Mater thesis |

Cocophony Mapper

Taking care of the sound level of intensive care

Graduation team
Student: Yoon Lee
Chair: Elif Ozcan
Mentor: Zoltan Rusak

Delft, May, 2019
Delft University of Technology
Master Integrated Product Design

Contents

Background

Project direction

Study

Idea generation

Realization

Evaluation

Acknowledgements

Elif,
Thanks for having me as a Critical Alarm Lab member and let me have a chance to re-frame myself as a scientific researcher for the first time. I really appreciate your trust that you showed in me and your guidance for this project as well as becoming a scientific design researcher.

Zoltan,
I am really grateful that I could have you as my mentor. You supported me in all different aspects and I could overcome many stumbling blocks that I came across with tons of consultations with you. Thanks for all your patience and your kindnesses.

My mom, dad, and auntie,
Thank you so much for all your support and your belief that you have shown in me. You all are the best supporters and the best family that I have had in my life and you will ever be.

My brother Ban,
Congrats on your marriage and I wish all the good luck to your newly-born family's happiness. I never thought I would follow through the paths in academia as you do, but I am glad now we have much more to talk about and communicate more often because of it.

Danielle
Thank you for being with me all the time and being such a sincere friend of mine. I had so much fun with you even though our work plans used to be changed into another fun day and I never regret it.

Ginny and Tim
Thanks for being a second family to me. I am so grateful for the time that we talked, studied and cooked together, and I really appreciate all your million-bucks-advice.

Jiyoung
Thank you for being my great support system all the time and your dropping-bys and suggestions. You are such a positive, caring, kind person and I am fortunate to have you here.

Jiwon
Thank you for giving me great advice all the time. I learn a lot from your positive mindset, and I really appreciate you for being such a good phase maker and a good counselor.

Veronika
Thank you for being a great work/lunch/coffee buddy all the time. I am glad that we could spend so much time together, supporting each other.

Nathalie
Thank you so much for being such a good listener and being a good counselor all the time. You are such a thoughtful person to talk to, and I really enjoy good talks with you.

Heidi
Congrats on your surprise-engagement with Ruben. I am really happy for you guys. Thanks for your amazing cooks all the time and keep up with your sense of humor which makes me cry all the time.

Jooyoung
Thanks for introducing me to Critical Alarm Lab, so I could eventually graduate in here. You inspires me a lot, finding your own ways and I am always learning a lot from you.

Grace and Dan
Thanks for being my Korean community here. It is already sad that you guys are going back soon, but I wish all the bests to you guys.

Deanne, Seanne, and Rose from TU Delft,
Thank you so much for your great help. I must say I was so lucky to have you for my graduation. You always went extra miles to polish the work and I really appreciate it.

Kyoungah and Soohyun,
Thank you so much for flying over all the way from Korea to the Netherlands to support me. All Korean food, booze that you brought here helped me a lot, getting through many difficult days. I must say you are the best things happened to me at work.

Hyesol
Thanks for being my best buddy for 13 years for now! Your nice messages and calls helped me a lot getting over homesick.

YooKyung
Thanks for being my support system for 13 years. Your sweet messages always cheer me up.

Youngmin
I am so lucky to have a friend like you who clicks together in a life-perspective. Thank you so much for your moral supports all the time.

Yoonwon
Your ever-positive mindset always teaches me a lot. Thanks for cheering me up all the time and sending me fun messages.

Summary

The starting point of this graduation project is noises that nurses experience in the ICU (Intensive Care Unit). There is a defined medical symptom which is called “alarm fatigue” that refers to numb auditory senses, stress, low job satisfaction as a result of being exposed to excessive noise for the extended period, and it often leads to low job performances in the end. One report reveals that 65% of significant ICU incidents occur by not responding to alarms appropriately (Cho, Kim, Lee & Cho, 2016). Since the alarm is not a single source of noises in the ICU, this report first defines which sound group will be seen as a culprit of “noise fatigue” as an expanded concept of alarm fatigue in this project.

The way this graduation deals with noise in the ICU is different from the conventional approach of providing a design intervention as a solution. In this project, the focus is rather how sound stimuli and stress level of nurses can be precisely captured from the perspective of nurses and be correlated altogether so that the new system can function as an investigation tool for further design intervention to improve the sound environment in the ICU.

In this regard, the sound classification method using machine learning, heart rate tracking as a means of an objective stress level assessment, and emotion report as a subjective stress evaluation tool was studied. Furthermore, beacons, an indoor positioning detection device, was investigated and combined with the project concept, so that Cacophony Mapper can function as a mobile data collection platform with dynamic data summary and analysis.

After the exploration of possible technology options for this project, the way of combining opted technologies was contemplated. As a result, two devices for heart rate tracking and sound collection, one mobile application for emotion report and data analysis, and a web platform which function as a hub for the data summary were developed as Cacophony Mapper prototype.

In the evaluation phase, the reliability of the sound filter was tested by directly putting the WAV sound data and a dataset collected through a user test, separately. Two datasets were compared to find the significant impact of the environmental factors so that the external impact can be minimized by further optimization. Also, the usability of the overall system and the application was evaluated. It has been done to see the possibility of implementing the Cacophony Mapper system in a medical environment and find which aspect of usability should be improved for further design development.

Introduction

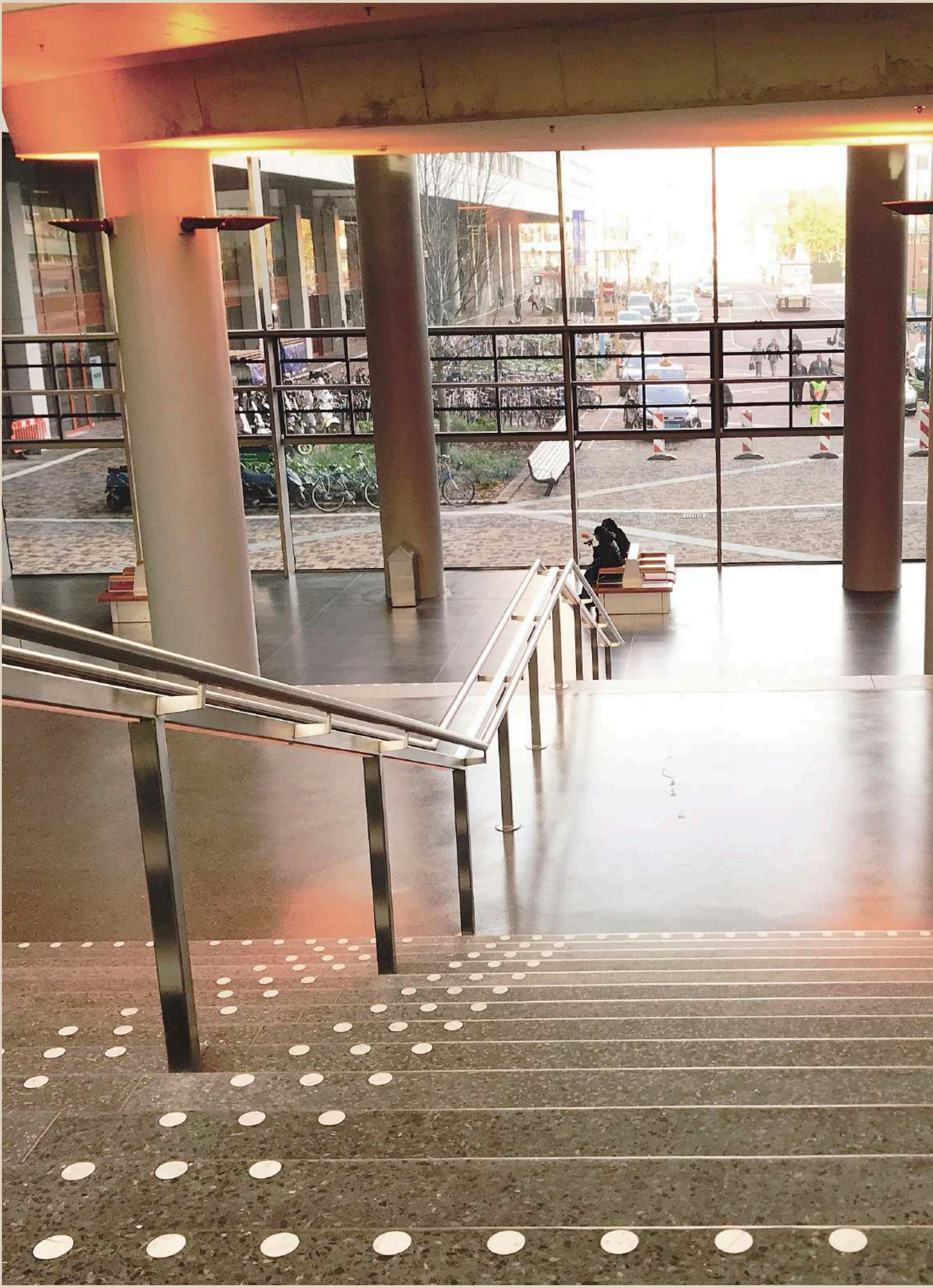
Intensive care units are acoustically hostile environments with high-tech medical devices and constant monitoring. As clinicians and patients have different types of involvement in the ICU, the emotions they feel for the same environment vary drastically (Farrell et al., 2005). For clinicians, taking good care of their patients and providing appropriate treatment is a top priority, so they pull their weight to perform their tasks efficiently as professionals. In this sense, the focus of clinicians tends to be upon productivity while they are doing their job and they often become indifferent about the surroundings or even the noise that they are making.

The soundscape in the ICU is somewhat complicated because it involves many stakeholders, situations and various sound sources, such as machinery sounds in the background, noise from speech, incidental sounds, as well as alarm sounds. Therefore, the ICU sound environment is accepted as a cacophony which is defined as a loud, unpleasant mixture of sounds and it makes people recognize the sound experience in the ICU something unpleasant and unharmonious. As cacophony turns sounds into pure noise and de-familiarize the whole concept of each sound with overlapping noise (McKee, 2006), the individual sounds often turn into a loud noise, so it impossible to distinguish a culprit of the unpleasant sound experience.

When it comes to nurses, their sound experience in the ICU can be even more severe as they are always surrounded by all different kinds of alarms from various pieces of machinery. As a result, nurses tend to lose sensitivity in their auditory ability as time goes by due to the alarm fatigue, which can lead to nurses' poor job performance in the end since alarms eventually fail to drag clinicians' attention. Also, regardless of how nurses try to give the best care for their patients or how good work ethics they have, they are sometimes unintentionally loud due to symptoms from the sound fatigue, and they do not seem aware how it will be accepted from patients and how it will affect their patients' recovery.

Therefore, a way of defining a noise fatigue, which is an expanded concept of an alarm fatigue, in the ICU has been contemplated and nurses has been chosen as a focus group for this graduation research as they have mobility, so the data collected by them is more interactive to the ICU surrounding, compared to static sensors attached to walls. Thus, this research project with the technical challenge will focus on collecting the sound data from surroundings from nurses' perspective, who are proactive stakeholders for patient's care with mobility for collecting sound data from all over the ICU, including ward, corridor as well as working stations for nurses.

Eventually, a platform for the sound analysis in ICU will be introduced to make a better awareness of overall sound produce in the ICU, which will enable further design interventions of noise care. It will not only help awareness of the criticality of the sound in the care unit but also trigger a further design development for behavior change, that leads to more pleasant sound experience in general. Therefore, a new culture of taking care of the noise in the ICU will be made, and it will contribute a lot for improving the quality of work environment for nurses as well as the general perception of patients and stakeholders about the ICU experience as a whole.



Project process

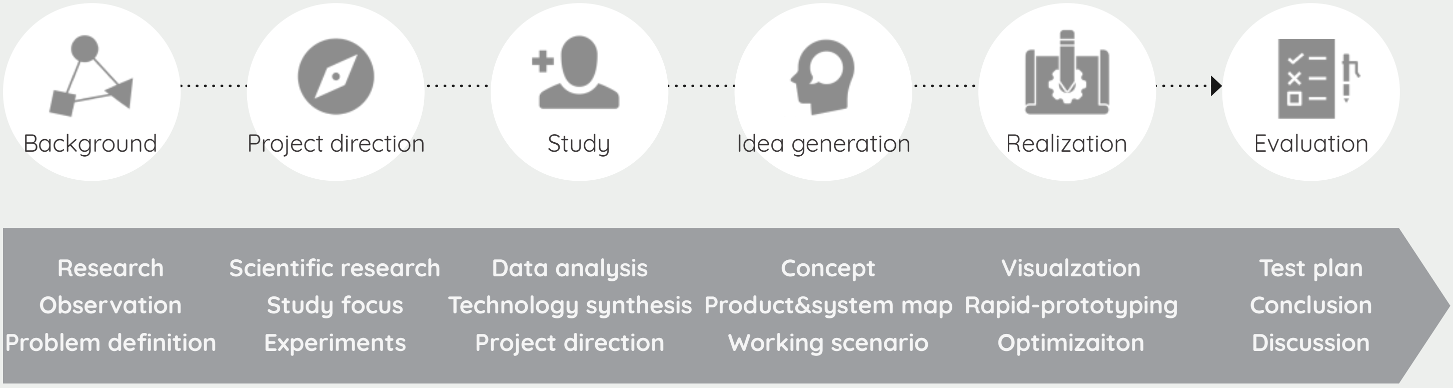


Figure 1. Process of the graduation project

The main focus of the graduation project is to:

- Design a sound sensor that is attached to nurses with mobility so they can collect sound information from the ICU
- Design an interactive platform to show collected sound data from nurses
- Design an interactive sound map that shows summarized sound data with more intuitive visuals

Background

To achieve these goals, I went through broadly six steps in my graduation process. To specify the problem, desktop research has been done, and observations were conducted in Erasmus MC in the first phase for the background research.

Project direction

After defining the main topic of this graduation project, scientific research was conducted as well as defining the focus of further study. In this process, three different studies were planned to set the specific project direction, regarding stress detection, sound classification, and indoor positioning and how those technologies are going to be intertwined in the end.

Study

Within three studies, I developed a deeper understanding of technologies which could be used for my project. In this phase, scientific research was mostly done, and various data was collected through heart rate tracker, microphone. Analysis methods were contemplated for the data analysis. Also, various physical and psychological stress analysis methods were studied, and sound filtering technologies have been tested to find out sound properties for the sound classification filter generation. As a result, project direction was settled in this stage, and it led to the idea generation phase.

Idea generation

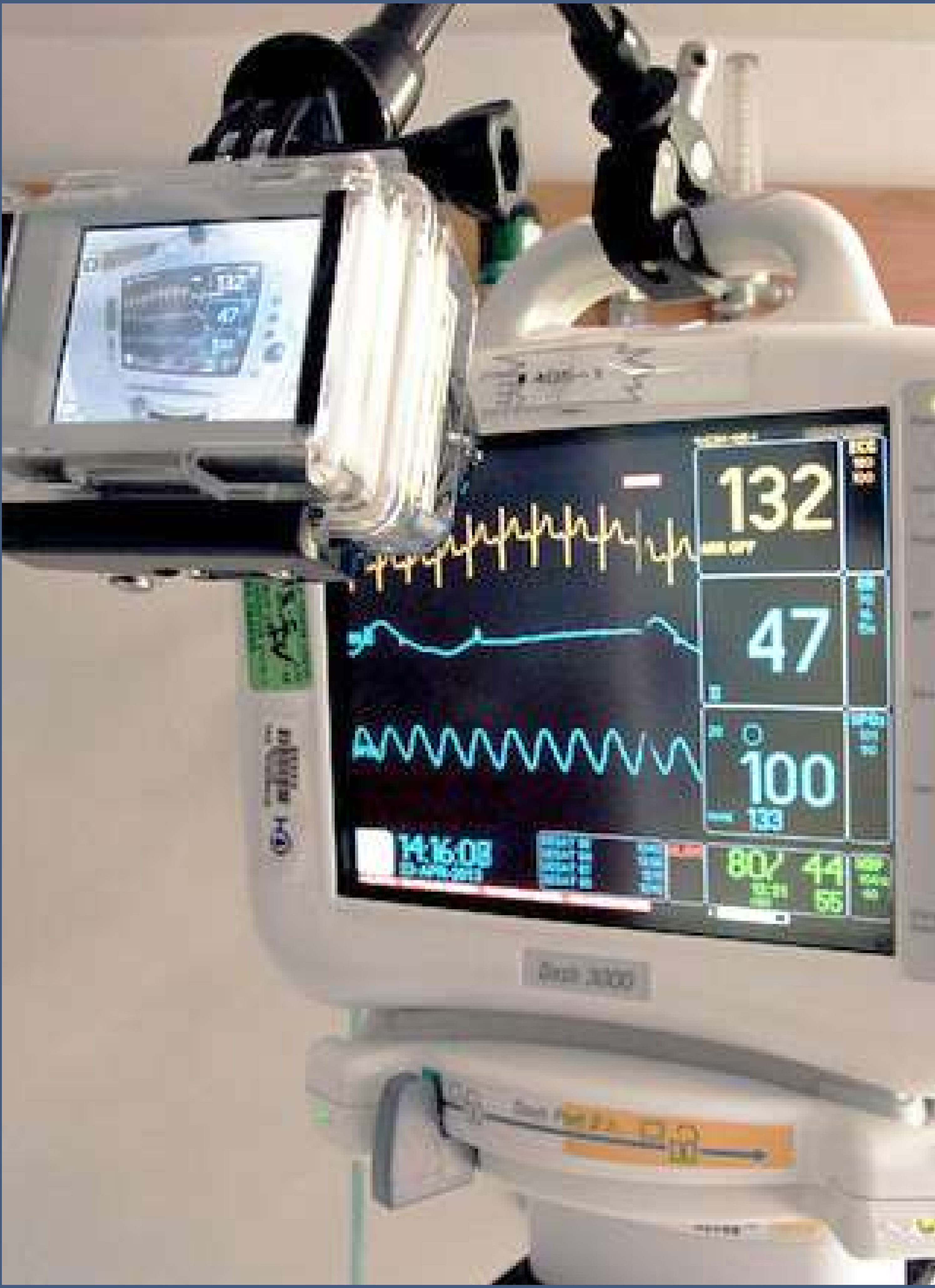
Through the idea generation process, the more robust concept of the project was figured out with a reliable product idea and a product & system architecture. Also, a system working scenario was created so I could have an overview of products and system which should be implemented and how they can be combined in one system.

Realization

In the realization phase, various forms and visuals have been created for the product and the application design. Also, an application structure map was created to have an overview of screens and various functions for the application implementation. When it comes to hardware, both aesthetic prototype and optimized prototype with better durability for testing were made. The second model was created with a rapid prototyping method, complimenting the shortage of 3D printing material and layered printing structure, which was found from the aesthetic prototype.

Evaluation

Finally, within the physical prototypes and an implemented application, test plans for reliability and usability of the system were made, and user tests have been conducted with students in the Industrial design department, TU Delft. All in all, evaluation has been done, and it led to a discussion for further development of an overall system for Cacophony Mapper.



Inhabitants in the ICU

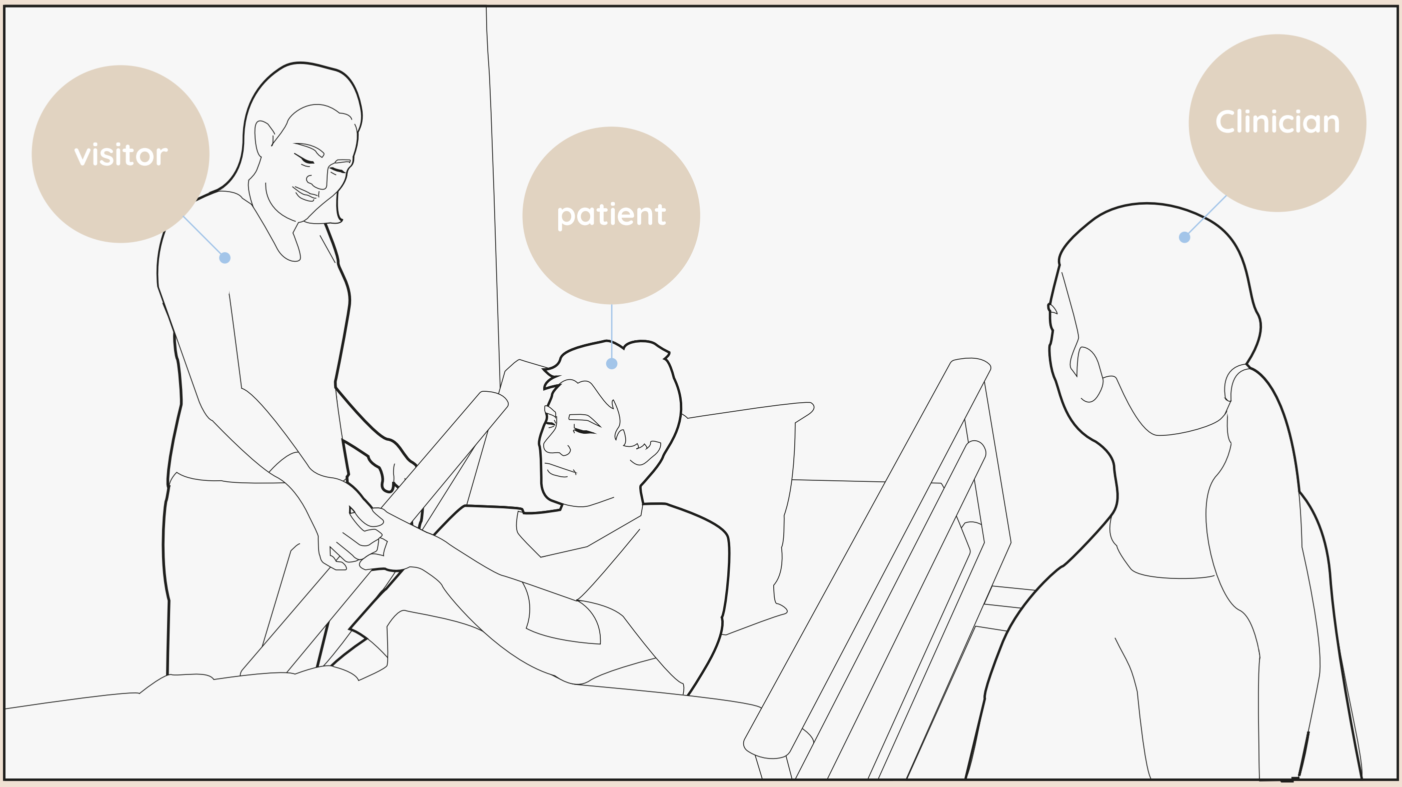


Figure 2. Major stakeholders in the ICU: patients, clinicians, and visitors

There are broadly three groups of inhabitants, clinicians, patients, and visitors, in the ICU. As the level of involvements, situational contexts and emotions that they face are different, those three groups of people happen to have all different perspectives and understanding about the same situation, and it goes for sound experiences in the ICU, too.

Clinicians

For clinicians, giving the best care for patients is their top priority. While they are working in the ICU, clinicians professionally perform tasks with rational mindsets, and they see situations with rather less emotional involvement than patients or patients' family members since clinicians cannot perceive patients' serious health condition as something tragic because those events happen for them on a daily basis while they perform their job.

Patients

For patients or visitors, being in the ICU is something traumatizing and desperate, as patients are hospitalized in the ICU when they are in their most inferior health conditions. Patients in the ICU often situated in their wards with severe injuries from significant incidents, after a big incident or a surgery. Often, patients in the ICU are even in a coma without any consciousness. Therefore, the priority of them is getting back to a healthy and normal life after receiving quality care, so their focus is taking a good rest with medical help and professional care of clinicians.

Visitors

ICU visitors find their experience in the ICU sad and desperate since they are usually family members or good friends of patients. Thus, they are sensitive about patients' physical conditions, and they put considerable importance on quality care. However, their access to patients or medical information is rather limited because the ICU aims for private and intensive care.

Why nurses?

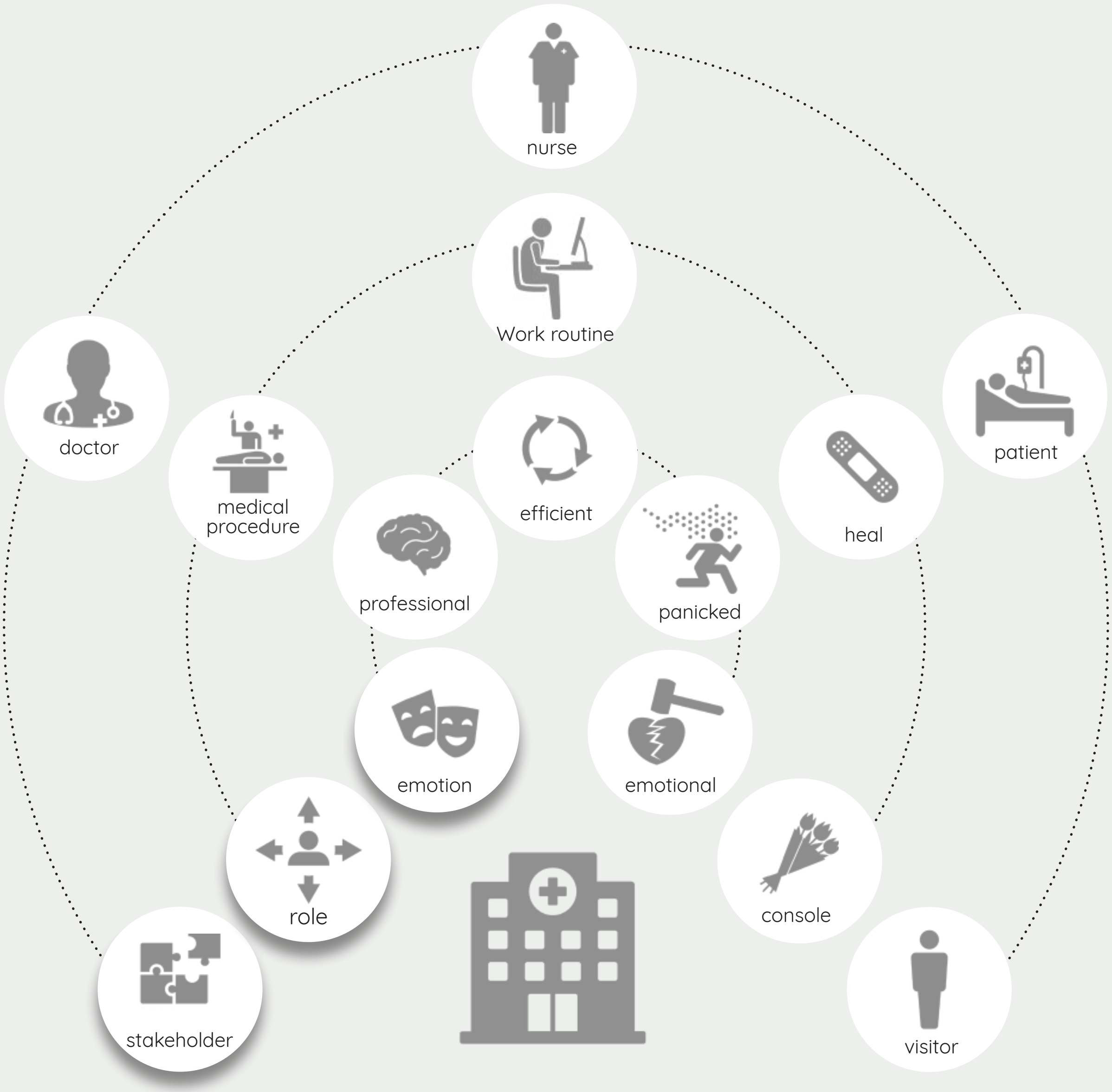


Figure 3. Different roles, different engagements of stakeholders

In this graduation project, the focus group has been chosen as nurses than other clinicians, patients, and visitors. In many cases, various design interventions were made from the perspective of patients, since they are the ones who are in the receiving end of the medical service.

However, as nurses are proactive inhabitants in the ICU environment with great involvement and significant impacts on the quality care of patients eventually, it is vital to form nurses' perspective. Since nurses pose a significant impact on the hospital atmosphere, the better sound experience of nurses can eventually form a positive loop of all inhabitants' experience, that lead to pleasant ICU experience as a whole.

Also, nurses are captive audiences of all sound stimuli in the unit since they work for there. They happen to be situated in a poor sound environment for an extended period than other inhabitants and they still need to perform their tasks as professionals while they are exposed to various noises coming from multiple sound sources, such as machinery, alarms, people's conversation, footsteps, door slam, or even a careless metal gadget dropping sound.

In this sense, this project will see the sound experience and their responses from the perspective of nurses. Since they are the ones with the most mobility in the unit, this approach will enable more conative sound analysis than conventional static sound collection methods. As nurses can collect flooding information while covering various situations in the unit, more valid information will be collected so that the collected data can be used for further design intervention for sound improvement.

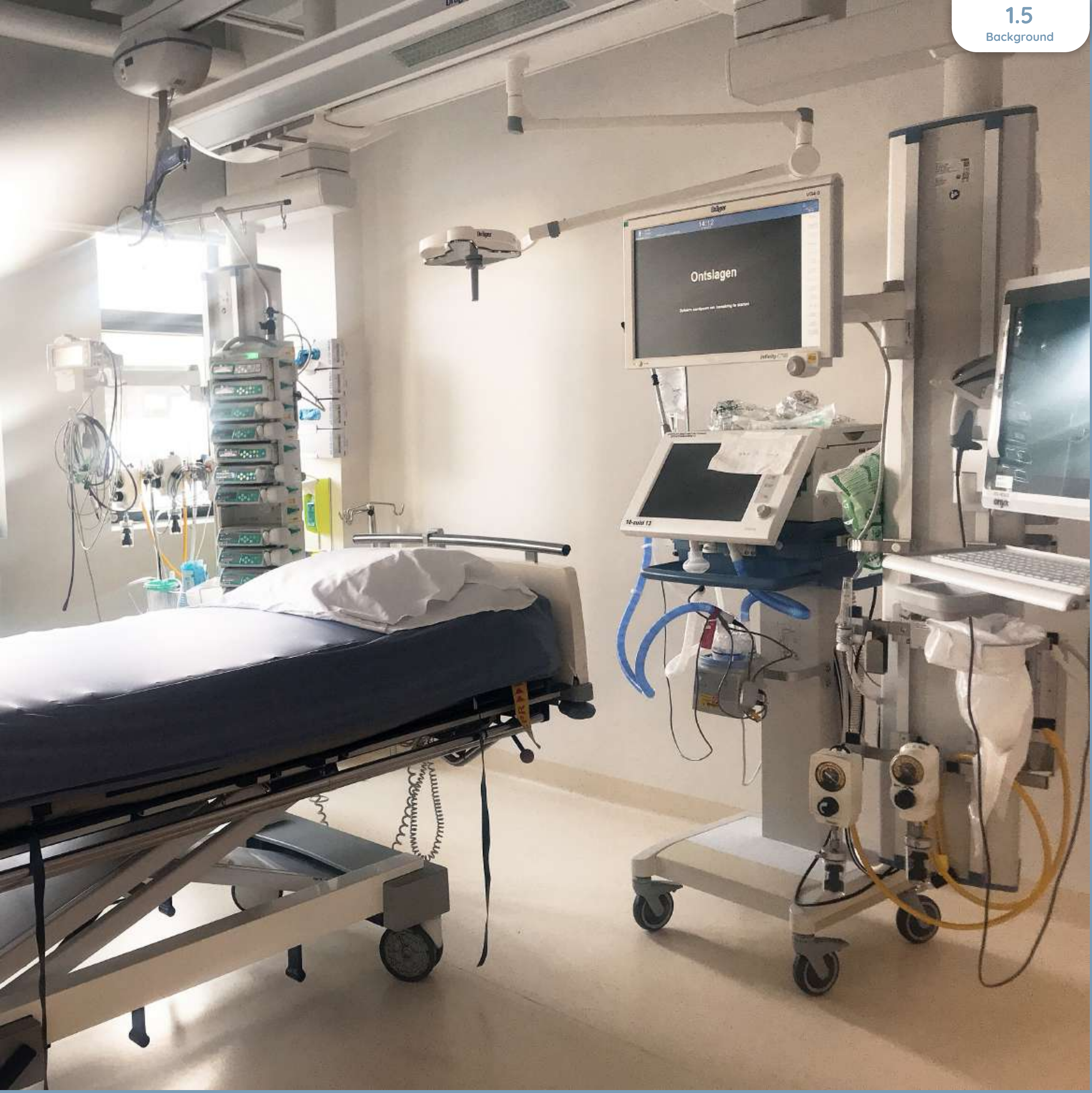


Figure 4. Erasmus MC ICU (Intensive Care Unit)

Intensive Care Unit Environment

There are two different types of the ICU, one with a private care unit which has nursing stations outside of each ward, and one with a shared room so nurses take care of all patients suffer from various symptoms all together (Konkani, Oakley & Penprase, 2014). The trend of ICU layout is changing from the latter type into the former type as private unit enables more personalized care with one on one care from clinicians with more privacy. Since the general ward is a big open area, everyone is exposed to all different stimulus from other patients, which vary from constant coughing or grunting noise to severe symptoms from patients with critical health conditions. Therefore, the healing of patients in the open type of ward can be highly dependent on each other's status.

In this project, the ICU in Erasmus MC has been observed, which has individual care units with nursing stations in the corridor. This type of layout has been introduced to Erasmus MC recently through the ICU renovation in 2017, due to increasing needs of individualized and private care. Also, it is expected that other big hospitals will follow through this global trend due to its effective operations and increasing needs of more private care. Therefore, this project will be done under the assumption that the individualized care unit will be widespread in the near future.

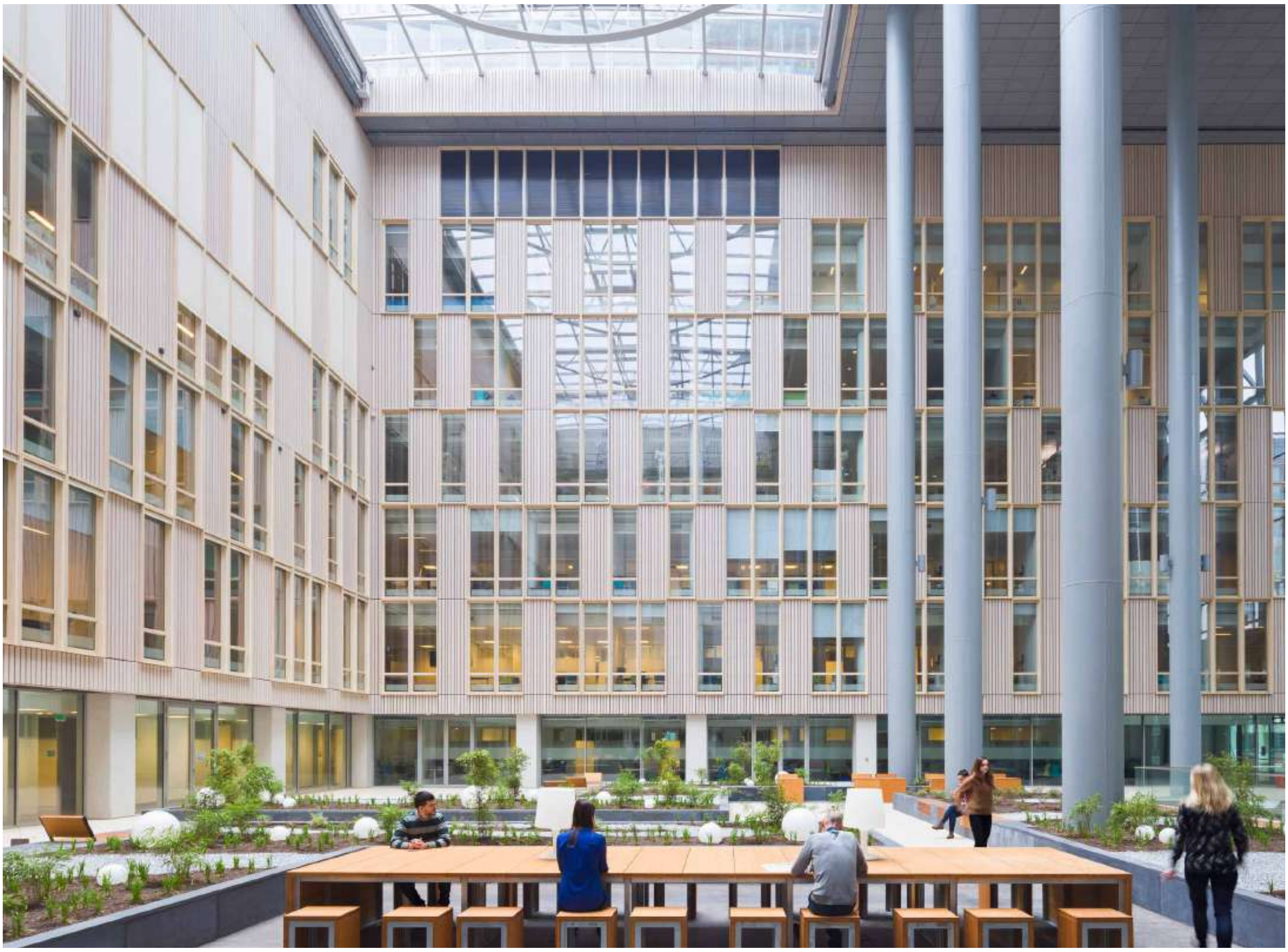


Figure 5. Renovated Erasmus MC

ICU in Erasmus MC

The ICU was introduced to Erasmus MC in the 60s, and Erasmus MC has been playing an essential role as a primary medical provider in the Netherlands. Before the emergence of the ICU, the distribution of patients was decided by patients' symptoms without consideration of their criticality level. However, with the introduction of the ICU, patients started to be hospitalized all together in the same room, which required clinicians to have a much broader knowledge of the medical area. Erasmus MC started to categorize patients by their symptoms so that nurses can give more specialized care afterward.

After the layout renovation in 2017, personal wards have been aligned in a row in the ICU, with nursing stations right outside in the corridor, which has been designated to two nurses in a pair. The aim of the new layout is giving one on one care to patients who are in critical conditions. Therefore, nurses stay in nursing stations, looking at patient monitors, which consistently indicate the patient's status. Also, nurses always carry a beeper, which gives continuous updates about the patient's needs and medical status.

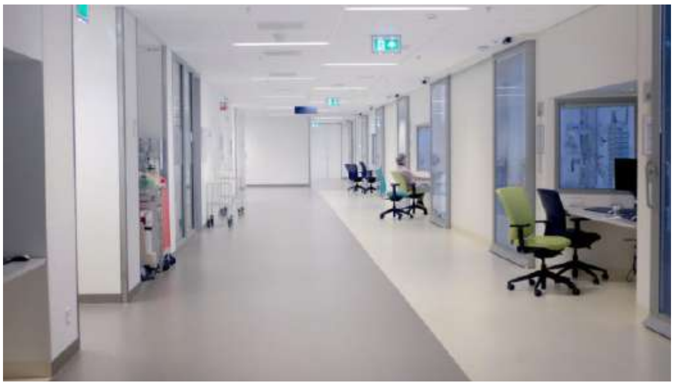


Figure 6. Erasmus MC ICU corridor and nursing stations

Soundscape in the ICU

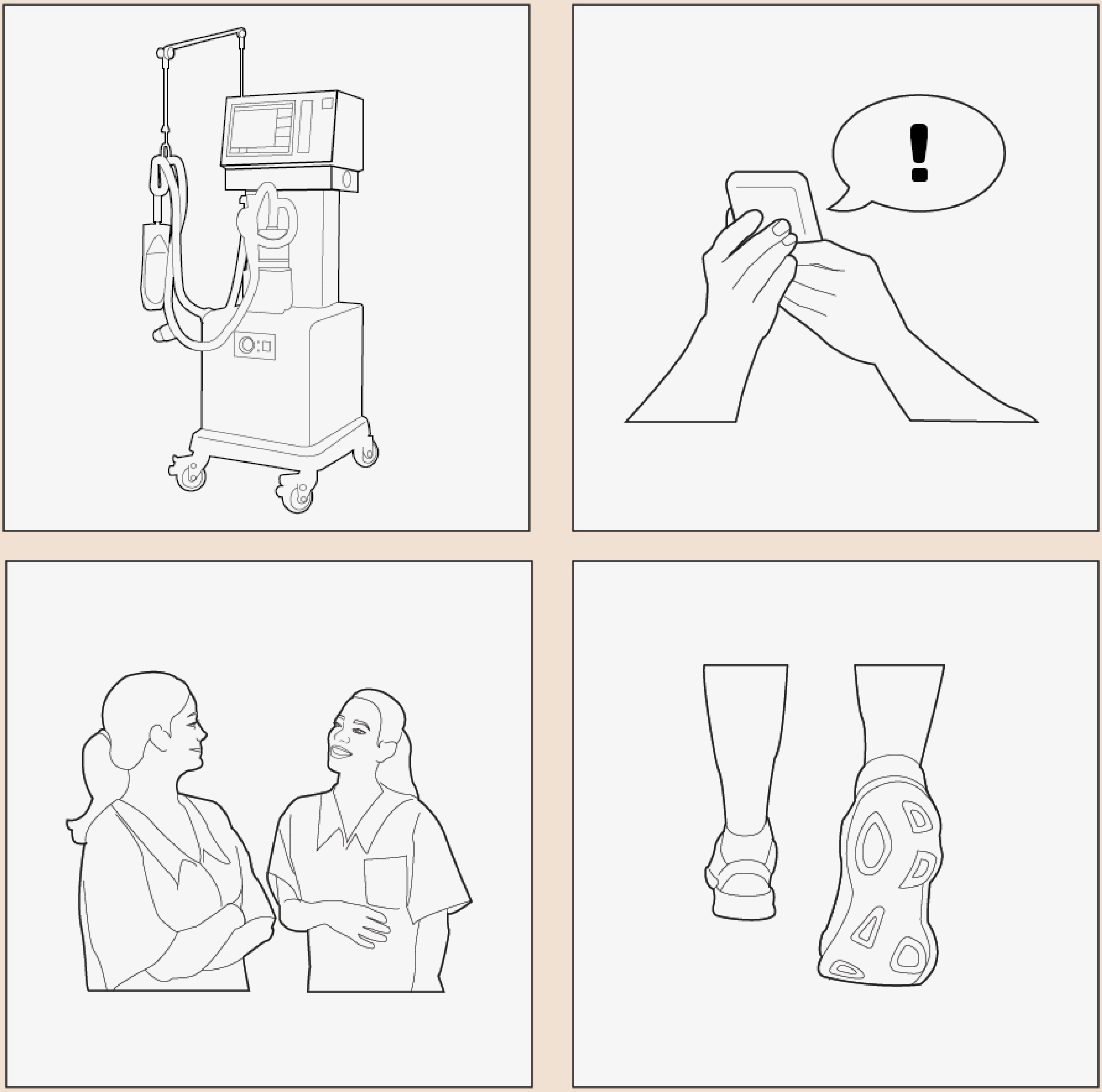


Figure 7. Major sounds in the ICU: (Clockwise from top left) machinery noise in background, alarms, conversational noise, incidental noise

As defined as unharmonious, unpleasant noise, cacophony is a mixture of noises from various sound sources. The soundscape of the ICU is often shown as a pure cacophony since people in the ICU easily feel unpleasant about the sound surroundings due to machinery noise in the background, conversations, alarms, and incidental sounds.

It is not easy to define the primary culprit of the cacophony since it is impossible to distinguish one sound source from others as the whole sound is absorbed as a chunk (Alain, 2007). In this sense, individual sound sources in the ICU has been studied, and they are categorized into four different groups, as shown in the figure above.

As one of the purposes of this graduation project is to understand sound surroundings in the ICU and give a reliable visual indication about it, the soundscape has been observed as a whole. A summary of various sound sources in the ICU and the sound categorization idea will be elaborated in chapter 2.2 and 3.9

Various alarms in the ICU

According to a research done by Drew et al. in 2014, alarms consist of three different sorts of beeps, which is composed of patient status arrhythmia alarms, patient status parameter limit violation alarms, and system status technical alarms (Drew et al., 2014).

Figure 8. Various alarms from diverse warning situation

Patient status arrhythmia alarms

Patient status arrhythmia alarms make three beeps continuously when there is a warning. It indicates that the patient is in a crisis level while the machine makes two beeps for advisory and one beep for the message or inaudible text. For example, asystole, ventricular fibrillation, ventricular tachycardia, and ventricular bradycardia is considered as crises. Accelerated ventricular rhythm, pause, gives off a warning sign, while bigeminy and trigeminy deliver messages of patients' parameters with one beep.

Patient status parameter limit violation alarm goes off selectively, and it indicates there is a warning. In this status, an advisory message will be given with two beeps while one beep shows a message or inaudible text. Those alarms include heart rate, invasive arterial pressure, and respiratory rate abnormality, while both noninvasive blood pressure and peripheral oxygen saturation gives advisory alerts.

System status technical alarms

System status technical alarms give a warning, a message, or an inaudible text selectively. Those alarms include warning sound, which indicates ECG leads fail, respiratory leads fail, arrhythmia suspend, invasive pressure sensor fails, failures or excessive pressure of noninvasive blood, while message alarms are given for various artifacts and lead fail with continuous foghorn tone.

Alarm fatigue

What is alarm fatigue?

The alarm fatigue is one of the primary starting points of this project as it poses one of the most significant occupational hazards that clinicians in the ICU suffer from. One report showed that 46% of nurses responded that alarm fatigue is the major struggles in the ICU environment in terms of their job performances (Konkani, Oakley & Penprase, 2014). Alarm fatigue is a symptom that nurses and clinicians become numb about the sound stimuli, mainly because of numerous alarms from various medical machinery in the unit. Even though the alarm is designed to give off beeping noises for nurses to be aware of the medical indication of their patients, nurses tend not to pay that much attention to them as much they should do in real situations.

What are the leading causes of alarm fatigue?

One of the main reason for increasing alarm fatigue is that the number and sorts of alarms from devices have been increased. The sort of the alarm increased from 6 in 1983 to 40 in 2011, so the frequency of the alerts is significantly higher than that past 20 years (Cho, Kim, Lee & Cho, 2016). Not only because of an increased number of beeping sounds, but many studies have also shown that alarms are often considered as something minor from clinicians because of the high rate of false alarms, which brings cry wolf effect (Cho, Kim, Lee & Cho, 2016). After showing attention to alarms several times and realize beeping was nothing crucial, perceived priority level of alarms decreases, so they are not keen on answering to them later on. Also, a study shows that excessive amount of non-actionable alarms makes nurses lose concentration on sound alerts, so nurses are desensitized due to sensory .overload (Salous et al., 2017) on their auditory ability, which results in ignoring or delaying in responses to alarms. All in all, it is shown that the increase in the number of alarms, false alarms, and non-actionable alarms are the leading three causes of alarm fatigue in the ICU.

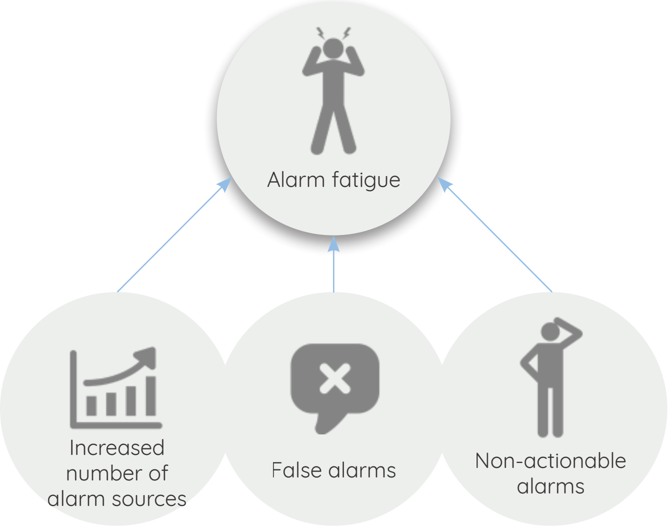


Figure 9. Main causes of alarm fatigue

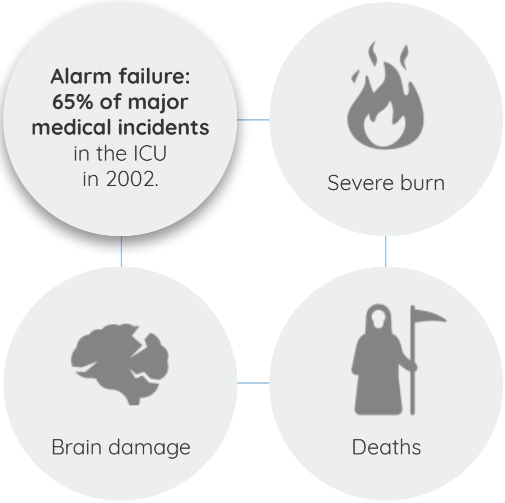

Alarm failure: 65% of major medical incidents in the ICU in 2002.
Severe burn
Brain damage
Deaths

Figure 10. Alarm failure incidents in the ICU

Criticality of alarm fatigue in the ICU

The impact of alarm fatigue is severe because the ICU is a place for patients who are in critical health conditions. Even though the majority of alarms are false alarms and non-actionable alarms, there are still calls for patient's critical health status. However, there are many hazardous cases reported as a result of alarm fatigue. A study states that 65 percent of major medical incidents in the ICU in 2002, was triggered because clinicians did not respond to alarms appropriately, which resulted in the severe burn, brain damage and deaths in extreme cases (Cho, Kim, Lee & Cho, 2016). It is because alarm signals can be hardly distinguishable from each other as only 31% of nurses reported that they could distinguish one alarm from others (Cho, Kim, Lee & Cho, 2016). Therefore, clinicians often misunderstand important cues as something trivial, so they ignore critical signals from the medical device.

Also, alarm fatigue is one of the significant reasons that nurses have lower attention to alarms. Being exposed to excessive alarms and sounds for the extensive period, clinicians happen to have poor physical and psychological health, which lead low job performance and low engagements to their job, forming negative loops in quality of the medical service.

Physical and psychological impacts of alarm fatigue

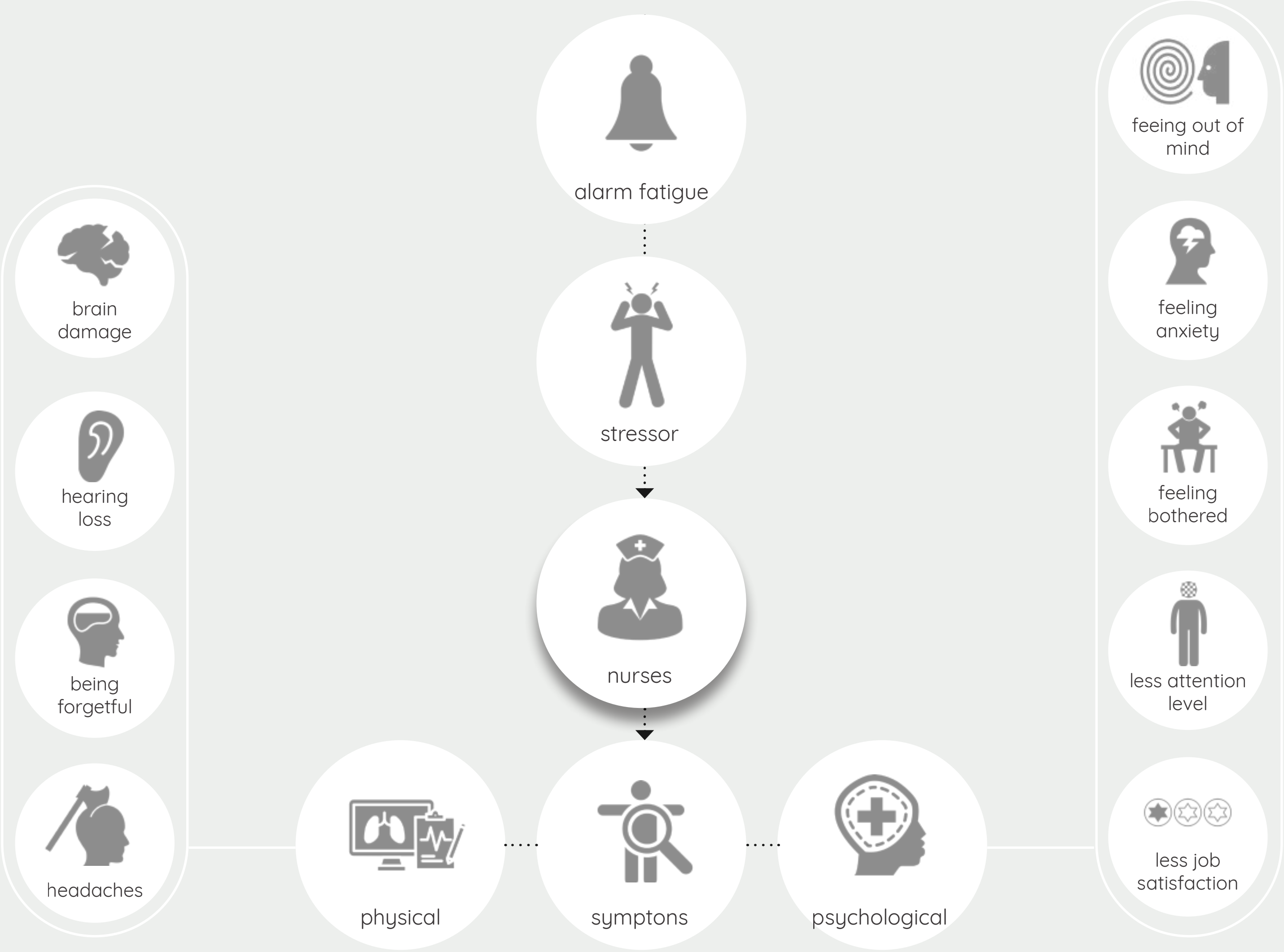


Figure 11. Physical and psychological symptoms of alarm fatigue

Symptoms of alarm fatigue

Physical and psychological impacts of alarm fatigue were studied by Cho et al. in 2016. Through the questionnaire in the research, responses have been collected from 77 nurses with 5-point scale answers. Nurses' perspective about clinical alarms, mostly related to emotional fatigue, was the main focus of this questionnaire and nurses were asked to scale eight sentences from 1 to 5, corresponding to their feelings about alarms.

As a result, "feeling bothered in everything by clinical alarms" rated 3.9, with 0.8 of standard deviation. Also, "feeling anxiety" rated 3.7 while "feeling out of my mind" recorded 3.6, rating 0.8 and 0.9 each for SD, respectively. "Having trouble paying attention" was the following symptom, rating 3.3 points and "being forgetful" recorded 3.2 with each 0.8 of SD. Furthermore, "feeling bad" and "having headaches" gained scored 3.1 of each, showing the various psychological and emotional impact of alarm fatigue that nurses are fighting against (Cho, Kim, Lee & Cho, 2016).

Previous sound improvement approaches

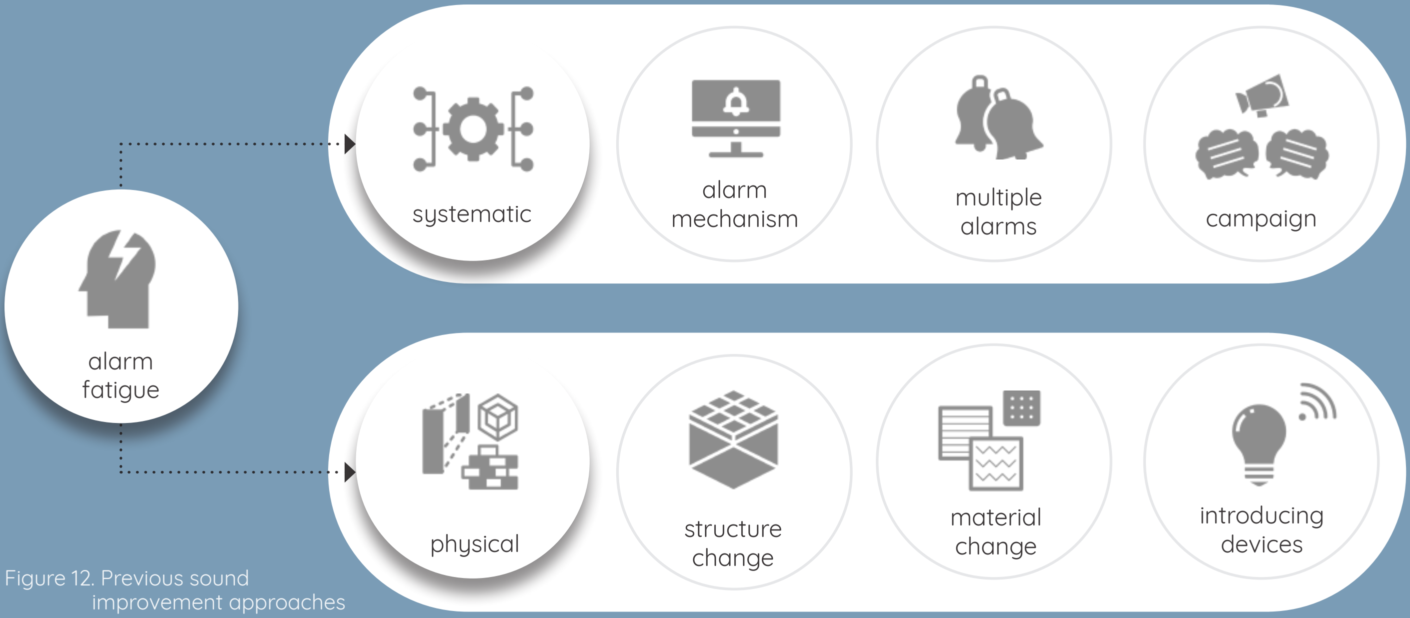


Figure 12. Previous sound improvement approaches

Most previous studies regarding sound level in the ICU have been focusing on defining the sound level in the ICU and find the solution to reduce the level. Therefore, diverse methods, which vary from introducing the device to making a systematic change, have been implemented (Konkani, Oakley & Penprase, 2014).

- introducing lighting device for excessive noise
- switching alarm mechanism
- closing door campaign, using earplug and earmuff
- introducing individualized alarms
- introducing new building materials
- changing ceiling structure and shape

Even though various ideas have been tried and tested, it has been proven that the impact of the change is trivial, or the validity of the impact was disappeared in a certain period after the design intervention. Also, some solutions were not feasible to implement due to the high budget requirements and practicality in the real environment. Additionally, since most research has been done from the perspective of patients, it is hard to apply the same design for clinicians as a design intervention for them.

Why is noise in the ICU problematic?

According to a previous study, the noise level of hospital, in general, has been increased from the past 50 years (Stafford, Haverland & Bridges, 2014), which exceeds the healthy sound criteria of WHO. Even though WHO, which is an organization to give international guidance regarding health and global medical approaches, recommends daytime sound level in the hospital as 45dB and night hour level as 35dB (Stafford, Haverland & Bridges, 2014), the sound level in the ICU is reported to exceed 80dB (Cho, Kim, Lee & Cho, 2016). The value is almost the twice the recommended figure and ICU sound environment is even more severe than the hospital in general(Khademi et al., 2011).

It poses physical and psychologocial threat to inhabitants in the ICU with tremendous impacts. When it comes to patients, they suffer from sleep deprivation due to the noise in the ICU with their immobile physical condition (Stafford, Haverland & Bridges, 2014), which lead to depression and post-traumatic syndromes even after the recovery from their primary symptoms. For nurses, the influence can be even more significant as the ICU is their work environment, so they happen to be exposed to the noise for a more extended period, which often lead to chronic physical and psychological symptoms. The severe sound experience in the ICU can lead to less job satisfaction, hearing loss and even pose health threat to clinicians (Drew et al., 2014).

Figure 13. Noise in the ICU: threats for both nurses and patients

Noise categorization

Primary noise sources in the ICU has been defined as four sound categories, which is machinery noise in the background, alarms, conversational noise, and incidental noise.

Machinery noise in the background

As ICU is filled with a lot of medical pieces of machinery, it is not too rare to listen to the sound of the electronic ventilator as background noise, continuously pumping air to patients that rely on their breathing. Also, the sound of air-conditioning, the sound of a coffee machine can also be a part of this category. As the sound is not too loud and something constant like a piece of background music, people tend to be indifferent about the sound. It is the same as when we become almost numb about white noises that we encounter every day because we only accept it as our surroundings and do not deeply engage in the sound.

Alarms

An alarm is one of the major noises sources in the ICU. Sudden beeping sound often makes people get annoyed as it startles people, even though the sound was designed to make a warning. It often triggers annoyance because some nurses respond to alarm slow, so it often takes some time to go off. Also, the sound level and consistency often do not meet the importance that alarms are carrying. Therefore, people do not find alarms not “alarming” any longer in many cases, and people even often feel fatigues out of them (Otenio et al., 2007). This can lead to serious ramification because many patients’ health is dependent on those machines, while people become less sensitive about the alarm it produces because of the alarm fatigue.

Conversational noise

Speech is usually made among clinicians as an ICU environment is highly protective and private. Also, the accessibility to the unit is relatively limited, so visitors cannot be in the unit except for the official visiting hours. To make sure patients’ medical status, nurses always need to stand by in front of the patient rooms, sitting in front of the private unit as a pair, while looking at the check-up monitor. In most cases, nurses try to keep themselves silent, but one of the problems is that the sound production in the unit is highly dependent on themselves since no one would dare to point out someone is loud. As a quiet environment is expected in the ICU, patients get annoyance once the noise from speech occurs, and they often suffer from the speech that their peers make.

Incidental noise

ICU can be either relatively silent or noisy, as the situation inside of ICU can vary wildly. When a patient with a severe condition comes to the unit, the sound environment can also be very hectic. In this case, the general sound level will go high up with the patient’s rolling bed carelessly coming into the room, noises from many clinicians’ running, as well as careless door slams. Since each event is hardly expected and the pattern of the incidental sounds is irregular, it is usually considered as a one-time event even though inhabitants in the ICU are irritated by the sound.

Figure 14. Noise category:
(from top to bottom) background noise, alarms, conversational noise, incidental noise

Project focus: Noise fatigue

The main focus of the study has been decided as noise fatigue in the ICU, which is a broadened concept of alarm fatigue. The difference of two is that the subject of the noise fatigue covers overall sound in the ICU while that of alarm fatigue is only confined to alarm noises. Even though there has been a lot studied which defines alarm fatigue of nurses, an alarm is not only a severe sound factor which contributes to extreme sound experiences of nurses in the ICU. In this sense, this study will provide a solid foundation to study noise fatigue for further design interventions by providing a sound sensing technology and the data analysis platform. Also, by using the Cacophony Mapper system, nurses will be able to aware of the sound environment that they are exposed to and start to think about the sound that they produce by themselves. It means that this project can contribute to forming a pleasant sound culture in the ICU.

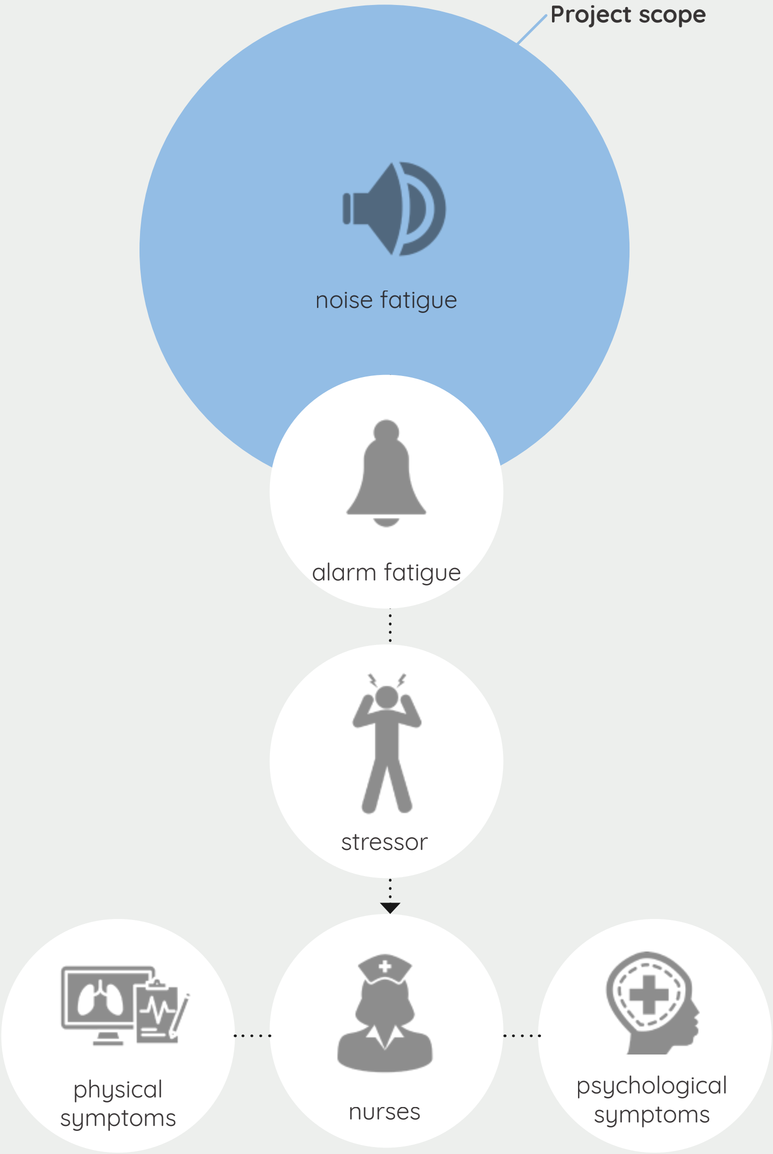


Figure 15. Physical and psychological symptoms of alarm fatigue

Noise fatigue evaluation

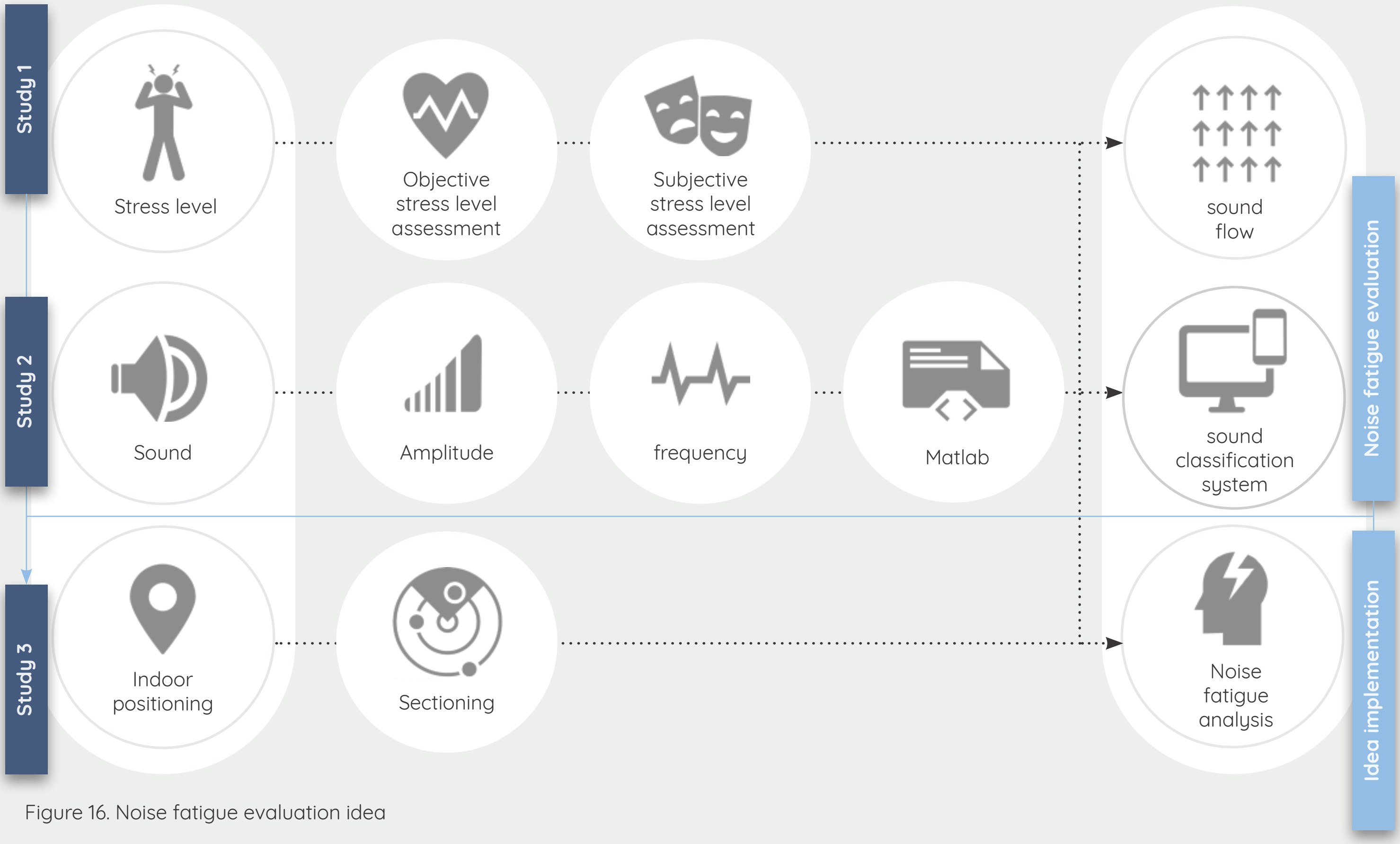


Figure 16. Noise fatigue evaluation idea

Monitoring fatigue

As one of the primary goals of this research is to find a correlation between hectic auditory environment and the noise fatigue that nurses experience in the ICU, it is vital to find a measure to check the fatigue level of nurses which can be linked to the sound level fluctuation in the environment. To measure the fatigue level of nurses, subjective and objective measures which are widely used to check the fatigue level will be studied and interrelation between nurses' fatigue swings and acoustic flow in the unit will be assessed in the later phases.

The first part of the study will deal with how fatigue can be quantified so the stress data can be correlated to the sound analysis in the end. Thus, the first part of the study was planned as a study for objective and subjective stress level assessment separately, and the study process and insights will be available from chapter 3.1 to 3.7.

Secondly, sound properties will be analyzed, and a sound classification filter will be created with the gained knowledge about the ICU sound. This filter will eventually enable the sound categorization, and the whole study and filter development process will be available from chapter 3.8 to 3.13.

Finally, there will be a study of indoor positioning, which will enable sound flow analysis in line with the previous two studies. This technology will enable real-time sound heat map function, and the concept will be available from chapter 3.14 to 3.16.

Design vision and program of requirements

To set up the clear project goal and to plan what should be studied through this graduation project, design vision, and program of requirements have been contemplated and was made into a list.

Design vision

- Creating a tool which can validate noise fatigue in the ICU
- Creating a sound and stress level tracking sensor
- Creating a system which clearly shows the correlation between sound stimuli and stress
- Creating a platform with intuitive visual which nurses can quickly learn to use
- Creating a system which makes nurses think of the sound the environment in the ICU

Program of requirements

- The system collects sound and stress information from the perspective of nurses
- The system collects dynamic information through a device attached to nurses, and collected dataset should function as an integrated information
- The system should be able to classify the sound category
- The system should be able to detect nurses' stress level
- The system should not hinder nurses' work routine
- The way of using the system should be intuitive to understand
- The system should be easy to learn
- The design of the system should follow the hospital regulation
- The design of the system should show a clear relation with the medical environment
- The system should be used continuously for a long time

Objective assessment of stress

Various objective stress level assessment tools were investigated to find the method which can be used for Cacophony Mapper system development.

VAS (Visual analog scales)

Visual analog scales measure the fatigue level by marking the accuracy level of respondents' answers to see the changes in mood and activation level of test participants (Monk, 1989). Respondents are asked to mark the length of the figure, which is given each time differently, and the correct level will be traced over the time to see the flow of the concentration level. As this is a straightforward method for both questioner and respondents, this test is easy to be taken and also the further processing for the data does not take too long because the scale for the analysis is already given in numbers.

Resting heart rate

Collecting the resting heart rate(RHR) is an excellent way to keep track of fatigue level of nurses because there is an intimate connection between heart and human brain (Thayer et al., 2012). After collecting average pulses from neck, chin, or wrist three times every day, collected numbers are compared. To collect the precise resting heart rate, morning hour is usually ideal for getting reliable data for the comparison. If resting heart rate shows 7 or more beats differences than the average per minute, it means that the test participant did not fully recover from the previous fatigue. As pulse rate collection can be done very fast and easy, it is a quick and easy way of collecting fatigue level. Also, as it indicates a physical fatigue level with numbers, processing the dataset can be quickly done.

Measuring eye movements

Measuring eye movements is meaningful in the way that it shows constant data change about the fatigue and concentration level. Many studies show that eye blinking patterns differ depending on the fatigue level. For example, if a person focuses on one object or stares one place for an extended period than the average, it means that the concentration level is low (Caffier et al., 2003). Even though tracking eye movements look simple in experiments, it can be tricky to use this technique, combining with a daily routine. It is because the camera should be around participants' eye area during the whole day for consistent measurement, so the appliance of the hardware is confined to glasses or headgears.

Saliva measures

Saliva measure can be a beneficial tool because it is quick, easy, and painless to use. This method is based on Hyperion's study in 2012, and it gives a precise chemical data related to the stress levels, such as cortisol, dehydroepiandrosterone, testosterone, chromogranin A, 3-methoxy-4-hydroxyphenyglycol, alpha-amylase, and secretory immunoglobulin A (IZAWA et al., 2007). However, even though it gives a precise dataset, it is not easy to use saliva measures as a tool for constant stress tracking, because the usage of it is not common and the measure is hard to be positioned in users' daily rituals.

Figure 17. EEG: an example of objective assessment of stress

EEG (Electroencephalogram)

EEG is a device which is widely used to measure the fatigue and cognition level of people using brain wave, which shows relatively precise result comparing to other methods. EEG is meaningful in the way it enables qualitative and quantitative evaluation at the same time, using signals acquired from brain activity (Hosseini, Khalilzadeh & Changiz, 2010). For the measurement, the EEG device is attached to the head or forehead from one to several spots and sense patterns of brain activity. With technological advancements, EEG is not considered as a high-tech device anymore so that the product line-up can differ from a widely used home device for concentration level tracking to a professional gadget in big hospitals.

After the investigation, heart rate measurement was chosen as an objective stress level assessment tool. Since I did not want to have a questionnaire as an objective stress level assessment tool so the measurement method can be differentiated than subjective stress level assessment tool, VAS was excluded from the option. When considering the practicality, wearing a big measurement device for EEG or an eye movement, tracker did not seem to be feasible in the ICU environment since it can hinder nurses' daily activities. Also, taking saliva samples several times a day was against the program of requirements on the previous page that the system should not hinder nurses' work routine. All in all, heart rate tracker was selected as a measurement tool, and further study of heart rate detection will be available from the next page.

Heart rate detection tools

Heart rate tracker: an objective fatigue measurement tool

Heart rate tracker was chosen as a tool for objective stress assessment because of its easy accessibility and high applicability for the system. When looking at its role in the market, heart rate sensors have been widely adopted in many smartwatches and health trackers. In terms of heart rate detection mechanism, they follow two different methods, one for using LED lights and the other for using electrical current.

Photoplethysmography (PPG)

Photoplethysmography(PPG) is the most common measurement for the heart rate tracking using LED sensor, which has also been adopted in Apple watch and Fitbit, that gives more accurate test results than Bioelectrical impedance analysis (BIA) method and enables continuous measurement with incessant physical movements. The basic principle is emitting green and red lights through the wrist band and check the number of each light reflected or absorbed on the skin. As blood, which has a red color, reflects red lights while it absorbs green lights, the amount of absorbed green light is checked hundreds of times for one second, so it shows user's heartbeat per minute by detecting the expansion and contraction of the blood vessels. As this method uses light emission and absorbance as its primary mechanism, tattoo or any disturbance for light can affect the quality of the sensing. Also, severe movements can affect the test quality, too.

Bioelectrical impedance analysis (BIA)

Unlike Apple watch and Fitbit, Jawbone uses electrical current as its heartbeat measurement mechanism. BIA is more commonly used for measuring body composition than heart rate since it only collects sitting heart rate unlike PPG, so it is not appropriate for measurement with a lot of activities. Even though BIA is a widespread method in the medical field and it has its benefit of efficient battery use, Kyle points out in his previous research that BIA still lacks standardized method and quality control procedures (Kyle, 2004).

Fitbit Alta HR

After the literature review, a PPG sensor was chosen for heart rate detection for Cacophony Mapper. Fitbit Alta HR, a sports watch which has launched geared toward heart rate detection, has been chosen for further testing. As using a wrist band is banned for nurses while they are at work because of the possibility of cross-infection, different placement scenarios for Fitbit were made, and pros and cons of positioning the tracker on each spot were compared.

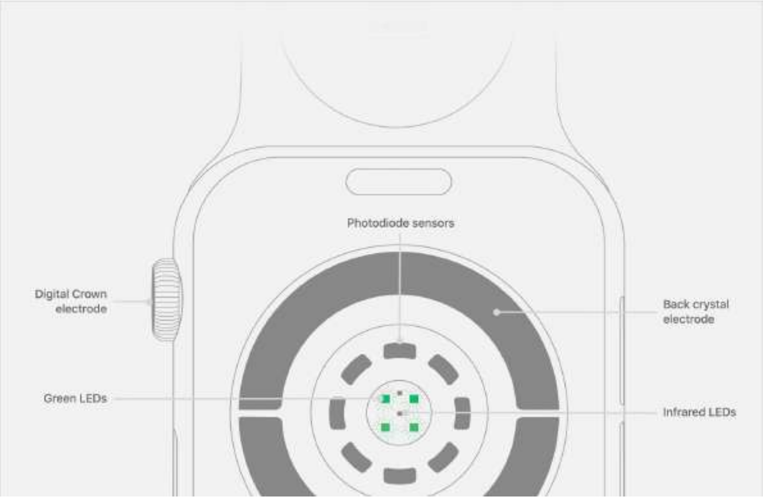


Figure 18. heart rate detection method with PPG sensor

Figure 19. heart rate detection method with BIA sensor

Figure 20. Fitbit Alta HR

Placement of heartrate tracker

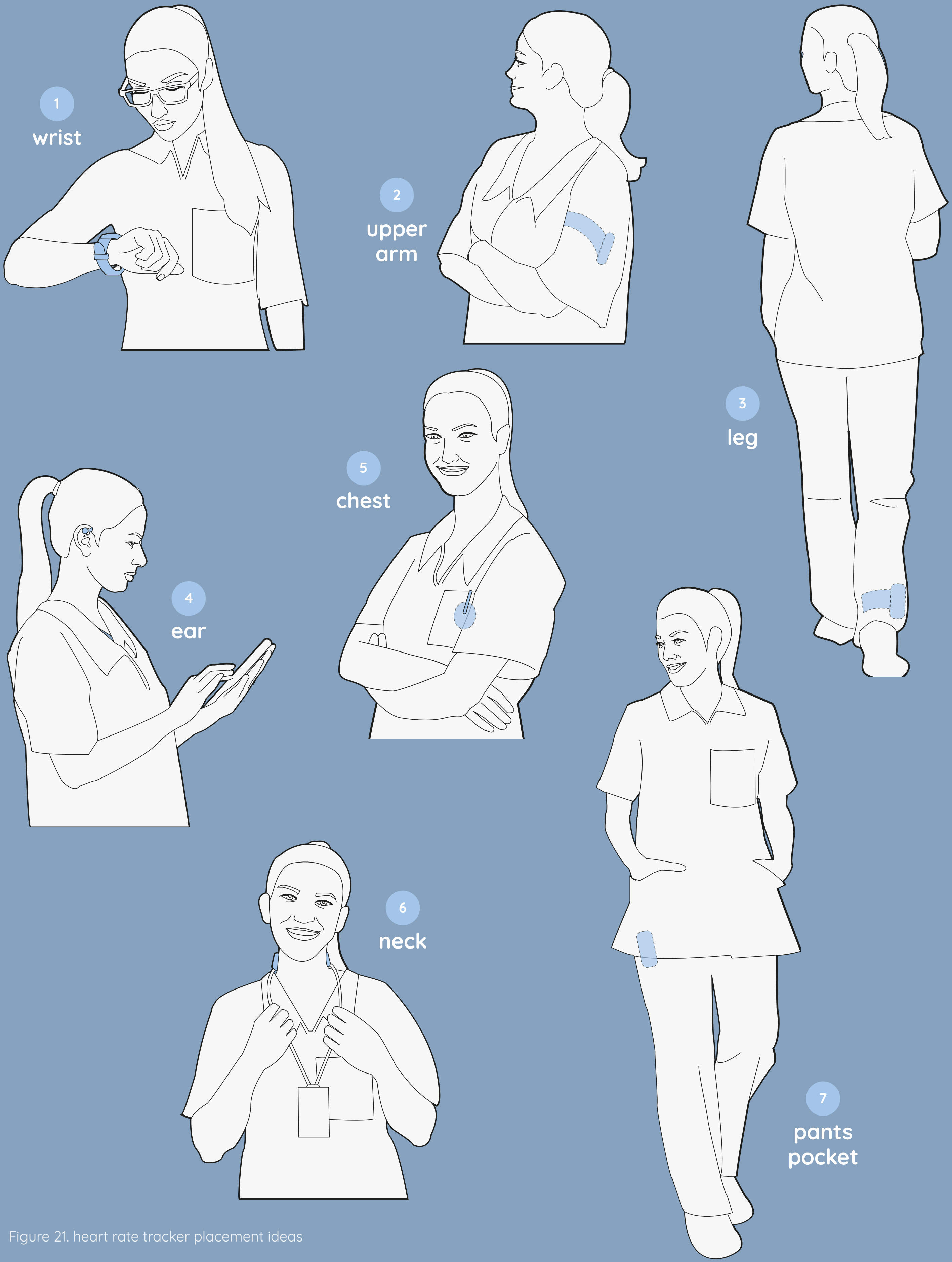


Figure 21. heart rate tracker placement ideas

1 wrist

The wrist is a common area for putting health trackers because it is an ideal place for positioning an interface and collect the heart rate data at the same time, using the same device. Apple watch, Samsung Galaxy watch, Fitbit, Shaomi Mi band are significant players in the digital health tracker market, and they have designated applications for the heart rate fluctuation analysis so that users can keep track of their activities, as well as sleep quality tracking and steps tracking function. However, wearing a wrist band as a nurse in the ICU is against the hospital regulation because of the possible cross-infection among patients. Additionally, the wrist is not the most accurate place for detection, since the area involves many activities from constant hand movements. Also, when a device covers the wrist too tight, then it can hinder bloodstreams, which contradicts the accuracy of the sensing.

2 upper arm

In many cases, the upper arm is considered as a body part which can substitute the use of wrist, giving almost the same usability with better accuracy. Since wearing a device on the upper arm area does not hinder user activity as much as other areas, the upper arm provides precise tracking result. Because of its perks, it is easy to find sports armbands that are launched geared toward especially the upper arm area.

3 leg

Even though the mass of the leg consists of the majority of our body, it is not an ideal place for the heart rate detection since the blood perfusion is lower in leg comparing to other body parts. It is because of the distance from the heart and its dense muscle tissue composition in the thigh and calf area (Wearable, 2018). Also, since the area involves lots of movements, it can hinder accurate detection. All in all, leg area has been excluded from the user scenario because of expected low acceptance level since the placement of the device reminds users of trackers for convicts, and it would form negative responses to the overall project.

4 ear

Because of its intense bloodstream, ears are considered as the best part for the heart rate tracking. On ears, arterioles are located between the antitragus and concha, so it enables more intensified heart rate detection (J. A. C. Patterson et al., 2009). Also, if the sound collection could be taken at the same spot as the user's ear, then collected sound data reflects the same amount of sound interference in the real environment, enabling higher data validity. However, ear detection is not commonly used because of various size and shapes of ears, so one device cannot fit ideally for everyone perfectly. Also, the biggest problem is that earbuds or ear clip can act as artifacts when a person need to speak or listen to something or someone, so the device should be designed in the way it does not harm users' daily use. As nurses' daily work routine involves communication with other people, this option has been excluded from the final option because of the feasibility.

5 chest

The chest is considered as a good body part for the accurate heart rate detection as it is physically near to heart. However, as PPG sensor needs to be attached to the body, with tight chest band inside of clothing, it did not fit the criteria of easy usability. As it is not sure if the detection above clothing area gives accurate data as much as attaching the sensor to the bare body, the user test involves detection in two different spots, on the bare chest and chest pocket to find out the best placement option.

6 neck

According to previous researches, forehead also works perfectly in accuracy wise because it has steady bloodstream and there is no hindrance for continuous measurement comparing to other body parts. The downside of it is that the usability is not as good as using other parts, such as using a wrist band, because users are not used to putting something on their above-head areas. In this regard, the neck was considered to substitute for forehead since it has a large vessel and stable bloodstream in it so that it can collect a precise dataset. Therefore, introducing a necklace type of device has been contemplated for the user scenario. To prove the usability of the product idea and the accuracy of the detection, user test with Fitbit Alta HR will be done in the next chapter.

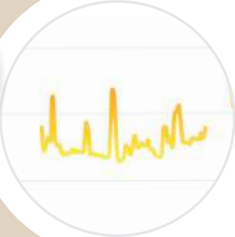
7 pants pocket

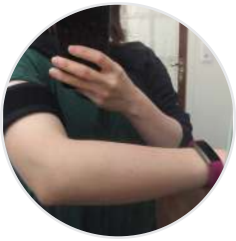
Putting the heart rate detection device to pants pocket has been considered as one of the user scenarios because of its easy usability. However, there is a doubt about the accuracy level of detection in this case. Since the device can only contact users through a layer of cloth when users put it on or put it inside of a pants pocket, which leads to doubt of expected accuracy level reduction. Furthermore, the way of attaching the device on or inside of the pants pocket was one thing that should be taken into account because bending, sitting posture, or walking can affect the overall detection quality too.

Additionally, study for the upper inner arm and palm side of the wrist area was planned as a part of the placement test. All in all, upper outer arm, upper inner arm, ear, chest pocket, pants pocket, bare chest, and the palm side of the wrist area were chosen for the heart rate detection in comparison with wrist area, which will be conducted with Fitbit Alta HR device.

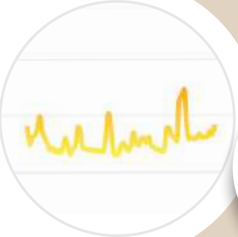
Placement of heartrate tracker testing

wrist





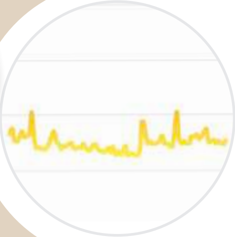
average
1 beat
difference

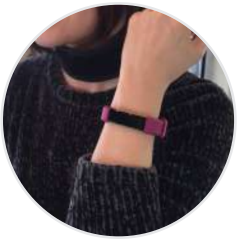


Test date: 14.02.19
11:50-18:05

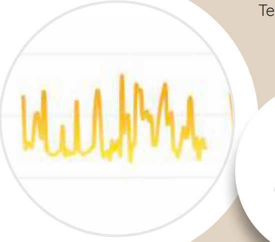
upper outer arm

wrist





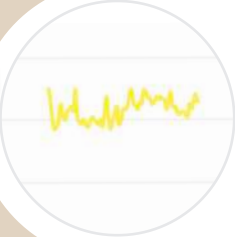
average
7 beats
difference

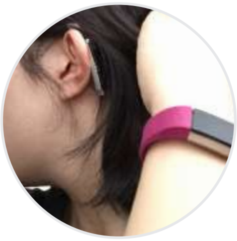


Test date: 15.02.19
7:40-15:10

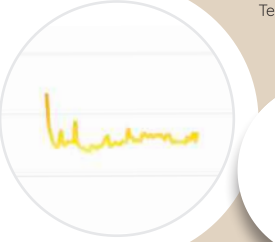
neck

wrist





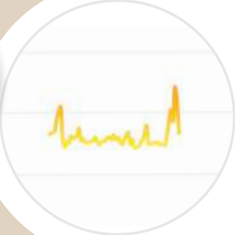
average
3 beats
difference

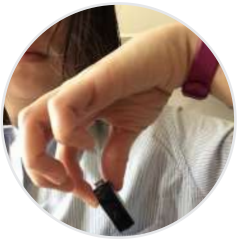


Test date: 16.02.19
15:55-22:05

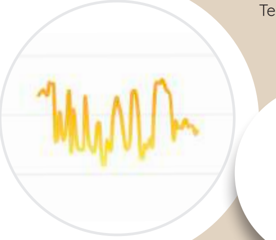
ear

wrist





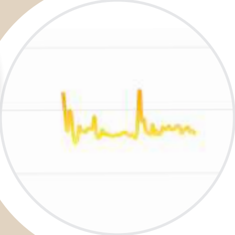
average
36 beats
difference

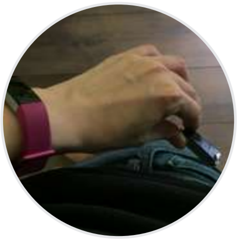


Test date: 17.02.19
13:15-18:40

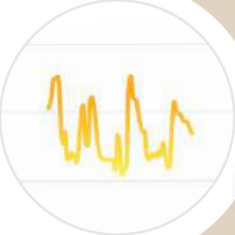
chest pocket

wrist





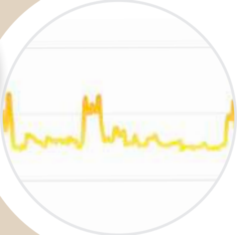
average
11 beats
difference

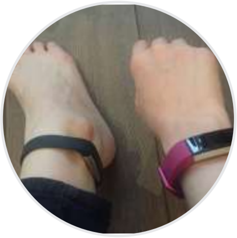


Test date: 21.02.19
12:05-17:30

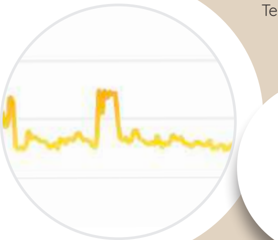
pants pocket

wrist





average
3 beats
difference



Test date: 22.02.19
11:50-20:30

ankle

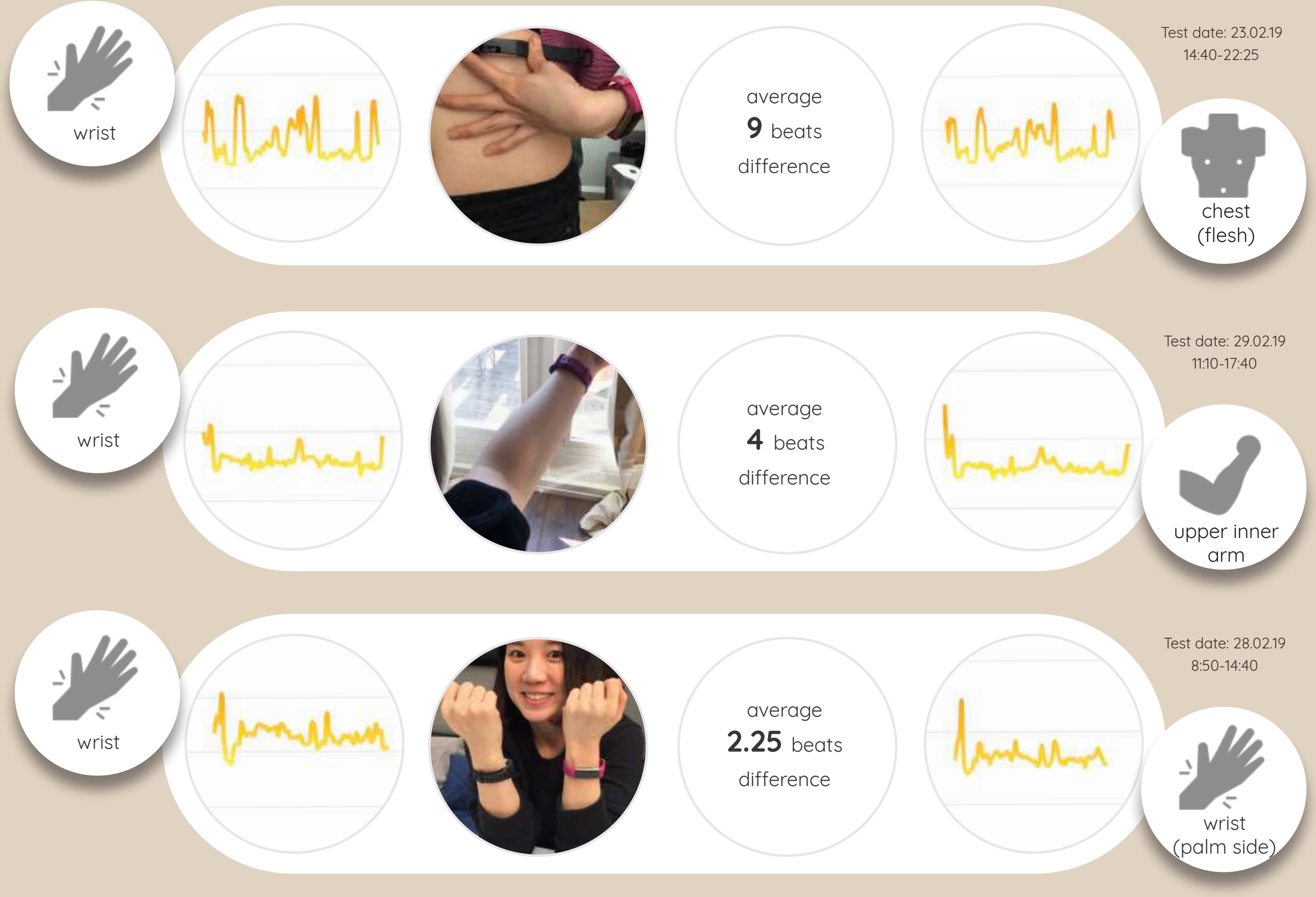


Figure 22.Heart rate tracker placement test

Placement validity test

Placement of the device has been contemplated as a part of an embodiment. Accuracy, usability, and user acceptance have been considered as three main subjects of this user test. Fitbit Altra HR was chosen as a testing device because of its reliability, which was contemplated as a heart rate detection method in the previous chapter.

For the testing, two Fitbit devices were purchased in different colors, and one of them was put on a wrist, while the other was attached to nine different spots for the data comparison. The test was conducted on the upper outer arm, upper inner arm, ear, chest, neck, chest pocket, pants pocket, ankle, and palm side of the wrist area in comparison to the wrist area, and heart rate data were collected from two spots simultaneously.

As Fitbit Altra HR is aimed to be put on the user's wrist area, the number measured from the wrist was used as a reference point. I compared numbers collected from the other spots, so the accuracy or the measurement from other body parts can be assessed by seeing the difference than the wrist. Each test was taken on different days, for more than a minimum of 5 hours a day. During the test, test participants were asked to perform their daily routine using a laptop, and dynamic physical movement was refrained from during the test because it does not reflect nurses' work routine.

Test result

During each experiment, the heart rate records were marked every 5 minutes. The average heart rate difference (Average heart rate difference = Average |value from wrist-value from variable|) was calculated, and the whole dataset is available in Appendix A.

As a result, upper outer arm area showed the smallest gap than the wrist, while palm side of the wrist area marked the second smallest number with 2.25 beats difference, which was followed by ear and upper inner arm which had three beats difference each. The neck showed seven beats difference than the average heart rate collected from the wrist while chest recorded nine beats gap. The number collected from the pants pocket and chest pocket had a more significant gap than other areas, having 11 and 36 beats gaps respectively, with the lowest accordance among all tested areas. It is because of the fact Fitbit device did not touch the flesh all the time which is required for PPG sensing.

Even though ear has been considered for the implementation at first, it was excluded from the option because of physical discomfort during the test. Also, wearing something could hinder conversation, which is a considerable part of the nurses' work routine. Also, both chest and pants pocket area was regarded as ideal spots because of high applicability, but the test result showed the least accuracy in those two areas because of the cotton layer between the skin and the tracker. All in all, the outer upper arm has been chosen as a place for heart rate tracker because of its high accuracy, usability, and acceptance.

Subjective assessment of stress

Self reporting: subjective fatigue assessment tool

While heart rate tracking is used as a means of objective stress level assessment tool, various self-reporting tools were studied to be applied as a subjective fatigue assessment tool. After the collection of the sound and heart rate, those data set will be compared with emotion self-reports from users, so researchers can understand which sound triggers which stress responses; in the end, to validate noise fatigue as a result.

Wellness questionnaires

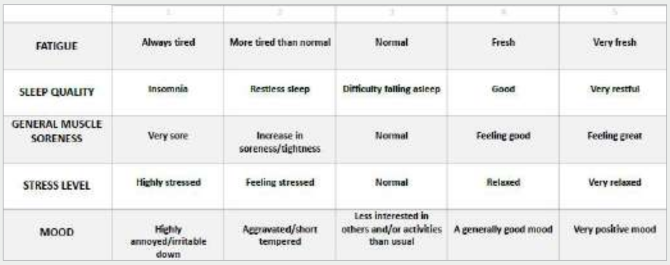
Wellness questionnaires is a measure which is often used to look into the fatigue level of athletes to measure the perceived physical fatigue level. Wellness questionnaire is beneficial in the way it covers external factors than physical symptoms, such as sleep status from the previous night and personal emotional status (Martin Buchheit, 2015). However, the questions are highly inclined to report physical symptoms than emotional symptoms. Thus, it has a limitation that it cannot cover the psychological aspect of noise fatigue.

Rating of a fatigue scale

Rating of fatigue scale has been investigated as an effective way to record the subjective feelings of respondents regarding their fatigue levels. As people tend to have diverse personal factors, it can be hard for users to pick a single number which stands for their fatigue level as a whole. Also, since the perceived level of fatigue differs depending on individuals and situational interpretation, it is hard to combine multiple users' responses and make it into a valid dataset for further analysis.

Emotion report

Emotion report was studied because previous questionnaires were more oriented to physical fatigue assessment. Also, they asked respondents to generalize their fatigue level and mark it into a flat answer as one number, so it could not fully reflect users' subjective responses. However, the noise fatigue that I tried to look into through subjective assessment was more inclined to mental fatigue since physical fatigue assessment will be taken care of by using heart rate tracker. Therefore, PrEmo and Circumplex of affect model, which focuses more on users' emotion, were reviewed additionally.



	1	2	3	4	5
FATIGUE	Always tired	More tired than normal	Normal	Fresh	Very fresh
SLEEP QUALITY	Insomnia	Restless sleep	Difficulty falling asleep	Good	Very restful
GENERAL MUSCLE SORENESS	Very sore	Increase in soreness/tightness	Normal	Feeling good	Feeling great
STRESS LEVEL	Highly stressed	Feeling stressed	Normal	Relaxed	Very relaxed
MOOD	Highly annoyed/irritable down	Aggravated/short tempered	Less interested in others and/or activities than usual	A generally good mood	Very positive mood

Figure 23. Wellness questionnaire from Martin Buchheit (2015)

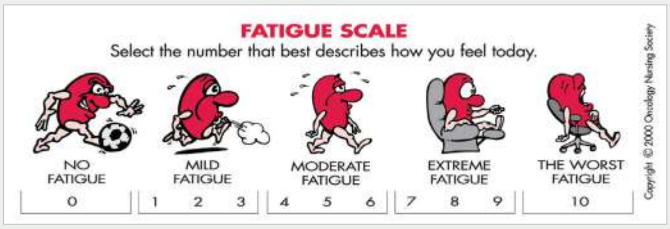

FATIGUE SCALE
Select the number that best describes how you feel today.
NO FATIGUE
0
MILD FATIGUE
1
2
3
MODERATE FATIGUE
4
5
6
EXTREME FATIGUE
7
8
9
THE WORST FATIGUE
10
Copyright © 2000 Oncology Nursing Society

Figure 24. Rating of fatigue scale from Micklewright et al. (2017)

Figure 25. Various emotions

Emotion report tools

Product Emotion Measurement Tool (PrEmo)

PrEmo was developed as a non-verbal self-reporting tool by SUSAGROUP and has been widely used for user perception and emotion assessment in various industries. The major perk of using PrEmo is coming from the fact that it uses visuals for various emotions, so people can easily access to the testing with a little literacy about the test. Also, as a facial expression is something international, and it can be perceived intuitively without further descriptions, it can be used for many test participants from various backgrounds, covering various age groups, educational levels, ethnicity, and geographical locations. However, this questionnaire was excluded from the implementation because this assessment tool is mainly focusing on the evaluation of physical products than the environment or situations.

Circumplex of affect model

The circumplex model of affect was studied for emotion report function development of the Cacophony mapper system. The model has positive and negative emotion as its polar opposite in X-axis while having the level of arousal in its Y axis. I especially focused on unpleasant-intense quadrant than other three areas because the purpose of the emotion report is to define the noise fatigue of nurses in the ICU. Since the focus should be finding negative emotion triggered by sound stimuli in the medical surroundings, I decided to develop this quadrant as a emotion report interface on the application. Also, mild emotions were excluded from the focus area because emotions given in this area are far from emotions that nurses would feel about the sound experience in the ICU.

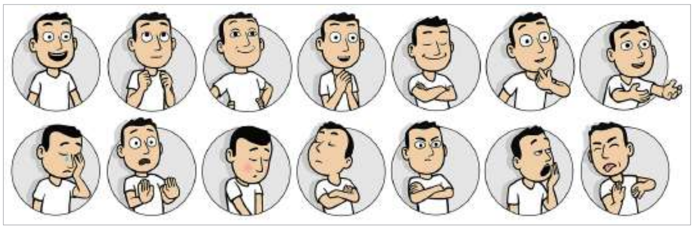


Figure 26. Premo from SUSA Group

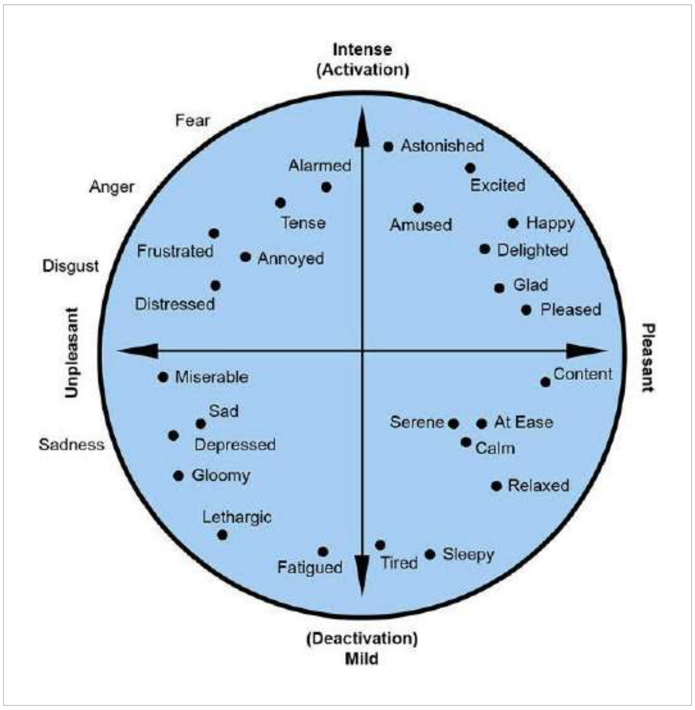
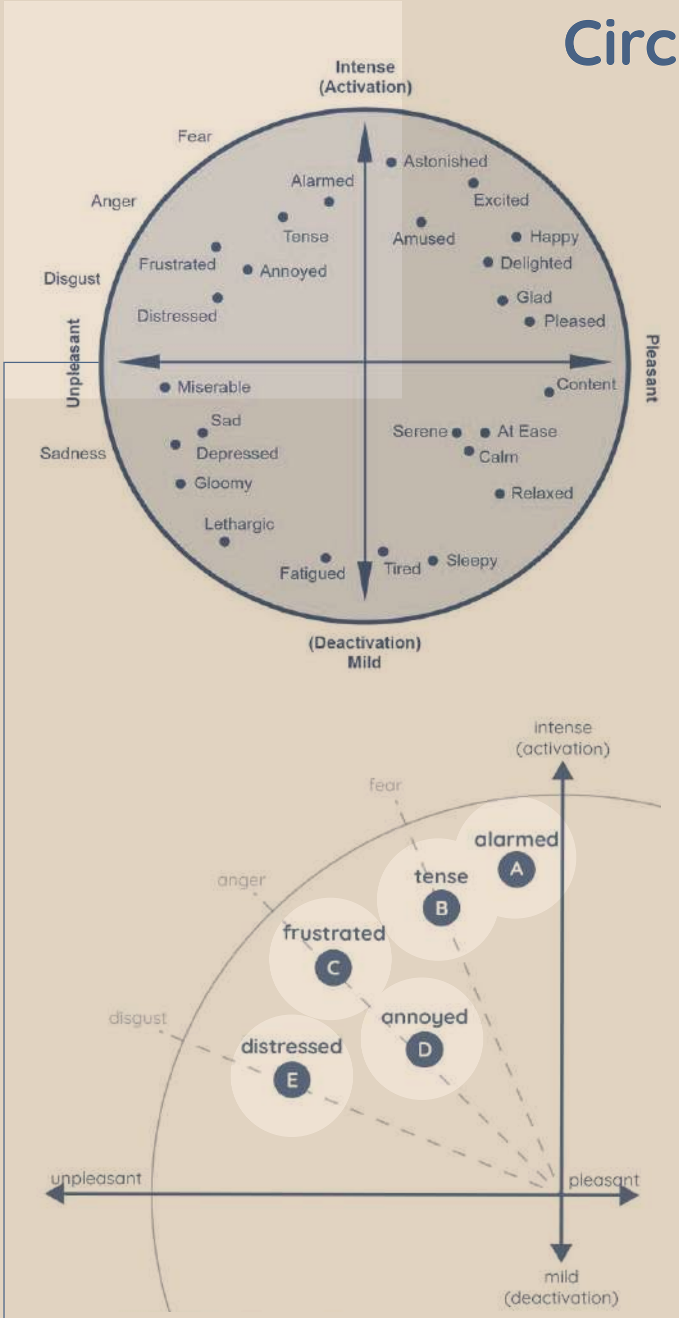


Figure 27. Circumplex of affect from Jonathan Posner et al. (2005)

Circumplex model of affect



Emotion report: circumplex model of affect

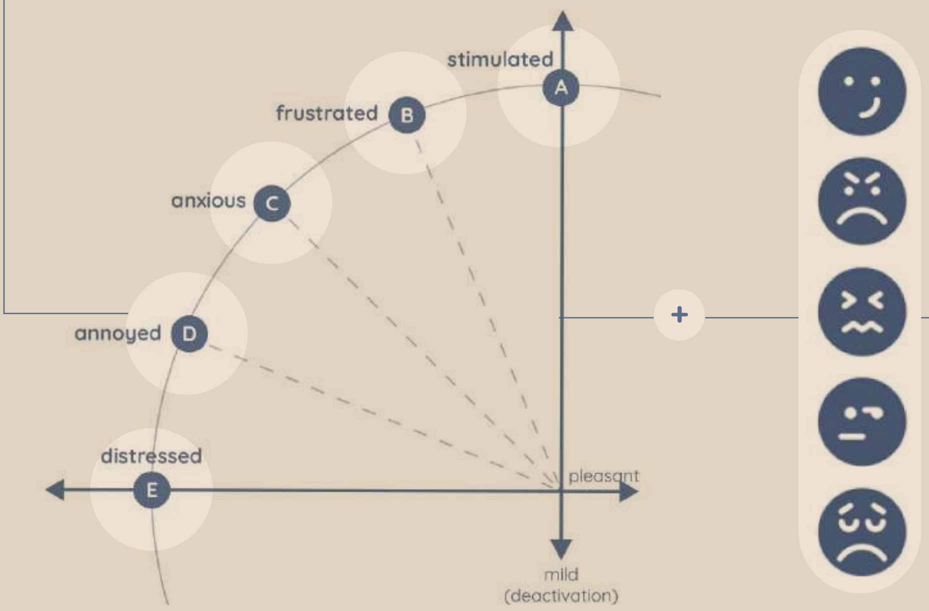
Circumplex model of affect has been chosen as a tool for an emotion report. Only emotions in the negative-intense area in the model have been chosen for more clarity of the reporting process as the rest of the emotions in the model was not relevant to ICU sound environment in general. Also, the quadrant of the model was used because the purpose of this project is to create a system which can validate noise fatigue in the ICU, which involves the collection of noise stimuli in the unit and negative responses toward it.

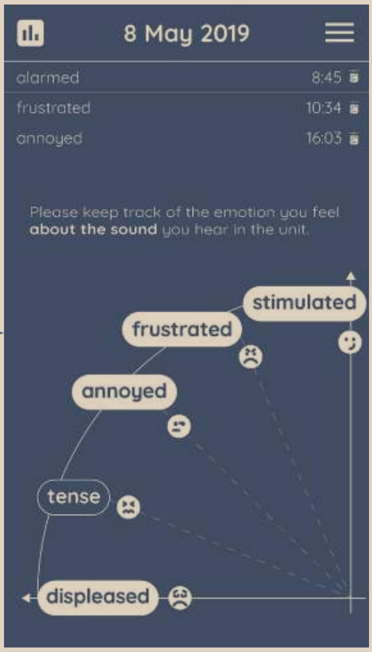
In the unpleasant-intense emotion zone, five emotions that are relevant to the ICU sound was selected, and those emotions are alarmed, tense, frustrated, annoyed, and distressing emotions.

To make the interface of the reporting function more clear, chosen emotions were re-distributed to each extreme, so emotions have clearer gaps with each other, and users can easily differentiate one from the others. Also, indication lines for each angle were deleted for clarity of the overall layout.

Finally, emotion icons were added right next to words, so users quickly get what the emotion is about as an added description. Emoji icon that is commonly used in messenger applications had been considered to be used at first but excluded from the choice in the end, because many people use the same emoji for various situations, and it means they can be read the same emoji differently depending on their prior experiences with those icons.

Figure 28. Appliance of Circumplex of affect to the interface





8 May 2019

alarmed 8:45
frustrated 10:34
annoyed 16:03

Please keep track of the emotion you feel **about the sound** you hear in the unit.

Basic sound properties in the ICU

Before working on the sound classification, basic sound knowledge and sound properties in the ICU were studied.

dB (Decibel)

Decibel refers to the quantified amount of sound which is translated into a comparative unit. As human beings cannot perceive the sound level proportionally even if the sound level has been changed on a regular scale, the concept of dB was created. Since we cannot perceive ten times amplified version of sound as ten times bigger than the original sound, dB takes other scales into accounts, such as voltage ratio, wavelength, power, and amplitude ratio so that we can refer to sound in a more readable and understandable manner. In this study, dB will be additionally used to figure out the noise component in the ICU. Later, dB information will be combined with other sound datasets so that the sensor can capture more precise sound information from the ICU surroundings.

Frequency

Frequency refers to the number of reoccurrences of the sound wave per one minute, so the pitch of each sound can be quantified, having its unit as Hz (Hertz) or rpm (revolutions per minute). As sound is a result of constant vibration through a medium from sound sources, each sound has its unique frequency depending on the sound source and the environment, showing various patterns of different range and waveforms.

As mentioned in a previous research of Busch-Vishniac, each sound in the ICU can be differentiated by analyzing waveforms of their own. For example, a wave pattern with low-frequency level can be expected from sound sources, such as blown winds, air conditioning, and ventilator sound in the ICU environment. Also, as mentioned in the same research, human speech is concentrated on 300 to 3000 Hz frequency range with irregular patterns comparing to regular machinery noise (Busch-Vishniac, 2006). In this regard, if we look into frequency properties of each sound group, it means that creating a sound classifier which can differenciate each sound category is possible.

General sound range in the ICU

According to research done in Johns Hopkins in 2005, the sound spectrum of the ICU was positioned between 63 Hz to 1000 Hz, and the sound level always exceeded 45dB. Also, more than half of the entire exploration showed that the sound level exceeded 52-59dB (Busch-Vishniac et al., 2005). Also, the sound level of the ICU showed more extreme record than the average hospital sound, marking more than 85dB at all sites, scoring 51dB as its overnight average after evening shift (Darbyshire & Young, 2013).

To find essential sound properties of each sound group which can be used for the sound categorization, in the end, frequency range, waveform, sound spectrum, and amplitude range have been studied, so those acoustic features can be used to filter each group from a mixture of various sound sources.

Previous research indicated that the sound which has a frequency range of 63 to 1000 Hz could be found on the octave band of 50 to 60 dB (Busch-Vishniac et al., 2005). The same study indicated that sound which has more amplitude than 60 dB has a lower frequency than 63Hz, while the frequency between 1000 to 16000Hz is corresponding to the sound group which has lower sound dB than 50dB (Busch-Vishniac et al., 2005). Within this knowledge, the sound information in the ICU, mostly the amplitude range, from various research papers could have been translated into the frequency range, so I could get to know the general the band range of each sound.

Sound data confidentiality

As the project is highly dependent on information gained through data logging, confidentiality, and privacy of the recorded data have been contemplated as a vital part of this project. For the acceptance of the system and the project itself, several ways of improving trustworthiness in the data confidentiality have been considered.

Therefore, it was necessary for users to make sure that the data accessibility will be confined only to academic and research purpose, and the data will be accessed only in a distorted format so that no one can listen to the original recording. At first, deleting data right after each analysis was considered, but the data can be used for a longer time for making a more extended period of data comparison. So, I decided to use a filter and save the sound as non-human-readable, and non-reversible form, so sound researchers gain only sound information that they need to achieve through the experiment, not the complete sound recording itself.

How to categorize sounds in the ICU?

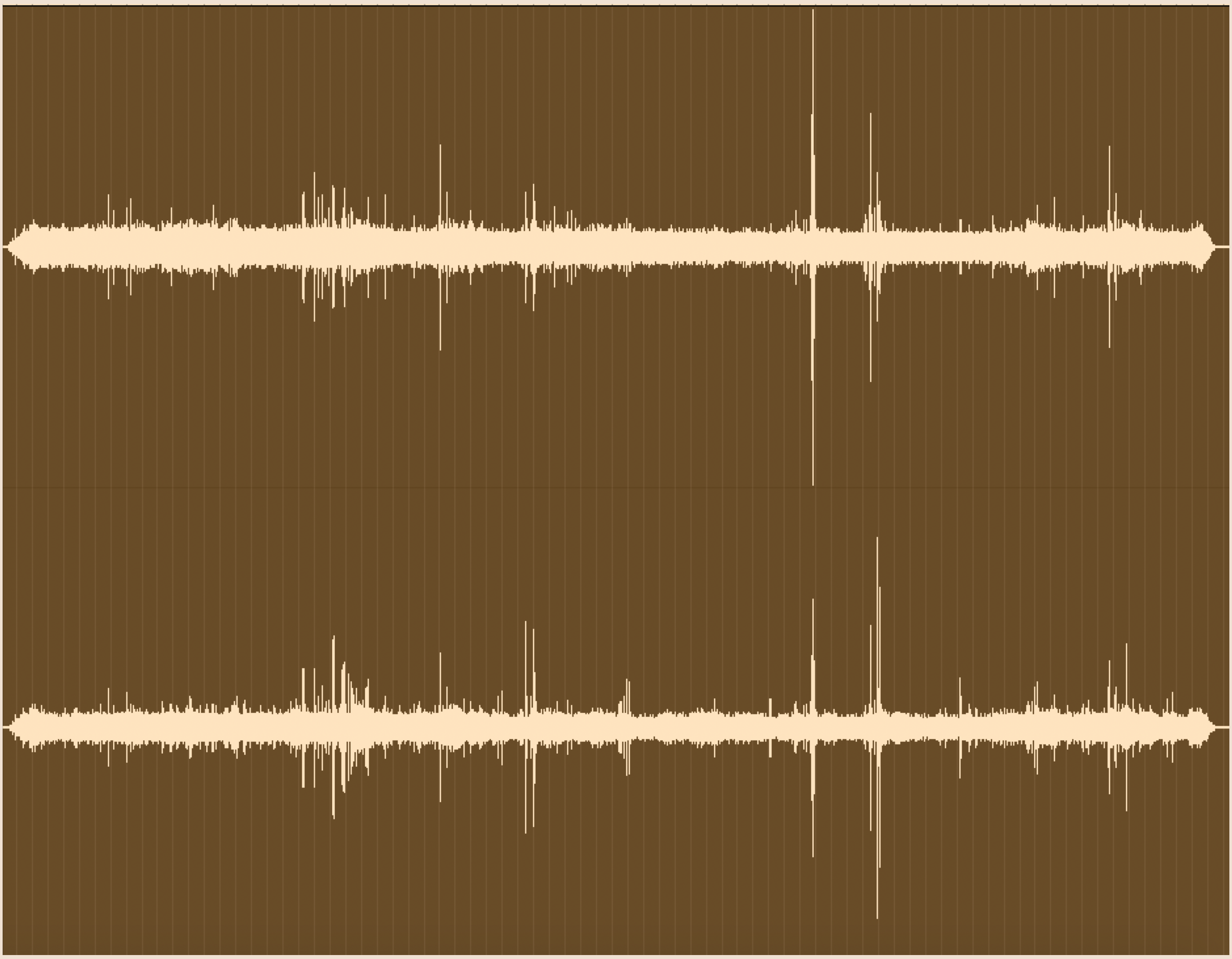


Figure 29. ICU Sound recording

Methodology: which sound properties should be taken into account for the sound evaluation?

For the sound analysis, the basic concept of sound, such as frequency and decibel, and relevant sound information in the hospital environment was studied to get the basic concept and idea of the project. A further step is to collect sound samples from the ICU and analyze them with various sound filters, such as sound wave, sound spectrum, and FFT(Fast Fourier Transform) filters to figure out each sound categories' unique sound properties to enable the categorization.

The first step is the collection of ICU sound samples. To create a sound filter, a total of 43 sound files which were recorded in the ICU from BBC radio station were reviewed. After listening to them, I divided them into eight categories, which is background sound, ventilator sound, conversational sound, alarms, footsteps, door slam, trolley dragging sound, and object clashing sounds. I labeled them into 200 different sound samples, so each groups' sound properties can be studied. A more detailed process is available in chapter 3.10 and 3.11.

After the labeling, various sound filters were applied to find the sound properties of each group. After finding frequency peaks and power peaks by applying sound samples in FFT (Fast Fourier Transform) filter, a collected dataset was applied into classification filter in Matlab to find out the best algorithm which is best to enable sound classification with the highest accuracy. A description of sound classification filter generation is available in chapter 3.12.

Succeedingly, sound classification filter was optimized in chapter 3.13 by enlarging sample size and improving the sound quality of the sample to get better accuracy of the filter. As a result of three stages of optimization, a sound classification filter which has 71.4% of accuracy was created, and the whole process is available in the next chapter.

Sound sampling

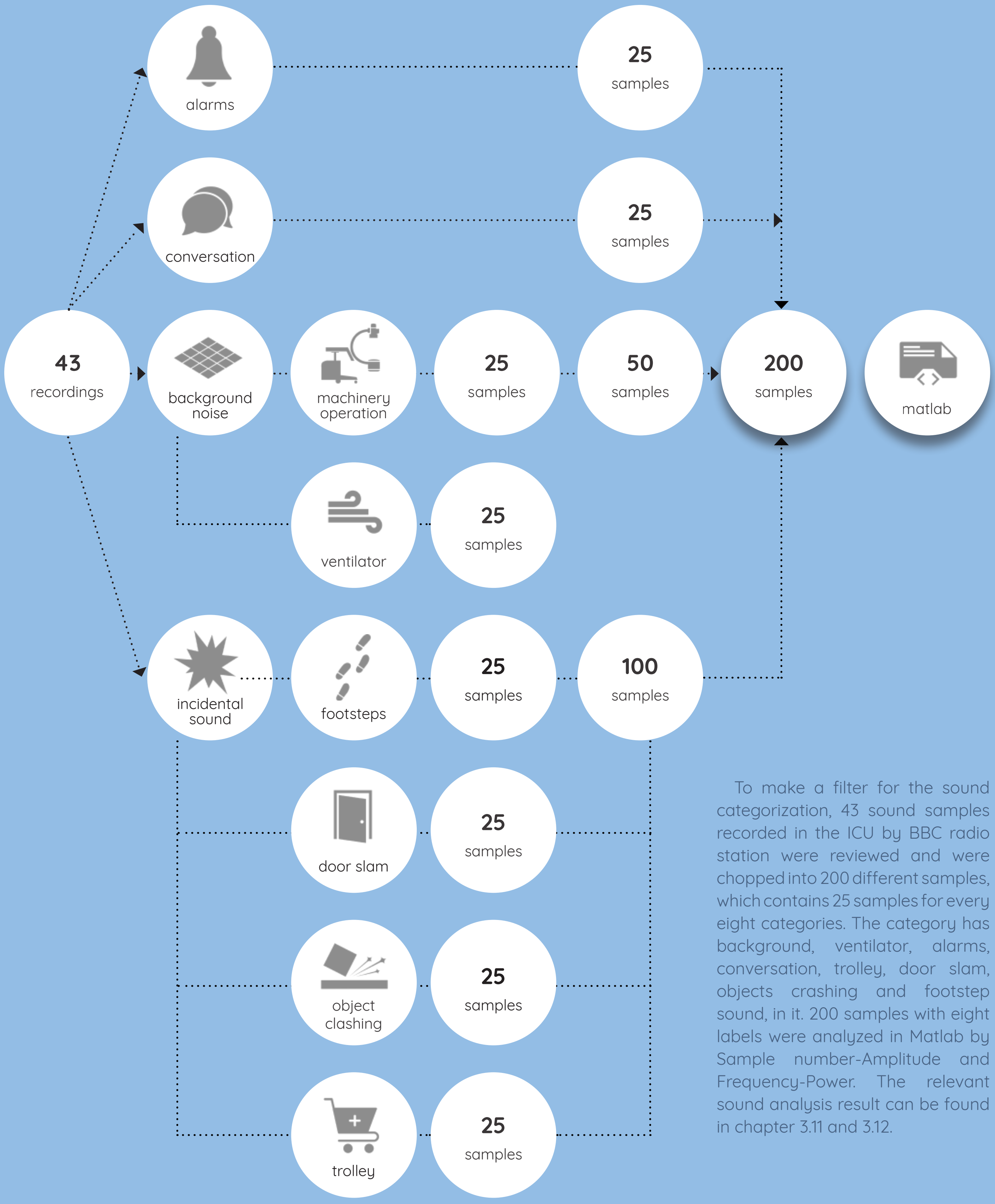


Figure 30. ICU Sound sampling process

To make a filter for the sound categorization, 43 sound samples recorded in the ICU by BBC radio station were reviewed and were chopped into 200 different samples, which contains 25 samples for every eight categories. The category has background, ventilator, alarms, conversation, trolley, door slam, objects crashing and footstep sound, in it. 200 samples with eight labels were analyzed in Matlab by Sample number-Amplitude and Frequency-Power. The relevant sound analysis result can be found in chapter 3.11 and 3.12.

Sound processing: Sample number-Amplitude

In the Sample number-Amplitude analysis, the time scale in the data is compressed by 10 to raise the pitch and make the sound more clearly audible. 200 sound samples from 8 categories have been applied to this filter, and each pattern has been analyzed in this chapter.

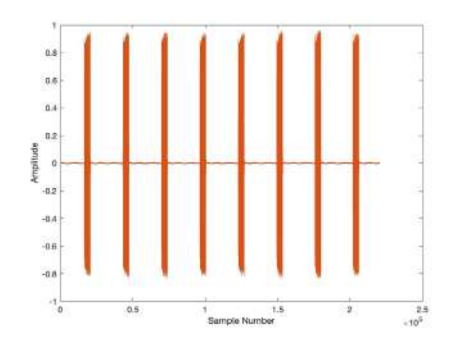


Figure 31. Alarm 19

Alarms

Alarms have the most apparent patterns compared to other sound sources because of its regular phasing of the peaks and the amplitude range, which lies between -0.8 and 0.8. The graph shows more obvious patterns when there is less noise in the recording, while added environmental sounds to alarms form a thicker orange horizontal area in the middle, overlapping with the pattern of alarms.

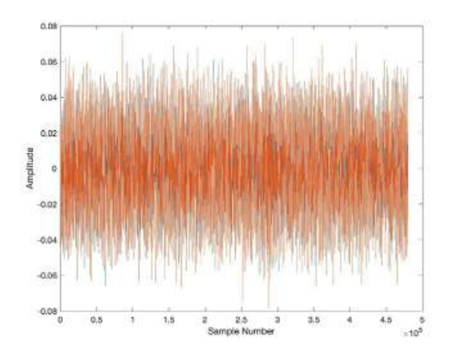


Figure 32. Background 27

Background

Background sample shows more amplitude fluctuation with more significant and fuzzier orange area in the middle compared to other categories. It shows more severe fluctuations because of the frequent amplitude level change when the recording has more noise in it. On the other hand, when the recording is more clear, the height of the orange area is narrower with smaller amplitude range. Even though quiet surroundings were cropped as samples to show the general tranquil ICU atmosphere in the background, various sound sources that are included in the recording still have impacts on the general shape of the sound pattern. It is because background noise is a combination of various sounds from other categories with diverse patterns and the doninancy of each component matters to the pattern of background sound recording.

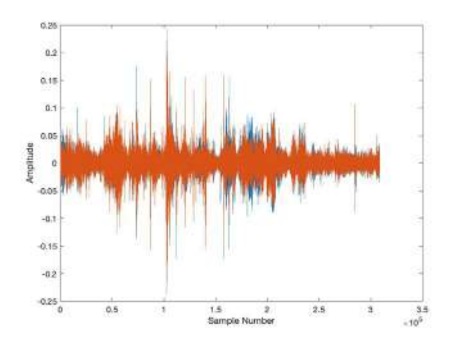


Figure 33. Conversation 8

Conversation

The conversation shows irregular patterns both in the sample number range and the amplitude range. The range of the amplitude varies depending on samples, as the dominance of the conversation in the recording is an essential factor. While dominant background sound in conversation sample forms a big fat orange area in the middle, metal-clashing sounds in conversation sample make prominent peaks. Therefore, it is crucial to include pure conversation sound among conversation sample group as well as sound samples from real surroundings so that the created classification algorithm can reflect both patterns.

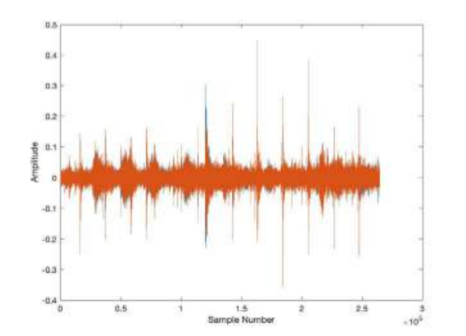


Figure 34. Footsteps 11

Footsteps

Footsteps make regular patterns in sample number axis, while amplitude shows various values even in the same sample. The peak range of the amplitude varies because people walk in different types of shoes with different sole materials with their own steps. When high heel shows more extreme peaks in a graph, light footsteps with soft rubber sole shows smaller fluctuation in the amplitude range. The pattern forms a narrower orange area in the horizontal center line when the recording has less noise in it, while more noises form a bigger orange zone.

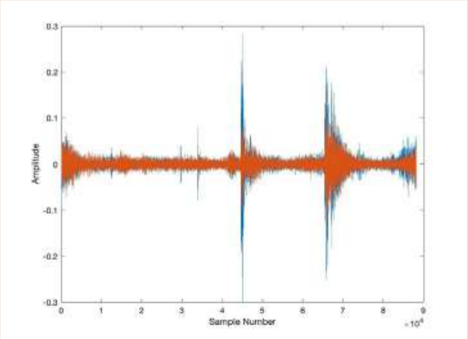


Figure 35. Object clashing 16

Objects clashing

Objects clashing sound usually forms triangular patterns with sudden peaks with high amplitude values, and it narrows down drastically. The number of triangular pattern in the recording shows the number of clashes and how hard objects bumps into each other. Also, the clarity of the pattern is dependent on the material of objects since dull material tends to make one clear sound with very sharp borderline in a triangular pattern while metal clashing often makes small vibration follows to the main clashing, which leads to a fuzzy outline in a triangular shape.

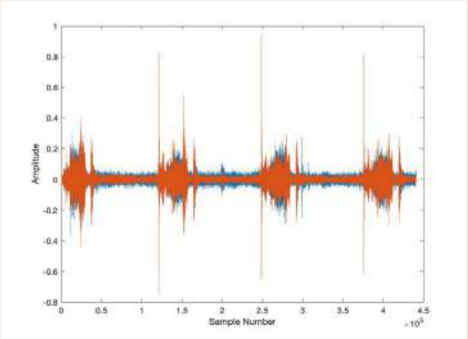


Figure 36. Ventilator 13

Ventilator

Ventilator sound forms recurring diamond-shaped patterns with following next peaks with small triangular shape with narrow end on the right-hand side. This pattern is made because of the sequence of the pumping mechanism. The pattern of the ventilator is comparably noticeable than other sample groups because it generates recurring patterns as alarms do.

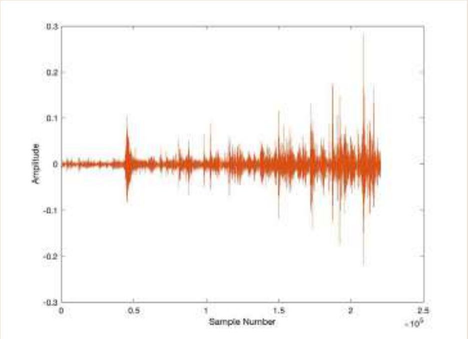


Figure 37. Trolley 20

Trolley

The trolley shows an irregular pattern with severe amplitude fluctuations. The overall pattern sometimes increases and decreases depending on the changing distance of the trolley to the microphone for the recording. If there is more rattling sound from the recording, the graph shows higher peaks, while the dominance of the general noise in the recording leads to the more significant orange area in the middle area.

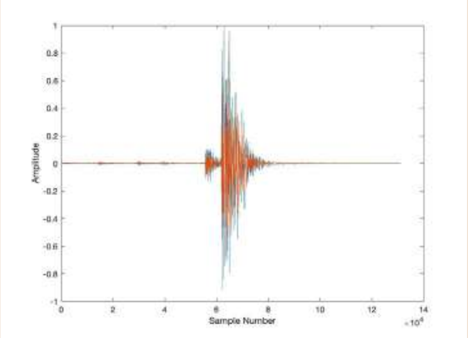


Figure 38. Door slam 7

Door slam

Door slam makes a triangular pattern which has a sudden peak with an immediate decrease. Sometimes one more triangular form is made right next to the bigger triangle on the left. It is because the door makes the small fricative sound when it hits the door frame first and makes a more prominent sound at the moment that the door is completely shut. As can be seen from other recordings in various categories, the general noise level in the recording affects the thickness of the orange area, and more noises cause bolder thickness in the middle zone in the graph.

Sound processing:
Fast Fourier Transform(FFT)

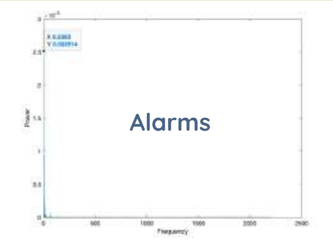


Figure 39. Alarm 19

Max F	Ave F peaks	Max P	Ave P peaks
356.4	43.1363	169.6	8.1417

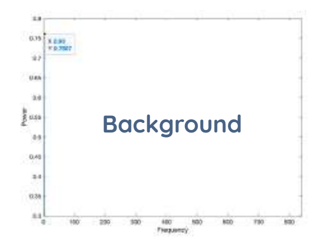


Figure 40. Background 27

Max F	Ave F peaks	Max P	Ave P peaks
1563	80.1389	1.08E+01	1.95E-01

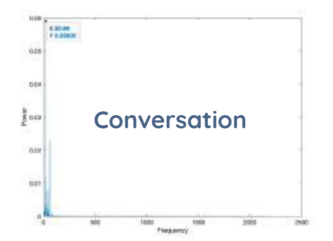


Figure 41. Conversation 8

Max F	Ave F peaks	Max P	Ave P peaks
1563	50.7277	9.68E-01	8.76E-02

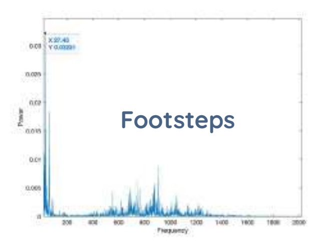
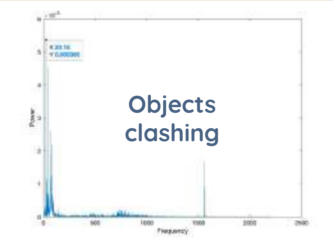


Figure 42. Footsteps 11

Max F	Ave F peaks	Max P	Ave P peaks
1564	106.8879	3.32E-01	4.17E-02

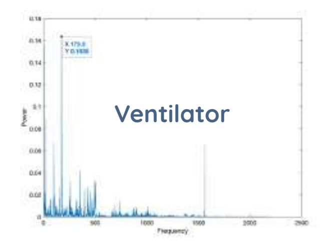
FFT (Fast Fourier Transform)

Fast Fourier transformation has been studied and applied to sound samples since it is a concept which helps decomposing combined frequencies from a synthesized sound. As every sound has different notes with various air pressures, each sound has its unique patterns of their own as well as their frequency ranges. However, if multiple sounds occur at once, they add up altogether and the sound is recognized a sum. When two higher pitched-notes adds up together, the wave shows spiked high frequency, while lower pitched-note and higher pitched-note cancel out each other and makes a different pattern than those two individual sounds (Bracewell, 1989). Therefore, Fourier transformation was considered as a means of finding pure sound sources for further sound categorization.



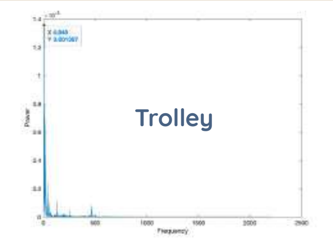
Max F	Ave F peaks	Max P	Ave P peaks
1014	104.411	2.55E-01	2.27E-02

Figure 43. Object clashing 16



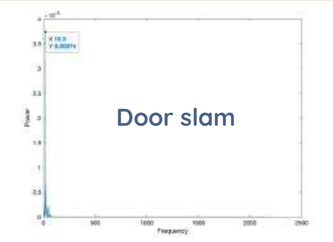
Max F	Ave F peaks	Max P	Ave P peaks
1562	350.859	1.45E+00	1.86E-01

Figure 44. Ventilator 13



Max F	Ave F peaks	Max P	Ave P peaks
961.5	49.0448	2.32E+00	1.88E-01

Figure 45. Trolley 20



Max F	Ave F peaks	Max P	Ave P peaks
86.13	12.70465	9.27E+00	4.01E-01

Figure 46. Door slam 7

By applying 200 sound samples into FFT filter in Matlab, each three of frequency peaks and power peaks were came up with. As a result, ventilator showed the highest average frequency peak of 350.859, followed by footsteps which have average frequency peak of 106.8879 and 104.411 of object clashing. Door slam sound showed the least average frequency peaksof 12.70465 and alarms and trolley dragging sound showed the second and the third smallest average frequency peak. In terms of the maximum frequency value, background, footsteps, and ventilator sound recorded the highest value. As the highest value can show the biased number due to the use of the same sample source, each frequency peak and power peak values were listed from the highest to the third highest value.

Sound classification filter creation
: with 8 categories

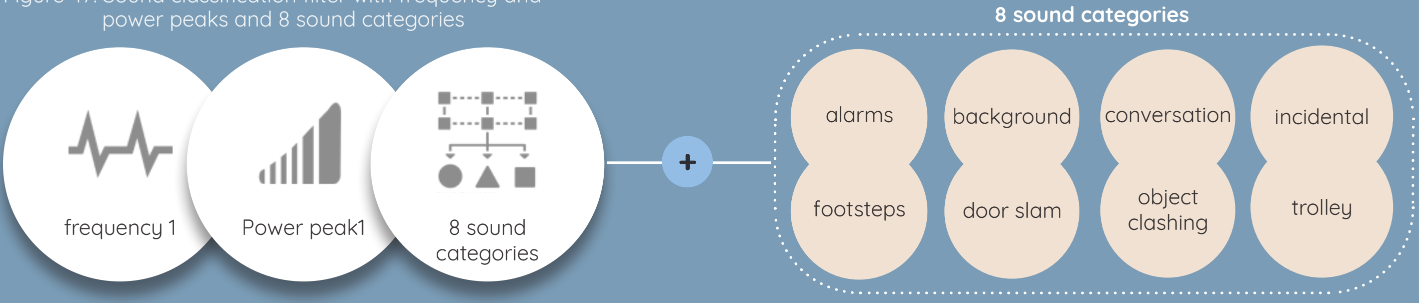
Sound classification with Matlab (1st attempt)

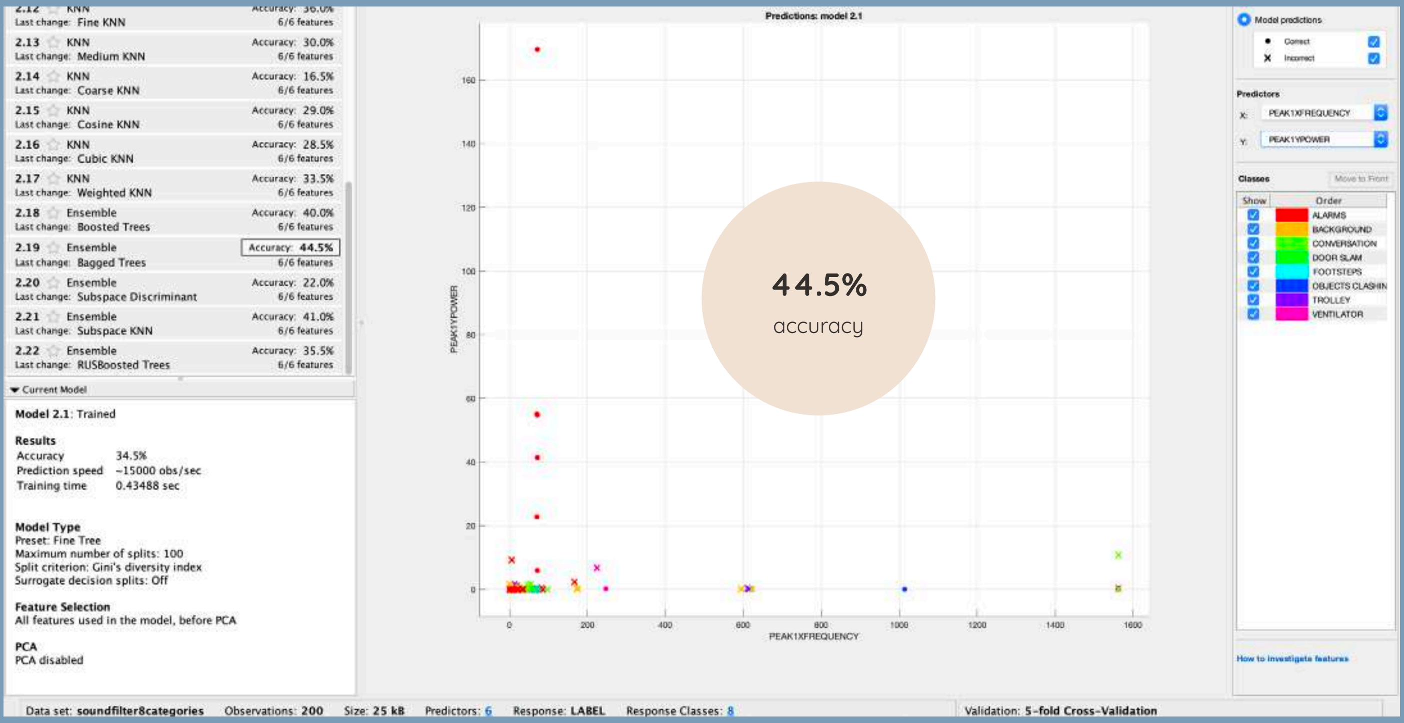
Top three power peaks of 200 samples were listed on Excel and exported to Matlab to create a sound classification filter. By using the classification learner, various algorithm with different accuracy were created with machine learning feature in Matlab. In this process, various options were reviewed whether samples should be divided into eight groups or four groups and whether using variables of Frequency-Frequency, Frequency-Power, or Power-Power combination will make a filter with better accuracy as a result. The exploration process is available in Figure 41 to 45.

Using 8 sample categories

With 8 sample categorization, three peak values of background, ventilator, alarms, conversation, trolley, door slam, objects clashing and footstep sound from each 25 samples were used in machine learning train process. As a result, using Frequency-Power as a variable showed the best accuracy of 44.5%, which was followed by the accuracy of the combination of Frequency-Frequency which marked 39.5%. Lastly, using Power-Power variables for classification showed the lowest accuracy of 31%.

Figure 47. Sound classification filter with frequency and power peaks and 8 sound categories





2.12 KNN Accuracy: 36.0%
Last change: Fine KNN 6/6 features

2.13 KNN Accuracy: 30.0%
Last change: Medium KNN 6/6 features

2.14 KNN Accuracy: 16.5%
Last change: Coarse KNN 6/6 features

2.15 KNN Accuracy: 29.0%
Last change: Cosine KNN 6/6 features

2.16 KNN Accuracy: 28.5%
Last change: Cubic KNN 6/6 features

2.17 KNN Accuracy: 33.5%
Last change: Weighted KNN 6/6 features

2.18 Ensemble Accuracy: 40.0%
Last change: Boosted Trees 6/6 features

2.19 Ensemble Accuracy: 44.5%
Last change: Bagged Trees 6/6 features

2.20 Ensemble Accuracy: 22.0%
Last change: Subspace Discriminant 6/6 features

2.21 Ensemble Accuracy: 41.0%
Last change: Subspace KNN 6/6 features

2.22 Ensemble Accuracy: 35.5%
Last change: RUSBoosted Trees 6/6 features

Model 2.1: Trained

Results
Accuracy 34.5%
Prediction speed ~15000 obs/sec
Training time 0.43488 sec

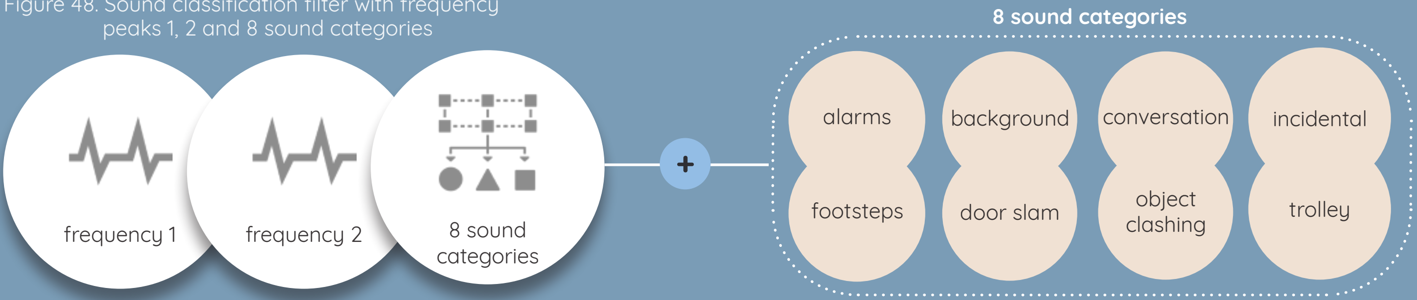
Model Type
Preset: Fine Tree
Maximum number of splits: 100
Split criterion: Gini's diversity index
Surrogate decision splits: Off

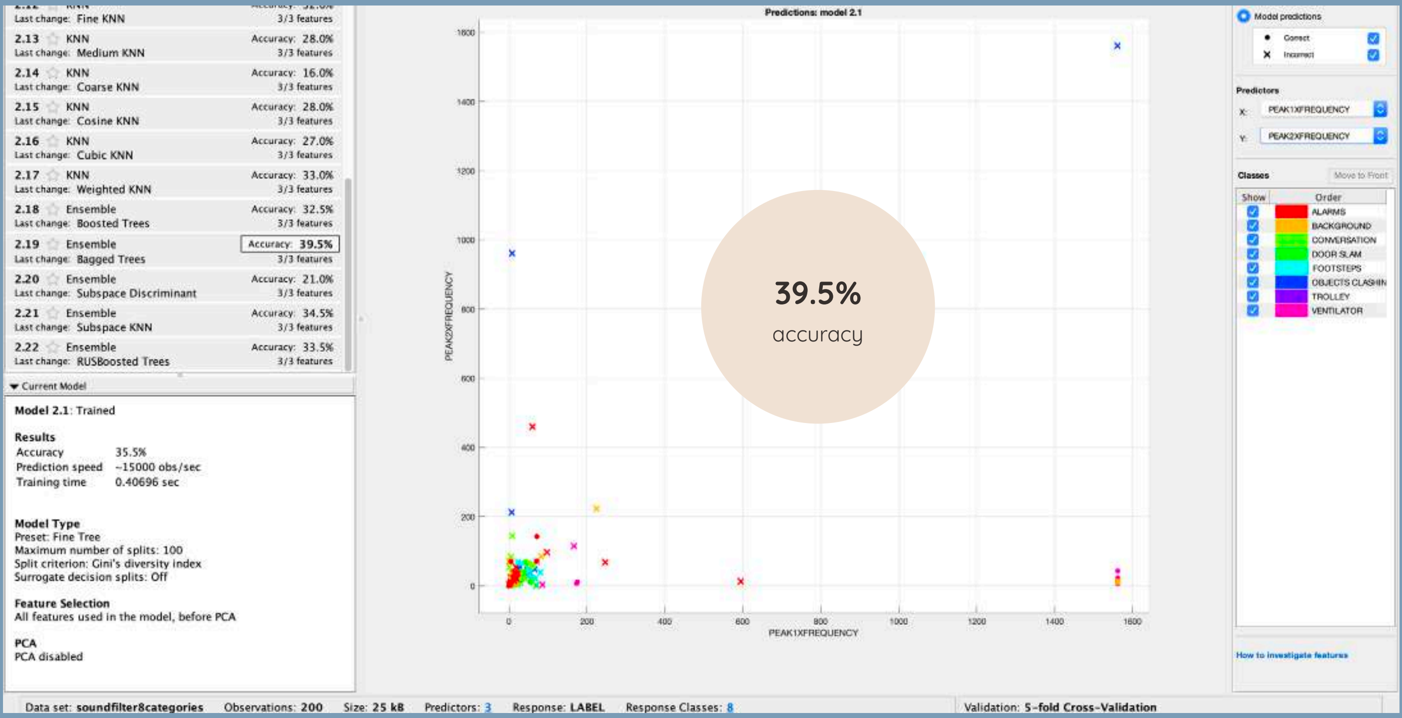
Feature Selection
All features used in the model, before PCA

PCA
PCA disabled

Data set: **soundfilter8categories** Observations: **200** Size: **25 kB** Predictors: 6 Response: **LABEL** Response Classes: 8 Validation: **5-fold Cross-Validation**

Figure 48. Sound classification filter with frequency peaks 1, 2 and 8 sound categories





2.13 KNN Accuracy: 28.0%
Last change: Medium KNN 3/3 features

2.14 KNN Accuracy: 16.0%
Last change: Coarse KNN 3/3 features

2.15 KNN Accuracy: 28.0%
Last change: Cosine KNN 3/3 features

2.16 KNN Accuracy: 27.0%
Last change: Cubic KNN 3/3 features

2.17 KNN Accuracy: 33.0%
Last change: Weighted KNN 3/3 features

2.18 Ensemble Accuracy: 32.5%
Last change: Boosted Trees 3/3 features

2.19 Ensemble Accuracy: 39.5%
Last change: Bagged Trees 3/3 features

2.20 Ensemble Accuracy: 21.0%
Last change: Subspace Discriminant 3/3 features

2.21 Ensemble Accuracy: 34.5%
Last change: Subspace KNN 3/3 features

2.22 Ensemble Accuracy: 33.5%
Last change: RUSBoosted Trees 3/3 features

Current Model

Model 2.1: Trained

Results
Accuracy 35.5%
Prediction speed ~15000 obs/sec
Training time 0.40696 sec

Model Type
Preset: Fine Tree
Maximum number of splits: 100
Split criterion: Gini's diversity index
Surrogate decision splits: Off

Feature Selection
All features used in the model, before PCA

PCA
PCA disabled

Model predictions
● Correct ☑
× Incorrect ☑

Predictors
X: PEAK1XFREQUENCY
Y: PEAK2XFREQUENCY

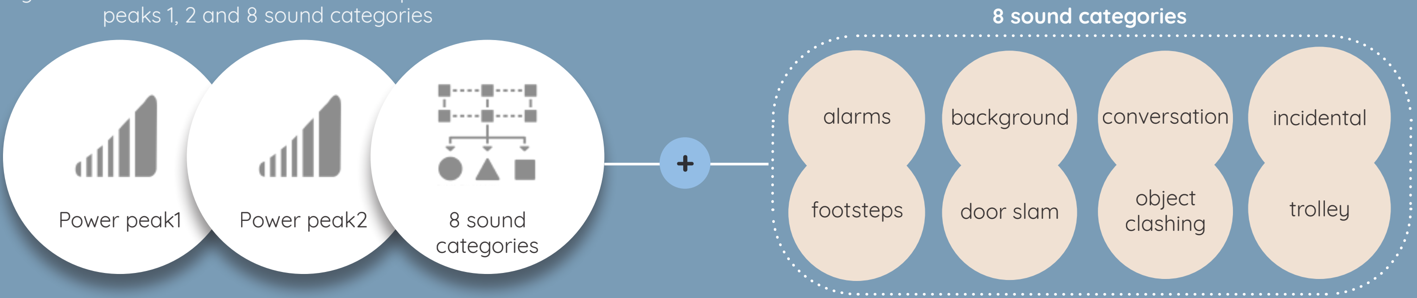
Classes Move to Front

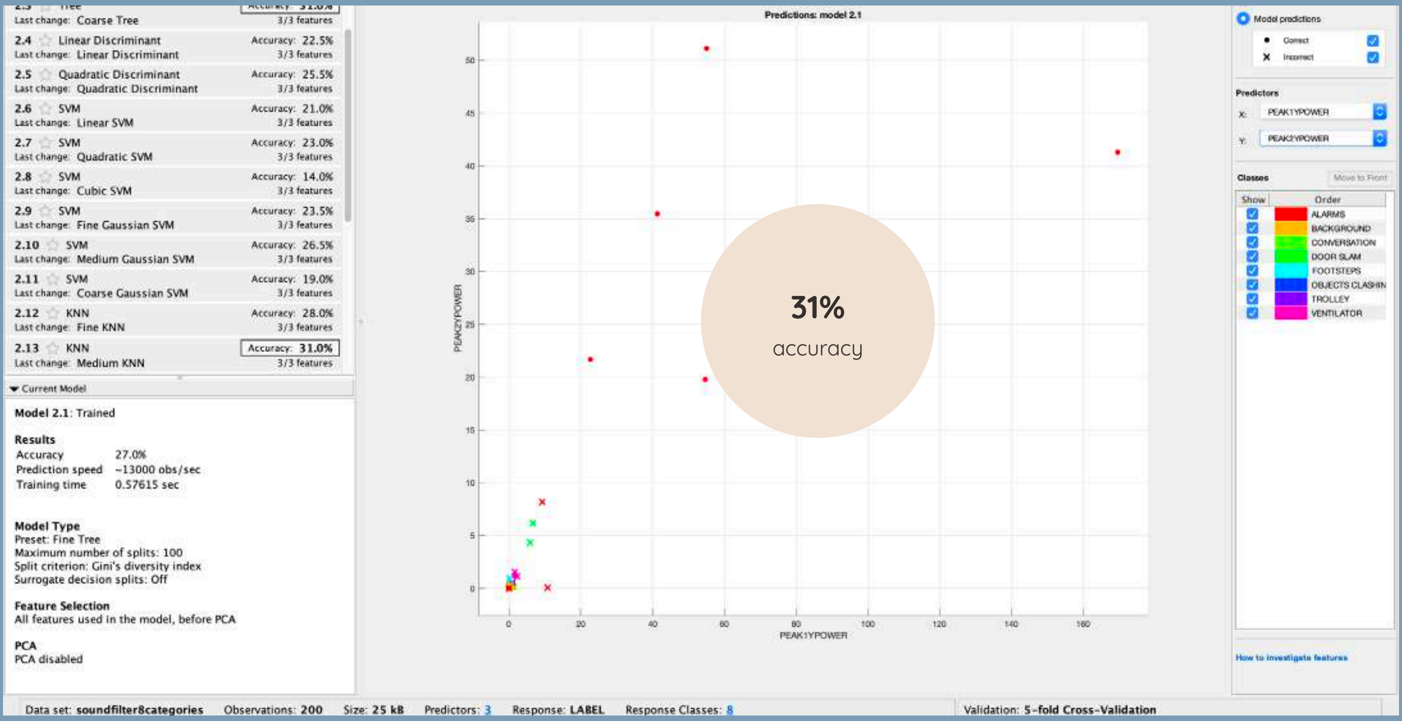
Show	Order
☑	🟥 ALARMS
☑	🟧 BACKGROUND
☑	🟩 CONVERSATION
☑	🟩 DOOR SLAM
☑	🟦 FOOTSTEPS
☑	🟦 OBJECTS CLASHIN
☑	🟪 TROLLEY
☑	🟪 VENTILATOR

How to investigate features

Data set: **soundfilter8categories** Observations: **200** Size: **25 kB** Predictors: 3 Response: **LABEL** Response Classes: 8 Validation: **5-fold Cross-Validation**

Figure 49. Sound classification filter with power peaks 1, 2 and 8 sound categories





Last change: Coarse Tree 3/3 features

2.4 Linear Discriminant Accuracy: 22.5%
Last change: Linear Discriminant 3/3 features

2.5 Quadratic Discriminant Accuracy: 25.5%
Last change: Quadratic Discriminant 3/3 features

2.6 SVM Accuracy: 21.0%
Last change: Linear SVM 3/3 features

2.7 SVM Accuracy: 23.0%
Last change: Quadratic SVM 3/3 features

2.8 SVM Accuracy: 14.0%
Last change: Cubic SVM 3/3 features

2.9 SVM Accuracy: 23.5%
Last change: Fine Gaussian SVM 3/3 features

2.10 SVM Accuracy: 26.5%
Last change: Medium Gaussian SVM 3/3 features

2.11 SVM Accuracy: 19.0%
Last change: Coarse Gaussian SVM 3/3 features

2.12 KNN Accuracy: 28.0%
Last change: Fine KNN 3/3 features

2.13 KNN Accuracy: 31.0%
Last change: Medium KNN 3/3 features

Current Model

Model 2.1: Trained

Results
Accuracy 27.0%
Prediction speed ~13000 obs/sec
Training time 0.57615 sec

Model Type
Preset: Fine Tree
Maximum number of splits: 100
Split criterion: Gini's diversity index
Surrogate decision splits: Off

Feature Selection
All features used in the model, before PCA

PCA
PCA disabled

Model predictions
● Correct ☑
× Incorrect ☑

Predictors
X: PEAK1YPOWER
Y: PEAK2YPOWER

Classes Move to Front

Show	Order
☑	🟥 ALARMS
☑	🟧 BACKGROUND
☑	🟩 CONVERSATION
☑	🟩 DOOR SLAM
☑	🟦 FOOTSTEPS
☑	🟦 OBJECTS CLASHIN
☑	🟪 TROLLEY
☑	🟪 VENTILATOR

How to investigate features

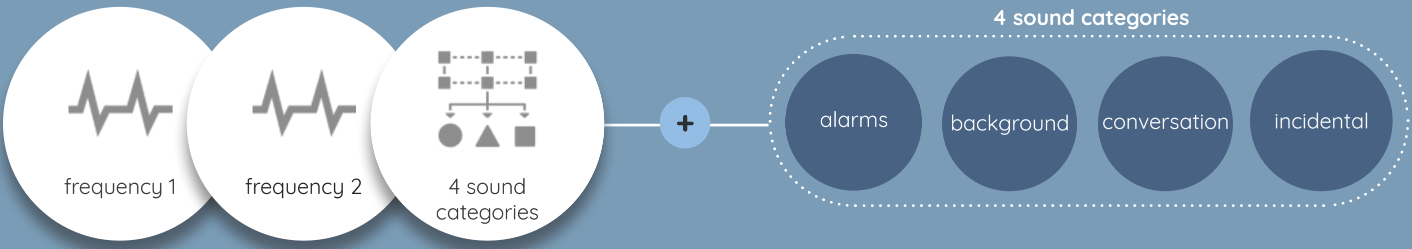
Data set: **soundfilter8categories** Observations: **200** Size: **25 kB** Predictors: 3 Response: **LABEL** Response Classes: 8 Validation: **5-fold Cross-Validation**

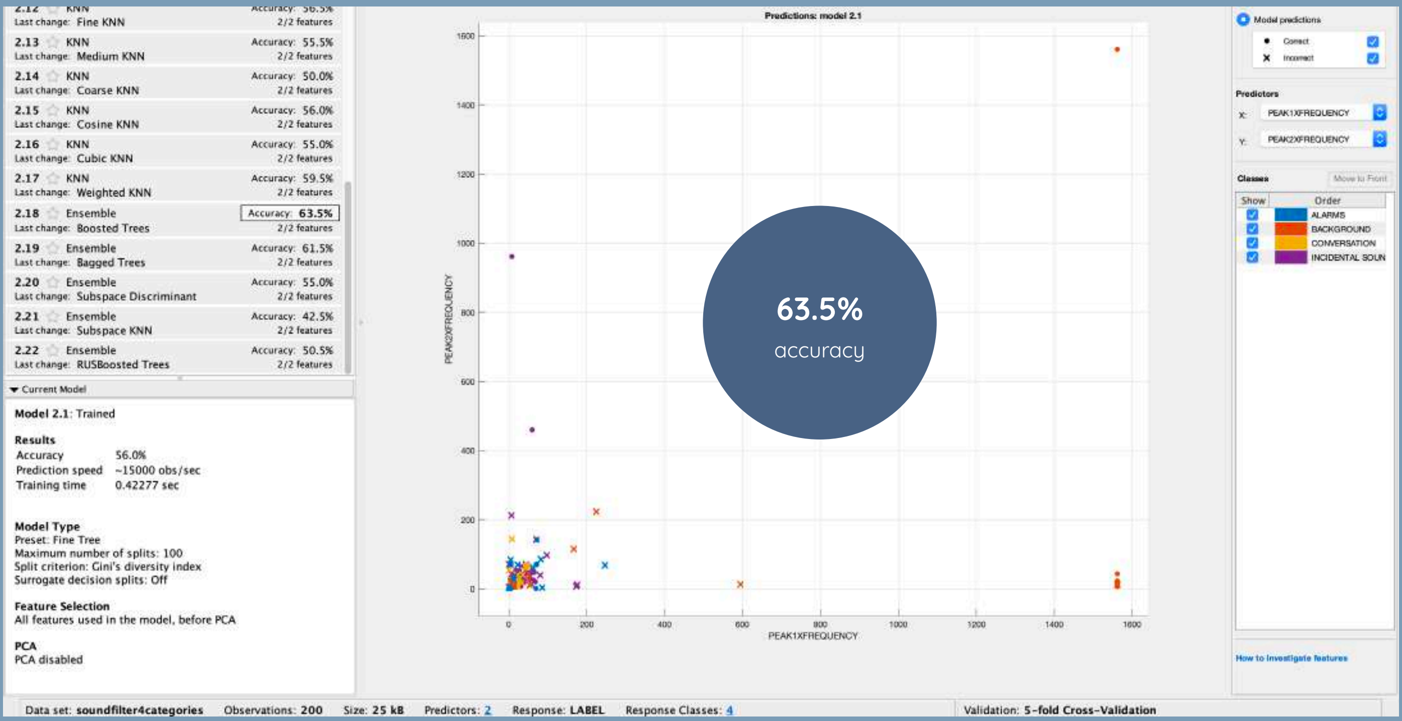
Sound classification filter creation
: with 4 categories

Using 4 sample categories

To improve the accuracy of the overall classification, 8 categories were combined into 4 categories of background, alarms, conversation, and incidental sounds. After using 4 sample group categorization, the background has 50 samples while alarms and conversation have 25 of each and sound samples of trolley dragging noise, door slam, objects clashing, and footsteps were merged into an incidental sound group with 100 samples. All in all, using Frequency-Frequency variables in training process showed a dominant accuracy of 63.5%, followed by Frequency-Power of 62% and Power-Power combination which rated 55.5% of accuracy. All in all, I decided to use 4 sample categorization than 8 sample categorization because of its better accuracy.

Figure 50. Sound classification filter with frequency peaks 1, 2 and 4 sound categories





2.12 KNN Accuracy: 56.5%
Last change: Fine KNN 2/2 features

2.13 KNN Accuracy: 55.5%
Last change: Medium KNN 2/2 features

2.14 KNN Accuracy: 50.0%
Last change: Coarse KNN 2/2 features

2.15 KNN Accuracy: 56.0%
Last change: Cosine KNN 2/2 features

2.16 KNN Accuracy: 55.0%
Last change: Cubic KNN 2/2 features

2.17 KNN Accuracy: 59.5%
Last change: Weighted KNN 2/2 features

2.18 Ensemble Accuracy: 63.5%
Last change: Boosted Trees 2/2 features

2.19 Ensemble Accuracy: 61.5%
Last change: Bagged Trees 2/2 features

2.20 Ensemble Accuracy: 55.0%
Last change: Subspace Discriminant 2/2 features

2.21 Ensemble Accuracy: 42.5%
Last change: Subspace KNN 2/2 features

2.22 Ensemble Accuracy: 50.5%
Last change: RUSBoosted Trees 2/2 features

▼ Current Model

Model 2.1: Trained

Results

Accuracy	56.0%
Prediction speed	~15000 obs/sec
Training time	0.42277 sec

Model Type
Preset: Fine Tree
Maximum number of splits: 100
Split criterion: Gini's diversity index
Surrogate decision splits: Off

Feature Selection
All features used in the model, before PCA

PCA
PCA disabled

● Model predictions
● Correct ☑
✕ Incorrect ☑

Predictors
X: PEAK1XFREQUENCY
Y: PEAK2XFREQUENCY

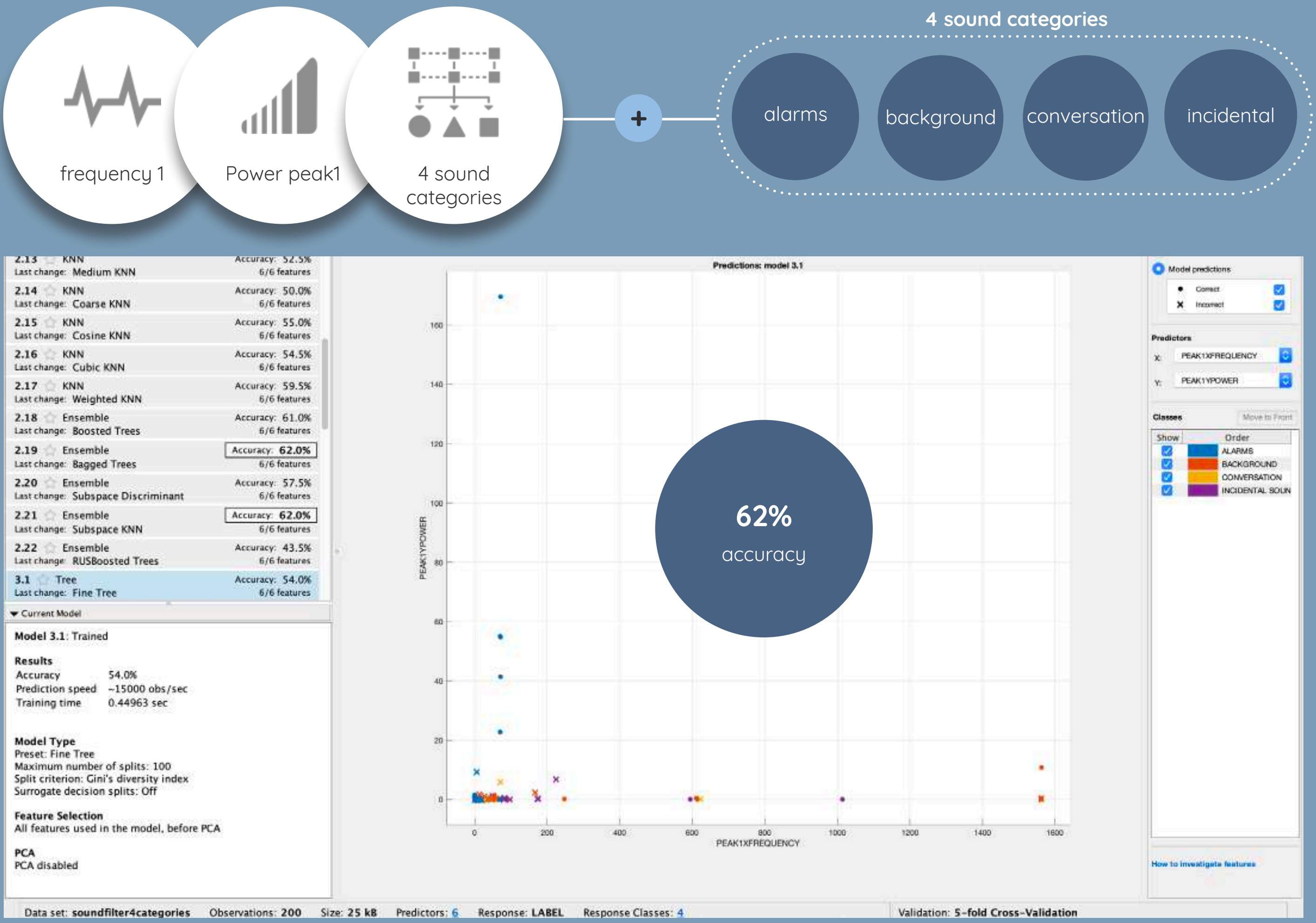
Classes Move to Front

Show	Order
☑	🟦 ALARMS
☑	🟥 BACKGROUND
☑	🟨 CONVERSATION
☑	🟪 INCIDENTAL SOUN

How to investigate features

Data set: **soundfilter4categories** Observations: **200** Size: **25 kB** Predictors: 2 Response: **LABEL** Response Classes: 4 Validation: **5-fold Cross-Validation**

Figure 51. Sound classification filter with frequency and power peaks and 4 sound categories



2.13 ☆ KNN — Accuracy: 52.5%
Last change: Medium KNN — 6/6 features

2.14 ☆ KNN — Accuracy: 50.0%
Last change: Coarse KNN — 6/6 features

2.15 ☆ KNN — Accuracy: 55.0%
Last change: Cosine KNN — 6/6 features

2.16 ☆ KNN — Accuracy: 54.5%
Last change: Cubic KNN — 6/6 features

2.17 ☆ KNN — Accuracy: 59.5%
Last change: Weighted KNN — 6/6 features

2.18 ☆ Ensemble — Accuracy: 61.0%
Last change: Boosted Trees — 6/6 features

2.19 ☆ Ensemble — Accuracy: **62.0%**
Last change: Bagged Trees — 6/6 features

2.20 ☆ Ensemble — Accuracy: 57.5%
Last change: Subspace Discriminant — 6/6 features

2.21 ☆ Ensemble — Accuracy: **62.0%**
Last change: Subspace KNN — 6/6 features

2.22 ☆ Ensemble — Accuracy: 43.5%
Last change: RUSBoosted Trees — 6/6 features

3.1 ☆ Tree — Accuracy: 54.0%
Last change: Fine Tree — 6/6 features

▾ Current Model

Model 3.1: Trained

Results

Accuracy	54.0%
Prediction speed	~15000 obs/sec
Training time	0.44963 sec

Model Type
Preset: Fine Tree
Maximum number of splits: 100
Split criterion: Gini's diversity index
Surrogate decision splits: Off

Feature Selection
All features used in the model, before PCA

PCA
PCA disabled

○ Model predictions
● Correct ☑
✕ Incorrect ☑

Predictors
X: PEAK1XFREQUENCY
Y: PEAK1YPOWER

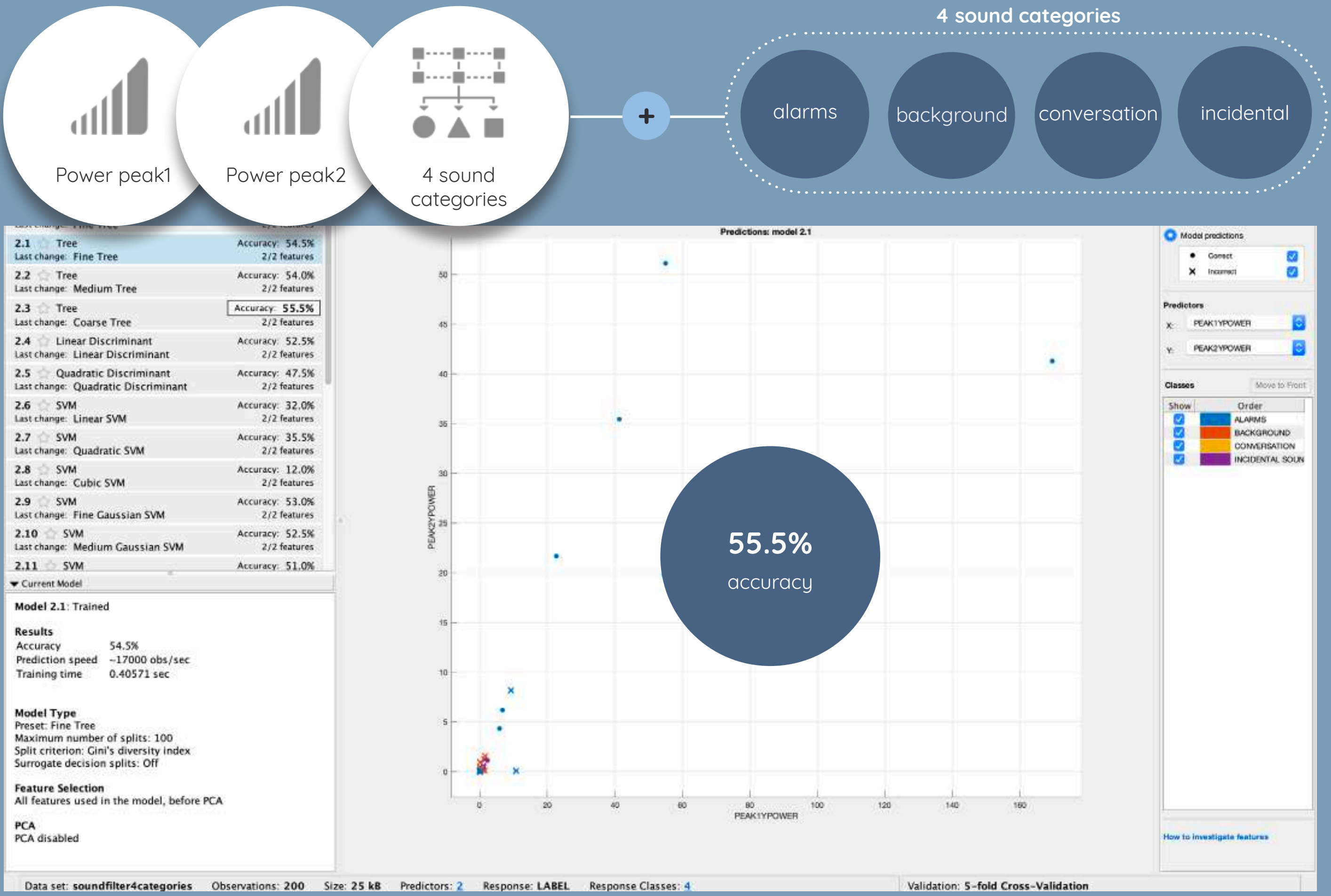
Classes — Move to Front

Show	Order
☑	ALARMS
☑	BACKGROUND
☑	CONVERSATION
☑	INCIDENTAL SOUN

How to investigate features

Data set: **soundfilter4categories** Observations: **200** Size: **25 kB** Predictors: 6 Response: **LABEL** Response Classes: 4 Validation: **5-fold Cross-Validation**

Figure 52. Sound classification filter with power peaks 1, 2 and 4 sound categories



2.1 ☆ Tree — Accuracy: 54.5%
Last change: Fine Tree — 2/2 features

2.2 ☆ Tree — Accuracy: 54.0%
Last change: Medium Tree — 2/2 features

2.3 ☆ Tree — Accuracy: **55.5%**
Last change: Coarse Tree — 2/2 features

2.4 ☆ Linear Discriminant — Accuracy: 52.5%
Last change: Linear Discriminant — 2/2 features

2.5 ☆ Quadratic Discriminant — Accuracy: 47.5%
Last change: Quadratic Discriminant — 2/2 features

2.6 ☆ SVM — Accuracy: 32.0%
Last change: Linear SVM — 2/2 features

2.7 ☆ SVM — Accuracy: 35.5%
Last change: Quadratic SVM — 2/2 features

2.8 ☆ SVM — Accuracy: 12.0%
Last change: Cubic SVM — 2/2 features

2.9 ☆ SVM — Accuracy: 53.0%
Last change: Fine Gaussian SVM — 2/2 features

2.10 ☆ SVM — Accuracy: 52.5%
Last change: Medium Gaussian SVM — 2/2 features

2.11 ☆ SVM — Accuracy: 51.0%

▾ Current Model

Model 2.1: Trained

Results

Accuracy	54.5%
Prediction speed	~17000 obs/sec
Training time	0.40571 sec

Model Type
Preset: Fine Tree
Maximum number of splits: 100
Split criterion: Gini's diversity index
Surrogate decision splits: Off

Feature Selection
All features used in the model, before PCA

PCA
PCA disabled

○ Model predictions
● Correct ☑
✕ Incorrect ☑

Predictors
X: PEAK1YPOWER
Y: PEAK2YPOWER

Classes — Move to Front

Show	Order
☑	ALARMS
☑	BACKGROUND
☑	CONVERSATION
☑	INCIDENTAL SOUN

How to investigate features

Data set: **soundfilter4categories** Observations: **200** Size: **25 kB** Predictors: 2 Response: **LABEL** Response Classes: 4 Validation: **5-fold Cross-Validation**

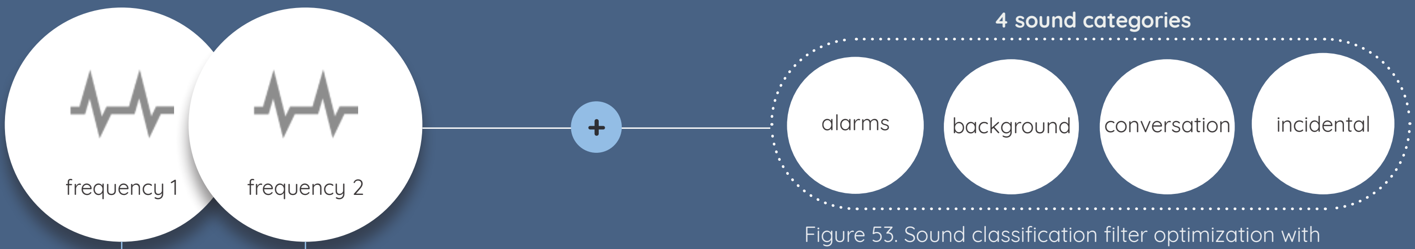
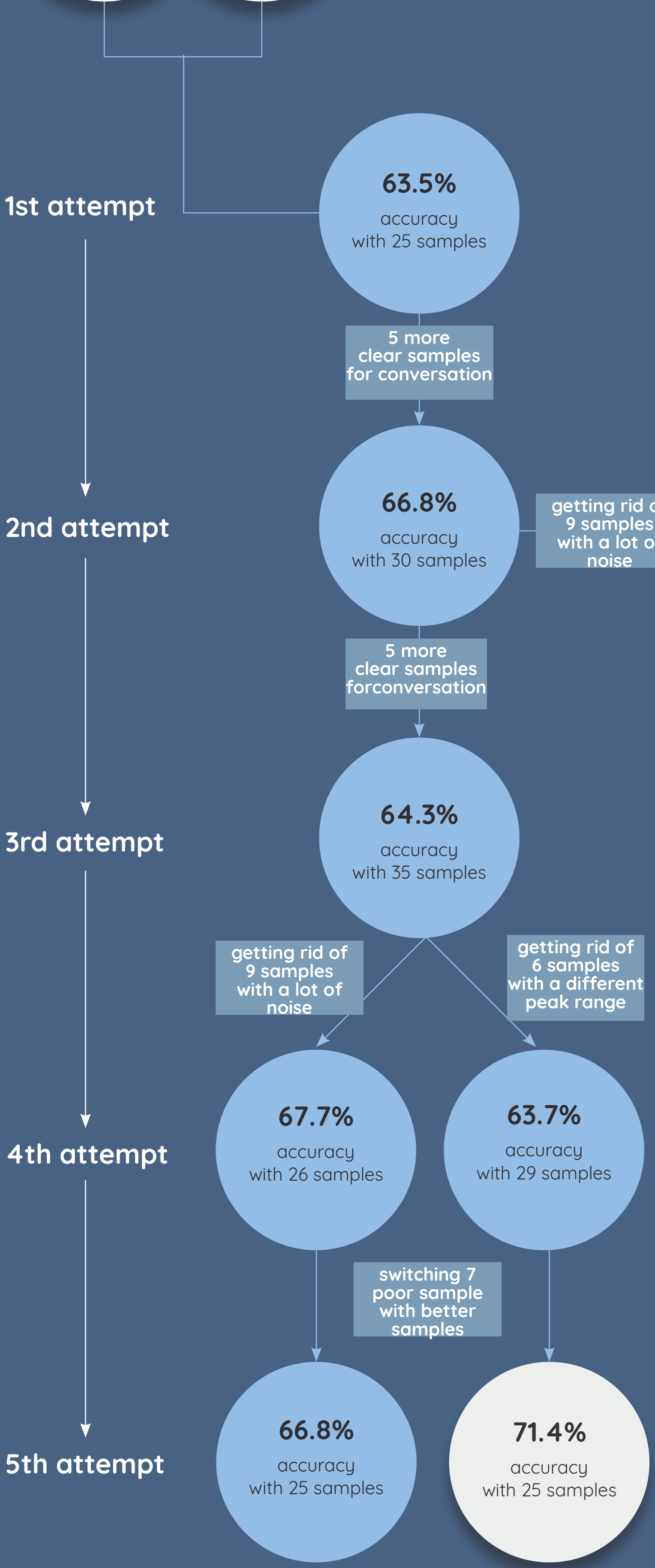


Figure 53. Sound classification filter optimization with frequency peaks 1, 2 and 4 sound categories

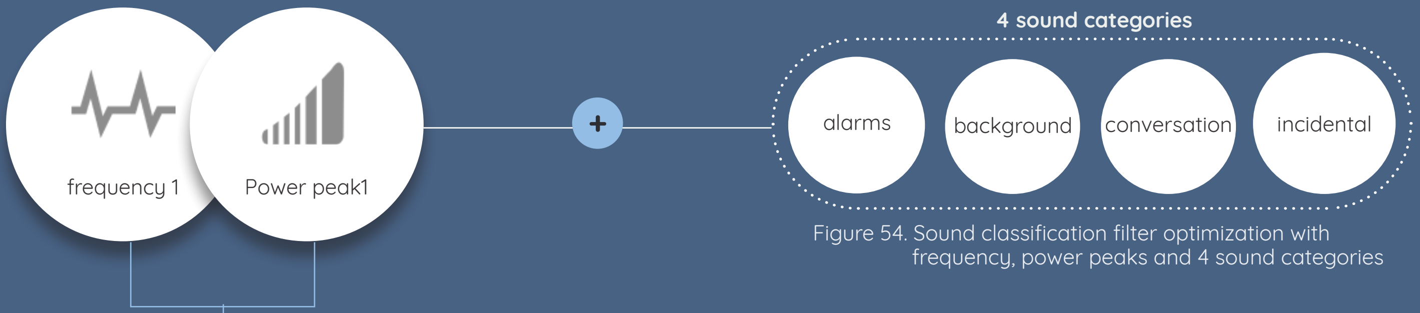


Sound classification filter optimization

Optimization for improving the filter accuracy

To improve the reliability of the algorithm in the classification system, classification filter optimization was done. In this process, I decided to use 4 sample categorization instead of 8 sample categorization because the elaborated classification was not needed and 4 sample categorization showed the better overall accuracy, so I used it as a starting point of the optimization process.

Optimization was conducted by substituting poor quality sample with better ones, which well represents the group. Also, the sample size was enlarged in some groups so that the sensor can reflect the sound group better.



frequency 1

Power peak1

+

4 sound categories

alarms

background

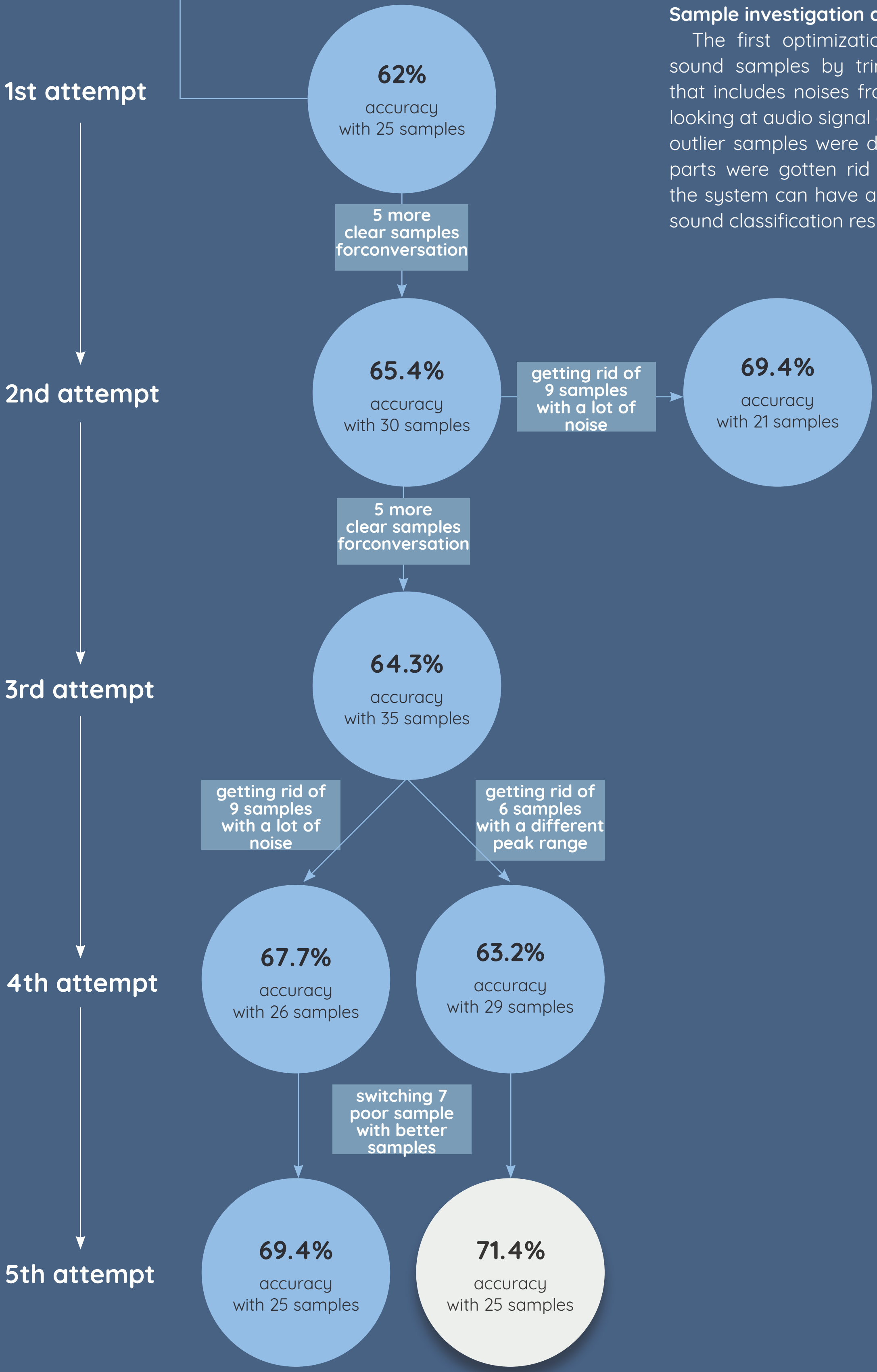
conversation

incidental

Figure 54. Sound classification filter optimization with frequency, power peaks and 4 sound categories

Sample investigation and re-sampling

The first optimization process was resampling sound samples by trimming parts of recordings that includes noises from other sound sources. By looking at audio signal analysis results from Matlab, outlier samples were discovered, and unnecessary parts were gotten rid of from recordings so that the system can have a more accurate filter for the sound classification result.



1st attempt

62% accuracy with 25 samples

5 more clear samples forconversation

2nd attempt

65.4% accuracy with 30 samples

getting rid of 9 samples with a lot of noise

69.4% accuracy with 21 samples

5 more clear samples forconversation

3rd attempt

64.3% accuracy with 35 samples

getting rid of 9 samples with a lot of noise

getting rid of 6 samples with a different peak range

4th attempt

67.7% accuracy with 26 samples

63.2% accuracy with 29 samples

switching 7 poor sample with better samples

5th attempt

69.4% accuracy with 25 samples

71.4% accuracy with 25 samples

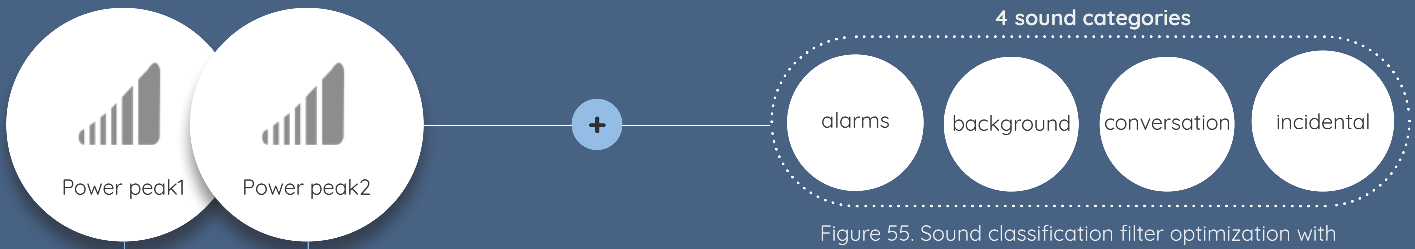
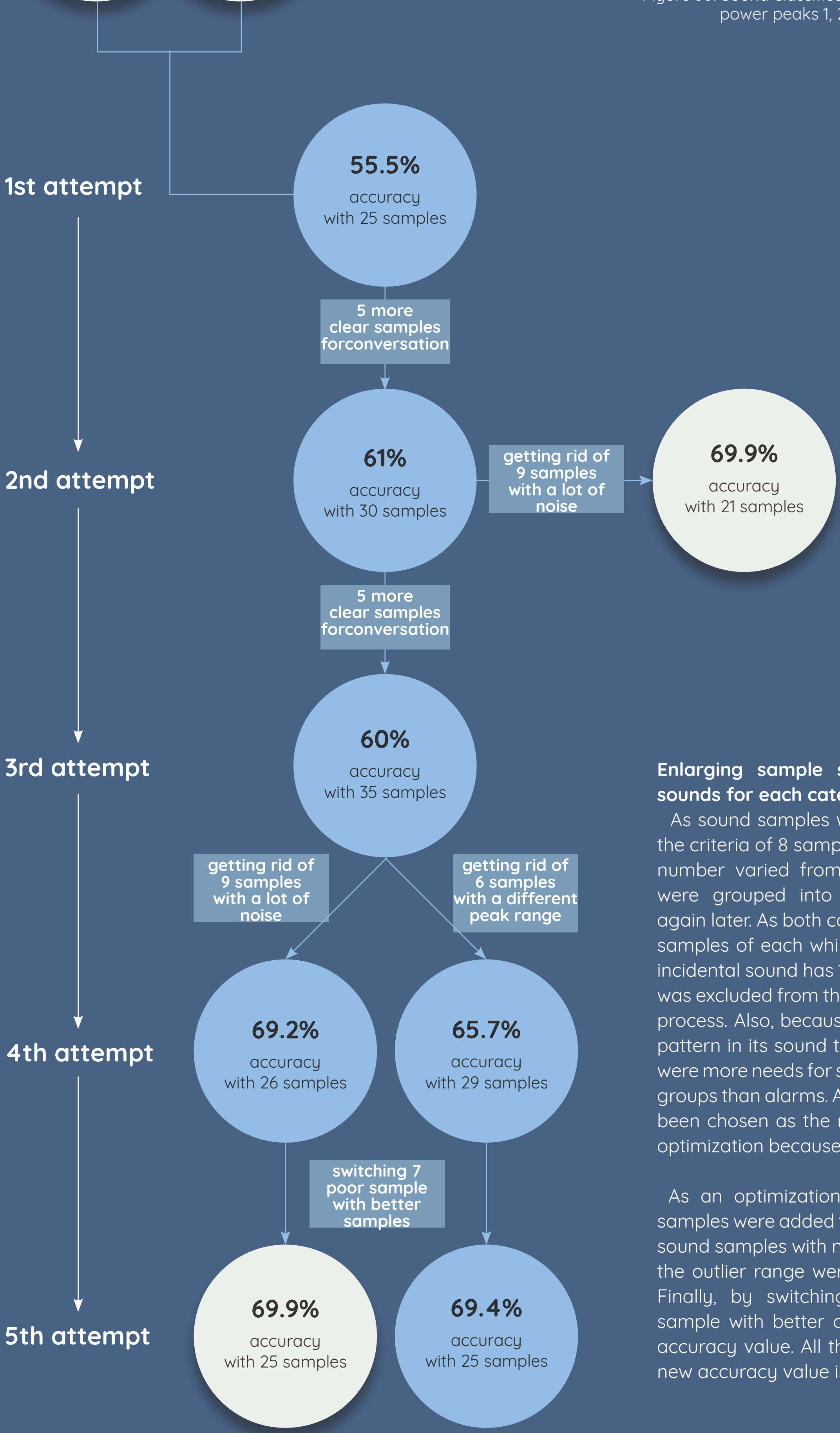


Figure 55. Sound classification filter optimization with power peaks 1, 2 and 4 sound categories



Enlarging sample size by adding more clear sounds for each category

As sound samples were originally collected under the criteria of 8 sample categorization, each sample number varied from 25 to 100 because samples were grouped into four samples categorization again later. As both conversation and alarm have 25 samples of each while the background has 50 and incidental sound has 100, the incidental sound group was excluded from the sample number enlargement process. Also, because alarms have a more regular pattern in its sound than other sound groups, there were more needs for sample improvements for other groups than alarms. As a result, the conversation has been chosen as the main target of the sound filter optimization because of its irregular sound patterns.

As an optimization procedure, five clear sound samples were added to filters for two times, and nine sound samples with noises and six sound samples in the outlier range were gotten rid of from samples. Finally, by switching seven poor sound quality sample with better ones, the result showed better accuracy value. All the process of re-sampling and new accuracy value is available from Figure 53-55.

Indoor positioning methods

In this chapter, indoor positioning will be introduced as a part of the project scope. First, GPS technology was looked into, but it turned out that GPS is a position detection technology based on satellite signals so that GPS cannot be used indoors as the satellite signals cannot penetrate walls of buildings. Also, the accuracy range of GPS is not suitable for indoor use. Therefore, various indoor positioning solutions were studied to find the best option for Cacophony Mapper.

Visual markers

Using a visual marker was considered as an indoor positioning method in the first stage. A method using visual markers uses a camera recording of surroundings for indoor positioning analysis. The camera detects specific visual markers in the environment and determines the user location based on the database from previous recordings. It is cost efficient as it only needs a functioning camera which will be attached to nurses for position detection. However, the hospital is an environment where people are sensitive about confidentiality and privacy; therefore, the option was excluded from the user scenario. Using a blur or a distortion filter was considered to apply to the camera, but acceptance of clinicians and patients remained as the biggest problem.

Beacon installments

Installing beacons in the ICU was contemplated because of its easy deployment and quick connection with Bluetooth. Also, many beacons adopt different combinations of built-in sensors, such as accelerometer, temperature sensor, ambient light sensor, magnetometer, and pressure sensor, they could be combined with the further concept of Cacophony Mapper. Also, since most beacons provide area sectioning function, it is possible to define each zone in the ICU with beacons. By combining sound data and position data, it is possible to create a sound heat map as a form of sound analysis. Also, because beacon’s various analysis options, such as the name of the visitor, the number of visits, and duration of stay, there is room to combine more information proactively with the Cacophony Mapper concept later.

Magnetic positioning

When it goes for magnetic positioning, the most significant benefit is that there is no need to install any hardware in the environment as every place has its unique magnetic fingerprint. Also, it has 1 to 2 meters of accuracy, and it is exact and cost-effective method at the same time. However, the downside of it is that the hospital has various metallic equipment, that can hinder detection based on a magnetic field. Also, since many pieces of machinery emit electromagnetic waves in the ICU, it can be crucial to the accuracy of the magnetic positioning method, so magnetic positioning was excluded from the implementation option, too.

Wi-fi (wireless fidelity) signals

The Wi-fi signals can be used as an indoor positioning method. However, it was excluded from the implementation option because of its limited proximity range of 5-15 meters, which exceeds the width of the ICU corridor.

Li-fi (light fidelity)

Li-fi uses the signals from LED lighting positioned on ceilings, so the data transmission through light emission can be changed into a form which is non-visible for human eyesight. Since it does not transmit with electromagnetic waves, Li-fi is often used for aircraft cabins and hospital environment. However, Li-fi was excluded from the embodiment scenario because B2B service providers mainly provide the Li-fi technology in more significant installation volume. Also, those service providers provide not only the Li-fi devices but also the installment plan as a package, so it was not an available option for this graduation project.

How inner positioning can be used?

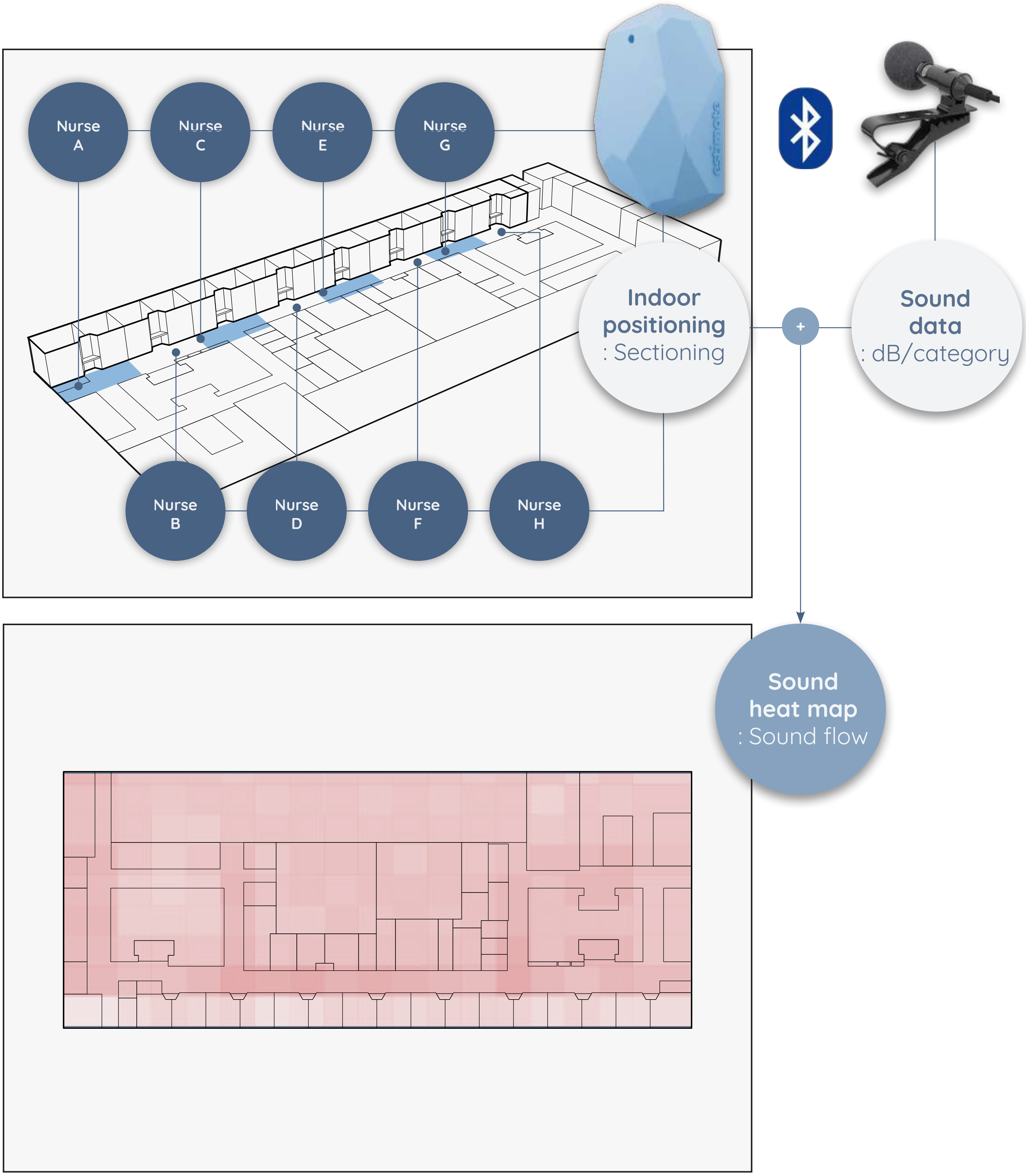


Figure 56. Idea to combine Inner positioning and sound data

Sound heat map

From the previous technological research, beacons were introduced to Cacophony Mapper concept to find the indoor position information of ICU nurses. Since beacons offer sectioning function, they detect nurses' movement and provide fluid position information of their flow. Sound decibel and sound category will also be continuously collected through the sound sensor with a Cacophony Mapper device, while position information is collected. Thus, the heat map will be created with multiple data that nurses send out real-time.

How beacons can be distributed in the ICU?

Distribution of the beacons was contemplated because the proximity range of them could be set up differently for each device. Therefore, the range of the sectioning was contemplated by looking at the layout of ICU units. As can be seen from Figure 57, one floor has several different departments (which will be referred to as units in this report). In this chapter, the distribution of beacons was planned based on the unit layout of Erasmus MC. After calculating the total length of each unit's corridor and the distance between desks, which is basically a one working station, the range of the area has a width of 4.8m and height of 3.3m. In this scenario, if someone enters into Zone A from Zone B, beacons will detect this movement, and those positioning information will be sent to the system. As sound and stress level detection will be done in the meantime, all collected information will be integrated into the system, and a heat map will be provided as one of data analysis.

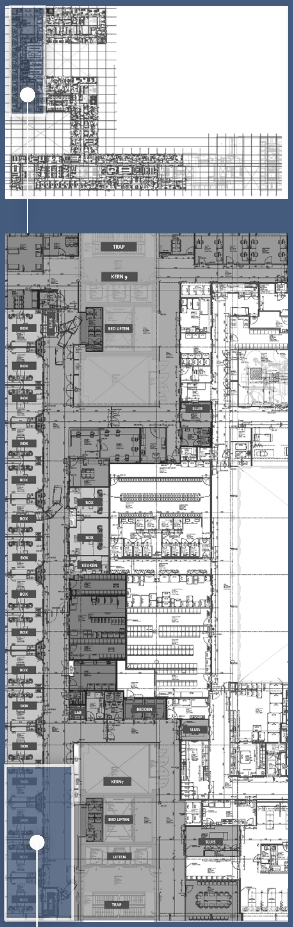
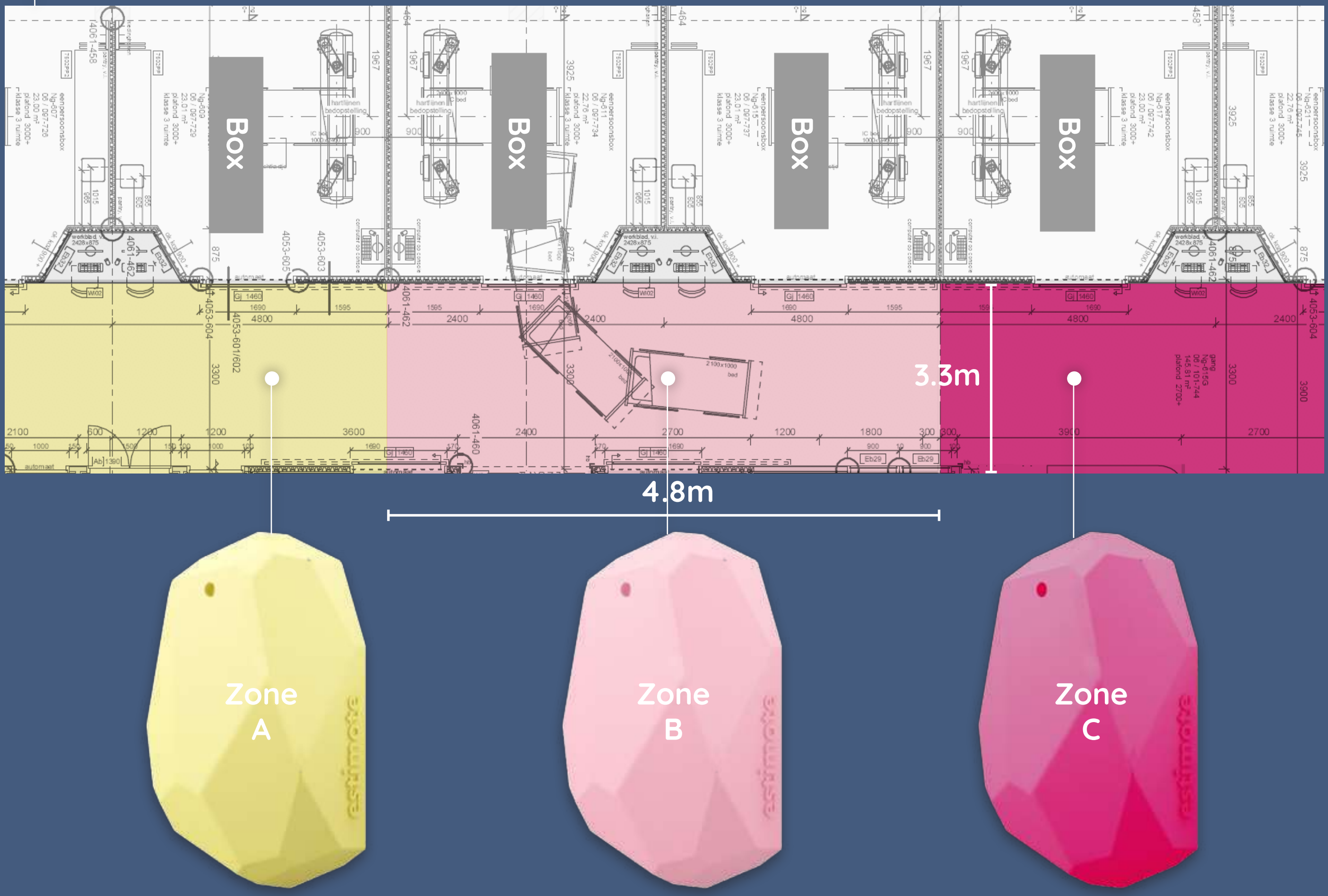


Figure 57. Beacons deployment plan for Erasmus MC



Cacophony Mapper concept

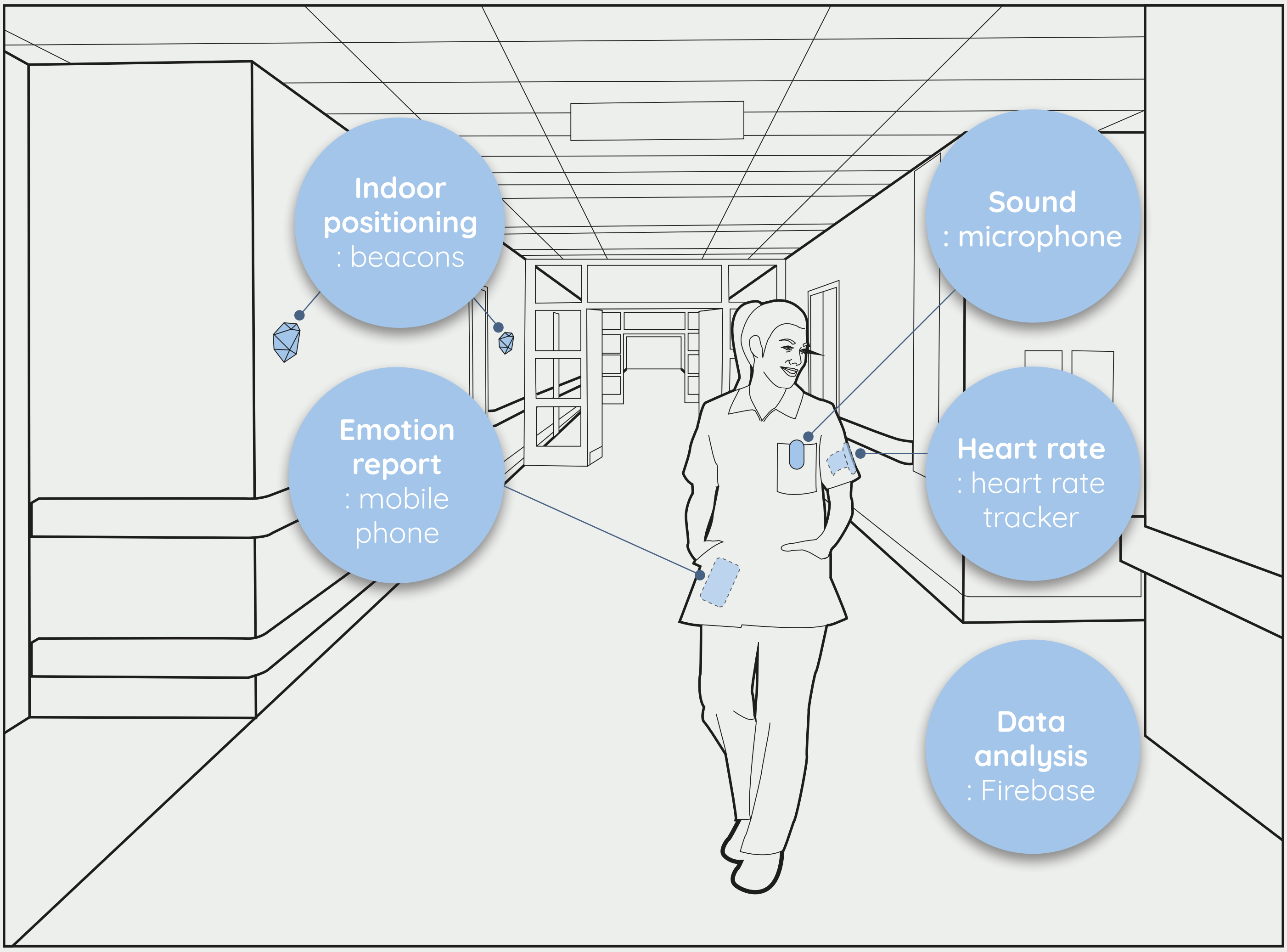


Figure 58. Cacophony mapper concept

Cacophony mapper concept involves mainly five different sub-components in the system, a microphone for the sound collection, a heart rate tracker for the objective stress level assessment, beacons for indoor positioning, a mobile phone for emotion report function as subjective stress assessment tool and data analysis, and Firebase for data summary and further data analysis.

How product and system will be intertwined?

As can be seen from Figure 58 and 59 for product and system architecture, there will be a heart rate tracker, which will be attached to the nurses' outer upper arm. For heart rate collection, PPG heart sensor will be used in a tracker, and collected data will be delivered to the system through Bluetooth. Also, there will be a microphone component which will be clipped to nurses' chest pocket, which uses Bluetooth for the data delivery. Also, there will be a mobile phone for the nurses' emotion report, which gives data analysis at the same time. Furthermore, beacons will be attached to ICU walls, providing indoor position sectioning. Finally, Sound, heart rate, emotion report, and indoor positioning data set will be available from Firebase altogether, which functions as a platform for collected data set for Cacophony Mapper. Further data analysis will be done within the data set collected from Firebase. Data analysis screens for mobile were designed differently for nurse mode, and researcher mode and further explanation is available in chapter 5.9 and 5.10.

Product & system architecture

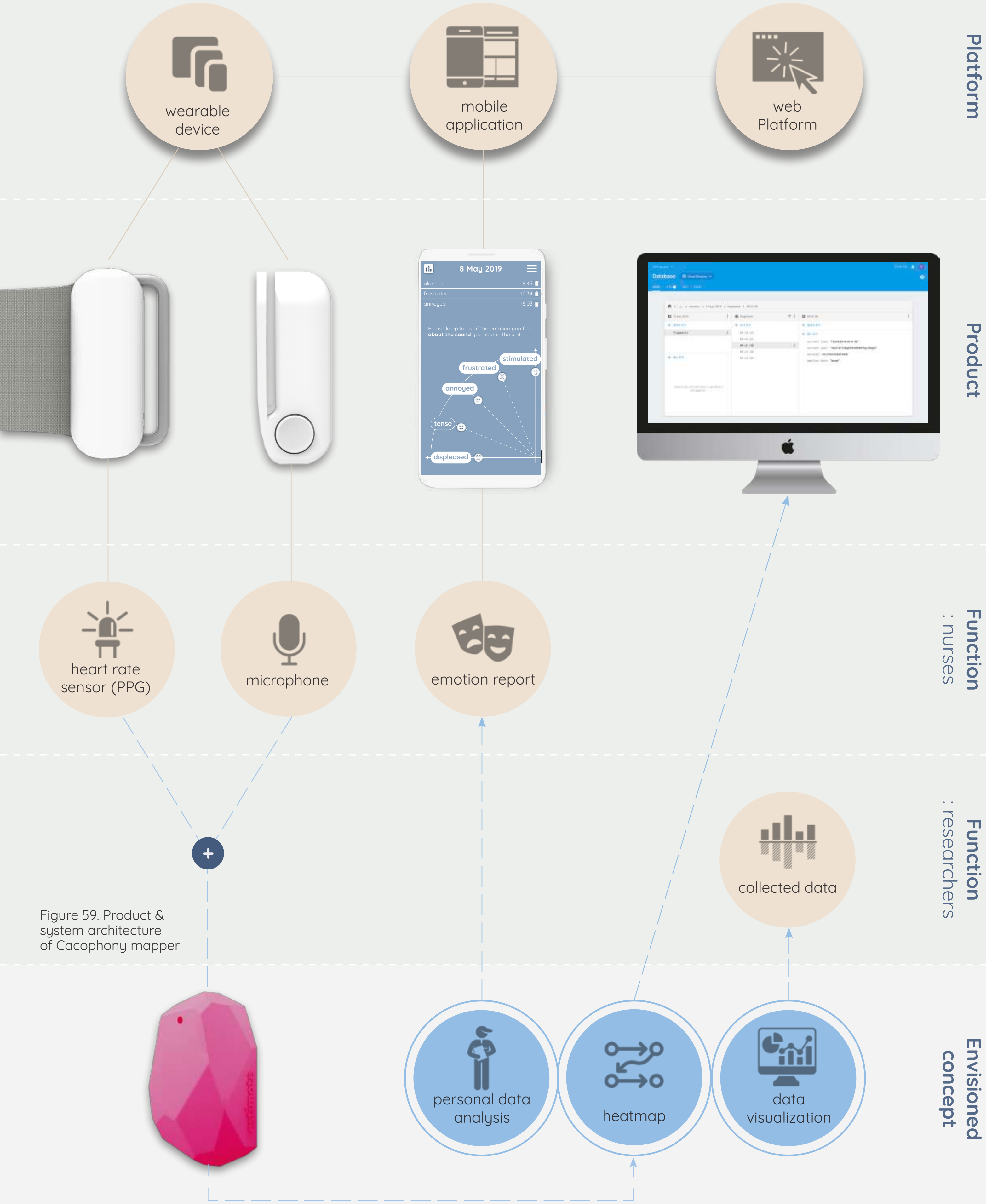


Figure 59. Product & system architecture of Cacophony mapper

Product properties

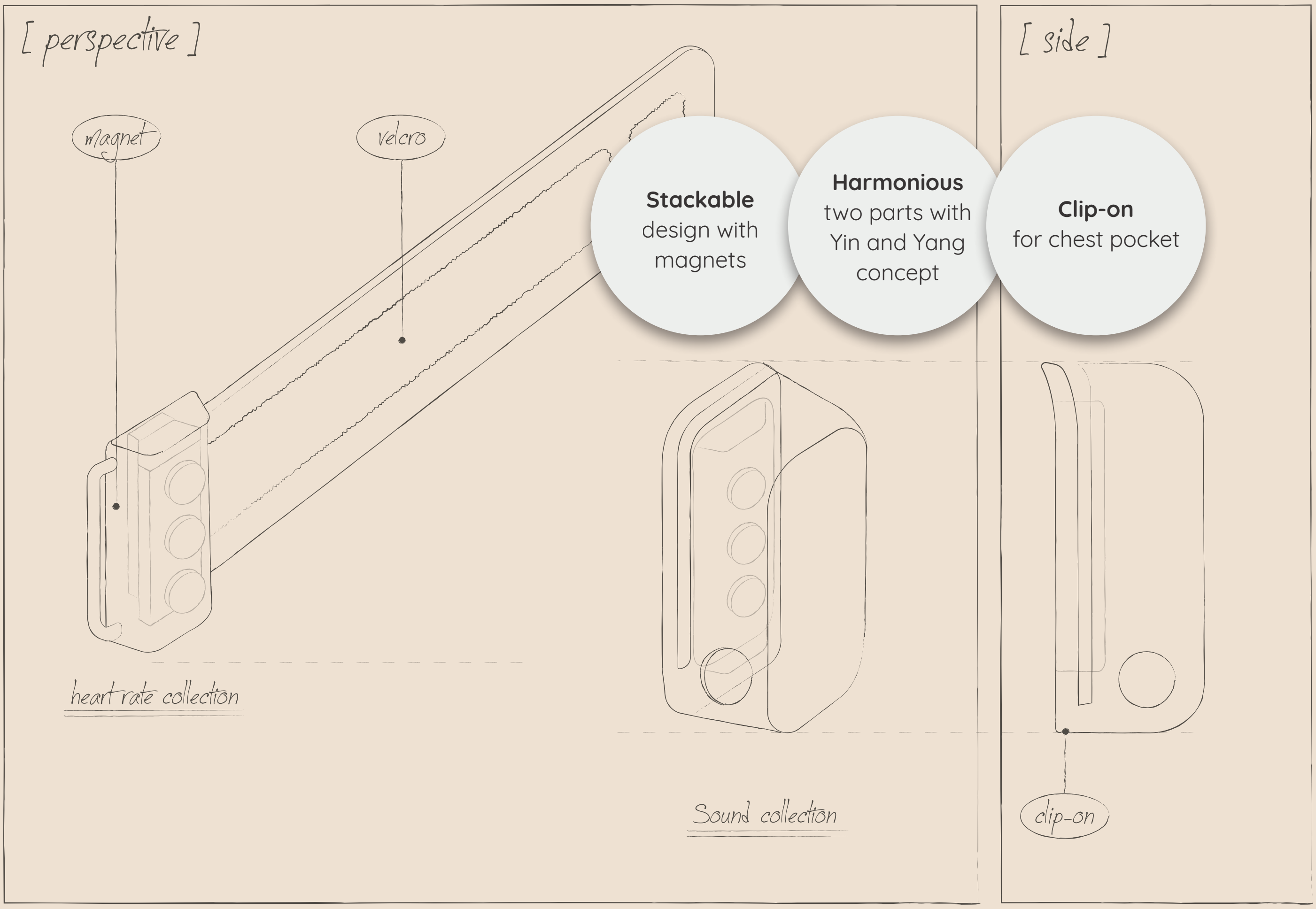

[perspective]
magnet
velcro
heart rate collection
Sound collection
[side]
clip-on
Stackable design with magnets
Harmonious two parts with Yin and Yang concept
Clip-on for chest pocket

Figure 60. Cacophony mapper product idea sketch

In this chapter, the distribution of the sound sensor and the heart rate collection component will be introduced as a part of hardware properties. For the quality recording for a precise sound classification, a microphone should be located outside of nurses' clothing while heart rate can only be accurately collected when a device touch users' skin. To overcome this structural challenge, multiple user scenarios have been considered and various versions of sketches were made.

As a result, I decided to separate the heart rate tracker and the sound collection part, so the quality sound and the precise heart rate data can be collected. For the compact storage and the harmonious look, an idea of designing a stackable set was chosen as a hardware design concept, and magnets were embedded to complete the idea.

For easy use, heart rate tracker part has a look of a sports armband with velcro, so users can easily pick up how to wear it without further instruction. Also, an interview (see Appendix E) was conducted with a nurse with 3D-printed draft hardware, and many insights were gained through that interview, and further ideation based on this interview process is available in chapter 4.4.

Working scenario

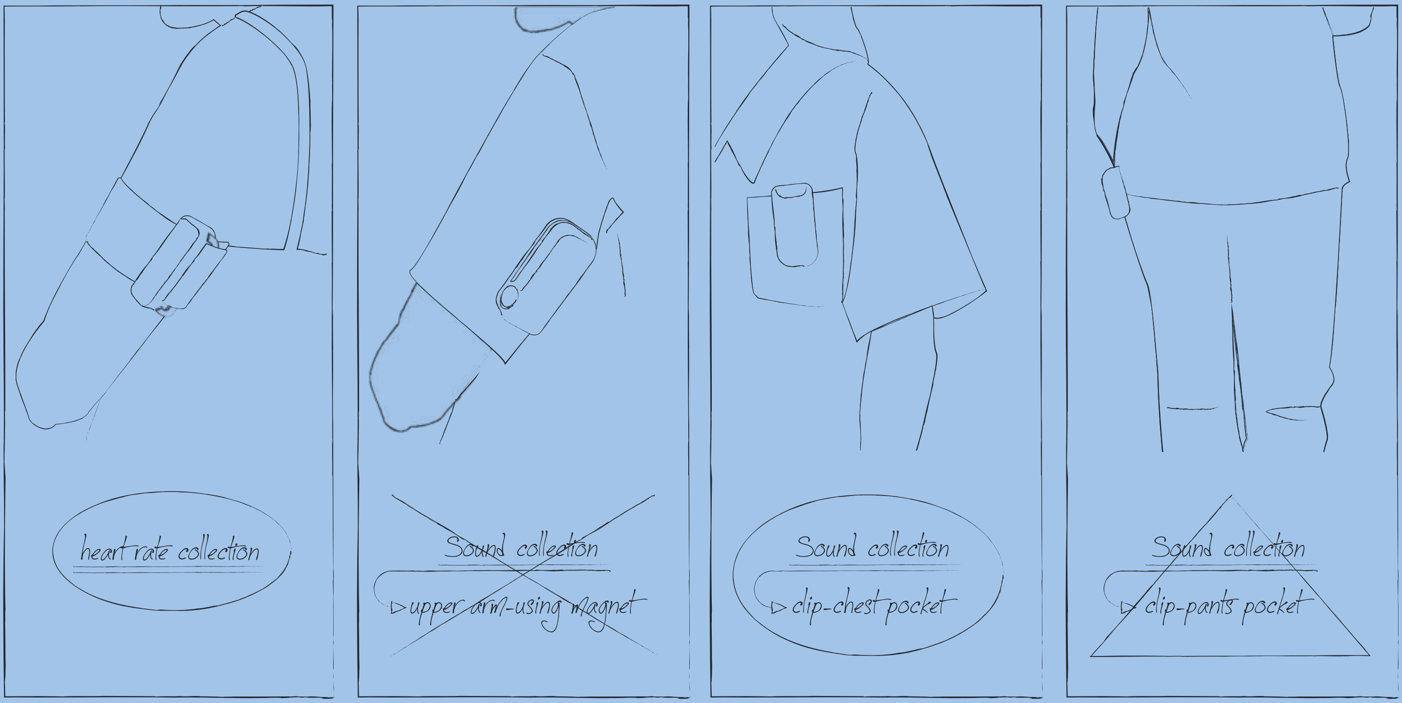


Figure 61. Cacophony mapper product working scenario

For the heart rate collection, the armband format has been chosen based on its excellent acceptance from its familiar usability. In this chapter, the positioning of the sound collection part will be mainly discussed.

As an idea, applying the sound device on the sleeve over the heart rate detection device within magnets, was considered. However, this idea was excluded from the option after having an interview with a nurse, which can be found in Appendix F. According to the interview, wearing something over the sleeve is forbidden for nurses in the hospital because of the possibility of cross-infection, even though the chest pocket area can be freely used, even over the clothing area.

Therefore, I came up with the second scenario of clipping the sound device on the nurses' chest pocket. Even in this option, applying magnet in two different parts, and clicking together will stay as a part of the concept. This is mainly because having three different devices, including a mobile phone, for one system can be complicated for users and embedding magnets inside and storing devices together can be one right solution for a storing issue.

Clipping the sound device on the pants pocket was considered as an option too, but it was soon excluded from the user scenario because of its poor recording quality from bustling sound from the walk. Also, if users need to sit or bend, there is a high chance that the clipping part will be damaged in the long run. Furthermore, the far distance from users' ear area was one reason that this option was excluded from the user scenario because sound analysis from the sound collection system and nurses' emotion report would show more different tendency if they are physically far, which can be a crucial factor for misleading research result.

Form exploration

Twelve forms were modeled and printed before deciding on shapes of the final prototype. Shapes around the range of 15*25*50mm for sound sensor part and 18*25*60mm for heart rate detection part were modeled and printed. In this process, the size and distribution of inner components were considered, as well as the width of the elastic armband and the size of coin magnets. Rounded shapes were mainly studied because the device should touch users' skin for a long time.

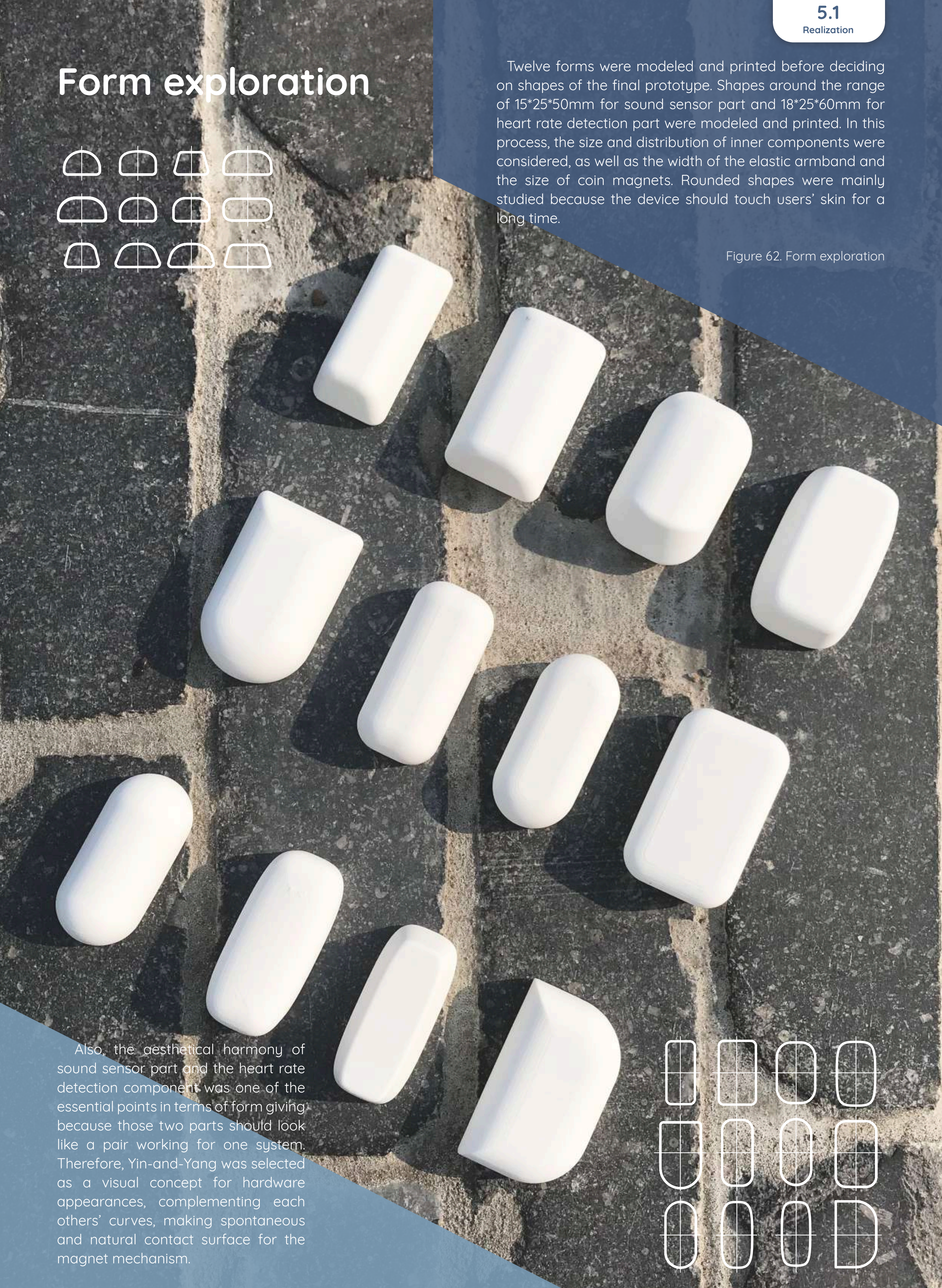


Figure 62. Form exploration

Also, the aesthetical harmony of sound sensor part and the heart rate detection component was one of the essential points in terms of form giving because those two parts should look like a pair working for one system. Therefore, Yin-and-Yang was selected as a visual concept for hardware appearances, complementing each others' curves, making spontaneous and natural contact surface for the magnet mechanism.

3D modeling

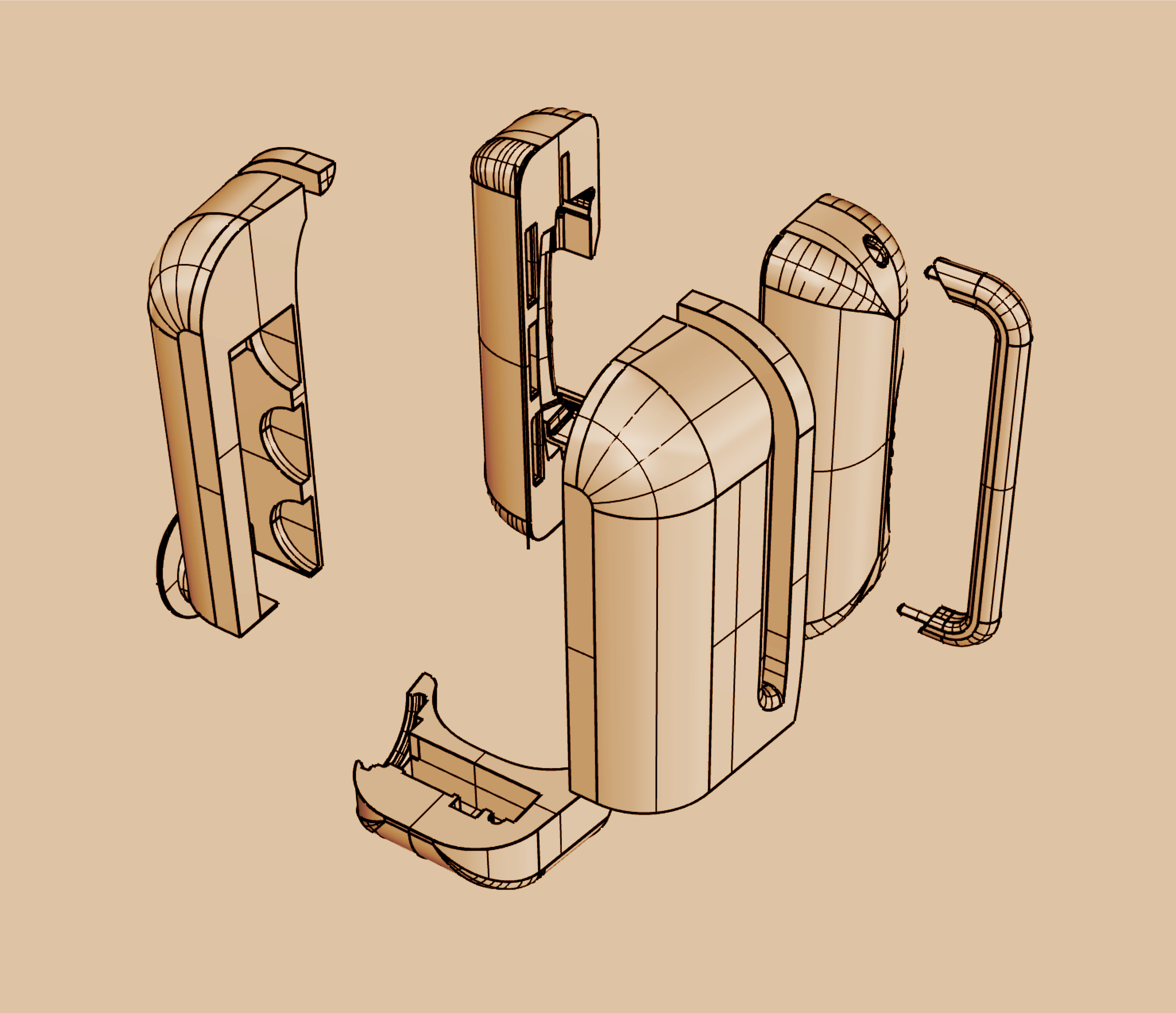


Figure 63. 3D-modeled components

3D models were created in Rhino Ceros program within the scale criteria, as mentioned in chapter 5.1.

First, the mass for inner components was created for fastening inner components effectively, so existing inner components can be fit into the model without too much or too tight gaps. After successfully testing holding parts for inner components by several trial and errors, outer shells were created, and the inner fastening part was subtracted from the outer shell so that it can function as a product casing. In this phase, spaces for magnets were modeled as well as the space for armband belting part and an on/off button for the sound sensor. Also, products were divided into several pieces that they can be easily 3d printed and assembled later.

Rendering

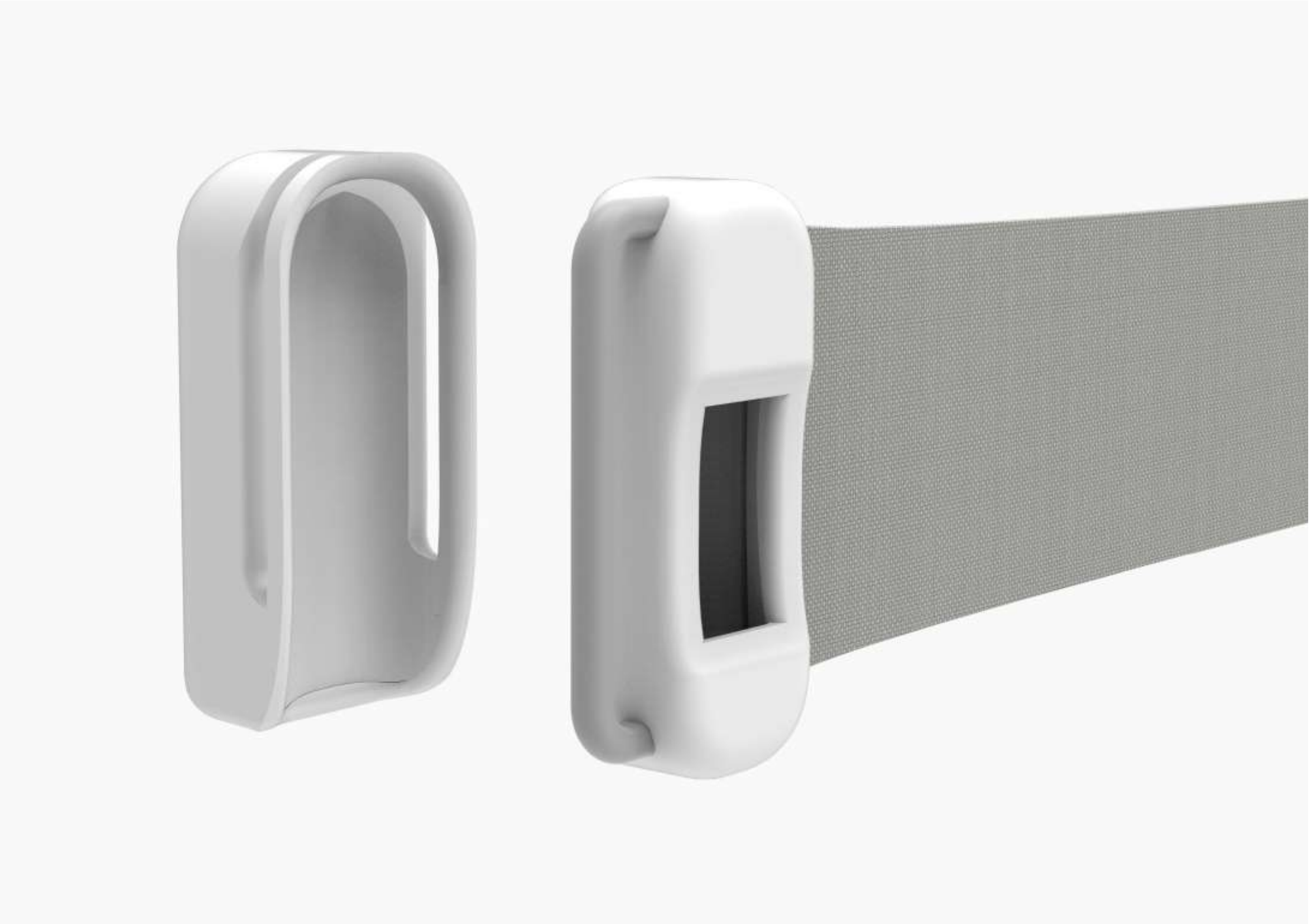


Figure 64. 3D-rendered components

3D models were made, mainly focusing on two parts for sound collection, and heart rate detection can be stackable with a minimal structure. As there should be inner components for electronics and magnets, tolerance gaps were applied to holding structures for inner components. The inner curve for sound collection part and outer curve for heart rate detection part were designed in line with each other, so they can be clicked together without any interfering surface, allowing a more harmonious look. Also, soft curves were applied to edges for those devices, so that users would not feel any discomfort because of angles on the edges.

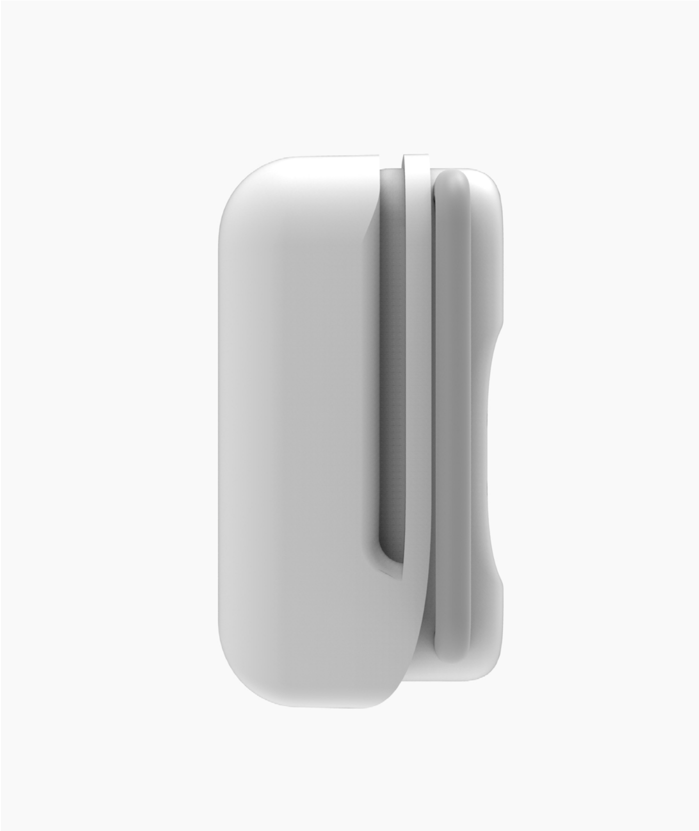
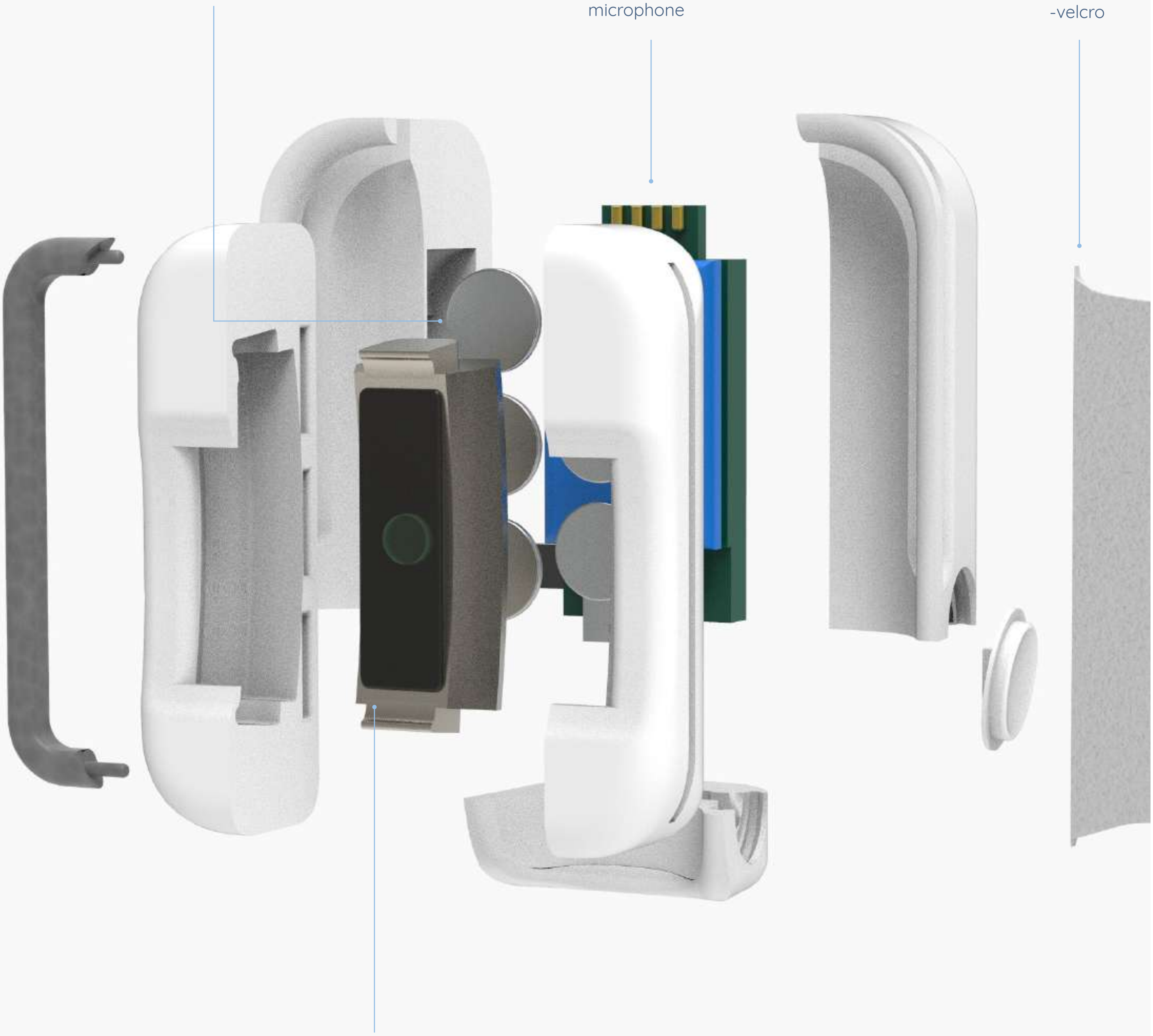


Figure 65. Stackable structure for storage

Inner components



magnets

-neodymium magnet
(10mm X 1mm * 6)

bluetooth sound sensor

-Nolan bluetooth wireless microphone

elastic armband

-neoprene
-velcro

heartrate sensor

-Fitbit Alta HR

In this project, making a more precise system in a limited time frame of 20 weeks had more priority than building up all internal electronics from scratch by myself. Thus, existing devices were implemented in my product, Fitbit Alta HR for the heartrate collection and Nolan Bluetooth microphone for the sound collection. Additionally, six magnets were embedded to click two separate parts, and neoprene elastic armband with velcro was implemented in the prototype.

Figure 66. Disassembled parts with inner components

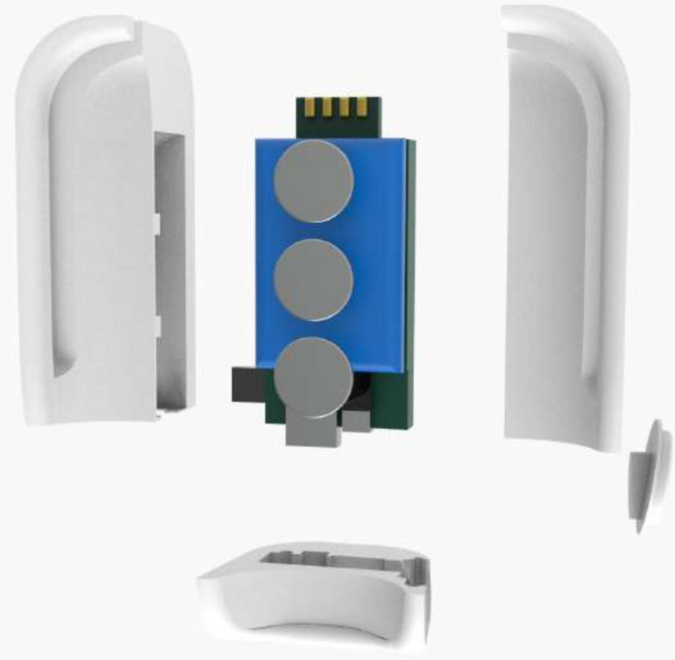


Figure 67. Disassmbled sound collection part

Sound sensor

Sound sensor casing was 3d printed in broadly four different parts including a button. As inner components, there are a PCB board with a Bluetooth, a microphone, a lithium battery, and a USB slot for charging. Additionally, there are three coin magnets implemented inside which has 10mm as its diameter and 1mm at its height.

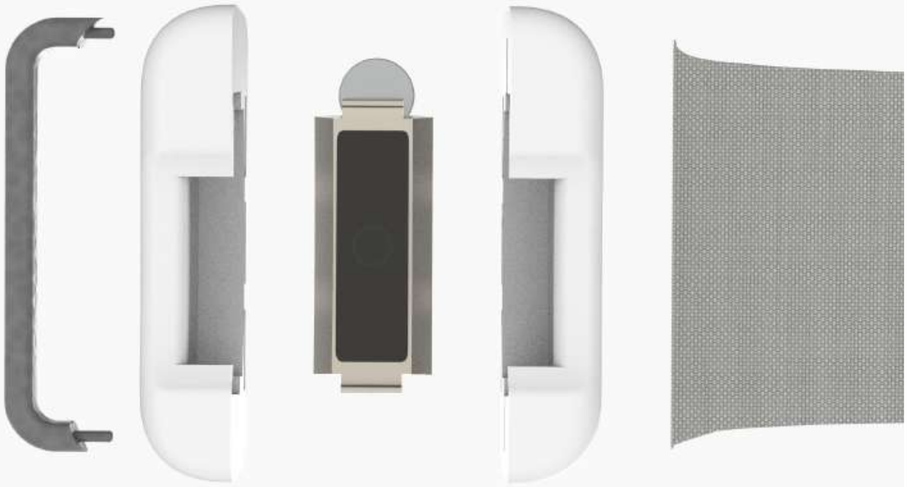


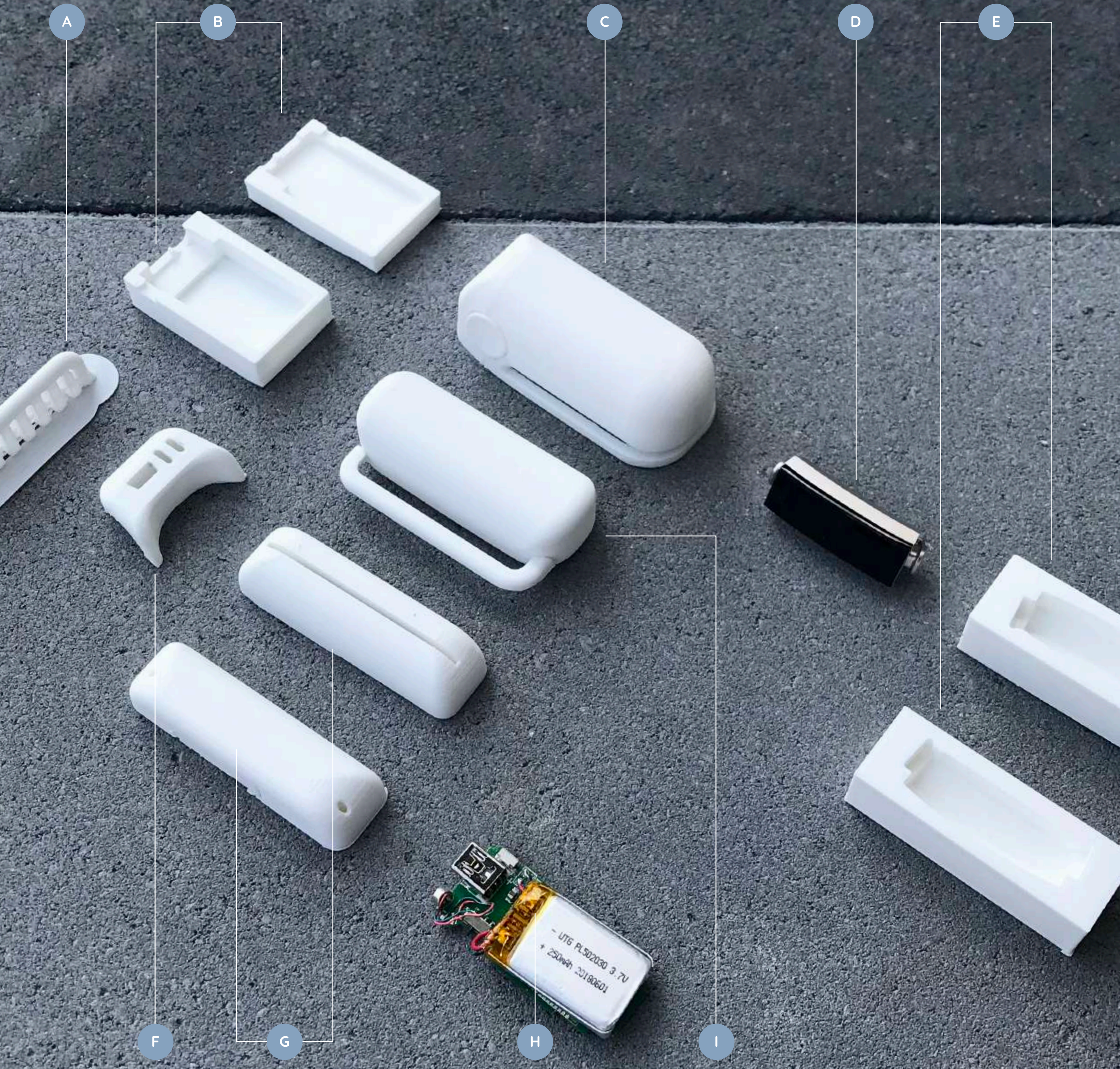
Figure 68. Disassmbled heart rate detection part

Heart rate sensor

Heart rate sensor casing was printed into three pieces, two for the main body and one for belting component. Fitbit Alta HR with PPG sensor was implemented as a central component. Also, neoprene material armband was introduced for fastening heartbeat to users' upper arm. Three coin magnets were embedded, and a armband hook was bolted to the main body later.

Figure 69. 3D-printed parts and inner components

A armband hook
B sound sensor inner holding part
C form mock-up: sound sensor
D Fitbit Alta HR (Heart rate sensor)
E Heart rate sensor inner holding part
F sound sensor lid part
G heart rate sensor outer shell
H microphone
I form mock-up: heart rate sensor



Final Design

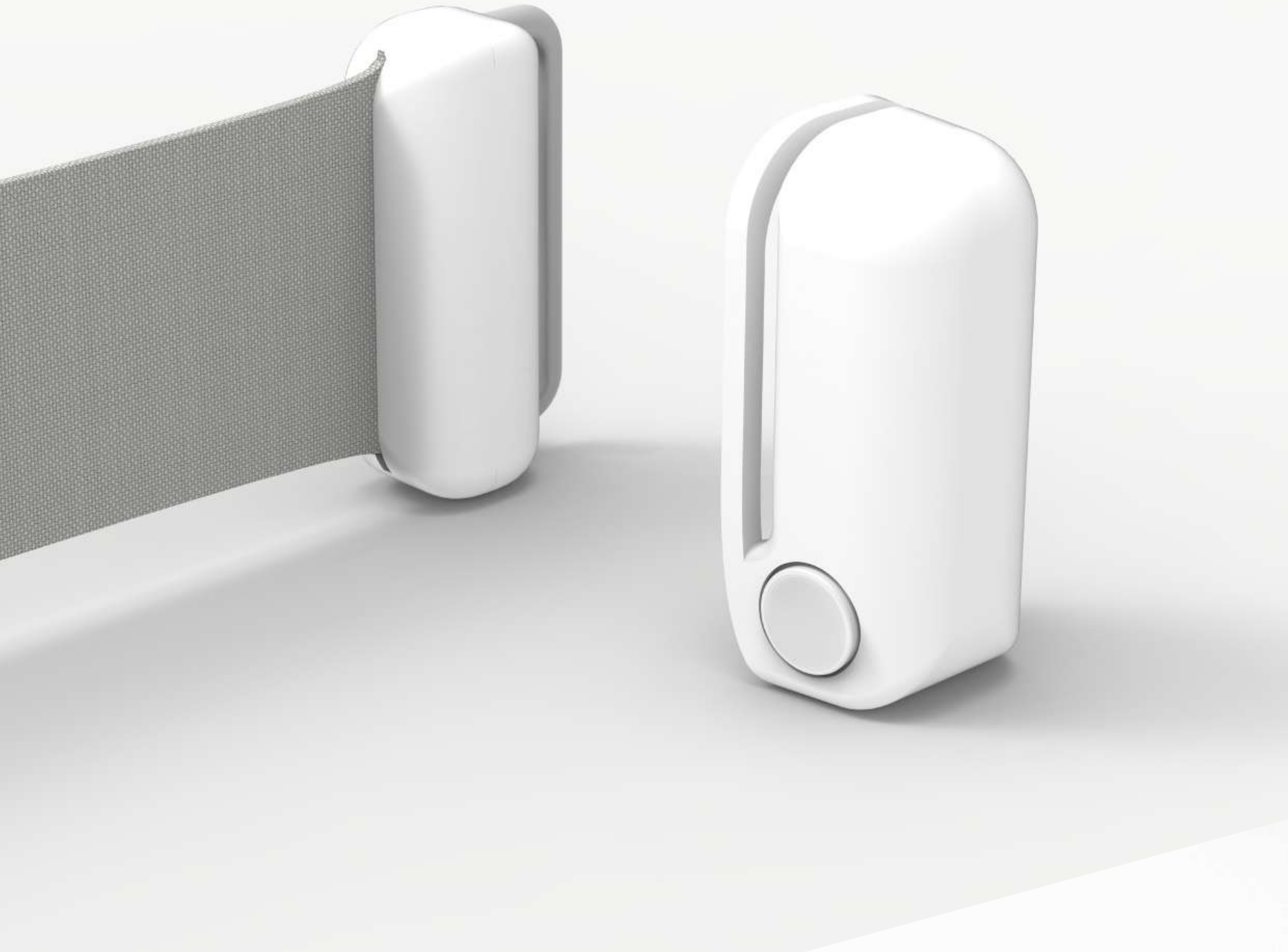
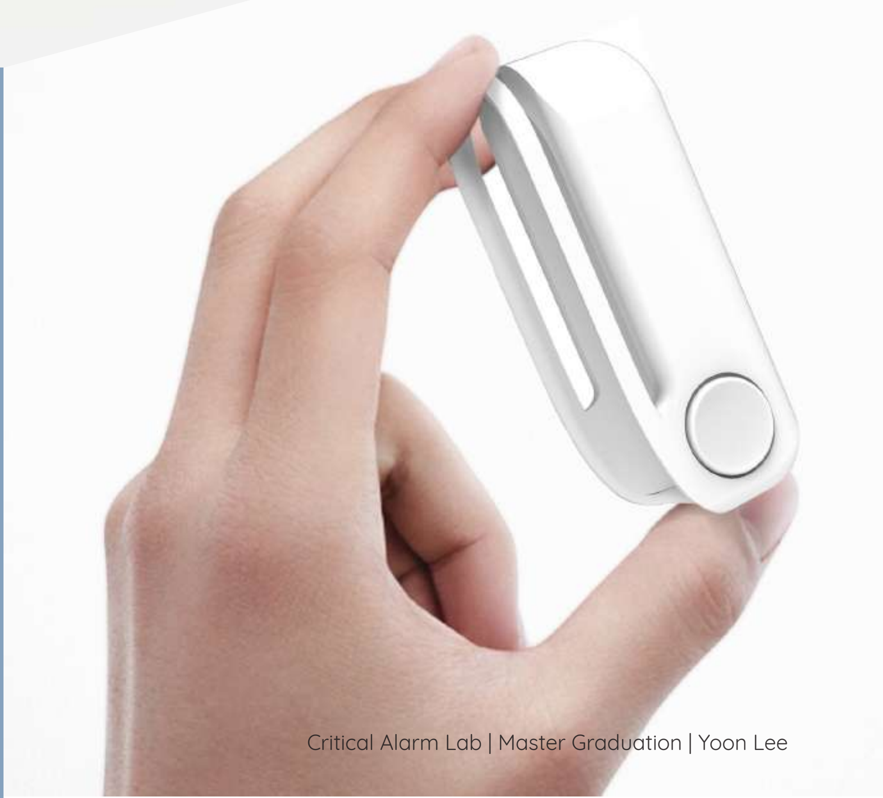
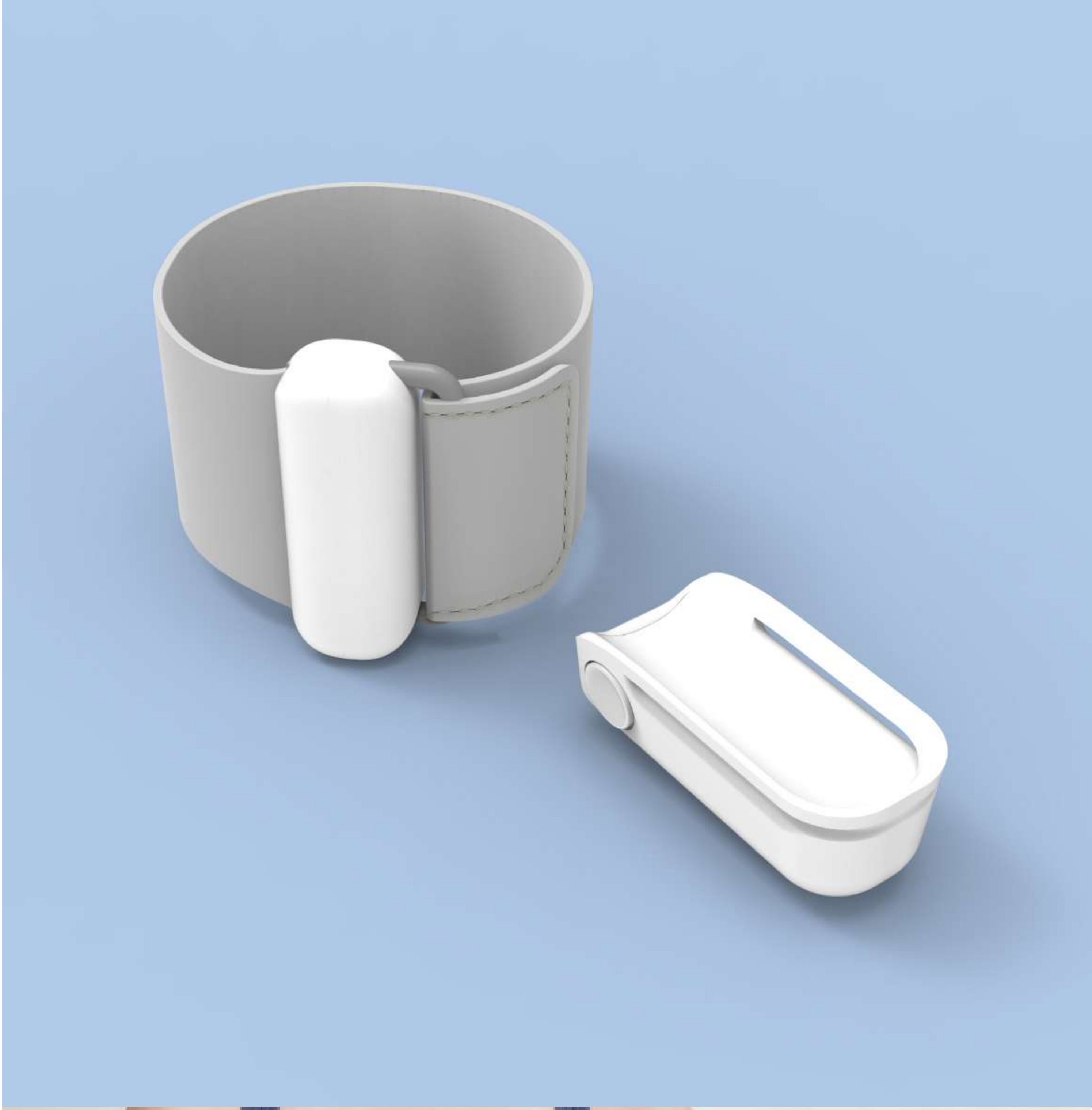


Figure 70. Cacophony mapper final design

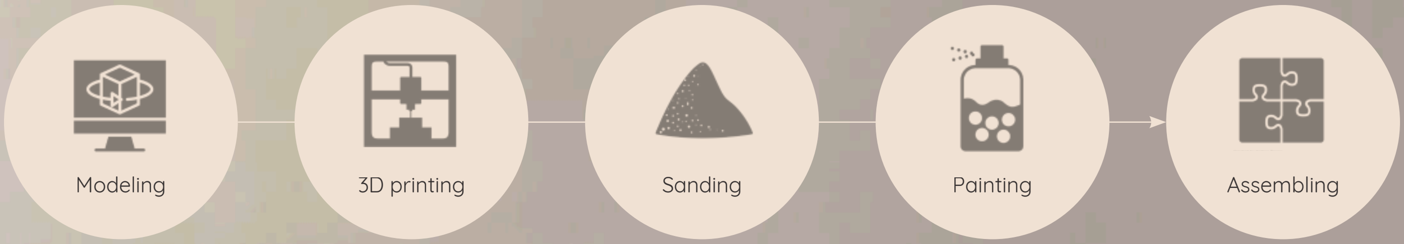
Cacophony Mapper

Taking care of
the sound level of
intensive care





Prototyping process



Modeling → 3D printing → Sanding → Painting → Assembling

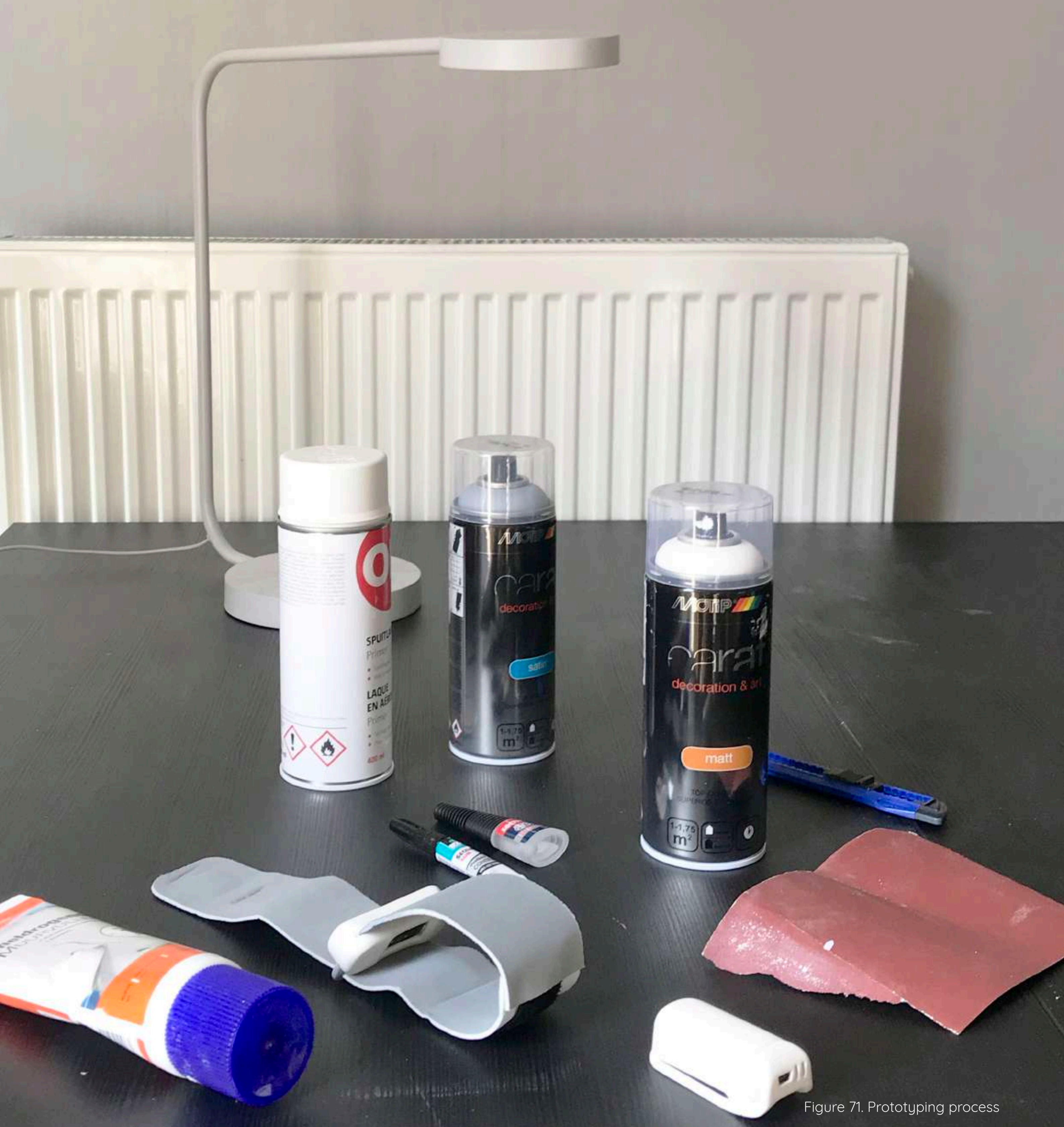


Figure 71. Prototyping process

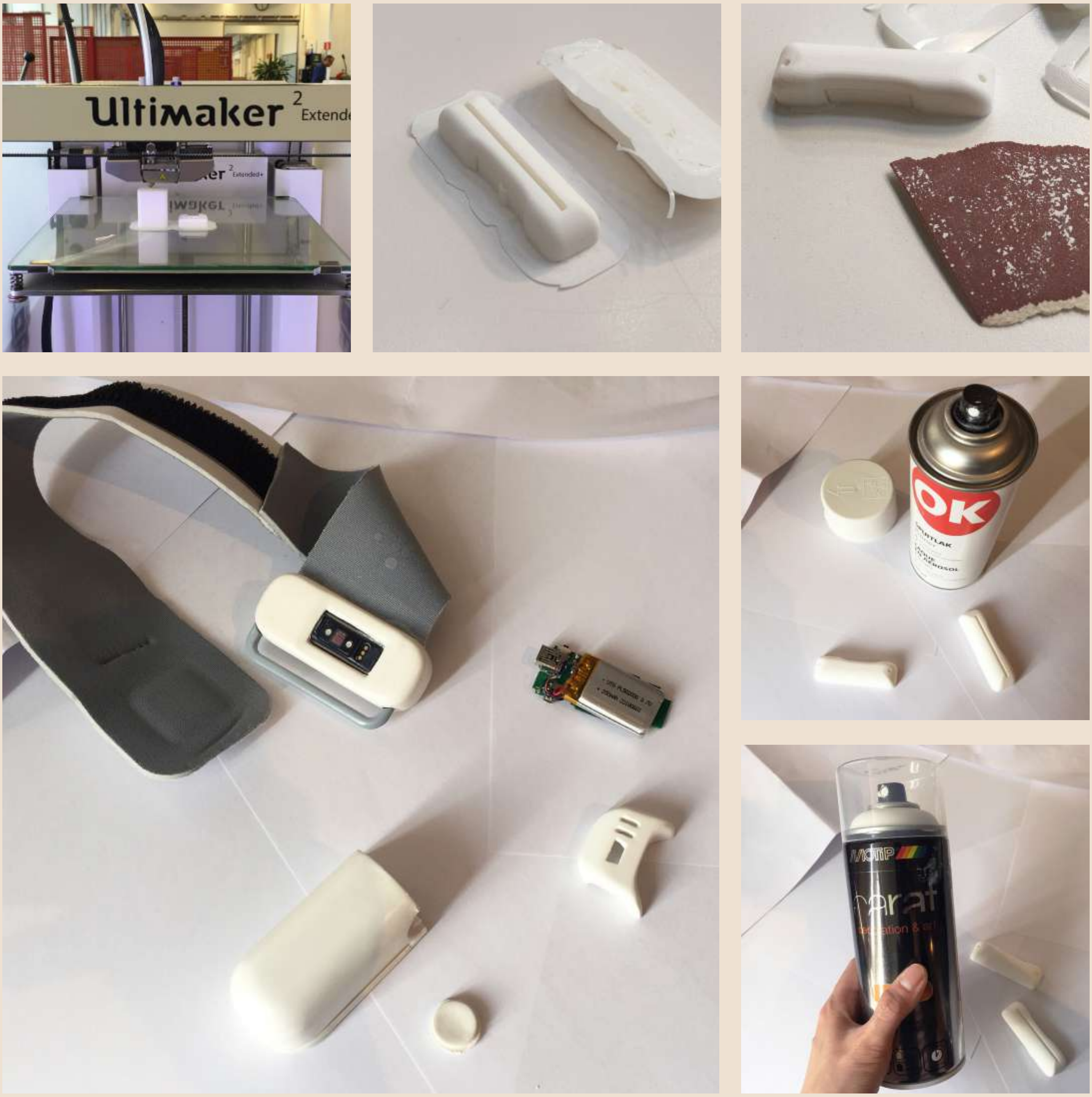


Figure 72. (clockwise from top leftt)3D printing process, 3D printed parts, sanding process, sparying primer, spraying paint, printed parts

There are broadly five steps in the prototyping process. First, 3D modeling was finished using Rhino Ceros program for the sound sensor part and the microphone part separately, having inner spaces for components and gaps for the button assembly. After the modeling, the rendered file was 3D printed within PLA material with Cura Ultramaker 2 and various revisions were made until I got the final model which suits all dimensions of the internal parts as well as a functioning button. After 3D printing, sanding was done with sandpaper with #180 to get rid of bumps on the surface and #320 for the final touch. Primer was applied to the prototype, and it was re-sanded to get a smooth surface and finally, white spray paint was applied as a final coat. After then, all the inner components were filled into the casing, and a final assembly attachment was made within bolting parts and glue.

Aesthetical prototype

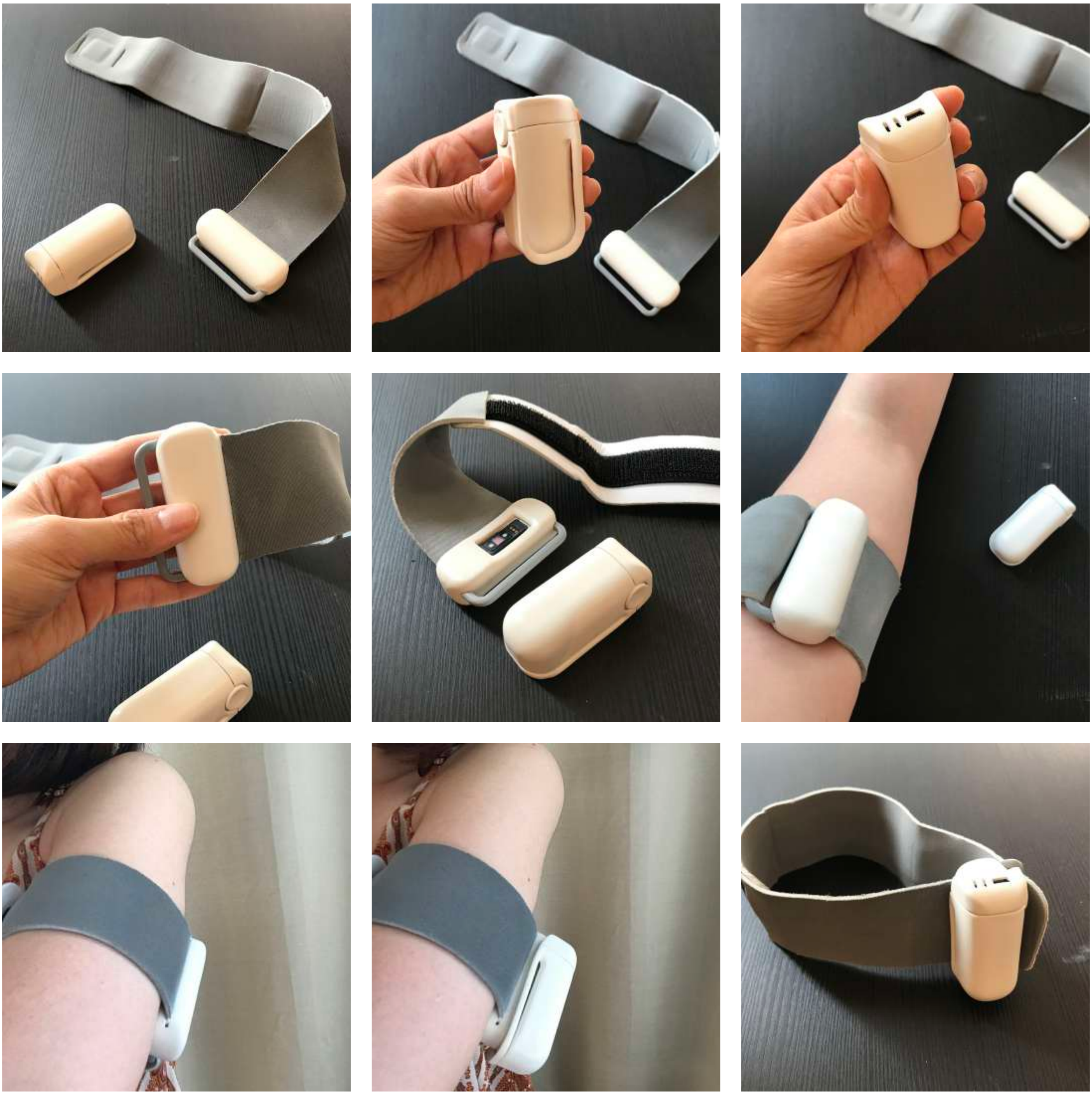


Figure 73. Aesthetic prototype

The aesthetical prototype was created with PLA material. Because of the fragile nature of the PLA material and the layered structure from the 3D printer, there was a limitation, especially in terms of the durability of the model. When it comes to clipping part, and the belting hook for an elastic band, layered structure with thin shape showed its limitation in strength. Therefore, I had to work on the second prototyping, mainly focusing on more firm structure in the model for the user test.

Rapid-prototyping for testing

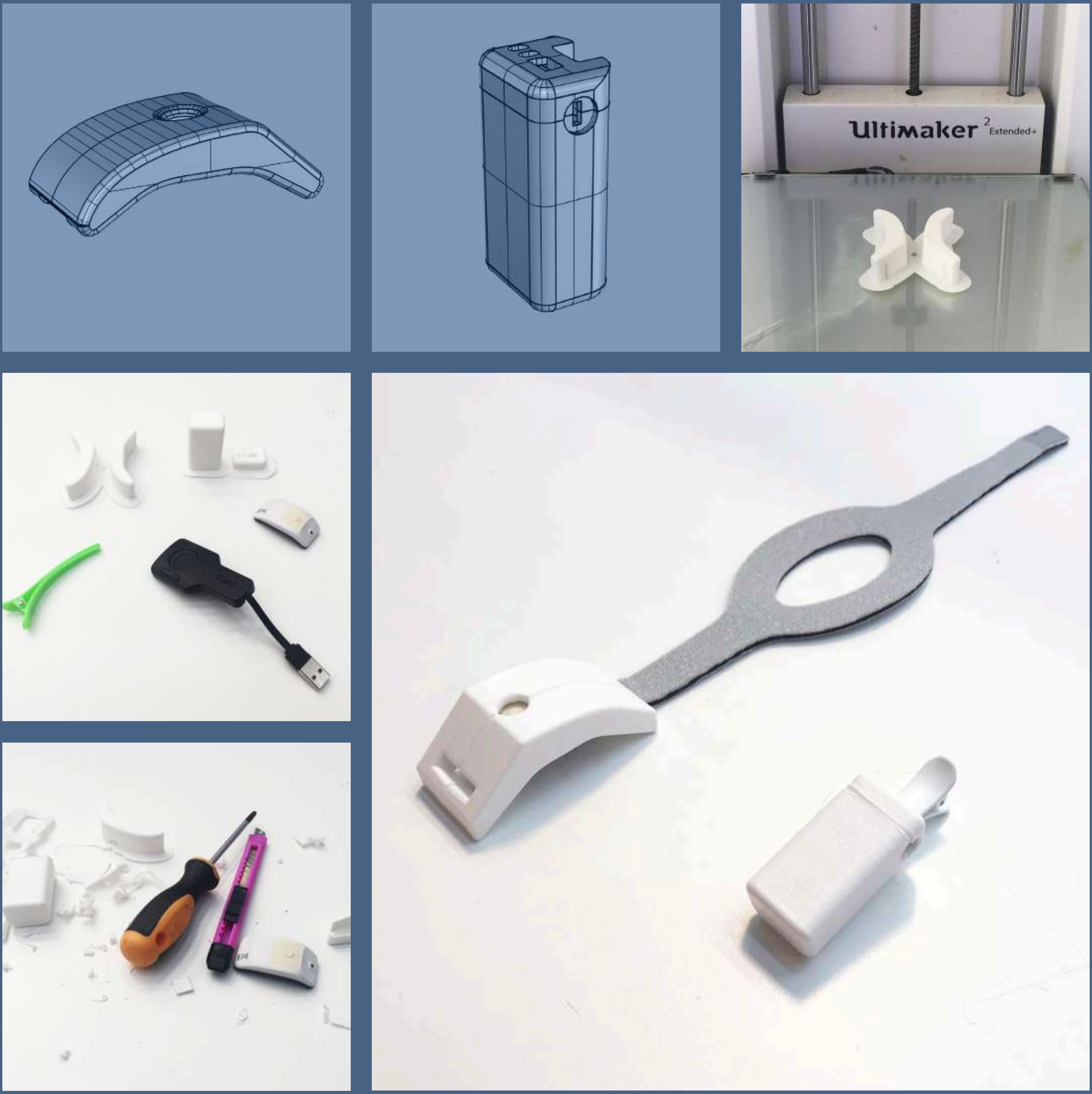


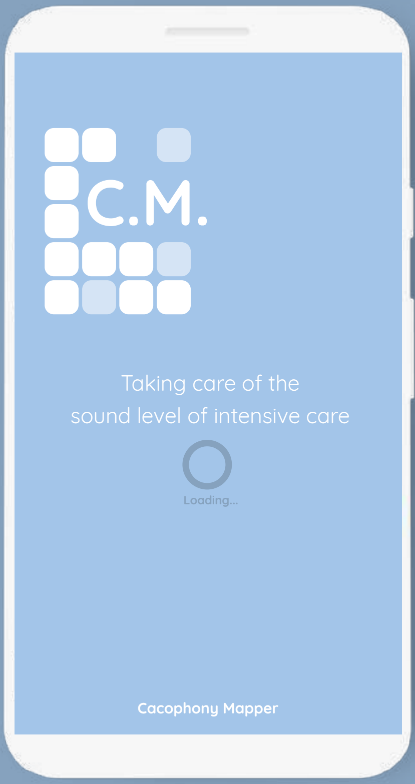
Figure 74. Rapid-prototyping process

To complement the durability issue of the clipping part and the hook part shown in the aesthetic prototype, another model was 3D-printed with stronger structure and minimal shape. An existing clip was implemented with a rapid-prototyping method to get rid of an extra hassle of creating a 3D model for a functioning clipping part from scratch. Also, the hook part was designed as a part of the body structure, so it does not need to be an added structure, improving the fragility shown in the previous aesthetic prototype.

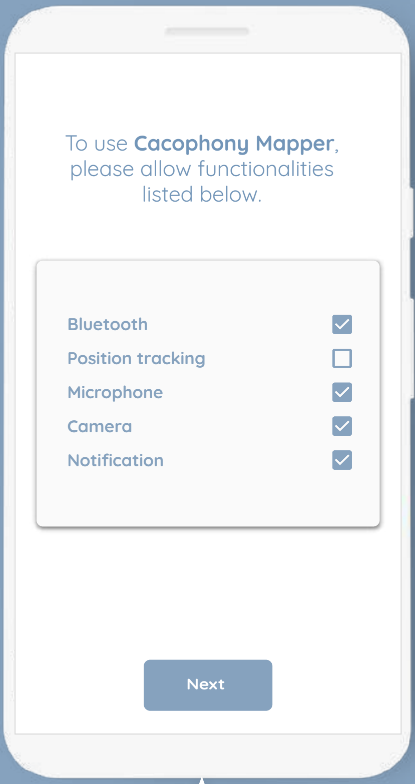
Nurse mode

Authorities

On the first page, users are asked to allow authorities needed for the application use. Bluetooth, position tracking, microphone, and notification will be enabled by ticking boxes.

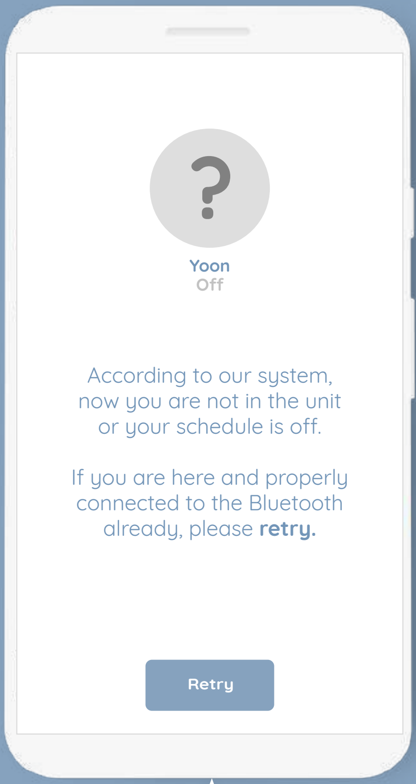


Taking care of the sound level of intensive care



To use **Cacophony Mapper**, please allow functionalities listed below.

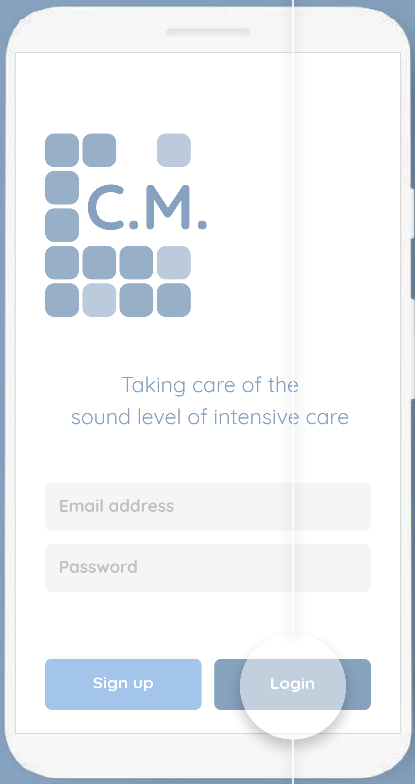
- Bluetooth ☑
- Position tracking ☐
- Microphone ☑
- Camera ☑
- Notification ☑



Yoon
Off

According to our system, now you are not in the unit or your schedule is off.

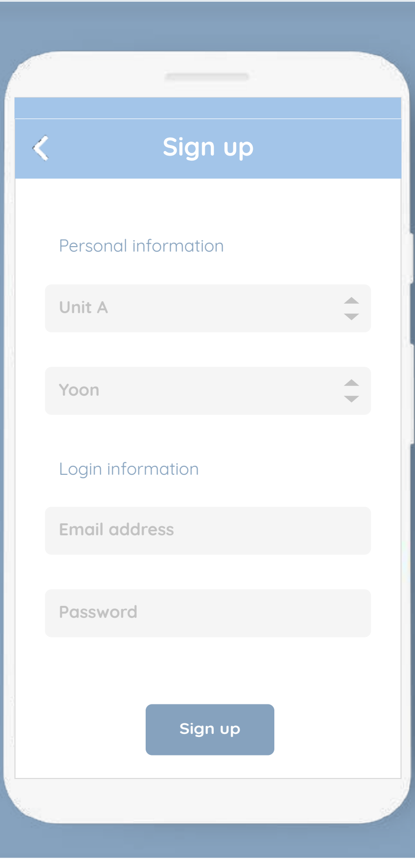
If you are here and properly connected to the Bluetooth already, please **retry.**



Taking care of the sound level of intensive care

Email address

Password



Sign up

Personal information

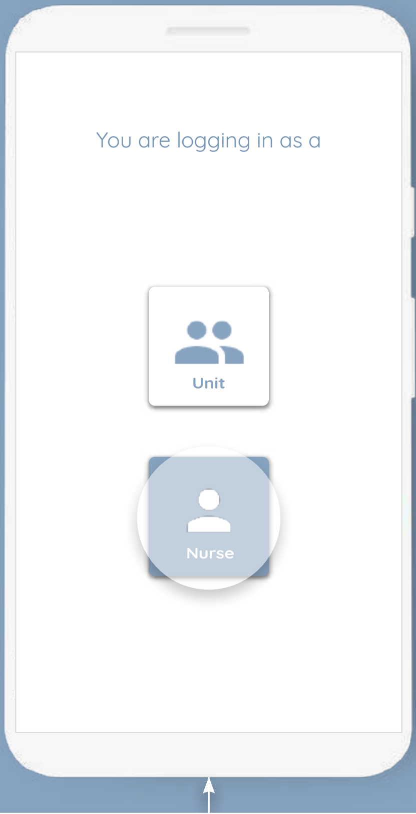
Unit A

Yoon

Login information

Email address

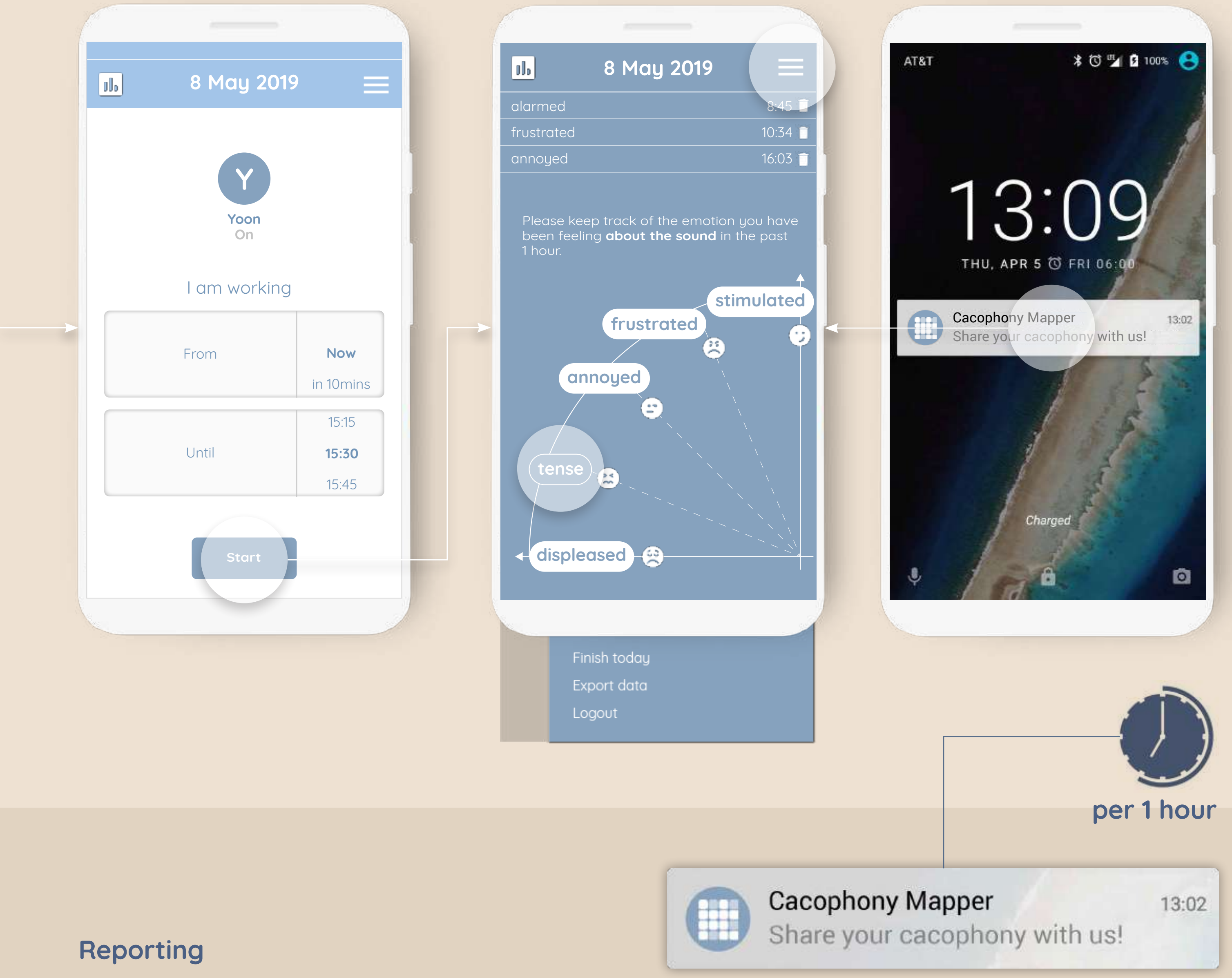
Password



You are logging in as a

Login

For the first time users, the app will ask users to sign up. In the sign up page, users need to select their hospital unit and name which is already registered on the system. Then, users can complete their sign up process by filling in their email address and password. There are two different modes for the app, one for nurses with reporting function and the other for sound researchers mainly with analysis function. In this project, only functions for nurses were embodied for the testing and analysis functions were covered as envisioned design.



Reporting

If users start Cacophony Mapper app with a nurse mode, nurses are asked to select the time of their work slot, so the system can recognize when to start and stop the recording and sending out notifications every hour. The time setting is divided into quarters, which is the same as the way nurses divide their time table; The morning work slot starts from 7 am to 15:30 pm and the afternoon work shift starts from 14:45 to 23:15, while night work shift starts at 22:45 pm to 7:15 am. Once users finish setting up their work slot and click the start button below, then they will go immediately to the emotion reporting page.

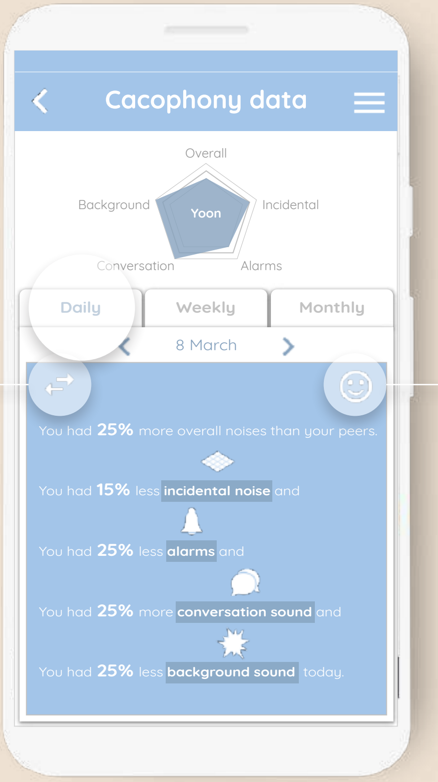
In emotion report page, users can select which negative emotions were triggered by sound events in the past 1 hour. This immediate report page designed was developed based on a paper about momentary reporting tool guidance, and it emphasizes that immediate reporting is effective than retrospective assessment because responses can reflect respondents' less-biased experiences in natural surroundings (Stone & Shiffman, 2002). Therefore, the reporting page was designed in one single layer with five emotion buttons, so nurses can tell immediately how they felt about their sound experience. If users tab into a specific emotion button, a report will be stacked in the top column, and users can easily delete their accidental report by clicking the trash bin button on the right-hand side.

In the menu button on the top right, users can find finish today button to quit their reporting in the middle. Also, users can export the data, or they can log out from their account.

Notification

For quantified research, the way of increasing the number of emotion reports from users needed to be contemplated. Sending out notifications and deciding its interval was the main issue. Thus, whether if sending out alerts on time base, or event base makes better reporting rates was contemplated. As a result, time-based notification was chosen because sound-based alerts cannot collect users' immediate responses based on sound events. Even though users try to tab the button right away, there is a time gap between the sound event and the time of the report. Instead, the system sends a notification every hour saying, "Share your cacophony with us.". If users tab into it, then users can get into an emotion report page without the app extra hassle of running the app.

Daily / Weekly / Monthly analysis



Cacophony data

Daily | Weekly | Monthly

8 March

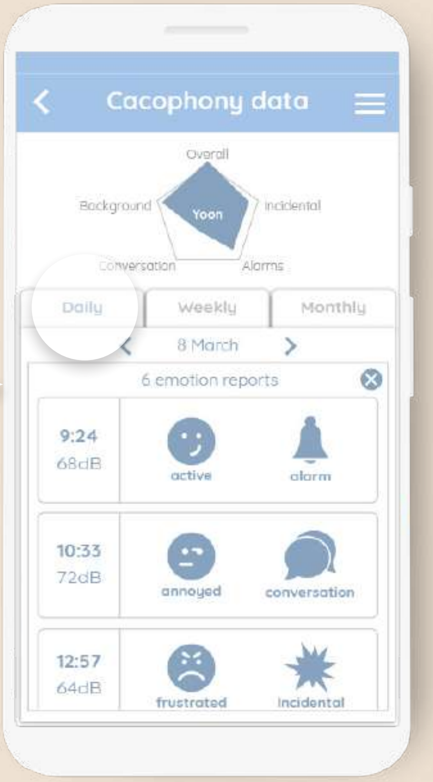
You had **25%** more overall noises than your peers.

You had **15%** less **incidental noise** and

You had **25%** less **alarms** and

You had **25%** more **conversation sound** and

You had **25%** less **background sound** today.



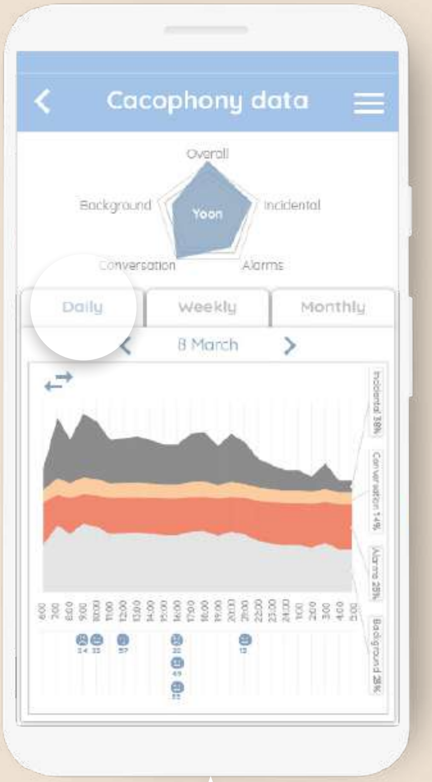
Cacophony data

Daily | Weekly | Monthly

8 March

6 emotion reports

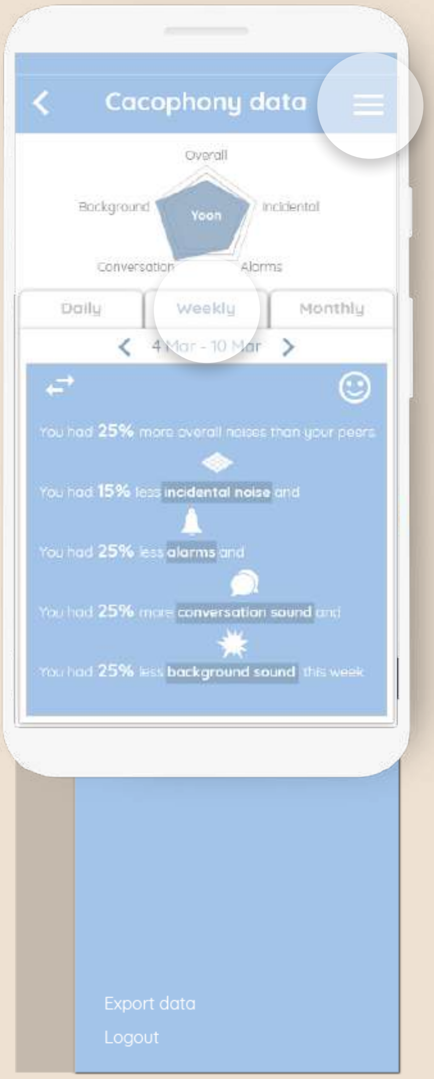
9:24 68dB	active	alarm
10:33 72dB	annoyed	conversation
12:57 64dB	frustrated	incidental



Cacophony data

Daily | Weekly | Monthly

8 March



Cacophony data

Daily | Weekly | Monthly

4 Mar - 10 Mar

You had **25%** more overall noises than your peers.

You had **15%** less **incidental noise** and

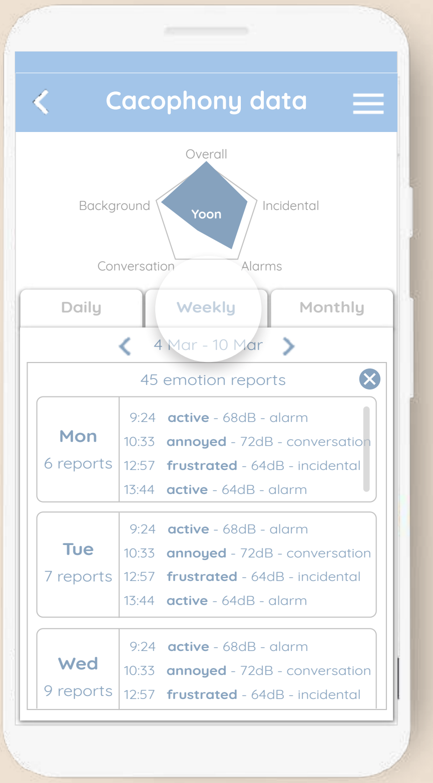
You had **25%** less **alarms** and

You had **25%** more **conversation sound** and

You had **25%** less **background sound** this week

Export data

Logout



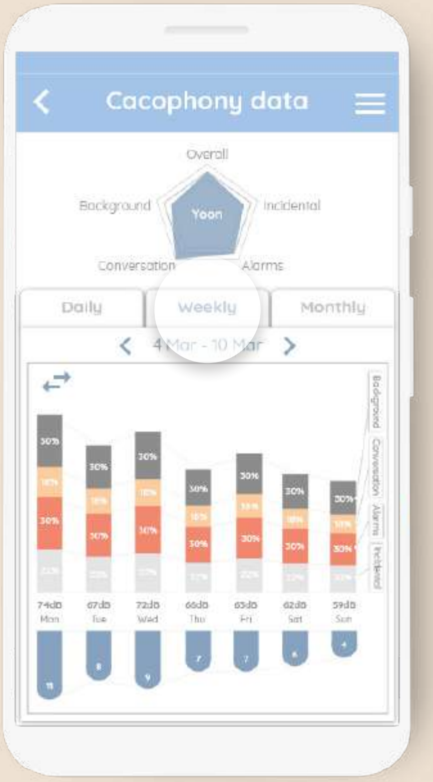
Cacophony data

Daily | Weekly | Monthly

4 Mar - 10 Mar

45 emotion reports

Mon 6 reports	9:24 **active** - 68dB - alarm 10:33 **annoyed** - 72dB - conversation 12:57 **frustrated** - 64dB - incidental 13:44 **active** - 64dB - alarm
Tue 7 reports	9:24 **active** - 68dB - alarm 10:33 **annoyed** - 72dB - conversation 12:57 **frustrated** - 64dB - incidental 13:44 **active** - 64dB - alarm
Wed 9 reports	9:24 **active** - 68dB - alarm 10:33 **annoyed** - 72dB - conversation 12:57 **frustrated** - 64dB - incidental



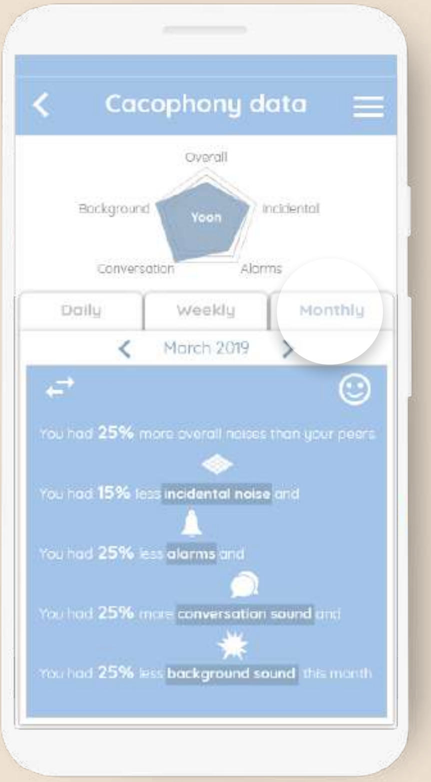
Cacophony data

Daily | Weekly | Monthly

4 Mar - 10 Mar

The analysis which will be given to nurses and sound researchers are different. For nurses, I found that giving an overview of their own sound experience is vital than giving statistical facts in terms of motivation of using Cacophony Mapper application. Self-preoccupation is an essential factor for self-reporting since reports about their personality or reflection about themselves lead to a quality response with more frequent and diligent answers (Handbook of research methods in personality psychology, 2007). To keep users' motivation for regular reporting, I found that having personal feedbacks about their own experiences is essential for users. Therefore, on this page, written comments about the sound experience were given instead of statistical analysis. Also, a comparison among users' data and others' data in terms of the sound exposure was made with percentile information, saying, "You were 25% more exposed to an incidental sound than your peers today.".

Also, data is available in a daily, weekly, and monthly basis and the total number of reports are collected as well as sound category, decibel, and heart rate. In this way, the correlation among sound, subjective, and objective assessment of stress level can be compared. It will let nurses know how excessive sound stimuli is affecting their stress level in the work environment, which can lead to better awareness of the sound in the ICU environment.



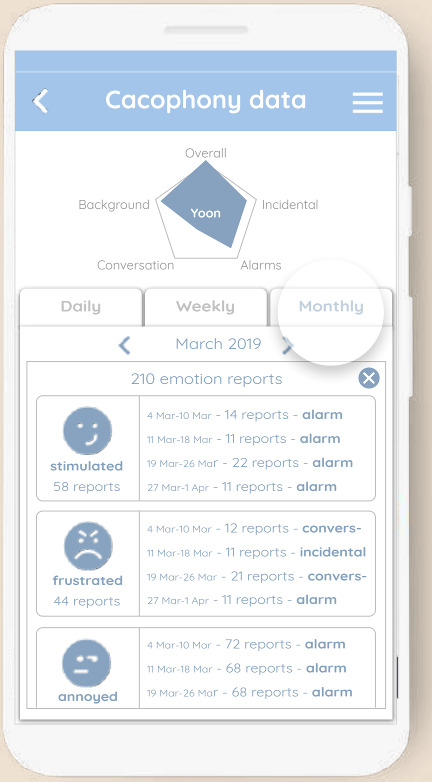
You had **25%** more overall noises than your peers

You had **15%** less **incidental noise** and

You had **25%** less **alarms** and

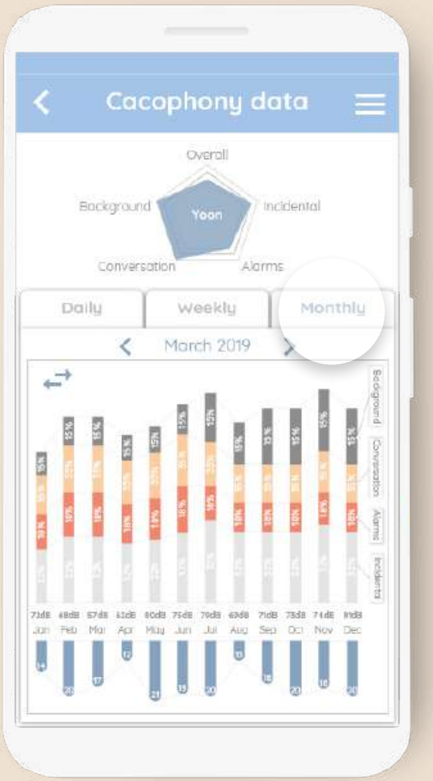
You had **25%** more **conversation sound** and

You had **25%** less **background sound** this month



210 emotion reports

stimulated 58 reports	4 Mar-10 Mar - 14 reports - **alarm** 11 Mar-18 Mar - 11 reports - **alarm** 19 Mar-26 Mar - 22 reports - **alarm** 27 Mar-1 Apr - 11 reports - **alarm**
frustrated 44 reports	4 Mar-10 Mar - 12 reports - **convers-** 11 Mar-18 Mar - 11 reports - **incidental** 19 Mar-26 Mar - 21 reports - **convers-** 27 Mar-1 Apr - 11 reports - **alarm**
annoyed	4 Mar-10 Mar - 72 reports - **alarm** 11 Mar-18 Mar - 68 reports - **alarm** 19 Mar-26 Mar - 68 reports - **alarm**



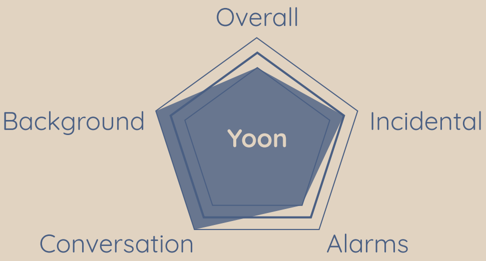


Figure 68. Pentagon of comparing sound experience with other users

Analysis for nurses

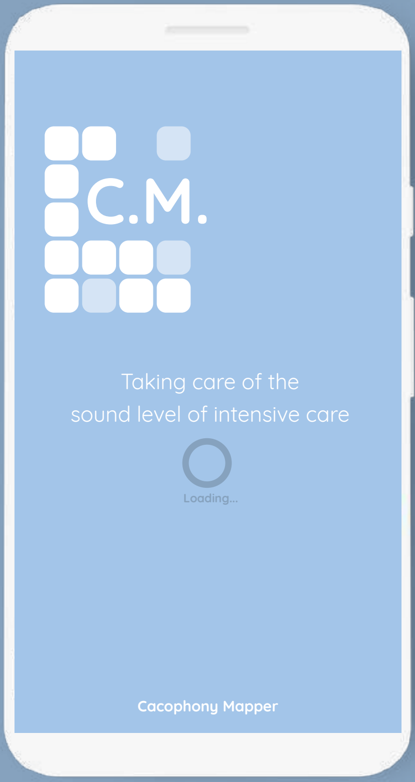
Feedback for nurses is primarily geared toward improving their motivation to report more frequently. Therefore, by looking at a pentagon visual overview of users' sound experiences, users can easily get to know how their sound experience was different from others. The three levels of indication of more/average/less sound are given through the pentagon graphic as well as the written text below, talking about the significance of your sound experience today.

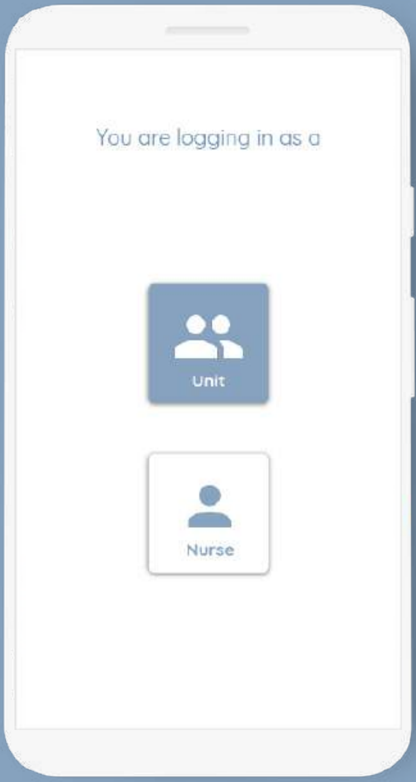
If users click on the crossing arrows button on the top left in the blue box, it shows the statistical analysis with bar charts, illustrating sound fluctuation. In each bar, four sound categories are divided into four colors with numbers written in tags on the right-hand side. Also, the blue chart on the bottom shows the number of emotion reports from nurses through the day, week, or a month. By making a comparison of two bar charts' fluctuations, nurses get to know how sound can affect overall emotional responses of themselves while working in the ICU.

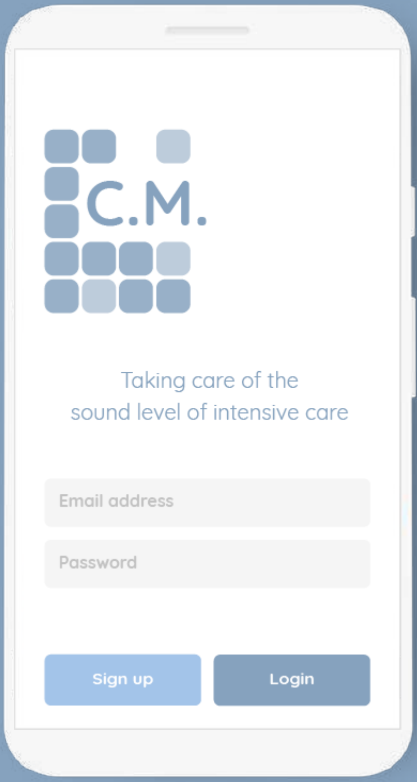
If a user clicks the smile button on the right-hand side in the blue box, the screen changes into an emotion report record page of the day, week, or a month. Listed emotions follow the order of the most reports to the least reports, and users can check which sound was dominant while they report their negative emotions. By doing so, nurses can think about the sound that they produce by themselves, that it can eventually lead to a better sound awareness and more pleasant sound environment in the ICU.

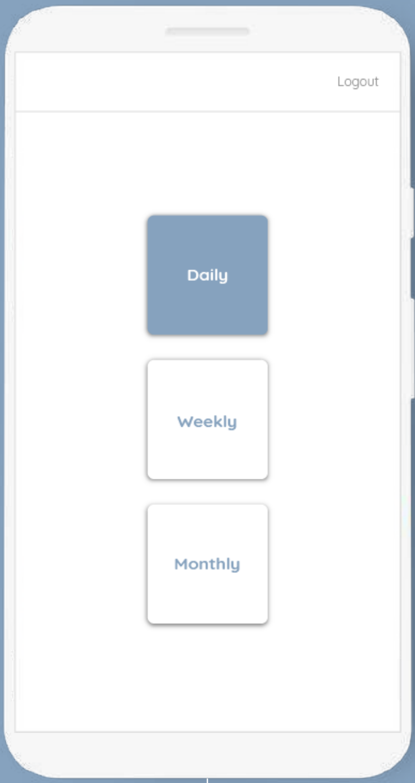
Researcher mode: envisioned concept

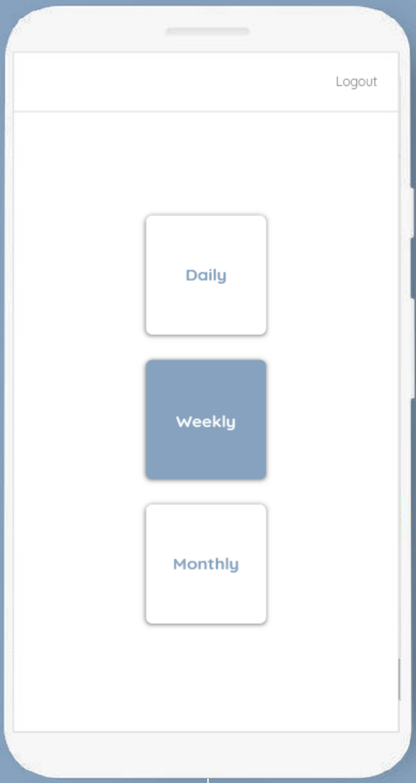
The data sets given to researchers are different from the data which is provided to nurses'. For researchers, making a good comparison of sound environment depending on dates, days, units, and individuals is essential. By investigating the organized dataset, finding general rules about the sound events in the ICU can be one of their priorities. Also, assumingly, sound researchers check those data on a daily or regular basis; thus, I made daily, weekly, and monthly buttons as the application's upper layer and made tabs for overall, by unit, and by a nurse as its sub-category so that researchers can compare the data more efficiently.

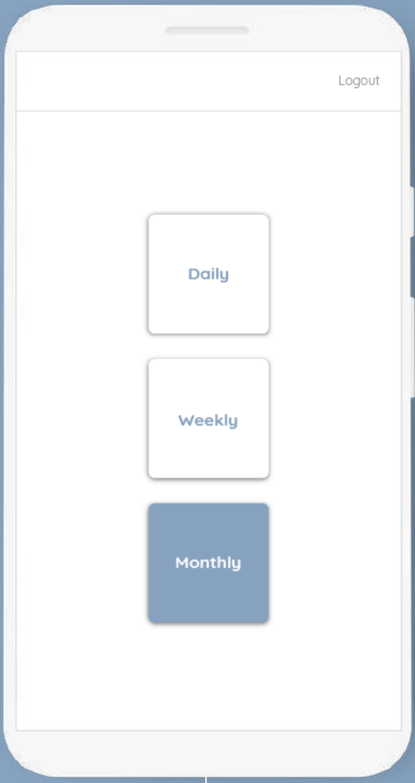






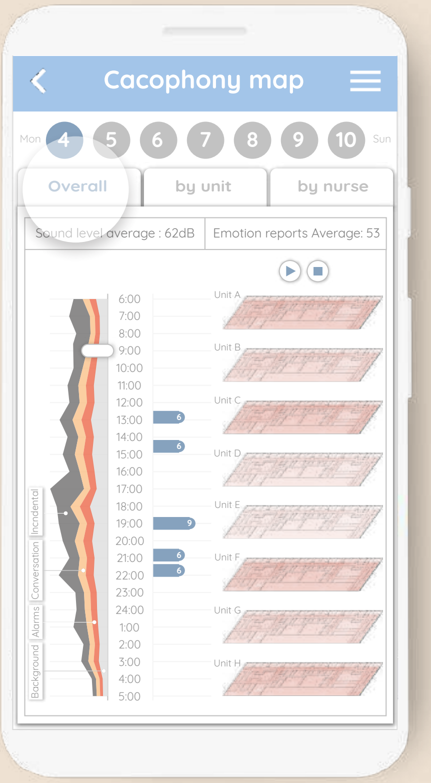




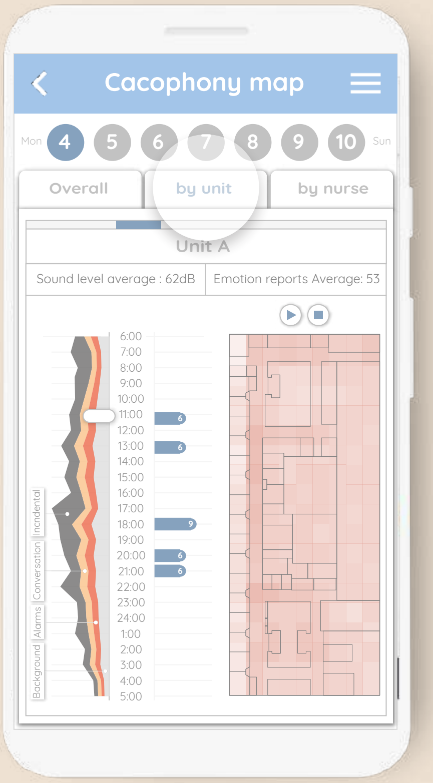


Daily

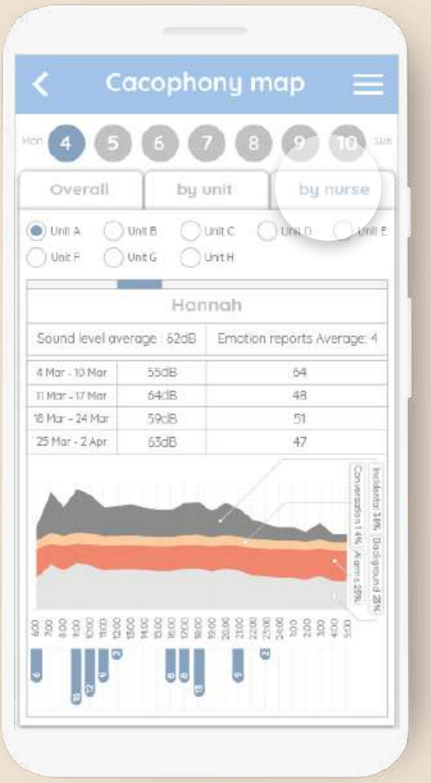
Overall



by unit



by nurse



Sound level average : 62dB		Emotion reports Average: 4
4 Mar - 10 Mar	55dB	64
11 Mar - 17 Mar	64dB	48
18 Mar – 24 Mar	59dB	51
25 Mar - 2 Apr	63dB	47

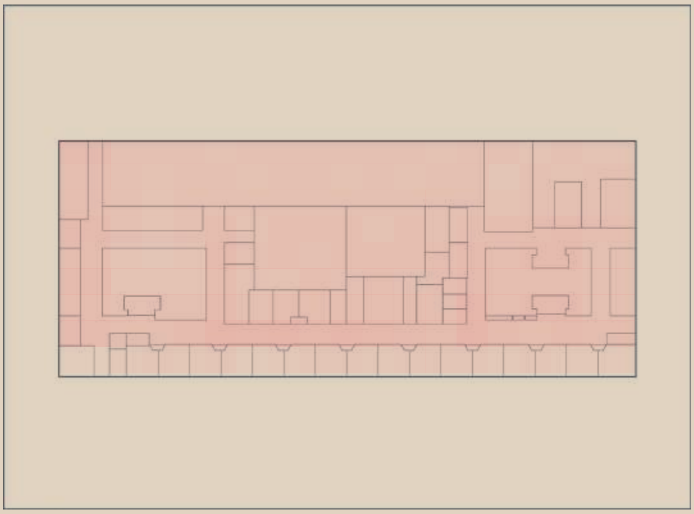


Figure 75. Heatmap concept

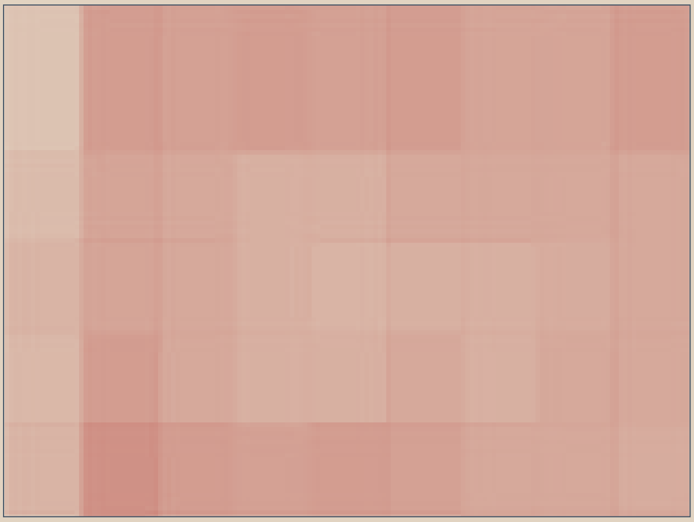


Figure 76. Heatmap scale: by transparency

Daily analysis

Under the "overall" tab, users can find the sound and the emotion report data they want to look into. Average sound level is given in decibel, and the number of the emotion report is shown right next to it, so researchers can easily find the correlation between sound level and the number of nurses' complaints toward it. The graph shown on the left-hand side shows changing sound level in a 24-hour-time-frame as well as nurses' sound reports. On the right-hand side, there are sound heat maps for various units. Therefore, depending on where users put a white toggle button on the timeline, they can easily see the sound flow of the overall unit on the heat map. Additionally, the flow by the time can be automatically shown by clicking the play button on the top of the heat map.

Under "by unit" tab, a sound heat map of a specific unit is available, and researchers can swipe the screen to see the next unit's data set.

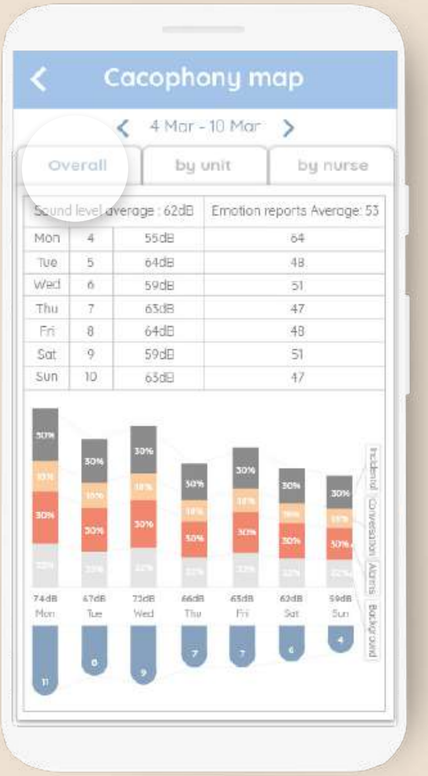
Under "by nurse" tab, researchers can choose the unit that they would like to investigate, and they can search for a dataset of individuals by swiping the screen.

Sound heat map

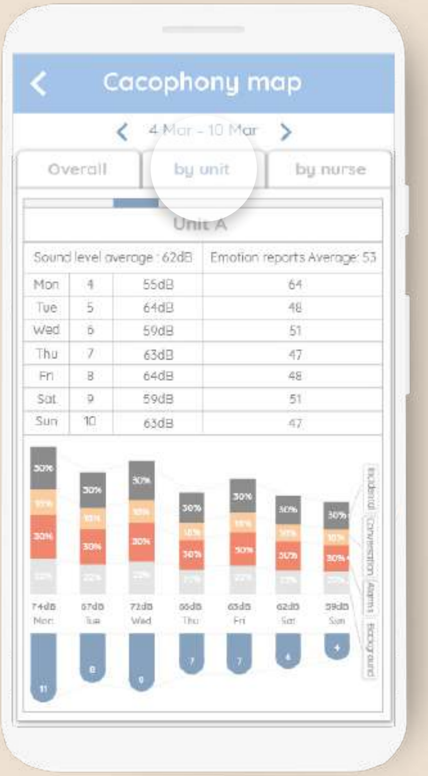
The sound heat map will be created by combining indoor position information and collected sound data. If users change the position of the toggle on the graph on the left-hand side, it will show the visualized sound analysis in different hours. Based on which data set researchers are looking into, the interface will show a group of the heat map, that shows the overall sound flow of all units, whereas clicking "by unit" tab will enable researchers to look at the sound flow of a specific unit. The graphic shows more shades of red color when there are more noises in the unit while a calm sound environment will show the more transparent color effect on the heat map.

Overall | by unit | by nurse

Weekly

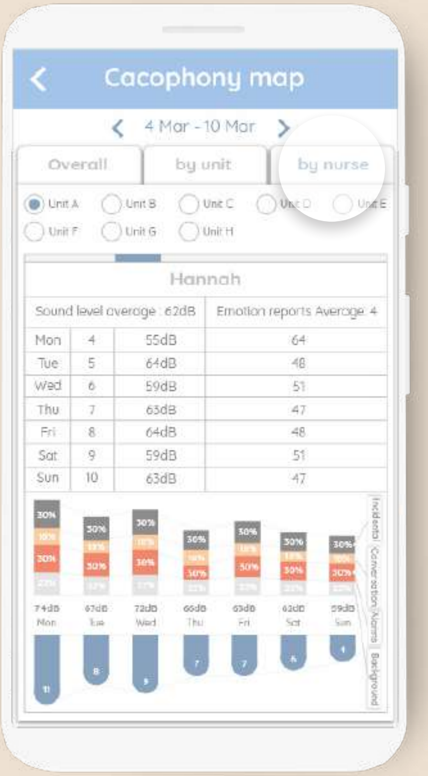


Sound level average : 62dB			Emotion reports Average: 53
Mon	4	55dB	64
Tue	5	64dB	48
Wed	6	59dB	51
Thu	7	63dB	47
Fri	8	64dB	48
Sat	9	59dB	51
Sun	10	63dB	47



Unit A

Sound level average : 62dB			Emotion reports Average: 53
Mon	4	55dB	64
Tue	5	64dB	48
Wed	6	59dB	51
Thu	7	63dB	47
Fri	8	64dB	48
Sat	9	59dB	51
Sun	10	63dB	47

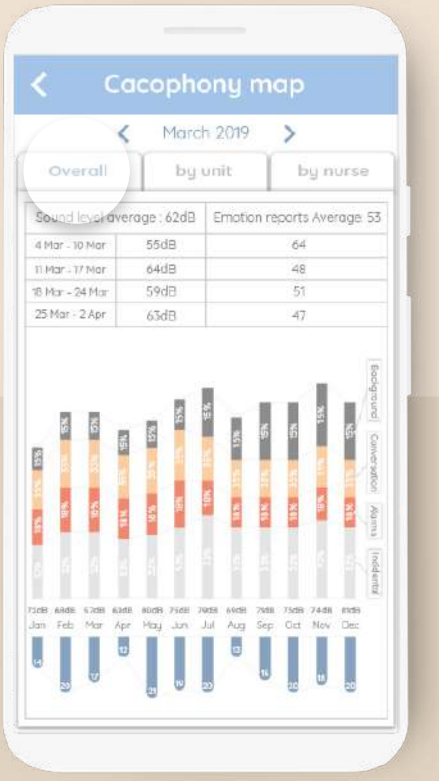


● Unit A ○ Unit B ○ Unit C ○ Unit D ○ Unit E
○ Unit F ○ Unit G ○ Unit H

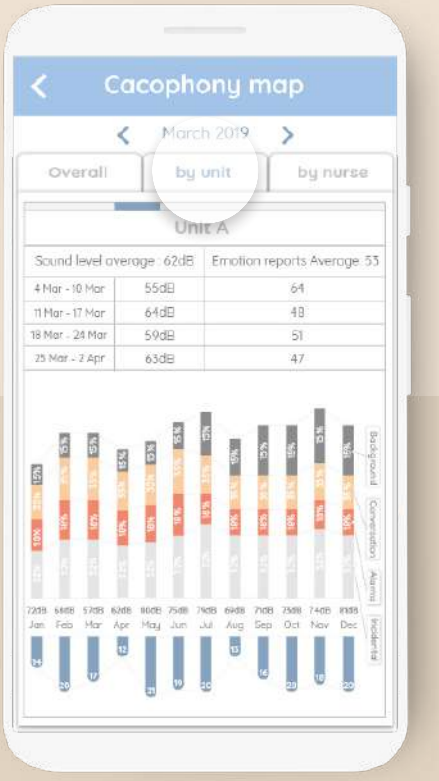
Hannah

Sound level average : 62dB			Emotion reports Average: 4
Mon	4	55dB	64
Tue	5	64dB	48
Wed	6	59dB	51
Thu	7	63dB	47
Fri	8	64dB	48
Sat	9	59dB	51
Sun	10	63dB	47

Monthly

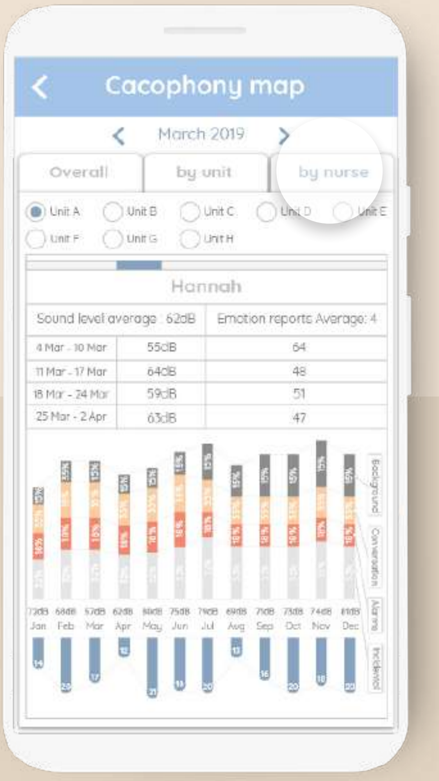


Sound level average : 62dB		Emotion reports Average: 53
4 Mar - 10 Mar	55dB	64
11 Mar - 17 Mar	64dB	48
18 Mar - 24 Mar	59dB	51
25 Mar - 2 Apr	63dB	47



Unit A

Sound level average : 62dB		Emotion reports Average: 53
4 Mar - 10 Mar	55dB	64
11 Mar - 17 Mar	64dB	48
18 Mar - 24 Mar	59dB	51
25 Mar - 2 Apr	63dB	47



● Unit A ○ Unit B ○ Unit C ○ Unit D ○ Unit E
○ Unit F ○ Unit G ○ Unit H

Hannah

Sound level average : 62dB		Emotion reports Average: 4
4 Mar - 10 Mar	55dB	64
11 Mar - 17 Mar	64dB	48
18 Mar - 24 Mar	59dB	51
25 Mar - 2 Apr	63dB	47

User scenario

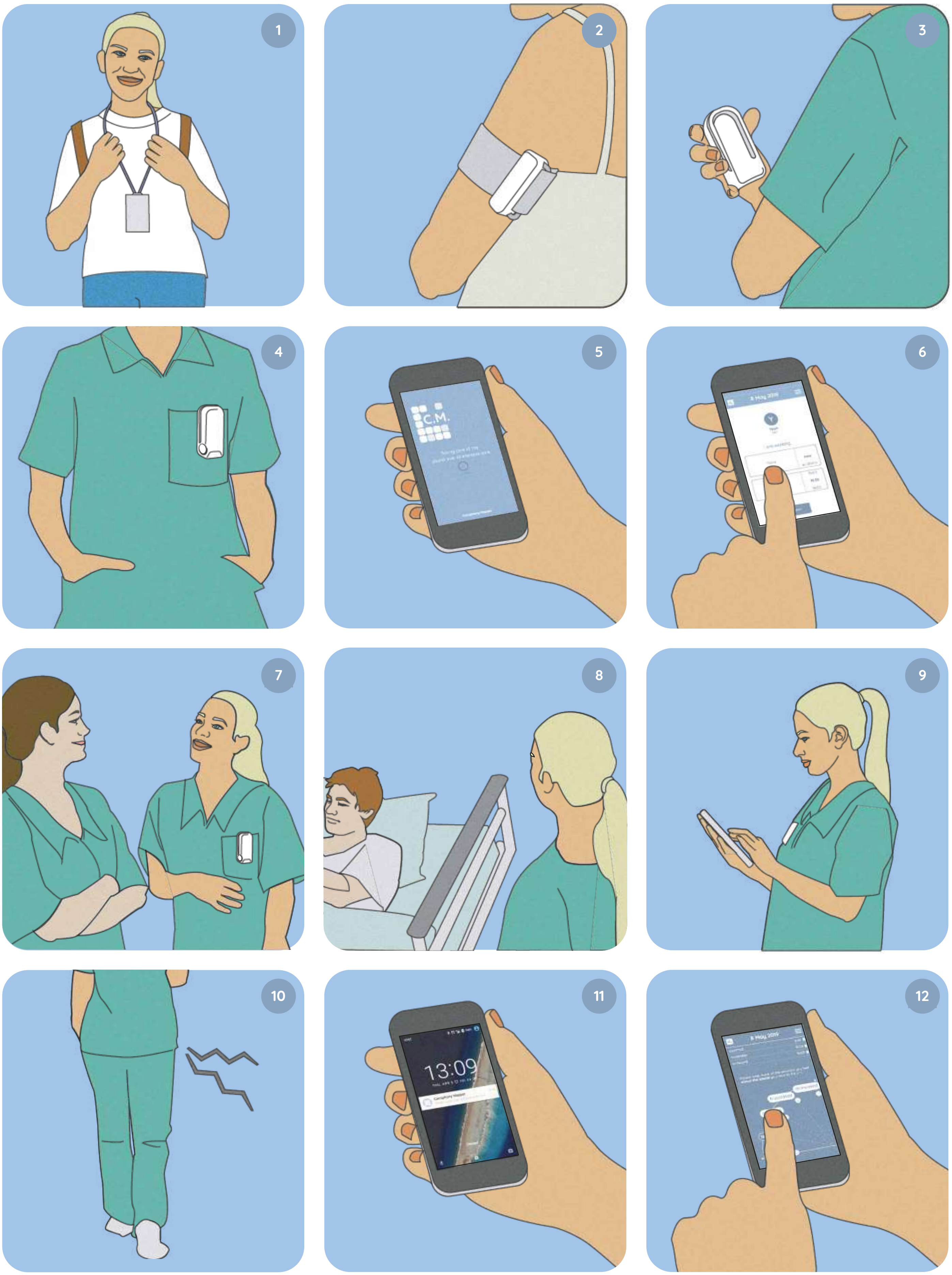


Figure 77. User scenario of using Cacophony mapper products and the application

User test: System reliability test (test 1, 2)

Introduction (test 1, 2 ,3)

The purpose of developing Cacophony Mapper system is to provide a platform which can find a correlation between sound stimuli and nurses' stress level as a response to the negative sound environment. Therefore, data collection for both sound and stress level is essential for the whole project. Thus, the data reliability is a crucial part of this project since the dataset will be a foundation which will enable validating noise fatigue as a result. Therefore, there were two aspects that I tried to test in this evaluation phase; the reliability of the sound classification system and the general usability of the system.

Since sound categorization has been mainly covered as a study subject for this project, I decided to put the sound classification system reliability as a top priority for the testing. Also, the system usability was chosen as a next test subject since Cacophony Mapper system is first introduced through this project, so how the system will be used and accepted by users is something needs to be investigated for further discussion. Thus, I divided the test plan into three parts, a system reliability test with a sound file, a system reliability test with a sound recording collected from test participants, and general usability test with test participants.

Participants (test 2, 3)

For test 1, only WAV sound recording will be tested, so the participant is not needed. For test 2 and 3, there will be 11 test participants from Industrial design department, TU Delft.

Method (test 1, 2)

As sound classification is one of the core functions of the Cacophony Mapper system, conducting a reliability test seemed to be the first necessary step to check for further development.

First of all, the system reliability will be tested by directly putting the WAV sound file to Matlab to see the accuracy level of the filter, getting rid of all impacts of external factors, such as the quality of the testing device or unexpected inputs to the system during the experiment. As Matlab classification learner showed 71.4% as its final accuracy level as a result of optimization in chapter 3.13, the accuracy of the filter will be tested by putting sound data to the sound classifier to see if the test result meet the accuracy level that system aims for. This process will be evaluated using a format shown in Figure 78.

Also, the system reliability will be tested with experiments with students at the Industrial design department, TU Delft, using the product and mobile application as its apparatus. To see the accuracy of the implemented sound classifier, sound recording will be done using the mobile application, and the classification will be conducted to see if the system can be successfully done with external factors too. Figure 79 will be filled in through this test, and the result will be compared with test result 1 to see to find on which level environmental factors, such as recording quality and noise level of the test environment, will affect the overall sound classification result (Figure80).

What will be tested?

Test 1: System reliability (using WAV)

-What is the overall sound classification success rate?
-What is the sound classification success rate of each group?

Test 2: System reliability (with students)

-What is the overall sound classification success rate?
-What is the sound classification success rate of each group?
-How the result different when using WAV file and sound recorded within the app?

Using sound file for testing (WAV file)

	Phase 1	Phase 2	Phase 3	Phas
Recording	Incidental	Conversation	Machinery	Alarms
Sensing result				
Correct				
Success rate				
Overall Success rate				

Figure 78. Test 1 format: classification success rate in WAV file test

Using a mobile app with participants

Participant X	Phase 1	Phase 2	Phase 3	Phas
Recording	Incidental	Conversation	Machinery	Alarms
Sensing result				
Correct				
Success rate				

.
.
.

Overall classification success rate	Incidental noise classification success rate	Conversation noise classification success rate	Machinery noise classification success rate	Alarm noise classification success rate

Figure 79. Test 2 format: classification success rate in user test

System Accuracy	WAV file	App recording
Incidental		
Conversation		
Machinery		
Alarms		
Overall		

Figure 80. Test result comparison: test1, test2

User test: System usability test (test 3)

Method (test 3)

First of all, the general product and system usability will be validated through this experiment before the further test is conducted in the real hospital environment with nurses. The test will be conducted in two steps, and the first part, it will be investigated to see if test participants can easily pick up what they should do to see the overall system usability. The process will be evaluated using the form in Figure 82, which is a test instruction paper with a scaling system. Also, through an after-test-survey in Figure 83, the application will be assessed whether if it was manageable to learn, clearly structured, easy to use, straightforward to use, and practical to use, which was formulated using word pairs from Attrakdiff scale shown in Figure 81.

What will be tested?

Test 3: System usability test (with students)

Through this system usability test, the following two questions will be answered:

-How users evaluate the overall usability of the Cacophony Mapper system (product & application)?

-How users evaluate the usability of Cacophony Mapper application?

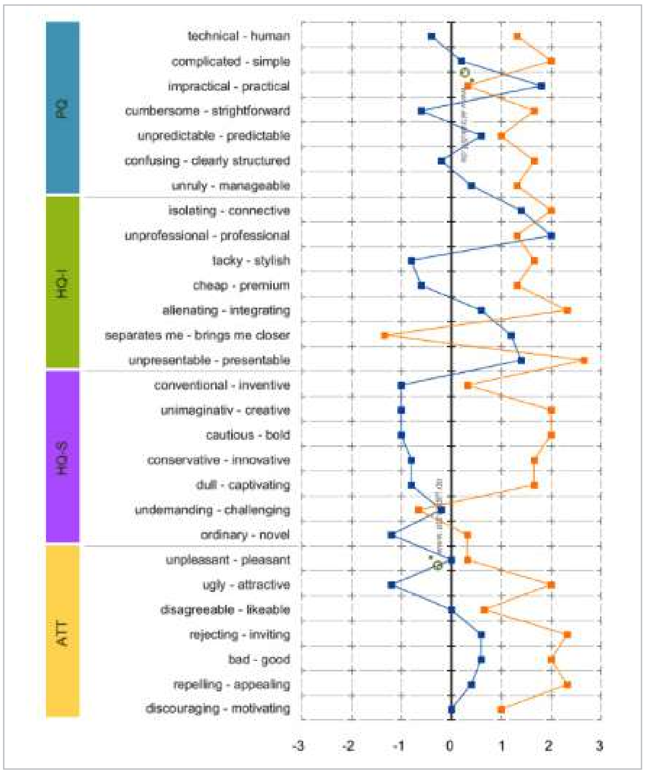


Figure 81. Attrakdiff scale

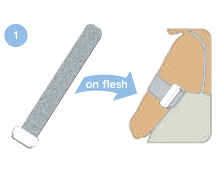
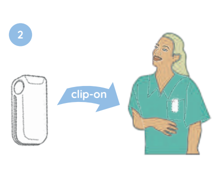
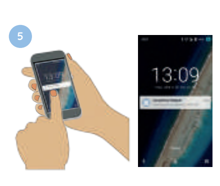
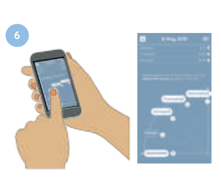
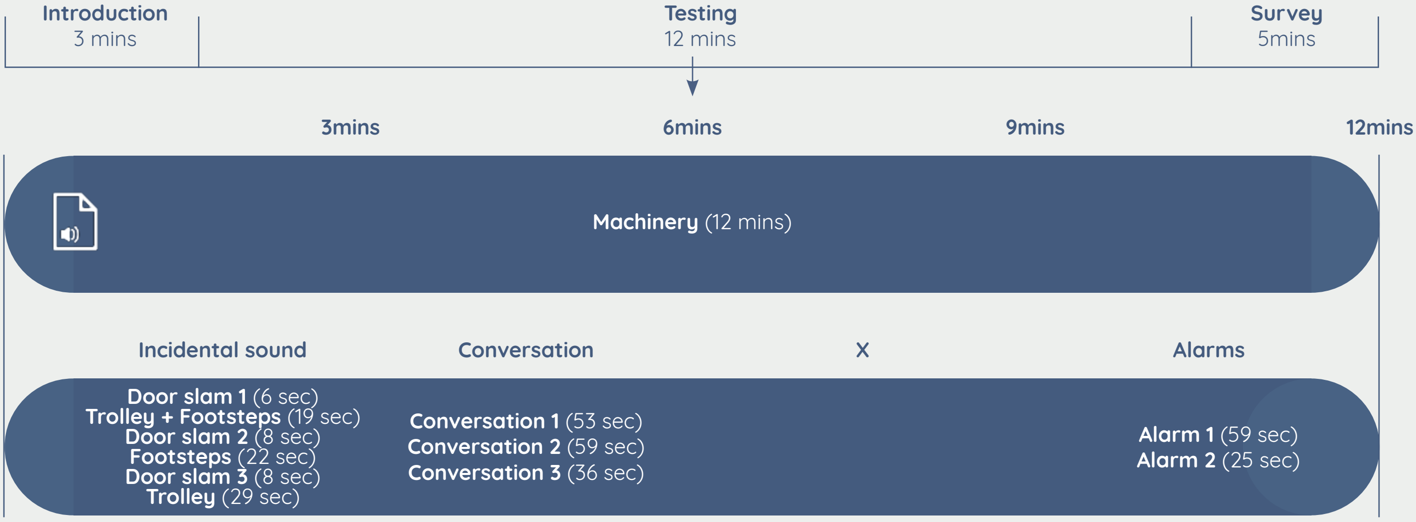
Step 1 heart rate tracker	Step 2 Sound collection part	Step 3 App: Login	Step 4 App: Time-setting	Step 5 App: Notification	Step 6 App: Emotion report
					
-/2.0	-/2.0	-/2.0	-/2.0	-/2.0	-/2.0
-/12					

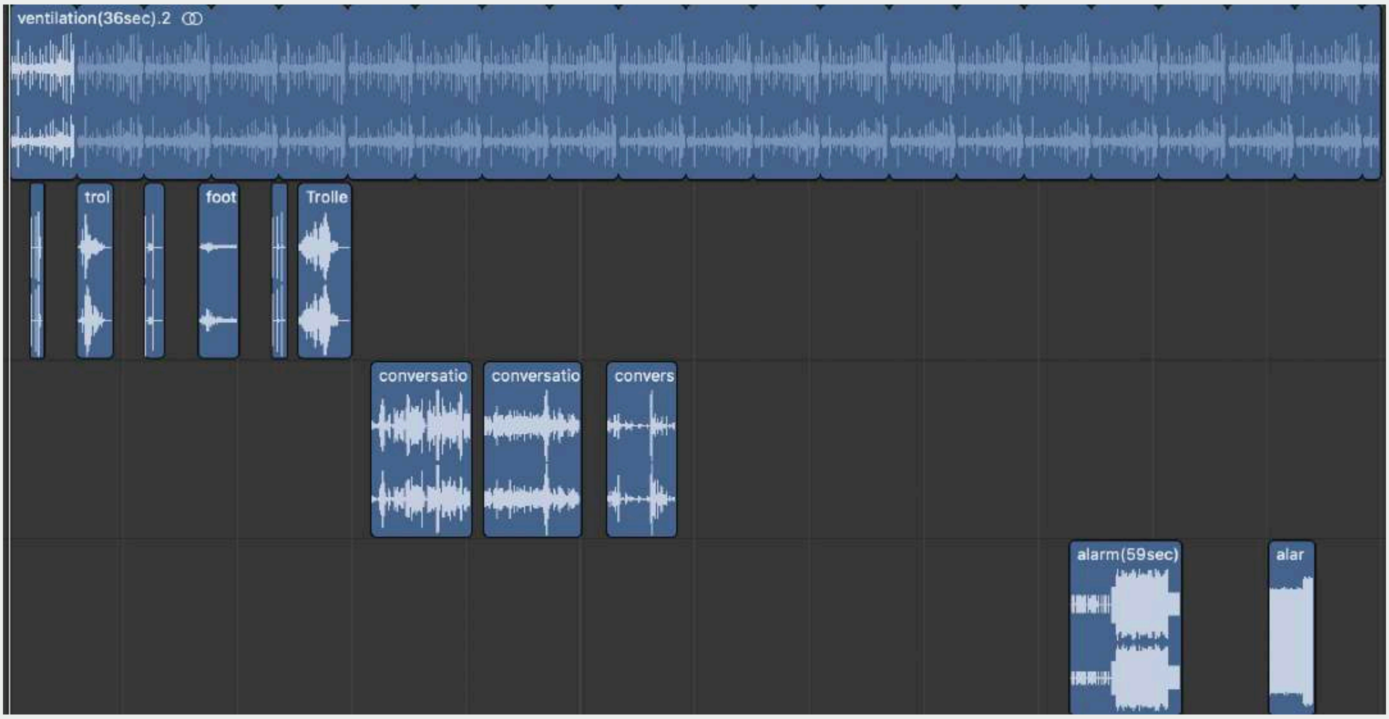
Figure 82. Usability test format1: Overall system usability assessment

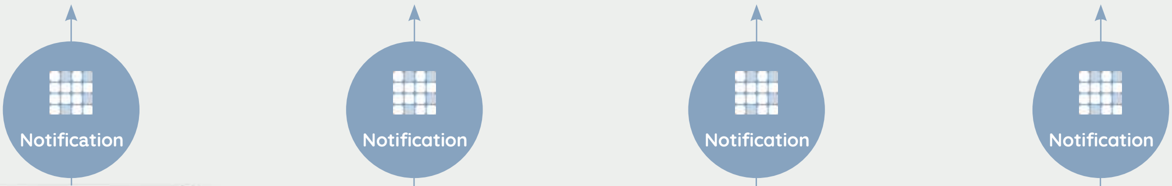
Cacophony Mapper application usability evaluation				
Cacophony Mapper app was **manageable to learn.**	Cacophony Mapper app was **clearly structured.**	Cacophony Mapper app was **easy to use.**	Cacophony Mapper app was **straightforward to use.**	I think it will be **practical** to use Cacophony Mapper app during work in a medical environment.
-/5.0	-/5.0	-/5.0	-/5.0	-/5.0
-/25				

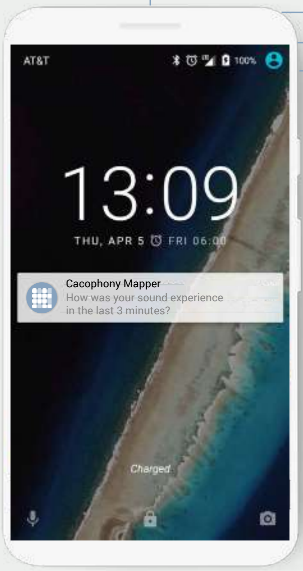
Figure 83. Usability test format2: Application usability assessment

User test plan: test 2, 3









AT&T

13:09

THU, APR 5 ⏰ FRI 06:00

Cacophony Mapper
How was your sound experience in the last 3 minutes?

Charged

Testing file
(12 mins)

Figure 84. User test sound file

User test plan: test 2, 3

Test plan

The duration of the test was designed for 20 minutes in total, having 3 minutes of introduction, 12 minutes of the testing, and 5 minutes for the after-test-survey.

For instruction for the test procedure, a printed A4 paper with graphics in the next two pages will be given to participants to give clear guidance of what they will expect through the experiment. Also, the after-test-survey is mainly based on Attrakkdiff scale in Figure 81, which is a design usability assessment tool using word pairs.

Also, in the testing stage, a sound recording, which has a duration of 12 minutes will be played (recording file available: Figure 84) and test participants will be asked to work on their daily work, using a laptop. In the meantime, they are going to be exposed to the sound recording. There will be notifications every 3 minutes, asking participants' emotional responses to their own sound experiences for the past 3 minutes. The guidance in the instruction paper explains six steps that the participants need to follow during the test, and participants fulfill six tasks, one by one and score the usability of each process.

The recording is composed with 11 sound samples, one with the background with dominant ventilator sound, a trolley, and footsteps, a trolley dragging sound, a footstep sound, three times of door slams, three different conversations, and two different alarms. In the meantime, the ventilator sound of the ICU will be played for 12 minutes as background noise.

In the first phase of the recording, three times of door slams, a combination of trolley dragging sound and footsteps, a recording of footsteps, and trolley dragging sound will be played for the first 3 minutes to stand for an incidental sound group. After 3 minutes, three different conversational noise will be played for the next 3 minutes. There will be only ventilator sound in the background played for 3 minutes in the third phase of the recording, and there will be 3 minutes of 2 different alarms played in the last phase. After each segment of 3 minutes, users will be asked to report their emotions toward their previous sound experiences through an application.

The test will be done by completing two sheets of a questionnaire, which has step-by-step instruction on the first page and general assessment of the application usability on the next page (available in Figure 85, 86). The questionnaire on the first page is giving guidance how the heart rate tracker and sound sensor should be worn, how to log in, how to set up the time for the detection, how to get access to report page by swiping notification, and how emotion report can be done. In the next page, the usability of the application will be assessed by asking whether the app was manageable to use, clearly structured, complicated to use, straightforward to use, or it would be practical to use it in the medical environment while working.

Figure 80. Attrakkdiff scale word pairs

User test plan : test3 questionnaire 1/2

Cacophony Mapper: Defining Noise Fatigue in the ICU

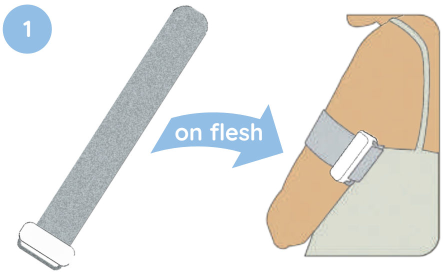
Name :

Score : **/ 12**

Gender : **Age :**

The purpose of this project is to find how sound stimuli can affect nurses' stress level and emotional responses in the ICU. Thus, you are going to be asked to wear a heart rate detection device and a sound collector during the test. Sound data will be distorted using a filter while recording, so no one can listen or restore the original sound that you don't need to worry about the privacy issue. The data will be only used for academic purpose. Please follow the instruction below and rate your usability of each process. The questionnaire on the next page should be filled in after the test. The duration of the recordning will be 12 minutes and you do not need to try to remember the sound from the recording. Thank you for your participation and enjoy!

1



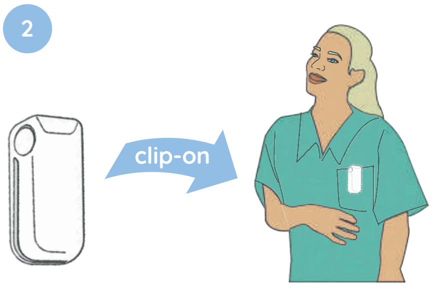
Why?

Please fasten the armband on your outer side of the bare upper arm for the heart rate collection.

Easy —— **Okay** —— **Difficult**

2 1 0

2



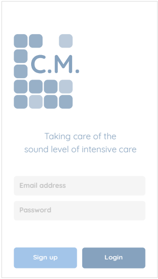
Why?

Please wear the clip the microphone device on your chest area.

Easy —— **Okay** —— **Difficult**

2 1 0

3



Taking care of the sound level of intensive care

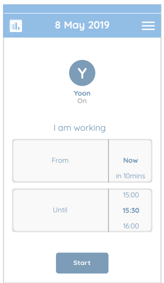
Why?

Please Sign in to Cacophony Mapper application, using Email and password written below:
Emai l: designer.yoon.lee@gmail.com
Password : dldbs

Easy —— **Okay** —— **Difficult**

2 1 0

4



I am working

From: Now / in 10mins

Until: 15:00 / 15:30 / 16:00

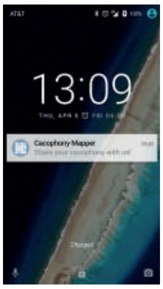
Why?

Please **start HB button** to connect the device to the Bluetooth. Assign the time slot **from now to 12 minutes** later and press **ttart rec button** and enjoy your work! :)

Easy —— **Okay** —— **Difficult**

2 1 0

5



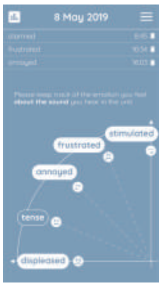
Why?

When there is a notification, please start the application by swiping the notification from the left to the rigt and report your emotion.

Easy —— **Okay** —— **Difficult**

2 1 0

6



Why?

Please press the emotion button which was relevant to your emotion related to your recent sound experience.

Easy —— **Okay** —— **Difficult**

2 1 0

Figure 85. Instruction & questionnaire for system usability assessment

User test plan : test3 questionnaire 2/2

The SUS(System Usability Scale) Survey for Cacophony Mapper application

1. Cacophony Mapper application was manageable to learn.

Strongly disagree ○ 1 ○ 2 ○ 3 ○ 4 ○ 5 Strongly agree

Why do you think so?

2. Cacophony Mapper application was clearly structured.

Strongly disagree ○ 1 ○ 2 ○ 3 ○ 4 ○ 5 Strongly agree

Why do you think so?

3. Cacophony Mapper application was complicated to use.

Strongly disagree ○ 1 ○ 2 ○ 3 ○ 4 ○ 5 Strongly agree

Why do you think so?

4. Cacophony Mapper interface was straghtforward to use.

Strongly disagree ○ 1 ○ 2 ○ 3 ○ 4 ○ 5 Strongly agree

Why do you think so?

5. I think it will be practical to used Cacophony Mapper during work in a medical environment.

Strongly disagree ○ 1 ○ 2 ○ 3 ○ 4 ○ 5 Strongly agree

Why do you think so?

Comments?

Figure 86. Application usability assessment

System reliability test process: test 1

Test 1
-What is the overall sound classification success rate?
-What is the sound classification success rate of each group?

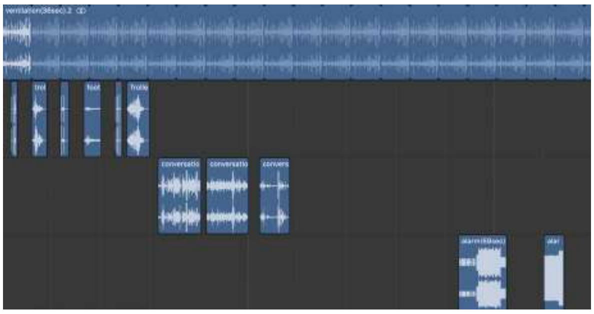


Figure 87. Sound track for user experiment

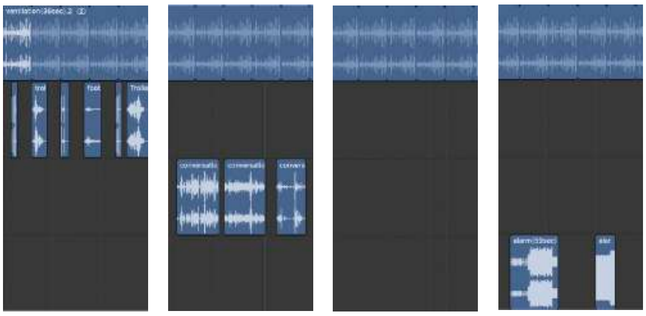


Figure 88. Segmented sound track which stand for incidental noise, conversation, machinery noise from the background and alarms

First, the sound for the testing has four different parts, that stand for incidental noise, conversation, machinery noise in the background, and alarms, respectively in sequence. The whole recording is 12 minutes, and every four segments take 3 minutes. Since each segment was included in four different categories, the recording could not go straightly to the sound classifier because the system will detect the four segments as a chunk so that it will give only one classification result. To avoid that, I divided one sound file into four segments so the classifier can detect each sound group more clearly.

Four sound files and 11 sound sources which compose the soundtrack were put into sample number-amplitude filter in Matlab audio toolbox for sound analysis first. Also, these files were applied to FFT(Fast Fourier Transform) filter to draw three frequency peaks and power peaks, for further sound investigation.

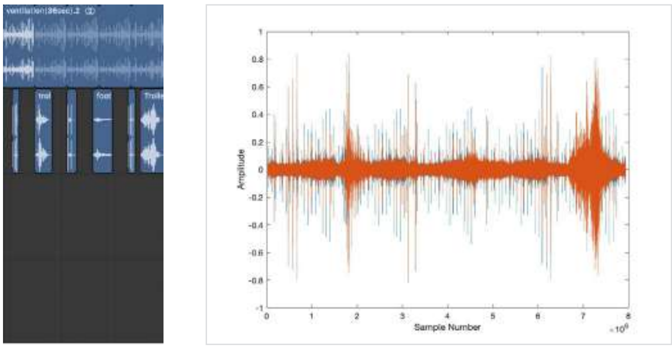


Figure 89. Sound track phase 1 which stands for incidental noise, applied to sample number-amplitude filter

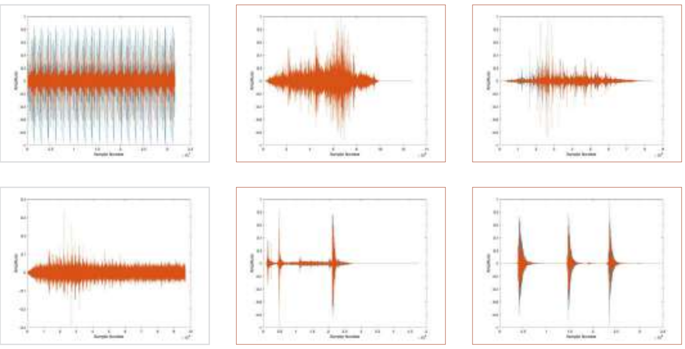


Figure 90. Sound sorces that compose a sound track, applied to sample number-amplitude filter: machinery noise, trolley passing sound, trolley passing and footsteps, footsteps, door slam 1, door slam 2 from the left top

The first phase of the soundtrack was applied to sample number-amplitude filter, with its sound elements of machinery noise in the background, trolley passing sound, trolley passing and footsteps, and two different door slam sounds, separately. The figure above shows the sound patterns that each sound makes. Even though six sound sources were combined into one sound, the pattern does not reflect the combination of 6 sound waveforms. It is because sound data of each sound source offsets each other, so the waveform of the combined soundtrack has its own pattern. The combined waveform showed the general shape of the incidental sound waveform, while separated files showed the typical pattern of its sub-categories, such as trolley dragging, door slam, and footsteps, which was investigated in chapter 3.11.

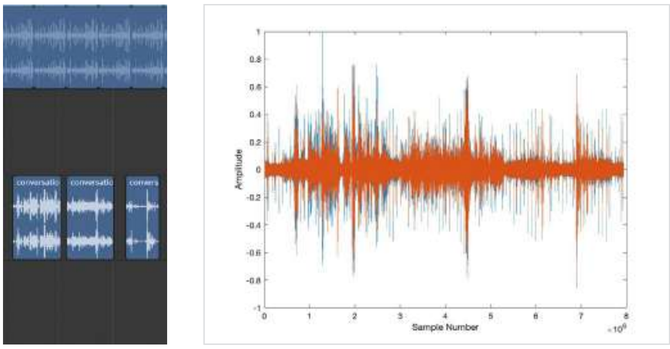


Figure 91. Sound track phase 2 which stands for conversation, applied to sample number-amplitude filter

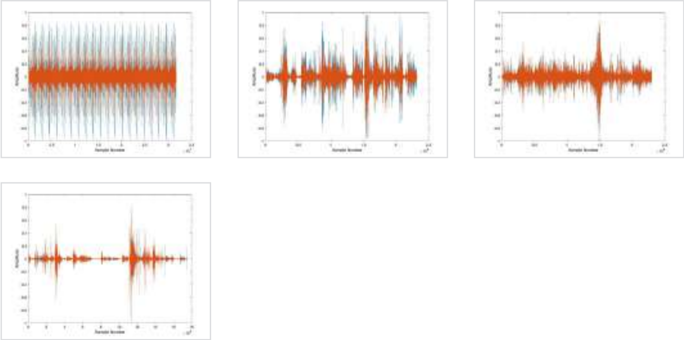


Figure 92. Sound sorces that compose a sound track, applied to sample number-amplitude filter: machinery noise, three conversation sound sources from the left top

The second phase of the soundtrack was applied to sample number-amplitude filter, with sound elements of machinery noise in the background, and three different conversation sound. As can be seen from the figure above, visual significance was not found in a combined file.

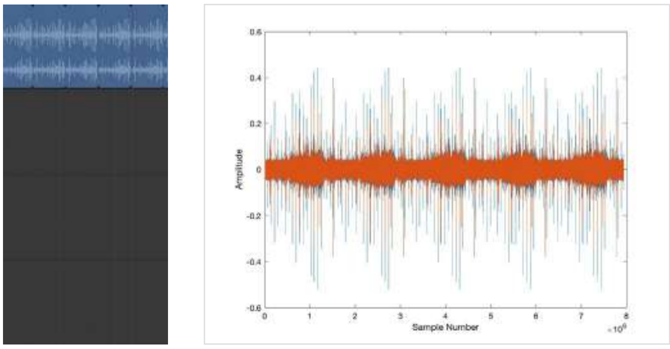


Figure 93. Sound track phase 3 which stands for machinery noise, applied to sample number-amplitude filter

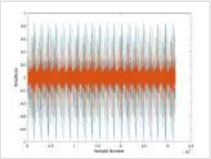


Figure 94. Sound sorces that compose a sound track, applied to sample number-amplitude filter: machinery noise

The waveform shows the repetitive pattern of ventilator sound. The significant point is that the waveform shows a difference when the source is collected from a synthesized file and from the original file. It seems that the sound synthesis process affected the quality of the recording even though it was also done in WAV form. The duration of the sound file could affect the shape of the waveform since the original file has around 30 seconds as its duration while the whole recording is 12 minutes long.

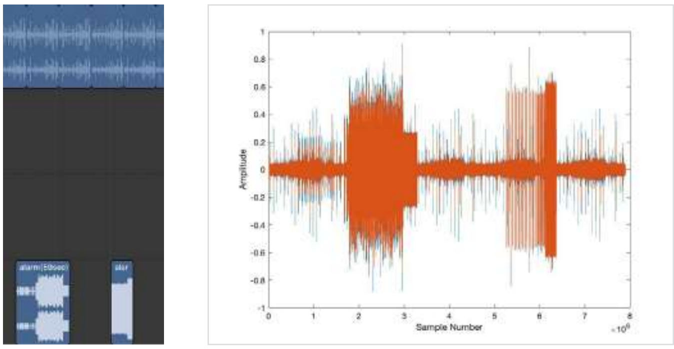


Figure 95. Sound track phase 4 which stands for alarms, applied to sample number-amplitude filter

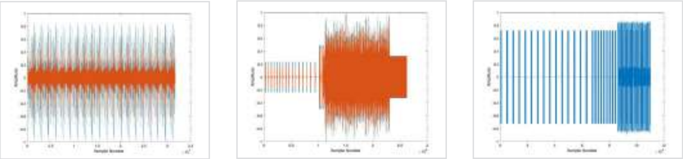
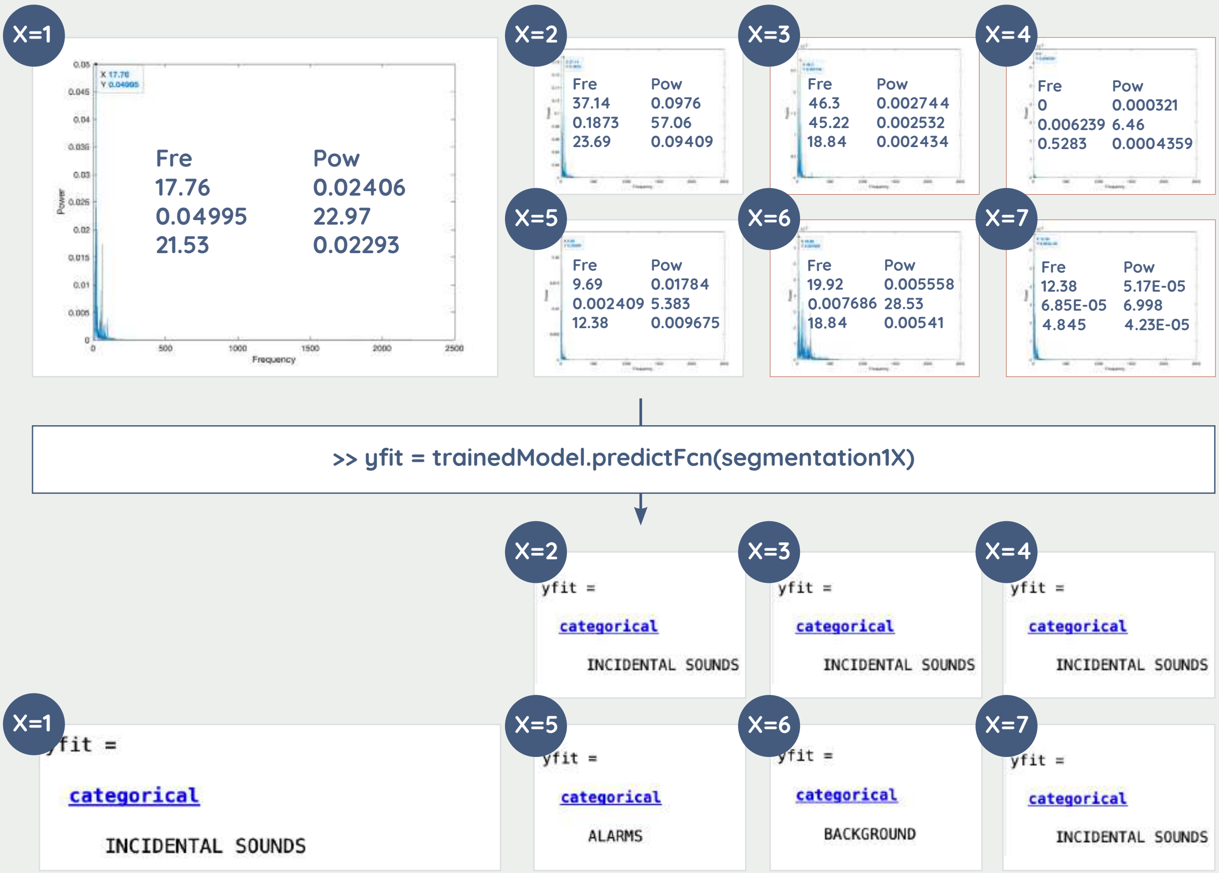


Figure 96. Sound sorces that compose a sound track, applied to sample number-amplitude filter: machinery noise, and two different alarms.

The alarms show a unique pattern in the waveform. Even though the sound offset was made in the sound synthesis process, the pattern shows a distinct pattern with a rigid shape. When comparing the pattern of original sound sources and the pattern of a synthesized file, the compiled file contains the pattern of original files in a more condensed way. The pattern looks artificial comparing to other sound sources, which show more angular and artificial shape in the sound pattern.

Secondly, the same sound sources were applied to FFT (Fast Fourier Transform) filter to extract three frequency peaks and three power peaks. Succeedingly, those sound peaks were applied sound classifier, which was created in chapter 3.12 to find the sound classification result.

Incidental



X=1

Fre	Pow
17.76	0.02406
0.04995	22.97
21.53	0.02293

X=2

Fre	Pow
37.14	0.0976
0.1873	57.06
23.69	0.09409

X=3

Fre	Pow
46.3	0.002744
45.22	0.002532
18.84	0.002434

X=4

Fre	Pow
0	0.000321
0.006239	6.46
0.5283	0.0004359

X=5

Fre	Pow
9.69	0.01784
0.002409	5.383
12.38	0.009675

X=6

Fre	Pow
19.92	0.005558
0.007686	28.53
18.84	0.00541

X=7

Fre	Pow
12.38	5.17E-05
6.85E-05	6.998
4.845	4.23E-05

```
>> yfit = trainedModel.predictFcn(segmentation1X)
```

X=2

```
yfit =

  categorical

     INCIDENTAL SOUNDS
```

X=3

```
yfit =

  categorical

     INCIDENTAL SOUNDS
```

X=4

```
yfit =

  categorical

     INCIDENTAL SOUNDS
```

X=1

```
yfit =

  categorical

     INCIDENTAL SOUNDS
```

X=5

```
yfit =

  categorical

     ALARMS
```

X=6

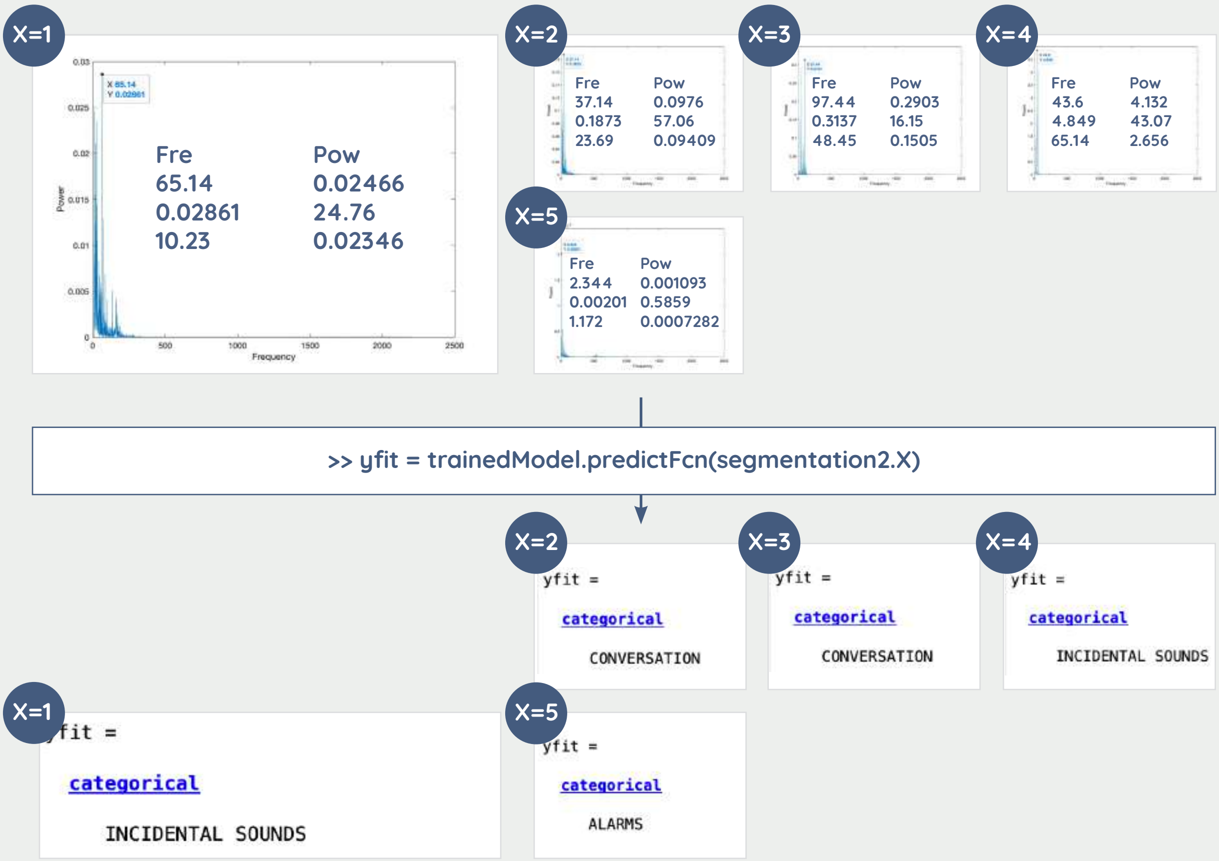
```
yfit =

  categorical

     BACKGROUND
```

X=7

```
yfit =

  categorical

     INCIDENTAL SOUNDS
```

Figure 97. Sound track phase 1, applied to FFT filter

Conversation



X=1

Fre	Pow
65.14	0.02466
0.02861	24.76
10.23	0.02346

X=2

Fre	Pow
37.14	0.0976
0.1873	57.06
23.69	0.09409

X=3

Fre	Pow
97.44	0.2903
0.3137	16.15
48.45	0.1505

X=4

Fre	Pow
43.6	4.132
4.849	43.07
65.14	2.656

X=5

Fre	Pow
2.344	0.001093
0.00201	0.5859
1.172	0.0007282

```
>> yfit = trainedModel.predictFcn(segmentation2.X)
```

X=2

```
yfit =

  categorical

     CONVERSATION
```

X=3

```
yfit =

  categorical

     CONVERSATION
```

X=4

```
yfit =

  categorical

     INCIDENTAL SOUNDS
```

X=1

```
yfit =

  categorical

     INCIDENTAL SOUNDS
```

X=5

```
yfit =

  categorical

     ALARMS
```

Figure 98. Sound track phase 2, applied to FFT filter

Machinery

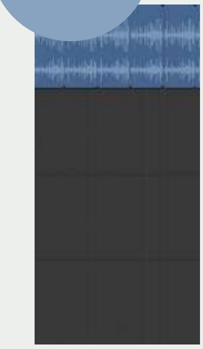
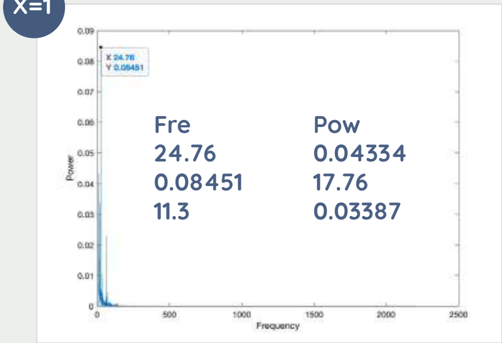


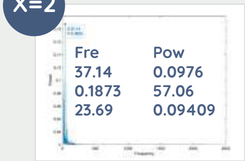
Figure 99. Sound track phase 3, applied to FFT filter

X=1



Fre	Pow
24.76	0.04334
0.08451	17.76
11.3	0.03387

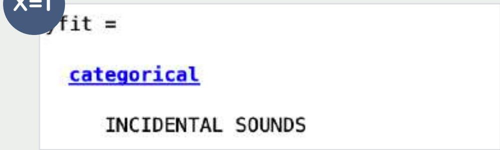
X=2



Fre	Pow
37.14	0.0976
0.1873	57.06
23.69	0.09409

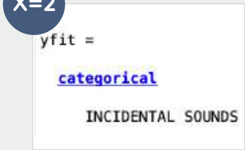
```
>> yfit = trainedModel.predictFcn(segmentation3.X)
```

X=1



```
yfit =

  categorical

     INCIDENTAL SOUNDS
```

X=2

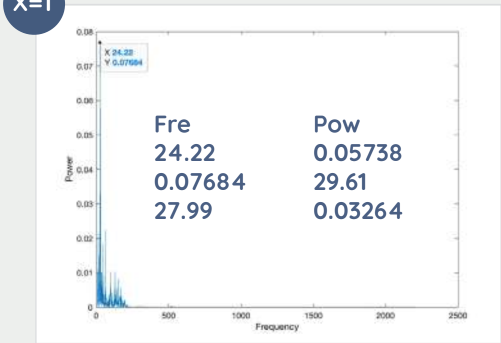


```
yfit =

  categorical

     INCIDENTAL SOUNDS
```

Alarms

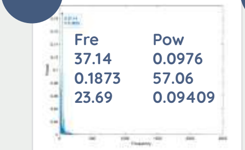
Figure 100. Sound track phase 3, applied to FFT filter Secondly, the same sound sources were

X=1



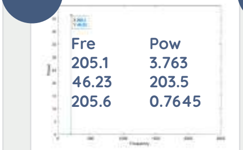
Fre	Pow
24.22	0.05738
0.07684	29.61
27.99	0.03264

X=2



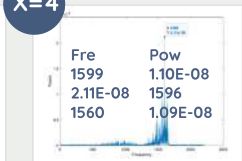
Fre	Pow
37.14	0.0976
0.1873	57.06
23.69	0.09409

X=3



Fre	Pow
205.1	3.763
46.23	203.5
205.6	0.7645

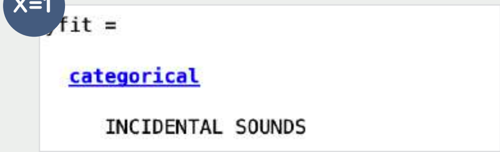
X=4



Fre	Pow
1599	1.10E-08
2.11E-08	1596
1560	1.09E-08

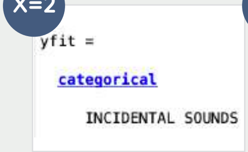
```
>> yfit = trainedModel.predictFcn(segmentation4.X)
```

X=1



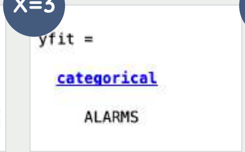
```
yfit =

  categorical

     INCIDENTAL SOUNDS
```

X=2



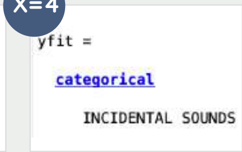
```
yfit =

  categorical

     INCIDENTAL SOUNDS
```

X=3



```
yfit =

  categorical

     ALARMS
```

X=4



```
yfit =

  categorical

     INCIDENTAL SOUNDS
```

Four synthesized sound crops of incidental sound, conversation, machinery, and alarm were applied to FFT filter to find the top three frequency peaks and power peaks. Dataset was applied to Matlab classifier using given code in figure97 to 100, to find out the sound category. In the same way, 11 separate sound sources were put into the same classifier to find out the accuracy of the sound classifier.

System reliability test result: test 1

Using sound file for testing (WAV file)

Using synthesized sound chunks

	Phase 1	Phase 2	Phase 3	Phase 4
Recording	Incidental (3min)	Conversation (3min)	Machinery (3min)	Alarms (3min)
Sensing result	Incidental	Incidental	Incidental	Incidental
Correct	Yes	No	No	No
Success rate	100%	0%	0%	0%
	25%			

Figure 101. Sound classification result: WAV

Result 1.1

The classification result shows that the accuracy of the filter when putting four synthesized sound source was 25%. The filter showed only one correct result in the first phase of the recording for incidental sound. All classification test result showed incidental sound as their classification results; it was because the sound category classifier recognizes 3 minutes of phases as one chunk. Since incidental sound filter has multiple sound groups, such as footsteps, object clashing sounds, trolley dragging sound, and door slams, it could have been possible that sound of 3 minutes has been recognized as incidental sound as a whole. For further investigation, I decided to run the classifier using 11 sound files which were used to make a synthesized sound recording for testing.

Using 11 separate sound sources

	Incidental					Conversation			Machinery	Alarms	
Recording	Trolley (29sec)	Trolley and Footsteps (19sec)	Footsteps (22sec)	Door slam 1 (6sec)	Door slam 2 (8sec)	Conversation1 (53sec)	Conversation2 (52sec)	Conversation3 (36sec)	Ventilator (12min)	Alarm 1 (59sec)	Alarm 2 (25sec)
Sensing result	Incidental	Incidental	Alarms	Background	Incidental	Conversation	Conversation	Alarms	Incidental	Alarms	Incidental
Correct	Yes	Yes	No	No	Yes	Yes	Yes	No	No	Yes	No
Success rate	60%					66.6%			0%	50%	
	54%										

Figure 102. Sound classification result: recordings from user test

Classification result showed that 3 out of the five incidental sounds were recognized correctly, recording 60% of accuracy level. When it comes to conversation, 2 out of the three results were correct, showing 66.6% of accordance, while one long ventilator sound of 12 minutes was not recognized as a machinery sound. Furthermore, only one sort of alarms was recognized out of two recordings. Except for the ventilator recording, the duration of each sound file did not exceed a minute, so it did not seem the sound composition of the recording was too complicated.

All in all, the classification filter showed 54% of the accuracy level, which showed more than twice better the performance than when using a sound chunk. I thought the accuracy varies depending on the sound input mainly because the classifier has only 200 sound recordings for its machine learning, and some groups only had 25 samples for its classification.

Therefore, 11 sets of test sound data for an individual sound group was applied to the classifier so that it could improve the classification result with the first optimization. Therefore, a new classifier with 66.2% of accuracy level was created (see Figure 103), and the same test was conducted to see if my assumption below was right.
; The low performance of the classifier is coming from small sound sample number and it can be solved by improving the sound classifier quality. By enlarging sample dataset with sounds which will be used for the sound synthesis, the accuracy of the classification will be improved.

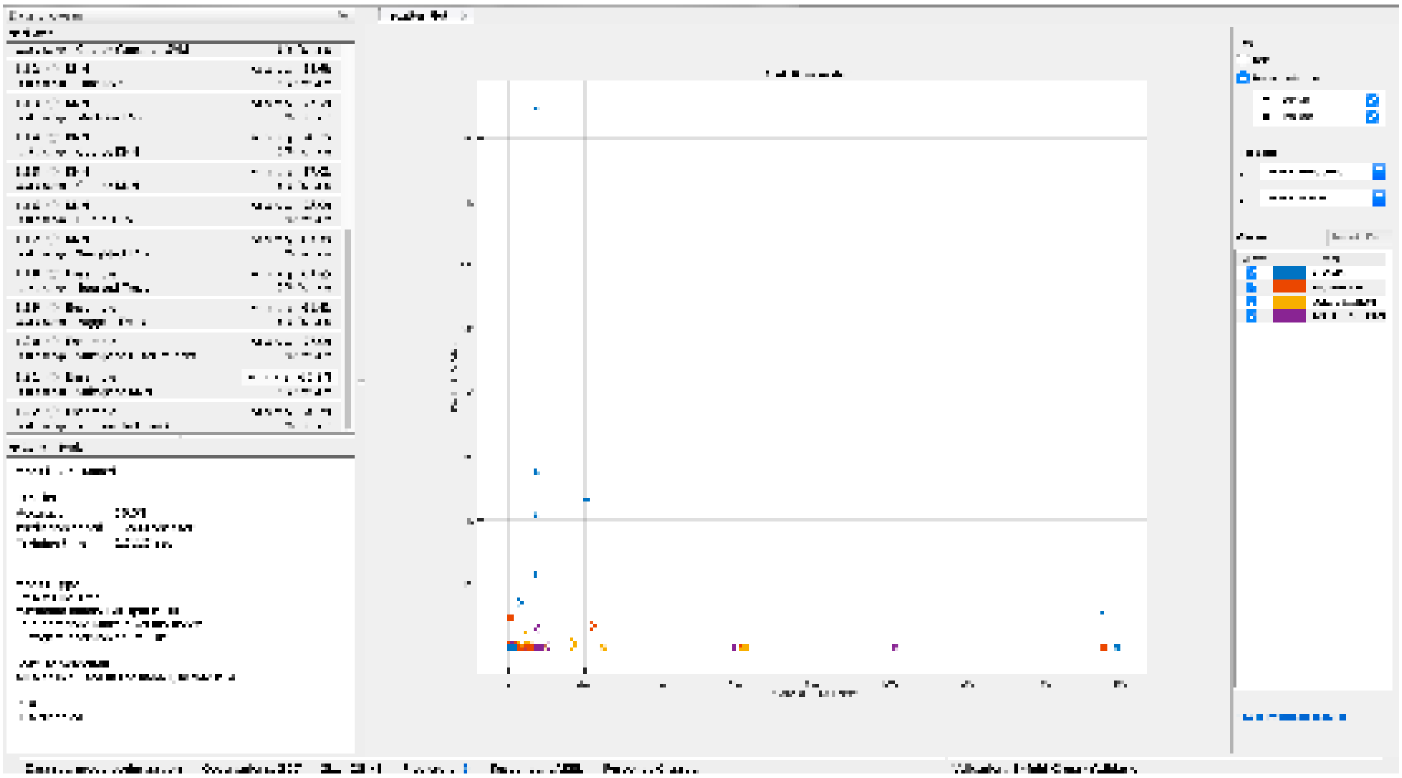


Figure 103. Sound classification filter optimization-1

Using synthesized sound chunks

	Phase 1	Phase 2	Phase 3	Phas
Recording	Incidental (3min)	Conversation (3min)	Machinery (3min)	Alarms (3min)
Sensing result	Incidental	Incidental	Incidental	Incidental
Correct	Yes	No	No	No
Success rate	100%	0%	0%	0%
	25%			

Figure 104. Sound classification result: WAV

Using 11 separate sound sources

	Incidental					Conversation			Machinery	Alarms	
Recording	Trolley (29sec)	Trolley and Footsteps (19sec)	Footsteps (22sec)	Door slam 1 (6sec)	Door slam 2 (8sec)	Conversation1 (53sec)	Conversation2 (52sec)	Conversation3 (36sec)	Ventilator (12min)	Alarm 1 (59sec)	Alarm 2 (25sec)
Sensing result	Background	Incidental	Incidental	Incidental	Incidental	Conversation	Conversation	Conversation	Background	Alarms	Alarms
Correct	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Success rate	80%					100%			100%	100%	
	90.9%										

Figure 105. Sound classification result: recordings from user test

Result 1.2

Two tests were conducted with new classifier with 66.6% of accuracy. The first test was conducted using four synthesized sound chunks for incidental, conversation, machinery, and alarms, and the overall result showed 25% of classification success rate. The classifier could only tell the incidental sounds like the same as the classification result before the sound filter optimization.

On the other hand, the test result with 11 separate sound sources showed a significant improvement in accuracy level, dramatically increasing to 90.9% from 54%. When looking at the result, all sound classification results were all correct except for one trial with incidental sound.

Since there was no significant improvement when using synthesized sound chunks, the way of improving the recognition quality of synthesized sound was contemplated. The main reason of the low job performance of the classifier seemed to be coming from the fact that there are so many sound events in 3 minutes of the recording, and the sound classifier recognizes the whole as one sound event, which leads to incorrect results.

Therefore, a way of dividing one sound chunk into every 10 seconds was devised in the sound processing phase to avoid the classifier detects 3 minutes of a recording as one sound. Therefore, the total of 72 sound samples with 10 seconds of duration in each recording was made. An expected downside of this processing method for this test-setup is that if there are gaps in between each sound event, there will be only machinery noise left in the background so the classifier will detect those sound gaps as background than the intended sound group, which lowers the classification success rate. Therefore, I decided to use the term sound dominancy than classification accuracy in this section. Since the whole recording is not filled with specific sound events, the most recognized sounds should be determined as the result of the classification than counting the overall classification success rate. Therefore, 72 sound samples were put into Matlab classifier, so the most recognized sound could be decided as each segment's dominant sound.

	Incidental								
Recording	**0-10 sec**	**11-20 sec**	**21-30 sec**	**31-40 sec**	**41-50 sec**	**51-60 sec**	**61-70 sec**	**71-80 sec**	**81-90 sec**
Sensing result	Incidental	Incidental	Conversation	Background	Incidental	Conversation	Incidental	Incidental	Incidental
Correct	Yes	Yes	No	No	Yes	No	Yes	Yes	Yes

	Incidental								
Recording	**91-100 sec**	**101-110 sec**	**111-120 sec**	**121-130 sec**	**131-140 sec**	**141-150 sec**	**151-160 sec**	**161-170 sec**	**171-180 sec**
Sensing result	Incidental	Incidental	Incidental	Background	Incidental	Incidental	Incidental	Alarms	Incidental
Correct	Yes	Yes	No	No	Yes	Yes	Yes	No	Yes
Dominant sound	**Incidental : 66.6% dominancy** (> Conversation : 11.1% = Background : 11.1% > Alarms : 5.5%)								

Figure 106. Sound classification result: phase 1-Incidental

	Conversation								
Recording	**0-10 sec**	**11-20 sec**	**21-30 sec**	**31-40 sec**	**41-50 sec**	**51-60 sec**	**61-70 sec**	**71-80 sec**	**81-90 sec**
Sensing result	Incidental	Incidental	Incidental	Conversation	Conversation	Conversation	Incidental	Incidental	Background
Correct	No	No	No	Yes	Yes	Yes	No	No	No

	Conversation								
Recording	**91-100 sec**	**101-110 sec**	**111-120 sec**	**121-130 sec**	**131-140 sec**	**141-150 sec**	**151-160 sec**	**161-170 sec**	**171-180 sec**
Sensing result	Incidental	Incidental	Alarms	Conversation	Incidental	Conversation	Incidental	Incidental	Incidental
Correct	No	No	No	Yes	No	Yes	No	No	No
Dominant sound	**Incidental : 61.1% dominancy** (**> Conversation : 27.7%** > Background : 5.5% = Alarms : 5.5%)								

Figure 107. Sound classification result: phase 2-Conversation

	Machinery								
Recording	**0-10 sec**	**11-20 sec**	**21-30 sec**	**31-40 sec**	**41-50 sec**	**51-60 sec**	**61-70 sec**	**71-80 sec**	**81-90 sec**
Sensing result	Incidental	Incidental	Incidental	Conversation	Conversation	Conversation	Incidental	Incidental	Background
Correct	No	No	No	No	No	No	No	No	Yes

	Machinery								
Recording	**91-100 sec**	**101-110 sec**	**111-120 sec**	**121-130 sec**	**131-140 sec**	**141-150 sec**	**151-160 sec**	**161-170 sec**	**171-180 sec**
Sensing result	Incidental	Incidental	Alarms	Conversation	Incidental	Conversation	Incidental	Incidental	Incidental
Correct	No	No	No	No	No	No	No	No	No
Dominant sound	**Incidental :** 61.1% dominancy (> Conversation : 27.7% **> Background : 5.5%** = Alarms : 5.5%)								

Figure 108. Sound classification result: phase 3-Machinery

Figure 106. Sound classification result: phase 4-Alarms

	Alarms								
Recording	**0-10 sec**	**11-20 sec**	**21-30 sec**	**31-40 sec**	**41-50 sec**	**51-60 sec**	**61-70 sec**	**71-80 sec**	**81-90 sec**
Sensing result	Incidental	Incidental	Alarms	Incidental	Alarms	Alarms	Alarms	Background	Incidental
Correct	No	No	Yes	No	Yes	Yes	Yes	No	No

	Alarms								
Recording	**91-100 sec**	**101-110 sec**	**111-120 sec**	**121-130 sec**	**131-140 sec**	**141-150 sec**	**151-160 sec**	**161-170 sec**	**171-180 sec**
Sensing result	Incidental	Incidental	Conversaion	Conversaion	Incidental	Incidental	Incidental	Incidental	Incidental
Correct	No	No	No	No	No	No	No	No	No
Dominant sound	**Incidental : 61.1% dominancy** (**> Alarms : 22.2%** > Conversation : 11.1% > Background : 5.5%)								

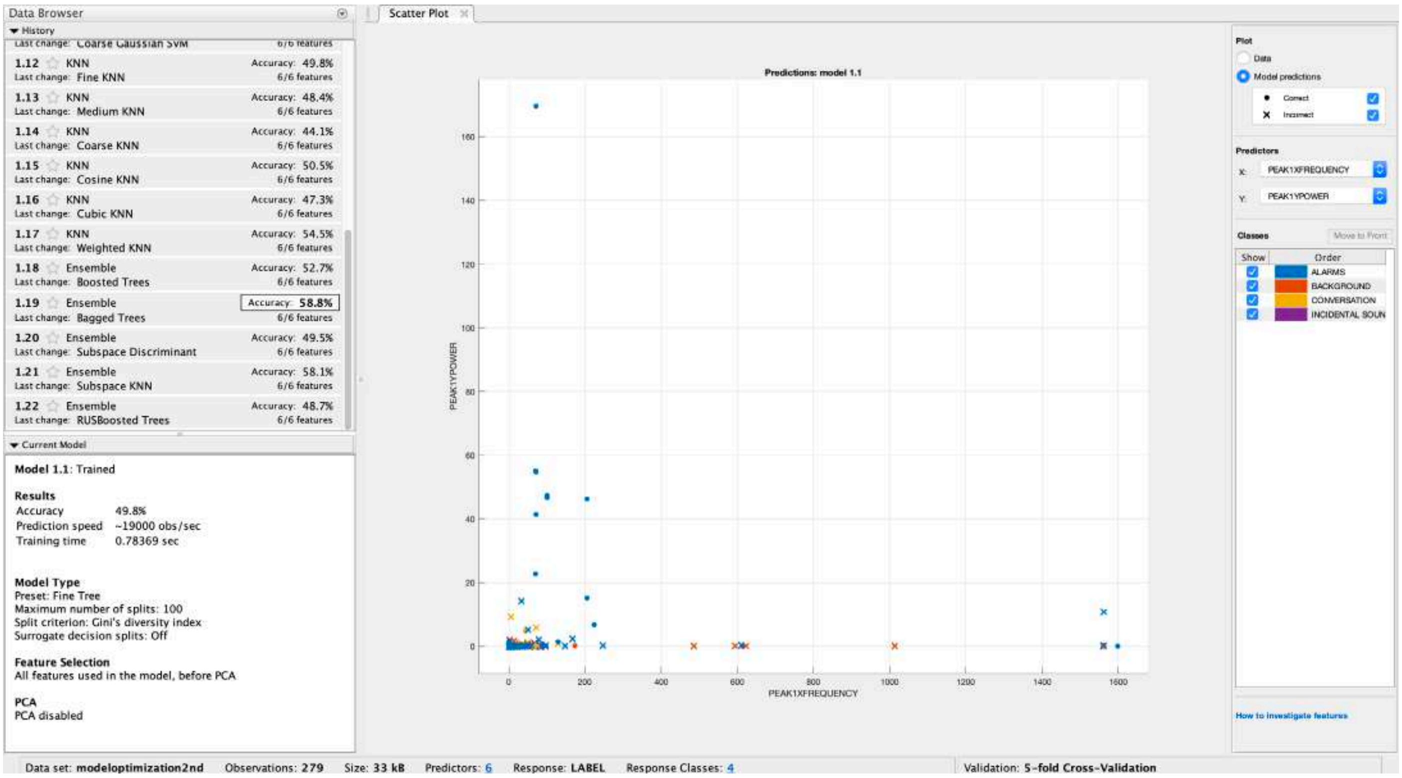
Overall: 30.5%

Result 1.3

As a result, Incidental sound group showed the most dominancy of 66.6% in the first phase of the recording, which stands for the incidental sound group. Incidental sound showed the domonancy of each of 61.1% for conversation, machinery, alarm phase, which was different than what those phases stand for. Even though the second dominant sound reflected its original sound in conversation and alarm segment, the overall dominance of the right sound group recorded 30.5%, which was way lower than when using separate sound sources.

As an optimization, the accuracy of the classifier was developed by including sound data from sample files. 72 cropped sound samples of each 10 seconds were added into Matlab classifier and re-trained. The optimized filter showed 58.8% of expected accuracy after its training (see Figure 107), and the same files were tested with the new classifier to see if optimization helped better classification result.

Figure 107. Sound classification filter optimization-2



Last change: Coarse Gaussian SVM 6/6 features

1.12 KNN Accuracy: 49.8%
Last change: Fine KNN 6/6 features

1.13 KNN Accuracy: 48.4%
Last change: Medium KNN 6/6 features

1.14 KNN Accuracy: 44.1%
Last change: Coarse KNN 6/6 features

1.15 KNN Accuracy: 50.5%
Last change: Cosine KNN 6/6 features

1.16 KNN Accuracy: 47.3%
Last change: Cubic KNN 6/6 features

1.17 KNN Accuracy: 54.5%
Last change: Weighted KNN 6/6 features

1.18 Ensemble Accuracy: 52.7%
Last change: Boosted Trees 6/6 features

1.19 Ensemble Accuracy: 58.8%
Last change: Bagged Trees 6/6 features

1.20 Ensemble Accuracy: 49.5%
Last change: Subspace Discriminant 6/6 features

1.21 Ensemble Accuracy: 58.1%
Last change: Subspace KNN 6/6 features

1.22 Ensemble Accuracy: 48.7%
Last change: RUSBoosted Trees 6/6 features

Model 1.1: Trained

Results
Accuracy 49.8%
Prediction speed ~19000 obs/sec
Training time 0.78369 sec

Model Type
Preset: Fine Tree
Maximum number of splits: 100
Split criterion: Gini's diversity index
Surrogate decision splits: Off

Feature Selection
All features used in the model, before PCA

PCA
PCA disabled

Data set: **modeloptimization2nd** Observations: **279** Size: **33 kB** Predictors: **6** Response: **LABEL** Response Classes: **4** Validation: **5-fold Cross-Validation**

	Incidental								
Recording	0-10 sec	11-20 sec	21-30 sec	31-40 sec	41-50 sec	51-60 sec	61-70 sec	71-80 sec	81-90 sec
Sensing result	Incidental	Background	Incidental	Incidental	Background	Incidental	Conversation	Incidental	Incidental
Correct	Yes	No	Yes	Yes	No	Yes	No	Yes	Yes

	Incidental								
Recording	91-100 sec	101-110 sec	111-120 sec	121-130 sec	131-140 sec	141-150 sec	151-160 sec	161-170 sec	171-180 sec
Sensing result	Incidental	Incidental	Incidental	Incidental	Incidental	Incidental	Incidental	Incidental	Incidental
Correct	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Success rate	**Incidental : 83.3%** (> Background : 11.1% > Conversation : 5.5% > Alarms : 0%)								

Figure 108. Sound classification result: phase 1-Incidental

	Conversation								
Recording	0-10 sec	11-20 sec	21-30 sec	31-40 sec	41-50 sec	51-60 sec	61-70 sec	71-80 sec	81-90 sec
Sensing result	Conversation	Conversation	Conversation	Conversation	Incidental	Alarms	Conversation	Conversation	Conversation
Correct	Yes	Yes	Yes	Yes	No	No	Yes	Yes	Yes

	Conversation								
Recording	91-100 sec	101-110 sec	111-120 sec	121-130 sec	131-140 sec	141-150 sec	151-160 sec	161-170 sec	171-180 sec
Sensing result	Conversation	Conversation	Conversation	Alarms	Incidental	Conversation	Conversaion	Background	Conversation
Correct	Yes	Yes	Yes	No	No	Yes	Yes	No	Yes
Success rate	**Conversation : 72.2%** (> Incidental : 11.1% = Alarms : 11.1% > Background : 5.5%)								

Figure 109. Sound classification result: phase 2-Conversation

	Machinery								
Recording	0-10 sec	11-20 sec	21-30 sec	31-40 sec	41-50 sec	51-60 sec	61-70 sec	71-80 sec	81-90 sec
Sensing result	Background	Background	Background	Background	Background	Background	Background	Background	Background
Correct	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

	Machinery								
Recording	91-100 sec	101-110 sec	111-120 sec	121-130 sec	131-140 sec	141-150 sec	151-160 sec	161-170 sec	171-180 sec
Sensing result	Background	Incidental	Incidental	Background	Alarms	Incidental	Background	Incidental	Background
Correct	Yes	No	No	Yes	No	No	Yes	No	Yes
Success rate	**Background : 72.2%** (> Incidental : 22.2% > Alarms : 5.5% > Conversation : 0%)								

Figure 110. Sound classification result: phase 3-Machinery

Figure 111. Sound classification result: phase 4-Alarms

	Alarms								
Recording	**0-10 sec**	**11-20 sec**	**21-30 sec**	**31-40 sec**	**41-50 sec**	**51-60 sec**	**61-70 sec**	**71-80 sec**	**81-90 sec**
Sensing result	Alarms	Alarms	Alarms	Conversation	Alarms	Alarms	Alarms	Alarms	Background
Correct	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	No

	Alarms								
Recording	**91-100 sec**	**101-110 sec**	**111-120 sec**	**121-130 sec**	**131-140 sec**	**141-150 sec**	**151-160 sec**	**161-170 sec**	**171-180 sec**
Sensing result	Alarms	Conversation	Background	Background	Incidental	Incidental	Background	Incidental	Background
Correct	Yes	No	No	No	No	No	No	No	No
Success rate	**Alarms : 44.4%** (> Background : 27.7% > Incidental : 16.6% > Conversation : 11.1%)								

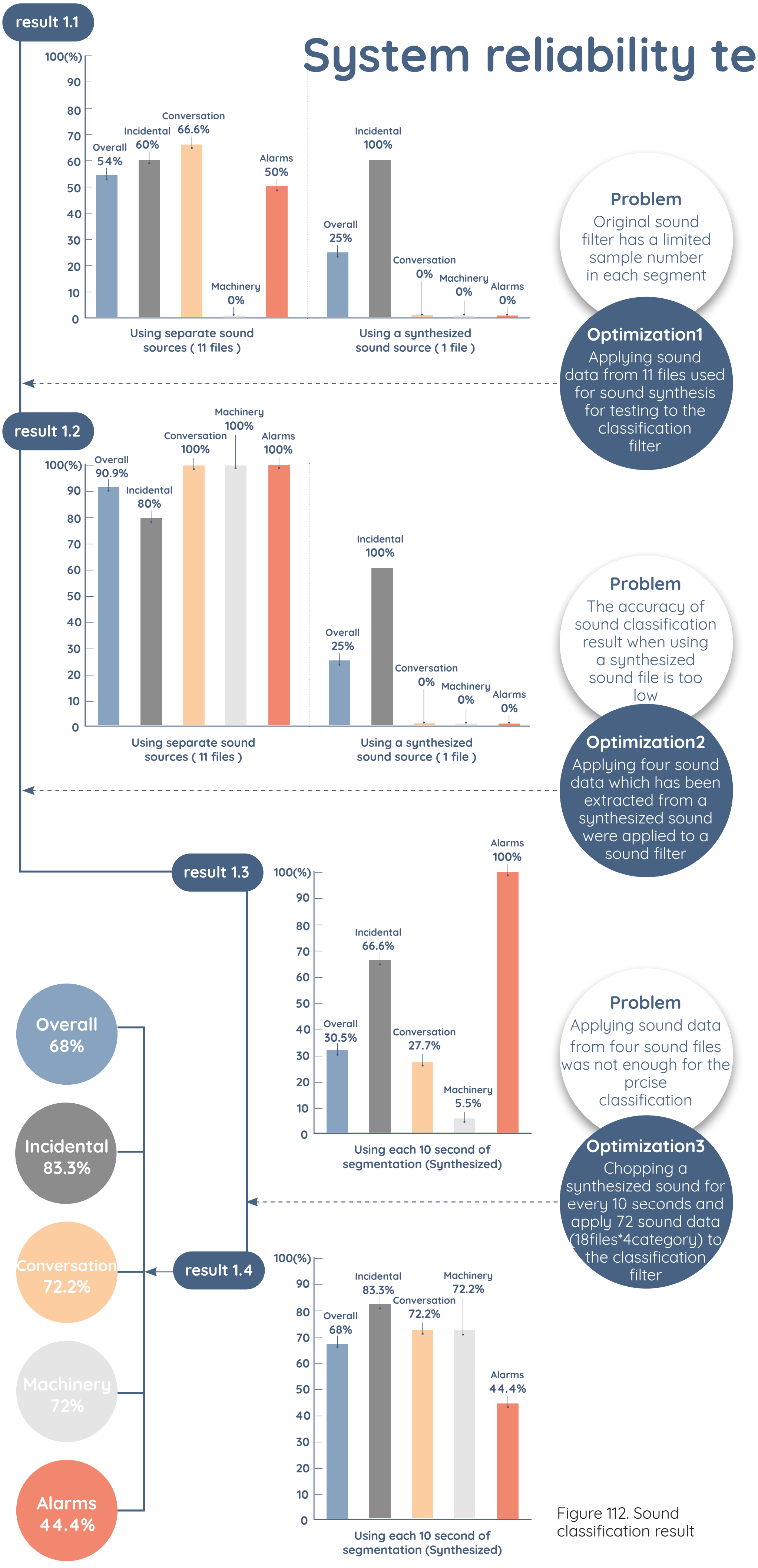
Overall: 68.0%

Result 1.4

After the second filter optimization, the overall accuracy of the classification result has improved drastically, showing 68.0% of the accordance rate, which is more than double the of the previous testing which had 30.5% of accordance rate. When it comes to incidental sound segment, it showed 83.3% of the accordance rate while the conversation and the machinery group showed 72.2%, respectively. Also, the alarm segment showed 44.4% of the match.

All in all, the accuracy of sound classification result with synthesized sound has been increased starting from 25% and ended up with 68%. The result shows that the 10-seconds-segmenting process was, and more relevant sound samples for classifier training will enable better classification quality.

System reliability test result: test 1



Problem
Original sound filter has a limited sample number in each segment

Optimization1
Applying sound data from 11 files used for sound synthesis for testing to the classification filter

Problem
The accuracy of sound classification result when using a synthesized sound file is too low

Optimization2
Applying four sound data which has been extracted from a synthesized sound were applied to a sound filter

Problem
Applying sound data from four sound files was not enough for the prcise classification

Optimization3
Chopping a synthesized sound for every 10 seconds and apply 72 sound data (18files*4category) to the classification filter

Figure 112. Sound classification result

Test 1 was conducted in four steps to see if the sound classifier works properly as it has been designed. In this process, several problems have been defined, especially regarding sound processing methods, and the filter was optimized together with the new processing methods so that optimized classifier can be used in test 2.

In result 1.1, using 11 separate sound sources scored overall of 54% accuracy while using synthesized sound sources showed 25% of the classification success rate. As the classifier has a total sample size of 200 and some group has only 25 samples in it, I thought applying the testing sound data to the filter can develop the quality of the classification.

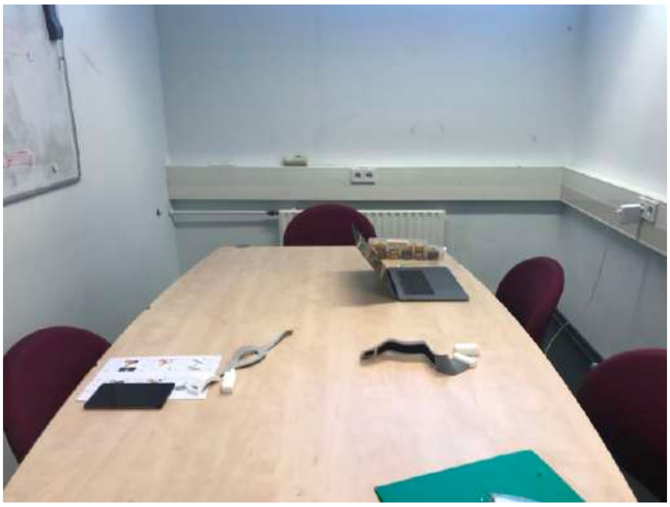
After the first optimization, the result 1.2 showed a significant improvement in the accuracy level of detection in 11 separate sound sources. However, there was no difference when using a synthesized sound source in the test, and the overall accuracy remained at 25%. I thought this difference occurs because the classifier uses only three frequency and three power peaks for the sound classification. Therefore, one recording of 3 minutes has too complicated sound components in one dataset. So, four sound segments of each 3 minutes were chopped into every 10 seconds, and 72 test samples were created.

After the second optimization, I focused on investigating synthesized sound dataset than separate sound sources since synthesized sound will be used in test 2 and the result will be compared to the result of test 1. The test result showed 30.5% of the accuracy, which showed 5.5% of the improvement than before the 10-second-sound-processing. The third optimization was done by including segmented dataset to the sound classifier so it can reflect the test dataset, complimenting the limitation of small sample groups.

The result 1.4 showed noticeable improvement after the third optimization, showing an overall accuracy of 68%. It showed significant improvements in each sound groups of incidental, conversation, and machinery sound too, except for alarms, which showed a plummeted classification success rate.

System reliability test process: test 2, 3

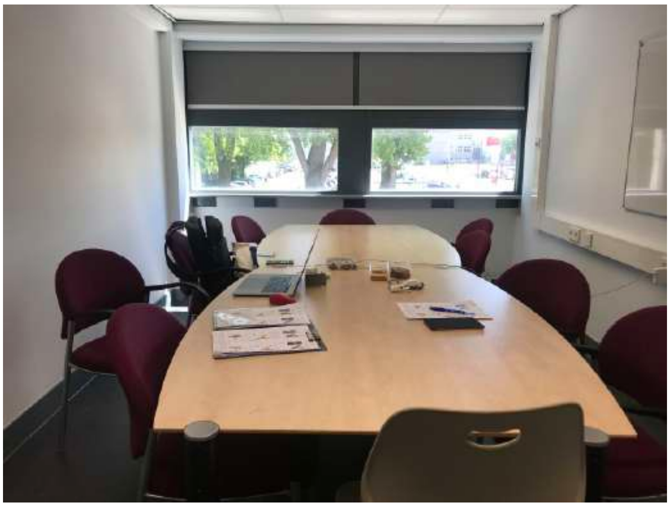
Test date: 14th May, 2019
Location: B-1-420, IDE, TU Delft



Test participants: Students (4)

Test date: 15th May, 2019
Location: D-1-900, IDE, TU Delft

Figure 113. Test setting



Test participants: Students (7)

Test materials: Mobile phone (Huawei Mediapad T3 7.0), Sound record device, Heart rate tracker, Instruction paper/Questionnaire , laptop (Macbook Pro 2018) as a speaker

The user test was conducted throughout two days, with 11 participants from the Industrial design department, TU Delft. The test was conducted with students in the faculty room, and there was a mobile phone (Huawei Mediapad T3 7.0), a sound record device, a heart rate tracking device, and a test sheet, which had instruction and questionnaire part, as test materials.

After 3 minutes of oral instruction, test devices, and a test sheet was given to participants. There were broadly six steps that test participants needed to follow, following guidance written on the instruction paper. After clicking the start recording button on the app, test participants were asked to refrain from talking or making big noises during the test to avoid additional impacts for the test result. The first part of the questionnaire was filled in while fulfilling six stages of tasks, and the second part of the questionnaire was finished after the test.

System reliability test result: test 2

Test 2
-What is the overall sound classification success rate?
-What is the sound classification success rate of each group?
-How the result different when using WAV file and sound recorded within the app?

Figure 114. Test result: participant 1

Participant1		
	Phase 1	Phase 2
Recording	Incidental	Conversation
Correct/total	1/18	0/18
Correct rate	5,6%	0%
	Phase 3	Phase 4
Recording	Machinery	Alarms
Correct/total	0/18	6/6
Correct rate	0%	100%
Overall: 11.7%		

Figure 115. Test result: participant 2

Participant2		
	Phase 1	Phase 2
Recording	Incidental	Conversation
Correct/total	4/18	0/18
Correct rate	22.2%	0%
	Phase 3	Phase 4
Recording	Machinery	Alarms
Correct/total	0/18	20/20
Correct rate	0%	100%
Overall: 32.4%		

Figure 116. Test result: participant 3

Participant3		
	Phase 1	Phase 2
Recording	Incidental	Conversation
Correct/total	2/18	0/18
Correct rate	11.1%	0%
	Phase 3	Phase 4
Recording	Machinery	Alarms
Correct/total	0/18	14/17
Correct rate	0%	82.3%
Overall: 22.5%		

Figure 117. Test result: participant 4

Participant4		
	Phase 1	Phase 2
Recording	Incidental	Conversation
Correct/total	3/18	0/18
Correct rate	16.7%	0%
	Phase 3	Phase 4
Recording	Machinery	Alarms
Correct/total	0/18	17/20
Correct rate	0%	85%
Overall: 27.0%		

Figure 118. Test result: participant 5

Participant5		
	Phase 1	Phase 2
Recording	Incidental	Conversation
Correct/total	4/18	4/18
Correct rate	22.2%	22.2%
	Phase 3	Phase 4
Recording	Machinery	Alarms
Correct/total	0/18	10/11
Correct rate	0%	90.9%
Overall: 27.7%		

Figure 119. Test result: participant 6

Participant6		
	Phase 1	Phase 2
Recording	Incidental	Conversation
Correct/total	3/18	1/18
Correct rate	16.7%	5.6%
	Phase 3	Phase 4
Recording	Machinery	Alarms
Correct/total	0/18	11/11
Correct rate	0%	100%
Overall: 23.1%		

Figure 120. Test result: participant 7

Participant7		
	Phase 1	Phase 2
Recording	Incidental	Conversation
Correct/total	3/18	3/18
Correct rate	16.7%	16.7%
	Phase 3	Phase 4
Recording	Machinery	Alarms
Correct/total	0/18	16/18
Correct rate	0%	88.9%
Overall: 30.6%		

Figure 121. Test result: participant 8

Participant8		
	Phase 1	Phase 2
Recording	Incidental	Conversation
Correct/total	0/18	2/18
Correct rate	0%	11.1%
	Phase 3	Phase 4
Recording	Machinery	Alarms
Correct/total	0/18	16/18
Correct rate	0%	88.9%
Overall: 25%		

Figure 122. Test result: participant 9

Participant9		
	Phase 1	Phase 2
Recording	Incidental	Conversation
Correct/total	0/18	3/18
Correct rate	0%	16.7%
	Phase 3	Phase 4
Recording	Machinery	Alarms
Correct/total	0/18	13/18
Correct rate	0%	72.2%
Overall: 22.2%		

Figure 123. Test result: participant 10

Participant10		
	Phase 1	Phase 2
Recording	Incidental	Conversation
Correct/total	2/18	2/18
Correct rate	11.1%	11.1%
	Phase 3	Phase 4
Recording	Machinery	Alarms
Correct/total	0/18	13/18
Correct rate	0%	72.2%
Overall: 23.6%		

Figure 124. Test result: participant 11

Participant11		
	Phase 1	Phase 2
Recording	Incidental	Conversation
Correct/total	1/18	0/18
Correct rate	5.6%	0%
	Phase 3	Phase 4
Recording	Machinery	Alarms
Correct/total	0/18	12/19
Correct rate	0%	63.2%
Overall: 17.8%		

The primary purpose of test 2 was to answer to following three questions listed below:

-What is the overall sound classification success rate?
-What is the sound classification success rate of each group?
-How the result different when using WAV file and sound recorded within the app?

With the sound classification filter, which was gained through three optimization process in test1, the user test was conducted with participants. Full data collected through eleven tests is available in Appendix H.

As can be seen from the overall value of each test, the classification success rate is not a high as the result of test 1. The categorization result shows the classifier mostly worked for alarms, but not for other sound segments. Especially when it comes to machinery segment, there was no success case in this experiment, and the possible reasons for the classification failure will be discussed in Chapter 6.4 for further discussion.

	Overall	Incidental	Conversation	Machinery	Alarms
Sample number	769	198	198	198	176
Correct number	183	23	15	0	148
Correct rate	23.8%	11.6%	7.6%	0%	84.1%

Figure 125. Test result (test 2)

As can be seen from the chart above, a total of 769 samples were collected through user test. If the test worked ideally, the total number of collected sample should have been 792, but there are some number difference because of the time setting function of the application. During the test, test participants were asked to set up the time from now to 12 minutes later, and it leads to the result the duration of the testing to be less than 12 minutes because when they press the start button, that is already after several seconds were passed from exact 0 second. All in all, the correct rate of the sound filter in test2 was 23.8%, which shows lower rate than the result of test 1. In test 2, the accordance rate of incidental sound was 11.6%, while that of conversation rated 7.6%. Alarms rated a significant correct rate of 84.1%, while machinery sound was not recognized at all during the test.

System Accuracy	WAV file (Test 1)	App recording (Test 2)
Incidental	83.3%	11.6%
Conversation	72.2%	7.6%
Machinery	72.2%	0%
Alarms	44.4%	84.1%
Overall	68%	23.8%

Figure 126. Test result comparison (test 1, 2)

The results of test 1 and test 2 were compared to find how the external factors and test setups can affect the overall test quality. As can be seen from Figure 126, the overall sound categorization success rate shows a significant gap in test 1 result and test 2 result.

While the accordance rate when using WAV file (test1) shows 68%, the testing result with participants (test2) using an app shows 23.8%. One of the salient points is that alarm shows 44.4% of the accuracy level in test 1, which is comparatively lower accuracy level than other categories, while the alarm shows 84.1% of the accuracy in test 2 result.

All in all, the overall test result shows that using WAV file enabled better categorization result than using recordings made via test device. For further discussion regarding the test result, go to chapter 6.4.

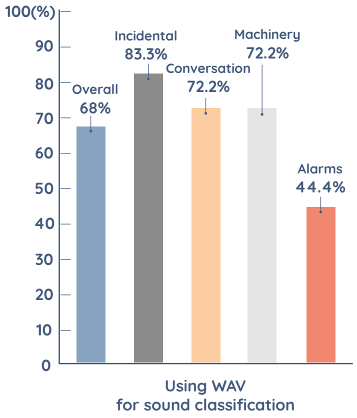


Figure 128. Test result 1: WAV file

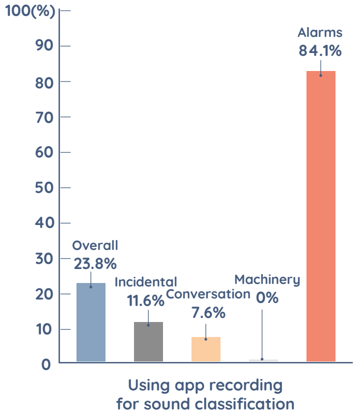


Figure 129. Test result 2: App recording

System usability test result : test3

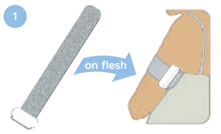
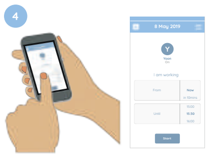
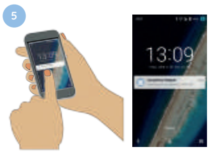
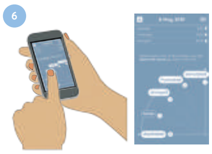
Step 1 heart rate tracker	Step 2 Sound collection part	Step 3 App: Login	Step 4 App: Time-setting	Step 5 App: Notification	Step 6 App: Emotion report
					
1.6/2.0	2.0/2.0	2.0/2.0	1.2/2.0	1.2/2.0	1.9/2.0
9.9/12					

Figure 129. Test result: Instruction & questionnaire for system usability assessment

Test 3: System usability test (with students)
-How users evaluate the overall usability of the Cacophony Mapper system (product & application)?
-How users evaluate the usability of Cacophony Mapper application?

The system usability of products and the application was evaluated through two different sections of a questionnaire. In the first section of the test, participants were asked to evaluate the product and the application usability by following six stages of tasks; putting heart rate tracker on their upper arm, clipping the sound collection device on their chest pocket, logging in to the application within given ID and password, setting up time for the report, getting access to the report page by swiping notification, and reporting emotions regarding their sound experience. Each step had 3 point scaling, starting from scale 0 which stands for difficult, 1 for okay, and 2 for easy. After going through each process, participants were asked to rate the usability of each step.

As a result, both tasks of clipping sound collection device and logging into application rated 2.0 out of 2.0. Users responded that attaching a device using a clip was easy for them. Also, all participants were used to login function since it is a standardized interface for web and app, so there was no extra learning step for this function.

Both time setting for the application and notification function rated 1.2 out of 2.0, which was the lowest among six steps. Some participants did not understand the meaning of the question saying, “please set the time slot from now to 12 minutes later from now on.”. Also, some struggled to calculate the time until 12 minutes later. This problem revealed because the interface for the application was different from an aimed interface since the testing app had every minute while the aimed version only had 15, 30, 45, 00 minutes for time setting as nurses’ timetable ends and starts only with this time frame.

Also, the notification function got comments that it is not noticeable enough. Since Huawei Mediapad T3 7.0 has been released as a tablet then a mobile, the device did not have a function to give off a buzz when there is an alert, so there was just a pop-up when there was a notification. Also, 5 participants pointed out that deciding the interval of notification would be essential because too frequent notification can be a significant stress for nurses because multiple notifications during the test already felt like an additional burden for them.

When it comes to the step for wearing a heart rate tracker, 2 participants reported that it is physically hard to wear the device with only one hand. Also, some struggled to wear a device when they were wearing a long sleeve shirt, but it does not seem to happen in the real hospital scenario as one nurse interviewed that nurses usually wear a sleeveless top underneath their work gown or some only wear their underwear. Also, sometimes I needed to repeat that participants need to wear the armband on their skin since the sensor only functions when it touches flesh area, but participants had no idea about it and often skipped the guidance in the introduction paper.

When it comes to the emotion report interface, which is a central function of this testing application, got 1.9 out of 2.0. Some participants mentioned that they did not understand how the page works at first since it looked not too familiar, but it turned out to be very easy to use this page after trying to press buttons several times. 3 participant mentioned that the screen was very intuitively structured, so they knew immediately what they needed to do. Also, one participant mentioned that it is not clear what each axis means, while one told that he almost missed the neutral button on the right bottom side of the screen since it was positioned differently than other emotions.

All in all, the first part of the usability test for the system of the product and the application marked 9.9 out of 12, which shows 82.5 in percentile.

Cacophony Mapper application usability evaluation				
Cacophony Mapper app was **manageable to learn.**	Cacophony Mapper app was **clearly structured.**	Cacophony Mapper app was **easy to use.**	Cacophony Mapper app was **straightforward to use.**	I think it will be **practical** to use Cacophony Mapper app during work in a medical environment.
4.5/5.0	**4.0/5.0**	**4.4/5.0**	**4.4/5.0**	**3.1/5.0**
20.4/25				

Figure 130. Test result: Application usability assessment

The second part of the usability test was for examining the Cacophony Mapper application usability. Five questions were asked using the pragmatic quality from Attrakdiff word pairs for the usability assessment.

The majority of participants responded that Cacophony Mapper app was manageable to learn, showing 4.4 out of 5.0 scale. Many participants said that even though there were unfamiliar interfaces in Cacophony Mapper app than other applications, it was easy to learn how to operate it since the overall interface was well structured and intuitive enough to understand. One participant mentioned that it is easy to learn as there were only a few depths in the application.

Secondly, users responded that Cacophony Mapper app was easy to use, showing 4.4 out of 5.0 scale. Most participants naturally recognized that emotion report was the primary function since it was a page that they had to stay for the longest, and they performed most of their tasks on that page. However, some mentioned that sometimes they feel differently than emotion options, or felt there was no middle ground between emotions. Also, some were curious about why buttons are distributed that way, and one missed the existence of the neutral button at first and recognized it later.

The questionnaire “Cacophony Mapper app was easy to use” rated 4.4 out of 5.0. The only negative comment was that the participant did not know how to react to connection failure of the heart rate tracker since Bluetooth sometimes did not detect the device immediately. It is mainly because the heart rate device turns itself off when participants do not click the connection button on the application for the designated time.

Clarity of the app structure was rated as 4.0 out of 5.0. One participant mentioned that start HB button and start Rec button could have been combined or at least positioned differently, so users would know the sequence that they need to follow. Since those buttons are located in parallel, test participants sometimes did not understand where to start and how to go to the next task without instruction.

The practicality of introducing the system to the medical environment was the point that overall participants gave the lowest score, which is 3.1 out of 5.0. Majority of participants found that it is not pleasant to be disturbed by alerts multiple times while they were fulfilling their daily work routine, and they reported that it was not easy to keep their concentration. Three participants pointed out that nurses may not easily find time to report in the middle of the work. Also, 5 participants mentioned that the alert was not noticeable since it just showed the pop-up screen without any buzz. One participant mentioned that nurses would feel tired of reporting their emotions if they do not find it beneficial for them, so keeping motivation for continuous reporting was considered as an essential factor as it has been expected.

Discussion

test 1: System reliability (with WAV file)

Since the sample size of the sound classification filter was limited to 200, the result showed comparatively low accuracy level to the sound which was not been used for the classification training. The sound filter was not as responsive to the sound that it comes across for the first time, so three stages of the optimization process were needed for further testing. Like several times of classifier updates were made to overcome the limitation of limited sample size, optimizations will be needed for the testing in the real hospital environment, too. As the filter was created mainly with ICU sounds collected from BBC Radio station, sounds from the real environment should be added to the classifier for further system implementation. Once the classification filter has enough sample size with the quality samples that reflect hospital sound environment better, the error rate will be significantly decreased as can be seen from classification success rate difference before and after optimization process in test 1.

Overall test setting (relevant to test 2,3)

The user test was conducted at the Industrial design department, TU Delft, with 11 students. Participants were aged between 21 to 31, and they are studying or finished studying the Industrial design department. One of the essential notes is that the test was conducted with a fairly limited number of participants with a limited scope with good educational level and good knowledge about operating new technologies. Since the test group tends to be more familiar with products with various concepts, they might have had a better understanding and high acceptance to the Cacophony Mapper concept, too. Also, since young people tend to easily learn how to operate various interfaces, it could have affected test participants' general responses to the system.

Also, since the product and application system was designed geared toward nurses who work in the ICU environment, it is essential to do the testing and get useful comments from a real hospital environment. Students' responses do not fully reflect nurses' responses because they worked on different tasks than nurses' during the test in a totally different environment. For example, students' task was a sedentary job, mostly working on their laptop, while nurses' daily obligation involves various physical activities, with different postures than a sitting posture.

Finally, the quality of the instruction paper was crucial especially for the first part of the usability test since more well-organized instruction paper could have functioned as a successful experiment tool, giving a better understanding of general system use and the purpose of the test. There could have been some added marks or arrows on the instruction graphics to help participants' understanding of their task.

test 2: System reliability (with participants)

The test result showed a significant drop in accuracy rate in test 2 compare to the result of test 1. The main reason seems to be related to external factors from the test set, such as the quality of the testing device, the size, and material of the testing room, the specification range setting of the microphone. As can be seen from test 1, the function of the classifier has shown 68% of accuracy, so the problem of the significant success rate drop should be found from external factors than the quality of the classifier itself. For example, the power peaks of the sound can differ depending on the quality of compiled WAV sound. Also, specifications of a speaker or a microphone could have been the leading cause since power peaks can be collected differently depending on various test device settings or test environments. Therefore, further calibration of the test device should be done to match the condition of recording and the original sound environment, so that more meaningful test output can be derived.

Furthermore, there were often sound interruptions during the test since test participants tried to ask something in the middle of the test, or there were some cases that participants needed further instructions. Also, participants made some noises during the test, which could have recorded together with the sound samplee, and eventually affected the classification result.

test 3: System usability

The general responses of the test participants toward the overall system usability were positive. Participants did not have significant difficulties using hardware, and they evaluated that the application structure and the interface was intuitive and easy to learn. Even though the interface did not look familiar for them, participants reported that they could pick up how to operate the app soon since the structure of the app is simple, and it does not have many layers of information in one page.

However, Huawei Mediapad T3 7.0, which is a small-sized tablet, was used as a device for the application usability test and it could have affected the test result in the way that it did not have a function to give off a buzz when sending a notification. Many test participants responded that they did not notice the notification because the screen only splashed, and the notification showed up on the screen for a few seconds and disappeared. Some mentioned they could not fully concentrate on their tasks because they felt like they needed to wait for the notification to come since there was no buzzing.

The test device could have affected the information delivery as well since it had a bigger screen size than usual mobile phones. Since there was no crammed information coming from a small screen, users' perceived usability could have been different than using a regular smartphone in the experiment.

One important note that needs to be taken into consideration is that intervals of the notifications should be well distributed when Cacophony Mapper system is introduced in the real hospital setting. Many participants reported they were not sure if the system can naturally function in the real hospital environment while nurses are performing their work routines.

Furthermore, there was a doubt about how to make nurses' responses more regular and spontaneous because one participant mentioned that she would not feel the motivation to report if there are no rewards. Since the analysis page has not been implemented, providing a personal sound experience analyses to nurses as a part of the reward was not tested. However, it is still crucial that nurses do not lose motivation to keep reporting, so the application can continuously collect the user responses toward sound stimuli.

Making a communication page among nurses about their sound experiences has been considered as one of the design interventions, but it has not been developed as a concept because developing a system for sound collection and analysis was more of a focus in this graduation project. In this regard, it will be great to include design interventions in the system in further system development, too.

Conclusion

Intensive care unit in the hospital is a hostile environment for everyone. The focus of my project was especially nurses who work for the ICU and I especially focused on their negative auditory experiences. Even though nurses are familiar with the hospital surroundings and working at the ICU on a daily basis, they can be more vulnerable to sounds than other inhabitants in the way that they are captive audiences of every sound stimuli in the ICU.

Both on an emotional level physical level, their auditory burden is excessive because of various sound sources that they are exposed to; such as machinery noise in the background, the conversation of peers and other inhabitants, alarms incessantly coming from various machines, and incidental sounds such as objects dropping sounds or door slams. They do not have much autonomy to those sounds, and there is a well-defined symptom which is called “alarm fatigue” which refers to a medical symptom that leads to less auditory ability, massive stress, less job satisfaction, and less job performance in the end as a result of constant exposure to excessive alarm noises. In fact, a research pointed out that 65% of the primary medical incident in the ICU happened by not responding to alarms appropriately.

To figure out this problem, what I focused on was not the design solution to reduce noises, but how noises can be captured and other sound sources than alarms can be defined as a source of stress too, so new system can function as a foundation to start design interventions. Therefore, what I aimed for was developing a sound tracker which can categorize the sound and detect the sound level from the surrounding, and combine the collected sound information with other indicators that show stress level of nurses. Thus, I decided to develop a mobile sensor which could be attached to nurses so it can collect more dynamic ICU information from the perspective of nurses.

Therefore, what needed to be done was developing a sound classification filter and connect it with hardware design. Also, the objective and subjective stress level were decided to be collected through heart rate tracker and emotion report of nurses, respectively. For the emotion report function, there should have been an application, and ways of improving the motivation for regular reporting was contemplated. There is a firebase server which functions as a hub to summarize all the data set, so collected information could be analyzed as a result. Because of the time limitation, the analysis pages of the app for the sound researchers were not implemented in the prototype, but screen design was introduced in the report for further development.

Through multiple rounds of the system reliability test, it has been concluded that enlarging the sample size using the sound files collected from the real hospital surroundings is needed. Because of the limitation of the sample size, the first round of the test result showed only 25% of the accuracy, but it soared up to 68% in the final round test.

The second test for the system reliability with students showed that external factors in the test set are essential, too. The quality and the setting of the test devices, such as a speaker and a microphone could immensely affect the test result since the accuracy level plummeted to 23.8% in the user test. It is strongly suspected that the power peaks produced through the microphone were affected by external factors, such as a specification of the microphone or the size and the material of the testing room. Also, there are other factors, such as test participants’ sound productions, sound play time gap, and recording time mismatch.

The usability test result showed that the Cacophony Mapper concept is not difficult to understand, and most participants could cope well with the application even though they were first-time users. The assessment of the usability of the overall system (product and application) rated 9.9 out of 12, while the assessment of the usability of the app scored 20.4 out of 25. The most reviews said the app was intuitive and easy to learn, but their concerns were mostly about how this system could be applied to the real hospital environment, which remains as a further assignment.

Even though the classification result of the second test was lower than I expected, I still believe that the classification success rate can be still improved with valid sample numbers increase, samples processing method development, and test device calibration. With an improved accuracy level of the sound classifier and contemplation over test methodology, I firmly believe that Cacophony Mapper can function as a tool to define noise fatigue in the ICU. It will not only function as a detection tool but also will combine with further design interventions, so eventually, it can contribute to forming a positive sound environment in the ICU.

Recommendations

Data visualization for the web

The original scope of this graduation was finding negative sound stimuli in the ICU and analyze them. Even though the categorization and user stress analysis has been done through this project and data visualization for the application is there, I think making an analytical visualization for the web is still needed to be done after this project since the web will be a primary form of getting access to the data than an application for sound researchers. Web visualization was excluded after the midterm meeting to focus on reporting function of the application and visual design for the mobile app. However, since there is no platform which gives a clear overview of sound data and stress detection altogether in web format, and that information is only available in fault tree analysis format, I think developing a data visualization for the web is necessary.

An embodiment of heat map

An embodiment of the heat map is still needed to be done for the application. Even though sound heat map idea has been suggested as a part of the envisioned concept in the report and there is a screen design for them, it has not been implemented in the application because of a limited graduation time frame of 20 weeks. Though the implementation for the application has been taken care of, the heat map function is one of the core functionalities in Cacophony Mapper concept.

Sound categorization filter accuracy development

Sound categorization filter was created by putting 200 sound samples in the classification model in Matlab. Even though the filter was optimized by putting quality sound samples and enlarging sample numbers, it showed its limitation in real test environment. As optimization has been mainly done in a conversation and background noise group, there is still a room for development within incidental sound group which contains 4 sub-categories of footsteps, door slam, objects clashing, and trolley dragging sound. Also, system reliability should be continuosly developed by adding more samples collected from real hospital surroundings and optimize the processing rate of sound to get more accurate classification mechanism.

Testing sound filter in a real hospital environment

Furthermore, when creating a sound filter, it is essential that the sound filter works in a real-life situation, especially in the hospital environment. Even though the sound classifier reflects the sound of the hospital environment, the environment can still affect the sound classification success rate a lot, so it important to callibrate the device in the exact test environment and optimize the classifier several times in that environment together.

Electronics optimization

Due to the time limitation of the whole project, electronics were considered as the least priority, so electronics have been substituted with existing devices, such as Fitbit, Mio, and Nolan microphone. However, if the alignment of electronics can be changed or minimized, the structure of the whole product can be changed depending on the requirements of new settings. In this regards, electronics for the project can be looked into for the further hardware optimization.

Data visualization for web

Embodiment of heatmap

Sound filter accuracy development

Testing sound filter in hospital environment

Electronics optimization

Emotion report scope review

Throughout experiments, there were comments that emotion report options does not fully reflect the emotion that they feel about the sound environment. Even though circumplex of affect model has been simplified for easy reporting, it seems necessary to mainly look at other emotion expressions for the sound environment in the ICU. In fact, neutral button was included in interface for the user test since users would feel pressured to choose negative emotion because there were only five negative emotions.

Solutions for noise fatigue

Even though the purpose of this graduation project was developing a device which can used as a tool to validate noise fatigue, and sound categorization was mainly focused in implementation process, I think there still can be added functions to system than reporting. Three devices are already used for Cacophony Mapper system, including a mobile phone, and there can be a added value to the system if it can provide a design intervention for the pleasant sound environment in the ICU, too.

Consideration for the device material

As mass-production has not been considered for Cacophony mapper concept, the device has been 3D-printed through Ultramaker Cura 2. Therefore, there was a limitation of the material use as well as its layering structure. Since the structural limitation led to a delicate clipping part, other material or prototyping method can be considered for further development of the hardware.

Personal reflection

Quality of the work

In general, I am satisfied with the work that I have done within my graduation. Mainly, I am thrilled that I could organize various experiments myself to figure out my main focus "noise fatigue" from the perspective of a design researcher. Especially, heart rate placement test was one exciting thing in the way that I came up with several placement scenarios and look into the validity of my ideas myself. Also, sound classification filter creation was one significant progress that I made through my graduation. I am really proud that I could look into various sound properties and cluster them and used those features for the further creation of the sound classification filter. Now, I am confident enough to create a new classification system and optimize the classifier depending on my needs. I am satisfied that I made a practical system with pieces of knowledge that I gained through incessant scientific research.

Planning

I am also glad that I could finish my graduation project within designated 20 weeks. To some extent, the limited time frame was a shame because I had to narrow down the scope of the project and implementation area. Also, there were some topics which have not been covered while I think it is still important to look at, so I put them as recommendations for further development. However, I think I learned how I could organize my own schedule and resources, and I think this experience will help me a lot to practically organize my work as a design researcher in the future.

Personal ambition

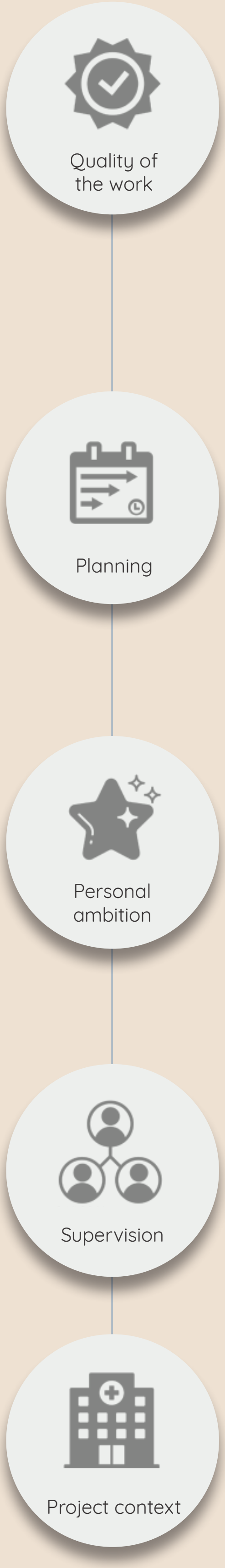
While I was working on the research project in the previous semester for Critical Alarm Lab, I wanted to build up a deeper understanding of the sound analysis method in the medical environment. Through bountiful literature review related to the sound and experiments that I organized and conducted myself, I think I have achieved what I was aiming for within this graduation project. Also, I believe that the gained knowledge will be a great resource that I will be able to use for my further research as a design research engineer. Also, I could learn how to implement design methodologies into a practical design, and I think it is significant progress that I made as a design researcher.

Supervision

I am pleased that my supervision team has supported me in various directions and I love the fact that the area that I could get help from my chair and my mentor was different based on their specialties. It was such a great experience that I could get very constructive feedbacks in every stage of my design development, so I could actually apply that advice to my progress. There was always food for through after meetings with my supervision team, and I could overcome numerous difficulties throughout those consultations.

Project context

As this project had an Intensive care unit of the hospital as its context, there has always been confidentiality and privacy issues that were clashing with my design decisions. The boundary and limitations coming from the context "hospital" made a little detour in many decisions, but it is still fruitful in the way this process taught me to think more about priorities, limitations, and project scope.



References

Alain, C. (2007). Breaking the wave: Effects of attention and learning on concurrent sound perception. Hearing Research, 229(1-2), 225-236. doi: 10.1016/j.heares.2007.01.011

Anttonen, Jenni & Surakka, Veikko. (2005). Emotions and heart rate while sitting on a chair. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. 491-499. 10.1145/1054972.1055040.

Boeijen, A., & Daalhuizen, J. (2017). Delft design guide (4th ed.). BIS.

Bracewell, R. (1989). The Fourier Transform. Scientific American, 260(6), 86-95. doi: 10.1038/scientificamerican0689-86

Busch-Vishniac, Ilene & E West, James & Barnhill, Colin & Hunter, Tyrone & Orellana, Douglas & Chivukula, Ram. (2006). Noise levels in Johns Hopkins Hospital. The Journal of the Acoustical Society of America. 118. 3629-45. 10.1121/1.2118327.

Caffier, P., Erdmann, U., & Ullsperger, P. (2003). Experimental evaluation of eye-blink parameters as a drowsiness measure. European Journal Of Applied Physiology, 89(3), 319-325. doi: 10.1007/s00421-003-0807-5

Cho, O., Kim, H., Lee, Y., & Cho, I. (2016). Clinical Alarms in Intensive Care Units: Perceived Obstacles of Alarm Management and Alarm Fatigue in Nurses. Healthcare Informatics Research, 22(1), 46. doi: 10.4258/hir.2016.22.1.46

Darbyshire, J., & Young, J. (2013). An investigation of sound levels on intensive care units with reference to the WHO guidelines. Critical Care, 17(5), R187. doi: 10.1186/cc12870

Drew, B., Harris, P., Zègre-Hemsey, J., Mammone, T., Schindler, D., & Salas-Boni, R. et al. (2014). Insights into the Problem of Alarm Fatigue with Physiologic Monitor Devices: A Comprehensive Observational Study of Consecutive Intensive Care Unit Patients. Plos ONE, 9(10), e110274. doi: 10.1371/journal.pone.0110274

Edworthy, J., Özcan, E., van Egmond, R., & Jansen, R. Alarm Philosophy for ESOC Mission Control Rooms in Darmstadt. ESA Project, (AO 1-7223/12/F/MOS)

Farrell, M.E., Joseph, D.H. and Schwartz-Barcott, D., 2005, January. Visiting hours in the ICU: finding the balance among patient, visitor and staff needs. In Nursing forum. Blackwell Publishing, Ltd.

Handbook of research methods in personality psychology. (2007). Choice Reviews Online, 45(04), 45-2315-45-2315. doi: 10.5860/choice.45-2315

Hosseini, S., Khalilzadeh, M., & Changiz, S. (2010). EMOTIONAL STRESS RECOGNITION SYSTEM FOR AFFECTIVE COMPUTING BASED ON BIO-SIGNALS. Journal Of Biological Systems, 18(spec01), 101-114. doi: 10.1142/s0218339010003640

IZAWA, S., SHIROTSUKI, K., SUGAYA, N., OGAWA, N., SUZUKI, K., & NOMURA, S. (2007). The Application of Saliva to an Assessment of Stress: Procedures for Collecting and Analyzing Saliva and Characteristics of Salivary Substances. Japanese Journal Of Complementary And Alternative Medicine, 4(3), 91-101. doi: 10.1625/jcam.4.91

J. A. C. Patterson, D. C. McIlwraith and G. Yang, “A Flexible, Low Noise Reflective PPG Sensor Platform for Ear-Worn Heart Rate Monitoring,” 2009 Sixth International Workshop on Wearable and Implantable Body Sensor Networks, Berkeley, CA, 2009, pp. 286-291.
doi: 10.1109/BSN.2009.16

Khademi, G., Roudi, M., Shah Farhat, A., & Shahabian, M. (2011). Noise pollution in intensive care units and emergency wards. Iranian journal of otorhinolaryngology, 23(65), 141-8.

Konkani, A., Oakley, B., & Penprase, B. (2014). Reducing Hospital ICU Noise: A Behavior-Based Approach. Journal Of Healthcare Engineering, 5(2), 229-246. doi: 10.1260/2040-2295.5.2.229

Kyle, U. (2004). Bioelectrical impedance analysis?part I: review of principles and methods. Clinical Nutrition, 23(5), 1226-1243. doi: 10.1016/j.clnu.2004.06.004

Lawson, N., Thompson, K., Saunders, G., Saiz, J., Richardson, J., & Brown, D. et al. (2019). Sound Intensity and Noise Evaluation in a Critical Care Unit.

Lin, J., Keogh, E., Wei, L., & Lonardi, S. (2007). Experiencing SAX: a novel symbolic representation of time series. Data Mining And Knowledge Discovery, 15(2), 107-144. doi: 10.1007/s10618-007-0064-z

McKee, H. (2006). Sound matters: Notes toward the analysis and design of sound in multimodal webtexts. Computers And Composition, 23(3), 335-354. doi: 10.1016/j.compcom.2006.06.003

Monk, T. (1989). A visual analogue scale technique to measure global vigor and affect. Psychiatry Research, 27(1), 89-99. doi: 10.1016/0165-1781(89)90013-9

Micklewright, D., St Clair Gibson, A., Gladwell, V., Al Salman, A. (2017) Development and Validity of the Rating-of-Fatigue Scale. Sports Med 47: 2375.

Otenio MH, Cremer E, and Claro EMT. Noise level in a 222 bed hospital in the 18th health region-pr. Revista Brasileira de Otorrinolaringologia. 2007

Özcan, E., van Egmond, R.,Gentner, A., & Favart, C. (in press). Incorporating brand identity in the design of auditory displays: The case of Toyota Motors Europe. in Foundations in Sound Design (ed. Michael Filimowicz). Routledge.

Posner, J., Russell, J. A., & Peterson, B. S. (2005). The circumplex model of affect: an integrative approach to affective neuroscience, cognitive development, and psychopathology. Development and psychopathology, 17(3), 715–734. doi:10.1017/S0954579405050340

Salous, M., Alkhawaldeh, J., Kewan, S., Aburashideh, H., Hani, D., & Alzayyat, A. (2017). Nurses' Attitudes Related To Alarm Fatigue in Critical Care Units: A Systematic Review. IOSR Journal Of Nursing And Health Science, 06(02), 62-66. doi: 10.9790/1959-0602076266

Selvaraj, N., Jaryal, A., Santhosh, J., Deepak, K., & Anand, S. (2008). Assessment of heart rate variability derived from finger-tip photoplethysmography as compared to electrocardiography. Journal Of Medical Engineering & Technology, 32(6), 479-484. doi: 10.1080/03091900701781317

Stafford, A., Haverland, A., & Bridges, E. (2014). Noise in the ICU. AJN, American Journal Of Nursing, 114(5), 57-63. doi: 10.1097/01.naj.0000446780.99522.90

Stone, A., & Shiffman, S. (2002). Capturing momentary, self-report data: A proposal for reporting guidelines. Annals Of Behavioral Medicine, 24(3), 236-243. doi: 10.1207/s15324796abm2403_09

Tania F. Gallo, Stuart J. Cormack, Tim J. Gabbett & Christian H. Lorenzen
(2016) Pre-training perceived wellness impacts training output in Australian football players,Journal of Sports Sciences, 34:15, 1445-1451, DOI:10.1080/02640414.2015.1119295

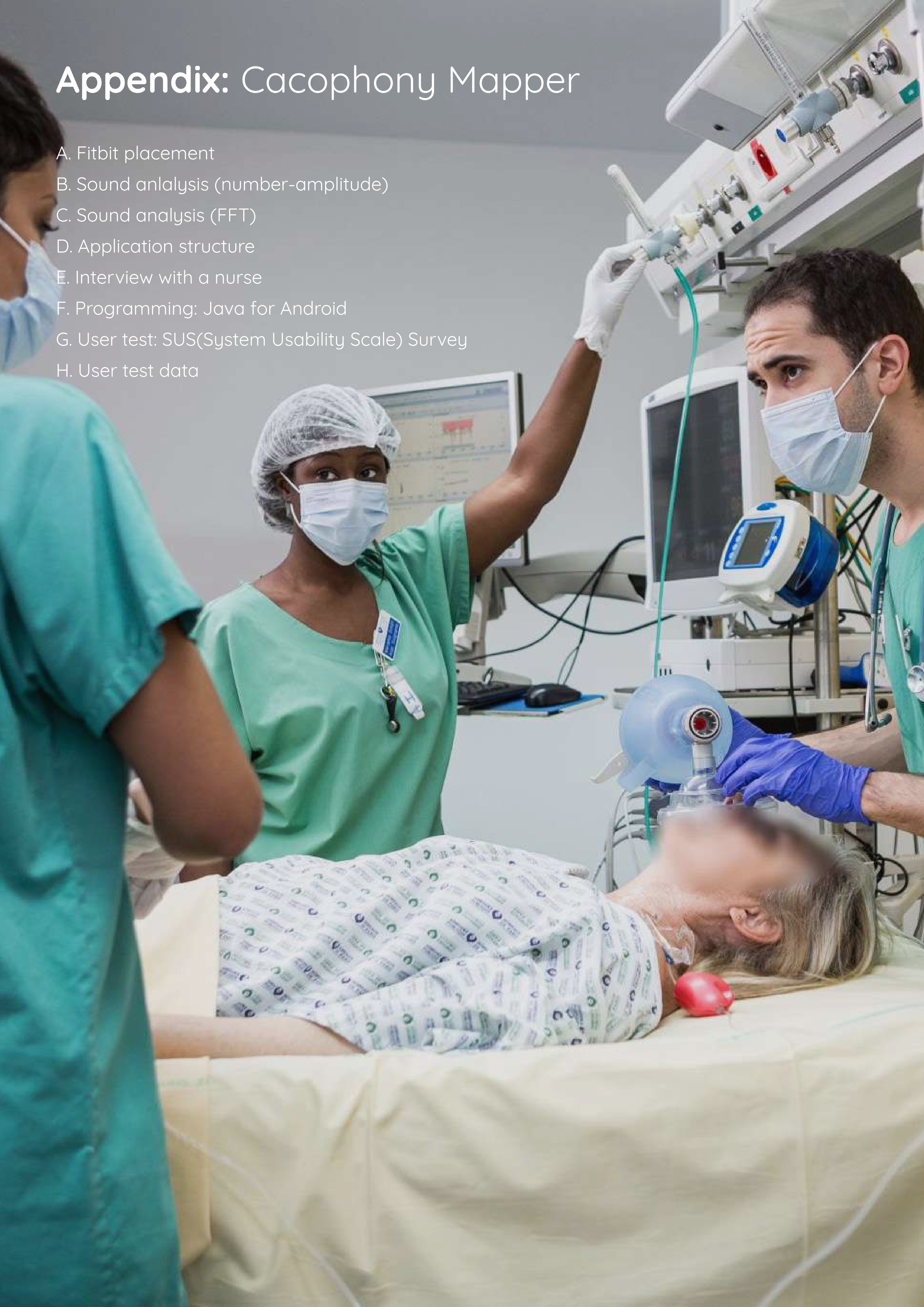
Thayer, J., Åhs, F., Fredrikson, M., Sollers, J., & Wager, T. (2012). A meta-analysis of heart rate variability and neuroimaging studies: Implications for heart rate variability as a marker of stress and health. Neuroscience & Biobehavioral Reviews, 36(2), 747-756. doi: 10.1016/j.neubiorev.2011.11.009

http://attrakdiff.de/index-en.html

https://www.premotool.com/

https://www.wareable.com/wearable-tech/where-is-the-best-place-to-track-heart-rate-877

Appendix: Cacophony Mapper



Fitbit placement: wrist-upper outer arm (14/02/2019)

time	HR wrist	HR u.o.arm	difference
11:50-11:55	91	90	1
11:55-12:00	72	74	2
12:00-12:05	82	80	2
12:05-12:10	82	82	0
12:10-12:15	79	87	8
12:15-12:20	113	99	14
12:20-12:25	82	85	3
12:25-12:30	76	79	3
12:30-12:35	76	74	2
12:35-12:40	71	69	2
12:40-12:45	69	71	2
12:45-12:50	70	67	3
12:50-12:55	70	69	1
12:55-13:00	71	66	5
13:00-13:05	73	70	3
13:05-13:10	71	79	8
13:10-13:15	71	72	1
13:15-13:20	72	65	7
13:20-13:25	71	70	1
13:25-13:30	72	72	0
13:30-13:35	93	89	4
13:35-13:40	88	85	3
13:40-13:45	69	70	1
13:45-13:50	70	70	0
13:50-13:55	69	69	0
13:55-14:00	69	69	0
14:00-14:05	68	68	0
14:05-14:10	69	69	0
14:10-14:15	67	67	0
14:15-14:20	67	68	1
14:20-14:25	67	67	0
14:25-14:30	83	81	2
14:30-14:35	118	101	17
14:35-14:40	106	91	15
14:40-14:45	71	66	5
14:45-14:50	64	64	0
14:50-14:55	65	66	1
14:55-15:00	67	72	5
15:00-15:05	76	75	1
15:05-15:10	74	70	4
15:10-15:15	69	68	1
15:15-15:20	74	74	0
15:20-15:25	83	84	1
15:25-15:30	75	74	1
15:30-15:35	74	73	1
15:35-15:40	78	79	1
15:40-15:45	76	79	3
15:45-15:50	76	75	1
15:50-15:55	80	80	0
15:55-16:00	73	72	1
16:00-16:05	73	71	2
16:05-16:10	69	69	0
16:10-16:15	68	69	1
16:15-16:20	67	67	0
16:20-16:25	68	68	0
16:25-16:30	89	88	1
16:30-16:35	90	88	2
16:35-16:40	78	81	3
16:40-16:45	82	80	2
16:45-16:50	80	76	4
16:50-16:55	85	111	26
16:55-17:00	102	117	15
17:00-17:05	106	122	16
17:05-17:10	89	89	0
17:10-17:15	78	78	0
17:15-17:20	78	79	1
17:20-17:25	78	78	0
17:25-17:30	77	80	3
17:30-17:35	81	83	2
17:35-17:40	81	79	2
17:40-17:45	80	80	0
17:45-17:50	80	80	0
17:50-17:55	85	85	0
17:55-18:00	81	82	1
18:00-18:05	90	91	1

Sum difference
1

Fitbit placement: wrist-neck (15/02/2019)

time	HR wrist	HR u.o.arm	difference
7:40-7:45	80	73	7
7:45-7:50	87	78	9
7:50-7:55	76	75	1
7:55-8:00	75	74	1
8:00-8:05	79	85	6
8:05-8:10	84	101	17
8:10-8:15	79	102	23
8:15-8:20	80	88	8
8:20-8:25	81	81	0
8:25-8:30	101	119	18
8:30-8:35	77	76	1
8:35-8:40	69	70	1
8:40-8:45	70	68	2
8:45-8:50	68	68	0
8:50-8:55	68	68	0
8:55-9:00	69	69	0
9:00-9:05	73	71	2
9:05-9:10	79	72	7
9:10-9:15	84	91	7
9:15-9:20	75	67	8
9:20-9:25	73	72	1
9:25-9:30	68	68	0
9:30-9:35	71	70	1
9:35-9:40	72	69	3
9:40-9:45	74	72	2
9:45-9:50	74	111	37
9:50-9:55	69	71	2
9:55-10:00	68	70	2
10:00-10:05	70	68	2
10:05-10:10	69	69	0
10:10-10:15	73	BLANK	#VALUE!
10:15-10:20	69	71	2
10:20-10:25	67	73	6
10:25-10:30	67	75	8
10:30-10:35	72	109	37
10:35-10:40	72	120	48
10:40-10:45	67	106	39
10:45-10:50	67	75	8
10:50-10:55	65	66	1
10:55-11:00	66	69	3
11:00-11:05	69	71	2
11:05-11:10	65	64	1
11:10-11:15	70	79	9
11:15-11:20	63	71	8
11:20-11:25	66	75	9
11:25-11:30	63	79	16
11:30-11:35	65	67	2
11:35-11:40	71	132	61
11:40-11:45	62	85	23
11:45-11:50	64	99	35
11:50-11:55	64	100	36
11:55-12:00	62	62	0
12:00-12:05	62	76	14
12:05-12:10	69	75	6
12:10-12:15	93	125	32
12:15-12:20	86	120	34
12:20-12:25	75	120	45
12:25-12:30	73	97	24
12:30-12:35	75	102	27
12:35-12:40	73	103	30
12:40-12:45	74	95	21
12:45-12:50	75	99	24
12:50-12:55	80	75	5
12:55-13:00	73	75	2
13:00-13:05	71	86	15
13:05-13:10	71	101	30
13:10-13:15	70	111	41
13:15-13:20	82	118	36
13:20-13:25	101	125	24
13:25-13:30	76	92	16
13:30-13:35	73	94	21
13:35-13:40	75	102	27
13:40-13:45	73	72	1
13:45-13:50	73	71	2
13:50-13:55	79	66	13
13:55-14:00	79	73	6
14:00-14:05	81	81	0
14:05-14:10	77	90	13
14:10-14:15	79	81	2
14:15-14:20	83	86	3
14:20-14:25	86	114	28
14:25-14:30	74	BLANK	#VALUE!
14:30-14:35	75	73	2

Sum difference
7

Fitbit placement: wrist-ear (16/02/2019)

time	HR wrist	HR u.o.arm	difference
15:55-16:00	88	91	3
16:00-16:05	83	84	1
16:05-16:10	75	77	2
16:10-16:15	71	75	4
16:15-16:20	72	73	1
16:20-16:25	75	78	3
16:25-16:30	81	86	5
16:30-16:35	76	75	1
16:35-16:40	75	75	0
16:40-16:45	78	78	0
16:45-16:50	79	77	2
16:50-16:55	80	78	2
16:55-17:00	78	77	1
17:00-17:05	87	93	6
17:05-17:10	76	76	0
17:10-17:15	74	74	0
17:15-17:20	73	72	1
17:20-17:25	75	72	3
17:25-17:30	73	71	2
17:30-17:35	72	70	2
17:35-17:40	74	70	4
17:40-17:45	76	74	2
17:45-17:50	75	74	1
17:50-17:55	74	72	2
17:55-18:00	73	72	1
18:00-18:05	75	74	1
18:5-18:10	72	74	2
18:10-18:15	77	77	0
18:15-18:20	84	81	3
18:20-18:25	76	82	6
18:25-18:30	71	73	2
18:30-18:35	74	74	0
18:35-18:40	77	76	1
18:40-18:45	80	77	3
18:45-18:50	73	75	2
18:50-18:55	75	75	0
18:55-19:00	79	78	1
19:00-19:05	79	77	2
19:05-19:10	80	77	3
19:10-19:15	85	87	2
19:15-19:20	87	81	6
19:20-19:25	82	81	1
19:25-19:30	80	78	2
19:30-19:35	79	79	0
19:35-19:40	82	78	4
19:40-19:45	81	78	3
19:45-19:50	87	83	4
19:50-19:55	86	84	2
19:55-20:00	84	82	2
20:00-20:05	83	81	2
20:05-20:10	81	83	2
20:10-20:15	82	81	1
20:15-20:20	84	82	2
20:20-20:25	83	83	0
20:25-20:30	84	82	2
20:30-20:35	83	83	0
20:35-20:40	83	81	2
20:40-20:45	79	79	0
20:45-20:50	82	83	1
20:50-20:55	86	83	3
20:55-21:00	80	79	1
21:00-21:05	80	76	4
21:05-21:10	78	78	0
21:10-21:15	77	77	0
21:15-21:20	78	77	1
21:20-21:25	78	77	1
21:25-21:30	77	77	0
21:30-21:35	76	76	0
21:35-21:40	81	81	0
21:40-21:45	79	78	1
21:45-21:50	82	81	1
21:50-21:55	85	84	1
21:55-22:00	82	77	5
22:00-22:05	82	85	3

Sum difference
3

Fitbit placement: wrist-chest pocket (17/02/2019)

time	HR wrist	HR u.o.arm	difference
13:15-13:20	82	118	36
13:20-13:25	83	120	37
13:25-13:30	84	121	37
13:30-13:35	85	118	33
13:35-13:40	94	116	22
13:40-13:45	101	127	26
13:45-13:50	105	126	21
13:50-13:55	77	126	49
13:55-14:00	72	77	5
14:00-14:05	77	80	3
14:05-14:10	75	102	27
14:10-14:15	81	88	7
14:15-14:20	79	78	1
14:20-14:25	80	83	3
14:25-14:30	81	98	17
14:30-14:35	87	120	33
14:35-14:40	77	68	9
14:40-14:45	79	104	25
14:45-14:50	76	BLANK	#VALUE!
14:50-14:55	76	73	3
14:55-15:00	76	70	6
15:00-15:05	77	BLANK	#VALUE!
15:05-15:10	77	70	7
15:10-15:15	80	115	35
15:15-15:20	76	98	22
15:20-15:25	74	75	1
15:25-15:30	76	72	4
15:30-15:35	77	68	9
15:35-15:40	77	71	6
15:40-15:45	80	89	9
15:45-15:50	80	93	13
15:50-15:55	78	69	9
15:55-16:00	82	57	25
16:00-16:05	78	59	19
16:05-16:10	80	78	2
16:10-16:15	76	72	4
16:15-16:20	74	69	5
16:20-16:25	78	94	16
16:25-16:30	82	106	24
16:30-16:35	84	114	30
16:35-16:40	75	107	32
16:45-16:50	71	BLANK	#VALUE!
16:50-16:55	72	75	3
16:55-17:00	72	70	2
17:00-17:05	76	72	4
17:05-17:10	77	83	6
17:10-17:15	84	119	35
17:15-17:20	89	120	31
17:20-17:25	74	BLANK	#VALUE!
17:25-17:30	74	99	25
17:30-17:35	75	64	11
17:35-17:40	75	62	13
17:40-17:45	76	74	2
17:45-17:50	74	71	3
17:50-17:55	74	BLANK	#VALUE!
17:55-18:00	74	79	5
18:00-18:05	73	72	1
18:5-18:10	82	106	24
18:10-18:15	92	126	34
18:15-18:20	102	127	25
18:20-18:25	92	126	34
18:25-18:30	121	129	8
18:30-18:35	109	126	17
18:35-18:40	82	121	39

Sum difference
36

Fitbit placement: wrist-pants pocket (21/02/2019)

time	HR wrist	HR u.o.arm	difference
12:05-12:10	114	103	11
12:10-12:15	98	104	6
12:15-12:20	83	110	27
12:20-12:25	99	124	25
12:25-12:30	95	125	30
12:30-12:35	79	126	47
12:35-12:40	74	BLANK	#VALUE!
12:40-12:45	77	77	0
12:45-12:50	87	85	2
12:50-12:55	83	65	18
12:55-13:00	83	66	17
13:00-13:05	82	70	12
13:05-13:10	85	67	18
13:10-13:15	85	63	22
13:15-13:20	85	74	11
13:20-13:25	94	96	2
13:25-13:30	81	105	24
13:30-13:35	77	78	1
13:35-13:40	80	75	5
13:40-13:45	84	106	22
13:45-13:50	80	111	31
13:50-13:55	81	100	19
13:55-14:00	80	BLANK	#VALUE!
14:00-14:05	79	70	9
14:05-14:10	79	66	13
14:10-14:15	79	65	14
14:15-14:20	80	65	15
14:20-14:25	79	65	14
14:25-14:30	81	65	16
14:30-14:35	81	64	17
14:35-14:40	82	57	25
14:40-14:45	80	83	3
14:45-14:50	79	72	7
14:50-14:55	80	54	26
14:55-15:00	78	57	21
15:00-15:05	77	61	16
15:05-15:10	90	118	28
15:10-15:15	115	126	11
15:15-15:20	91	110	19
15:20-15:25	92	109	17
15:25-15:30	83	101	18
15:30-15:35	81	BLANK	#VALUE!
15:35-15:40	80	86	6
15:40-15:45	88	85	3
15:45-15:50	80	66	14
15:50-15:55	83	65	18
15:55-16:00	80	69	11
16:00-16:05	81	69	12
16:05-16:10	86	67	19
16:10-16:15	84	71	13
16:15-16:20	85	68	17
16:20-16:25	84	70	14
16:25-16:30	84	68	16
16:30-16:35	84	59	25
16:35-16:40	89	61	28
16:45-16:50	86	107	21
16:50-16:55	81	100	19
16:55-17:00	83	BLANK	#VALUE!
17:00-17:05	85	95	10
17:05-17:10	86	93	7
17:10-17:15	82	93	11
17:15-17:20	83	91	8
17:20-17:25	81	86	5
17:25-17:30	81	83	2

Sum difference
11

Fitbit placement: wrist-ankle (22/02/2019)

time	HR wrist	HR u.o.arm	difference
11:50-11:55	103	106	3
11:55-12:00	87	93	6
12:00-12:05	92	96	4
12:05-12:10	113	117	4
12:10-12:15	98	113	15
12:15-12:20	76	77	1
12:20-12:25	74	74	0
12:25-12:30	75	75	0
12:30-12:35	73	74	1
12:35-12:40	76	77	1
12:40-12:45	74	88	14
12:45-12:50	84	83	1
12:50-12:55	84	84	0
12:55-13:00	82	83	1
13:00-13:05	82	81	1
13:05-13:10	79	80	1
13:10-13:15	79	80	1
13:15-13:20	78	78	0
13:20-13:25	76	75	1
13:25-13:30	75	76	1
13:30-13:35	82	78	4
13:35-13:40	81	74	7
13:40-13:45	79	82	3
13:45-13:50	80	81	1
13:50-13:55	78	78	0
13:55-14:00	80	82	2
14:00-14:05	80	82	2
14:05-14:10	78	81	3
14:10-14:15	80	85	5
14:15-14:20	77	87	10
14:20-14:25	82	84	2
14:25-14:30	79	81	2
14:30-14:35	80	81	1
14:35-14:40	78	80	2
14:40-14:45	74	77	3
14:45-14:50	76	75	1
14:50-14:55	85	82	3
14:55-15:00	112	123	11
15:00-15:05	100	104	4
15:05-15:10	100	123	23
15:10-15:15	107	117	10
15:15-15:20	102	122	20
15:20-15:25	101	120	19
15:25-15:30	113	121	8
15:30-15:35	94	97	3
15:35-15:40	82	83	1
15:40-15:45	79	81	2
15:45-15:50	82	83	1
15:50-15:55	79	79	0
15:55-16:00	79	80	1
16:00-16:05	88	88	0
16:05-16:10	89	89	0
16:10-16:15	83	83	0
16:15-16:20	83	81	2
16:20-16:25	87	80	7
16:25-16:30	76	78	2
16:30-16:35	77	77	0
16:35-16:40	78	79	1
16:45-16:50	83	77	6
16:50-16:55	79	84	5
16:55-17:00	75	75	0
17:00-17:05	76	78	2
17:05-17:10	76	77	1
17:10-17:15	76	80	4
17:15-17:20	81	85	4
17:20-17:25	81	82	1
17:25-17:30	81	81	0
17:30-17:35	83	83	0
17:35-17:40	79	79	0
17:40-17:45	77	77	0
17:45-17:50	76	77	1
17:50-17:55	75	75	0
17:55-18:00	76	77	1
18:00-18:05	75	79	4
18:5-18:10	77	78	1
18:10-18:15	77	77	0
18:15-18:20	77	76	1
18:20-18:25	78	77	1
18:25-18:30	79	80	1
18:30-18:35	72	74	2
18:35-18:40	75	74	1
18:40-18:45	74	74	0
18:45-18:50	74	76	2
18:50-18:55	75	76	1
18:55-19:00	73	72	1
19:00-19:05	74	74	0
19:05-19:10	74	75	1
19:10-19:15	74	77	3
19:15-19:20	76	77	1
19:20-19:25	77	78	1
19:25-19:30	74	78	4
19:30-19:35	73	74	1
19:35-19:40	73	76	3
19:40-19:45	73	75	2
19:45-19:50	74	77	3
19:50-19:55	73	75	2
19:55-20:00	74	74	0
20:00-20:05	76	75	1
20:05-20:10	76	76	0
20:10-20:15	93	107	14
20:15-20:20	98	121	23
20:20-20:25	95	126	31
20:25-20:30	109	127	18

Sum difference
3

Fitbit placement: wrist-chest (flesh) (23/02/2019)

time	HR wrist	HR u.o.arm	difference
14:40-14:45	100	109	9
14:45-14:50	90	89	1
14:50-14:55	126	111	15
14:55-15:00	114	107	7
15:00-15:05	119	110	9
15:05-15:10	98	97	1
15:10-15:15	77	75	2
15:15-15:20	74	74	0
15:20-15:25	84	81	3
15:25-15:30	81	81	0
15:30-15:35	79	80	1
15:35-15:40	88	84	4
15:40-15:45	73	70	3
15:45-15:50	71	71	0
15:50-15:55	71	71	0
15:55-16:00	70	70	0
16:00-16:05	70	72	2
16:05-16:10	116	106	10
16:10-16:15	131	116	15
16:15-16:20	125	120	5
16:20-16:25	129	108	21
16:25-16:30	115	103	12
16:30-16:35	82	80	2
16:35-16:40	84	82	2
16:45-16:50	73	74	1
16:50-16:55	77	72	5
16:55-17:00	79	80	1
17:00-17:05	82	83	1
17:05-17:10	81	85	4
17:10-17:15	79	80	1
17:15-17:20	86	86	0
17:20-17:25	83	83	0
17:25-17:30	90	88	2
17:30-17:35	89	88	1
17:35-17:40	86	86	0
17:40-17:45	87	87	0
17:45-17:50	98	93	5
17:50-17:55	103	100	3
17:55-18:00	100	101	1
18:00-18:05	84	84	0
18:5-18:10	84	81	3
18:10-18:15	78	76	2
18:15-18:20	79	78	1
18:20-18:25	90	97	7
18:25-18:30	102	107	5
18:30-18:35	110	109	1
18:35-18:40	106	97	9
18:40-18:45	102	98	4
18:45-18:50	104	92	12
18:50-18:55	118	104	14
18:55-19:00	97	101	4
19:00-19:05	121	105	16
19:05-19:10	108	99	9
19:10-19:15	104	91	13
19:15-19:20	82	87	5
19:20-19:25	81	89	8
19:25-19:30	93	86	7
19:30-19:35	109	97	12
19:35-19:40	130	124	6
19:40-19:45	118	112	6
19:45-19:50	77	76	1
19:50-19:55	76	71	5
19:55-20:00	80	74	6
20:00-20:05	76	74	2
20:05-20:10	75	73	2
20:10-20:15	76	76	0
20:15-20:20	80	79	1
20:20-20:25	77	75	2
20:25-20:30	80	82	2
20:30-20:35	79	87	8
20:35-20:40	81	79	2
20:40-20:45	83	82	1
20:45-20:50	81	83	2
20:50-20:55	75	76	1
20:55-21:00	79	79	0
21:00-21:05	76	74	2
21:05-21:10	75	79	4
21:10-21:15	79	81	2
21:15-21:20	82	81	1
21:20-21:25	81	84	3
21:25-21:30	116	103	13
21:30-21:35	81	81	0
21:35-21:40	75	77	2
21:40-21:45	73	75	2
21:45-21:50	75	76	1
21:50-21:55	76	78	2
21:55-22:00	75	75	0
22:00-22:05	79	80	1
22:05-22:10	125	117	8
22:10-22:15	126	115	11
22:15-22:20	105	99	6
22:20-22:25	87	91	4

Sum difference
9

Fitbit placement: wrist-wrist(palm side) (28/02/2019)

time	HR wrist	HR u.o.arm	difference
8:50-8:55	83	80	3
8:55-9:00	91	92	1
9:00-9:05	88	97	9
9:05-9:10	101	122	21
9:10-9:15	79	82	3
9:15-9:20	71	72	1
9:20-9:25	70	71	1
9:25-9:30	69	73	4
9:30-9:35	77	74	3
9:35-9:40	78	81	3
9:40-9:45	78	79	1
9:45-9:50	88	92	4
9:50-9:55	84	90	6
9:55-10:00	82	82	0
10:00-10:05	83	82	1
10:05-10:10	83	84	1
10:10-10:15	82	78	4
10:15-10:20	85	81	4
10:20-10:25	87	84	3
10:25-10:30	87	86	1
10:30-10:35	86	85	1
10:35-10:40	86	82	4
10:40-10:45	84	84	0
10:45-10:50	85	84	1
10:50-10:55	85	86	1
10:55-11:00	86	84	2
11:00-11:05	82	80	2
11:05-11:10	87	85	2
11:10-11:15	87	93	6
11:15-11:20	84	84	0
11:20-11:25	84	84	0
11:25-11:30	81	82	1
11:30-11:35	81	82	1
11:35-11:40	81	82	1
11:40-11:45	81	81	0
11:45-11:50	82	80	2
11:50-11:55	79	79	0
11:55-12:00	78	79	1
12:00-12:05	78	76	2
12:05-12:10	79	78	1
12:10-12:15	88	90	2
12:15-12:20	85	83	2
12:20-12:25	76	77	1
12:25-12:30	77	76	1
12:30-12:35	79	78	1
12:35-12:40	78	82	4
12:40-12:45	92	93	1
12:45-12:50	82	89	7
12:50-12:55	82	82	0
12:55-13:00	79	78	1
13:00-13:05	79	79	0
13:05-13:10	78	77	1
13:10-13:15	80	81	1
13:15-13:20	82	81	1
13:20-13:25	81	80	1
13:25-13:30	82	81	1
13:30-13:35	80	79	1
13:35-13:40	78	79	1
13:40-13:45	73	81	8
13:45-13:50	80	80	0
13:50-13:55	79	75	4
13:55-14:00	81	81	0
14:00-14:05	82	82	0
14:05-14:10	85	81	4
14:10-14:15	79	82	3
14:15-14:20	80	84	4
14:20-14:25	87	88	1
14:25-14:30	83	84	1
14:30-14:35	84	81	3
14:35-14:40	76	77	1

Sum difference

2.25

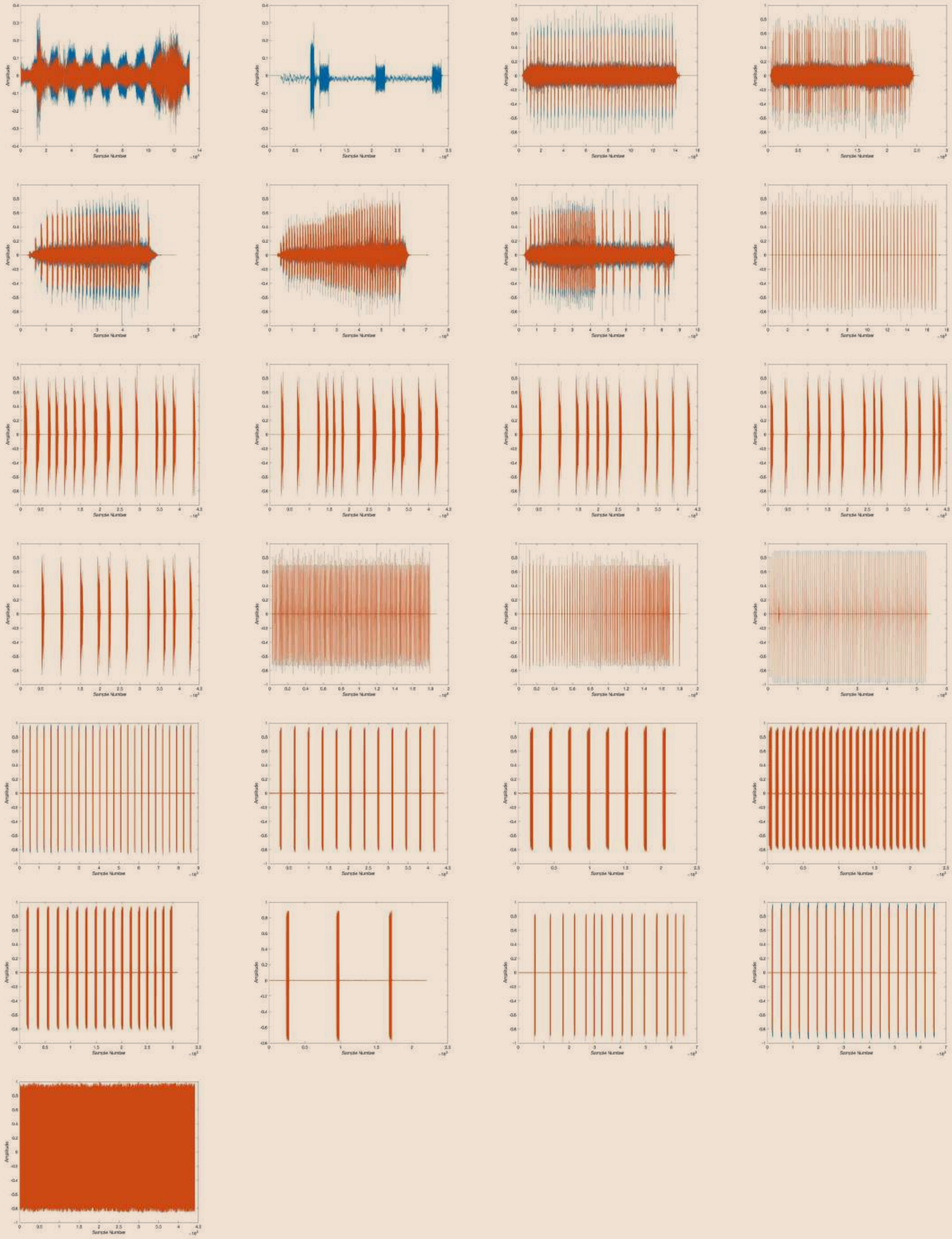
Fitbit placement: wrist-upper inner arm (29/02/2019)

time	HR wrist	HR u.o.arm	difference
11:10-11:15	91	87	4
11:15-11:20	92	88	4
11:20-11:25	105	102	3
11:25-11:30	91	82	9
11:30-11:35	72	72	0
11:35-11:40	72	72	0
11:40-11:45	77	73	4
11:45-11:50	83	80	3
11:50-11:55	82	81	1
11:55-12:00	82	82	0
12:00-12:05	79	81	2
12:05-12:10	79	80	1
12:10-12:15	80	81	1
12:15-12:20	79	79	0
12:20-12:25	79	80	1
12:25-12:30	78	82	4
12:30-12:35	76	79	3
12:35-12:40	80	81	1
12:40-12:45	77	78	1
12:45-12:50	83	80	3
12:50-12:55	72	71	1
12:55-13:00	77	76	1
13:00-13:05	76	77	1
13:05-13:10	73	76	3
13:10-13:15	78	78	0
13:15-13:20	74	73	1
13:20-13:25	76	77	1
13:25-13:30	77	77	0
13:30-13:35	88	86	2
13:35-13:40	75	74	1
13:40-13:45	78	77	1
13:45-13:50	78	78	0
13:50-13:55	77	78	1
13:55-14:00	77	77	0
14:00-14:05	78	78	0
14:05-14:10	75	78	3
14:10-14:15	78	78	0
14:15-14:20	79	80	1
14:20-14:25	77	80	3
14:25-14:30	80	84	4
14:30-14:35	86	88	2
14:35-14:40	94	93	1
14:40-14:45	90	86	4
14:45-14:50	79	82	3
14:50-14:55	81	83	2
14:55-15:00	81	82	1
15:00-15:05	81	86	5
15:05-15:10	81	82	1
15:10-15:15	80	80	0
15:15-15:20	78	78	0
15:20-15:25	74	76	2
15:25-15:30	77	77	0
15:30-15:35	77	77	0
15:35-15:40	81	81	0
15:40-15:45	82	80	2
15:45-15:50	77	76	1
15:50-15:55	76	79	3
15:55-16:00	79	79	0
16:00-16:05	77	75	2
16:05-16:10	76	78	2
16:10-16:15	75	78	3
16:15-16:20	77	77	0
16:20-16:25	75	75	0
16:25-16:30	76	76	0
16:30-16:35	75	76	1
16:35-16:40	76	79	3
16:40-16:45	79	79	0
16:45-16:50	84	85	1
16:50-16:55	68	75	7
16:55-17:00	75	75	0
17:00-17:05	74	74	0
17:05-17:10	75	76	1
17:10-17:15	73	74	1
17:15-17:20	76	77	1
17:20-17:25	72	73	1
17:25-17:30	73	75	2
17:30-17:35	73	75	2
17:35-17:40	79	80	1

Sum difference
4

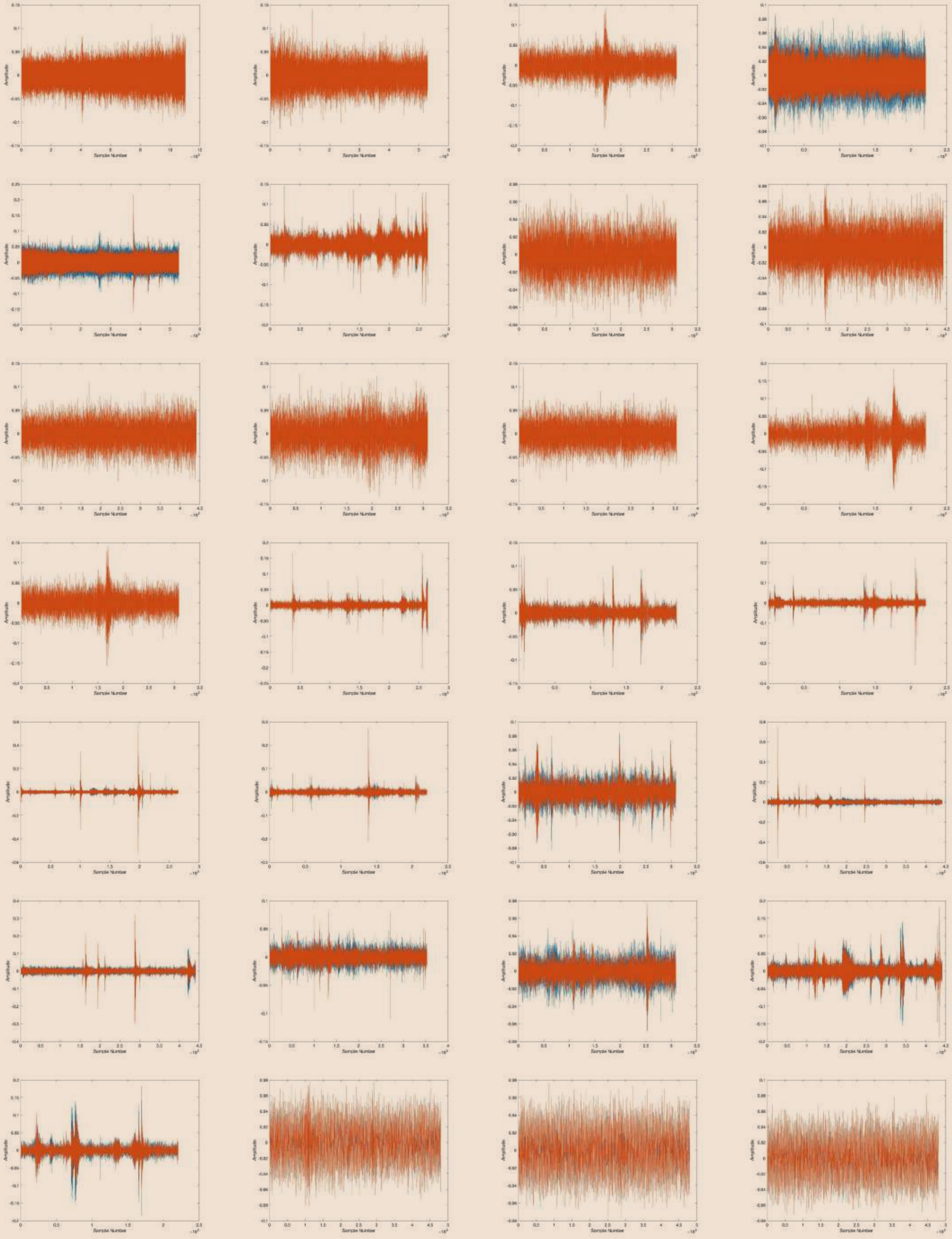
Sound Analysis: Alarm

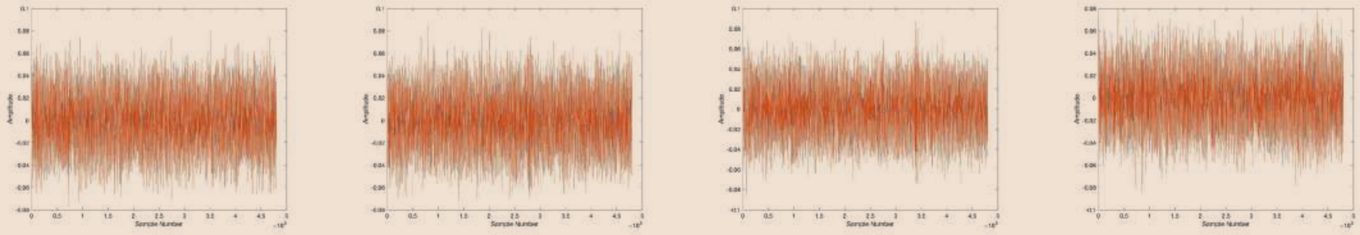
Matlab Audio Signal Analysis with 25 samples



Sound Analysis: Background

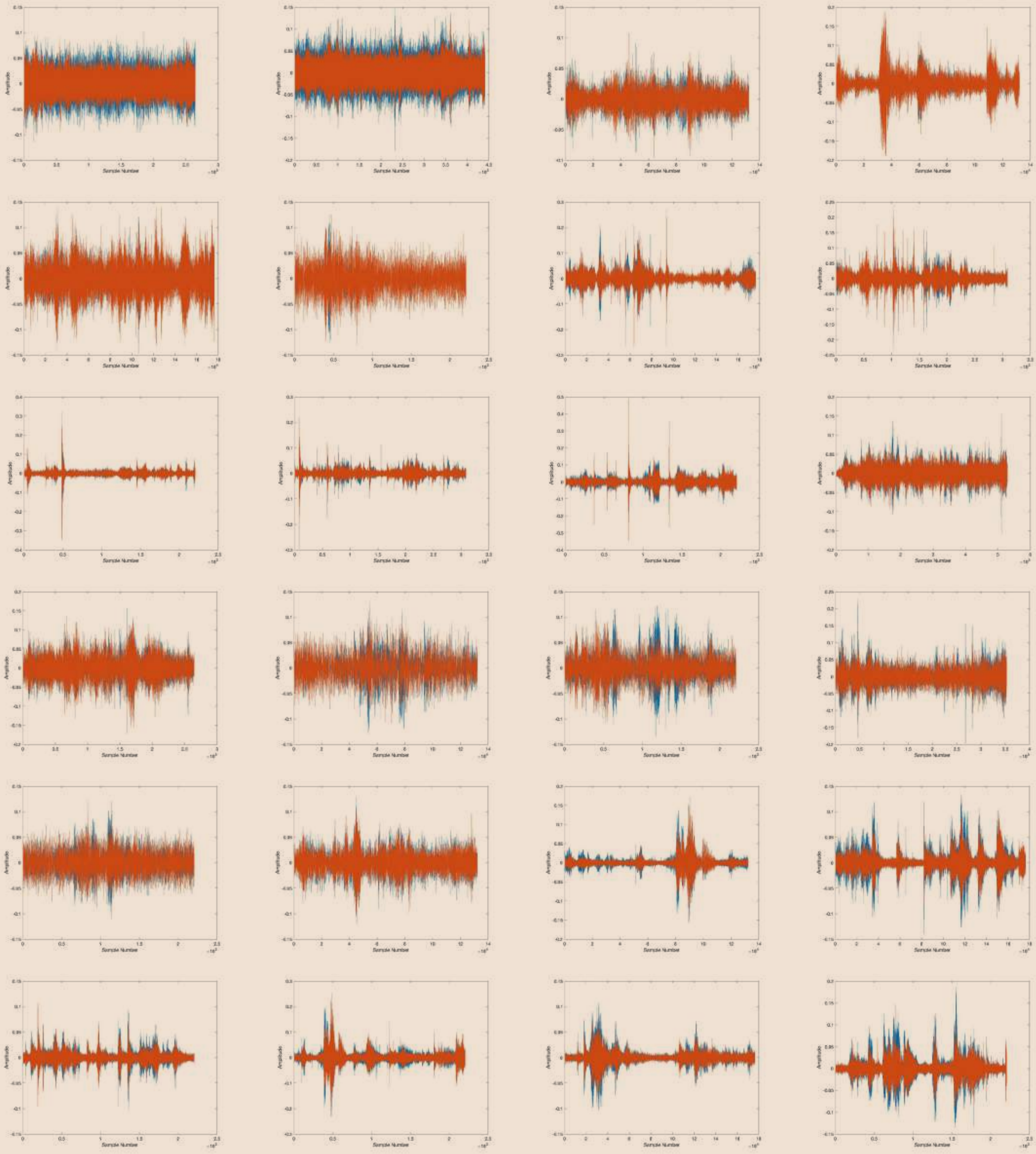
Matlab Audio Signal Analysis with 32 samples

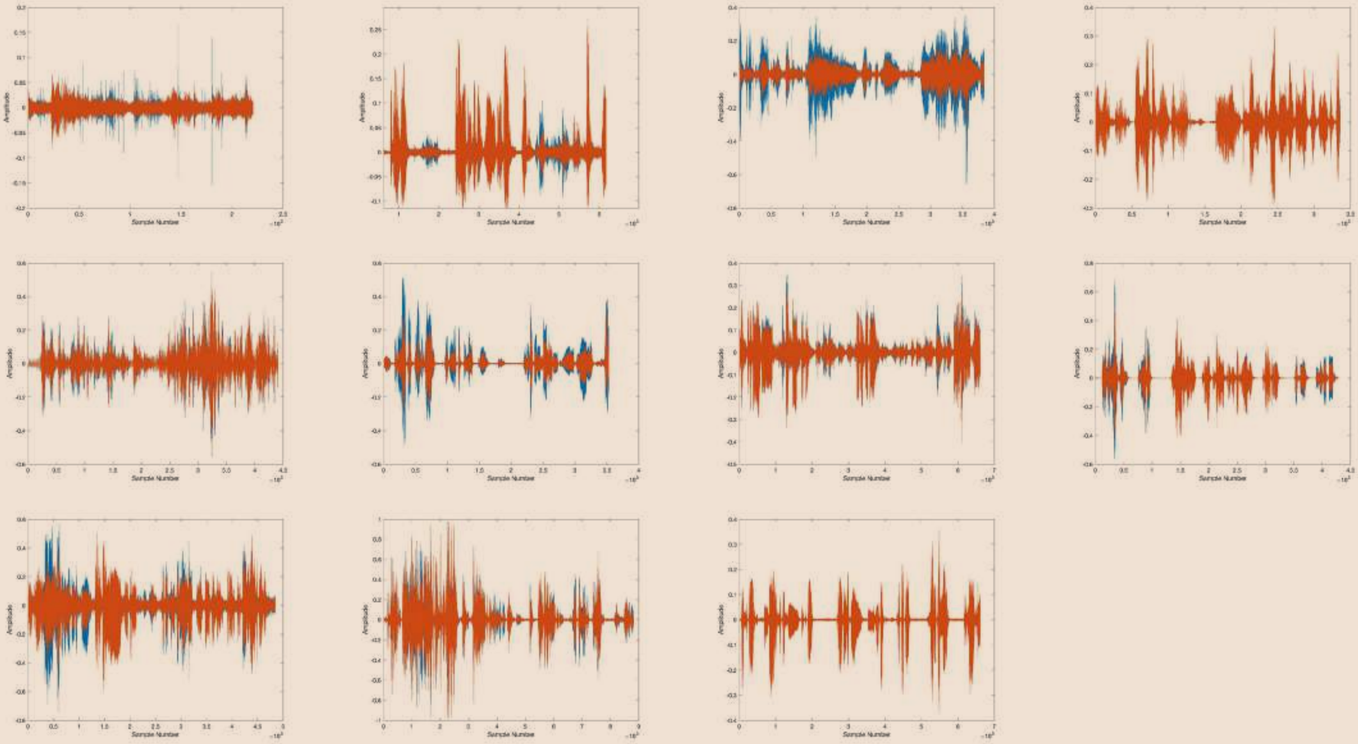




Sound Analysis: Conversation

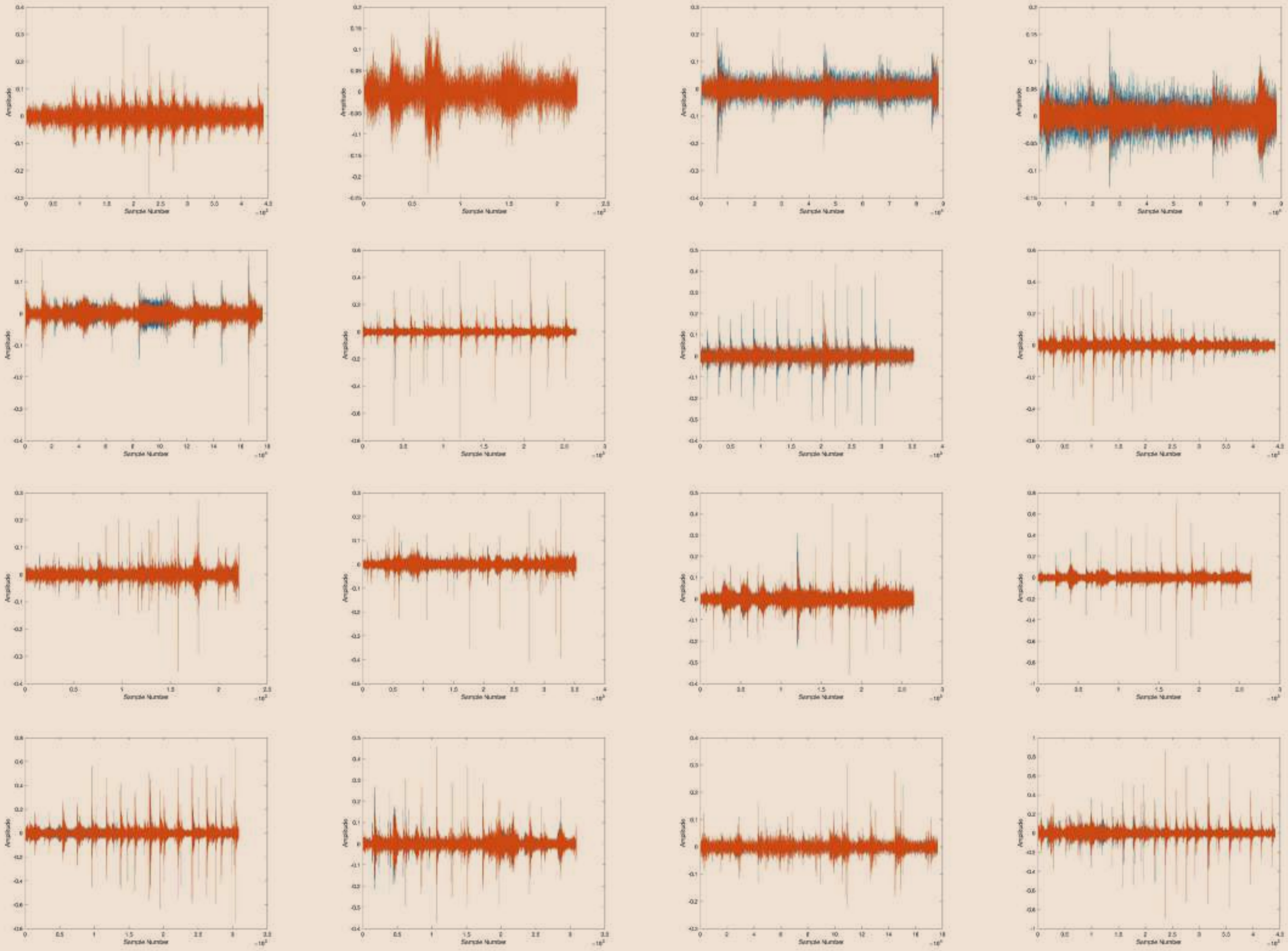
Matlab Audio Signal Analysis with 35 samples

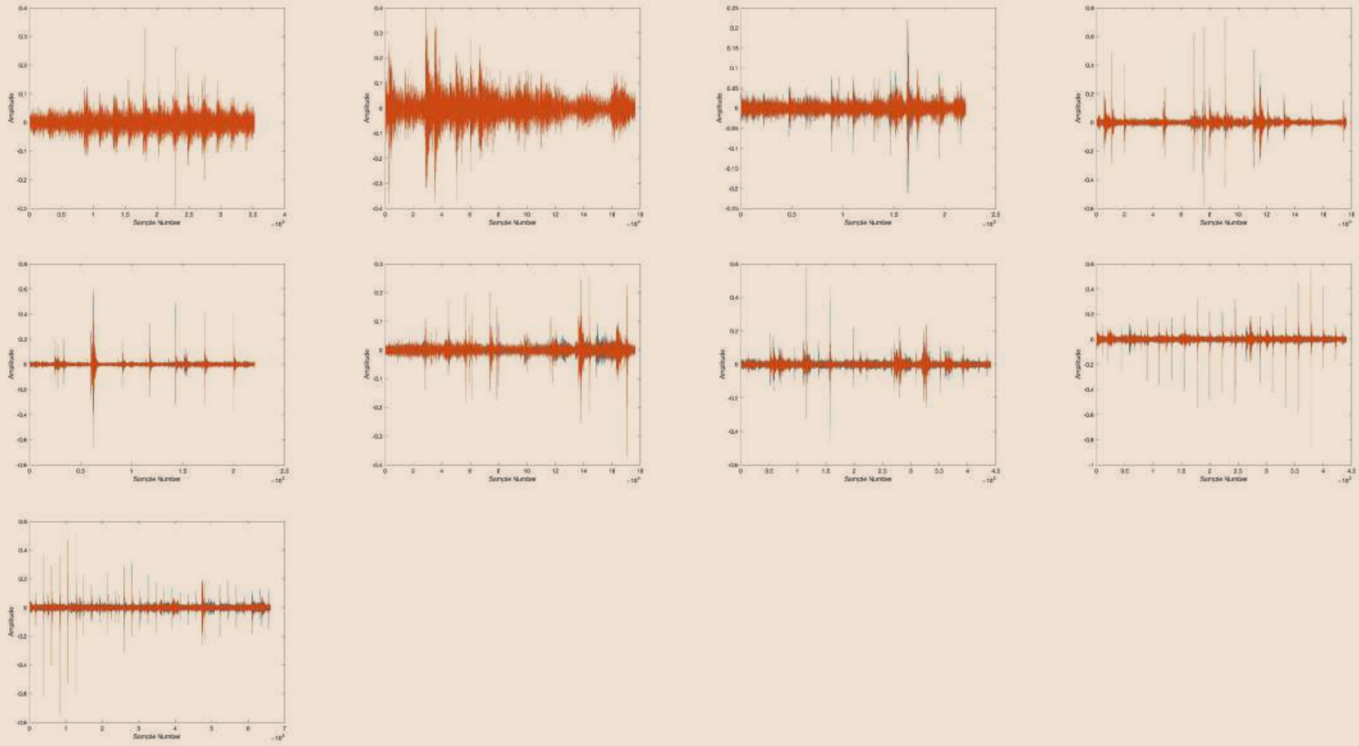




Sound Analysis: Footsteps

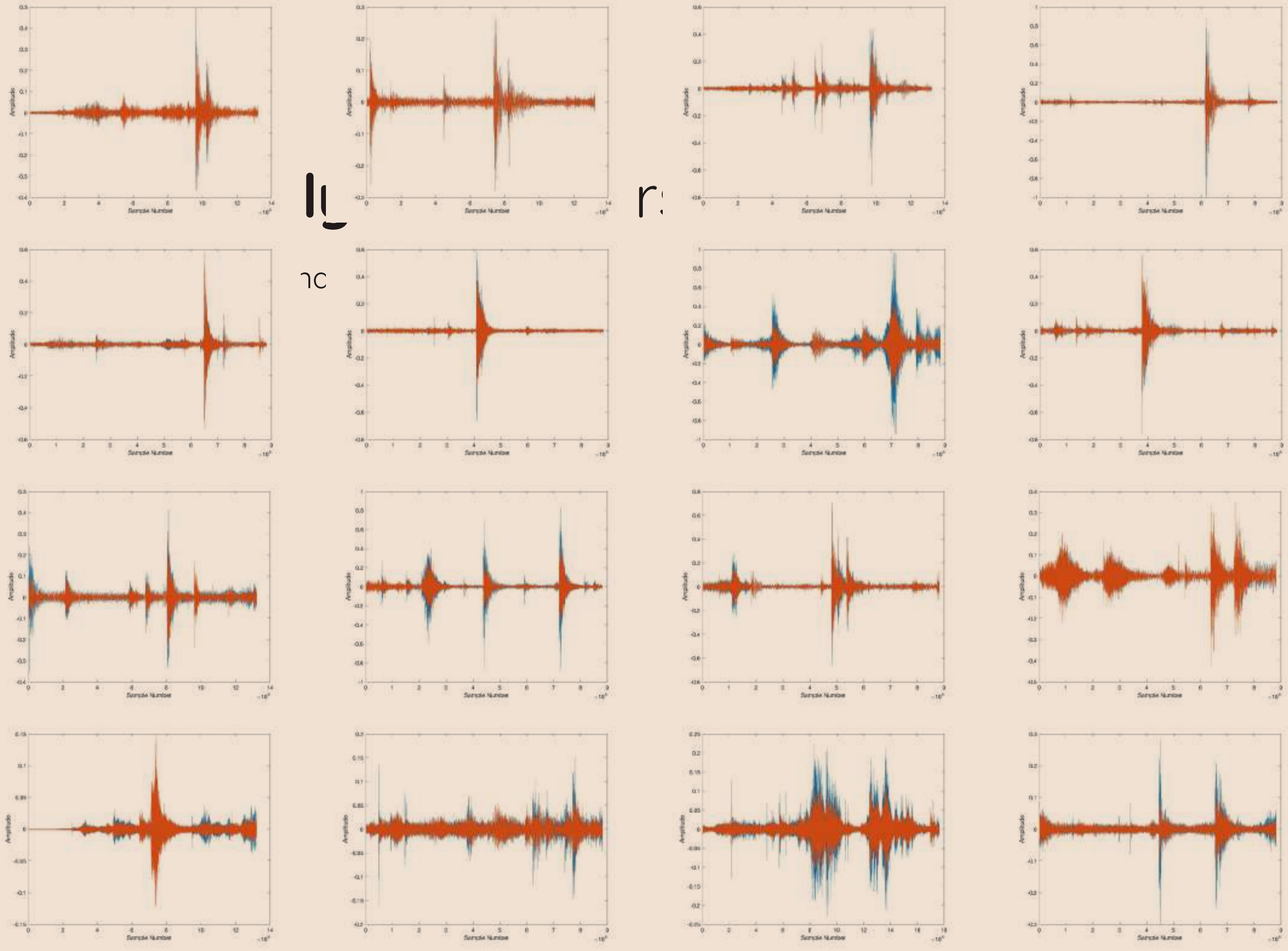
Matlab Audio Signal Analysis with 25 samples

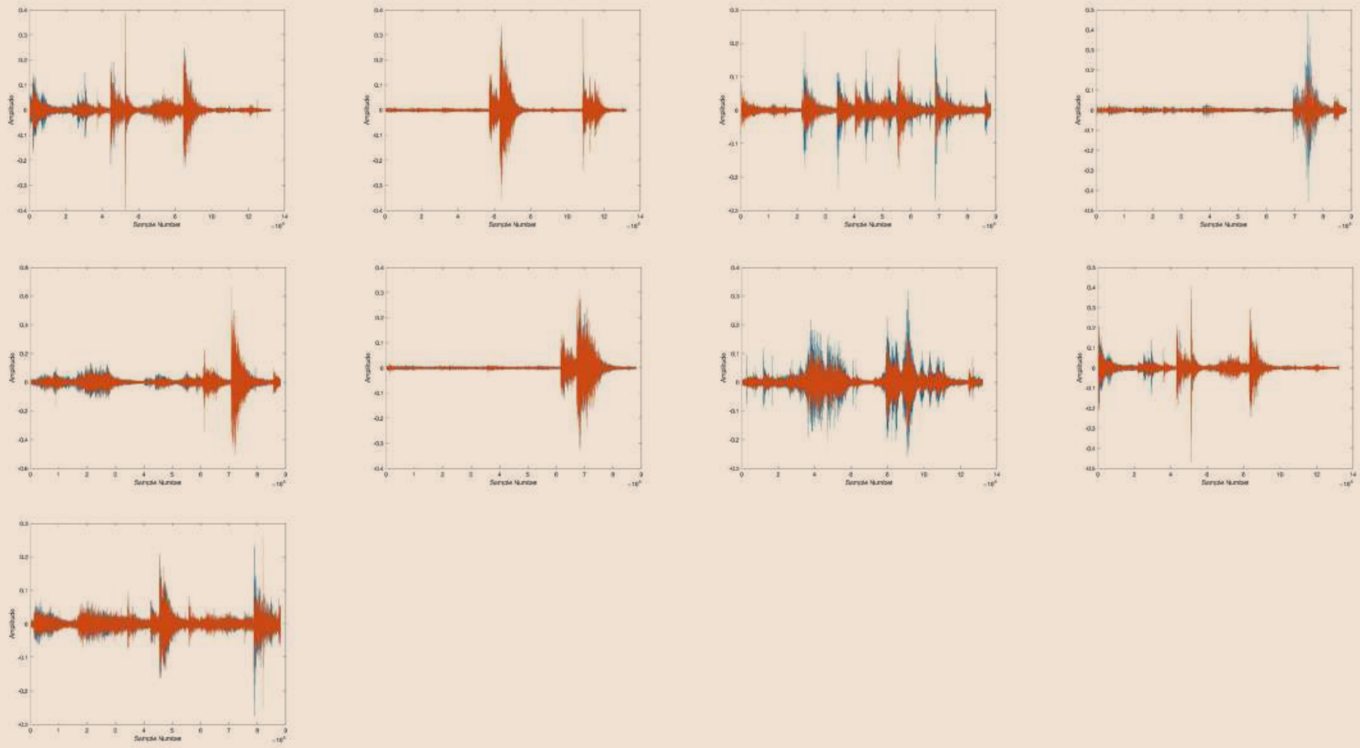




Sound Analysis: Objects clashing

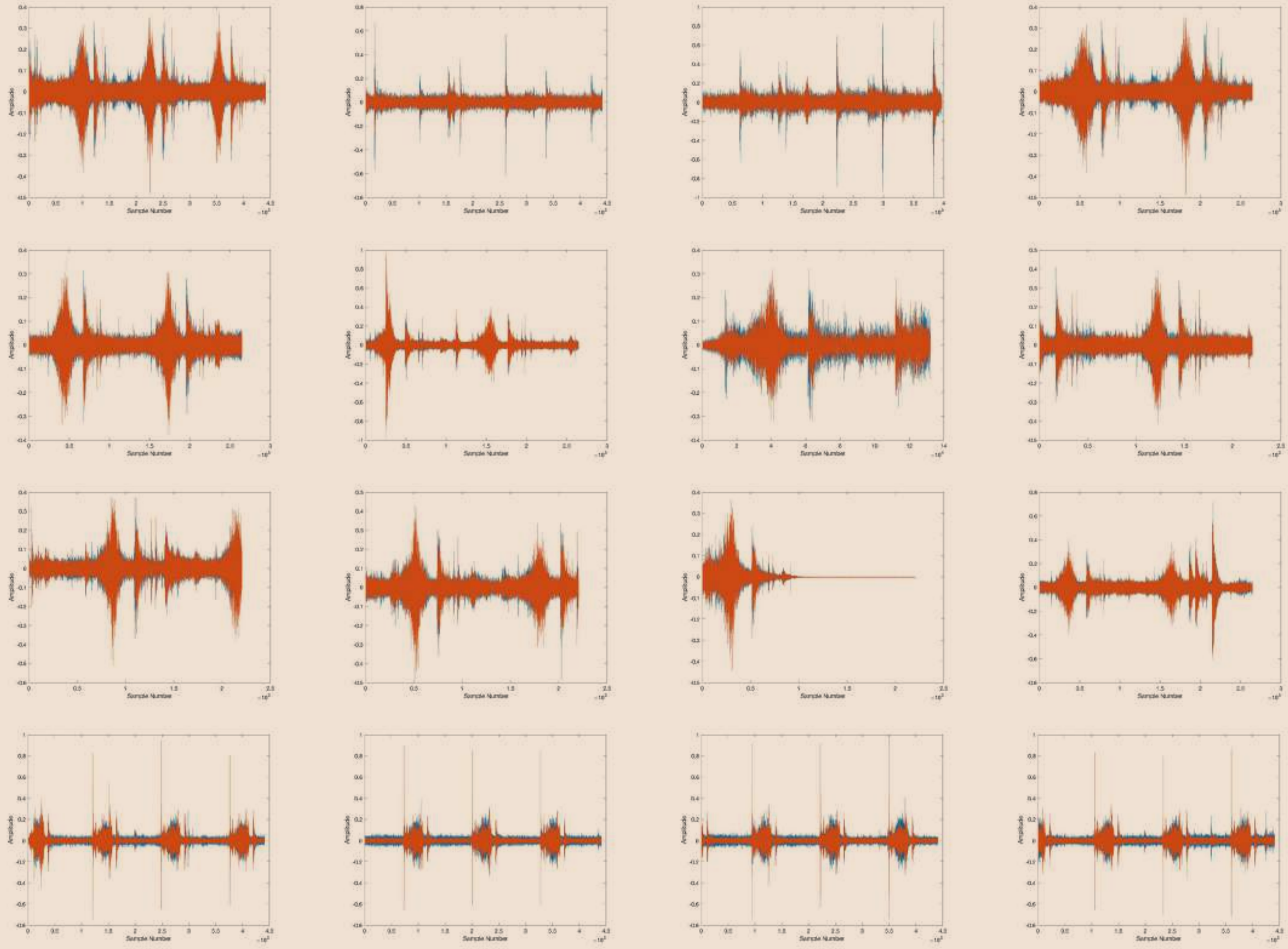
Matlab Audio Signal Analysis with 25 samples

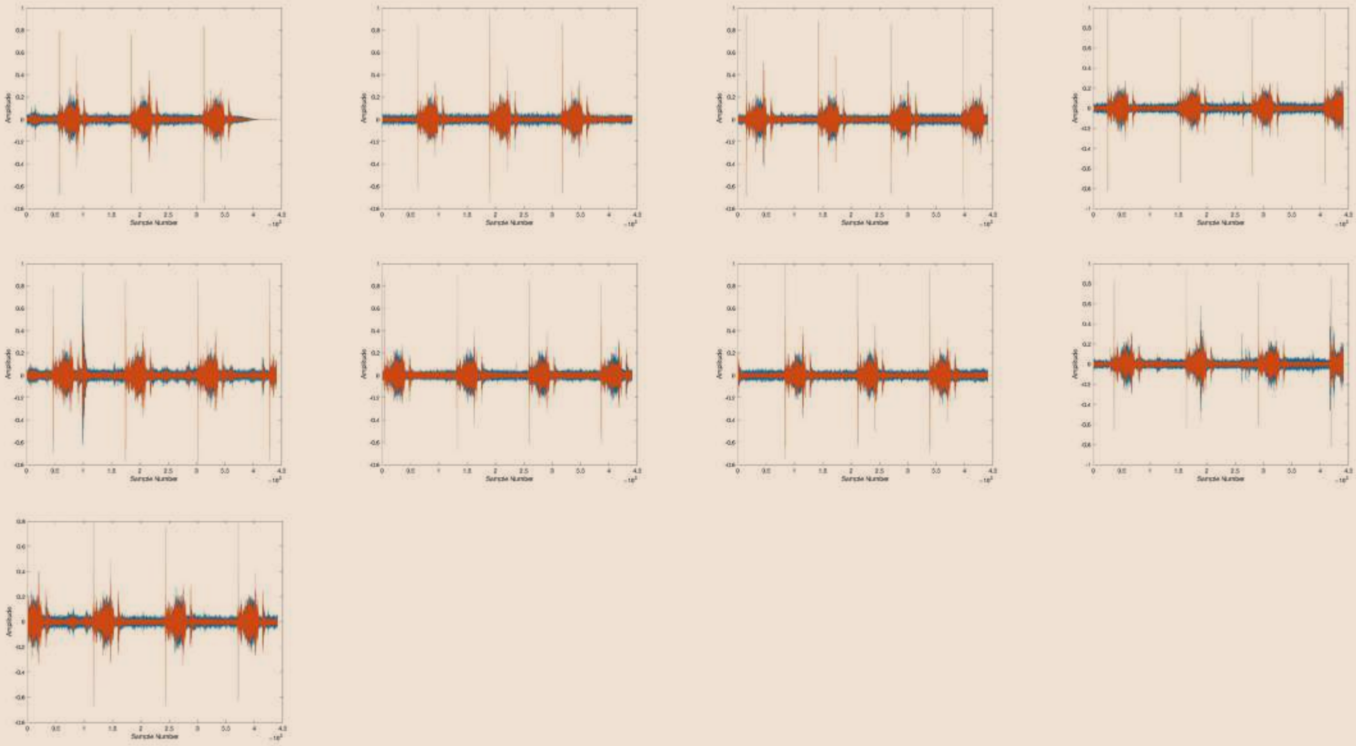




Sound Analysis: Ventilator

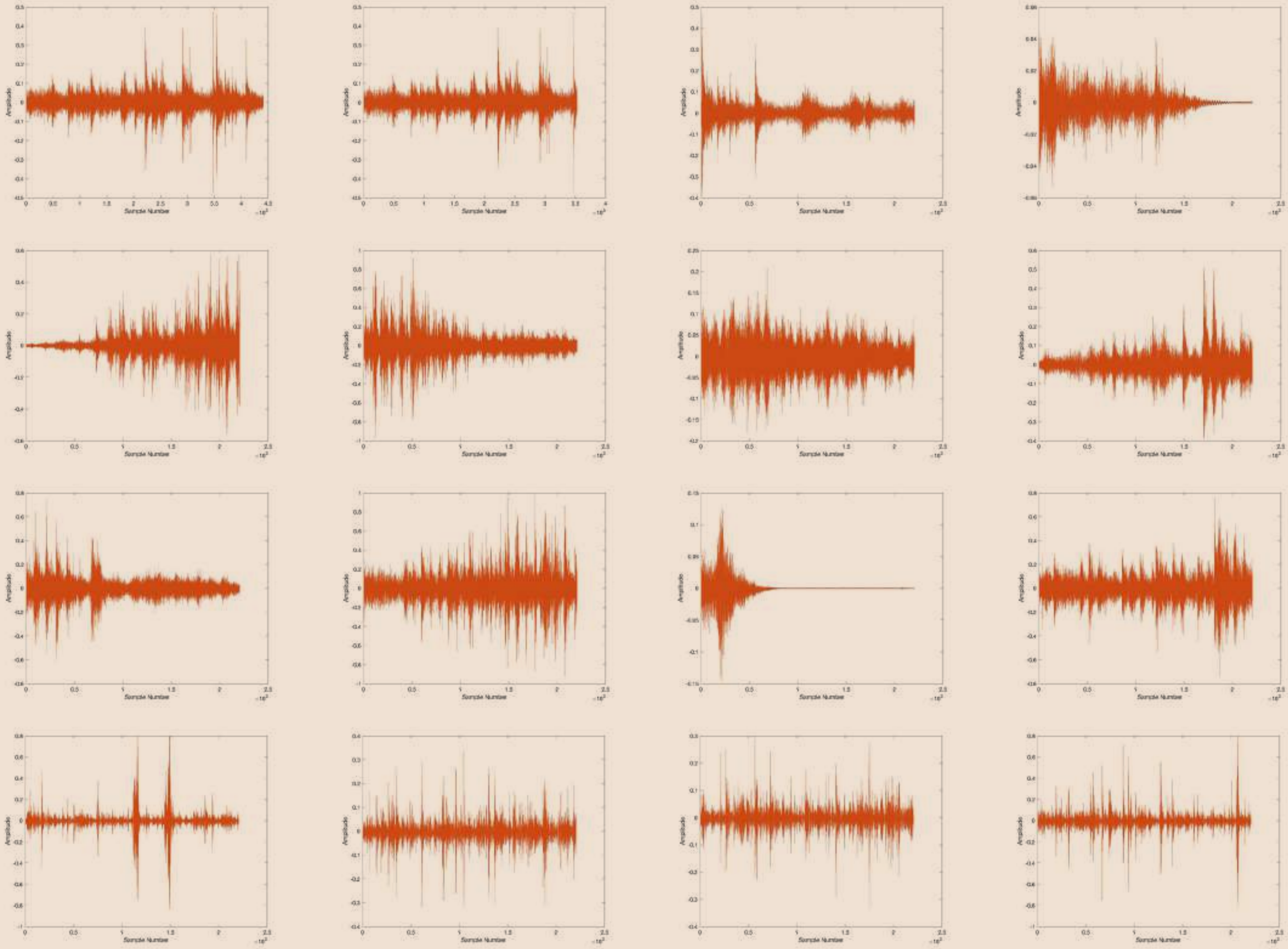
Matlab Audio Signal Analysis with 25 samples

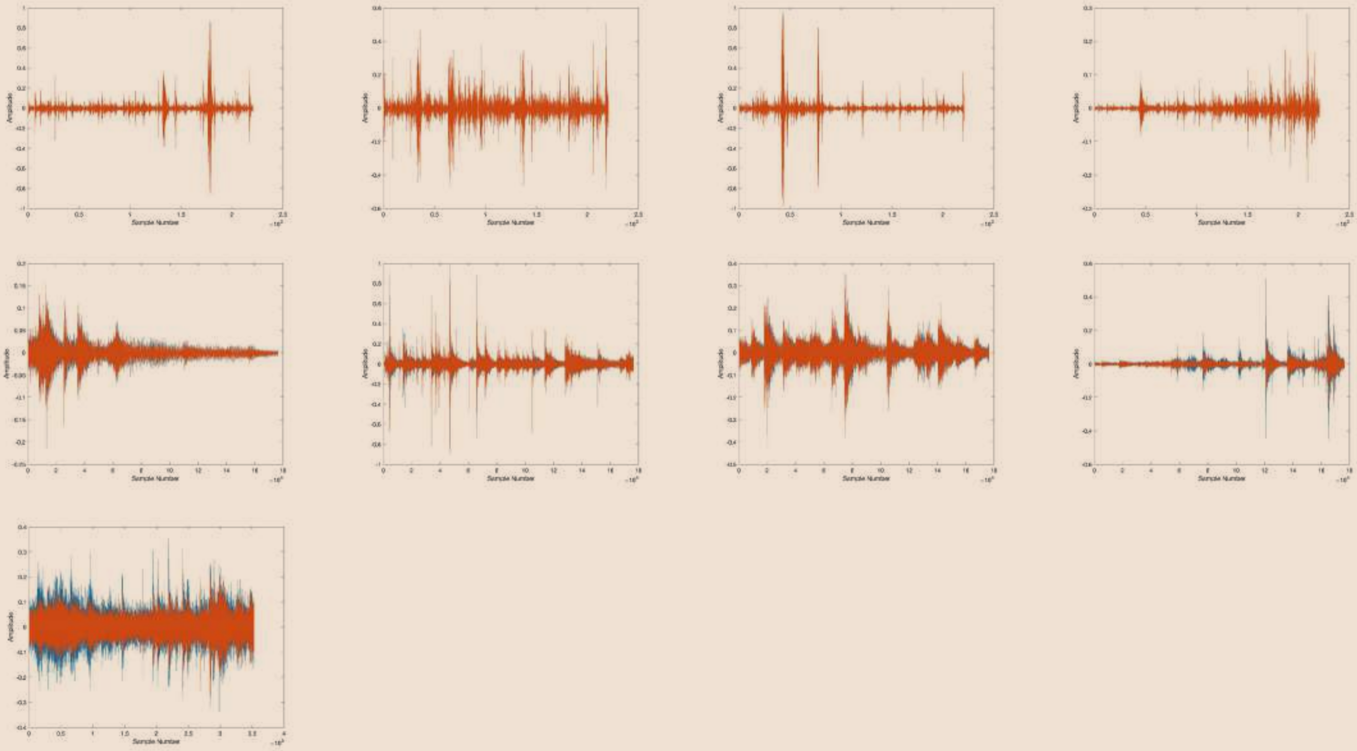




Sound Analysis: Trolley

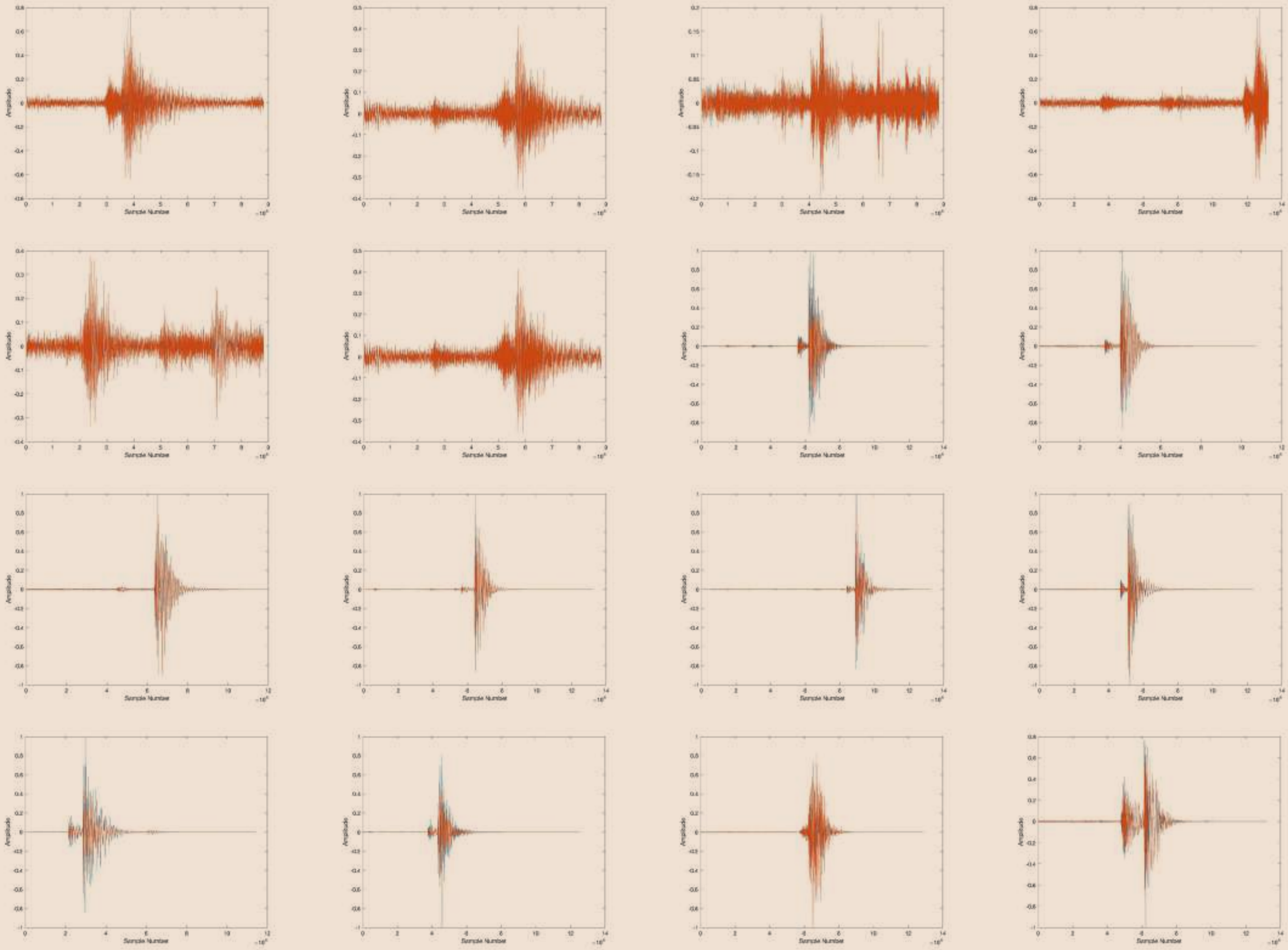
Matlab Audio Signal Analysis with 25 samples

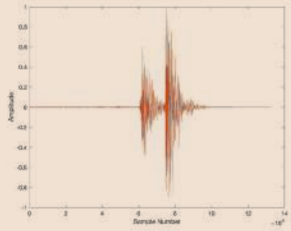


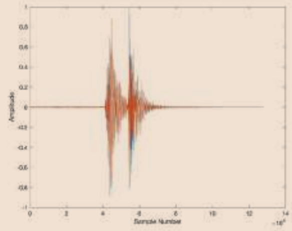


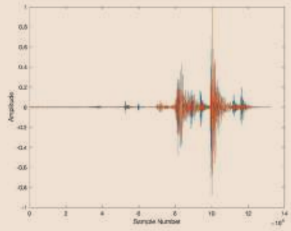
Sound Analysis: Door slam

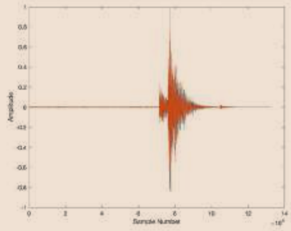
Matlab Audio Signal Analysis with 25 samples

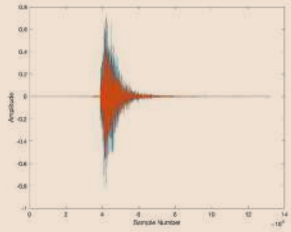


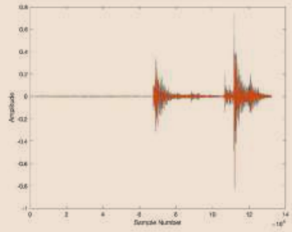

Amplitude
Sample Number
0 2 4 6 8 10 12 14

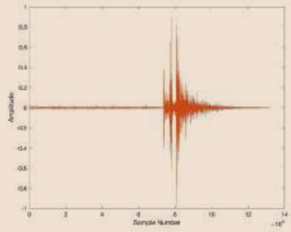

Amplitude
Sample Number
0 2 4 6 8 10 12 14

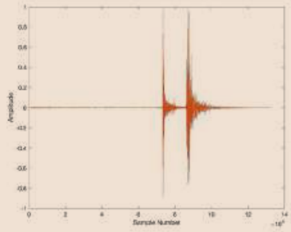

Amplitude
Sample Number
0 2 4 6 8 10 12 14

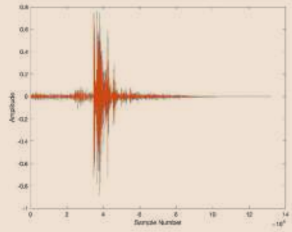

Amplitude
Sample Number
0 2 4 6 8 10 12 14


Amplitude
Sample Number
0 2 4 6 8 10 12 14


Amplitude
Sample Number
0 2 4 6 8 10 12 14

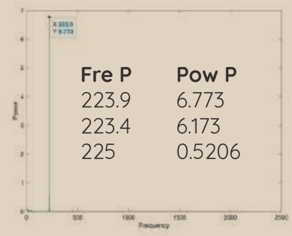

Amplitude
Sample Number
0 2 4 6 8 10 12 14


Amplitude
Sample Number
0 2 4 6 8 10 12 14

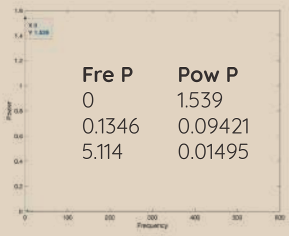

Amplitude
Sample Number
0 2 4 6 8 10 12 14

Sound Analysis: Alarm

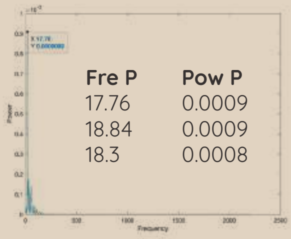
Matlab FFT(Fast Fourier Transform) Analysis with 25 samples



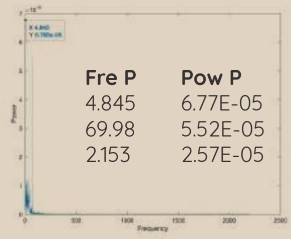
Fre P	Pow P
223.9	6.773
223.4	6.173
225	0.5206



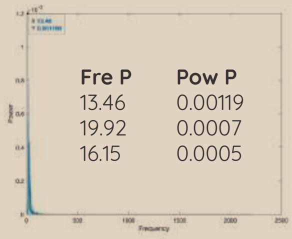
Fre P	Pow P
0	1.539
0.1346	0.09421
5.114	0.01495



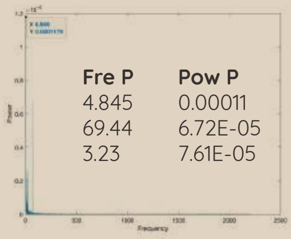
Fre P	Pow P
17.76	0.0009
18.84	0.0009
18.3	0.0008



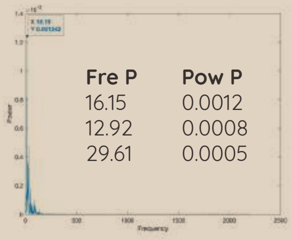
Fre P	Pow P
4.845	6.77E-05
69.98	5.52E-05
2.153	2.57E-05



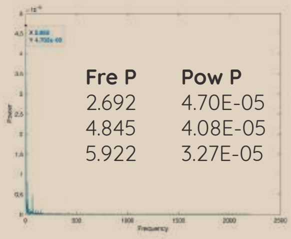
Fre P	Pow P
13.46	0.00119
19.92	0.0007
16.15	0.0005



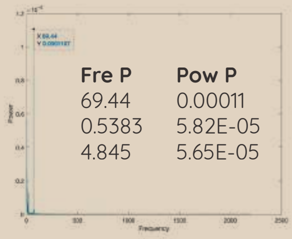
Fre P	Pow P
4.845	0.00011
69.44	6.72E-05
3.23	7.61E-05



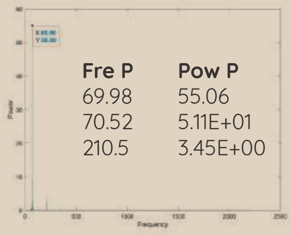
Fre P	Pow P
16.15	0.0012
12.92	0.0008
29.61	0.0005



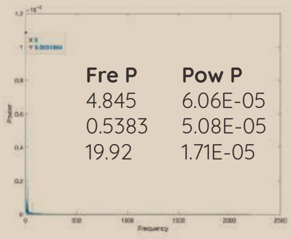
Fre P	Pow P
2.692	4.70E-05
4.845	4.08E-05
5.922	3.27E-05



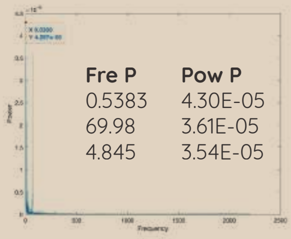
Fre P	Pow P
69.44	0.00011
0.5383	5.82E-05
4.845	5.65E-05



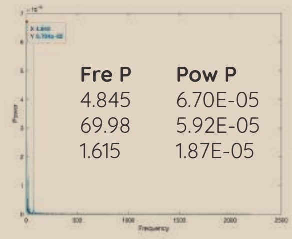
Fre P	Pow P
69.98	55.06
70.52	5.11E+01
210.5	3.45E+00



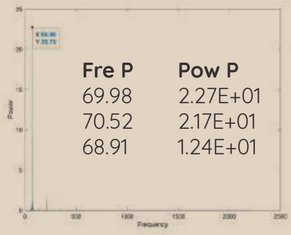
Fre P	Pow P
4.845	6.06E-05
0.5383	5.08E-05
19.92	1.71E-05



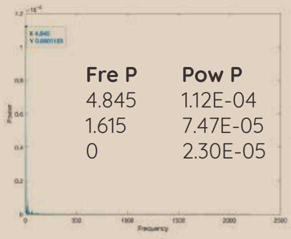
Fre P	Pow P
0.5383	4.30E-05
69.98	3.61E-05
4.845	3.54E-05



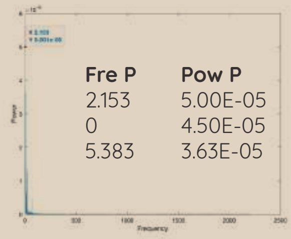
Fre P	Pow P
4.845	6.70E-05
69.98	5.92E-05
1.615	1.87E-05



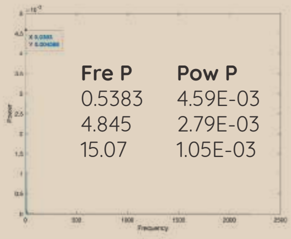
Fre P	Pow P
69.98	2.27E+01
70.52	2.17E+01
68.91	1.24E+01



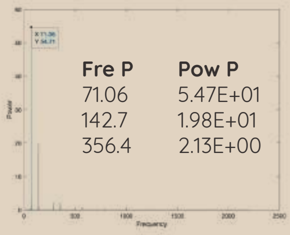
Fre P	Pow P
4.845	1.12E-04
1.615	7.47E-05
0	2.30E-05



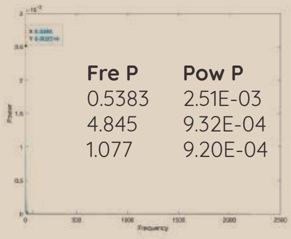
Fre P	Pow P
2.153	5.00E-05
0	4.50E-05
5.383	3.63E-05



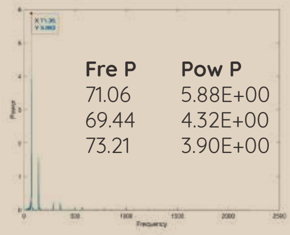
Fre P	Pow P
0.5383	4.59E-03
4.845	2.79E-03
15.07	1.05E-03



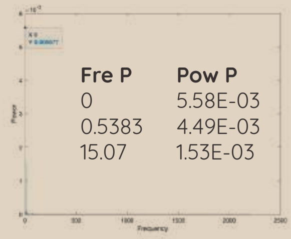
Fre P	Pow P
71.06	5.47E+01
142.7	1.98E+01
356.4	2.13E+00



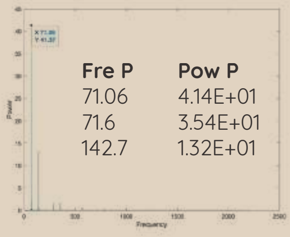
Fre P	Pow P
0.5383	2.51E-03
4.845	9.32E-04
1.077	9.20E-04



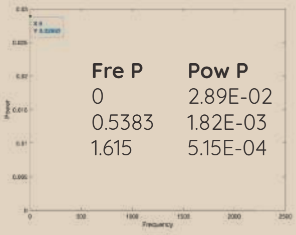
Fre P	Pow P
71.06	5.88E+00
69.44	4.32E+00
73.21	3.90E+00



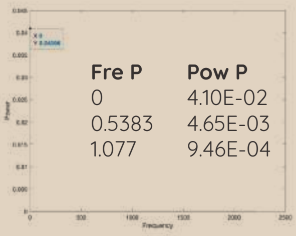
Fre P	Pow P
0	5.58E-03
0.5383	4.49E-03
15.07	1.53E-03



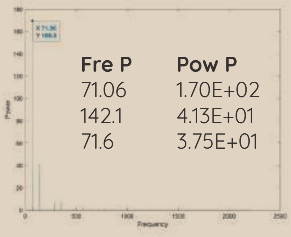
Fre P	Pow P
71.06	4.14E+01
71.6	3.54E+01
142.7	1.32E+01



Fre P	Pow P
0	2.89E-02
0.5383	1.82E-03
1.615	5.15E-04



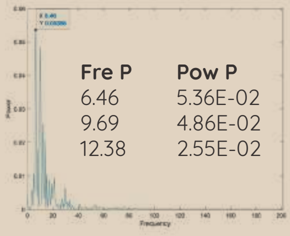
Fre P	Pow P
0	4.10E-02
0.5383	4.65E-03
1.077	9.46E-04



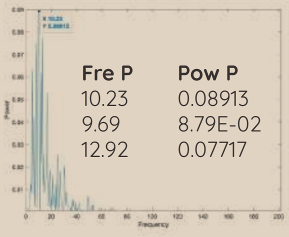
Fre P	Pow P
71.06	1.70E+02
142.1	4.13E+01
71.6	3.75E+01

Sound Analysis: Background

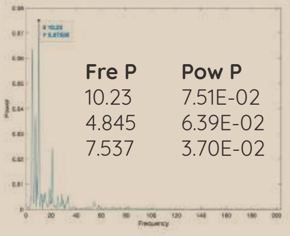
Matlab FFT(Fast Fourier Transform) Analysis with 32 samples



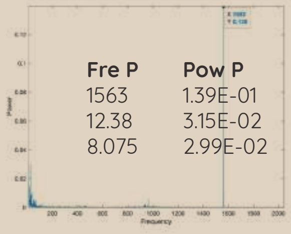
Fre P	Pow P
6.46	5.36E-02
9.69	4.86E-02
12.38	2.55E-02



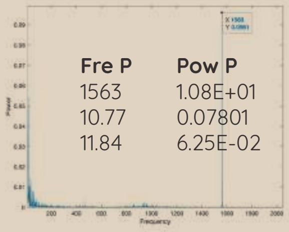
Fre P	Pow P
10.23	0.08913
9.69	8.79E-02
12.92	0.07717



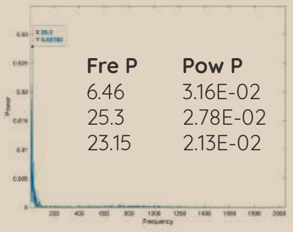
Fre P	Pow P
10.23	7.51E-02
4.845	6.39E-02
7.537	3.70E-02



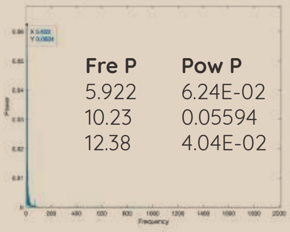
Fre P	Pow P
1563	1.39E-01
12.38	3.15E-02
8.075	2.99E-02



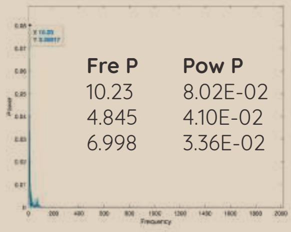
Fre P	Pow P
1563	1.08E+01
10.77	0.07801
11.84	6.25E-02



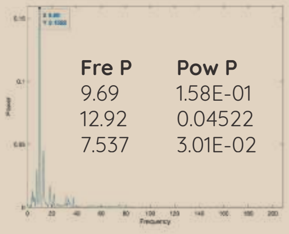
Fre P	Pow P
6.46	3.16E-02
25.3	2.78E-02
23.15	2.13E-02



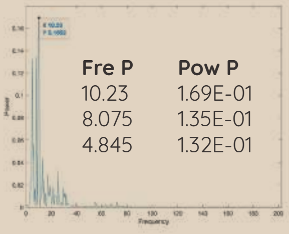
Fre P	Pow P
5.922	6.24E-02
10.23	0.05594
12.38	4.04E-02



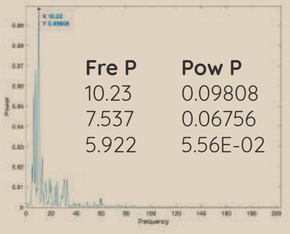
Fre P	Pow P
10.23	8.02E-02
4.845	4.10E-02
6.998	3.36E-02



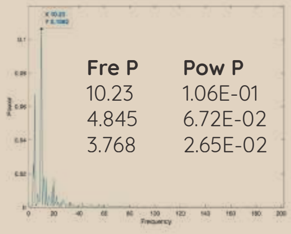
Fre P	Pow P
9.69	1.58E-01
12.92	0.04522
7.537	3.01E-02



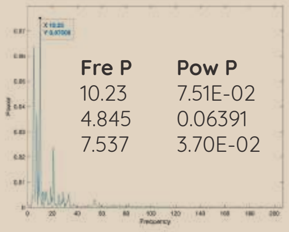
Fre P	Pow P
10.23	1.69E-01
8.075	1.35E-01
4.845	1.32E-01



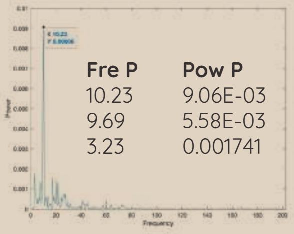
Fre P	Pow P
10.23	0.09808
7.537	0.06756
5.922	5.56E-02



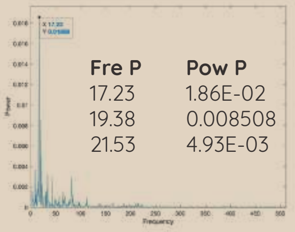
Fre P	Pow P
10.23	1.06E-01
4.845	6.72E-02
3.768	2.65E-02



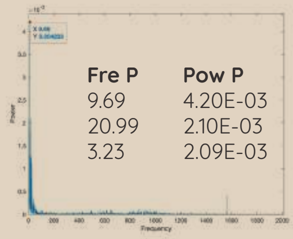
Fre P	Pow P
10.23	7.51E-02
4.845	0.06391
7.537	3.70E-02



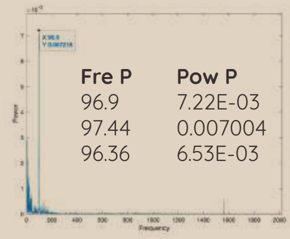
Fre P	Pow P
10.23	9.06E-03
9.69	5.58E-03
3.23	0.001741



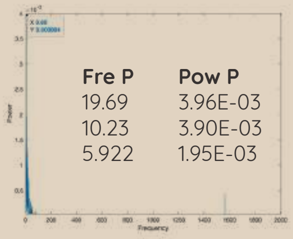
Fre P	Pow P
17.23	1.86E-02
19.38	0.008508
21.53	4.93E-03



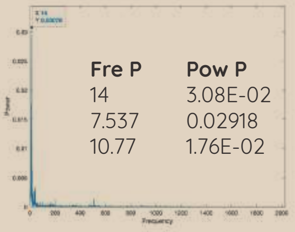
Fre P	Pow P
9.69	4.20E-03
20.99	2.10E-03
3.23	2.09E-03



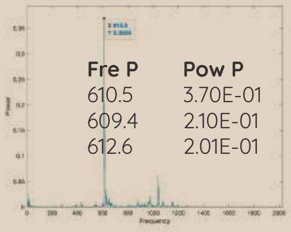
Fre P	Pow P
96.9	7.22E-03
97.44	0.007004
96.36	6.53E-03



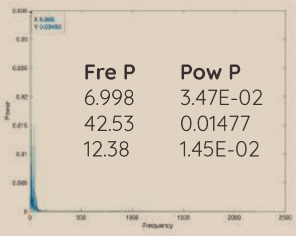
Fre P	Pow P
19.69	3.96E-03
10.23	3.90E-03
5.922	1.95E-03



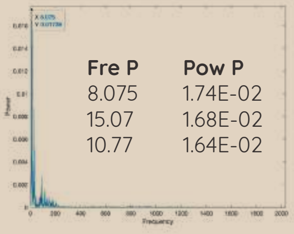
Fre P	Pow P
14	3.08E-02
7.537	0.02918
10.77	1.76E-02



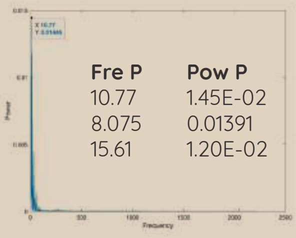
Fre P	Pow P
610.5	3.70E-01
609.4	2.10E-01
612.6	2.01E-01



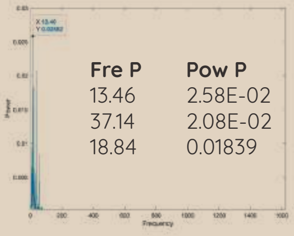
Fre P	Pow P
6.998	3.47E-02
42.53	0.01477
12.38	1.45E-02



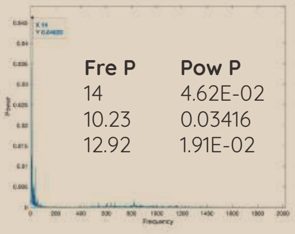
Fre P	Pow P
8.075	1.74E-02
15.07	1.68E-02
10.77	1.64E-02



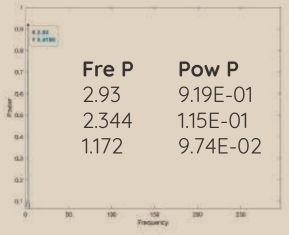
Fre P	Pow P
10.77	1.45E-02
8.075	0.01391
15.61	1.20E-02



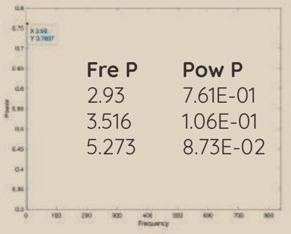
Fre P	Pow P
13.46	2.58E-02
37.14	2.08E-02
18.84	0.01839



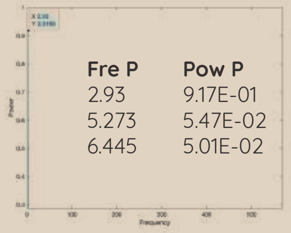
Fre P	Pow P
14	4.62E-02
10.23	0.03416
12.92	1.91E-02



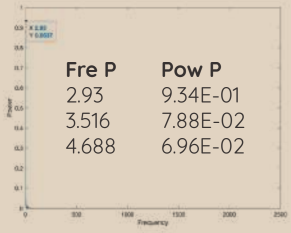
Fre P	Pow P
2.93	9.19E-01
2.344	1.15E-01
1.172	9.74E-02



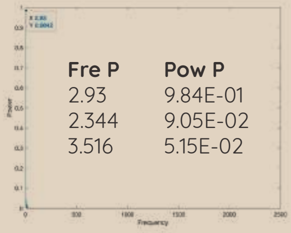
Fre P	Pow P
2.93	7.61E-01
3.516	1.06E-01
5.273	8.73E-02



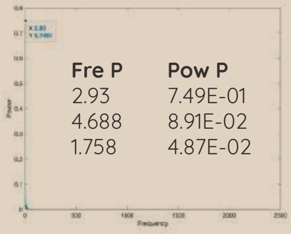
Fre P	Pow P
2.93	9.17E-01
5.273	5.47E-02
6.445	5.01E-02



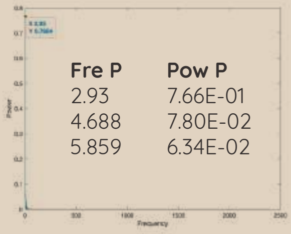
Fre P	Pow P
2.93	9.34E-01
3.516	7.88E-02
4.688	6.96E-02



Fre P	Pow P
2.93	9.84E-01
2.344	9.05E-02
3.516	5.15E-02



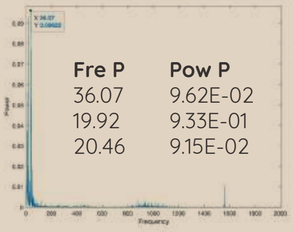
Fre P	Pow P
2.93	7.49E-01
4.688	8.91E-02
1.758	4.87E-02



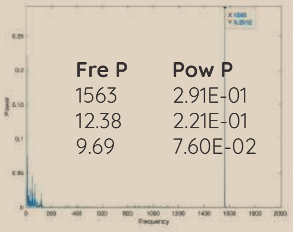
Fre P	Pow P
2.93	7.66E-01
4.688	7.80E-02
5.859	6.34E-02

Sound Analysis: Conversation

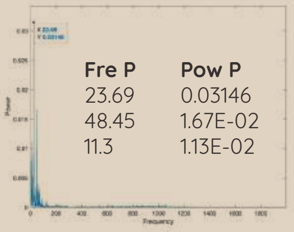
Matlab FFT(Fast Fourier Transform) Analysis with 35 samples



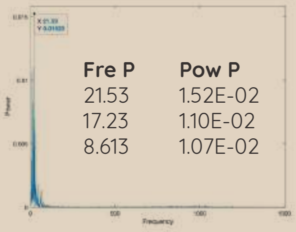
Fre P	Pow P
36.07	9.62E-02
19.92	9.33E-01
20.46	9.15E-02



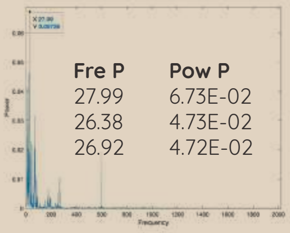
Fre P	Pow P
1563	2.91E-01
12.38	2.21E-01
9.69	7.60E-02



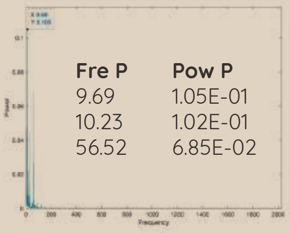
Fre P	Pow P
23.69	0.03146
48.45	1.67E-02
11.3	1.13E-02



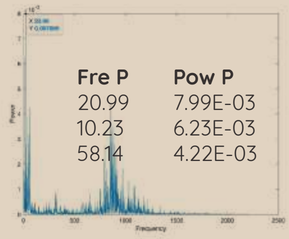
Fre P	Pow P
21.53	1.52E-02
17.23	1.10E-02
8.613	1.07E-02



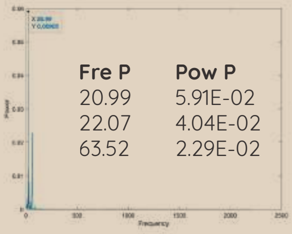
Fre P	Pow P
27.99	6.73E-02
26.38	4.73E-02
26.92	4.72E-02



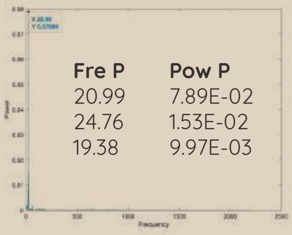
Fre P	Pow P
9.69	1.05E-01
10.23	1.02E-01
56.52	6.85E-02



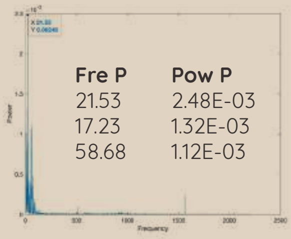
Fre P	Pow P
20.99	7.99E-03
10.23	6.23E-03
58.14	4.22E-03



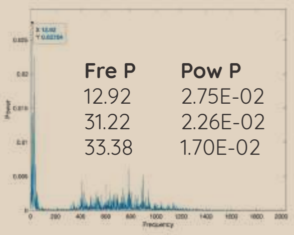
Fre P	Pow P
20.99	5.91E-02
22.07	4.04E-02
63.52	2.29E-02



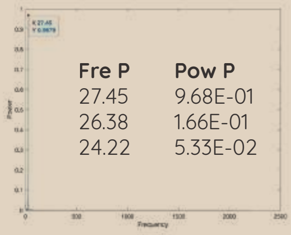
Fre P	Pow P
20.99	7.89E-02
24.76	1.53E-02
19.38	9.97E-03



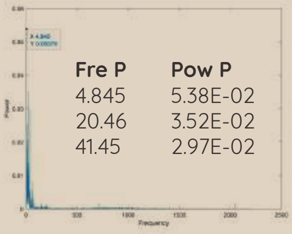
Fre P	Pow P
21.53	2.48E-03
17.23	1.32E-03
58.68	1.12E-03



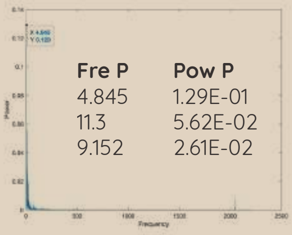
Fre P	Pow P
12.92	2.75E-02
31.22	2.26E-02
33.38	1.70E-02



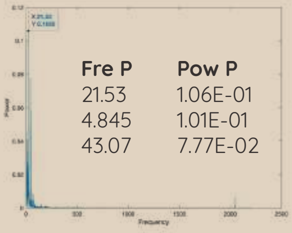
Fre P	Pow P
27.45	9.68E-01
26.38	1.66E-01
24.22	5.33E-02



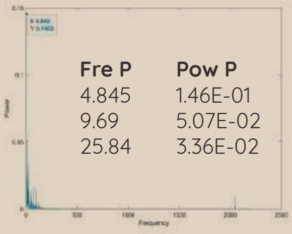
Fre P	Pow P
4.845	5.38E-02
20.46	3.52E-02
41.45	2.97E-02



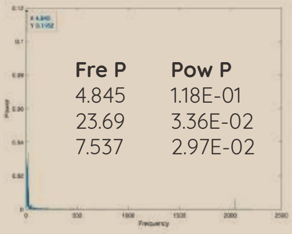
Fre P	Pow P
4.845	1.29E-01
11.3	5.62E-02
9.152	2.61E-02



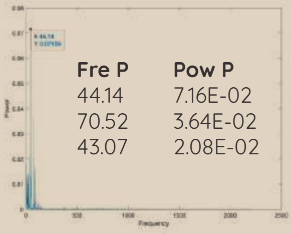
Fre P	Pow P
21.53	1.06E-01
4.845	1.01E-01
43.07	7.77E-02



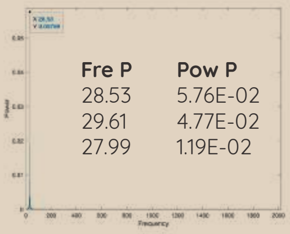
Fre P	Pow P
4.845	1.46E-01
9.69	5.07E-02
25.84	3.36E-02



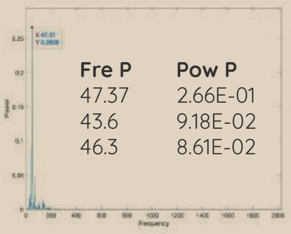
Fre P	Pow P
4.845	1.18E-01
23.69	3.36E-02
7.537	2.97E-02



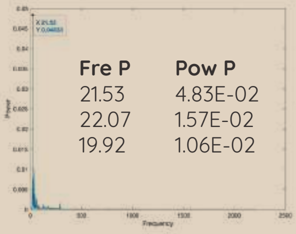
Fre P	Pow P
44.14	7.16E-02
70.52	3.64E-02
43.07	2.08E-02



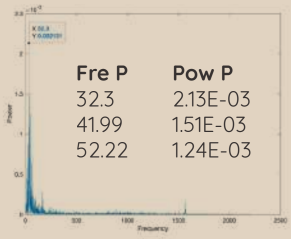
Fre P	Pow P
28.53	5.76E-02
29.61	4.77E-02
27.99	1.19E-02



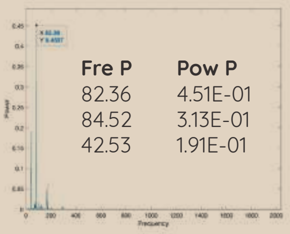
Fre P	Pow P
47.37	2.66E-01
43.6	9.18E-02
46.3	8.61E-02



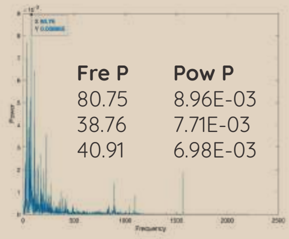
Fre P	Pow P
21.53	4.83E-02
22.07	1.57E-02
19.92	1.06E-02



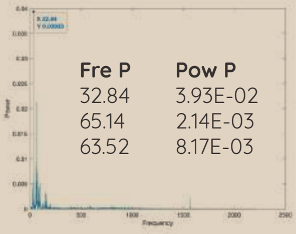
Fre P	Pow P
32.3	2.13E-03
41.99	1.51E-03
52.22	1.24E-03



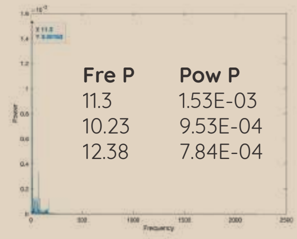
Fre P	Pow P
82.36	4.51E-01
84.52	3.13E-01
42.53	1.91E-01



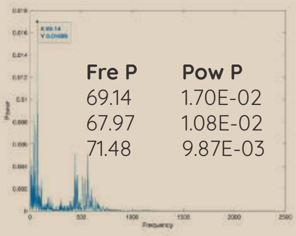
Fre P	Pow P
80.75	8.96E-03
38.76	7.71E-03
40.91	6.98E-03



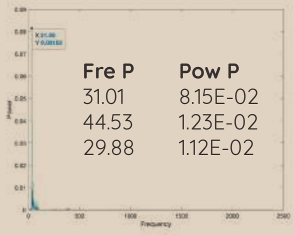
Fre P	Pow P
32.84	3.93E-02
65.14	2.14E-03
63.52	8.17E-03



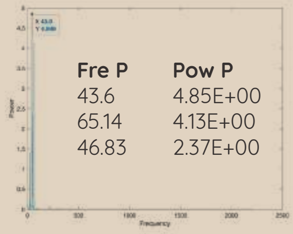
Fre P	Pow P
11.3	1.53E-03
10.23	9.53E-04
12.38	7.84E-04



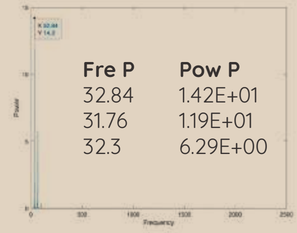
Fre P	Pow P
69.14	1.70E-02
67.97	1.08E-02
71.48	9.87E-03



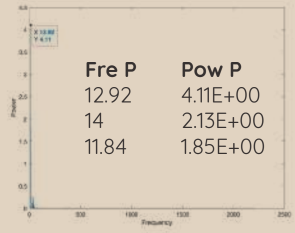
Fre P	Pow P
31.01	8.15E-02
44.53	1.23E-02
29.88	1.12E-02



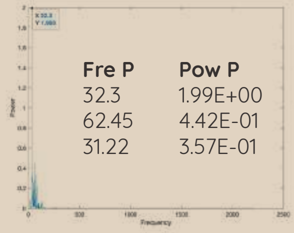
Fre P	Pow P
43.6	4.85E+00
65.14	4.13E+00
46.83	2.37E+00



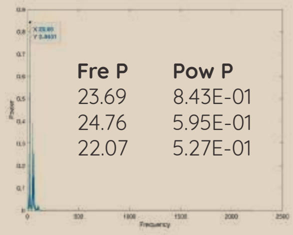
Fre P	Pow P
32.84	1.42E+01
31.76	1.19E+01
32.3	6.29E+00



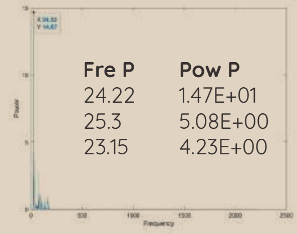
Fre P	Pow P
12.92	4.11E+00
14	2.13E+00
11.84	1.85E+00



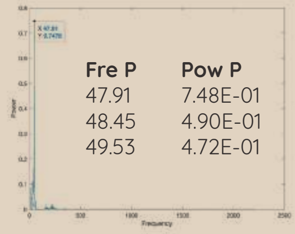
Fre P	Pow P
32.3	1.99E+00
62.45	4.42E-01
31.22	3.57E-01



Fre P	Pow P
23.69	8.43E-01
24.76	5.95E-01
22.07	5.27E-01



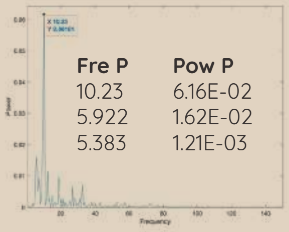
Fre P	Pow P
24.22	1.47E+01
25.3	5.08E+00
23.15	4.23E+00



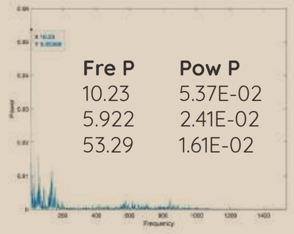
Fre P	Pow P
47.91	7.48E-01
48.45	4.90E-01
49.53	4.72E-01

Sound Analysis: Footsteps

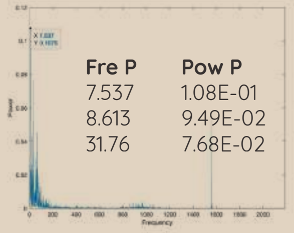
Matlab FFT(Fast Fourier Transform) Analysis with 25 samples



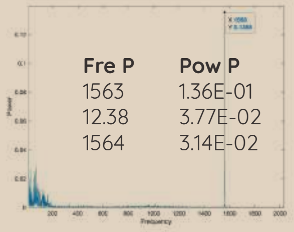
Fre P	Pow P
10.23	6.16E-02
5.922	1.62E-02
5.383	1.21E-03



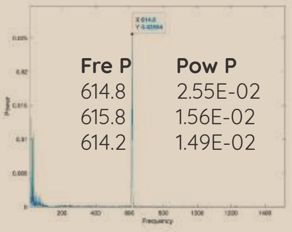
Fre P	Pow P
10.23	5.37E-02
5.922	2.41E-02
53.29	1.61E-02



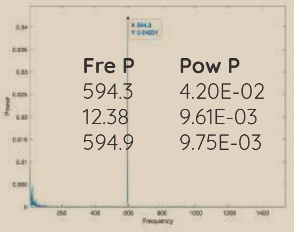
Fre P	Pow P
7.537	1.08E-01
8.613	9.49E-02
31.76	7.68E-02



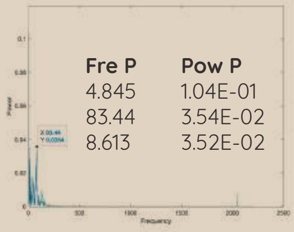
Fre P	Pow P
1563	1.36E-01
12.38	3.77E-02
1564	3.14E-02



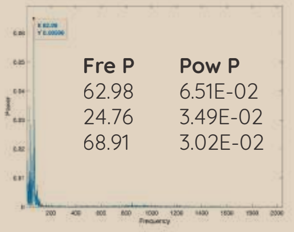
Fre P	Pow P
614.8	2.55E-02
615.8	1.56E-02
614.2	1.49E-02



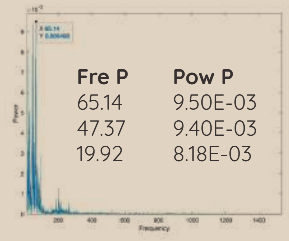
Fre P	Pow P
594.3	4.20E-02
12.38	9.61E-03
594.9	9.75E-03



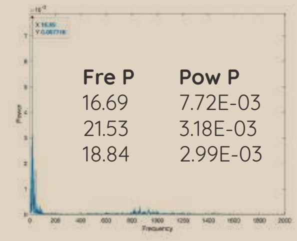
Fre P	Pow P
4.845	1.04E-01
83.44	3.54E-02
8.613	3.52E-02



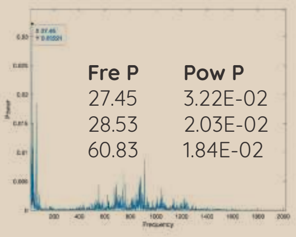
Fre P	Pow P
62.98	6.51E-02
24.76	3.49E-02
68.91	3.02E-02



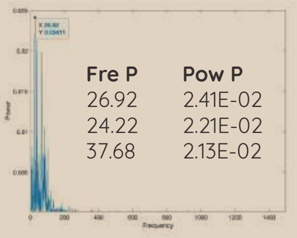
Fre P	Pow P
65.14	9.50E-03
47.37	9.40E-03
19.92	8.18E-03



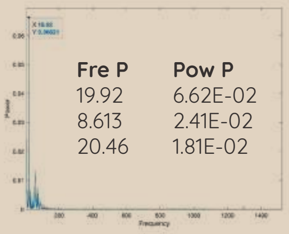
Fre P	Pow P
16.69	7.72E-03
21.53	3.18E-03
18.84	2.99E-03



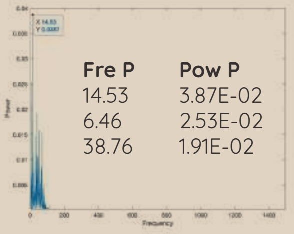
Fre P	Pow P
27.45	3.22E-02
28.53	2.03E-02
60.83	1.84E-02



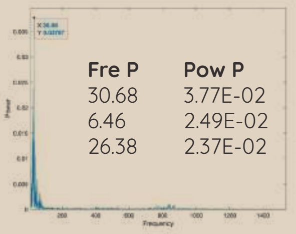
Fre P	Pow P
26.92	2.41E-02
24.22	2.21E-02
37.68	2.13E-02



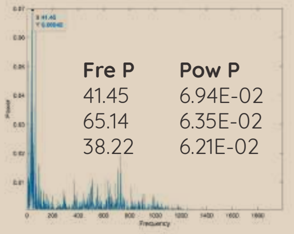
Fre P	Pow P
19.92	6.62E-02
8.613	2.41E-02
20.46	1.81E-02



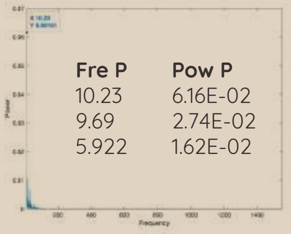
Fre P	Pow P
14.53	3.87E-02
6.46	2.53E-02
38.76	1.91E-02



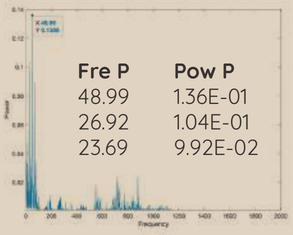
Fre P	Pow P
30.68	3.77E-02
6.46	2.49E-02
26.38	2.37E-02



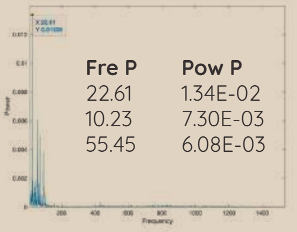
Fre P	Pow P
41.45	6.94E-02
65.14	6.35E-02
38.22	6.21E-02



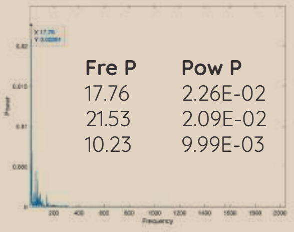
Fre P	Pow P
10.23	6.16E-02
9.69	2.74E-02
5.922	1.62E-02



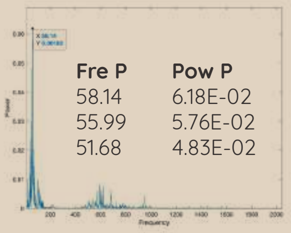
Fre P	Pow P
48.99	1.36E-01
26.92	1.04E-01
23.69	9.92E-02



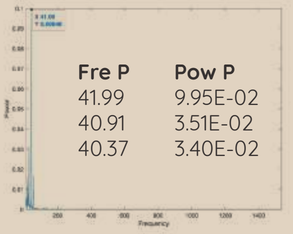
Fre P	Pow P
22.61	1.34E-02
10.23	7.30E-03
55.45	6.08E-03



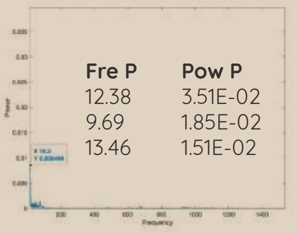
Fre P	Pow P
17.76	2.26E-02
21.53	2.09E-02
10.23	9.99E-03



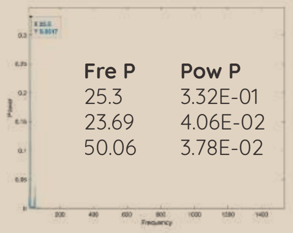
Fre P	Pow P
58.14	6.18E-02
55.99	5.76E-02
51.68	4.83E-02



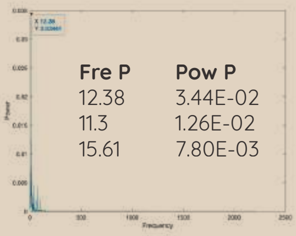
Fre P	Pow P
41.99	9.95E-02
40.91	3.51E-02
40.37	3.40E-02



Fre P	Pow P
12.38	3.51E-02
9.69	1.85E-02
13.46	1.51E-02



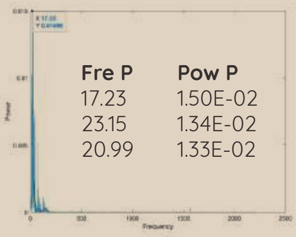
Fre P	Pow P
25.3	3.32E-01
23.69	4.06E-02
50.06	3.78E-02



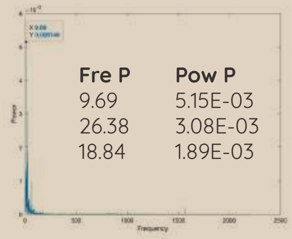
Fre P	Pow P
12.38	3.44E-02
11.3	1.26E-02
15.61	7.80E-03

Sound Analysis: Objects clashing

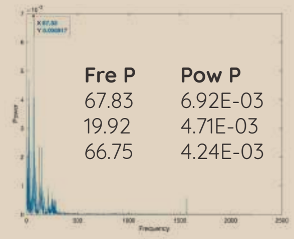
Matlab FFT(Fast Fourier Transform) Analysis with 25 samples



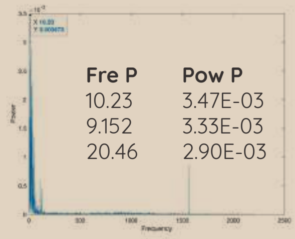
Fre P	Pow P
17.23	1.50E-02
23.15	1.34E-02
20.99	1.33E-02



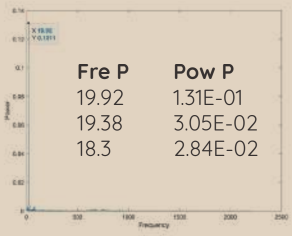
Fre P	Pow P
9.69	5.15E-03
26.38	3.08E-03
18.84	1.89E-03



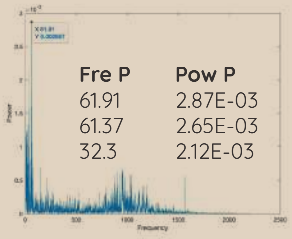
Fre P	Pow P
67.83	6.92E-03
19.92	4.71E-03
66.75	4.24E-03



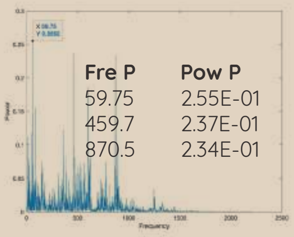
Fre P	Pow P
10.23	3.47E-03
9.152	3.33E-03
20.46	2.90E-03



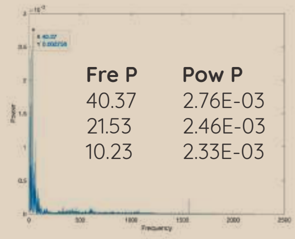
Fre P	Pow P
19.92	1.31E-01
19.38	3.05E-02
18.3	2.84E-02



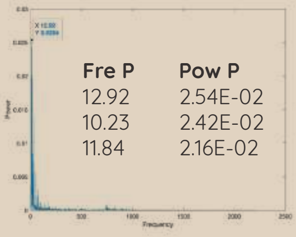
Fre P	Pow P
61.91	2.87E-03
61.37	2.65E-03
32.3	2.12E-03



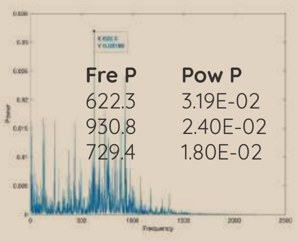
Fre P	Pow P
59.75	2.55E-01
459.7	2.37E-01
870.5	2.34E-01



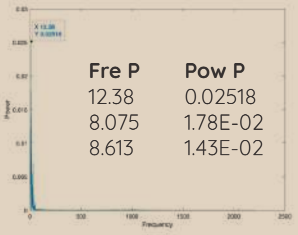
Fre P	Pow P
40.37	2.76E-03
21.53	2.46E-03
10.23	2.33E-03



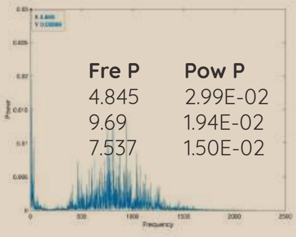
Fre P	Pow P
12.92	2.54E-02
10.23	2.42E-02
11.84	2.16E-02



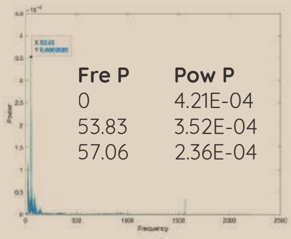
Fre P	Pow P
622.3	3.19E-02
930.8	2.40E-02
729.4	1.80E-02



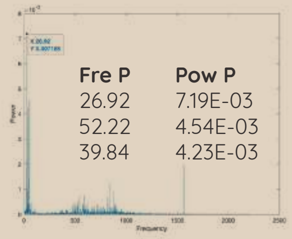
Fre P	Pow P
12.38	0.02518
8.075	1.78E-02
8.613	1.43E-02



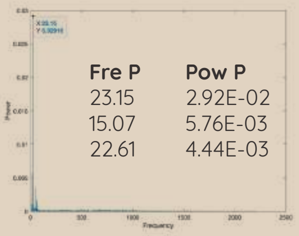
Fre P	Pow P
4.845	2.99E-02
9.69	1.94E-02
7.537	1.50E-02



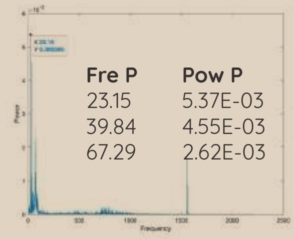
Fre P	Pow P
0	4.21E-04
53.83	3.52E-04
57.06	2.36E-04



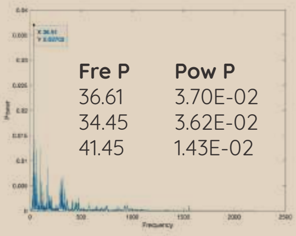
Fre P	Pow P
26.92	7.19E-03
52.22	4.54E-03
39.84	4.23E-03



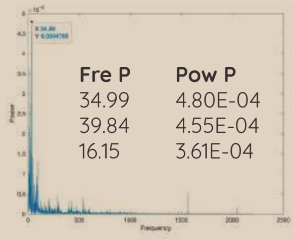
Fre P	Pow P
23.15	2.92E-02
15.07	5.76E-03
22.61	4.44E-03



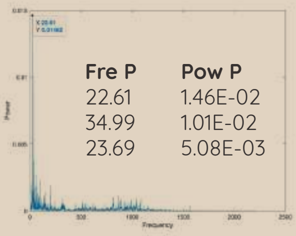
Fre P	Pow P
23.15	5.37E-03
39.84	4.55E-03
67.29	2.62E-03



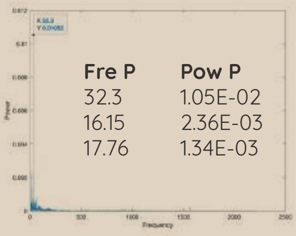
Fre P	Pow P
36.61	3.70E-02
34.45	3.62E-02
41.45	1.43E-02



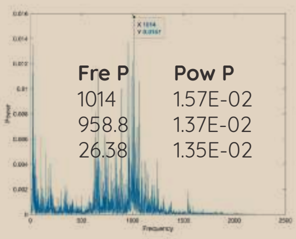
Fre P	Pow P
34.99	4.80E-04
39.84	4.55E-04
16.15	3.61E-04



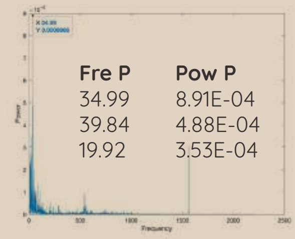
Fre P	Pow P
22.61	1.46E-02
34.99	1.01E-02
23.69	5.08E-03



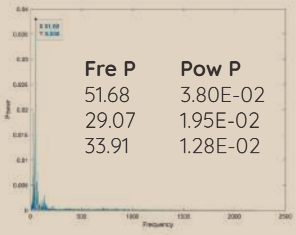
Fre P	Pow P
32.3	1.05E-02
16.15	2.36E-03
17.76	1.34E-03



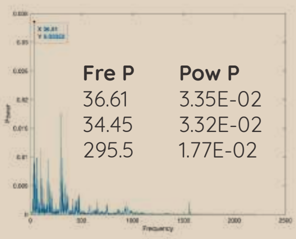
Fre P	Pow P
1014	1.57E-02
958.8	1.37E-02
26.38	1.35E-02



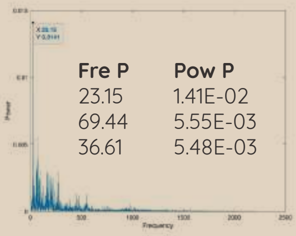
Fre P	Pow P
34.99	8.91E-04
39.84	4.88E-04
19.92	3.53E-04



Fre P	Pow P
51.68	3.80E-02
29.07	1.95E-02
33.91	1.28E-02



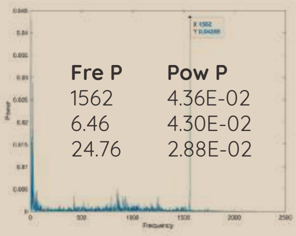
Fre P	Pow P
36.61	3.35E-02
34.45	3.32E-02
295.5	1.77E-02



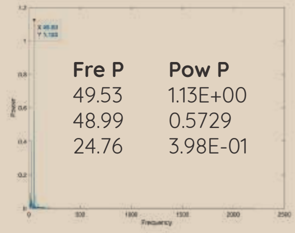
Fre P	Pow P
23.15	1.41E-02
69.44	5.55E-03
36.61	5.48E-03

Sound Analysis: Ventilator

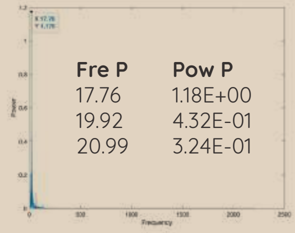
Matlab FFT(Fast Fourier Transform) Analysis with 25 samples



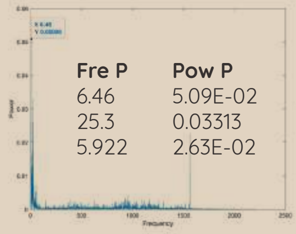
Fre P	Pow P
1562	4.36E-02
6.46	4.30E-02
24.76	2.88E-02



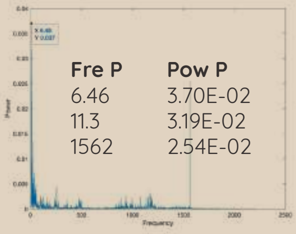
Fre P	Pow P
49.53	1.13E+00
48.99	0.5729
24.76	3.98E-01



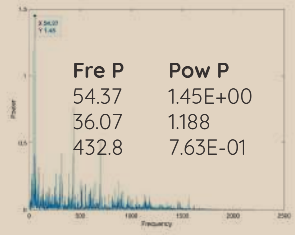
Fre P	Pow P
17.76	1.18E+00
19.92	4.32E-01
20.99	3.24E-01



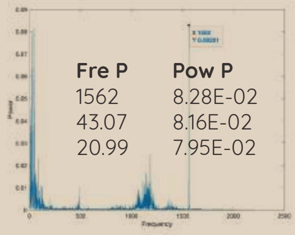
Fre P	Pow P
6.46	5.09E-02
25.3	0.03313
5.922	2.63E-02



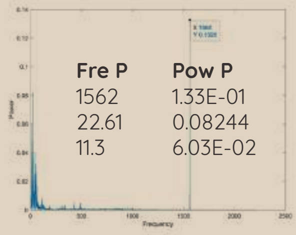
Fre P	Pow P
6.46	3.70E-02
11.3	3.19E-02
1562	2.54E-02



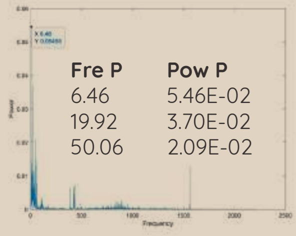
Fre P	Pow P
54.37	1.45E+00
36.07	1.188
432.8	7.63E-01



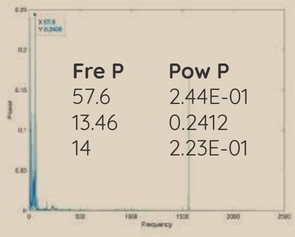
Fre P	Pow P
1562	8.28E-02
43.07	8.16E-02
20.99	7.95E-02



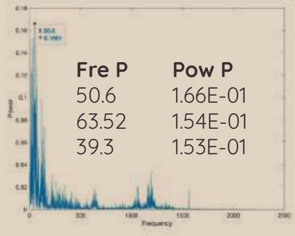
Fre P	Pow P
1562	1.33E-01
22.61	0.08244
11.3	6.03E-02



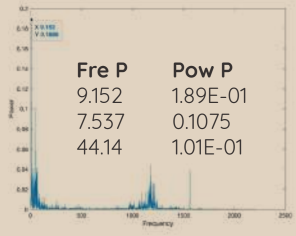
Fre P	Pow P
6.46	5.46E-02
19.92	3.70E-02
50.06	2.09E-02



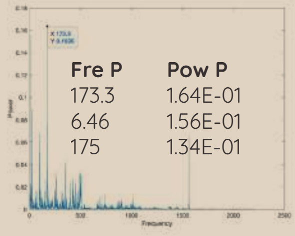
Fre P	Pow P
57.6	2.44E-01
13.46	0.2412
14	2.23E-01



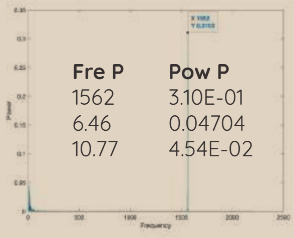
Fre P	Pow P
50.6	1.66E-01
63.52	1.54E-01
39.3	1.53E-01



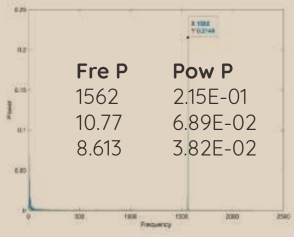
Fre P	Pow P
9.152	1.89E-01
7.537	0.1075
44.14	1.01E-01



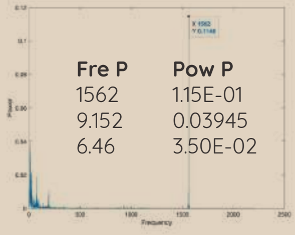
Fre P	Pow P
173.3	1.64E-01
6.46	1.56E-01
175	1.34E-01



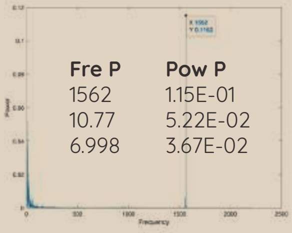
Fre P	Pow P
1562	3.10E-01
6.46	0.04704
10.77	4.54E-02



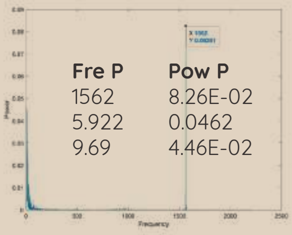
Fre P	Pow P
1562	2.15E-01
10.77	6.89E-02
8.613	3.82E-02



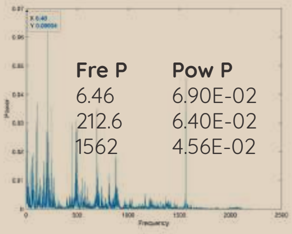
Fre P	Pow P
1562	1.15E-01
9.152	0.03945
6.46	3.50E-02



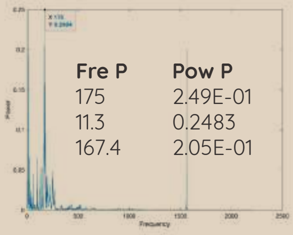
Fre P	Pow P
1562	1.15E-01
10.77	5.22E-02
6.998	3.67E-02



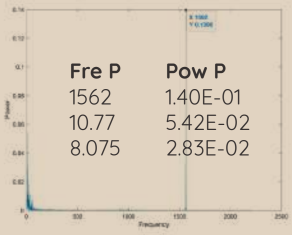
Fre P	Pow P
1562	8.26E-02
5.922	0.0462
9.69	4.46E-02



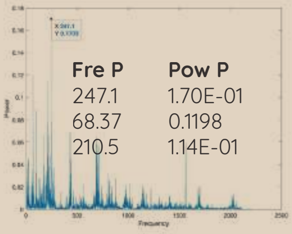
Fre P	Pow P
6.46	6.90E-02
212.6	6.40E-02
1562	4.56E-02



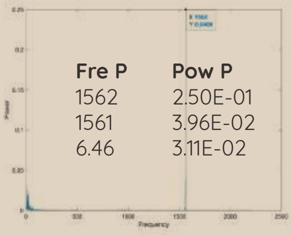
Fre P	Pow P
175	2.49E-01
11.3	0.2483
167.4	2.05E-01



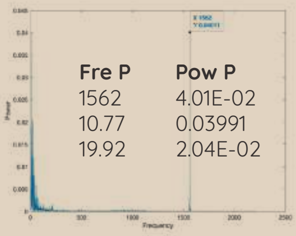
Fre P	Pow P
1562	1.40E-01
10.77	5.42E-02
8.075	2.83E-02



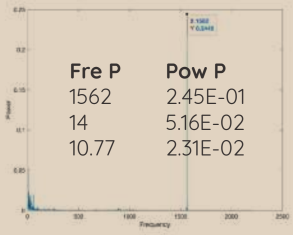
Fre P	Pow P
247.1	1.70E-01
68.37	0.1198
210.5	1.14E-01



Fre P	Pow P
1562	2.50E-01
1561	3.96E-02
6.46	3.11E-02



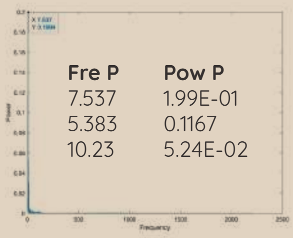
Fre P	Pow P
1562	4.01E-02
10.77	0.03991
19.92	2.04E-02



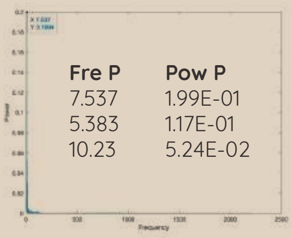
Fre P	Pow P
1562	2.45E-01
14	5.16E-02
10.77	2.31E-02

Sound Analysis: Trolley

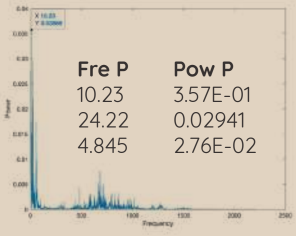
Matlab FFT(Fast Fourier Transform) Analysis with 25 samples



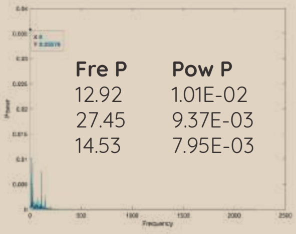
Fre P	Pow P
7.537	1.99E-01
5.383	0.1167
10.23	5.24E-02



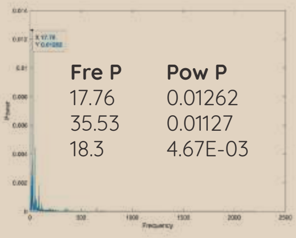
Fre P	Pow P
7.537	1.99E-01
5.383	1.17E-01
10.23	5.24E-02



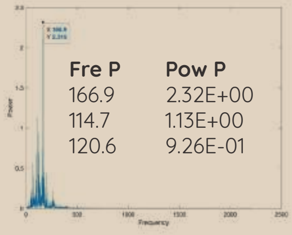
Fre P	Pow P
10.23	3.57E-01
24.22	0.02941
4.845	2.76E-02



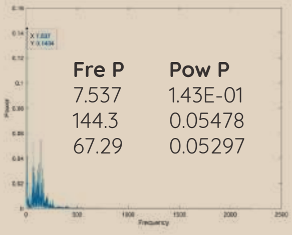
Fre P	Pow P
12.92	1.01E-02
27.45	9.37E-03
14.53	7.95E-03



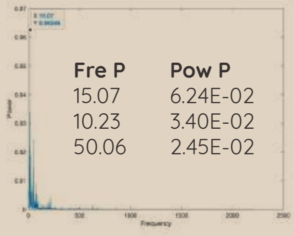
Fre P	Pow P
17.76	0.01262
35.53	0.01127
18.3	4.67E-03



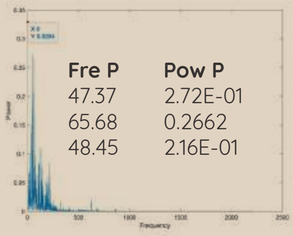
Fre P	Pow P
166.9	2.32E+00
114.7	1.13E+00
120.6	9.26E-01



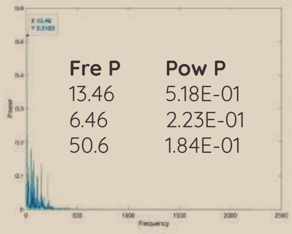
Fre P	Pow P
7.537	1.43E-01
144.3	0.05478
67.29	0.05297



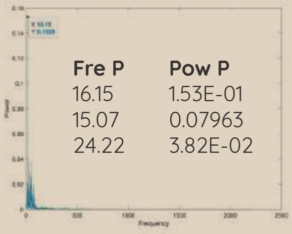
Fre P	Pow P
15.07	6.24E-02
10.23	3.40E-02
50.06	2.45E-02



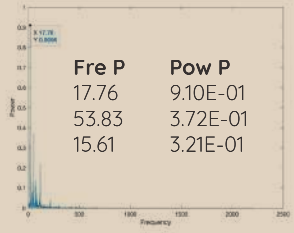
Fre P	Pow P
47.37	2.72E-01
65.68	0.2662
48.45	2.16E-01



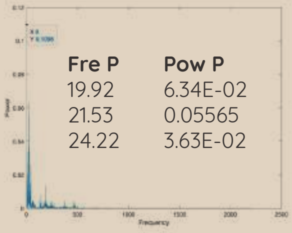
Fre P	Pow P
13.46	5.18E-01
6.46	2.23E-01
50.6	1.84E-01



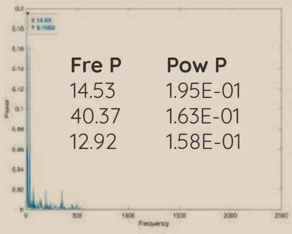
Fre P	Pow P
16.15	1.53E-01
15.07	0.07963
24.22	3.82E-02



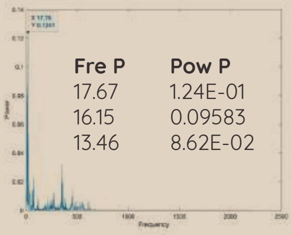
Fre P	Pow P
17.76	9.10E-01
53.83	3.72E-01
15.61	3.21E-01



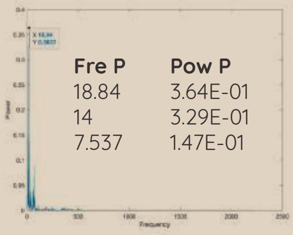
Fre P	Pow P
19.92	6.34E-02
21.53	0.05565
24.22	3.63E-02



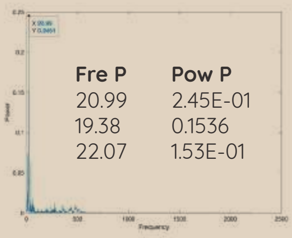
Fre P	Pow P
14.53	1.95E-01
40.37	1.63E-01
12.92	1.58E-01



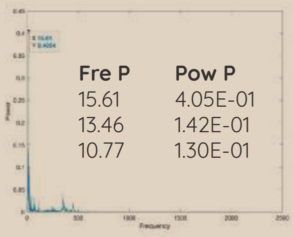
Fre P	Pow P
17.67	1.24E-01
16.15	0.09583
13.46	8.62E-02



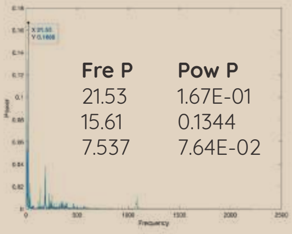
Fre P	Pow P
18.84	3.64E-01
14	3.29E-01
7.537	1.47E-01



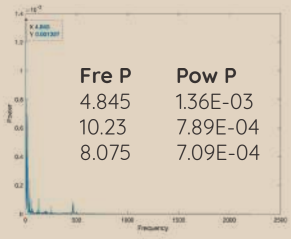
Fre P	Pow P
20.99	2.45E-01
19.38	0.1536
22.07	1.53E-01



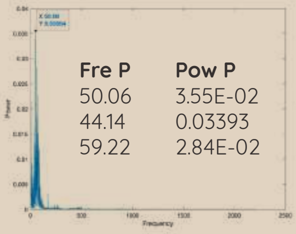
Fre P	Pow P
15.61	4.05E-01
13.46	1.42E-01
10.77	1.30E-01



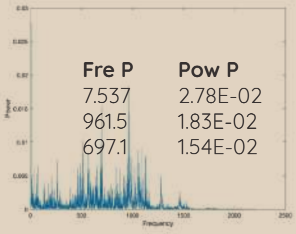
Fre P	Pow P
21.53	1.67E-01
15.61	0.1344
7.537	7.64E-02



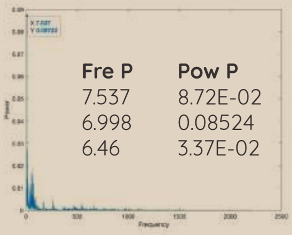
Fre P	Pow P
4.845	1.36E-03
10.23	7.89E-04
8.075	7.09E-04



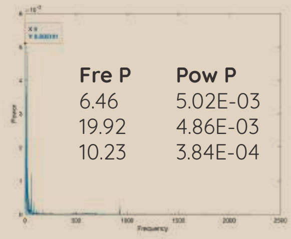
Fre P	Pow P
50.06	3.55E-02
44.14	0.03393
59.22	2.84E-02



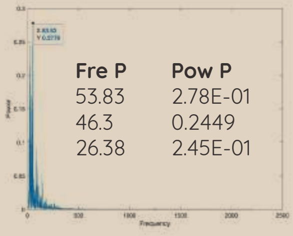
Fre P	Pow P
7.537	2.78E-02
961.5	1.83E-02
697.1	1.54E-02



Fre P	Pow P
7.537	8.72E-02
6.998	0.08524
6.46	3.37E-02



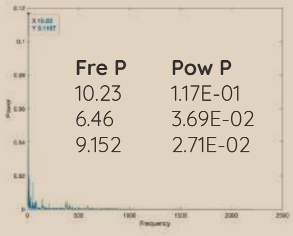
Fre P	Pow P
6.46	5.02E-03
19.92	4.86E-03
10.23	3.84E-04



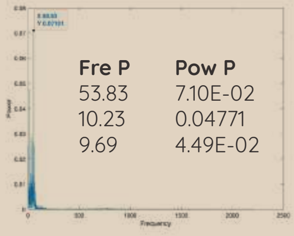
Fre P	Pow P
53.83	2.78E-01
46.3	0.2449
26.38	2.45E-01

Sound Analysis: Door slam

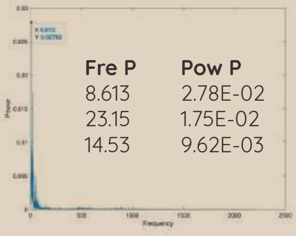
Matlab FFT(Fast Fourier Transform) Analysis with 25 samples



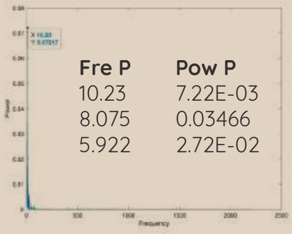
Fre P	Pow P
10.23	1.17E-01
6.46	3.69E-02
9.152	2.71E-02



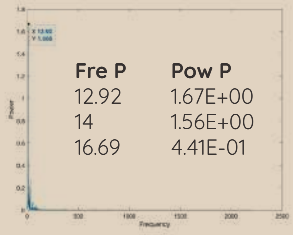
Fre P	Pow P
53.83	7.10E-02
10.23	0.04771
9.69	4.49E-02



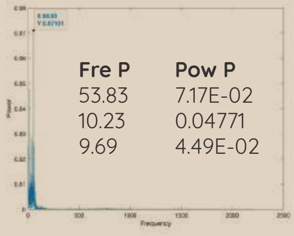
Fre P	Pow P
8.613	2.78E-02
23.15	1.75E-02
14.53	9.62E-03



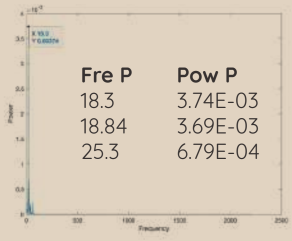
Fre P	Pow P
10.23	7.22E-03
8.075	0.03466
5.922	2.72E-02



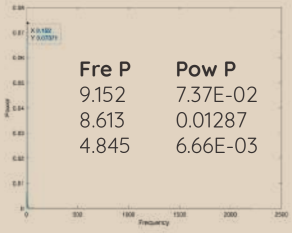
Fre P	Pow P
12.92	1.67E+00
14	1.56E+00
16.69	4.41E-01



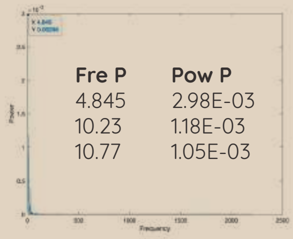
Fre P	Pow P
53.83	7.17E-02
10.23	0.04771
9.69	4.49E-02



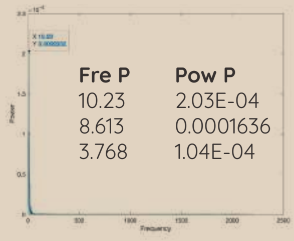
Fre P	Pow P
18.3	3.74E-03
18.84	3.69E-03
25.3	6.79E-04



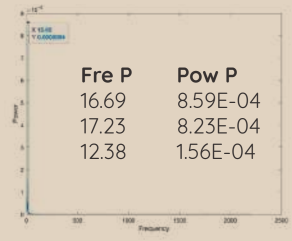
Fre P	Pow P
9.152	7.37E-02
8.613	0.01287
4.845	6.66E-03



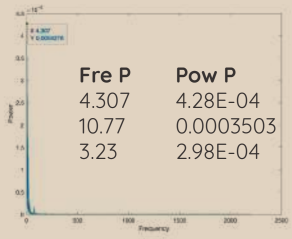
Fre P	Pow P
4.845	2.98E-03
10.23	1.18E-03
10.77	1.05E-03



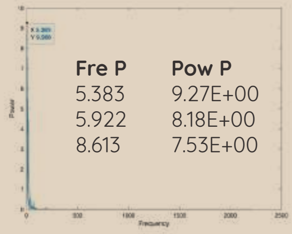
Fre P	Pow P
10.23	2.03E-04
8.613	0.0001636
3.768	1.04E-04



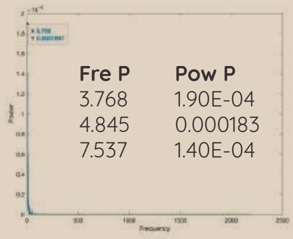
Fre P	Pow P
16.69	8.59E-04
17.23	8.23E-04
12.38	1.56E-04



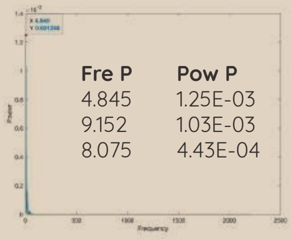
Fre P	Pow P
4.307	4.28E-04
10.77	0.0003503
3.23	2.98E-04



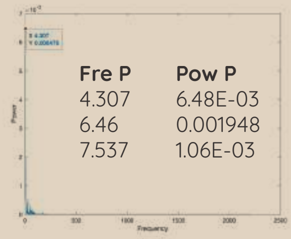
Fre P	Pow P
5.383	9.27E+00
5.922	8.18E+00
8.613	7.53E+00



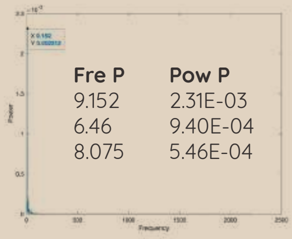
Fre P	Pow P
3.768	1.90E-04
4.845	0.000183
7.537	1.40E-04



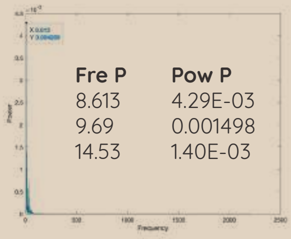
Fre P	Pow P
4.845	1.25E-03
9.152	1.03E-03
8.075	4.43E-04



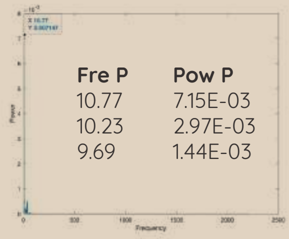
Fre P	Pow P
4.307	6.48E-03
6.46	0.001948
7.537	1.06E-03



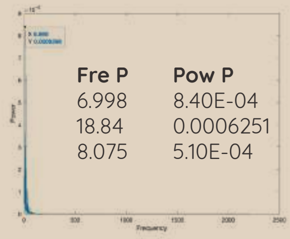
Fre P	Pow P
9.152	2.31E-03
6.46	9.40E-04
8.075	5.46E-04



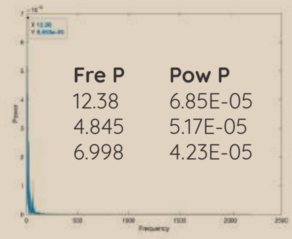
Fre P	Pow P
8.613	4.29E-03
9.69	0.001498
14.53	1.40E-03



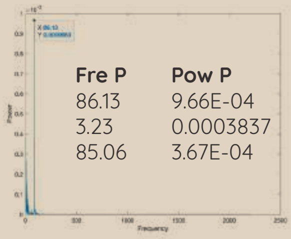
Fre P	Pow P
10.77	7.15E-03
10.23	2.97E-03
9.69	1.44E-03



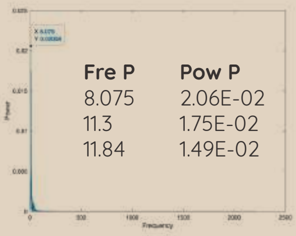
Fre P	Pow P
6.998	8.40E-04
18.84	0.0006251
8.075	5.10E-04



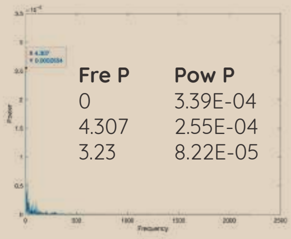
Fre P	Pow P
12.38	6.85E-05
4.845	5.17E-05
6.998	4.23E-05



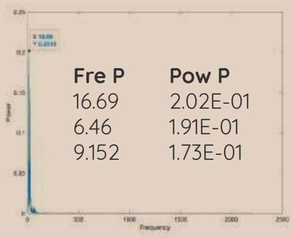
Fre P	Pow P
86.13	9.66E-04
3.23	0.0003837
85.06	3.67E-04



Fre P	Pow P
8.075	2.06E-02
11.3	1.75E-02
11.84	1.49E-02

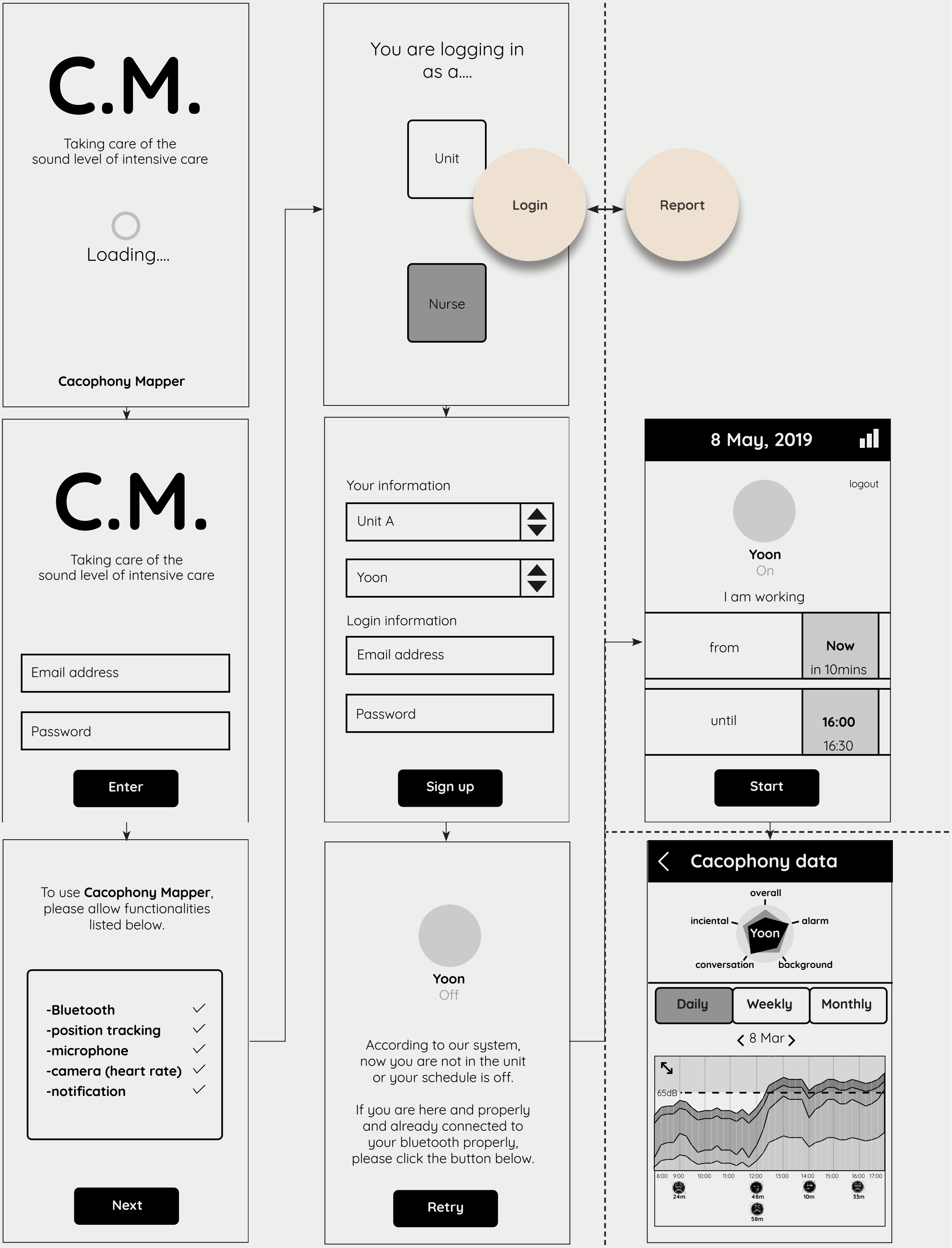


Fre P	Pow P
0	3.39E-04
4.307	2.55E-04
3.23	8.22E-05



Fre P	Pow P
16.69	2.02E-01
6.46	1.91E-01
9.152	1.73E-01

Application structure: nurse mode


C.M.
Taking care of the sound level of intensive care
Loading....
Cacophony Mapper
C.M.
Taking care of the sound level of intensive care
Email address
Password
Enter

To use **Cacophony Mapper**, please allow functionalities listed below.

-Bluetooth ✓
-position tracking ✓
-microphone ✓
-camera (heart rate) ✓
-notification ✓

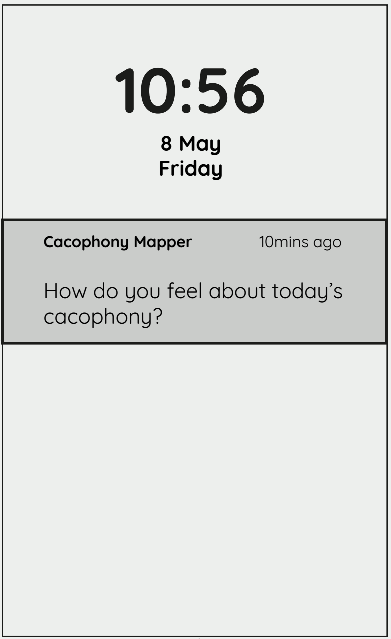
You are logging in as a....

Your information

Login information

According to our system, now you are not in the unit or your schedule is off.

If you are here and properly and already connected to your bluetooth properly, please click the button below.

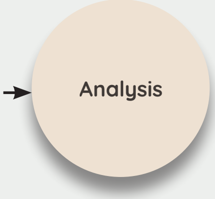


10:56

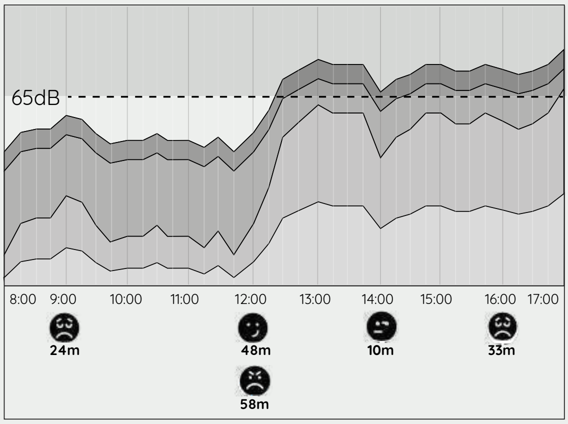
8 May
Friday

Cacophony Mapper 10mins ago

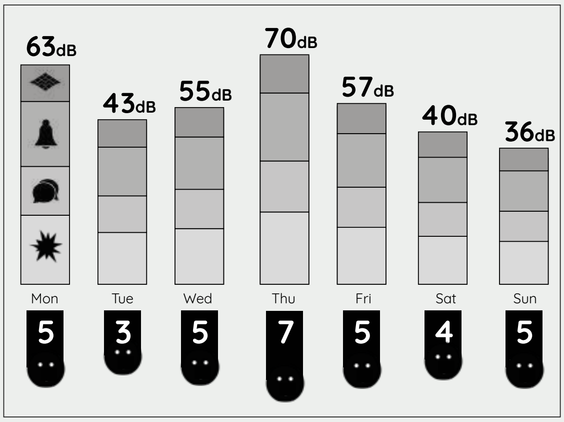
How do you feel about today's cacophony?



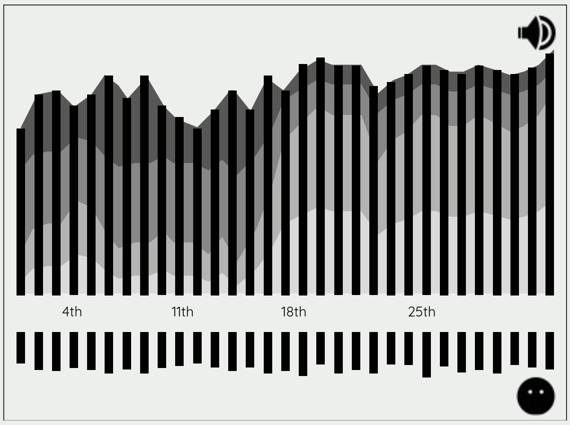
Daily

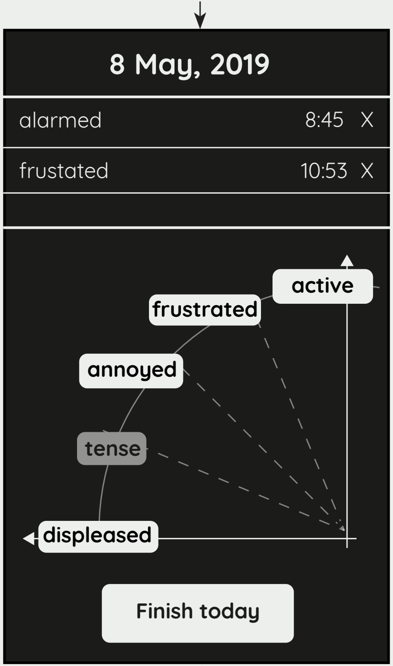


Weekly



Monthly



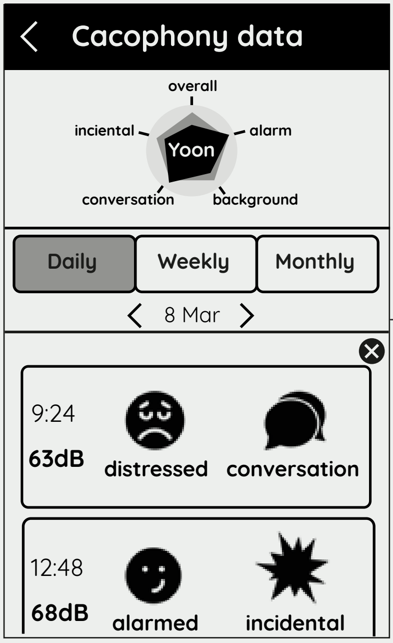


8 May, 2019

alarmed 8:45 X

frustated 10:53 X

Finish today



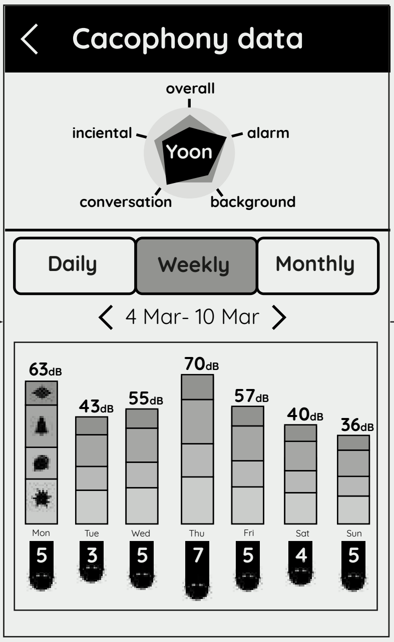
Cacophony data

Daily | Weekly | Monthly

8 Mar

9:24
63dB
distressed
conversation

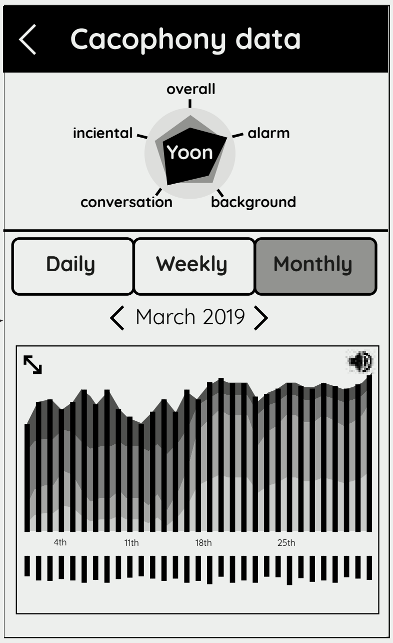
12:48
68dB
alarmed
incidental



Cacophony data

Daily | Weekly | Monthly

4 Mar- 10 Mar

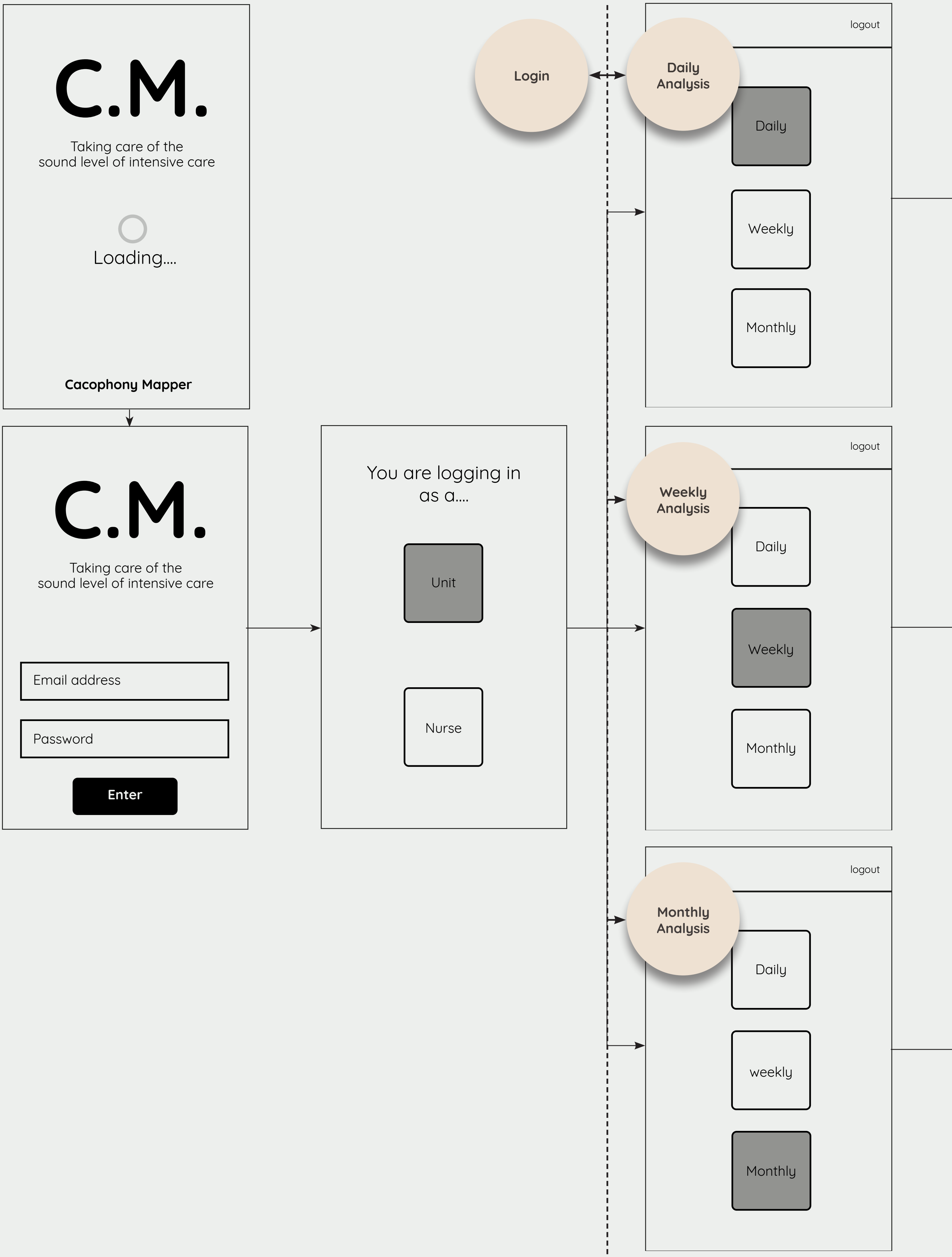


Cacophony data

Daily | Weekly | Monthly

March 2019

Application structure: researcher mode



C.M.

Taking care of the
sound level of intensive care

Loading....

Cacophony Mapper

C.M.

Taking care of the
sound level of intensive care

Email address

Password

Enter

You are logging in
as a....

Unit

Nurse

Login

Daily Analysis

logout

Daily

Weekly

Monthly

Weekly Analysis

logout

Daily

Weekly

Monthly

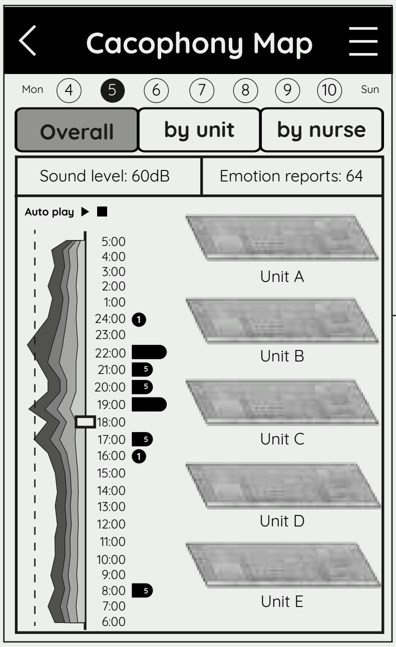
Monthly Analysis

logout

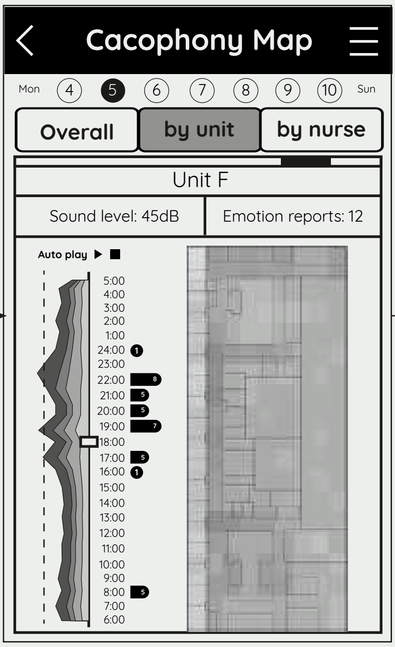
Daily

weekly

Monthly

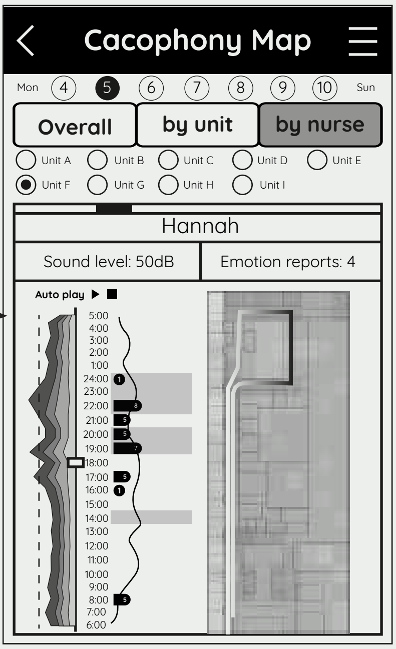


Sound level: 60dB | Emotion reports: 64



Unit F

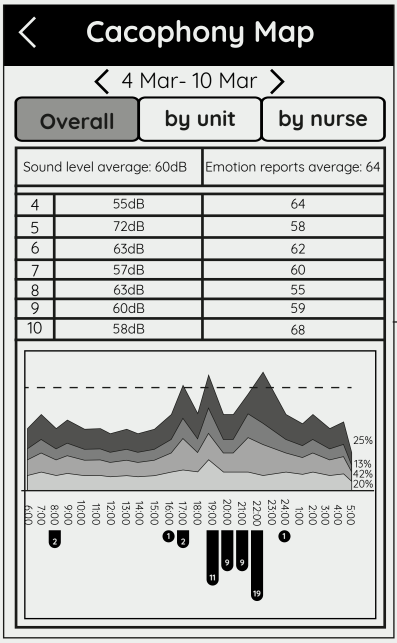
Sound level: 45dB | Emotion reports: 12



○ Unit A ○ Unit B ○ Unit C ○ Unit D ○ Unit E
● Unit F ○ Unit G ○ Unit H ○ Unit I

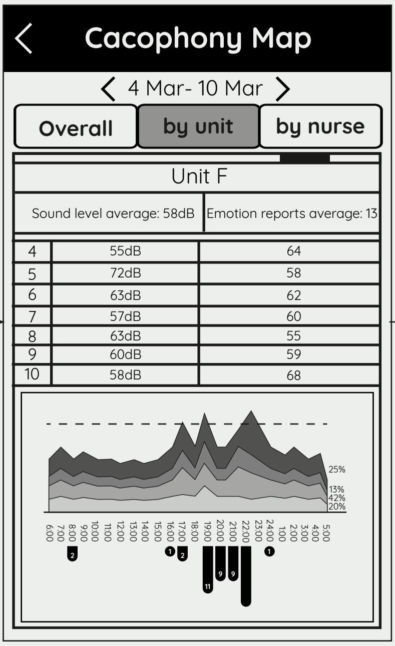
Hannah

Sound level: 50dB | Emotion reports: 4



Sound level average: 60dB | Emotion reports average: 64

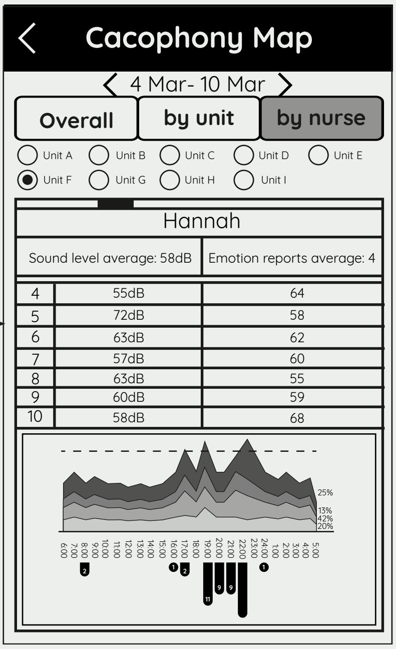
4	55dB	64
5	72dB	58
6	63dB	62
7	57dB	60
8	63dB	55
9	60dB	59
10	58dB	68



Unit F

Sound level average: 58dB | Emotion reports average: 13

4	55dB	64
5	72dB	58
6	63dB	62
7	57dB	60
8	63dB	55
9	60dB	59
10	58dB	68

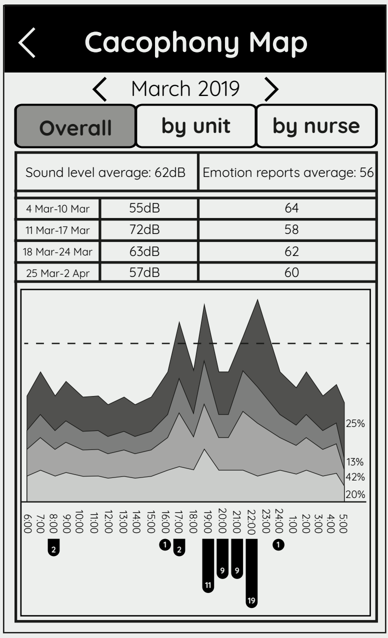


○ Unit A ○ Unit B ○ Unit C ○ Unit D ○ Unit E
● Unit F ○ Unit G ○ Unit H ○ Unit I

Hannah

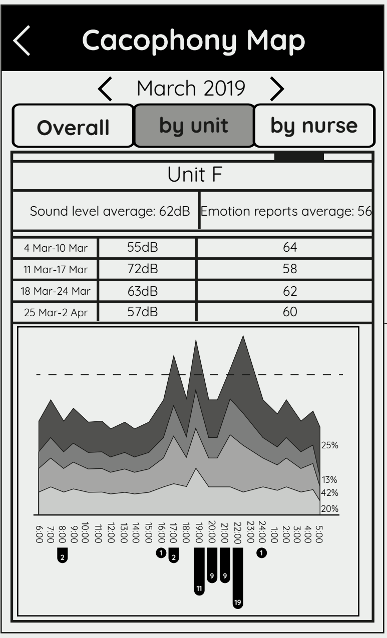
Sound level average: 58dB | Emotion reports average: 4

4	55dB	64
5	72dB	58
6	63dB	62
7	57dB	60
8	63dB	55
9	60dB	59
10	58dB	68



Sound level average: 62dB | Emotion reports average: 56

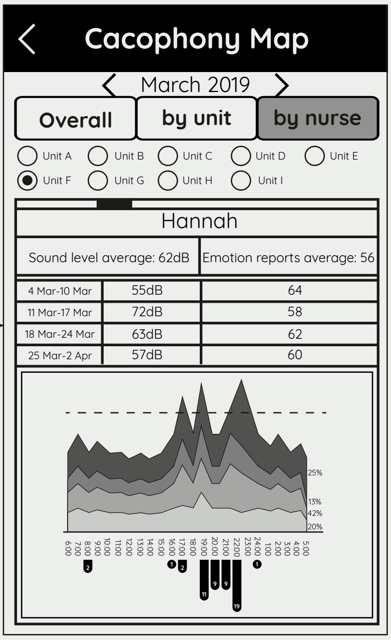
4 Mar-10 Mar	55dB	64
11 Mar-17 Mar	72dB	58
18 Mar-24 Mar	63dB	62
25 Mar-2 Apr	57dB	60



Unit F

Sound level average: 62dB | Emotion reports average: 56

4 Mar-10 Mar	55dB	64
11 Mar-17 Mar	72dB	58
18 Mar-24 Mar	63dB	62
25 Mar-2 Apr	57dB	60



○ Unit A ○ Unit B ○ Unit C ○ Unit D ○ Unit E
● Unit F ○ Unit G ○ Unit H ○ Unit I

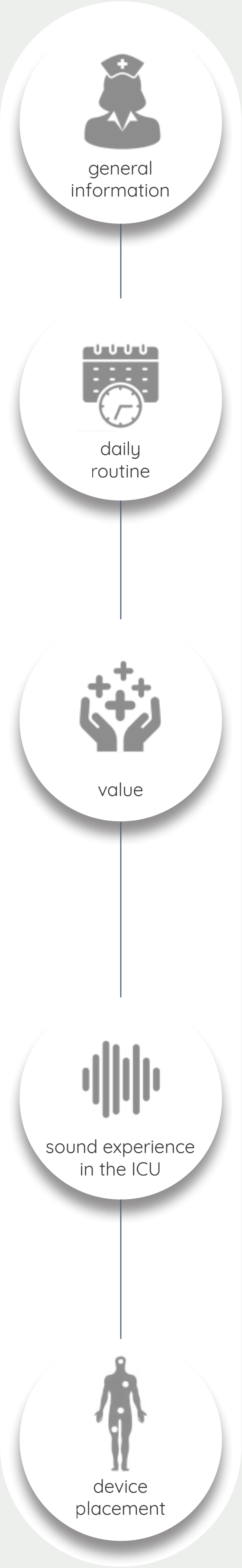
Hannah

Sound level average: 62dB | Emotion reports average: 56

4 Mar-10 Mar	55dB	64
11 Mar-17 Mar	72dB	58
18 Mar-24 Mar	63dB	62
25 Mar-2 Apr	57dB	60

Appendix 4

Questionaire for an interview with a nurse



General Information

1. name?
2. age?
3. work place?
4. working experience?

Daily Routine

1. What time do you start and finish at work? How long do you work a day? How many days do you work a week?
2. What kind of work do you do to take care of your patients?
3. Are you usually busy while you perform your job?
4. Is it stressful to perform your job? What is the main source of your stress?
5. Do you take some break while you are working?

Value

1. How do you feel about working as a nurse in the ICU? How would you define your passion as a nurse?
2. How do you feel about your medical team?
3. How do you feel about your patients?
4. How do you feel about your patient’s family?
5. Are you interested in improving a hospital system? Is there any im provements going on?

Sound experience in the ICU

1. How do you feel about the noise level of operating sound of machines in general?
2. How do you feel about the noise level of alarms from machines?
3. How do you feel about the noise level of general conversation in the ICU? (Clinicians, visitors...)
4. How do you feel about the noise level of incidental sounds in the ICU? (footsteps, door slam, objects clashing, trolley...)
5. Are you aware of the term “Sound fatigue” or “ Noise fatigue”? After listening to the definition and the symptom of the sound, can you come up with similar experiences of your own?

Device placement

1. Are you willing to try new gadgets to improve the sound environment in the ICU?
2. Is there any area that you cannot use your digital gadgets specifically?

Personal Information

Maeijke van dee Tol (23)
Aleijne ziekenhuis Leidendorp
Working experience of 3 years

Hospital Routine

day shift: 7:00-15:30
evening shift: 14:45-23:15
night shift: 22:45-7:15

Day shift starts from 7:00 to 15:30, evening shift starts from 14:45 to 23:15, and night shift starts from 22:45 to 7:15. There are overlapping hours between slots, and nurses usually hand over the patient 's medical records and important notes.

Day shift involves a lot of physical labor than other shifts since nurses need to wash patients and tide them up. Time preference really depends on people since one only goes for the night shifts while someone prefers day shift.

Value

Becoming a nurse used to be my childhood dream. I like to have challenges in my work feel achivements by them.

I love good interactions my patiens and visitors. Some can be annoying but generally, they are nice good people.

Society in the hospital can be highly hierarchical because there are different classes of nurses and doctors.

Sound experience in the ICU

Becoming a nurse used to be my childhood dream. I like to have challenges in my work feel achivements by them. I would like to change the hospital culture positively if there is a chance

Important notes for Cacophony Mapper

“shift changes by 15 minues in day/evening/night hours, so keep that in mind when you design a function with time.”

“Nurses are not allowed to wear something on their arms because of possible infection. Keeping something in chest pocket is totally fine.”

“Changes can be slow in the hospital because no one wants to put an additional burden on their shoulders. However, if they are used to the new system, then they will follow through the new rules and what you need to is to find the way to make nurses get used to the system by making it simple and nice.”

“Some people might prefer to wear the sound collection part on inside the pocket because they want to look tidy. I think you should as people’s preference and apply it to your final design.”

Programming: Java for Android

Complex 1D

```
package nl.dut.ide.software.soundfilterrealgit;
/**
 * Construct a 1-D complex data sequence.
 */
public class Complex1D
{
/**
 * <em>x</em>[<em>i</em>] is the real part of <em>i</em>-th complex data.
 */
  public double x[];
/**
 * <em>y</em>[<em>i</em>] is the imaginary part of <em>i</em>-th complex data.
 */
  public double y[];
}
```

Complex Double FFT

```
package nl.dut.ide.software.soundfilterrealgit;
/**
 * FFT transform of a complex periodic sequence.
 * @author Baoshe Zhang
 * @author Astronomical Instrument Group of University of Lethbridge.
 */
public class ComplexDoubleFFT extends ComplexDoubleFFT_Mixed
{
/**
 * <em>norm_factor</em> can be used to normalize this FFT transform. This is because
 * a call of forward transform (<em>ft</em>) followed by a call of backward transform
 * (<em>bt</em>) will multiply the input sequence by <em>norm_factor</em>.
 */
    public double norm_factor;
    private double wavetable[];
    private int ndim;

/**
 * Construct a wavenumber table with size <em>n</em> for Complex FFT.
 * The sequences with the same size can share a wavenumber table. The prime
 * factorization of <em>n</em> together with a tabulation of the trigonometric functions
 * are computed and stored.
 *
 * @param  n  the size of a complex data sequence. When <em>n</em> is a multiplication of small
 * numbers (4, 2, 3, 5), this FFT transform is very efficient.
 */
    public ComplexDoubleFFT(int n)
    {
        ndim = n;
        norm_factor = n;
        if(wavetable == null || wavetable.length !=(4*ndim+15))
        {
            wavetable = new double[4*ndim + 15];
        }
        cffti(ndim, wavetable);
    }

/**
 * Forward complex FFT transform.
 *
 * @param x  2*<em>n</em> real double data representing <em>n</em> complex double data.
 * As an input parameter, <em>x</em> is an array of 2*<em>n</em> real
 * data representing <em>n</em> complex data. As an output parameter, <em>x</em> represents <em>n</em>
 * FFT'd complex data. Their relation as follows:
 * <br>
 *  x[2*i] is the real part of <em>i</em>-th complex data;
 * <br>
 *  x[2*i+1] is the imaginary part of <em>i</em>-the complex data.
 *
 */
    public void ft(double x[])
    {
       if(x.length != 2*ndim)
           throw new IllegalArgumentException("The length of data can not match that of the wavetable");
       cfftf(ndim, x, wavetable);
    }

/**
 * Forward complex FFT transform.
 *
 * @param x  an array of <em>n</em> Complex data
 */
    public void ft(Complex1D x)
    {
       if(x.x.length != ndim)
           throw new IllegalArgumentException("The length of data can not match that of the wavetable");
       double[] y = new double[2*ndim];
       for(int i=0; i<ndim; i++)
       {
           y[2*i] = x.x[i];
           y[2*i+1] = x.y[i];
       }
       cfftf(ndim, y, wavetable);
       for(int i=0; i<ndim; i++)
       {
           x.x[i]=y[2*i];
           x.y[i]=y[2*i+1];
       }
    }

/**
 * Backward complex FFT transform. It is the unnormalized inverse transform of <em>ft</em>(double[]).
 *
 * @param x  2*<em>n</em> real double data representing <em>n</em> complex double data.
 *
 * As an input parameter, <em>x</em> is an array of 2*<em>n</em>
 * real data representing <em>n</em> complex data. As an output parameter, <em>x</em> represents
 * <em>n</em> FFT'd complex data. Their relation as follows:
 * <br>
 *  x[2*<em>i</em>] is the real part of <em>i</em>-th complex data;
 * <br>
 *  x[2*<em>i</em>+1] is the imaginary part of <em>i</em>-the complex data.
 *
 */
    public void bt(double x[])
    {
       if(x.length != 2*ndim)
           throw new IllegalArgumentException("The length of data can not match that of the wavetable");
        cfftb(ndim, x, wavetable);
    }
/**
 * Backward complex FFT transform. It is the unnormalized inverse transform of <em>ft</em>(Complex1D[]).
 *
 *
 * @param x  an array of <em>n</em> Complex data
 */
    public void bt(Complex1D x)
    {
        if(x.x.length != ndim)
            throw new IllegalArgumentException(“The length of data can not match that of the wavetable”);
        double[] y = new double[2*ndim];
        for(int i=0; i<ndim; i++)
        {
            y[2*i] = x.x[i];
            y[2*i+1] = x.y[i];
        }
        cfftb(ndim, y, wavetable);
        for(int i=0; i<ndim; i++)
        {
            x.x[i]=y[2*i];
            x.y[i]=y[2*i+1];
        }
    }
}
```

Complex Double FFT _Mixed

```
package nl.dut.ide.software.soundfilterrealgit;
/**
 * @author Baoshe Zhang
 * @author Astronomical Instrument Group of University of Lethbridge.
 */
class ComplexDoubleFFT_Mixed
{
/*----------------------------------------------------------------------
  passf2: Complex FFT's forward/backward processing of factor 2;
  isign is +1 for backward and -1 for forward transforms
 ----------------------------------------------------------------------*/

    void passf2(int ido, int l1, final double cc[], double ch[], final double wtable[], int offset, int isign)
                    /*isign==+1 for backward transform*/
    {
        int     i, k, ah, ac;
        double  ti2, tr2;
        int iw1;

        iw1 = offset;
        if(ido<=2)
        {
                for(k=0; k<l1; k++)
                {
                    ah=k*ido;
                    ac=2*k*ido;
                    ch[ah]=cc[ac]+cc[ac+ido];
                    ch[ah+ido*l1]=cc[ac]-cc[ac+ido];
                    ch[ah+1]=cc[ac+1]+cc[ac+ido+1];
                    ch[ah+ido*l1+1]=cc[ac+1]-cc[ac+ido+1];
                }
        }
        else
        {
                for(k=0; k<l1; k++)
                {
                    for(i=0; i<ido-1; i+=2)
                    {
                            ah=i+k*ido;
                            ac=i+2*k*ido;
                            ch[ah]=cc[ac]+cc[ac+ido];
                            tr2=cc[ac]-cc[ac+ido];
                            ch[ah+1]=cc[ac+1]+cc[ac+1+ido];
                            ti2=cc[ac+1]-cc[ac+1+ido];
                            ch[ah+l1*ido+1]=wtable[i+iw1]*ti2+isign*wtable[i+1+iw1]*tr2;
                            ch[ah+l1*ido]=wtable[i+iw1]*tr2-isign*wtable[i+1+iw1]*ti2;
                    }
                }
        }
    }
/*----------------------------------------------------------------------
  passf3: Complex FFT's forward/backward processing of factor 3;
  isign is +1 for backward and -1 for forward transforms
 ----------------------------------------------------------------------*/
    void passf3(int ido, int l1, final double cc[], double ch[], final double wtable[], int offset, int isign)
    {
        final double taur=-0.5;
        final double taui=0.866025403784439;
        int     i, k, ac, ah;
        double  ci2, ci3, di2, di3, cr2, cr3, dr2, dr3, ti2, tr2;
        int iw1, iw2;

        iw1 = offset;
        iw2 = iw1 + ido;

        if(ido==2)
        {
                for(k=1; k<=l1; k++)
                {
                    ac=(3*k-2)*ido;
                    tr2=cc[ac]+cc[ac+ido];
                    cr2=cc[ac-ido]+taur*tr2;
                    ah=(k-1)*ido;
                    ch[ah]=cc[ac-ido]+tr2;

                    ti2=cc[ac+1]+cc[ac+ido+1];
                    ci2=cc[ac-ido+1]+taur*ti2;
                    ch[ah+1]=cc[ac-ido+1]+ti2;

                    cr3=isign*taui*(cc[ac]-cc[ac+ido]);
                    ci3=isign*taui*(cc[ac+1]-cc[ac+ido+1]);
                    ch[ah+l1*ido]=cr2-ci3;
                    ch[ah+2*l1*ido]=cr2+ci3;
                    ch[ah+l1*ido+1]=ci2+cr3;
                    ch[ah+2*l1*ido+1]=ci2-cr3;
                }
        }
        else
        {
                for(k=1; k<=l1; k++)
                {
                    for(i=0; i<ido-1; i+=2)
                    {
```

```
                                    ac=i+(3*k-2)*ido;
                                    tr2=cc[ac]+cc[ac+ido];
                                    cr2=cc[ac-ido]+taur*tr2;
                                    ah=i+(k-1)*ido;
                                    ch[ah]=cc[ac-ido]+tr2;
                                    ti2=cc[ac+1]+cc[ac+ido+1];
                                    ci2=cc[ac-ido+1]+taur*ti2;
                                    ch[ah+1]=cc[ac-ido+1]+ti2;
                                    cr3=isign*taui*(cc[ac]-cc[ac+ido]);
                                    ci3=isign*taui*(cc[ac+1]-cc[ac+ido+1]);
                                    dr2=cr2-ci3;
                                    dr3=cr2+ci3;
                                    di2=ci2+cr3;
                                    di3=ci2-cr3;
                                    ch[ah+l1*ido+1]=wtable[i+iw1]*di2+isign*wtable[i+1+iw1]*dr2;
                                    ch[ah+l1*ido]=wtable[i+iw1]*dr2-isign*wtable[i+1+iw1]*di2;
                                    ch[ah+2*l1*ido+1]=wtable[i+iw2]*di3+isign*wtable[i+1+iw2]*dr3;
                                    ch[ah+2*l1*ido]=wtable[i+iw2]*dr3-isign*wtable[i+1+iw2]*di3;
                        }
                    }
        }
    }

/*----------------------------------------------------------------------
  passf4: Complex FFT's forward/backward processing of factor 4;
  isign is +1 for backward and -1 for forward transforms
  ----------------------------------------------------------------------*/
    void passf4(int ido, int l1, final double cc[], double ch[], final double wtable[], int offset, int isign)
    {
        int i, k, ac, ah;
        double  ci2, ci3, ci4, cr2, cr3, cr4, ti1, ti2, ti3, ti4, tr1, tr2, tr3, tr4;
        int iw1, iw2, iw3;
        iw1 = offset;
        iw2 = iw1 + ido;
        iw3 = iw2 + ido;

        if(ido==2)
        {
                    for(k=0; k<l1; k++)
                    {
                        ac=4*k*ido+1;
                        ti1=cc[ac]-cc[ac+2*ido];
                        ti2=cc[ac]+cc[ac+2*ido];
                                tr4=cc[ac+3*ido]-cc[ac+ido];
                                ti3=cc[ac+ido]+cc[ac+3*ido];
                                tr1=cc[ac-1]-cc[ac+2*ido-1];
                        tr2=cc[ac-1]+cc[ac+2*ido-1];
                                ti4=cc[ac+ido-1]-cc[ac+3*ido-1];
                        tr3=cc[ac+ido-1]+cc[ac+3*ido-1];
                                ah=k*ido;
                                ch[ah]=tr2+tr3;
                                ch[ah+2*l1*ido]=tr2-tr3;
                                ch[ah+1]=ti2+ti3;
                                ch[ah+2*l1*ido+1]=ti2-ti3;
                                ch[ah+l1*ido]=tr1+isign*tr4;
                                ch[ah+3*l1*ido]=tr1-isign*tr4;
                                ch[ah+l1*ido+1]=ti1+isign*ti4;
                                ch[ah+3*l1*ido+1]=ti1-isign*ti4;
                    }
        }
        else
        {
                    for(k=0; k<l1; k++)
                            {
                                for(i=0; i<ido-1; i+=2)
                                {
                                        ac=i+1+4*k*ido;
                                        ti1=cc[ac]-cc[ac+2*ido];
                                        ti2=cc[ac]+cc[ac+2*ido];
                                        ti3=cc[ac+ido]+cc[ac+3*ido];
                                        tr4=cc[ac+3*ido]-cc[ac+ido];
                                        tr1=cc[ac-1]-cc[ac+2*ido-1];
                                        tr2=cc[ac-1]+cc[ac+2*ido-1];
                                        ti4=cc[ac+ido-1]-cc[ac+3*ido-1];
                                        tr3=cc[ac+ido-1]+cc[ac+3*ido-1];
                                        ah=i+k*ido;
                                        ch[ah]=tr2+tr3;
                                        cr3=tr2-tr3;
                                        ch[ah+1]=ti2+ti3;
                                        ci3=ti2-ti3;
                                        cr2=tr1+isign*tr4;
                                        cr4=tr1-isign*tr4;
                                        ci2=ti1+isign*ti4;
                                        ci4=ti1-isign*ti4;
                                        ch[ah+l1*ido]=wtable[i+iw1]*cr2-isign*wtable[i+1+iw1]*ci2;
                                        ch[ah+l1*ido+1]=wtable[i+iw1]*ci2+isign*wtable[i+1+iw1]*cr2;
                                        ch[ah+2*l1*ido]=wtable[i+iw2]*cr3-isign*wtable[i+1+iw2]*ci3;
                                        ch[ah+2*l1*ido+1]=wtable[i+iw2]*ci3+isign*wtable[i+1+iw2]*cr3;
                                        ch[ah+3*l1*ido]=wtable[i+iw3]*cr4-isign*wtable[i+1+iw3]*ci4;
                                        ch[ah+3*l1*ido+1]=wtable[i+iw3]*ci4+isign*wtable[i+1+iw3]*cr4;
                                }
                            }
                    }
    }

/*----------------------------------------------------------------------
  passf5: Complex FFT's forward/backward processing of factor 5;
  isign is +1 for backward and -1 for forward transforms
  ----------------------------------------------------------------------*/
    void passf5(int ido, int l1, final double cc[], double ch[], final double wtable[], int offset, int isign)
              /*isign==-1 for forward transform and+1 for backward transform*/
    {
                final double tr11=0.309016994374947;
                final double ti11=0.951056516295154;
                final double tr12=-0.809016994374947;
                final double ti12=0.587785252292473;
                int     i, k, ac, ah;
                double  ci2, ci3, ci4, ci5, di3, di4, di5, di2, cr2, cr3, cr5, cr4,
                    ti2, ti3, ti4, ti5, dr3, dr4, dr5, dr2, tr2, tr3, tr4, tr5;
      int iw1, iw2, iw3, iw4;

      iw1 = offset;
      iw2 = iw1 + ido;
      iw3 = iw2 + ido;
      iw4 = iw3 + ido;

                if(ido==2)
                {
                    for(k=1; k<=l1;++k)
                    {
                            ac=(5*k-4)*ido+1;
                            ti5=cc[ac]-cc[ac+3*ido];
                            ti2=cc[ac]+cc[ac+3*ido];
                            ti4=cc[ac+ido]-cc[ac+2*ido];
                            ti3=cc[ac+ido]+cc[ac+2*ido];
                            tr5=cc[ac-1]-cc[ac+3*ido-1];
                            tr2=cc[ac-1]+cc[ac+3*ido-1];
                tr4=cc[ac+ido-1]-cc[ac+2*ido-1];
                tr3=cc[ac+ido-1]+cc[ac+2*ido-1];
                ah=(k-1)*ido;
                ch[ah]=cc[ac-ido-1]+tr2+tr3;
                ch[ah+1]=cc[ac-ido]+ti2+ti3;
                cr2=cc[ac-ido-1]+tr11*tr2+tr12*tr3;
                ci2=cc[ac-ido]+tr11*ti2+tr12*ti3;
                cr3=cc[ac-ido-1]+tr12*tr2+tr11*tr3;
                ci3=cc[ac-ido]+tr12*ti2+tr11*ti3;
                cr5=isign*(ti11*tr5+ti12*tr4);
                ci5=isign*(ti11*ti5+ti12*ti4);
                cr4=isign*(ti12*tr5-ti11*tr4);
                ci4=isign*(ti12*ti5-ti11*ti4);
                ch[ah+l1*ido]=cr2-ci5;
                ch[ah+4*l1*ido]=cr2+ci5;
                ch[ah+l1*ido+1]=ci2+cr5;
                ch[ah+2*l1*ido+1]=ci3+cr4;
                ch[ah+2*l1*ido]=cr3-ci4;
                ch[ah+3*l1*ido]=cr3+ci4;
                ch[ah+3*l1*ido+1]=ci3-cr4;
                ch[ah+4*l1*ido+1]=ci2-cr5;
          }
        }
        else
        {
          for(k=1; k<=l1; k++)
          {
                for(i=0; i<ido-1; i+=2)
                {
                        ac=i+1+(k*5-4)*ido;
                        ti5=cc[ac]-cc[ac+3*ido];
                        ti2=cc[ac]+cc[ac+3*ido];
                        ti4=cc[ac+ido]-cc[ac+2*ido];
                        ti3=cc[ac+ido]+cc[ac+2*ido];
                        tr5=cc[ac-1]-cc[ac+3*ido-1];
                        tr2=cc[ac-1]+cc[ac+3*ido-1];
                        tr4=cc[ac+ido-1]-cc[ac+2*ido-1];
                        tr3=cc[ac+ido-1]+cc[ac+2*ido-1];
                        ah=i+(k-1)*ido;
                        ch[ah]=cc[ac-ido-1]+tr2+tr3;
                        ch[ah+1]=cc[ac-ido]+ti2+ti3;
                        cr2=cc[ac-ido-1]+tr11*tr2+tr12*tr3;

                        ci2=cc[ac-ido]+tr11*ti2+tr12*ti3;
                        cr3=cc[ac-ido-1]+tr12*tr2+tr11*tr3;

                        ci3=cc[ac-ido]+tr12*ti2+tr11*ti3;
                        cr5=isign*(ti11*tr5+ti12*tr4);
                        ci5=isign*(ti11*ti5+ti12*ti4);
                        cr4=isign*(ti12*tr5-ti11*tr4);
                        ci4=isign*(ti12*ti5-ti11*ti4);
                        dr3=cr3-ci4;
                        dr4=cr3+ci4;
                        di3=ci3+cr4;
                        di4=ci3-cr4;
                        dr5=cr2+ci5;
                        dr2=cr2-ci5;
                        di5=ci2-cr5;
                        di2=ci2+cr5;
                        ch[ah+l1*ido]=wtable[i+iw1]*dr2-isign*wtable[i+1+iw1]*di2;
                        ch[ah+l1*ido+1]=wtable[i+iw1]*di2+isign*wtable[i+1+iw1]*dr2;
                        ch[ah+2*l1*ido]=wtable[i+iw2]*dr3-isign*wtable[i+1+iw2]*di3;
                        ch[ah+2*l1*ido+1]=wtable[i+iw2]*di3+isign*wtable[i+1+iw2]*dr3;
                        ch[ah+3*l1*ido]=wtable[i+iw3]*dr4-isign*wtable[i+1+iw3]*di4;
                        ch[ah+3*l1*ido+1]=wtable[i+iw3]*di4+isign*wtable[i+1+iw3]*dr4;
                        ch[ah+4*l1*ido]=wtable[i+iw4]*dr5-isign*wtable[i+1+iw4]*di5;
                        ch[ah+4*l1*ido+1]=wtable[i+iw4]*di5+isign*wtable[i+1+iw4]*dr5;
                }
          }
     }
  }

/*------------------------------------------------------------------
 passfg: Complex FFT's forward/backward processing of general factor;
 isign is +1 for backward and -1 for forward transforms
 ------------------------------------------------------------------*/
  void passfg(int nac[], int ido, int ip, int l1, int idl1,
           final double cc[], double c1[], double c2[], double ch[], double ch2[],
           final double wtable[], int offset, int isign)
  {
     int idij, idlj, idot, ipph, i, j, k, l, jc, lc, ik, nt, idj, idl, inc, idp;
     double  wai, war;
     int iw1;

     iw1 = offset;
         idot=ido / 2;
     nt=ip*idl1;
     ipph=(ip+1)/ 2;
     idp=ip*ido;
     if(ido>=l1)
     {
           for(j=1; j<ipph; j++)
           {
             jc=ip-j;
             for(k=0; k<l1; k++)
             {
                    for(i=0; i<ido; i++)
                    {
                      ch[i+(k+j*l1)*ido]=cc[i+(j+k*ip)*ido]+cc[i+(jc+k*ip)*ido];
                      ch[i+(k+jc*l1)*ido]=cc[i+(j+k*ip)*ido]-cc[i+(jc+k*ip)*ido];
                    }
             }
           }
           for(k=0; k<l1; k++)
             for(i=0; i<ido; i++)
                   ch[i+k*ido]=cc[i+k*ip*ido];
     }
     else
     {
           for(j=1; j<ipph; j++)
           {
             jc=ip-j;
             for(i=0; i<ido; i++)
             {
                   for(k=0; k<l1; k++)
                   {
                     ch[i+(k+j*l1)*ido]=cc[i+(j+k*ip)*ido]+cc[i+(jc+k*ip)*ido];
                     ch[i+(k+jc*l1)*ido]=cc[i+(j+k*ip)*ido]-cc[i+(jc+k*ip)*ido];
                   }
             }
           }
           for(i=0; i<ido; i++)
             for(k=0; k<l1; k++)
                   ch[i+k*ido]=cc[i+k*ip*ido];
     }

     idl=2-ido;
```

```
        inc=0;
        for(l=1; l<ipph; l++)
        {
                lc=ip-l;
                idl+=ido;
                for(ik=0; ik<idl1; ik++)
                {
                    c2[ik+l*idl1]=ch2[ik]+wtable[idl-2+iw1]*ch2[ik+idl1];
                    c2[ik+lc*idl1]=isign*wtable[idl-1+iw1]*ch2[ik+(ip-1)*idl1];
                }
                idlj=idl;
                inc+=ido;
                for(j=2; j<ipph; j++)
                {
                    jc=ip-j;
                    idlj+=inc;
                    if(idlj>idp) idlj-=idp;
                    war=wtable[idlj-2+iw1];
                    wai=wtable[idlj-1+iw1];
                    for(ik=0; ik<idl1; ik++)
                    {
                            c2[ik+l*idl1]+=war*ch2[ik+j*idl1];
                            c2[ik+lc*idl1]+=isign*wai*ch2[ik+jc*idl1];
                    }
                }
        }
        for(j=1; j<ipph; j++)
               for(ik=0; ik<idl1; ik++)
                   ch2[ik]+=ch2[ik+j*idl1];
        for(j=1; j<ipph; j++)
        {
                jc=ip-j;
                for(ik=1; ik<idl1; ik+=2)
                {
                    ch2[ik-1+j*idl1]=c2[ik-1+j*idl1]-c2[ik+jc*idl1];
                    ch2[ik-1+jc*idl1]=c2[ik-1+j*idl1]+c2[ik+jc*idl1];
                    ch2[ik+j*idl1]=c2[ik+j*idl1]+c2[ik-1+jc*idl1];
                    ch2[ik+jc*idl1]=c2[ik+j*idl1]-c2[ik-1+jc*idl1];
                }
        }
        nac[0]=1;
        if(ido==2) return;
        nac[0]=0;
        for(ik=0; ik<idl1; ik++) c2[ik]=ch2[ik];
        for(j=1; j<ip; j++)
        {
                for(k=0; k<l1; k++)
                {
                    c1[(k+j*l1)*ido+0]=ch[(k+j*l1)*ido+0];
                    c1[(k+j*l1)*ido+1]=ch[(k+j*l1)*ido+1];
                }
        }
        if(idot<=l1)
        {
                idij=0;
                for(j=1; j<ip; j++)
                {
                    idij+=2;
                    for(i=3; i<ido; i+=2)
                    {
                            idij+=2;
                            for(k=0; k<l1; k++)
                            {
                                c1[i-1+(k+j*l1)*ido]=
                                        wtable[idij-2+iw1]*ch[i-1+(k+j*l1)*ido]-
                                        isign*wtable[idij-1+iw1]*ch[i+(k+j*l1)*ido];
                                c1[i+(k+j*l1)*ido]=
                                        wtable[idij-2+iw1]*ch[i+(k+j*l1)*ido]+
                                        isign*wtable[idij-1+iw1]*ch[i-1+(k+j*l1)*ido];
                            }
                    }
                }
        }
        else
        {
                idj=2-ido;
                for(j=1; j<ip; j++)
                {
                    idj+=ido;
                    for(k=0; k<l1; k++)
                    {
                            idij=idj;
                            for(i=3; i<ido; i+=2)
                            {
                                idij+=2;
                                c1[i-1+(k+j*l1)*ido]=
                                        wtable[idij-2+iw1]*ch[i-1+(k+j*l1)*ido]-
                                        isign*wtable[idij-1+iw1]*ch[i+(k+j*l1)*ido];
                                c1[i+(k+j*l1)*ido]=
                                        wtable[idij-2+iw1]*ch[i+(k+j*l1)*ido]+
                                        isign*wtable[idij-1+iw1]*ch[i-1+(k+j*l1)*ido];
                            }
                    }
                }
        }
      }

/*---------------------------------------------------------
  cfftf1: further processing of Complex forward FFT
 --------------------------------------------------------*/
     void cfftf1(int n, double c[], final double wtable[], int isign)
     {
        int     idot, i;
        int     k1, l1, l2;
        int     na, nf, ip, iw, ido, idl1;
        int[]  nac = new int[1];

        int     iw1, iw2;
        double[] ch = new double[2*n];

        iw1=2*n;
        iw2=4*n;
        System.arraycopy(wtable, 0, ch, 0, 2*n);

        nac[0] = 0;

        nf=(int)wtable[1+iw2];
        na=0;
        l1=1;
        iw=iw1;
        for(k1=2; k1<=nf+1; k1++)
        {
                ip=(int)wtable[k1+iw2];
                l2=ip*l1;
                ido=n / l2;
                idot=ido+ido;
                idl1=idot*l1;
```

```
           if(ip==4)
           {
             if(na==0)
         {
           passf4(idot, l1, c, ch, wtable, iw, isign);
         }
             else
         {
           passf4(idot, l1, ch, c, wtable, iw, isign);
         }
             na=1-na;
           }
           else if(ip==2)
           {
             if(na==0)
         {
             passf2(idot, l1, c, ch, wtable, iw, isign);
         }
             else
         {
             passf2(idot, l1, ch, c, wtable, iw, isign);
         }
             na=1-na;
           }
           else if(ip==3)
           {
             if(na==0)
         {
            passf3(idot, l1, c, ch, wtable, iw, isign);
         }
             else
         {
            passf3(idot, l1, ch, c, wtable, iw, isign);
         }
             na=1-na;
           }
           else if(ip==5)
           {
             if(na==0)
         {
            passf5(idot, l1, c, ch, wtable, iw, isign);
         }
             else
         {
            passf5(idot, l1, ch, c, wtable, iw, isign);
         }
             na=1-na;
           }
           else
           {
             if(na==0)
         {
            passfg(nac, idot, ip, l1, idl1, c, c, c, ch, ch, wtable, iw, isign);
         }
             else
         {
            passfg(nac, idot, ip, l1, idl1, ch, ch, ch, c, c, wtable, iw, isign);
         }
             if(nac[0] !=0) na=1-na;
           }
           l1=l2;
           iw+=(ip-1)*idot;
     }
     if(na==0) return;
     for(i=0; i<2*n; i++) c[i]=ch[i];
  }
/*---------------------------------------------------------
 cfftf: Complex forward FFT
 --------------------------------------------------------*/
  void cfftf(int n, double c[], double wtable[])
  {
     cfftf1(n, c, wtable, -1);
  }
/*---------------------------------------------------------
 cfftb: Complex borward FFT
 --------------------------------------------------------*/
  void cfftb(int n, double c[], double wtable[])
  {
     cfftf1(n, c, wtable, +1);
  }
/*---------------------------------------------------------
 cffti1: further initialization of Complex FFT
 --------------------------------------------------------*/
  void cffti1(int n, double wtable[])
  {

     final int[] ntryh = {3, 4, 2, 5};
     final double twopi=2.0D*Math.PI;
     double  argh;
     int     idot, ntry=0, i, j;
     double  argld;
     int     i1, k1, l1, l2, ib;
     double  fi;
     int     ld, ii, nf, ip, nl, nq, nr;
     double  arg;
     int     ido, ipm;

      nl=n;
      nf=0;
      j=0;

   factorize_loop:
      while(true)
      {
        j++;
        if(j<=4)
              ntry=ntryh[j-1];
        else
              ntry+=2;
        do
        {
              nq=nl / ntry;
              nr=nl-ntry*nq;
              if(nr !=0) continue factorize_loop;
              nf++;
              wtable[nf+1+4*n]=ntry;
              nl=nq;
              if(ntry==2 && nf !=1)
              {
                for(i=2; i<=nf; i++)
                {
                      ib=nf-i+2;
                      wtable[ib+1+4*n]=wtable[ib+4*n];
                }
```

```
                        wtable[2+4*n]=2;
                    }
            } while(nl !=1);
            break factorize_loop;
        }
        wtable[0+4*n]=n;
        wtable[1+4*n]=nf;
        argh=twopi /(double)n;
        i=1;
        l1=1;
        for(k1=1; k1<=nf; k1++)
        {
                ip=(int)wtable[k1+1+4*n];
                ld=0;
                l2=l1*ip;
                ido=n / l2;
                idot=ido+ido+2;
                ipm=ip-1;
                for(j=1; j<=ipm; j++)
                {
                    i1=i;
                    wtable[i-1+2*n]=1;
                    wtable[i+2*n]=0;
                    ld+=l1;
                    fi=0;
                    argld=ld*argh;
                    for(ii=4; ii<=idot; ii+=2)
                    {
                            i+=2;
                            fi+=1;
                            arg=fi*argld;
                            wtable[i-1+2*n]=Math.cos(arg);
                            wtable[i+2*n]=Math.sin(arg);
                    }
                    if(ip>5)
                    {
                            wtable[i1-1+2*n]=wtable[i-1+2*n];
                            wtable[i1+2*n]=wtable[i+2*n];
                    }
                }
                l1=l2;
        }

    }
/*---------------------------------------------------------
  cffti:  Initialization of Real forward FFT
 ---------------------------------------------------------*/
    void cffti(int n, double wtable[])
    {
        if(n==1) return;
        cffti1(n, wtable);
    }       /*cffti*/
}
```

Connect_Mio

```
package nl.dut.ide.software.soundfilterrealgit;

import android.annotation.SuppressLint;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.bluetooth.BluetoothGatt;
import android.bluetooth.BluetoothGattCallback;
import android.bluetooth.BluetoothGattCharacteristic;
import android.bluetooth.BluetoothGattDescriptor;
import android.bluetooth.BluetoothGattService;
import android.bluetooth.BluetoothManager;
import android.bluetooth.le.BluetoothLeScanner;
import android.bluetooth.le.ScanCallback;
import android.bluetooth.le.ScanResult;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.os.AsyncTask;
import android.os.Bundle;
import android.os.Handler;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.text.method.ScrollingMovementMethod;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;

import com.google.firebase.database.DatabaseReference;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import java.util.UUID;

public class ConnectMio extends AppCompatActivity {

    private static final String TAG = “ConnectMio”;

    private int heartRateValue;

    DatabaseReference databasereference;
    // bluetooth
    BluetoothManager btManager;
    BluetoothAdapter btAdapter;
    BluetoothLeScanner btScanner;
    Button startScanningButton;
    Button stopScanningButton;
    TextView peripheralTextView;
    private final static int REQUEST_ENABLE_BT = 1;
    private static final int PERMISSION_REQUEST_COARSE_LOCATION = 1;

    Boolean btScanning = false;
    int deviceIndex = 0;
    ArrayList<BluetoothDevice> devicesDiscovered = new ArrayList<BluetoothDevice>();
    EditText deviceIndexInput;
    Button connectToDevice;
    Button disconnectDevice;
    BluetoothGatt bluetoothGatt;

    UUID HEART_RATE_SERVICE_UUID = convertFromInteger(0x180D);
    UUID HEART_RATE_MEASUREMENT_CHAR_UUID = convertFromInteger(0x2A37);
    UUID HEART_RATE_CONTROL_POINT_CHAR_UUID = convertFromInteger(0x2A39);
    UUID CLIENT_CHARACTERISTIC_CONFIG_UUID = convertFromInteger(0x2902);


    public final static String ACTION_GATT_CONNECTED =
          “com.example.bluetooth.le.ACTION_GATT_CONNECTED”;
    public final static String ACTION_GATT_DISCONNECTED =
          “com.example.bluetooth.le.ACTION_GATT_DISCONNECTED”;
    public final static String ACTION_GATT_SERVICES_DISCOVERED =
          “com.example.bluetooth.le.ACTION_GATT_SERVICES_DISCOVERED”;
    public final static String ACTION_DATA_AVAILABLE =
          “com.example.bluetooth.le.ACTION_DATA_AVAILABLE”;
    public final static String EXTRA_DATA =
          “com.example.bluetooth.le.EXTRA_DATA”;

    public Map<String, String> uuids = new HashMap<String, String>();

    // Stops scanning after 5 seconds.
    private Handler mHandler = new Handler();
    private static final long SCAN_PERIOD = 150000;

    @SuppressLint(“WrongViewCast”)

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_connect_mio);

        //bluetooth
        peripheralTextView = (TextView) findViewByld(R.id.PeripheralTextView);
        peripheralTextView.setMovementMethod(new ScrollingMovementMethod());
        deviceIndexInput = (EditText) findViewByld(R.id.InputIndex);
        deviceIndexInput.setText(“0”);
        connectToDevice = (Button) findViewByld(R.id.ConnectButton);
        connectToDevice.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                connectToDeviceSelected();
            }
        });

        disconnectDevice = (Button) findViewByld(R.id.DisconnectButton);
        disconnectDevice.setVisibility(View.INVISIBLE);
        disconnectDevice.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                disconnectDeviceSelected();
            }
        });

        startScanningButton = (Button) findViewByld(R.id.StartScanningButton);
        startScanningButton.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                startScanning();
            }
        });

        stopScanningButton = (Button) findViewByld(R.id.StopScanningButton);
        stopScanningButton.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                stopScanning();
            }
        });
        stopScanningButton.setVisibility(View.INVISIBLE);

        btManager = (BluetoothManager) getSystemService(Context.BLUETOOTH_SERVICE);
        btAdapter = btManager.getAdapter();
        btScanner = btAdapter.getBluetoothLeScanner();

        if (btAdapter != null && !btAdapter.isEnabled()) {
            Intent enableIntent = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
            startActivityForResult(enableIntent, REQUEST_ENABLE_BT);
        }

    }

    // Device scan callback.
    private ScanCallback leScanCallback = new ScanCallback() {
        @Override
        public void onScanResult(int callbackType, ScanResult result) {
            if (result.getDevice().getName() != null && result.getDevice().getName().startsWith(“MIO”)) {
                peripheralTextView.append(“Index: “ + deviceIndex + “, Device Name: “ + result.getDevice().getName() + “ rssi: “ + result.getRssi() + “,
MAC: “ + result.getDevice().getAddress() + “\n”);
                // filtering only MIO devices
                devicesDiscovered.add(result.getDevice());
                deviceIndex++;

                // auto scroll for text view
                final int scrollAmount = peripheralTextView.getLayout().getLineTop(peripheralTextView.getLineCount()) - peripheralTextView.getH-
eight();
                // if there is no need to scroll, scrollAmount will be <=0
                if (scrollAmount > 0) {
                    peripheralTextView.scrollTo(0, scrollAmount);
                }
            }
        }
    };


    // Device connect call back
    private final BluetoothGattCallback btleGattCallback = new BluetoothGattCallback() {

        @Override
        public void onCharacteristicChanged(BluetoothGatt gatt, final BluetoothGattCharacteristic characteristic) {
            // this will get called anytime you perform a read or write characteristic operation

            final byte[] teste = characteristic.getValue();

            final String batida = teste.toString();

            final String result = “result”;

            int format = BluetoothGattCharacteristic.FORMAT_UINT8;

            setHeartRateValue(characteristic.getIntValue(format, 1));

            String TAG = “d”;

            Log.d(TAG, String.format(“Received heart rate: %d”, getHeartRateValue()));

            Log.v(result, batida);

            ConnectMio.this.runOnUiThread(new Runnable() {
                public void run() {
                    peripheralTextView.append(“value of sensor (BPM) “ + getHeartRateValue() + “\n”);
                }
            });

        }
        @Override
        public void onConnectionStateChange(final BluetoothGatt gatt, final int status, final int newState) {
            // this will get called when a device connects or disconnects
            System.out.println(newState);
            switch (newState) {
                case 0:
                    ConnectMio.this.runOnUiThread(new Runnable() {
                        public void run() {
                            peripheralTextView.append("device disconnected\n");
                            connectToDevice.setVisibility(View.VISIBLE);
                            disconnectDevice.setVisibility(View.INVISIBLE);
                        }
                    });
                    break;
                case 2:
                    ConnectMio.this.runOnUiThread(new Runnable() {
                        public void run() {
                            peripheralTextView.append("device connected\n");
                            connectToDevice.setVisibility(View.INVISIBLE);
                            disconnectDevice.setVisibility(View.VISIBLE);
                        }
                    });

                    // discover services and characteristics for this device
                    bluetoothGatt.discoverServices();

                    break;
                default:
                    ConnectMio.this.runOnUiThread(new Runnable() {
                        public void run() {
                            peripheralTextView.append("we encounterned an unknown state, uh oh\n");
                        }
                    });
                    break;
            }
        }

        @Override
        public void onServicesDiscovered(final BluetoothGatt gatt, final int status) {
            // this will get called after the client initiates a            BluetoothGatt.discoverServices() call
            ConnectMio.this.runOnUiThread(new Runnable() {
                public void run() {
                    peripheralTextView.append("device services have been discovered\n");
                }
            });
            displayGattServices(bluetoothGatt.getServices());

            BluetoothGattCharacteristic characteristic = gatt.getService(HEART_RATE_SERVICE_UUID).getCharacteristic(HEART_RATE_MEASUREMENT_CHAR_UUID);

            //BluetoothGattCharacteristic battery = gatt.getService(BATTERY_SERVICE).getCharacteristic(BATTERY_LEVEL);

            //gatt.readCharacteristic(battery);

            gatt.setCharacteristicNotification(characteristic, true);

            BluetoothGattDescriptor descriptor = characteristic.getDescriptor(CLIENT_CHARACTERISTIC_CONFIG_UUID);

            descriptor.setValue(BluetoothGattDescriptor.ENABLE_NOTIFICATION_VALUE);

            gatt.writeDescriptor(descriptor);
        }

        @Override
        public void onDescriptorWrite(BluetoothGatt gatt, BluetoothGattDescriptor descriptor, int status) {
            BluetoothGattCharacteristic characteristic = gatt.getService(HEART_RATE_SERVICE_UUID).getCharacteristic(HEART_RATE_CONTROL_POINT_CHAR_UUID);

            characteristic.setValue(new byte[]{1, 1});
            gatt.writeCharacteristic(characteristic);
        }

        @Override
        // Result of a characteristic read operation
        public void onCharacteristicRead(BluetoothGatt gatt,
                                         BluetoothGattCharacteristic characteristic,
                                         int status) {
            if (status == BluetoothGatt.GATT_SUCCESS) {
                broadcastUpdate(ACTION_DATA_AVAILABLE, characteristic);
            }
        }
    };

    private void broadcastUpdate(final String action,
                                 final BluetoothGattCharacteristic characteristic) {

        System.out.println(characteristic.getUuid());
    }

    @Override
    public void onRequestPermissionsResult(int requestCode, String permissions[], int[] grantResults) {
        switch (requestCode) {
            case PERMISSION_REQUEST_COARSE_LOCATION: {
                if (grantResults[0] == PackageManager.PERMISSION_GRANTED) {
                    System.out.println("coarse location permission granted");
                } else {
                    final AlertDialog.Builder builder = new AlertDialog.Builder(this);
                    builder.setTitle("Functionality limited");
                    builder.setMessage("Since location access has not been granted, this app will not be able to discover beacons when in the background.");
                    builder.setPositiveButton(android.R.string.ok, null);
                    builder.setOnDismissListener(new DialogInterface.OnDismissListener() {

                        @Override
                        public void onDismiss(DialogInterface dialog) {
                        }

                    });
                    builder.show();
                }
                return;
            }
        }
    }

    public void startScanning() {
        System.out.println("start scanning");
        btScanning = true;
        deviceIndex = 0;
        devicesDiscovered.clear();
        peripheralTextView.setText("");
        peripheralTextView.append("Started Scanning\n");
        startScanningButton.setVisibility(View.INVISIBLE);
        stopScanningButton.setVisibility(View.VISIBLE);
        AsyncTask.execute(new Runnable() {
            @Override
            public void run() {
                btScanner.startScan(leScanCallback);
            }
        });

        mHandler.postDelayed(new Runnable() {
            @Override
            public void run() {
                stopScanning();
            }
        }, SCAN_PERIOD);
    }

    public void stopScanning() {
        System.out.println(“stopping scanning”);
        peripheralTextView.append(“Stopped Scanning\n”);
        btScanning = false;
        startScanningButton.setVisibility(View.VISIBLE);
        stopScanningButton.setVisibility(View.INVISIBLE);
        AsyncTask.execute(new Runnable() {
            @Override
            public void run() {
                btScanner.stopScan(leScanCallback);
            }
        });
    }

    public void connectToDeviceSelected() {
        peripheralTextView.append(“Trying to connect to device at index: “ + deviceIndexInput.getText() + “\n”);
        int deviceSelected = Integer.parseInt(deviceIndexInput.getText().toString());
        bluetoothGatt = devicesDiscovered.get(deviceSelected).connectGatt(this, false, btleGattCallback);
    }

    public void disconnectDeviceSelected() {
        peripheralTextView.append(“Disconnecting from device\n”);
        bluetoothGatt.disconnect();
    }

    private void displayGattServices(List<BluetoothGattService> gattServices) {
        if (gattServices == null) return;

        // Loops through available GATT Services.
        for (BluetoothGattService gattService : gattServices) {

            final String uuid = gattService.getUuid().toString();
            System.out.println(“Service discovered: “ + uuid);
            ConnectMio.this.runOnUiThread(new Runnable() {
                public void run() {
                    peripheralTextView.append(“Service disovered: “ + uuid + “\n”);
                }
            });
            new ArrayList<HashMap<String, String>>();
            List<BluetoothGattCharacteristic> gattCharacteristics =
                    gattService.getCharacteristics();

            // Loops through available Characteristics.
            for (BluetoothGattCharacteristic gattCharacteristic :
                    gattCharacteristics) {

                final String charUuid = gattCharacteristic.getUuid().toString();
                System.out.println(“Characteristic discovered for service: “ + charUuid);
                ConnectMio.this.runOnUiThread(new Runnable() {
                    public void run() {
                        peripheralTextView.append(“Characteristic discovered for service: “ + charUuid + “\n”);
                    }
                });

            }
        }
    }

    @Override
    public void onStart() {
        super.onStart();
    }

    @Override
    public void onStop() {
        super.onStop();

    }

    public UUID convertFromInteger(int i) {
        final long MSB = 0x0000000000001000L;
        final long LSB = 0x800000805f9b34fbL;
        long value = i & 0xFFFFFFFF;
        return new UUID(MSB | (value << 32), LSB);
    }

    @Override
    protected void onPause() {
        super.onPause();
    }

    public int getHeartRateValue() {
        return heartRateValue;
    }

    public void setHeartRateValue(int heartRateValue) {
        this.heartRateValue = heartRateValue;
    }
}
```

```
package nl.dut.ide.software.soundfilterrealgit;

import android.content.Intent;
import android.support.annotation.NonNull;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.util.Patterns;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ProgressBar;
import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.FirebaseApp;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
```

Create account

```
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.FirebaseDatabase;

import com.google.firebase.firestore.DocumentReference;
import com.google.firebase.firestore.FirebaseFirestore;
import com.google.firebase.firestore.QueryDocumentSnapshot;
import com.google.firebase.firestore.QuerySnapshot;
import com.google.firebase.firestore.auth.User;

import java.util.HashMap;
import java.util.Map;


public class CreateAccount extends AppCompatActivity {
    //the onclick listener is already set in xml
    //this button is used to request focus
    private Button createNewAccount;

    EditText editTextUserEmail,editTextUserPassword,editTextUserUnit,editTextUserName,editTextUserPasswordCheck;
    private FirebaseAuth mAuth;
    private ProgressBar progressBarSignUp;
    private FirebaseFirestore db;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_create_account);

        //buttons
        createNewAccount = findViewById(R.id.create_new_account);

        //user input
        editTextUserEmail= (EditText) findViewById(R.id.user_email);
        editTextUserPassword= (EditText) findViewById(R.id.user_password);
        editTextUserPasswordCheck=(EditText) findViewById(R.id.user_password_repeated);
        editTextUserName = (EditText) findViewById(R.id.user_name);
        editTextUserUnit = (EditText) findViewById(R.id.user_unit_number);

        //firebase
        FirebaseApp.initializeApp(this);
        mAuth = FirebaseAuth.getInstance();
        db = FirebaseFirestore.getInstance();
        progressBarSignUp= (ProgressBar) findViewById(R.id.progress_bar_create_account);

    }

    @Override
    public void onStart() {
        super.onStart();
        //if there is already a user logged in we got to the home screen
        if(mAuth.getCurrentUser()!= null){
            alreadyLogInToHome();
        }
    }

    public void checkUserInput(View view){
        //we will check whether the user filled in all the field and if they did that correctly
        final String email =editTextUserEmail.getText().toString().trim().replaceAll(“ “,””);
        final String password=editTextUserPassword.getText().toString().trim().replaceAll(“ “,””);
        final String passwordCheck=editTextUserPasswordCheck.getText().toString().trim().replaceAll(“ “,””);
        final String unit=editTextUserUnit.getText().toString().trim().replaceAll(“ “,””);
        final String name=editTextUserName.getText().toString();

        if (email.isEmpty()){
            editTextUserEmail.setError(“Email is required”);
            editTextUserEmail.requestFocus();
            return;
        }

        if (!Patterns.EMAIL_ADDRESS.matcher(email).matches()){
            editTextUserEmail.setError(“Please enter a valid email”);
            editTextUserEmail.requestFocus();
            return;
        }

        if (unit.isEmpty()){
            editTextUserUnit.setError(“Unit number is required”);
            editTextUserUnit.requestFocus();
            return;
        }

        if (name.isEmpty()){
            editTextUserName.setError(“Name is required”);
            editTextUserName.requestFocus();
            return;
        }

        if (password.isEmpty()){
            editTextUserPassword.setError(“Password is required”);
            editTextUserPassword.requestFocus();
            return;
        }

        if (password.length()<=8){
            editTextUserPassword.setError(“Password should be at least 8 characters long”);
            editTextUserPassword.requestFocus();
            return;
        }

        String upperCaseChars = “(.*[A-Z].*)”;
        if (!password.matches(upperCaseChars )){
            editTextUserPassword.setError(“Password should contain at least one capital”);
            editTextUserPassword.requestFocus();
            return;
        }

        String lowerCaseChars = “(.*[a-z].*)”;
        if (!password.matches(lowerCaseChars )){
            editTextUserPassword.setError(“Password should contain at least one lower case letter”);
            editTextUserPassword.requestFocus();
            return;
        }

        String numbers = “(.*[0-9].*)”;
        if (!password.matches(numbers )){
            editTextUserPassword.setError(“Password should contain at least one number”);
            editTextUserPassword.requestFocus();
            return;
        }

        String specialChars = “(.*[,~,!,@,#,$,%,^,&,*,(,),-,_,=,+,[,{,],},|,;,:,<,>,/,?].*$)”;
        if (!password.matches(specialChars )){
            editTextUserPassword.setError(“Password should contain at least one special character”);
            editTextUserPassword.requestFocus();
            return;
        }
        if(!password.equals(passwordCheck) ){
            editTextUserPasswordCheck.setError(“Passwords do not match”);
            editTextUserPasswordCheck.requestFocus();
            return;

        }
        //we conclude that the user input is correct
        //thus we will show the progress bar and really start to create the account
        progressBarSignUp.setVisibility(View.VISIBLE);
        createAccount(email,password,unit,name);

    }

    private void createAccount(final String email, final String password, final String unit, final String name) {
        //first we create the account in firebase auth

        mAuth.createUserWithEmailAndPassword(email, password)
                .addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {
                    @Override
                    public void onComplete(@NonNull Task<AuthResult> task) {
                        if (task.isSuccessful()) {
                            storeUser(email, unit, name);

                        } else {
                            progressBarSignUp.setVisibility(View.INVISIBLE);
                            Toast.makeText(CreateAccount.this, “Registration failed, please try again”,
                                    Toast.LENGTH_LONG).show();
                        }
                    }
                });

    }

    private void storeUser(String email, String unit, String name) {
        //now that the user is created we can get their personal key
        // that is created in firebase to identify them
        final String key =FirebaseAuth.getInstance().getCurrentUser().getUid();
        if (key==null){
            editTextUserEmail.requestFocus();
            Toast.makeText(CreateAccount.this, “Registration failed no user id found”,
                    Toast.LENGTH_LONG).show();
        }
        else {
            //user key is found and thus we can save the data correctly now

            //file the content so that it can be send away
            Map<String, Object> userInformation = new HashMap<>();
            userInformation.put(“unit”, unit);
            userInformation.put(“name”, name);
            userInformation.put(“email”, email);

            db.collection(“users”).document(key).set(userInformation).addOnCompleteListener(new OnCompleteListener<Void>() {
                @Override
                public void onComplete(@NonNull Task<Void> task) {
                    if (task.isSuccessful()) {
                        Toast.makeText(CreateAccount.this, “Successfully registered”,
                                Toast.LENGTH_LONG).show();
                        goToTestMic();

                    } else {
                        Toast.makeText(CreateAccount.this, “Registration failed, please try again”,
                                Toast.LENGTH_LONG).show();
                        createNewAccount.requestFocus();
                    }
                }
            });
        }

    }

    private void goToTestMic() {
        Intent intentMic= new Intent(this, TestMic.class);
        intentMic.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
        startActivity(intentMic);
    }

    public void onClickGoToLogin(View view){
        Intent intent = new Intent(this , LoginActivity.class);
        startActivity(intent);
    }

    public void alreadyLogInToHome(){
        Intent intentHome= new Intent(this, Record.class);
        startActivity(intentHome);
    }

}
```

Emotion List Adapter

```
package nl.dut.ide.software.soundfilterrealgit;

import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.Button;
import android.widget.TextView;

import java.util.ArrayList;
import java.util.Map;

public class EmotionListAdapter extends BaseAdapter {
    private final ArrayList mData;

    public EmotionListAdapter(Map<Integer,String>map){
        mData = new ArrayList();
        mData.addAll(map.entrySet());
        //EmotionListAdapter receives data from the SelectEmotion class
        //and saves the values of the treemap in een new array list
    }

    @Override
    public int getCount() {
        return mData.size();
    }

    @Override
    public Map.Entry<String, String> getItem(int position) {
        return (Map.Entry) mData.get(position);
    }

    @Override
    public long getItemId(int position) {
        return 0;
        //Method not needed for what we want to do
        //but has to be here, because the base adapter wants it
    }

    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        final View result;

        if (convertView == null) {
            //tell the inflater to use the custom listview .
            result = LayoutInflater.from(parent.getContext()).inflate(R.layout.custom_listview_select_emotion, parent, false);
        } else {
            result = convertView;
        }

        //get one key/value pair of the map at a certain position
        Map.Entry<String, String> item = getItem(position);

        //set the text of the listview to the value of the mapping to show the user
        //set the tag of the button to the key of the mapping to help tracking the button in SelectEmotion.java
        ((Button) result.findViewById(R.id.delete_emotion_button)).setTag(item.getKey());
        ((TextView) result.findViewById(R.id.emotion_time_text)).setText(item.getValue());

        return result;
    }
}
```

Heart rate

```
package nl.dut.ide.software.soundfilterrealgit;

public class HeartRate {
    private int heartRateValue;
    private Long timevalue;


   //constructors

    public HeartRate(int heartRateValue, Long timevalue) {
        this.heartRateValue = heartRateValue;
        this.timevalue = timevalue;
    }


    //getters

    public int getHeartRateValue() {
        return heartRateValue;
    }

    public Long getTimevalue() {
        return timevalue;
    }
}
```

Login Activity

```
package nl.dut.ide.software.soundfilterrealgit;

import android.animation.Animator;
import android.animation.AnimatorListenerAdapter;
import android.annotation.TargetApi;
import android.content.Context;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.net.sip.SipSession;
import android.support.annotation.NonNull;
import android.support.design.widget.Snackbar;
import android.support.v7.app.AppCompatActivity;
import android.app.LoaderManager.LoaderCallbacks;

import android.content.CursorLoader;
import android.content.Loader;
import android.database.Cursor;
import android.net.Uri;
import android.os.AsyncTask;
import android.os.AsyncTask;

import android.os.Build;
import android.os.Bundle;
import android.provider.ContactsContract;
import android.text.TextUtils;
import android.util.Log;
import android.view.KeyEvent;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.inputmethod.EditorInfo;
import android.widget.ArrayAdapter;
import android.widget.AutoCompleteTextView;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ProgressBar;
import android.widget.TextView;
import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.FirebaseApp;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;

import java.util.ArrayList;
import java.util.List;

import static android.Manifest.permission.READ_CONTACTS;

public class LoginActivity extends AppCompatActivity{

    AutoCompleteTextView mEmailView;
    EditText mPasswordView;
    private ProgressBar progressBarSignIn;
    private FirebaseAuth mAuth;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);
        progressBarSignIn= (ProgressBar) findViewById(R.id.progress_bar_login_account);
        //Textviews
        mEmailView= (AutoCompleteTextView) findViewByld(R.id.email);
        mPasswordView= (EditText) findViewByld(R.id.password);

        //Firebase
        FirebaseApp.initializeApp(this);
        mAuth = FirebaseAuth.getInstance();
    }

    @Override
    public void onStart() {
        super.onStart();
        //if there is already a user logged in we got to the home screen
        if(mAuth.getCurrentUser()!= null&&TestMic.isPermission()==true){
            Intent intentHome= new Intent(this, Record.class);
            startActivity(intentHome);
        }
    }

    public void onClickSignIn(View view) {
        // Store values at the time of the login attempt.
        final String password = mPasswordView.getText().toString();
        final String email = mEmailView.getText().toString();


        //check the user input
        if (email.isEmpty()){
            mEmailView.setError("Email is required");
            mEmailView.requestFocus();
            return;
        }
        if (password.isEmpty()){
            mPasswordView.setError("Password is required");
            mPasswordView.requestFocus();
            return;
        }

        //set the progressbar on
        progressBarSignIn.setVisibility(View.VISIBLE);


        mAuth.signInWithEmailAndPassword(email, password)
                .addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {
                    @Override
                    public void onComplete(@NonNull Task<AuthResult> task) {
                        if (task.isSuccessful()) {
                            //hide the progressbar on
                            progressBarSignIn.setVisibility(View.INVISIBLE);

                            Toast.makeText(LoginActivity.this, "Login Successfully", Toast.LENGTH_SHORT).show();

                            //go to the record screen
                            Intent intent = new Intent(LoginActivity.this, Record.class);
                            //we clear all the activities before so that the user will ot go to login screen
                            //while they are still logged in. They have to sign out first
                            intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                            startActivity(intent);

                        } else {
                            //hide the progressbar
                            progressBarSignIn.setVisibility(View.INVISIBLE);

                            //Show the user why the login failed
                            Toast.makeText(getApplicationContext(), task.getException().getMessage(), Toast.LENGTH_LONG).show();
                        }
                    }
                });
    }

    public void onClickCreate(View v) {
        //when the user does not have an account, guide them to create account page
        Intent intent = new Intent(this , CreateAccount.class);
        startActivity(intent);
    }

}
```

Real Double FFT

```
package nl.dut.ide.software.soundfilterrealgit;
/**
 * FFT transform of a real periodic sequence.
 * @author Baoshe Zhang
 * @author Astronomical Instrument Group of University of Lethbridge.
*/
public class RealDoubleFFT extends RealDoubleFFT_Mixed
{
/**
 * <em>norm_factor</em> can be used to normalize this FFT transform. This is because
 * a call of forward transform (<em>ft</em>) followed by a call of backward transform
 * (<em>bt</em>) will multiply the input sequence by <em>norm_factor</em>.
*/
     public double norm_factor;
     private double wavetable[];
     private int ndim;

/**
 * Construct a wavenumber table with size <em>n</em>.
 * The sequences with the same size can share a wavenumber table. The prime
 * factorization of <em>n</em> together with a tabulation of the trigonometric functions
 * are computed and stored.
 *
 * @param  n  the size of a real data sequence. When <em>n</em> is a multiplication of small
 * numbers (4, 2, 3, 5), this FFT transform is very efficient.
*/
     public RealDoubleFFT(int n)
     {
          ndim = n;
          norm_factor = n;
          if(wavetable == null || wavetable.length !=(2*ndim+15))
          {
              wavetable = new double[2*ndim + 15];
          }
          rffti(ndim, wavetable);
     }

/**
 * Forward real FFT transform. It computes the discrete transform of a real data sequence.
 *
 * @param x an array which contains the sequence to be transformed. After FFT,
 * <em>x</em> contains the transform coeffients used to construct <em>n</em> complex FFT coeffients.
 * <br>
 * The real part of the first complex FFT coeffients is <em>x</em>[0]; its imaginary part
```

```
  * is 0. If <em>n</em> is even set <em>m</em> = <em>n</em>/2, if <em>n</em> is odd set
  * <em>m</em> = <em>n</em>/2, then for
  * <br>
  * <em>k</em> = 1, …, <em>m</em>-1 <br>
  * the real part of <em>k</em>-th complex FFT coeffients is <em>x</em>[2*<em>k</em>-1];
  * <br>
  * the imaginary part of <em>k</em>-th complex FFT coeffients is <em>x</em>[2*<em>k</em>-2].
  * <br>
  * If <em>n</em> is even,
  * the real of part of (<em>n</em>/2)-th complex FFT coeffients is <em>x</em>[<em>n</em>]; its imaginary part is 0.
  * The remaining complex FFT coeffients can be obtained by the symmetry relation:
  * the (<em>n</em>-<em>k</em>)-th complex FFT coeffient is the conjugate of <em>n</em>-th complex FFT coeffient.
  *
*/
    public void ft(double x[])
    {
        if(x.length != ndim)
            throw new IllegalArgumentException("The length of data can not match that of the wavetable");
        rfftf(ndim, x, wavetable);
    }

/**
  * Forward real FFT transform. It computes the discrete transform of a real data sequence.
  *
  * @param x an array which contains the sequence to be transformed. After FFT,
  * <em>x</em> contains the transform coeffients used to construct <em>n</em> complex FFT coeffients.
  * <br>
  * @param y the first complex (<em>n</em>+1)/2 (when <em>n</em> is odd) or (<em>n</em>/2+1) (when
  * <em>n</em> is even) FFT coeffients.
  * The remaining complex FFT coeffients can be obtained by the symmetry relation:
  * the (<em>n</em>-<em>k</em>)-th complex FFT coeffient is the conjugate of <em>n</em>-th complex FFT coeffient.
  *
*/
    public void ft(double x[], Complex1D y)
    {
        if(x.length != ndim)
            throw new IllegalArgumentException("The length of data can not match that of the wavetable");
        rfftf(ndim, x, wavetable);

        if(ndim%2 == 0)
        {
            y.x = new double[ndim/2 + 1];
            y.y = new double[ndim/2 + 1];
        }
        else
        {
            y.x = new double[(ndim+1)/2];
            y.y = new double[(ndim+1)/2];
        }


        y.x[0] = x[0];
        y.y[0] = 0.0D;
        for(int i=1; i<(ndim+1)/2; i++)
        {
            y.x[i] = x[2*i-1];
            y.y[i] = x[2*i];
        }
        if(ndim%2 == 0)
        {
            y.x[ndim/2] = x[ndim-1];
            y.y[ndim/2] = 0.0D;
        }

    }

/**
  * Backward real FFT transform. It is the unnormalized inverse transform of <em>ft</em>(double[]).
  *
  * @param x an array which contains the sequence to be transformed. After FFT,
  * <em>x</em> contains the transform coeffients. Also see the comments of <em>ft</em>(double[])
  * for the relation between <em>x</em> and complex FFT coeffients.
*/
    public void bt(double x[])
    {
        if(x.length != ndim)
            throw new IllegalArgumentException("The length of data can not match that of the wavetable");
        rfftb(ndim, x, wavetable);
    }
/**
  * Backward real FFT transform. It is the unnormalized inverse transform of <em>ft</em>(Complex1D, double[]).
  *
  * @param x  an array which contains the sequence to be transformed. When <em>n</em> is odd, it contains the first
  * (<em>n</em>+1)/2 complex data; when <em>n</em> is even, it contains (<em>n</em>/2+1) complex data.
  * @param y  the real FFT coeffients.
  * <br>
  * Also see the comments of <em>ft</em>(double[]) for the relation
  * between <em>x</em> and complex FFT coeffients.
*/
    public void bt(Complex1D x, double y[])
    {
        if(ndim%2 == 0)
        {
            if(x.x.length != ndim/2+1)
                throw new IllegalArgumentException("The length of data can not match that of the wavetable");
        }
        else
        {
            if(x.x.length != (ndim+1)/2)
                throw new IllegalArgumentException("The length of data can not match that of the wavetable");
        }

        y[0] = x.x[0];
        for(int i=1; i<(ndim+1)/2; i++)
        {
            y[2*i-1]=x.x[i];
            y[2*i]=x.y[i];
        }
        if(ndim%2 == 0)
        {
            y[ndim-1]=x.x[ndim/2];
        }
        rfftb(ndim, y, wavetable);
    }
}
```

```
package nl.dut.ide.software.soundfilterrealgit;
/**
  * @author Baoshe Zhang
  * @author Astronomical Instrument Group of University of Lethbridge.
*/
class RealDoubleFFT_Mixed
```

```
{

/*------------------------------------------------
  radf2: Real FFT's forward processing of factor 2
  ----------------------------------------------*/
    void radf2(int ido, int l1, final double cc[], double ch[],
                final double wtable[], int offset)
    {
       int     i, k, ic;
       double  ti2, tr2;
       int iw1;
       iw1 = offset;

             for(k=0; k<l1; k++)
       {
                ch[2*k*ido]=cc[k*ido]+cc[(k+l1)*ido];
                ch[(2*k+1)*ido+ido-1]=cc[k*ido]-cc[(k+l1)*ido];
       }
       if(ido<2) return;
       if(ido !=2)
       {
                for(k=0; k<l1; k++)
                {
                   for(i=2; i<ido; i+=2)
                   {
                            ic=ido-i;
                            tr2 = wtable[i-2+iw1]*cc[i-1+(k+l1)*ido]
                   +wtable[i-1+iw1]*cc[i+(k+l1)*ido];
                            ti2 = wtable[i-2+iw1]*cc[i+(k+l1)*ido]
                   -wtable[i-1+iw1]*cc[i-1+(k+l1)*ido];
                            ch[i+2*k*ido]=cc[i+k*ido]+ti2;
                            ch[ic+(2*k+1)*ido]=ti2-cc[i+k*ido];
                            ch[i-1+2*k*ido]=cc[i-1+k*ido]+tr2;
                            ch[ic-1+(2*k+1)*ido]=cc[i-1+k*ido]-tr2;
                   }
                }
                if(ido%2==1)return;
       }
       for(k=0; k<l1; k++)
       {
                ch[(2*k+1)*ido]=-cc[ido-1+(k+l1)*ido];
                ch[ido-1+2*k*ido]=cc[ido-1+k*ido];
       }
    }

/*------------------------------------------------
  radb2: Real FFT's backward processing of factor 2
  ----------------------------------------------*/
    void radb2(int ido, int l1, final double cc[], double ch[],
                final double wtable[], int offset)
    {
       int     i, k, ic;
       double  ti2, tr2;
       int iw1 = offset;

       for(k=0; k<l1; k++)
       {
                ch[k*ido]=cc[2*k*ido]+cc[ido-1+(2*k+1)*ido];
                ch[(k+l1)*ido]=cc[2*k*ido]-cc[ido-1+(2*k+1)*ido];
       }
       if(ido<2) return;
       if(ido !=2)
       {
                for(k=0; k<l1;++k)
                {
                   for(i=2; i<ido; i+=2)
                   {
                            ic=ido-i;
                            ch[i-1+k*ido]=cc[i-1+2*k*ido]+cc[ic-1+(2*k+1)*ido];
                            tr2=cc[i-1+2*k*ido]-cc[ic-1+(2*k+1)*ido];
                            ch[i+k*ido]=cc[i+2*k*ido]-cc[ic+(2*k+1)*ido];
                            ti2=cc[i+(2*k)*ido]+cc[ic+(2*k+1)*ido];
                            ch[i-1+(k+l1)*ido]=wtable[i-2+iw1]*tr2-wtable[i-1+iw1]*ti2;
                            ch[i+(k+l1)*ido]=wtable[i-2+iw1]*ti2+wtable[i-1+iw1]*tr2;
                   }
                }
                if(ido%2==1) return;
       }
       for(k=0; k<l1; k++)
       {
                ch[ido-1+k*ido]=2*cc[ido-1+2*k*ido];
                ch[ido-1+(k+l1)*ido]=-2*cc[(2*k+1)*ido];
       }
   }

/*------------------------------------------------
  radf3: Real FFT's forward processing of factor 3
  ----------------------------------------------*/
   void radf3(int ido, int l1, final double cc[], double ch[],
               final double wtable[], int offset)
   {
       final double taur=-0.5D;
       final double taui=0.866025403784439D;
       int     i, k, ic;
       double  ci2, di2, di3, cr2, dr2, dr3, ti2, ti3, tr2, tr3;
       int iw1, iw2;
       iw1 = offset;
       iw2 = iw1 + ido;

       for(k=0; k<l1; k++)
       {
                cr2=cc[(k+l1)*ido]+cc[(k+2*l1)*ido];
                ch[3*k*ido]=cc[k*ido]+cr2;
                ch[(3*k+2)*ido]=taui*(cc[(k+l1*2)*ido]-cc[(k+l1)*ido]);
                ch[ido-1+(3*k+1)*ido]=cc[k*ido]+taur*cr2;
       }
       if(ido==1) return;
       for(k=0; k<l1; k++)
       {
                for(i=2; i<ido; i+=2)
                {
                   ic=ido-i;
                   dr2 = wtable[i-2+iw1]*cc[i-1+(k+l1)*ido]
                +wtable[i-1+iw1]*cc[i+(k+l1)*ido];
                   di2 = wtable[i-2+iw1]*cc[i+(k+l1)*ido]
                -wtable[i-1+iw1]*cc[i-1+(k+l1)*ido];
                   dr3 = wtable[i-2+iw2]*cc[i-1+(k+l1*2)*ido]
                +wtable[i-1+iw2]*cc[i+(k+l1*2)*ido];
                   di3 = wtable[i-2+iw2]*cc[i+(k+l1*2)*ido]
                -wtable[i-1+iw2]*cc[i-1+(k+l1*2)*ido];
                   cr2 = dr2+dr3;
                   ci2 = di2+di3;
                   ch[i-1+3*k*ido]=cc[i-1+k*ido]+cr2;
                   ch[i+3*k*ido]=cc[i+k*ido]+ci2;
                   tr2=cc[i-1+k*ido]+taur*cr2;
                   ti2=cc[i+k*ido]+taur*ci2;
                   tr3=taui*(di2-di3);
```

```
                    ti3=taui*(dr3-dr2);
                    ch[i-1+(3*k+2)*ido]=tr2+tr3;
                    ch[ic-1+(3*k+1)*ido]=tr2-tr3;
                    ch[i+(3*k+2)*ido]=ti2+ti3;
                    ch[ic+(3*k+1)*ido]=ti3-ti2;
                }
        }
    }

/*------------------------------------------------------
  radb3: Real FFT's backward processing of factor 3
 ------------------------------------------------------*/
    void radb3(int ido, int l1, final double cc[], double ch[],
                    final double wtable[], int offset)
    {
        final double taur=-0.5D;
        final double taui=0.866025403784439D;
        int i, k, ic;
        double  ci2, ci3, di2, di3, cr2, cr3, dr2, dr3, ti2, tr2;
        int iw1, iw2;
        iw1 = offset;
        iw2 = iw1 + ido;

        for(k=0; k<l1; k++)
        {
                tr2=2*cc[ido-1+(3*k+1)*ido];
                cr2=cc[3*k*ido]+taur*tr2;
                ch[k*ido]=cc[3*k*ido]+tr2;
                ci3=2*taui*cc[(3*k+2)*ido];
                ch[(k+l1)*ido]=cr2-ci3;
                ch[(k+2*l1)*ido]=cr2+ci3;
        }
        if(ido==1) return;
        for(k=0; k<l1; k++)
        {
                for(i=2; i<ido; i+=2)
                {
                    ic=ido-i;
                    tr2=cc[i-1+(3*k+2)*ido]+cc[ic-1+(3*k+1)*ido];
                    cr2=cc[i-1+3*k*ido]+taur*tr2;
                    ch[i-1+k*ido]=cc[i-1+3*k*ido]+tr2;
                    ti2=cc[i+(3*k+2)*ido]-cc[ic+(3*k+1)*ido];
                    ci2=cc[i+3*k*ido]+taur*ti2;
                    ch[i+k*ido]=cc[i+3*k*ido]+ti2;
                    cr3=taui*(cc[i-1+(3*k+2)*ido]-cc[ic-1+(3*k+1)*ido]);
                    ci3=taui*(cc[i+(3*k+2)*ido]+cc[ic+(3*k+1)*ido]);
                    dr2=cr2-ci3;
                    dr3=cr2+ci3;
                    di2=ci2+cr3;
                    di3=ci2-cr3;
                    ch[i-1+(k+l1)*ido] = wtable[i-2+iw1]*dr2
                              -wtable[i-1+iw1]*di2;
                    ch[i+(k+l1)*ido] = wtable[i-2+iw1]*di2
                            +wtable[i-1+iw1]*dr2;
                    ch[i-1+(k+2*l1)*ido] = wtable[i-2+iw2]*dr3
                               -wtable[i-1+iw2]*di3;
                    ch[i+(k+2*l1)*ido] = wtable[i-2+iw2]*di3
                              +wtable[i-1+iw2]*dr3;
                }
        }
    }

/*------------------------------------------------------
  radf4: Real FFT's forward processing of factor 4
 ------------------------------------------------------*/
    void radf4(int ido, int l1, final double cc[], double ch[],
                    final double wtable[], int offset)
    {
        final double hsqt2=0.7071067811865475D;
        int i, k, ic;
        double  ci2, ci3, ci4, cr2, cr3, cr4, ti1, ti2, ti3, ti4, tr1, tr2, tr3, tr4;
        int iw1, iw2, iw3;
        iw1 = offset;
        iw2 = offset + ido;
        iw3 = iw2 + ido;
        for(k=0; k<l1; k++)
        {
                tr1=cc[(k+l1)*ido]+cc[(k+3*l1)*ido];
                tr2=cc[k*ido]+cc[(k+2*l1)*ido];
                ch[4*k*ido]=tr1+tr2;
                ch[ido-1+(4*k+3)*ido]=tr2-tr1;
                ch[ido-1+(4*k+1)*ido]=cc[k*ido]-cc[(k+2*l1)*ido];
                ch[(4*k+2)*ido]=cc[(k+3*l1)*ido]-cc[(k+l1)*ido];
        }
        if(ido<2) return;
        if(ido !=2)
        {
                for(k=0; k<l1; k++)
                {
                    for(i=2; i<ido; i+=2)
                    {
                                    ic=ido-i;
                                    cr2 = wtable[i-2+iw1]*cc[i-1+(k+l1)*ido]
                      +wtable[i-1+iw1]*cc[i+(k+l1)*ido];
                                    ci2 = wtable[i-2+iw1]*cc[i+(k+l1)*ido]
                      -wtable[i-1+iw1]*cc[i-1+(k+l1)*ido];
                                    cr3 = wtable[i-2+iw2]*cc[i-1+(k+2*l1)*ido]
                      +wtable[i-1+iw2]*cc[i+(k+2*l1)*ido];
                                    ci3 = wtable[i-2+iw2]*cc[i+(k+2*l1)*ido]
                      -wtable[i-1+iw2]*cc[i-1+(k+2*l1)*ido];
                                    cr4 = wtable[i-2+iw3]*cc[i-1+(k+3*l1)*ido]
                      +wtable[i-1+iw3]*cc[i+(k+3*l1)*ido];
                                    ci4 = wtable[i-2+iw3]*cc[i+(k+3*l1)*ido]
                      -wtable[i-1+iw3]*cc[i-1+(k+3*l1)*ido];
                                    tr1=cr2+cr4;
                                    tr4=cr4-cr2;
                                    ti1=ci2+ci4;
                                    ti4=ci2-ci4;
                                    ti2=cc[i+k*ido]+ci3;
                                    ti3=cc[i+k*ido]-ci3;
                                    tr2=cc[i-1+k*ido]+cr3;
                                    tr3=cc[i-1+k*ido]-cr3;
                                    ch[i-1+4*k*ido]=tr1+tr2;
                                    ch[ic-1+(4*k+3)*ido]=tr2-tr1;
                                    ch[i+4*k*ido]=ti1+ti2;
                                    ch[ic+(4*k+3)*ido]=ti1-ti2;
                                    ch[i-1+(4*k+2)*ido]=ti4+tr3;
                                    ch[ic-1+(4*k+1)*ido]=tr3-ti4;
                                    ch[i+(4*k+2)*ido]=tr4+ti3;
                                    ch[ic+(4*k+1)*ido]=tr4-ti3;
                    }
                }
                if(ido%2==1) return;
        }
        for(k=0; k<l1; k++)
        {
                ti1=-hsqt2*(cc[ido-1+(k+l1)*ido]+cc[ido-1+(k+3*l1)*ido]);
```

```
            tr1=hsqt2*(cc[ido-1+(k+l1)*ido]-cc[ido-1+(k+3*l1)*ido]);
            ch[ido-1+4*k*ido]=tr1+cc[ido-1+k*ido];
                  ch[ido-1+(4*k+2)*ido]=cc[ido-1+k*ido]-tr1;
                  ch[(4*k+1)*ido]=ti1-cc[ido-1+(k+2*l1)*ido];
                  ch[(4*k+3)*ido]=ti1+cc[ido-1+(k+2*l1)*ido];
        }
    }

/*------------------------------------------------
  radb4: Real FFT's backward processing of factor 4
  -----------------------------------------------*/
    void radb4(int ido, int l1, final double cc[], double ch[],
                 final double wtable[], int offset)
    {
        final double sqrt2=1.414213562373095D;
        int i, k, ic;
        double  ci2, ci3, ci4, cr2, cr3, cr4;
        double  ti1, ti2, ti3, ti4, tr1, tr2, tr3, tr4;
        int iw1, iw2, iw3;
        iw1 = offset;
        iw2 = iw1 + ido;
        iw3 = iw2 + ido;

        for(k=0; k<l1; k++)
        {
                tr1=cc[4*k*ido]-cc[ido-1+(4*k+3)*ido];
                tr2=cc[4*k*ido]+cc[ido-1+(4*k+3)*ido];
                tr3=cc[ido-1+(4*k+1)*ido]+cc[ido-1+(4*k+1)*ido];
                tr4=cc[(4*k+2)*ido]+cc[(4*k+2)*ido];
                ch[k*ido]=tr2+tr3;
                ch[(k+l1)*ido]=tr1-tr4;
                ch[(k+2*l1)*ido]=tr2-tr3;
                ch[(k+3*l1)*ido]=tr1+tr4;
        }
        if(ido<2) return;
        if(ido !=2)
        {
                for(k=0; k<l1;++k)
                {
                    for(i=2; i<ido; i+=2)
                    {
                                        ic=ido-i;
                                        ti1=cc[i+4*k*ido]+cc[ic+(4*k+3)*ido];
                                        ti2=cc[i+4*k*ido]-cc[ic+(4*k+3)*ido];
                                        ti3=cc[i+(4*k+2)*ido]-cc[ic+(4*k+1)*ido];
                                        tr4=cc[i+(4*k+2)*ido]+cc[ic+(4*k+1)*ido];
                                        tr1=cc[i-1+4*k*ido]-cc[ic-1+(4*k+3)*ido];
                                        tr2=cc[i-1+4*k*ido]+cc[ic-1+(4*k+3)*ido];
                                        ti4=cc[i-1+(4*k+2)*ido]-cc[ic-1+(4*k+1)*ido];
                                        tr3=cc[i-1+(4*k+2)*ido]+cc[ic-1+(4*k+1)*ido];
                                        ch[i-1+k*ido]=tr2+tr3;
                                        cr3=tr2-tr3;
                                        ch[i+k*ido]=ti2+ti3;
                                        ci3=ti2-ti3;
                                        cr2=tr1-tr4;
                                        cr4=tr1+tr4;
                                        ci2=ti1+ti4;
                                        ci4=ti1-ti4;
                                        ch[i-1+(k+l1)*ido] = wtable[i-2+iw1]*cr2
                                    -wtable[i-1+iw1]*ci2;
                                        ch[i+(k+l1)*ido] = wtable[i-2+iw1]*ci2
                                  +wtable[i-1+iw1]*cr2;
                                        ch[i-1+(k+2*l1)*ido] = wtable[i-2+iw2]*cr3
                                      -wtable[i-1+iw2]*ci3;
                                        ch[i+(k+2*l1)*ido] = wtable[i-2+iw2]*ci3
                                    +wtable[i-1+iw2]*cr3;
                                        ch[i-1+(k+3*l1)*ido] = wtable[i-2+iw3]*cr4
                                      -wtable[i-1+iw3]*ci4;
                                        ch[i+(k+3*l1)*ido] = wtable[i-2+iw3]*ci4
                                    +wtable[i-1+iw3]*cr4;
                    }
                }
                if(ido%2==1) return;
        }
        for(k=0; k<l1; k++)
        {
                ti1=cc[(4*k+1)*ido]+cc[(4*k+3)*ido];
                ti2=cc[(4*k+3)*ido]-cc[(4*k+1)*ido];
                tr1=cc[ido-1+4*k*ido]-cc[ido-1+(4*k+2)*ido];
                tr2=cc[ido-1+4*k*ido]+cc[ido-1+(4*k+2)*ido];
                ch[ido-1+k*ido]=tr2+tr2;
                ch[ido-1+(k+l1)*ido]=sqrt2*(tr1-ti1);
                ch[ido-1+(k+2*l1)*ido]=ti2+ti2;
                ch[ido-1+(k+3*l1)*ido]=-sqrt2*(tr1+ti1);
        }
    }

/*------------------------------------------------
  radf5: Real FFT's forward processing of factor 5
  -----------------------------------------------*/
    void radf5(int ido, int l1, final double cc[], double ch[],
                 final double wtable[], int offset)
    {
        final double tr11=0.309016994374947D;
        final double ti11=0.951056516295154D;
        final double tr12=-0.809016994374947D;
        final double ti12=0.587785252292473D;
        int      i, k, ic;
        double  ci2, di2, ci4, ci5, di3, di4, di5, ci3, cr2, cr3, dr2, dr3,
             dr4, dr5, cr5, cr4, ti2, ti3, ti5, ti4, tr2, tr3, tr4, tr5;
        int iw1, iw2, iw3, iw4;
        iw1 = offset;
        iw2 = iw1 + ido;
        iw3 = iw2 + ido;
        iw4 = iw3 + ido;

        for(k=0; k<l1; k++)
        {
                cr2=cc[(k+4*l1)*ido]+cc[(k+l1)*ido];
                ci5=cc[(k+4*l1)*ido]-cc[(k+l1)*ido];
                cr3=cc[(k+3*l1)*ido]+cc[(k+2*l1)*ido];
                ci4=cc[(k+3*l1)*ido]-cc[(k+2*l1)*ido];
                ch[5*k*ido]=cc[k*ido]+cr2+cr3;
                ch[ido-1+(5*k+1)*ido]=cc[k*ido]+tr11*cr2+tr12*cr3;
                ch[(5*k+2)*ido]=ti11*ci5+ti12*ci4;
                ch[ido-1+(5*k+3)*ido]=cc[k*ido]+tr12*cr2+tr11*cr3;
                ch[(5*k+4)*ido]=ti12*ci5-ti11*ci4;
        }
        if(ido==1) return;
        for(k=0; k<l1;++k)
        {
                for(i=2; i<ido; i+=2)
                {
                      ic=ido-i;
                            dr2 = wtable[i-2+iw1]*cc[i-1+(k+l1)*ido]
                   +wtable[i-1+iw1]*cc[i+(k+l1)*ido];
                            di2 = wtable[i-2+iw1]*cc[i+(k+l1)*ido]
```

```
              -wtable[i-1+iw1]*cc[i-1+(k+l1)*ido];
                    dr3 = wtable[i-2+iw2]*cc[i-1+(k+2*l1)*ido]
              +wtable[i-1+iw2]*cc[i+(k+2*l1)*ido];
                    di3 = wtable[i-2+iw2]*cc[i+(k+2*l1)*ido]
              -wtable[i-1+iw2]*cc[i-1+(k+2*l1)*ido];
                    dr4 = wtable[i-2+iw3]*cc[i-1+(k+3*l1)*ido]
              +wtable[i-1+iw3]*cc[i+(k+3*l1)*ido];
                    di4 = wtable[i-2+iw3]*cc[i+(k+3*l1)*ido]
              -wtable[i-1+iw3]*cc[i-1+(k+3*l1)*ido];
                    dr5 = wtable[i-2+iw4]*cc[i-1+(k+4*l1)*ido]
              +wtable[i-1+iw4]*cc[i+(k+4*l1)*ido];
                    di5 = wtable[i-2+iw4]*cc[i+(k+4*l1)*ido]
              -wtable[i-1+iw4]*cc[i-1+(k+4*l1)*ido];
                    cr2=dr2+dr5;
                    ci5=dr5-dr2;
                    cr5=di2-di5;
                    ci2=di2+di5;
                    cr3=dr3+dr4;
                    ci4=dr4-dr3;
                    cr4=di3-di4;
                    ci3=di3+di4;
                    ch[i-1+5*k*ido]=cc[i-1+k*ido]+cr2+cr3;
                    ch[i+5*k*ido]=cc[i+k*ido]+ci2+ci3;
                    tr2=cc[i-1+k*ido]+tr11*cr2+tr12*cr3;
                    ti2=cc[i+k*ido]+tr11*ci2+tr12*ci3;
                    tr3=cc[i-1+k*ido]+tr12*cr2+tr11*cr3;
                    ti3=cc[i+k*ido]+tr12*ci2+tr11*ci3;
                    tr5=ti11*cr5+ti12*cr4;
                    ti5=ti11*ci5+ti12*ci4;
                    tr4=ti12*cr5-ti11*cr4;
                    ti4=ti12*ci5-ti11*ci4;
                    ch[i-1+(5*k+2)*ido]=tr2+tr5;
                    ch[ic-1+(5*k+1)*ido]=tr2-tr5;
                    ch[i+(5*k+2)*ido]=ti2+ti5;
                    ch[ic+(5*k+1)*ido]=ti5-ti2;
                    ch[i-1+(5*k+4)*ido]=tr3+tr4;
                    ch[ic-1+(5*k+3)*ido]=tr3-tr4;
                    ch[i+(5*k+4)*ido]=ti3+ti4;
                    ch[ic+(5*k+3)*ido]=ti4-ti3;
            }
     }
   }

/*-------------------------------------------------
  radb5: Real FFT's backward processing of factor 5
 -------------------------------------------------*/
   void radb5(int ido, int l1, final double cc[], double ch[],
                   final double wtable[], int offset)
   {
     final double tr11=0.309016994374947D;
     final double ti11=0.951056516295154D;
     final double tr12=-0.809016994374947D;
     final double ti12=0.587785252292473D;
     int      i, k, ic;
     double  ci2, ci3, ci4, ci5, di3, di4, di5, di2, cr2, cr3, cr5, cr4,
         ti2, ti3, ti4, ti5, dr3, dr4, dr5, dr2, tr2, tr3, tr4, tr5;
     int iw1, iw2, iw3, iw4;
     iw1 = offset;
     iw2 = iw1 + ido;
     iw3 = iw2 + ido;
     iw4 = iw3 + ido;

     for(k=0; k<l1; k++)
     {
                  ti5=2*cc[(5*k+2)*ido];
                  ti4=2*cc[(5*k+4)*ido];
                  tr2=2*cc[ido-1+(5*k+1)*ido];
                  tr3=2*cc[ido-1+(5*k+3)*ido];
                  ch[k*ido]=cc[5*k*ido]+tr2+tr3;
                  cr2=cc[5*k*ido]+tr11*tr2+tr12*tr3;
                  cr3=cc[5*k*ido]+tr12*tr2+tr11*tr3;
                  ci5=ti11*ti5+ti12*ti4;
                  ci4=ti12*ti5-ti11*ti4;
                  ch[(k+l1)*ido]=cr2-ci5;
                  ch[(k+2*l1)*ido]=cr3-ci4;
                  ch[(k+3*l1)*ido]=cr3+ci4;
                  ch[(k+4*l1)*ido]=cr2+ci5;
     }
     if(ido==1) return;
     for(k=0; k<l1;++k)
     {
            for(i=2; i<ido; i+=2)
            {
                    ic=ido-i;
                    ti5=cc[i+(5*k+2)*ido]+cc[ic+(5*k+1)*ido];
                    ti2=cc[i+(5*k+2)*ido]-cc[ic+(5*k+1)*ido];
                    ti4=cc[i+(5*k+4)*ido]+cc[ic+(5*k+3)*ido];
                    ti3=cc[i+(5*k+4)*ido]-cc[ic+(5*k+3)*ido];
                    tr5=cc[i-1+(5*k+2)*ido]-cc[ic-1+(5*k+1)*ido];
                    tr2=cc[i-1+(5*k+2)*ido]+cc[ic-1+(5*k+1)*ido];
                    tr4=cc[i-1+(5*k+4)*ido]-cc[ic-1+(5*k+3)*ido];
                    tr3=cc[i-1+(5*k+4)*ido]+cc[ic-1+(5*k+3)*ido];
                    ch[i-1+k*ido]=cc[i-1+5*k*ido]+tr2+tr3;
                    ch[i+k*ido]=cc[i+5*k*ido]+ti2+ti3;
                    cr2=cc[i-1+5*k*ido]+tr11*tr2+tr12*tr3;

                    ci2=cc[i+5*k*ido]+tr11*ti2+tr12*ti3;
                    cr3=cc[i-1+5*k*ido]+tr12*tr2+tr11*tr3;

                    ci3=cc[i+5*k*ido]+tr12*ti2+tr11*ti3;
                    cr5=ti11*tr5+ti12*tr4;
                    ci5=ti11*ti5+ti12*ti4;
                    cr4=ti12*tr5-ti11*tr4;
                    ci4=ti12*ti5-ti11*ti4;
                    dr3=cr3-ci4;
                    dr4=cr3+ci4;
                    di3=ci3+cr4;
                    di4=ci3-cr4;
                    dr5=cr2+ci5;
                    dr2=cr2-ci5;
                    di5=ci2-cr5;
                    di2=ci2+cr5;
                    ch[i-1+(k+l1)*ido] = wtable[i-2+iw1]*dr2
                      -wtable[i-1+iw1]*di2;
                    ch[i+(k+l1)*ido] = wtable[i-2+iw1]*di2
                     +wtable[i-1+iw1]*dr2;
                    ch[i-1+(k+2*l1)*ido] = wtable[i-2+iw2]*dr3
                       -wtable[i-1+iw2]*di3;
                    ch[i+(k+2*l1)*ido] = wtable[i-2+iw2]*di3
                      +wtable[i-1+iw2]*dr3;
                    ch[i-1+(k+3*l1)*ido] = wtable[i-2+iw3]*dr4
                       -wtable[i-1+iw3]*di4;
                    ch[i+(k+3*l1)*ido] = wtable[i-2+iw3]*di4
                      +wtable[i-1+iw3]*dr4;
                    ch[i-1+(k+4*l1)*ido] = wtable[i-2+iw4]*dr5
                       -wtable[i-1+iw4]*di5;
                    ch[i+(k+4*l1)*ido] = wtable[i-2+iw4]*di5
                    +wtable[i-1+iw4]*dr5;
           }
    }
  }

/*-------------------------------------------------------
 radfg: Real FFT's forward processing of general factor
 --------------------------------------------------------*/
  void radfg(int ido, int ip, int l1, int idl1, double cc[],
           double c1[], double c2[], double ch[], double ch2[],
           final double wtable[], int offset)
  {
     final double twopi=2.0D*Math.PI; //6.28318530717959;
     int     idij, ipph, i, j, k, l, j2, ic, jc, lc, ik, is, nbd;
     double  dc2, ai1, ai2, ar1, ar2, ds2, dcp, arg, dsp, ar1h, ar2h;
     int iw1 = offset;

     arg=twopi / (double)ip;
     dcp=Math.cos(arg);
     dsp=Math.sin(arg);
     ipph=(ip+1)/ 2;
     nbd=(ido-1)/ 2;
     if(ido !=1)
     {
           for(ik=0; ik<idl1; ik++) ch2[ik]=c2[ik];
           for(j=1; j<ip; j++)
             for(k=0; k<l1; k++)
                   ch[(k+j*l1)*ido]=c1[(k+j*l1)*ido];
           if(nbd<=l1)
           {
             is=-ido;
             for(j=1; j<ip; j++)
             {
                   is+=ido;
                   idij=is-1;
                   for(i=2; i<ido; i+=2)
                   {
                     idij+=2;
                     for(k=0; k<l1; k++)
                     {
                            ch[i-1+(k+j*l1)*ido]=
                              wtable[idij-1+iw1]*c1[i-1+(k+j*l1)*ido]
                 +wtable[idij+iw1]*c1[i+(k+j*l1)*ido];
                            ch[i+(k+j*l1)*ido]=
                              wtable[idij-1+iw1]*c1[i+(k+j*l1)*ido]
                 -wtable[idij+iw1]*c1[i-1+(k+j*l1)*ido];
                     }
                   }
             }
           }
           else
           {
             is=-ido;
             for(j=1; j<ip; j++)
             {
                   is+=ido;
                   for(k=0; k<l1; k++)
                   {
                     idij=is-1;
                     for(i=2; i<ido; i+=2)
                     {
                           idij+=2;
                           ch[i-1+(k+j*l1)*ido]=
                             wtable[idij-1+iw1]*c1[i-1+(k+j*l1)*ido]
                 +wtable[idij+iw1]*c1[i+(k+j*l1)*ido];
                           ch[i+(k+j*l1)*ido]=
                             wtable[idij-1+iw1]*c1[i+(k+j*l1)*ido]
                 -wtable[idij+iw1]*c1[i-1+(k+j*l1)*ido];
                     }
                   }
             }
           }
           if(nbd>=l1)
           {
             for(j=1; j<ipph; j++)
             {
                   jc=ip-j;
                   for(k=0; k<l1; k++)
                   {
                     for(i=2; i<ido; i+=2)
                     {
                           c1[i-1+(k+j*l1)*ido]=ch[i-1+(k+j*l1)*ido]+ch[i-1+(k+jc*l1)*ido];
                           c1[i-1+(k+jc*l1)*ido]=ch[i+(k+j*l1)*ido]-ch[i+(k+jc*l1)*ido];
                           c1[i+(k+j*l1)*ido]=ch[i+(k+j*l1)*ido]+ch[i+(k+jc*l1)*ido];
                           c1[i+(k+jc*l1)*ido]=ch[i-1+(k+jc*l1)*ido]-ch[i-1+(k+j*l1)*ido];
                     }
                   }
             }
           }
           else
           {
             for(j=1; j<ipph; j++)
             {
                   jc=ip-j;
                   for(i=2; i<ido; i+=2)
                   {
                     for(k=0; k<l1; k++)
                     {
                           c1[i-1+(k+j*l1)*ido]=
                             ch[i-1+(k+j*l1)*ido]+ch[i-1+(k+jc*l1)*ido];
                           c1[i-1+(k+jc*l1)*ido]=ch[i+(k+j*l1)*ido]-ch[i+(k+jc*l1)*ido];
                           c1[i+(k+j*l1)*ido]=ch[i+(k+j*l1)*ido]+ch[i+(k+jc*l1)*ido];
                           c1[i+(k+jc*l1)*ido]=ch[i-1+(k+jc*l1)*ido]-ch[i-1+(k+j*l1)*ido];
                     }
                   }
             }
           }
     }
     else
     {
           for(ik=0; ik<idl1; ik++) c2[ik]=ch2[ik];
     }
     for(j=1; j<ipph; j++)
     {
           jc=ip-j;
           for(k=0; k<l1; k++)
           {
             c1[(k+j*l1)*ido]=ch[(k+j*l1)*ido]+ch[(k+jc*l1)*ido];
             c1[(k+jc*l1)*ido]=ch[(k+jc*l1)*ido]-ch[(k+j*l1)*ido];
           }
     }

     ar1=1;
     ai1=0;
     for(l=1; l<ipph; l++)
     {
           lc=ip-l;
```

```
                  ar1h=dcp*ar1-dsp*ai1;
                  ai1=dcp*ai1+dsp*ar1;
                  ar1=ar1h;
                  for(ik=0; ik<idl1; ik++)
                  {
                      ch2[ik+l*idl1]=c2[ik]+ar1*c2[ik+idl1];
                      ch2[ik+lc*idl1]=ai1*c2[ik+(ip-1)*idl1];
                  }
                  dc2=ar1;
                  ds2=ai1;
                  ar2=ar1;
                  ai2=ai1;
                  for(j=2; j<ipph; j++)
                  {
                      jc=ip-j;
                      ar2h=dc2*ar2-ds2*ai2;
                      ai2=dc2*ai2+ds2*ar2;
                      ar2=ar2h;
                      for(ik=0; ik<idl1; ik++)
                      {
                                  ch2[ik+l*idl1]+=ar2*c2[ik+j*idl1];
                                  ch2[ik+lc*idl1]+=ai2*c2[ik+jc*idl1];
                      }
                  }
          }
          for(j=1; j<ipph; j++)
                 for(ik=0; ik<idl1; ik++)
                    ch2[ik]+=c2[ik+j*idl1];

          if(ido>=l1)
          {
                 for(k=0; k<l1; k++)
                 {
                    for(i=0; i<ido; i++)
                    {
                                cc[i+k*ip*ido]=ch[i+k*ido];
                    }
                 }
          }
          else
          {
                  for(i=0; i<ido; i++)
                  {
                      for(k=0; k<l1; k++)
                      {
                                  cc[i+k*ip*ido]=ch[i+k*ido];
                      }
                  }
          }
          for(j=1; j<ipph; j++)
          {
                  jc=ip-j;
                  j2=2*j;
                  for(k=0; k<l1; k++)
                  {
                      cc[ido-1+(j2-1+k*ip)*ido]=ch[(k+j*l1)*ido];
                      cc[(j2+k*ip)*ido]=ch[(k+jc*l1)*ido];
                  }
          }
          if(ido==1) return;
          if(nbd>=l1)
          {
                  for(j=1; j<ipph; j++)
                  {
                      jc=ip-j;
                      j2=2*j;
                      for(k=0; k<l1; k++)
                      {
                                  for(i=2; i<ido; i+=2)
                                  {
                                      ic=ido-i;
                                      cc[i-1+(j2+k*ip)*ido]=ch[i-1+(k+j*l1)*ido]+ch[i-1+(k+jc*l1)*ido];
                                      cc[ic-1+(j2-1+k*ip)*ido]=ch[i-1+(k+j*l1)*ido]-ch[i-1+(k+jc*l1)*ido];
                                      cc[i+(j2+k*ip)*ido]=ch[i+(k+j*l1)*ido]+ch[i+(k+jc*l1)*ido];
                                      cc[ic+(j2-1+k*ip)*ido]=ch[i+(k+jc*l1)*ido]-ch[i+(k+j*l1)*ido];
                                  }
                      }
                  }
          }
          else
          {
                  for(j=1; j<ipph; j++)
                  {
                      jc=ip-j;
                      j2=2*j;
                      for(i=2; i<ido; i+=2)
                      {
                                  ic=ido-i;
                                  for(k=0; k<l1; k++)
                                  {
                                      cc[i-1+(j2+k*ip)*ido]=ch[i-1+(k+j*l1)*ido]+ch[i-1+(k+jc*l1)*ido];
                                      cc[ic-1+(j2-1+k*ip)*ido]=ch[i-1+(k+j*l1)*ido]-ch[i-1+(k+jc*l1)*ido];
                                      cc[i+(j2+k*ip)*ido]=ch[i+(k+j*l1)*ido]+ch[i+(k+jc*l1)*ido];
                                      cc[ic+(j2-1+k*ip)*ido]=ch[i+(k+jc*l1)*ido]-ch[i+(k+j*l1)*ido];
                                  }
                      }
                  }
          }
      }
/*---------------------------------------------------------
  radbg: Real FFT's backward processing of general factor
  --------------------------------------------------------*/
      void radbg(int ido, int ip, int l1, int idl1, double cc[], double c1[],
                    double c2[], double ch[], double ch2[], final double wtable[], int offset)
      {
          final double twopi=2.0D*Math.PI; //6.28318530717959;
          int      idij, ipph, i, j, k, l, j2, ic, jc, lc, ik, is;
          double  dc2, ai1, ai2, ar1, ar2, ds2;
          int      nbd;
          double  dcp, arg, dsp, ar1h, ar2h;
          int iw1 = offset;

          arg=twopi / (double)ip;
          dcp=Math.cos(arg);
          dsp=Math.sin(arg);
          nbd=(ido-1)/ 2;
          ipph=(ip+1)/ 2;
          if(ido>=l1)
          {
                  for(k=0; k<l1; k++)
                  {
                      for(i=0; i<ido; i++)
                      {
                                  ch[i+k*ido]=cc[i+k*ip*ido];
                      }
                  }
```

```
}
else
{
        for(i=0; i<ido; i++)
        {
           for(k=0; k<l1; k++)
           {
                    ch[i+k*ido]=cc[i+k*ip*ido];
           }
        }
}
for(j=1; j<ipph; j++)
{
        jc=ip-j;
        j2=2*j;
        for(k=0; k<l1; k++)
        {
           ch[(k+j*l1)*ido]=cc[ido-1+(j2-1+k*ip)*ido]+cc[ido-1+(j2-1+k*ip)*ido];
           ch[(k+jc*l1)*ido]=cc[(j2+k*ip)*ido]+cc[(j2+k*ip)*ido];
        }
}

if(ido !=1)
{
        if(nbd>=l1)
        {
           for(j=1; j<ipph; j++)
           {
                    jc=ip-j;
                    for(k=0; k<l1; k++)
                    {
                       for(i=2; i<ido; i+=2)
                       {
                                ic=ido-i;
                                ch[i-1+(k+j*l1)*ido]=cc[i-1+(2*j+k*ip)*ido]+cc[ic-1+(2*j-1+k*ip)*ido];
                                ch[i-1+(k+jc*l1)*ido]=cc[i-1+(2*j+k*ip)*ido]-cc[ic-1+(2*j-1+k*ip)*ido];
                                ch[i+(k+j*l1)*ido]=cc[i+(2*j+k*ip)*ido]-cc[ic+(2*j-1+k*ip)*ido];
                                ch[i+(k+jc*l1)*ido]=cc[i+(2*j+k*ip)*ido]+cc[ic+(2*j-1+k*ip)*ido];
                       }
                    }
           }
        }
        else
        {
           for(j=1; j<ipph; j++)
           {
                    jc=ip-j;
                    for(i=2; i<ido; i+=2)
                    {
                       ic=ido-i;
                       for(k=0; k<l1; k++)
                       {
                                ch[i-1+(k+j*l1)*ido]=cc[i-1+(2*j+k*ip)*ido]+cc[ic-1+(2*j-1+k*ip)*ido];
                                ch[i-1+(k+jc*l1)*ido]=cc[i-1+(2*j+k*ip)*ido]-cc[ic-1+(2*j-1+k*ip)*ido];
                                ch[i+(k+j*l1)*ido]=cc[i+(2*j+k*ip)*ido]-cc[ic+(2*j-1+k*ip)*ido];
                                ch[i+(k+jc*l1)*ido]=cc[i+(2*j+k*ip)*ido]+cc[ic+(2*j-1+k*ip)*ido];
                       }
                    }
           }
        }
}

ar1=1;
ai1=0;
for(l=1; l<ipph; l++)
{
        lc=ip-l;
        ar1h=dcp*ar1-dsp*ai1;
        ai1=dcp*ai1+dsp*ar1;
        ar1=ar1h;
        for(ik=0; ik<idl1; ik++)
        {
          c2[ik+l*idl1]=ch2[ik]+ar1*ch2[ik+idl1];
          c2[ik+lc*idl1]=ai1*ch2[ik+(ip-1)*idl1];
        }
        dc2=ar1;
        ds2=ai1;
        ar2=ar1;
        ai2=ai1;
        for(j=2; j<ipph; j++)
        {
          jc=ip-j;
          ar2h=dc2*ar2-ds2*ai2;
          ai2=dc2*ai2+ds2*ar2;
          ar2=ar2h;
          for(ik=0; ik<idl1; ik++)
          {
                   c2[ik+l*idl1]+=ar2*ch2[ik+j*idl1];
                   c2[ik+lc*idl1]+=ai2*ch2[ik+jc*idl1];
          }
        }
}
for(j=1; j<ipph; j++)
{
        for(ik=0; ik<idl1; ik++)
        {
          ch2[ik]+=ch2[ik+j*idl1];
        }
}
for(j=1; j<ipph; j++)
{
        jc=ip-j;
        for(k=0; k<l1; k++)
        {
           ch[(k+j*l1)*ido]=c1[(k+j*l1)*ido]-c1[(k+jc*l1)*ido];
           ch[(k+jc*l1)*ido]=c1[(k+j*l1)*ido]+c1[(k+jc*l1)*ido];
        }
}

if(ido==1) return;
if(nbd>=l1)
{
        for(j=1; j<ipph; j++)
        {
           jc=ip-j;
           for(k=0; k<l1; k++)
           {
                    for(i=2; i<ido; i+=2)
                    {
                       ch[i-1+(k+j*l1)*ido]=c1[i-1+(k+j*l1)*ido]-c1[i+(k+jc*l1)*ido];
                       ch[i-1+(k+jc*l1)*ido]=c1[i-1+(k+j*l1)*ido]+c1[i+(k+jc*l1)*ido];
                       ch[i+(k+j*l1)*ido]=c1[i+(k+j*l1)*ido]+c1[i-1+(k+jc*l1)*ido];
                       ch[i+(k+jc*l1)*ido]=c1[i+(k+j*l1)*ido]-c1[i-1+(k+jc*l1)*ido];
                    }
           }
        }
}
```

```
        else
        {
                for(j=1; j<ipph; j++)
                {
                    jc=ip-j;
                    for(i=2; i<ido; i+=2)
                    {
                                for(k=0; k<l1; k++)
                                {
                                    ch[i-1+(k+j*l1)*ido]=c1[i-1+(k+j*l1)*ido]-c1[i+(k+jc*l1)*ido];
                                    ch[i-1+(k+jc*l1)*ido]=c1[i-1+(k+j*l1)*ido]+c1[i+(k+jc*l1)*ido];
                                    ch[i+(k+j*l1)*ido]=c1[i+(k+j*l1)*ido]+c1[i-1+(k+jc*l1)*ido];
                                    ch[i+(k+jc*l1)*ido]=c1[i+(k+j*l1)*ido]-c1[i-1+(k+jc*l1)*ido];
                                }
                    }
                }
        }
        for(ik=0; ik<idl1; ik++) c2[ik]=ch2[ik];
        for(j=1; j<ip; j++)
                for(k=0; k<l1; k++)
                    c1[(k+j*l1)*ido]=ch[(k+j*l1)*ido];
        if(nbd<=l1)
        {
                is=-ido;
                for(j=1; j<ip; j++)
                {
                    is+=ido;
                    idij=is-1;
                    for(i=2; i<ido; i+=2)
                    {
                                idij+=2;
                                for(k=0; k<l1; k++)
                                {
                                    c1[i-1+(k+j*l1)*ido] = wtable[idij-1+iw1]*ch[i-1+(k+j*l1)*ido]
                                         -wtable[idij+iw1]*ch[i+(k+j*l1)*ido];
                                    c1[i+(k+j*l1)*ido] = wtable[idij-1+iw1]*ch[i+(k+j*l1)*ido]
                                       +wtable[idij+iw1]*ch[i-1+(k+j*l1)*ido];
                                }
                    }
                }
        }
        else
        {
                is=-ido;
                for(j=1; j<ip; j++)
                {
                    is+=ido;
                    for(k=0; k<l1; k++)
                    {
                                idij=is-1;
                                for(i=2; i<ido; i+=2)
                                {
                                    idij+=2;
                                    c1[i-1+(k+j*l1)*ido] = wtable[idij-1+iw1]*ch[i-1+(k+j*l1)*ido]
                                         -wtable[idij+iw1]*ch[i+(k+j*l1)*ido];
                                    c1[i+(k+j*l1)*ido] = wtable[idij-1+iw1]*ch[i+(k+j*l1)*ido]
                                       +wtable[idij+iw1]*ch[i-1+(k+j*l1)*ido];
                                }
                    }
                }
        }
    }

/*---------------------------------------------------------
  rfftf1: further processing of Real forward FFT
  -------------------------------------------------------*/
    void rfftf1(int n, double c[], final double wtable[], int offset)
    {
        int     i;
        int     k1, l1, l2, na, kh, nf, ip, iw, ido, idl1;

        double[] ch = new double[n];
        System.arraycopy(wtable, offset, ch, 0, n);

        nf=(int)wtable[1+2*n+offset];
        na=1;
        l2=n;
        iw=n-1+n+offset;
        for(k1=1; k1<=nf;++k1)
        {
                kh=nf-k1;
                ip=(int)wtable[kh+2+2*n+offset];
                l1=l2 / ip;
                ido=n / l2;
                idl1=ido*l1;
                iw-=(ip-1)*ido;
                na=1-na;
                if(ip==4)
                {
                    if(na==0)
              {
                radf4(ido, l1, c, ch, wtable, iw);
              }
                    else
              {
                radf4(ido, l1, ch, c, wtable, iw);
              }
                }
                else if(ip==2)
                {
                    if(na==0)
              {
                radf2(ido, l1, c, ch, wtable, iw);
              }
                    else
              {
                radf2(ido, l1, ch, c, wtable, iw);
              }
                }
                else if(ip==3)
                {
                    if(na==0)
              {
                  radf3(ido, l1, c, ch, wtable, iw);
              }
                    else
              {
                  radf3(ido, l1, ch, c, wtable, iw);
              }
                }
                else if(ip==5)
                {
                    if(na==0)
              {
                 radf5(ido, l1, c, ch, wtable, iw);
              }
                    else
```

```
         {
            radf5(ido, l1, ch, c, wtable, iw);
         }
           }
           else
           {
         if(ido==1) na=1-na;
             if(na==0)
             {
           radfg(ido, ip, l1, idl1, c, c, c, ch, ch, wtable, iw);
                   na=1;
             }
             else
             {
           radfg(ido, ip, l1, idl1, ch, ch, ch, c, c, wtable, iw);
                   na=0;
             }
           }
           l2=l1;
     }
     if(na==1) return;
     for(i=0; i<n; i++) c[i]=ch[i];
  }

/*-------------------------------------------------------
 rfftb1: further processing of Real backward FFT
 -------------------------------------------------------*/
  void rfftb1(int n, double c[], final double wtable[], int offset)
  {
     int     i;
     int     k1, l1, l2, na, nf, ip, iw, ido, idl1;

     double[] ch = new double[n];
     System.arraycopy(wtable, offset, ch, 0, n);

     nf=(int)wtable[1+2*n+offset];
     na=0;
     l1=1;
     iw=n+offset;
     for(k1=1; k1<=nf; k1++)
     {
           ip=(int)wtable[k1+1+2*n+offset];
           l2=ip*l1;
           ido=n / l2;
           idl1=ido*l1;
           if(ip==4)
           {
             if(na==0)
         {
             radb4(ido, l1, c, ch, wtable, iw);
         }
             else
         {
             radb4(ido, l1, ch, c, wtable, iw);
         }
             na=1-na;
           }
           else if(ip==2)
           {
             if(na==0)
         {
            radb2(ido, l1, c, ch, wtable, iw);
         }
             else
         {
            radb2(ido, l1, ch, c, wtable, iw);
         }
             na=1-na;
           }
           else if(ip==3)
           {
             if(na==0)
         {
             radb3(ido, l1, c, ch, wtable, iw);
         }
             else
         {
             radb3(ido, l1, ch, c, wtable, iw);
         }
             na=1-na;
           }
           else if(ip==5)
           {
             if(na==0)
         {
             radb5(ido, l1, c, ch, wtable, iw);
         }
             else
         {
             radb5(ido, l1, ch, c, wtable, iw);
         }
             na=1-na;
           }
           else
           {
             if(na==0)
         {
             radbg(ido, ip, l1, idl1, c, c, c, ch, ch, wtable, iw);
         }
             else
         {
             radbg(ido, ip, l1, idl1, ch, ch, ch, c, c, wtable, iw);
         }
             if(ido==1) na=1-na;
           }
           l1=l2;
           iw+=(ip-1)*ido;
     }
     if(na==0) return;
     for(i=0; i<n; i++) c[i]=ch[i];
  }

/*-------------------------------------------------------
 rfftf: Real forward FFT
 -------------------------------------------------------*/
  void rfftf(int n, double r[], double wtable[])
  {
     if(n==1) return;
     rfftf1(n, r, wtable, 0);
  }       /*rfftf*/

/*-------------------------------------------------------
 rfftf: Real backward FFT
 -------------------------------------------------------*/
  void rfftb(int n, double r[], double wtable[])
  {
     if(n==1) return;
```

```
        rfftb1(n, r, wtable, 0);
    } /*rfftb*/

/*----------------------------------------------------------
   rfftil: further initialization of Real FFT
  -----------------------------------------------------------*/
    void rffti1(int n, double wtable[], int offset)
    {

        final int[] ntryh= new int[] {4, 2, 3, 5};
        final double twopi=2.0D*Math.PI;
        double  argh;
        int     ntry=0, i, j;
        double  argld;
        int     k1, l1, l2, ib;
        double  fi;
        int     ld, ii, nf, ip, nl, is, nq, nr;
        double  arg;
        int     ido, ipm;
        int     nfm1;

        nl=n;
        nf=0;
        j=0;

      factorize_loop:
        while(true)
        {
            ++j;
            if(j<=4)
                        ntry=ntryh[j-1];
            else
                        ntry+=2;
            do
            {
                        nq=nl / ntry;
                        nr=nl-ntry*nq;
                        if(nr !=0) continue factorize_loop;
                        ++nf;
                wtable[nf+1+2*n+offset]=ntry;

                        nl=nq;
                        if(ntry==2 && nf !=1)
                        {
                            for(i=2; i<=nf; i++)
                            {
                                        ib=nf-i+2;
                        wtable[ib+1+2*n+offset]=wtable[ib+2*n+offset];
                            }
                    wtable[2+2*n+offset]=2;
                        }
            }while(nl !=1);
            break factorize_loop;
        }
        wtable[0+2*n+offset] = n;
        wtable[1+2*n+offset] = nf;
        argh=twopi /(double)(n);
        is=0;
        nfm1=nf-1;
        l1=1;
        if(nfm1==0) return;
        for(k1=1; k1<=nfm1; k1++)
        {
            ip=(int)wtable[k1+1+2*n+offset];
                    ld=0;
                    l2=l1*ip;
                    ido=n / l2;
                    ipm=ip-1;
                    for(j=1; j<=ipm;++j)
                    {
                        ld+=l1;
                        i=is;
                        argld=(double)ld*argh;

                        fi=0;
                        for(ii=3; ii<=ido; ii+=2)
                        {
                                    i+=2;
                                    fi+=1;
                                    arg=fi*argld;
                    wtable[i-2+n+offset] = Math.cos(arg);
                    wtable[i-1+n+offset] = Math.sin(arg);
                        }
                        is+=ido;
                    }
                    l1=l2;
        }
    } /*rffti1*/

/*----------------------------------------------------------
   rffti: Initialization of Real FFT
  -----------------------------------------------------------*/
    void rffti(int n, double wtable[])  /* length of wtable = 2*n + 15 */
    {
        if(n==1) return;
        rffti1(n, wtable, 0);
    } /*rffti*/

}
```

```
package nl.dut.ide.software.soundfilterrealgit;

import android.Manifest;
import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.app.TimePickerDialog;
import android.content.Context;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.net.Uri;
import android.os.Build;
import android.os.Handler;
import android.support.annotation.NonNull;
import android.support.design.widget.NavigationView;
import android.support.v4.app.ActivityCompat;
import android.support.v4.app.NotificationCompat;
import android.support.v4.app.NotificationManagerCompat;
import android.support.v4.content.ContextCompat;
import android.support.v4.view.GravityCompat;
import android.support.v4.widget.DrawerLayout;
import android.support.v7.app.ActionBarDrawerToggle;
```

Record

```
import android.support.v7.app.AppCompatActivity;

import java.nio.ByteBuffer;
import java.text.DateFormat;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.Date;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

import android.media.AudioFormat;
import android.media.AudioRecord;
import android.media.MediaRecorder;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.TextView;
import android.widget.TimePicker;
import android.widget.Toast;

import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.firestore.DocumentSnapshot;
import com.google.firebase.storage.FirebaseStorage;
import com.google.firebase.storage.StorageReference;
import com.google.firebase.storage.UploadTask;
import com.webianks.library.scroll_choice.ScrollChoice;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.firestore.FirebaseFirestore;

//Zoltan, the things that are included for place and heart rate can be found by doing ctrl+F "Zoltan"

public class Record extends AppCompatActivity {
    private TextView name, nameMenu, firstLetter, firstLetterMenu;                              //Textview to show the name of the user

    //Recording
    private String TAG = "Record";                          //TAG is used for Logcat, to easily collect the text send to myself
    private static final int RECORDER_SAMPLERATE = 44100;   //Every second 44100 audio samples are taken from the analogue signal, this is
the only sample rate appropriate for all phones
    private static final int RECORDER_CHANNELS = AudioFormat
        .CHANNEL_IN_MONO;                                   //Mono is used when you have one channel your audio samples have to go to (for example,
one speaker)
    private static final int RECORDER_AUDIO_ENCODING =
        AudioFormat.ENCODING_PCM_16BIT;                     //Amount of bits used for Pulse-Code-Modulation, 16 is teh most common one
    private AudioRecord recorder = null;                    //Make a new, empty Audio recorder
    private Thread recordingThread = null;                   //Make a new thread for the Audio recorder
    private static boolean isRecording;                     //To check if certain methods have to be done
    int bufferSize = AudioRecord.getMinBufferSize(           //Buffersize for Zoltan
        RECORDER_SAMPLERATE, RECORDER_CHANNELS,
        RECORDER_AUDIO_ENCODING);                           //The number of bytes you can take at the time, getMinBufferSize calculates the mini-
mum needed with the specific phone that is used
    final int REQUEST_PERMISSION = 1000;                    //To ask permission for recording again, even though this is done at the test microphone
Activity (the app did not work once because it had forgotten he had permission, so this is included)

    //Save to firebase
    private FirebaseStorage mStorageRef;
    private FirebaseFirestore db;
    private String url;
    byte[] bData;
    private String key;
    int saveFirebaseTime =10*60*1000;                       //Saying the recording to Firebase will happen every 10 minutes
    int saveOtherDataFirebaseTime=1000;                         //Saving other data to Firebase will happen every second

    //Distort recording=fast fourier transform
    double[] toTransform;
    private RealDoubleFFT transformer;                      //Make new Fast Fourier Transformer

    //EmotionSelection
    public static boolean isIsRecording() {
        return isRecording;
    }

    public static void setIsRecording(boolean isRecording) {
        Record.isRecording = isRecording;
    }

    //Calculate decibels
    private double lastPressure;                            //To calculate the decibels
    private double decibel;
    private double decibelReal;
    private static double reference = 0.00002;              //Air pressure (is reference value for microphone)

    //Calculate heart rate
    private int heartRate;                                  //For Zoltan

    //Determine place
    private String place;                                   //For Zoltan

    //ScrollChoice
    List<String> from = new ArrayList<>();
    ScrollChoice scrollChoiceFrom;
    private Handler recordHandler = new Handler();
    int startTime = 10 * 60 * 1000;                         //Time to wait before start recording when clicking start button, this is 10 min when the user has
not changed anything because I set the ScrollChoice on default on 10 min
    int stopTime = 0;                                       //Time to wait before stop recording

    //SelectTimeBox
    TextView DisplayTime;
    private int CalendarHour, CalendarMinute;
    Calendar calendar;
    TimePickerDialog timepickerdialog;

    //While Recording
    String CHANNEL_ID = "SelectEmotionNotificationChannel"; //Channel for push notification to select emotion
    String CHANNEL_ID_2 = "IsRecordingNotificationChannel"; //Channel for push notification to see the app is recording

    //rest
    private DrawerLayout drawer;
    private TextView dateTextview;
    int showNotificationTime=3*60*1000;                     //Notification will be send every hour

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_record);

        //Firebase
        mStorageRef = FirebaseStorage.getInstance();
        db = FirebaseFirestore.getInstance();


        //when you sign out of select emotion, recording will be stopped
        //since this is in this screen, the onCreate is called
        //however there is no user logged in anymore, so getting the current user will result in null

        //we make this variable to catch the mistake of logging out
        //we tried may ways to solve this, we ended up with this
        //we know it is not the best way, but ik works
        boolean userAlreadyLoggedOut = true;


        try {
            key = FirebaseAuth.getInstance().              //Get the current user
                    getCurrentUser().getUid();
        } catch (NullPointerException e) {
            //only when you come from select emotion you call the onCreate without a user
            //since we are still in the process of logging out
            //we wil redirect the user to the login page here
            Log.v("Firestore", "nullpointer after logout ");
            Intent intent = new Intent(this, LoginActivity.class);
            startActivity(intent);
            userAlreadyLoggedOut = false;
        }

        //when there is no user logged in you will not exceute this code
        //this prevents a lot of errors
        //after completing the onCreate the earlier stated intent will direct to user to the right screen
        if (userAlreadyLoggedOut) {


            //Notifications
            createNotificationChannel();                        //Create channel for notification to select emotion

            //While recording
            if (isIsRecording()) {
                intent();                                   //Go to SelectEmotion page if app is recording, user does not need to do anything on Record page
                showRecording();                                //Show in notifications bar on the phone of the user that the app is recording
            }

            //Toolbar
            //get date and set it on the actionbar
            android.support.v7.widget.Toolbar toolbarDateNurse = //Import the toolbar
                    findViewById(R.id.toolbar_date);
            dateTextview = findViewById(R.id.date);
            setSupportActionBar(toolbarDateNurse);
            Calendar cal = Calendar.getInstance();
            SimpleDateFormat sdf = new SimpleDateFormat("dd MMM, yyyy");
            String date = sdf.format(cal.getTime());
            getSupportActionBar().setTitle(date);
            dateTextview.setText(date);                         //Set the date in the toolbar

            //include drawermenu
            drawer = findViewById(R.id.menu_drawer_layout);
            ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(this, drawer,
                    R.string.navigation_drawer_open, R.string.navigation_drawer_close);
            drawer.addDrawerListener(toggle);
            toggle.syncState();

            //Name
            name = findViewById(R.id.nameTextView);
            firstLetter = findViewById(R.id.first_letter_name_user_record);

            NavigationView navigationView = (NavigationView) findViewById(R.id.menu_view);
            nameMenu = navigationView.findViewById(R.id.name_user_menu_header);
            firstLetterMenu = navigationView.findViewById(R.id.first_letter_name_user_menu_header);
            readName();                                         //Read the name from Firebase and show it on the screen

            //Recording
            (findViewById(R.id.startRecButton)).                //Set the onClickListener btnClick on the start button
                    setOnClickListener(btnClick);
            if (!checkPermissionMic()) {                        //Check if there is still permission to record
                requestPermission();                            //If not, request permission
            }

            //ScrollChoice
            scrollChoiceFrom = findViewById(R.id.fromScroll);
            loadChoices();
            scrollChoiceFrom.addItems(from, 1);     //At default, the second item is selected (0 is the first item)
            scrollChoiceFrom.setOnItemSelectedListener(        //Set a listener on the scroll choice > when the users selects something else, teh
recording should start after another time
                    new ScrollChoice.OnItemSelectedListener() {
                        @Override
                        public void onItemSelected(ScrollChoice scrollChoice, int position, String name) {
                            startTime(position);                //The position in the list decides how long the app has to wait before starting to record (0=0 sec,
1=10 min etc.)
                        }
                    });

            //SelectTimeBox
            DisplayTime = findViewById(R.id.untilTime);
            Calendar calOnCreate = Calendar.getInstance();      //Make the current time the default
            Date timeOnCreate = calOnCreate.getTime();
            SimpleDateFormat onCreate = new SimpleDateFormat("HH:mm");
            String currentTime = onCreate.format(timeOnCreate).toString();
            DisplayTime.setText(currentTime);
            (findViewById(R.id.untilTime)).
                    setOnClickListener(selectTime);             //When the user ticks on the time, (s)he has to be able to select another time
            stopTime(currentTime);                              //The default is the current time, so when the user does not change the time, this is when the
recording has to end


            //Give notification when whole app stops, this service will be running in the background and will the last thing that is done when the
app is swiped away
            Intent intent = new Intent(this, StopService.class);
            startService(intent);
        }
    }

    //Name---------------------------------------------------------------------------------------------------------------------------
    private void readName() {
        //look at firebase to see who is logged in
        Log.v("Firestore", "readName method start");
        FirebaseFirestore db = FirebaseFirestore.getInstance();

        if (key == null) {                                      //It is not possible, but when the user one way or another comes here without a key, this
code is included
            Intent intent = new Intent(this, LoginActivity.class);
            startActivity(intent);
        }

        db.collection("users").document(key).get().addOnCompleteListener(new OnCompleteListener<DocumentSnapshot>() {
            @Override
            public void onComplete(@NonNull Task<DocumentSnapshot> task) {
                if (task.isSuccessful()) {
                    DocumentSnapshot documentSnapshot = task.getResult();
                    String value = documentSnapshot.getString("name");
                    name.setText(value);
                    nameMenu.setText(value);
                    //now we set the first letter
                    String letterInPut = String.valueOf(value.charAt(0));
                    firstLetter.setText(letterInPut);
                    firstLetterMenu.setText(letterInPut);
                }
            }
        });
    }

    //Recording-----------------------------------------------------------------------------------------------------------------------
    private void startRecording() {
        recorder = new AudioRecord(MediaRecorder.AudioSource.MIC,   //Make a new recorder
                RECORDER_SAMPLERATE, RECORDER_CHANNELS,
                RECORDER_AUDIO_ENCODING, bufferSize * 2);
        Log.v(TAG, "Recording started");
        recorder.startRecording();                                  //Start recording
        setIsRecording(true);
        showRecording();                                          //Show in the notifications bar the app is recording

        recordingThread = new Thread(new Runnable() {
            public void run() {
                writeAudioDataToFile();                             //Read data from recorder and save it into a variable
            }
        }, "AudioRecorder Thread");
        recordingThread.start();
    }

    public void writeAudioDataToFile() {
        short sData[] = new short[bufferSize];//Short array to put data from buffer (audio) in

        while (isIsRecording()) {
            recorder.read(sData, 0, bufferSize);        //Save to audio data in the variable sData
            Log.v(TAG, "Short writing to file" + sData.toString());

            //Calculate decibels
            lastPressure = sData[sData.length - 1];
            double pressure = lastPressure / 9805.5336;              //This value is determined by calibrating the code with a decibel meter
            decibel = 20.0 * Math.log10(pressure / reference);
            boolean isNAN = Double.isNaN(decibel);                  //Check if decibel is a number (NaN=Not a Number=calculation went wrong)
            if (isNAN) {
                //do not update value when calculation went wrong
            } else {
                if (decibel <= 30.0) {
                    //do not update value, because this are rare mistakes that happen by using the AudioRecorder to get the decibels
                } else {
                    decibelReal = decibel;
                    Log.v(TAG, "decibel=" + decibel);
                }
            }

            //FFT
            int length = sData.length;
            toTransform = new double[length];
            transformer = new RealDoubleFFT(length);
            for (int i = 0; i < length; i++) {
                toTransform[i] = (double) sData[i] / Short.MAX_VALUE;
            }
            transformer.ft(toTransform);
            Log.v(TAG, "Transforming done");

            //Save to firebase
            bData = new byte[toTransform.length * Double.BYTES];
            for (int i = 0; i < toTransform.length; i++)            //Only bytes can be saved in Firebase, so we have to convert the doubles
                System.arraycopy(convertDoubleToByteArray(toTransform[i]), 0, bData, i * Double.BYTES, Double.BYTES);

            //Place and heartRate
            determinePlace();                                       //For Zoltan
            heartRate();                                          //For Zoltan
        }
        if (!isIsRecording()) stopRecording();

    }


    public void stopRecording() {//The interface is in here, because directly using a reference to the SelectEmotion Activity caused a circular
class referencing problem
        if (null != recorder) {
            setIsRecording(false);
            recorder.stop();
            recorder.release();
            recorder = null;                                      //Empty the recorder
            recordingThread = null;
            Log.v(TAG, "Recording stopped");
            NotificationManager mNotificationManager = (NotificationManager) getSystemService(Context.NOTIFICATION_SERVICE);
            mNotificationManager.cancel(0);                         //Delete the reminder to log your emotion
            mNotificationManager.cancel(1);                         //Delete the notification that shows the app is recording (because it is not anymore)
            Log.v(TAG, "notifications deleted");
        }
    }

    private Runnable record = new Runnable() {
        @Override
        public void run() {
            startRecording();
        }
    };
    private Runnable stopRecord = new Runnable() {
        @Override
        public void run() {
            stopRecording();
            intentRecord();
        }
    };

    private View.OnClickListener btnClick = new View.OnClickListener() {
        public void onClick(View v) {
            recordHandler.postDelayed(record, startTime);           //When the button is clicked, the recording has to begin at the time selected by
the user
            recordHandler.postDelayed(stopRecord, stopTime);        //", the recording has to stop at the time selected by the user
            recordHandler.postDelayed(showNotification, showNotificationTime);    //", the user has to get a reminder to log his/her emotion every
hour
            recordHandler.postDelayed(saveRecordingToFirebase, saveFirebaseTime);      //", the recording has to be saved to Firebase every 10
minutes
            recordHandler.postDelayed(saveOtherDataToFirebase, saveOtherDataFirebaseTime);           //", the decibels have to be saved to Fire-
base every second
            intent();                                             //", the user has to go to the SelectEmotion page, there is nothing more to be found on the Record
page for the user
        }
    };

    public void intent() {
        Intent intent = new Intent(this, SelectEmotion.class);
        startActivity(intent);
    }

    public void intentRecord() {
        Intent intentRecord = new Intent(this, Record.class);
        startActivity(intentRecord);
    }

    //Place--------------------------------------------------------------------------------------------------------------------------------
    public void determinePlace() {                                   //For Zoltan
        place = "place";
    }

    //Heart rate---------------------------------------------------------------------------------------------------------------------------
    public void heartRate() {                                    //For Zoltan
        heartRate = ConnectMio.getHeartRateValue();
    }

    //Save to Firebase-----------------------------------------------------------------------------------------------------------------------
    private Runnable saveRecordingToFirebase = new Runnable() {
        @Override
        public void run() {
            if (isIsRecording()) {
                saveToFirebaseStorage();
            }
        }
    };

    private byte[] convertDoubleToByteArray(double number) {        //Only bytes can be saved in Firebase, so we have to convert the double
array to a byte array
        ByteBuffer byteBuffer = ByteBuffer.allocate(Double.BYTES);
        byteBuffer.putDouble(number);
        return byteBuffer.array();
    }

    private void saveToFirebaseStorage() {
        //since audio files are more than 1mb we need to store them first before putting them
        //in a database. The database is not made for storing these bigger files

        //what is the current time?
        Calendar cal1 = Calendar.getInstance();
        Date time = cal1.getTime();
        SimpleDateFormat researcher = new SimpleDateFormat("dd MM yyyy HH:mm:ss");
        String soundtime = researcher.format(time).toString();

        //where the data has to go and what type of file is it
        //we make folders for each user and no more additional folders
        //this is not necessary since firestore provides the structure
        String path = "sound/" + key + "/" + soundtime + ".pcm";

        //what is the storage reference
        final StorageReference refFirestorage = mStorageRef.getReference(path);

        //Uploading to FireStorage (the bdata) and retrieving the download URL
        refFirestorage.putBytes(bData).addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {
            @Override
            public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
                refFirestorage.getDownloadUrl().addOnSuccessListener(new OnSuccessListener<Uri>() {
                    @Override
                    public void onSuccess(Uri uri) {
                        url = uri.toString();
                        Log.v(TAG, "upload to storage succesfull, link created");
                        saveToFirebaseFirestore();
                    }
                });
            }
        });
    }

    private void saveToFirebaseFirestore() {
        //now that we received the link with the place of the data we can store it in the database

        //document name is the date
        Calendar cal2 = Calendar.getInstance();
        SimpleDateFormat sdf = new SimpleDateFormat("dd MMM, yyyy");
        String date = sdf.format(cal2.getTime());

        //field name is the time
        Calendar cal3 = Calendar.getInstance();
        Date TimeNow = cal3.getTime();
        DateFormat dateFormat = new SimpleDateFormat("HH:mm:ss");
        String formattedTimeNow1 = dateFormat.format(TimeNow);

        //what is the current time? //put as value in field time
        Calendar cal1 = Calendar.getInstance();
        Date time = cal1.getTime();
        SimpleDateFormat researcher = new SimpleDateFormat("dd MM yyyy HH:mm:ss");
        String soundtime = researcher.format(time).toString();

        //making map with the fields with the corresponding values
        Map<String, Object> timeValuesSaveToFirestorage = new HashMap<>();
        timeValuesSaveToFirestorage.put("current user", key);
        timeValuesSaveToFirestorage.put("current time", soundtime);
        timeValuesSaveToFirestorage.put("link to fragment", url);

        db.collection("users").document(key).collection("sound").document(date)
                .collection("fragments").document(formattedTimeNow1).set(timeValuesSaveToFirestorage).addOnCompleteListener(new OnCom-
pleteListener<Void>() {
            @Override
            public void onComplete(@NonNull Task<Void> task) {
                if (task.isSuccessful()) {
                    Log.v(TAG, " link and data uploaded to firestore");
                } else {
                    Log.v(TAG, "upload to Firestore failed");
                }
            }
        });
        recordHandler.postDelayed(saveRecordingToFirebase, saveFirebaseTime);
        //Do this again after 10 minutes
        //we don't want to do this too often because then the structuring will be be unorganised
        //we don't want to do this too little because then we will miss data when the user
        // stops the recording
    }

    //Save decibels, place and heart rate to firebase
    private Runnable saveOtherDataToFirebase = new Runnable() {
        @Override
        public void run() {
            if (isIsRecording()) {
                saveOtherDataToFireBase(decibelReal);
                recordHandler.postDelayed(this, saveOtherDataFirebaseTime);
            }
        }
    };

    private void saveOtherDataToFireBase(final double decibel) {
        //we seperatly save the decibels for the researcher and to use them later in select emotion
        if (isIsRecording()) {
            //only when we are recording we want to save the data
            //document name is the date
            Calendar cal2 = Calendar.getInstance();
            SimpleDateFormat sdf = new SimpleDateFormat("dd MMM, yyyy");
            String date = sdf.format(cal2.getTime());

            //field name is the time
            Calendar cal3 = Calendar.getInstance();
            Date TimeNow = cal3.getTime();
            DateFormat dateFormat = new SimpleDateFormat(“HH:mm:ss”);
            String formattedTimeNow1 = dateFormat.format(TimeNow);

            //what is the current time? //put as value in field time
            Calendar cal1 = Calendar.getInstance();
            Date time = cal1.getTime();
            SimpleDateFormat researcher = new SimpleDateFormat(“dd MM yyyy HH:mm:ss”);
            String dbtime = researcher.format(time);

            //making the field with the corresponding values
            Map<String, Object> timeValuesSaveToFirestorage = new HashMap<>();
            timeValuesSaveToFirestorage.put(“current user”, key);
            timeValuesSaveToFirestorage.put(“current time”, dbtime);
            timeValuesSaveToFirestorage.put(“decibel”, decibel);
            timeValuesSaveToFirestorage.put(“place”, place);                    //For Zoltan
            timeValuesSaveToFirestorage.put(“heart rate”, heartRate);              //For Zoltan

            db.collection(“users”).document(key).collection(“relatedData”).document(date).collection(“fragments”).document(formattedTimeNow1).
set(timeValuesSaveToFirestorage).addOnCompleteListener(new OnCompleteListener<Void>() {
                @Override
                public void onComplete(@NonNull Task<Void> task) {
                    Log.v(TAG, “Decibels, place and heartRate uploaded to Firestore=” + decibel);
                }
            }).addOnFailureListener(new OnFailureListener() {
                @Override
                public void onFailure(@NonNull Exception e) {
                    Log.v(TAG, “Saving decibels, place and heartRate to Firestore failed”);
                }
            });
        }
    }

    //Check permission---------------------------------------------------------------------------------------------------------------------
    private boolean checkPermissionMic() {
        int record_audio_result = ContextCompat.checkSelfPermission(this, Manifest.permission.RECORD_AUDIO);
        return record_audio_result == PackageManager.PERMISSION_GRANTED;
    }

    private void requestPermission() {
        ActivityCompat.requestPermissions(this, new String[]{
                Manifest.permission.RECORD_AUDIO
        }, REQUEST_PERMISSION);
    }

    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {
        switch (requestCode) {
            case REQUEST_PERMISSION: {                            //Show to the user if (s)he gave access or denied it
                if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION_GRANTED)
                    Toast.makeText(this, “Permission granted”, Toast.LENGTH_SHORT).show();
                else Toast.makeText(this, “Permission denied”, Toast.LENGTH_SHORT).show();
            }
            break;
        }
    }

    //Time selection-----------------------------------------------------------------------------------------------------------------------------
    //Scroll choice (from)
    private void loadChoices() {
        from.add(“Now”);
        from.add(“in 10min”);
        from.add(“in 15min”);
        from.add(“in 30min”);
        from.add(“in 1hour”);
    }

    private int startTime(int pos) {
        if (pos == 0) {                                        //position 0=now, position 1=in 10min etc.
            startTime = 0;                                      //Start recording directly when startButton is clicked
        } else {
            if (pos == 1) {
                startTime = 60000 * 10;                            //Time is always in milliseconds
            } else {
                if (pos == 2) {
                    startTime = 60000 * 15;
                } else {
                    if (pos == 3) {
                        startTime = 60000 * 30;
                    } else {
                        startTime = 60000 * 60;
                    }
                }
            }
        }
        Log.v(TAG, “startTime=” + startTime);
        return startTime;
    }

    //SelectTimeBox (until)
    private View.OnClickListener selectTime = new View.OnClickListener() {
        public void onClick(View v) {                                //When the user clicks on the time, the TimeBox must be shown which automatically goes
to the current time
            calendar = Calendar.getInstance();
            CalendarHour = calendar.get(Calendar.HOUR_OF_DAY);
            CalendarMinute = calendar.get(Calendar.MINUTE);

            timepickerdialog = new TimePickerDialog(Record.this,
                    3,                                  //The third theme fittest best to our app
                    new TimePickerDialog.OnTimeSetListener() {
                        @Override
                        public void onTimeSet(TimePicker view, int hourOfDay, int minute) {
                            String endTime = String.format(“%02d:%02d”, //%02d makes sure the time on the textView is set to e.g. 07:05 instead of 7:5
when the user means five minutes past seven
                                    hourOfDay, minute);
                            DisplayTime.setText(endTime);
                            stopTime(endTime);
                        }
                    }, CalendarHour, CalendarMinute, true); //true means we use the 24h clock
            timepickerdialog.show();
        }
    };

    private int stopTime(String endTime) {                                  //Calculate the time the recorder has to record by using the current time and the
endTime selected by the user
        Log.v(TAG, “endTime=” + endTime);
        Calendar cal = Calendar.getInstance();
        Date TimeNow = cal.getTime();
        DateFormat dateFormat = new SimpleDateFormat(“HH:mm”);
        String formattedTimeNow = dateFormat.format(TimeNow);
        Log.v(TAG, “Now=” + formattedTimeNow);
        Date now = null;
        Date end = null;
        try {
            now = dateFormat.parse(formattedTimeNow);
            end = dateFormat.parse(endTime);
            long diff = end.getTime() - now.getTime();
            stopTime = (int) diff;
            Log.v(TAG, “stopTime=” + stopTime);
        } catch (ParseException e) {
            e.printStackTrace();
            Log.v(TAG, “Formatting time failed”);
        }
        return stopTime;
    }

    //Push notifications------------------------------------------------------------------------------------------------------------------------
    private Runnable showNotification = new Runnable() {
        @Override
        public void run() {
            if (isIsRecording()) {
                showNotification();
                recordHandler.postDelayed(this, showNotificationTime);
            }
        }
    };

    private void createNotificationChannel() {
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
            CharSequence name = “SelectEmotion”;
            int importance = NotificationManager.IMPORTANCE_HIGH;       //It is important the user does not forget to log his/her emotions
            NotificationChannel channel = new NotificationChannel(CHANNEL_ID, name, importance);
            NotificationManager notificationManager = getSystemService(NotificationManager.class);
            notificationManager.createNotificationChannel(channel);
        }
    }

    private void showNotification() {
        Intent intent = new Intent(this, SelectEmotion.class);
        intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK | Intent.FLAG_ACTIVITY_CLEAR_TASK);   //This makes sure that the phone sees the
SelectEmotion page as the beginning of the app use now, so the phone so not go back to Record when the user uses the back button
        PendingIntent pendingIntent = PendingIntent.getActivity(this, 0,  //When the user ticks on the notification, (s)he is directed to the SelectE-
motion page
                intent, 0);

        NotificationCompat.Builder builder = new NotificationCompat.Builder(this, CHANNEL_ID)
                .setSmallIcon(R.drawable.logo_cacophony)
                .setContentTitle(“Share your cacophony”)
                .setStyle(new NotificationCompat.BigTextStyle().                         //BigTextStyle allows the notification to be two rows of text instead
of being cut off on smaller phones
                        bigText(“How was your cacophony experience in the last hour?”))
                .setPriority(NotificationCompat.PRIORITY_HIGH)                        //It is important the user does not forget to log his/her emotions
                .setContentIntent(pendingIntent)                                  //Intent to SelectEmotion page
                .setAutoCancel(true);                                          //Notification is deleted when the user ticks on it

        if (isIsRecording()) {
            NotificationManagerCompat notificationManager = NotificationManagerCompat.from(this);
            notificationManager.notify(0, builder.build());                          //Notification ID is unique for every notification and will be used to delete
the right notifications at the right time
            Log.v(TAG, “Notification send”);
        }
    }

    //While Recording
    public void showRecording() {                                   //Notification to user to show the app is recording
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
            CharSequence name = “isRecording”;
            int importance = NotificationManager.IMPORTANCE_DEFAULT;
            NotificationChannel channel2 = new NotificationChannel(CHANNEL_ID_2, name, importance);
            NotificationManager notificationManager = getSystemService(NotificationManager.class);
            notificationManager.createNotificationChannel(channel2);
        }
        NotificationCompat.Builder builder2 = new NotificationCompat.Builder(this, CHANNEL_ID_2)
                .setSmallIcon(R.drawable.noun_recording_1614606)
                .setContentTitle(“Recording”)
                .setPriority(NotificationCompat.PRIORITY_DEFAULT)
                .setAutoCancel(true);

        if (isIsRecording() == true) {
            NotificationManagerCompat notificationManager2 = NotificationManagerCompat.from(this);
            notificationManager2.notify(1, builder2.build());
        }
    }
    //Rest----------------------------------------------------------------------------------------------------------------------------
    public void inflateMenuButtonClicked(View view) {
        //if the menu drawer is open close the drawer when the button is clicked
        //if the menu drawer is closed open the drawer when the button is clicked
        if (drawer.isDrawerOpen(GravityCompat.END)) {
            drawer.closeDrawer(GravityCompat.END);
        } else if (!drawer.isDrawerOpen(GravityCompat.END)) {
            drawer.openDrawer(GravityCompat.END);
        }
    }

    public void logOutButtonClicked(View view) {
        FirebaseAuth.getInstance().signOut();
        finish();                               //Make sure we finish this activity and ‘start over’ again
        Toast.makeText(this, “Logged out”,
                Toast.LENGTH_LONG).show();
        drawer.closeDrawer(GravityCompat.END);
        Intent intent = new Intent(this, LoginActivity.class);
        startActivity(intent);
    }

    //Back pressed modification-------------------------------------------------------------------------------------------------------------------
    @Override
    public void onBackPressed() {
        //If the back button is pressed while the menu drawer is open do not close the activity
        if (drawer.isDrawerOpen(GravityCompat.END)) {
            drawer.closeDrawer(GravityCompat.END);
        } else {
            super.onBackPressed();
        }
    }
```

```
package nl.dut.ide.software.soundfilterrealgit;

import android.graphics.Bitmap;
import android.graphics.Canvas;

import android.graphics.Paint;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.ImageView;
//Graph1
import android.widget.FrameLayout;
import com.jjoe64.graphview.BarGraphView;
import com.jjoe64.graphview.GraphView;
import com.jjoe64.graphview.GraphViewSeries;
import com.jjoe64.graphview.LineGraphView;
import android.graphics.Color;

public class Review extends AppCompatActivity {
    Canvas canvas;
    ImageView imageView;
    Paint paint;
    //Graph1
//      private FrameLayout mLayout=null;
//      private GraphViewSeries mGraphSeries=null;
//      private boolean isGraph=true;5

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_review);
        imageView=(ImageView)this.findViewByld(R.id.graph);
        Bitmap bitmap=Bitmap.createBitmap(256,100, Bitmap.Config.ARGB_8888);
        canvas=new Canvas(bitmap);
        paint=new Paint();
        paint.setColor(getResources().getColor(R.color.colorPrimaryDark));
        imageView.setImageBitmap(bitmap);

        //Graph1
//          mLayout=(FrameLayout)findViewByld(R.id.frameLayout);
    }

    public void publishProgress (double[]toTransform){
        for (int i = 0; i < toTransform.length; i++) {
            int x = i;
            int downy = (int) (100 - (toTransform[i] * 10));
            int upy = 100;

            canvas.drawLine(x, downy, x, upy, paint);
        }

        imageView.invalidate();
    }

        //Graph1
//      @Override
//      protected void onStart() {
//          super.onStart();
//          initGraphView();
//      }
//
//      private void initGraphView(){
//          GraphView graphView;
//          graphView = new BarGraphView(this, “Frequency Domain”);
//
//          mGraphSeries = new GraphViewSeries(new GraphView.GraphViewData[]{});
//          graphView.addSeries(mGraphSeries);
//
//          if (mLayout.getChildCount() > 0) {
//              mLayout.removeAllViews();
//          }
//          mLayout.addView(graphView);
//      }
//
//      public void updateGraphView(double[] toTransform,int length,short[]sData){
//          final GraphView.GraphViewData[] data = new GraphView.GraphViewData[length];
//
//              for (int i = 0; i < length; i++) {
//                  data[i] = new GraphView.GraphViewData(i, sData[i]);
//              }
//          Review.this.runOnUiThread(new Runnable() {
//              @Override
//              public void run() {
//                  mGraphSeries.resetData(data);
//              }
//          });
//      }
//
//      public void resetGraph(){
//          mGraphSeries.resetData(new GraphView.GraphViewData[]{});
//      }
}
```

Select
Emotion

```
package nl.dut.ide.software.soundfilterrealgit;

import android.content.Intent;
import android.support.annotation.NonNull;
import android.support.design.widget.NavigationView;
import android.support.v4.view.GravityCompat;
import android.support.v4.widget.DrawerLayout;
import android.support.v7.app.ActionBarDrawerToggle;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.firestore.DocumentReference;
import com.google.firebase.firestore.DocumentSnapshot;
import com.google.firebase.firestore.FirebaseFirestore;
import com.google.firebase.firestore.QueryDocumentSnapshot;
import com.google.firebase.firestore.QuerySnapshot;

import java.text.DateFormat;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.Date;
import java.util.HashMap;
import java.util.Map;
import java.util.TreeMap;

//Zoltan, search for ctrl+F ‘Zoltan’  to find the stuff about the decibels taken from 30 seconds ago instead of the highest

public class SelectEmotion extends AppCompatActivity{

    private TextView nameMenu,firstLetterMenu;
    private Button emotionButton;
    private int keyEmotion = -1;
    private DrawerLayout drawer;
    private TextView dateTextview;
    private int errorCounter;
    private Record record=new Record();
    String TAG=”SelectEmotion”;
    TreeMap<Integer,String>emotionsAndTimeMap=new TreeMap<>();
    TreeMap<Integer,String>getEmotionsAndTimeDatabaseMap=new TreeMap<>();


    //Firebase
    private FirebaseFirestore db;
    private String key;

    //Decibels
    Double deci;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_select_emotion);

        //Firebase
        db = FirebaseFirestore.getInstance();
        key= FirebaseAuth.getInstance().              //Get the current user
                getCurrentUser().getUid();

        //get date and set it on the actionbar
        dateTextview=findViewById(R.id.date);
        Calendar cal=Calendar.getInstance();
        SimpleDateFormat sdf=new SimpleDateFormat(“dd MMM, yyyy”);
        String date=sdf.format(cal.getTime());
        dateTextview.setText(date);

        //include drawermenu
        drawer = findViewById(R.id.menu_drawer_layout_emotion);
        ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(this, drawer,
                R.string.navigation_drawer_open, R.string.navigation_drawer_close);


        //Finding the right textview to set the name of the user later in readname()
        NavigationView navigationViewEmotion = (NavigationView) findViewById(R.id.menu_view_emotion);
        nameMenu = navigationViewEmotion.findViewById(R.id.name_user_menu_header_emotion);
        firstLetterMenu = navigationViewEmotion.findViewById(R.id.first_letter_name_user_menu_header_emotion);
        readName();
    }
    //Name------------------------------------------------------------------------------------------------------------------------
    private void readName() {
        //We will check who the current logged in user is in Firebase
        Log.v(“Firestore”, “readName method start”);

        Log.v(TAG,”key=”+key);

        db.collection(“users”).document(key).get().addOnCompleteListener(new OnCompleteListener<DocumentSnapshot>() {
            @Override
            public void onComplete(@NonNull Task<DocumentSnapshot> task) {
                if(task.isSuccessful()){
                    DocumentSnapshot documentSnapshot=task.getResult();
                    String value= documentSnapshot.getString(“name”);
                    nameMenu.setText(value);

                    //now we set the first letter
                    String letterInPut = String.valueOf(value.charAt(0));
                    firstLetterMenu.setText(letterInPut);
                }
            }
        });
    }

    //Emotion clicked---------------------------------------------------------------------------------------------------------------------------
    public void emotionClicked(View v) throws ParseException {
        //the clicked emotion button
        emotionButton = (Button) v;

        Calendar cal5 = Calendar.getInstance();
        Date TimeNow = cal5.getTime();
        DateFormat dateFormat = new SimpleDateFormat(“HH:mm:ss”);
        String formattedTimeNow2 = dateFormat.format(TimeNow);

        //get emotion of the button that is clicked
        String emotionSelected = String.valueOf(emotionButton.getText().toString());
        Log.d(TAG,emotionSelected);

        //give emotion to the show emotion method to add time and show it in the listview
        showEmotion(emotionSelected,formattedTimeNow2);

        //timeEmotionResearcher to firebase
        saveEmotionToFirebaseFirestore(emotionSelected,formattedTimeNow2);

        //save the decibels level of when the button was clicked
        getDecibel();
    }

    private void getDecibel() throws ParseException {
        //we will store the decibels every seconds for research purposes
        //we will also se this data to connect the clicked emotion to a decibel level
        //to find the right time we will have to get the data to make the right path to the decibel data

        //document name in Firebase is the date
        Calendar cal4 = Calendar.getInstance();
        SimpleDateFormat sdf = new SimpleDateFormat(“dd MMM, yyyy”);
        String date = sdf.format(cal4.getTime());

        //field name is the time
        DateFormat dateFormat = new SimpleDateFormat(“HH:mm:ss”);
        String formattedTimeNow2 = dateFormat.format(new Date());
        Date date2 = dateFormat.parse(formattedTimeNow2);
        Calendar cal5 = Calendar.getInstance();
        cal5.setTime(date2);
        cal5.add(Calendar.SECOND, -30);                    //Zoltan, this is where I decide I take the decibels from 30 seconds ago
        Date datereal=cal5.getTime();
        String formattedTimeNow3 = dateFormat.format(datereal);
        Log.v(TAG,”formattedTimenow3=”+formattedTimeNow3);

        DocumentReference dbref=db.collection(“users”).document(key).collection(“relatedData”)
                .document(date).collection(“fragments”).document(formattedTimeNow3);

        dbref.get().addOnSuccessListener(new OnSuccessListener<DocumentSnapshot>() {
            @Override
            public void onSuccess(DocumentSnapshot documentSnapshot) {
                if(documentSnapshot.exists()){
                    //when there is a document at th stated time
                    //we will store the data in a variable in java
                    deci=documentSnapshot.getDouble(“decibel”);     //Zoltan, here I get the decibel so I think you should get an array here
                    Log.v(TAG,”decibels=”+deci);
                }else{
                    //there is no document
                    // what to do then has to be explored further
                    Log.v(TAG,”No document for decibels”);
                }
            }
        }).addOnFailureListener(new OnFailureListener() {
            @Override
            public void onFailure(@NonNull Exception e) {
                Log.v(TAG,”Getting decibels failed”);
            }
        });
        //connecting to firebase failed
    }

    private void showEmotion(String emotionSelected,String timeEmotionClicked){
        //get time that emotion was clicked
        SimpleDateFormat user=new SimpleDateFormat(“HH:mm”);
        Calendar cal = Calendar.getInstance();
        Date time=cal.getTime();
        String timeEmotionUser=user.format(time);

        //put emotion and time to the treemap
        emotionsAndTimeMap.put(keyEmotion,emotionSelected+” “+timeEmotionUser);
        getEmotionsAndTimeDatabaseMap.put(keyEmotion,timeEmotionClicked);

        //lower the key with one so that the new entries in the treemap will be automatically sorted
        //on top of the treemap, so that the order does not have to be switched when inserting it in
        //the the listview
        keyEmotion=keyEmotion-1;

        //find the listview where the emotions and times should be displayed
        //give the treemap with the emotions and times to the EmotionListAdapter
        //set the view from the adapter to the listview where it should be displayed
        ListView listView = findViewById(R.id.list_emotion);
        EmotionListAdapter adapter = new EmotionListAdapter(emotionsAndTimeMap);
        listView.setAdapter(adapter);
    }

    public void saveEmotionToFirebaseFirestore(String emotionData, String timeEmotionClicked) {
        //save the data connected to the right user

        //document name is the date
        Calendar cal4 = Calendar.getInstance();
        SimpleDateFormat sdf = new SimpleDateFormat(“dd MMM, yyyy”);
        String date = sdf.format(cal4.getTime());

        //what is the current time? //put as value in field time
        Calendar cal1 = Calendar.getInstance();
        Date time=cal1.getTime();
        SimpleDateFormat researcher=new SimpleDateFormat(“dd/MM/yyyy HH:mm:ss”);
        String emotiontime = researcher.format(time);

        //making the fields and set the corresponding values
        //we put them all in one map since we can not add new fields once a document is created
        Map<String, Object> emotionValuesSaveToFirestorage = new HashMap<>();
        emotionValuesSaveToFirestorage.put(“current user”,key);
        emotionValuesSaveToFirestorage.put(“current time”,emotiontime);
        emotionValuesSaveToFirestorage.put(“emotion data”,emotionData);
        emotionValuesSaveToFirestorage.put(“decibel”,deci);

        db.collection(“users”).document(key).collection(“emotion”).document(date)
                .collection(“fragments”).document(timeEmotionClicked).set(emotionValuesSaveToFirestorage).addOnCompleteListener(new OnCom-
pleteListener<Void>() {
            @Override
            public void onComplete(@NonNull Task<Void> task) {
                if (task.isSuccessful()) {
                    //what to do when you uploaded the data?
                    Log.v(TAG, “ link and data uploaded to firestore EMOTION”);
                    //we will not inform the user about this, this will only distract them
                } else {
                    //what to do when failed to upload the data
                    Log.v(TAG, “upload to firestore failed EMOTION :(“);
                    //inform the user and ask them to try again
                    Toast.makeText(SelectEmotion.this, “Logging emotion failed, please try again”, Toast.LENGTH_LONG).show();
                }
            }
        });

    }

    //Delete emotion------------------------------------------------------------------------------------------------------------------------
    public void deleteEmotionClicked(View view){
        //get the button that was clicked (the listview has many buttons that call this method)
        LinearLayout vwParentRow = (LinearLayout)view.getParent();
        Button btnChild = (Button)vwParentRow.getChildAt(1);

        //get the tag that is given to the button and remove the entry of the treemap
        //with the same key value as the tag
        Log.d(“Tag”, String.valueOf(btnChild.getTag()));
        emotionsAndTimeMap.remove(btnChild.getTag());
        String timeEmotionClicked=getEmotionsAndTimeDatabaseMap.get(btnChild.getTag());

        //Refresh the view of the listview, without the just deleted entry pair of the treemap
        ListView listView = findViewById(R.id.list_emotion);
        EmotionListAdapter adapter = new EmotionListAdapter(emotionsAndTimeMap);
        listView.setAdapter(adapter);

        deleteEmotion(timeEmotionClicked);
    }

    public void deleteEmotion(String timeEmotionClicked) {
        //receive the time emotion clicked since this forms the reference
        // to the right emotion that has to be deleted

        //document name is the date
        Calendar cal4 = Calendar.getInstance();
        SimpleDateFormat sdf = new SimpleDateFormat(“dd MMM, yyyy”);
        String date = sdf.format(cal4.getTime());

        db.collection(“users”).document(key).collection(“emotion”).document(date)
                .collection(“fragments”).document(timeEmotionClicked)
                .delete()
                .addOnSuccessListener(new OnSuccessListener<Void>() {
                    @Override
                    public void onSuccess(Void aVoid) {
                        Toast.makeText(SelectEmotion.this, “Emotion deleted successfully”, Toast.LENGTH_LONG).show();
                        Log.v(TAG, “Emotion deleted successfully!”);
                    }
                })
                .addOnFailureListener(new OnFailureListener() {
```

```
                @Override
                public void onFailure(@NonNull Exception e) {
                    Toast.makeText(SelectEmotion.this, "Could not delete emotion", Toast.LENGTH_SHORT).show();
                    Log.w(TAG, "Error deleting document", e);
                }
            });
    }

    //Finish today----------------------------------------------------------------------------------------------------------------------------
    public void finishTodayButtonClicked(View view) {
        Record.setIsRecording(false);
        //Go to the recording screen
        Intent intent = new Intent(this, Record.class);
        startActivity(intent);
    }

    //Menu right side-------------------------------------------------------------------------------------------------------------------------
    public void inflateMenuButtonClicked(View view) {
        //if the menu drawer is open close the drawer when the button is clicked
        //if the menu drawer is closed open the drawer when the button is clicked
        if (drawer.isDrawerOpen(GravityCompat.END)){
            drawer.closeDrawer(GravityCompat.END);
        }else if(!drawer.isDrawerOpen(GravityCompat.END)){
            drawer.openDrawer(GravityCompat.END);
        }
    }

    //Back pressed---------------------------------------------------------------------------------------------------------------------------
    @Override
    public void onBackPressed() {
        //If the back button is pressed while the menu drawer is open do not close the activity
        if (drawer.isDrawerOpen(GravityCompat.END)) {
            drawer.closeDrawer(GravityCompat.END);
        }
        // else do no nothing, only when recording is stopped the user is allowed to go back
        //or when they press the finish today button
    }

    //Log out----------------------------------------------------------------------------------------------------------------------------------
    public void logOutButtonClickedEmotion(View view) {
        FirebaseAuth.getInstance().signOut();
        Log.v("FirebaseAuth", "user logged out on firebase ");
        Record.setIsRecording(false);
        Log.v("please", "stop recording had happened ");

        // make sure we finish this activity and 'start over' again
        // it wont be possible to use the back button to come back to this screen
        finish();
        Toast.makeText(this, "Logged out", Toast.LENGTH_LONG).show();
        drawer.closeDrawer(GravityCompat.END);

//        go the the login activity again
        Intent intentEmotionSignOut = new Intent(this, LoginActivity.class);
        startActivity(intentEmotionSignOut);
    }

}
```

Stop
Service

```
package nl.dut.ide.software.soundfilterrealgit;

import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.app.Service;
import android.content.Context;
import android.content.Intent;
import android.os.Build;
import android.os.IBinder;
import android.support.annotation.Nullable;
import android.support.v4.app.NotificationCompat;

public class StopService extends Service {
    @Nullable
    @Override
    public IBinder onBind(Intent intent) {
        return null;
    }

    String CHANNEL_ID_3 = "RecordingStoppedNotificationChannel";

    @Override
    public void onTaskRemoved(Intent rootIntent) {                  //When the app is moved away from the list by the user
        super.onTaskRemoved(rootIntent);

        NotificationManager mNotificationManager = (NotificationManager) getSystemService(Context.NOTIFICATION_SERVICE);
        mNotificationManager.cancel(0);                          //Delete the reminder to log your emotion
        mNotificationManager.cancel(1);                          //Delete the notification that shows the app is recording (because it is not anymore)

        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {       //Create a notification that says the app stopped, so when the working shift
of the user had not finished (s)he can start the recording again (app had stopped accidentally)
            CharSequence name = "recordingStopped";
            int importance = NotificationManager.IMPORTANCE_HIGH;
            NotificationChannel channel3 = new NotificationChannel(CHANNEL_ID_3, name, importance);
            NotificationManager notificationManager2 = getSystemService(NotificationManager.class);
            notificationManager2.createNotificationChannel(channel3);
        }
        NotificationCompat.Builder builder3 = new NotificationCompat.Builder(this, CHANNEL_ID_3)
                .setSmallIcon(R.drawable.logo_cacophony)
                .setContentTitle("Recording stopped")
                .setPriority(NotificationCompat.PRIORITY_HIGH)
                .setAutoCancel(true)
                .setStyle(new NotificationCompat.BigTextStyle().bigText("When your shift has not finished yet, please start the recording again."));

        mNotificationManager.notify(3, builder3.build());
    }
}
```

Test
Mic

```
package nl.dut.ide.software.soundfilterrealgit;

import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.graphics.Bitmap;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.media.MediaRecorder;
import android.os.Bundle;
import android.os.Environment;
import android.support.annotation.NonNull;
import android.support.v4.app.ActivityCompat;
import android.support.v4.content.ContextCompat;
import android.support.v7.app.AppCompatActivity;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import java.io.File;
import java.io.IOException;

public class TestMic extends AppCompatActivity {
    private TextView infoTextView;
    private MediaRecorder mTestRecorder = null;                   //Initialize a new media recorder
    private ImageView testMicButton,hiddenView;
    private Button continueButton;
    private int testFailedTimes=0;
    private int soundTooLowTimes=0;
    private String TAG=”Mic_test”;                         //TAG is used for Logcat, to easily collect the text send to myself
    private String on_off=”off”;
    final int REQUEST_PERMISSION=1000;
    private String test=””;
    private static boolean permission=false;

    public static boolean isPermission() {
        return permission;
    }

    public static void setPermission(boolean permission) {
        TestMic.permission = permission;
    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_test_mic);
        infoTextView=findViewByld(R.id.textTestMicResult);
        testMicButton=findViewByld(R.id.testMicButton);
        continueButton=findViewByld(R.id.continueButton);
        hiddenView=findViewByld(R.id.hiddenView);
        if(checkPermissionMic()){                             //When there is permission to use the MIC, the test can start
            testMicButton.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View v) {
                    if(on_off==”off”){                       //When app is not recording, start recording when button is pushed
                        if(testFailedTimes<3 && soundTooLowTimes<3) {   //When microphone test has not yet been done 3 times
                            Bitmap bitmap=recordingOnColour();          //Make the button red, like a real recorder
                            hiddenView.setImageBitmap(bitmap);
                            hiddenView.setVisibility(View.VISIBLE);
                            infoTextView.setText(“Recording started”);
                            on_off=”on”;
                            recordStart();                    //And start recording
                        }
                        else{
                            infoChange(“Reinstall the app or let your microphone be checked by a mechanic”);
                            testMicButton.setEnabled(false);           //You cannot test anymore
                        }
                    }
                    else{                                //When app is recording, stop recording when button is pushed
                        infoTextView.setText(“Recording ended”);
                        hiddenView.setVisibility(View.INVISIBLE);      //Delete the red circle on the recording button
                        recordStop();                          //Stop recording
                        checkAmplitude();                        //Check if the sound is loud enough
                        on_off=”off”;
                    }
                }
            });
        }
        else{
            requestPermission();
            testMicButton.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View v) {
                    if(on_off==”off”){                       //When app is not recording, start recording when button is pushed
                        if(testFailedTimes<3 && soundTooLowTimes<3) {   //When microphone test has not yet been done 3 times
                            Bitmap bitmap=recordingOnColour();          //Make the button red, like a real recorder
                            hiddenView.setImageBitmap(bitmap);
                            hiddenView.setVisibility(View.VISIBLE);
                            infoTextView.setText(“Recording started”);
                            on_off=”on”;
                            recordStart();                    //And start recording
                        }
                        else{
                            infoChange(“Reinstall the app or let your microphone be checked by a mechanic”);
                            testMicButton.setEnabled(false);           //You cannot test anymore
                        }
                    }
                    else{                                //When app is recording, stop recording when button is pushed
                        infoTextView.setText(“Recording ended”);
                        hiddenView.setVisibility(View.INVISIBLE);      //Delete the red circle on the recording button
                        recordStop();                          //Stop recording
                        checkAmplitude();                        //Check if the sound is loud enough
                        on_off=”off”;
                    }
                }
            });
        }
    }

    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {
        switch (requestCode){
            case REQUEST_PERMISSION:                              //Show to user if (s)he gave/denied permission
            {
                if(grantResults.length>0 && grantResults[0]==PackageManager.PERMISSION_GRANTED&&grantResults[1]==PackageManager.PER-
MISSION_GRANTED)
                {Toast.makeText(this,”Permission granted”,Toast.LENGTH_SHORT).show();
                Log.v(“toast”, “accepted”);
                setPermission(true);}
                else {
                    intent();
                    Log.v(“toast”, “denied”);
                    Toast.makeText(this,”Permission is needed to use this app”,Toast.LENGTH_LONG).show();}
            }
            break;
        }
    }

    private void requestPermission() {
        ActivityCompat.requestPermissions(this,new String[]{
                Manifest.permission.RECORD_AUDIO,Manifest.permission.WRITE_EXTERNAL_STORAGE
        },REQUEST_PERMISSION);
    }
    private boolean checkPermissionMic() {
        int record_audio_result= ContextCompat.checkSelfPermission(this,Manifest.permission.RECORD_AUDIO);
        int write_external_storage_result= ContextCompat.checkSelfPermission(this,Manifest.permission.WRITE_EXTERNAL_STORAGE);
        return record_audio_result==PackageManager.PERMISSION_GRANTED && write_external_storage_result==PackageManager.PERMIS-
SION_GRANTED;
    }

    public void continueButtonClicked(View view){
        Intent micWorksIntent = new Intent(this,Record.class);
        micWorksIntent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
        startActivity(micWorksIntent);
    }

    private void intent(){
        Intent intent = new Intent(this, LoginActivity.class);
        startActivity(intent);
    }

    private void recordStart() {
        if (mTestRecorder == null) {
            Log.v(TAG,”recording”);
            test = Environment.getExternalStorageDirectory().
                    getAbsolutePath()+”/mic_test_record.3gp”;              //Place to save the audio file temporarily
            mTestRecorder = new MediaRecorder();                          //Make a new media recorder
            mTestRecorder.setAudioSource(MediaRecorder.AudioSource.MIC);
            mTestRecorder.setOutputFormat(MediaRecorder.OutputFormat.THREE_GPP);
            mTestRecorder.setAudioEncoder(MediaRecorder.AudioEncoder.AMR_NB);
            mTestRecorder.setOutputFile(test);

            try {                                                 //Try-catch method must be used to use prepare()
                mTestRecorder.prepare();
                mTestRecorder.start();
                mTestRecorder.getMaxAmplitude();
            } catch (IOException e) {
                Log.e(TAG,”prepare() failed”);
            }
        }
    }

    private void recordStop() {
        if (mTestRecorder != null) {
            Log.v(TAG,”recording stopped”);
            mTestRecorder.stop();
        }
    }

    private void checkAmplitude() {
        if (mTestRecorder != null) {
            Log.v(TAG,”calculating amplitude”);
            double REFERENCE=0.00002;                              //Air pressure (is reference value for microphone)
            double amplitude = mTestRecorder.getMaxAmplitude();
            Log.v(TAG,”amplitude=”+amplitude);
            double pressure = amplitude / 90000.5336;                   //This value is determined by calibrating the code with a decibel meter
            Log.v(TAG,”pressure=”+pressure);
            double db = (20 * Math.log10(pressure / REFERENCE));    //Formula used by everyone on the internet
            String decibels=Double.toString(db);
            Log.v(TAG,”decibels=”+decibels);
            if (db >= 45) {                                      //Determined by experiment and internet research
                infoChange(“Sound level approved”);
                continueButton.setVisibility(View.VISIBLE);         //When the sound button was good, the user is allowed to continue to the real app
                File file = new File(Environment.getExternalStorageDirectory().getAbsolutePath()+”/mic_test_record.3gp”);
                boolean isSuccess = file.delete();                  //Delete the audio file from the phone
                Log.v(TAG,”file deleted from phone”);
            } else {
                infoChange(“Sound level too low, put your phone on a different place”);
                soundTooLowTimes = soundTooLowTimes + 1;
            }
        }
        else {
            infoChange(“Sound test failed, try again”);
            testFailedTimes=testFailedTimes+1;
            Log.e(TAG,”nothing recorded”);
        }
        mTestRecorder = null;
    }

    private void infoChange (String info){                           //Method used to show different texts to the user
        infoTextView.setText(info);
    }

    private Bitmap recordingOnColour(){                               //Create red circle on recording icon to make it look real
        Bitmap bitmap=Bitmap.createBitmap(200,200, Bitmap.Config.ARGB_8888);
        Canvas canvas=new Canvas(bitmap);
        Paint paint=new Paint(Paint.ANTI_ALIAS_FLAG);
        paint.setColor(Color.RED);
        canvas.drawCircle(100,100,75,paint);
        return bitmap;
    }
}
```

User test: SUS(System Usability Scale) Survey

Cacophony Mapper: Defining Noise Fatigue in the ICU

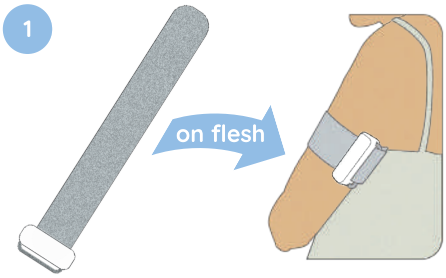
Name : Jiwon

Gender : Female **Age :** 26

Score : 11 **/ 12**

The purpose of this project is to find how sound stimuli can affect nurses' stress level and emotional responses in the ICU. Thus, you are going to be asked to wear a heart rate detection device and a sound collector during the test. Sound data will be distorted using a filter while recording, so no one can listen or restore the original sound that you don't need to worry about the privacy issue. The data will be only used for academic purpose. Please follow the instruction below and rate your usability of each process. The questionnaire on the next page should be filled in after the test. The duration of the recordning will be 12 minutes and you do not need to try to remember the sound from the recording. Thank you for your participation and enjoy!

1



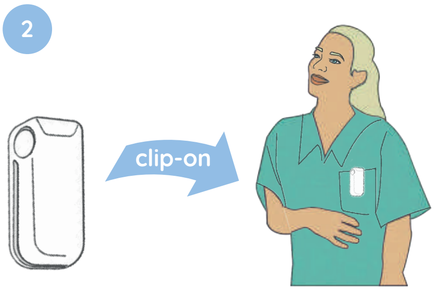
Please fasten the armband on your outer side of the bare upper arm for the heart rate collection.

Easy — **Okay** — **Difficult**

[✓ 2] 1 0

Why?

2



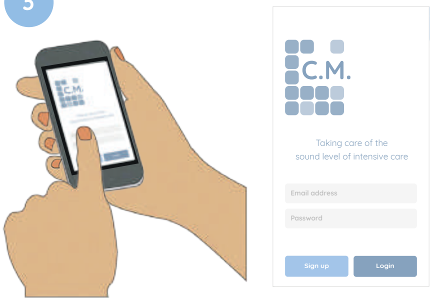
Please wear the clip the microphone device on your chest area.

Easy — **Okay** — **Difficult**

[✓ 2] 1 0

Why?

3



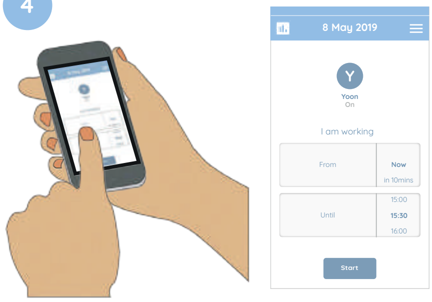
Please Sign in to Cacophony Mapper application, using Email and password written below:
Emai l: designer.yoon.lee@gmail.com
Password : dldbs

Easy — **Okay** — **Difficult**

[✓ 2] 1 0

Why?

4



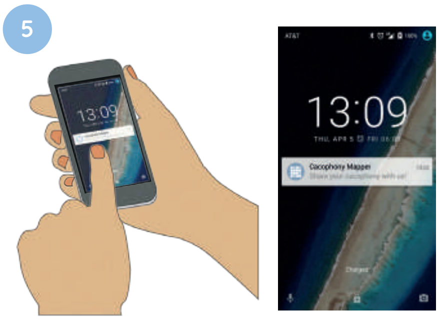
Please **start HB button** to connect the device to the Bluetooth. Assign the time slot **from now to 12 minutes** later and press **ttart rec button** and enjoy your work! :)

Easy — **Okay** — **Difficult**

[✓ 2] 1 0

Why?

5



When there is a notification, please start the application by swiping the notification from the left to the rigt and report your emotion.

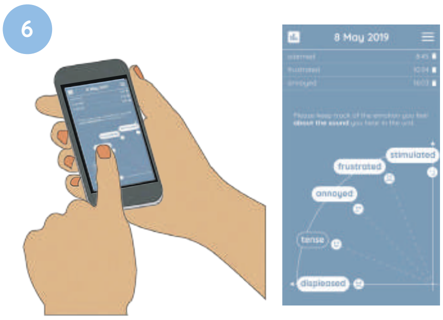
Easy — **Okay** — **Difficult**

2 [✓ 1] 0

Why?

Regular intervention made me difficult to concentrate on my work.

6



Please press the emotion button which was relevant to your emotion related to your recent sound experience.

Easy — **Okay** — **Difficult**

[✓ 2] 1 0

Why?

23/25

The SUS(System Usability Scale) Survey for Cacophony Mapper application

1. Cacophony Mapper application was manageable to learn.

Strongly disagree ○ 1 ○ 2 ○ 3 ○ 4 ☑ 5 Strongly agree

Why do you think so?

2. Cacophony Mapper application was clearly structured.

Strongly disagree ○ 1 ○ 2 ○ 3 ○ 4 ☑ 5 Strongly agree

Why do you think so?

3. Cacophony Mapper application was complicated to use.

Strongly disagree ☑ 1 ○ 2 ○ 3 ○ 4 ○ 5 Strongly agree

Why do you think so?

4. Cacophony Mapper interface was straghtforward to use.

Strongly disagree ○ 1 ○ 2 ○ 3 ○ 4 ☑ 5 Strongly agree

Why do you think so?

5. I think it will be practical to used Cacophony Mapper during work in a medical environment.

Strongly disagree ○ 1 ○ 2 ☑ 3 ○ 4 ○ 5 Strongly agree

Why do you think so?

Interval for the emotion report for this testing was too short for me. I had to switch my concentration too often.

Comments? App itself is very easy to use but I think nurses may feel to disrupted by having notification frequently.

Cacophony Mapper: Defining Noise Fatigue in the ICU

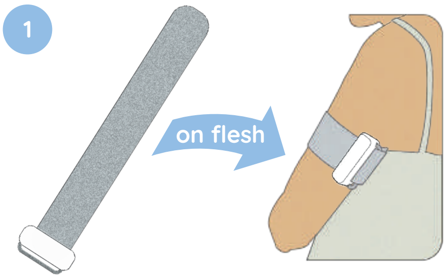
Name : Sanne

Gender : Female **Age :** 21

Score : 12 **/ 12**

The purpose of this project is to find how sound stimuli can affect nurses' stress level and emotional responses in the ICU. Thus, you are going to be asked to wear a heart rate detection device and a sound collector during the test. Sound data will be distorted using a filter while recording, so no one can listen or restore the original sound that you don't need to worry about the privacy issue. The data will be only used for academic purpose. Please follow the instruction below and rate your usability of each process. The questionnaire on the next page should be filled in after the test. The duration of the recordning will be 12 minutes and you do not need to try to remember the sound from the recording. Thank you for your participation and enjoy!

1



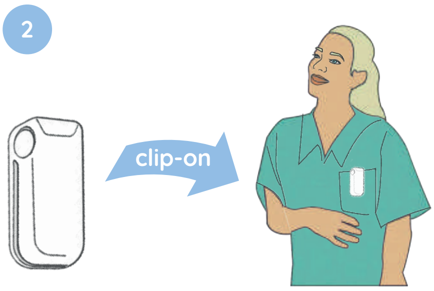
Please fasten the armband on your outer side of the bare upper arm for the heart rate collection.

Why?

Easy — **Okay** — **Difficult**

[✓ 2] 1 0

2



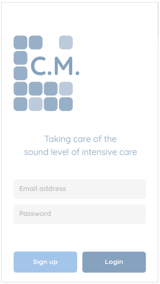
Please wear the clip the microphone device on your chest area.

Why?

Easy — **Okay** — **Difficult**

[✓ 2] 1 0

3



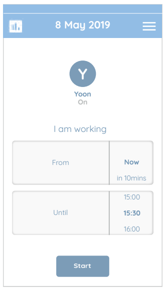
Please Sign in to Cacophony Mapper application, using Email and password written below:
Emai l: designer.yoon.lee@gmail.com
Password : dldbs

Why?

Easy — **Okay** — **Difficult**

[✓ 2] 1 0

4



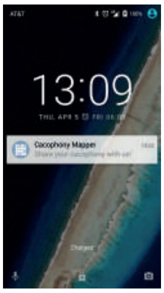
Please **start HB button** to connect the device to the Bluetooth. Assign the time slot **from now to 12 minutes** later and press **ttart rec button** and enjoy your work! :)

Why?

Easy — **Okay** — **Difficult**

[✓ 2] 1 0

5



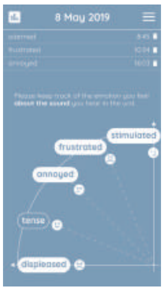
When there is a notification, please start the application by swiping the notification from the left to the rigt and report your emotion.

Why?

Easy — **Okay** — **Difficult**

[✓ 2] 1 0

6



Please press the emotion button which was relevant to your emotion related to your recent sound experience.

Why?

Easy — **Okay** — **Difficult**

[✓ 2] 1 0

23/25

The SUS(System Usability Scale) Survey for Cacophony Mapper application

1. Cacophony Mapper application was manageable to learn.

Strongly disagree ○ 1 ○ 2 ○ 3 ○ 4 ☑ 5 Strongly agree

Why do you think so?

2. Cacophony Mapper application was clearly structured.

Strongly disagree ○ 1 ○ 2 ○ 3 ○ 4 ☑ 5 Strongly agree

Why do you think so?

3. Cacophony Mapper application was complicated to use.

Strongly disagree ☑ 1 ○ 2 ○ 3 ○ 4 ○ 5 Strongly agree

Why do you think so?

4. Cacophony Mapper interface was straghtforward to use.

Strongly disagree ○ 1 ○ 2 ○ 3 ☑ 4 ○ 5 Strongly agree

Why do you think so?

The way emotion are presented is a bit fuzzy because you don't know
what are on each axis.

5. I think it will be practical to used Cacophony Mapper during work in a medical environment.

Strongly disagree ○ 1 ○ 2 ○ 3 ☑ 4 ○ 5 Strongly agree

Why do you think so?

I think it whould be easy because what you need to click is just emotion
and the process is quick and easy.

Comments?

Cacophony Mapper: Defining Noise Fatigue in the ICU

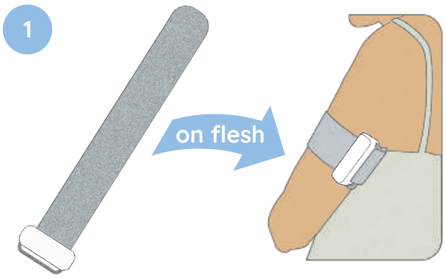
Name : Rob

Gender : Male **Age :** 23

Score : 9 **/ 12**

The purpose of this project is to find how sound stimuli can affect nurses' stress level and emotional responses in the ICU. Thus, you are going to be asked to wear a heart rate detection device and a sound collector during the test. Sound data will be distorted using a filter while recording, so no one can listen or restore the original sound that you don't need to worry about the privacy issue. The data will be only used for academic purpose. Please follow the instruction below and rate your usability of each process. The questionnaire on the next page should be filled in after the test. The duration of the recordning will be 12 minutes and you do not need to try to remember the sound from the recording. Thank you for your participation and enjoy!

1



Please fasten the armband on your outer side of the bare upper arm for the heart rate collection.

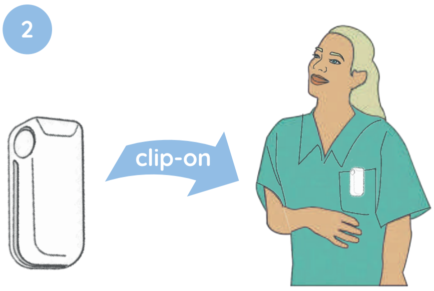
Easy ——— **Okay** ——— **Difficult**

2 [✓ 1] 0

Why?

One hand is difficult to coordinate.

2



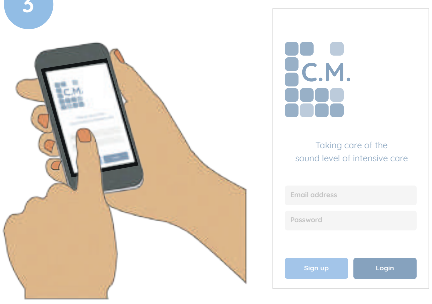
Please wear the clip the microphone device on your chest area.

Easy ——— **Okay** ——— **Difficult**

[✓ 2] 1 0

Why?

3



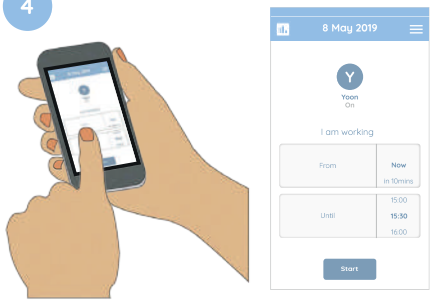
Please Sign in to Cacophony Mapper application, using Email and password written below:
Emai l: designer.yoon.lee@gmail.com
Password : dldbs

Easy ——— **Okay** ——— **Difficult**

[✓ 2] 1 0

Why?

4



Please **start HB button** to connect the device to the Bluetooth. Assign the time slot **from now to 12 minutes** later and press **ttart rec button** and enjoy your work! :)

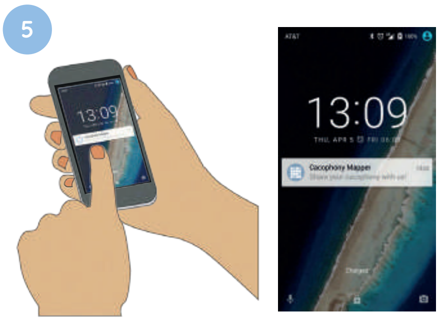
Easy ——— **Okay** ——— **Difficult**

2 [✓ 1] 0

Why?

The starting time was not set in now so it was not easy to understand a first.

5



When there is a notification, please start the application by swiping the notification from the left to the rigt and report your emotion.

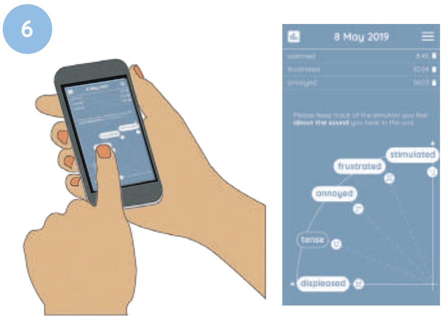
Easy ——— **Okay** ——— **Difficult**

2 1 [✓ 0]

Why?

I didn't notice them very well.

6



Please press the emotion button which was relevant to your emotion related to your recent sound experience.

Easy ——— **Okay** ——— **Difficult**

2 [✓ 1] 0

Why?

It seemed uneasy to understand at first.

17/25

The SUS(System Usability Scale) Survey for Cacophony Mapper application

1. Cacophony Mapper application was manageable to learn.

Strongly disagree ○ 1 ○ 2 ○ 3 [✓ 4] ○ 5 Strongly agree

Why do you think so?

It was a simple interface.

2. Cacophony Mapper application was clearly structured.

Strongly disagree ○ 1 ○ 2 [✓ 3] ○ 4 ○ 5 Strongly agree

Why do you think so?

No strong opinion on this since this is a part of the applicaiton.

3. Cacophony Mapper application was complicated to use.

Strongly disagree ○ 1 [✓ 2] ○ 3 ○ 4 ○ 5 Strongly agree

Why do you think so?

It was easy to use.

4. Cacophony Mapper interface was straghtforward to use.

Strongly disagree ○ 1 ○ 2 ○ 3 [✓ 4] ○ 5 Strongly agree

Why do you think so?

It was easy to use.

5. I think it will be practical to used Cacophony Mapper during work in a medical environment.

Strongly disagree ○ 1 [✓ 2] ○ 3 ○ 4 ○ 5 Strongly agree

Why do you think so?

I think notifications will distract me.

Comments?

Cacophony Mapper: Defining Noise Fatigue in the ICU

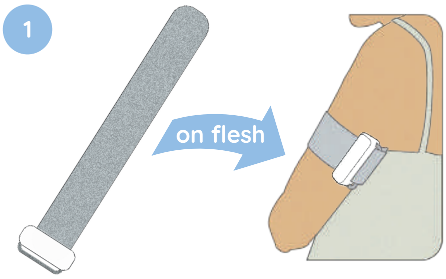
Name : Jooyoung

Gender : Female **Age :** 27

Score : 10 **/ 12**

The purpose of this project is to find how sound stimuli can affect nurses' stress level and emotional responses in the ICU. Thus, you are going to be asked to wear a heart rate detection device and a sound collector during the test. Sound data will be distorted using a filter while recording, so no one can listen or restore the original sound that you don't need to worry about the privacy issue. The data will be only used for academic purpose. Please follow the instruction below and rate your usability of each process. The questionnaire on the next page should be filled in after the test. The duration of the recordning will be 12 minutes and you do not need to try to remember the sound from the recording. Thank you for your participation and enjoy!

1



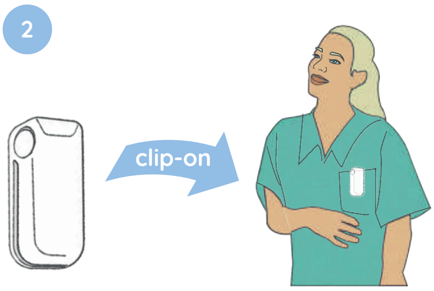
Please fasten the armband on your outer side of the bare upper arm for the heart rate collection.

Easy ———— **Okay** ———— **Difficult**

[✓ 2] 1 0

Why?

2



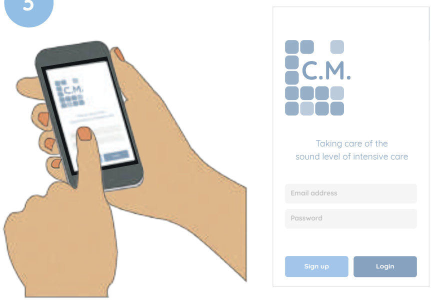
Please wear the clip the microphone device on your chest area.

Easy ———— **Okay** ———— **Difficult**

[✓ 2] 1 0

Why?

3



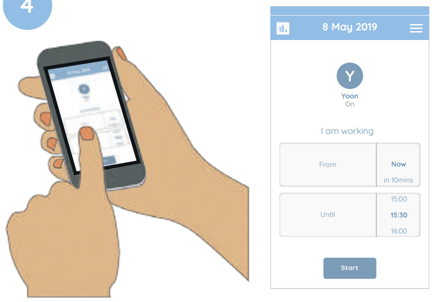
Please Sign in to Cacophony Mapper application, using Email and password written below:
Emai l: designer.yoon.lee@gmail.com
Password : dldbs

Easy ———— **Okay** ———— **Difficult**

[✓ 2] 1 0

Why?

4



Please **start HB button** to connect the device to the Bluetooth. Assign the time slot **from now to 12 minutes** later and press **ttart rec button** and enjoy your work! :)

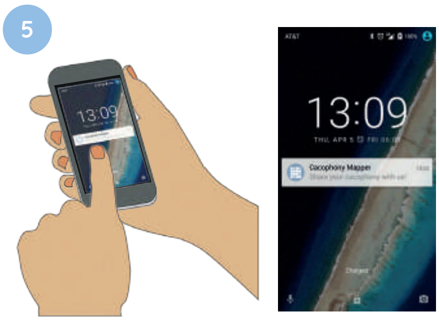
Easy ———— **Okay** ———— **Difficult**

2 [✓ 1] 0

Why?

I didn't know what HB meant.

5



When there is a notification, please start the application by swiping the notification from the left to the rigt and report your emotion.

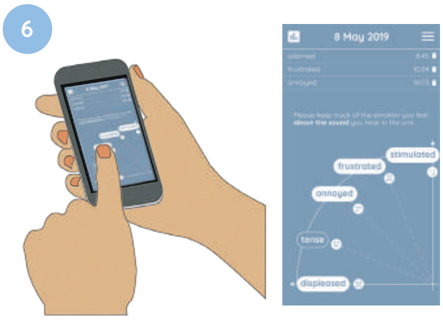
Easy ———— **Okay** ———— **Difficult**

2 [✓ 1] 0

Why?

The interaction was easy but there was no vibration, so I couldn't be noticed easily.

6



Please press the emotion button which was relevant to your emotion related to your recent sound experience.

Easy ———— **Okay** ———— **Difficult**

[✓ 2] 1 0

Why?

19 /25

The SUS(System Usability Scale) Survey for Cacophony Mapper application

1. Cacophony Mapper application was manageable to learn.

Strongly disagree ○ 1 ○ 2 ○ 3 [✓ 4] ○ 5 Strongly agree

Why do you think so?

2. Cacophony Mapper application was clearly structured.

Strongly disagree ○ 1 ○ 2 [✓ 3] ○ 4 ○ 5 Strongly agree

Why do you think so?

Buttons were really intuitive to click but wasn't sure why those buttons are positioned in that way.

3. Cacophony Mapper application was complicated to use.

Strongly disagree ○ 1 [✓ 2] ○ 3 ○ 4 ○ 5 Strongly agree

Why do you think so?

4. Cacophony Mapper interface was straghtforward to use.

Strongly disagree ○ 1 ○ 2 ○ 3 [✓ 4] ○ 5 Strongly agree

Why do you think so?

5. I think it will be practical to used Cacophony Mapper during work in a medical environment.

Strongly disagree ○ 1 ○ 2 ○ 3 [✓ 4] ○ 5 Strongly agree

Why do you think so?

I think it would be practical to use this applicaiton in the hospital environment, but notifications shouldn't bother clinicians too much.

Comments?

Cacophony Mapper: Defining Noise Fatigue in the ICU

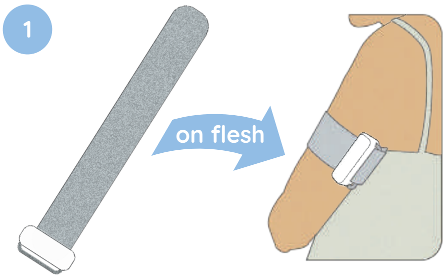
Name : Sebastiaan

Gender : Male **Age :** 26

Score : 10 **/ 12**

The purpose of this project is to find how sound stimuli can affect nurses' stress level and emotional responses in the ICU. Thus, you are going to be asked to wear a heart rate detection device and a sound collector during the test. Sound data will be distorted using a filter while recording, so no one can listen or restore the original sound that you don't need to worry about the privacy issue. The data will be only used for academic purpose. Please follow the instruction below and rate your usability of each process. The questionnaire on the next page should be filled in after the test. The duration of the recordning will be 12 minutes and you do not need to try to remember the sound from the recording. Thank you for your participation and enjoy!

1



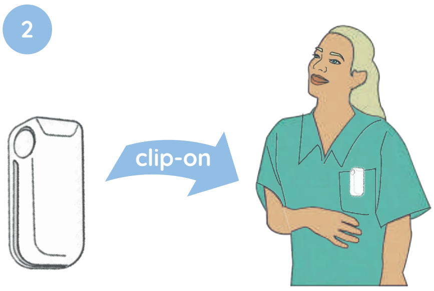
Please fasten the armband on your outer side of the bare upper arm for the heart rate collection.

Easy ——— **Okay** ——— **Difficult**

2 [✓ 1] 0

Why?

2



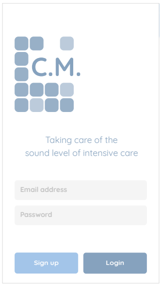
Please wear the clip the microphone device on your chest area.

Easy ——— **Okay** ——— **Difficult**

[✓ 2] 1 0

Why?

3



C.M.

Taking care of the sound level of intensive care

Email address

Password

Sign up Login

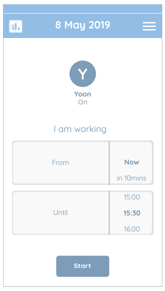
Please Sign in to Cacophony Mapper application, using Email and password written below:
Emai l: designer.yoon.lee@gmail.com
Password : dldbs

Easy ——— **Okay** ——— **Difficult**

[✓ 2] 1 0

Why?

4



8 May 2019

Y

Yoon
On

I am working

From	Now in 10mins
Until	15:00 15:30 16:00

Start

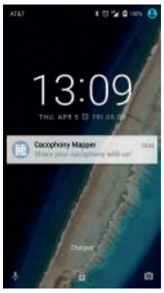
Please **start HB button** to connect the device to the Bluetooth. Assign the time slot **from now to 12 minutes** later and press **ttart rec button** and enjoy your work! :)

Easy ——— **Okay** ——— **Difficult**

2 [✓ 1] 0

Why?

5



13:09

THU, APR 5

Cacophony Mapper

Share your cacophony with us!

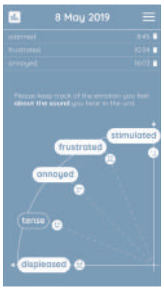
When there is a notification, please start the application by swiping the notification from the left to the rigt and report your emotion.

Easy ——— **Okay** ——— **Difficult**

[✓ 2] 1 0

Why?

6



8 May 2019

alarmed
frustrated
annoyed

Please keep track of the emotion you feel about the sound you hear in the unit

stimulated
frustrated
annoyed
tense
displeased

Please press the emotion button which was relevant to your emotion related to your recent sound experience.

Easy ——— **Okay** ——— **Difficult**

[✓ 2] 1 0

Why?

19 /25

The SUS(System Usability Scale) Survey for Cacophony Mapper application

1. Cacophony Mapper application was manageable to learn.

Strongly disagree ○ 1 ○ 2 ○ 3 [✓ 4] ○ 5 Strongly agree

Why do you think so?

2. Cacophony Mapper application was clearly structured.

Strongly disagree ○ 1 ○ 2 [✓ 3] ○ 4 ○ 5 Strongly agree

Why do you think so?

3. Cacophony Mapper application was complicated to use.

Strongly disagree ○ 1 [✓ 2] ○ 3 ○ 4 ○ 5 Strongly agree

Why do you think so?

4. Cacophony Mapper interface was straghtforward to use.

Strongly disagree ○ 1 ○ 2 ○ 3 [✓ 4] ○ 5 Strongly agree

Why do you think so?

I didn't see buttons at first, so I tried to press halfway the emotions. I think expressing a middle-ground options is missing (for example emotion between frustrated and neutral)

5. I think it will be practical to used Cacophony Mapper during work in a medical environment.

Strongly disagree ○ 1 [✓ 2] ○ 3 ○ 4 ○ 5 Strongly agree

Why do you think so?

I think it is disturbing to press buttons regularly.

Comments?

Cacophony Mapper: Defining Noise Fatigue in the ICU

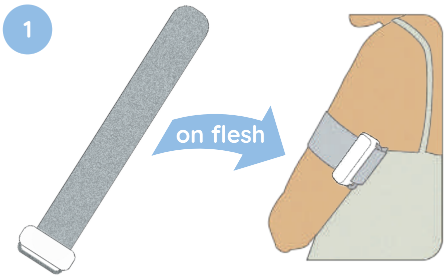
Name : Grace

Gender : Female **Age :** 32

Score : 11 **/ 12**

The purpose of this project is to find how sound stimuli can affect nurses' stress level and emotional responses in the ICU. Thus, you are going to be asked to wear a heart rate detection device and a sound collector during the test. Sound data will be distorted using a filter while recording, so no one can listen or restore the original sound that you don't need to worry about the privacy issue. The data will be only used for academic purpose. Please follow the instruction below and rate your usability of each process. The questionnaire on the next page should be filled in after the test. The duration of the recordning will be 12 minutes and you do not need to try to remember the sound from the recording. Thank you for your participation and enjoy!

1



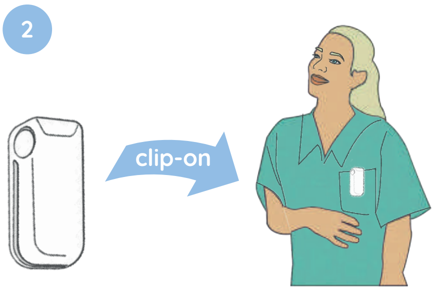
Please fasten the armband on your outer side of the bare upper arm for the heart rate collection.

Easy ——— **Okay** ——— **Difficult**

[✓ 2] 1 0

Why?

2



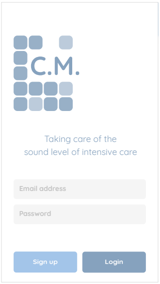
Please wear the clip the microphone device on your chest area.

Easy ——— **Okay** ——— **Difficult**

[✓ 2] 1 0

Why?

3



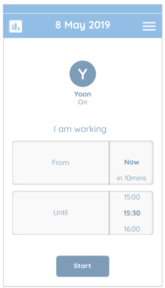
Please Sign in to Cacophony Mapper application, using Email and password written below:
Emai l: designer.yoon.lee@gmail.com
Password : dldbs

Easy ——— **Okay** ——— **Difficult**

[✓ 2] 1 0

Why?

4



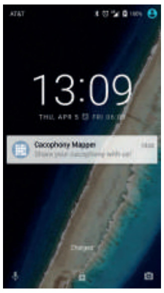
Please **start HB button** to connect the device to the Bluetooth. Assign the time slot **from now to 12 minutes** later and press **ttart rec button** and enjoy your work! :)

Easy ——— **Okay** ——— **Difficult**

2 [✓ 1] 0

Why?

5



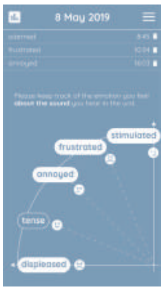
When there is a notification, please start the application by swiping the notification from the left to the rigt and report your emotion.

Easy ——— **Okay** ——— **Difficult**

[✓ 2] 1 0

Why?

6



Please press the emotion button which was relevant to your emotion related to your recent sound experience.

Easy ——— **Okay** ——— **Difficult**

[✓ 2] 1 0

Why?

21/25

The SUS(System Usability Scale) Survey for Cacophony Mapper application

1. Cacophony Mapper application was manageable to learn.

Strongly disagree ○ 1 ○ 2 ○ 3 [✓ 4] ○ 5 Strongly agree

Why do you think so?

It was not complicated in the beginning but time setting was a bit tricky. In general, it was easy to learn.

2. Cacophony Mapper application was clearly structured.

Strongly disagree ○ 1 ○ 2 ○ 3 [✓ 4] ○ 5 Strongly agree

Why do you think so?

3. Cacophony Mapper application was complicated to use.

Strongly disagree ○ 1 [✓ 2] ○ 3 ○ 4 ○ 5 Strongly agree

Why do you think so?

Only the first time was a bit complicated but was not too much to learn.

4. Cacophony Mapper interface was straghtforward to use.

Strongly disagree ○ 1 ○ 2 ○ 3 [✓ 4] ○ 5 Strongly agree

Why do you think so?

For me, emotion buttons are straigtforward to use.

5. I think it will be practical to used Cacophony Mapper during work in a medical environment.

Strongly disagree ○ 1 ○ 2 ○ 3 ○ 4 [✓ 5] Strongly agree

Why do you think so?

By using device, I can check the stress level and manage the stress in the end and it will improve the work environment in the end.

Comments?

Cacophony Mapper: Defining Noise Fatigue in the ICU

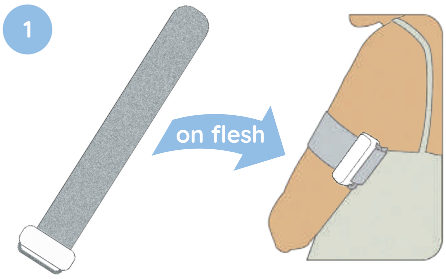
Name : Gyoung

Gender : Female **Age :** 29

Score : 11 / 12

The purpose of this project is to find how sound stimuli can affect nurses' stress level and emotional responses in the ICU. Thus, you are going to be asked to wear a heart rate detection device and a sound collector during the test. Sound data will be distorted using a filter while recording, so no one can listen or restore the original sound that you don't need to worry about the privacy issue. The data will be only used for academic purpose. Please follow the instruction below and rate your usability of each process. The questionnaire on the next page should be filled in after the test. The duration of the recordning will be 12 minutes and you do not need to try to remember the sound from the recording. Thank you for your participation and enjoy!

1



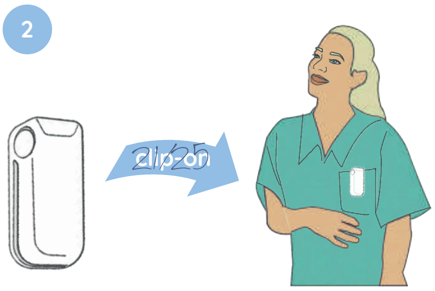
Please fasten the armband on your outer side of the bare upper arm for the heart rate collection.

Easy — **Okay** — **Difficult**

[✓ 2] 1 0

Why?

2



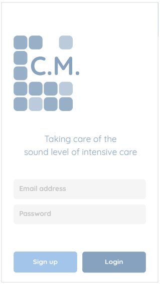
Please wear the clip the microphone device on your chest area.

Easy — **Okay** — **Difficult**

[✓ 2] 1 0

Why?

3



C.M.

Taking care of the sound level of intensive care

Email address

Password

Sign up Login

Please Sign in to Cacophony Mapper application, using Email and password written below:

Emai l: designer.yoon.lee@gmail.com

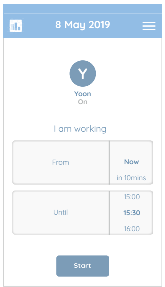
Password : dldbs

Easy — **Okay** — **Difficult**

[✓ 2] 1 0

Why?

4



8 May 2019

Y

Yoon

On

I am working

From | Now
in 10mins

Until | 15:00
15:30
16:00

Start

Please **start HB button** to connect the device to the Bluetooth. Assign the time slot **from now to 12 minutes** later and press **ttart rec button** and enjoy your work! :)

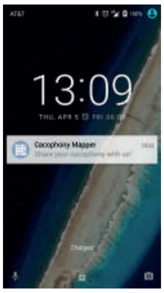
Easy — **Okay** — **Difficult**

2 [✓ 1] 0

Why?

I think the time setting interface should be more intuitive. (confusing to set up time for now and after.)

5



AT&T

13:09

THU, APR 5

Cacophony Mapper

Share your cacophony with us!

Charged

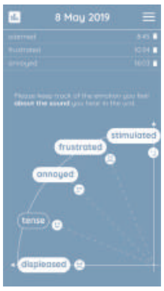
When there is a notification, please start the application by swiping the notification from the left to the rigt and report your emotion.

Easy — **Okay** — **Difficult**

[✓ 2] 1 0

Why?

6



8 May 2019

Please keep track of the emotion you feel about the sound you hear in the unit.

Please press the emotion button which was relevant to your emotion related to your recent sound experience.

Easy — **Okay** — **Difficult**

[✓ 2] 1 0

Why?

24/25

The SUS(System Usability Scale) Survey for Cacophony Mapper application

1. Cacophony Mapper application was manageable to learn.

Strongly disagree ○ 1 ○ 2 ○ 3 ○ 4 [✓ 5] Strongly agree

Why do you think so?

It was easy to follow because it was intuitive and self-explanatory interface.

2. Cacophony Mapper application was clearly structured.

Strongly disagree ○ 1 ○ 2 ○ 3 ○ 4 [✓ 5] Strongly agree

Why do you think so?

3. Cacophony Mapper application was complicated to use.

Strongly disagree [✓ 1] ○ 2 ○ 3 ○ 4 ○ 5 Strongly agree

Why do you think so?

4. Cacophony Mapper interface was straghtforward to use.

Strongly disagree ○ 1 ○ 2 ○ 3 ○ 4 [✓ 5] Strongly agree

Why do you think so?

5. I think it will be practical to used Cacophony Mapper during work in a medical environment.

Strongly disagree ○ 1 [✓ 2] ○ 3 ○ 4 ○ 5 Strongly agree

Why do you think so?

In case of real situation, it would be difficult to stop for a moment to record emotion.

Comments?

Cacophony Mapper: Defining Noise Fatigue in the ICU

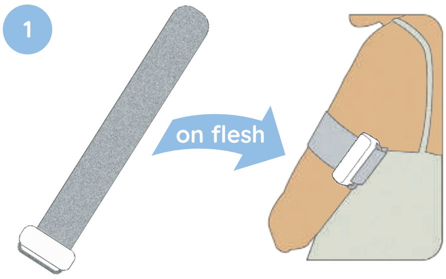
Name : Veronika

Gender : Female **Age :** 29

Score : 9 **/ 12**

The purpose of this project is to find how sound stimuli can affect nurses' stress level and emotional responses in the ICU. Thus, you are going to be asked to wear a heart rate detection device and a sound collector during the test. Sound data will be distorted using a filter while recording, so no one can listen or restore the original sound that you don't need to worry about the privacy issue. The data will be only used for academic purpose. Please follow the instruction below and rate your usability of each process. The questionnaire on the next page should be filled in after the test. The duration of the recordning will be 12 minutes and you do not need to try to remember the sound from the recording. Thank you for your participation and enjoy!

1



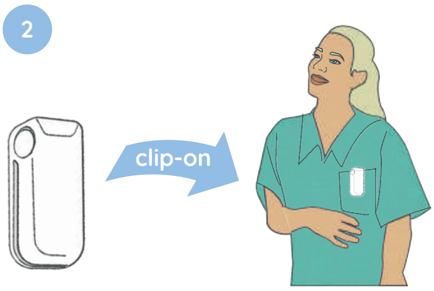
Please fasten the armband on your outer side of the bare upper arm for the heart rate collection.

Easy —— **Okay** —— **Difficult**

2 [✓ 1] 0

Why?

2



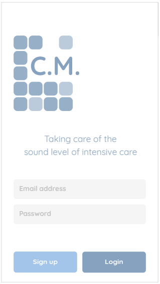
Please wear the clip the microphone device on your chest area.

Easy —— **Okay** —— **Difficult**

[✓ 2] 1 0

Why?

3


C.M.

Taking care of the sound level of intensive care

Email address

Password

Sign up | Login

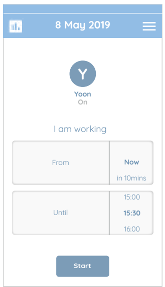
Please Sign in to Cacophony Mapper application, using Email and password written below:
Emai l: designer.yoon.lee@gmail.com
Password : dldbs

Easy —— **Okay** —— **Difficult**

[✓ 2] 1 0

Why?

4


8 May 2019

Y
Yoon
On

I am working

From: Now / in 10mins

Until: 15:00 / **15:30** / 16:00

Start

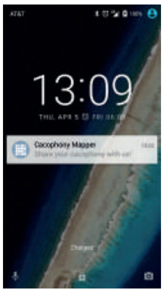
Please **start HB button** to connect the device to the Bluetooth. Assign the time slot **from now to 12 minutes** later and press **ttart rec button** and enjoy your work! :)

Easy —— **Okay** —— **Difficult**

2 [✓ 1] 0

Why?

5


AT&T

13:09

THU, APR 5

Cacophony Mapper
Share your cacophony with us!

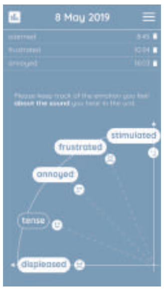
When there is a notification, please start the application by swiping the notification from the left to the rigt and report your emotion.

Easy —— **Okay** —— **Difficult**

2 [✓ 1] 0

Why?

6


8 May 2019

Please keep track of the emotion you feel **about the sound** you hear in the unit.

stimulated
frustrated
annoyed
tense
displeased

Please press the emotion button which was relevant to your emotion related to your recent sound experience.

Easy —— **Okay** —— **Difficult**

[✓ 2] 1 0

Why?

18/25

The SUS(System Usability Scale) Survey for Cacophony Mapper application

1. Cacophony Mapper application was manageable to learn.

Strongly disagree ○ 1 ○ 2 ○ 3 ☑ 4 ○ 5 Strongly agree

Why do you think so?

At first, I didn't understand the infreface but it was easy to learn.

2. Cacophony Mapper application was clearly structured.

Strongly disagree ○ 1 ○ 2 ○ 3 ☑ 4 ○ 5 Strongly agree

Why do you think so?

3. Cacophony Mapper application was complicated to use.

Strongly disagree ○ 1 ☑ 2 ○ 3 ○ 4 ○ 5 Strongly agree

Why do you think so?

4. Cacophony Mapper interface was straghtforward to use.

Strongly disagree ○ 1 ○ 2 ○ 3 ○ 4 ☑ 5 Strongly agree

Why do you think so?

5. I think it will be practical to used Cacophony Mapper during work in a medical environment.

Strongly disagree ☑ 1 ○ 2 ○ 3 ○ 4 ○ 5 Strongly agree

Why do you think so?

It was quite difficult to keep concentration while I tried to work on something.

Comments? The notification for emotion report should be more clear since the device didn't give a buzz in the testing

Cacophony Mapper: Defining Noise Fatigue in the ICU

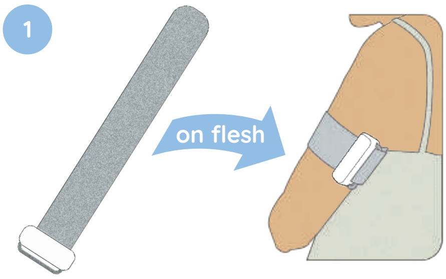
Name : Danielle

Gender : Female **Age :** 24

Score : / 12

The purpose of this project is to find how sound stimuli can affect nurses' stress level and emotional responses in the ICU. Thus, you are going to be asked to wear a heart rate detection device and a sound collector during the test. Sound data will be distorted using a filter while recording, so no one can listen or restore the original sound that you don't need to worry about the privacy issue. The data will be only used for academic purpose. Please follow the instruction below and rate your usability of each process. The questionnaire on the next page should be filled in after the test. The duration of the recordning will be 12 minutes and you do not need to try to remember the sound from the recording. Thank you for your participation and enjoy!

1



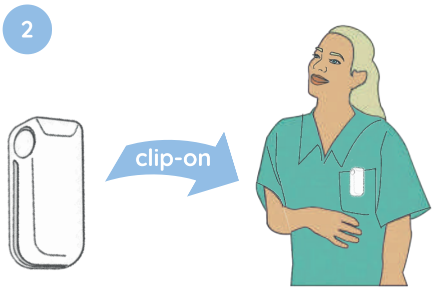
Please fasten the armband on your outer side of the bare upper arm for the heart rate collection.

Easy —— **Okay** —— **Difficult**

2 [✓ 1] 0

Why?

2



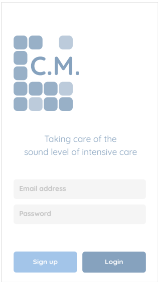
Please wear the clip the microphone device on your chest area.

Easy —— **Okay** —— **Difficult**

[✓ 2] 1 0

Why?

3



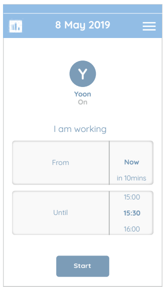
Please Sign in to Cacophony Mapper application, using Email and password written below:
Emai l: designer.yoon.lee@gmail.com
Password : dldbs

Easy —— **Okay** —— **Difficult**

[✓ 2] 1 0

Why?

4



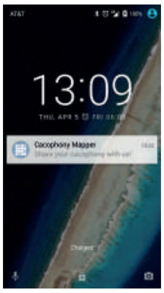
Please **start HB button** to connect the device to the Bluetooth. Assign the time slot **from now to 12 minutes** later and press **ttart rec button** and enjoy your work! :)

Easy —— **Okay** —— **Difficult**

2 [✓ 1] 0

Why?

5



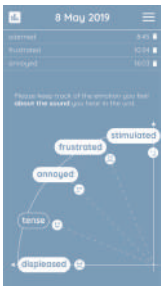
When there is a notification, please start the application by swiping the notification from the left to the rigt and report your emotion.

Easy —— **Okay** —— **Difficult**

2 [✓ 1] 0

Why?

6



Please press the emotion button which was relevant to your emotion related to your recent sound experience.

Easy —— **Okay** —— **Difficult**

[✓ 2] 1 0

Why?

21/25

The SUS(System Usability Scale) Survey for Cacophony Mapper application

1. Cacophony Mapper application was manageable to learn.

Strongly disagree ○ 1 ○ 2 ○ 3 [✓ 4] ○ 5 Strongly agree

Why do you think so?

2. Cacophony Mapper application was clearly structured.

Strongly disagree ○ 1 ○ 2 ○ 3 [✓ 4] ○ 5 Strongly agree

Why do you think so?

3. Cacophony Mapper application was complicated to use.

Strongly disagree [✓ 1] ○ 2 ○ 3 ○ 4 ○ 5 Strongly agree

Why do you think so?

4. Cacophony Mapper interface was straghtforward to use.

Strongly disagree ○ 1 ○ 2 ○ 3 [✓ 4] ○ 5 Strongly agree

Why do you think so?

5. I think it will be practical to used Cacophony Mapper during work in a medical environment.

Strongly disagree ○ 1 ○ 2 ○ 3 [✓ 4] ○ 5 Strongly agree

Why do you think so?

I think giving off vibration as a notification is improtant since the mobile will be in your pocket.

Comments?

Cacophony Mapper: Defining Noise Fatigue in the ICU

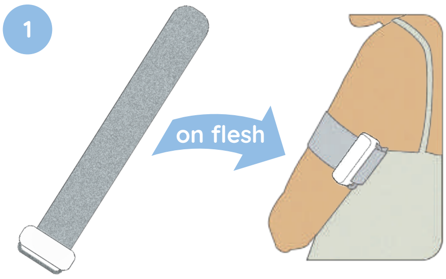
Name : Alex

Gender : Male **Age :** 25

Score : 10 **/ 12**

The purpose of this project is to find how sound stimuli can affect nurses' stress level and emotional responses in the ICU. Thus, you are going to be asked to wear a heart rate detection device and a sound collector during the test. Sound data will be distorted using a filter while recording, so no one can listen or restore the original sound that you don't need to worry about the privacy issue. The data will be only used for academic purpose. Please follow the instruction below and rate your usability of each process. The questionnaire on the next page should be filled in after the test. The duration of the recordning will be 12 minutes and you do not need to try to remember the sound from the recording. Thank you for your participation and enjoy!

1



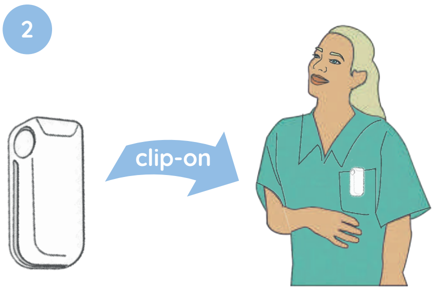
Why?

Please fasten the armband on your outer side of the bare upper arm for the heart rate collection.

Easy ——— **Okay** ——— **Difficult**

[✓ 2] 1 0

2



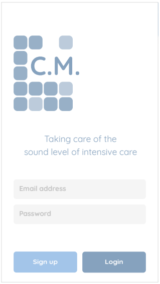
Why?

Please wear the clip the microphone device on your chest area.

Easy ——— **Okay** ——— **Difficult**

[✓ 2] 1 0

3


C.M.

Taking care of the sound level of intensive care

Email address

Password

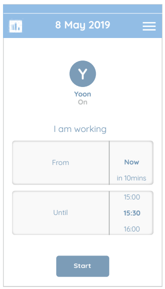
Why?

Please Sign in to Cacophony Mapper application, using Email and password written below:
Emai l: designer.yoon.lee@gmail.com
Password : dldbs

Easy ——— **Okay** ——— **Difficult**

[✓ 2] 1 0

4


8 May 2019

Y

Yoon
On

I am working

From: Now / in 10mins

Until: 15:00 / 15:30 / 16:00

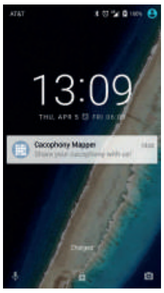
Why?

Please **start HB button** to connect the device to the Bluetooth. Assign the time slot **from now to 12 minutes** later and press **ttart rec button** and enjoy your work! :)

Easy ——— **Okay** ——— **Difficult**

2 [✓ 1] 0

5


AT&T

13:09

THU, APR 5

Cacophony Mapper

Share your cacophony with us!

Charged

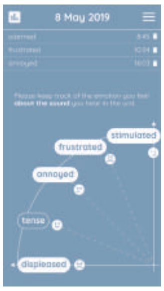
Why?

When there is a notification, please start the application by swiping the notification from the left to the rigt and report your emotion.

Easy ——— **Okay** ——— **Difficult**

2 [✓ 1] 0

6


8 May 2019

Why?

Please press the emotion button which was relevant to your emotion related to your recent sound experience.

Easy ——— **Okay** ——— **Difficult**

[✓ 2] 1 0

22/25

The SUS(System Usability Scale) Survey for Cacophony Mapper application

1. Cacophony Mapper application was manageable to learn.

Strongly disagree ○ 1 ○ 2 ○ 3 ○ 4 [✓ 5] Strongly agree

Why do you think so?

It is simple interface with a little layers.

2. Cacophony Mapper application was clearly structured.

Strongly disagree ○ 1 ○ 2 ○ 3 [✓ 4] ○ 5 Strongly agree

Why do you think so?

I think "Start HB" button and "Start Rec" button should be combined. It is weird to have those buttons in parallel.

3. Cacophony Mapper application was complicated to use.

Strongly disagree ○ 1 [✓ 2] ○ 3 ○ 4 ○ 5 Strongly agree

Why do you think so?

4. Cacophony Mapper interface was straghtforward to use.

Strongly disagree ○ 1 ○ 2 ○ 3 [✓ 4] ○ 5 Strongly agree

Why do you think so?

The "neutral" button was positioned a bit differently than others, so it took some time for me to recognize there was one.

5. I think it will be practical to used Cacophony Mapper during work in a medical environment.

Strongly disagree ○ 1 ○ 2 ○ 3 ○ 4 [✓ 5] Strongly agree

Why do you think so?

Since the only task is report their emotion, I think it can be done in the real environment too.

Comments?

Cacophony Mapper: Defining Noise Fatigue in the ICU

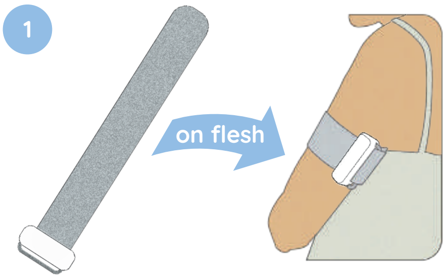
Name : Yiling

Gender : Female **Age :** 22

Score : 11 **/ 12**

The purpose of this project is to find how sound stimuli can affect nurses' stress level and emotional responses in the ICU. Thus, you are going to be asked to wear a heart rate detection device and a sound collector during the test. Sound data will be distorted using a filter while recording, so no one can listen or restore the original sound that you don't need to worry about the privacy issue. The data will be only used for academic purpose. Please follow the instruction below and rate your usability of each process. The questionnaire on the next page should be filled in after the test. The duration of the recordning will be 12 minutes and you do not need to try to remember the sound from the recording. Thank you for your participation and enjoy!

1



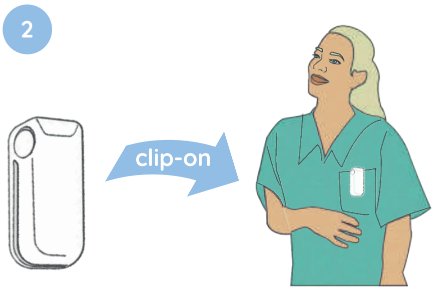
Please fasten the armband on your outer side of the bare upper arm for the heart rate collection.

Easy ——— **Okay** ——— **Difficult**

[✓ 2] 1 0

Why?

2



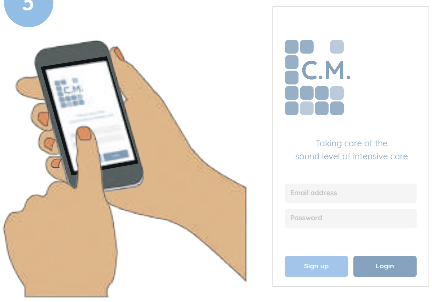
Please wear the clip the microphone device on your chest area.

Easy ——— **Okay** ——— **Difficult**

[✓ 2] 1 0

Why?

3



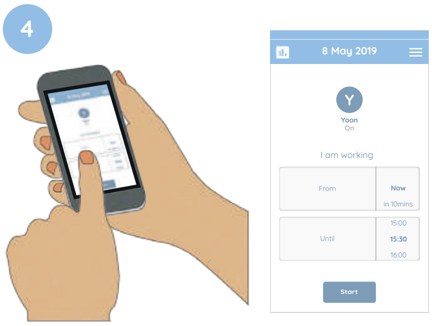
Please Sign in to Cacophony Mapper application, using Email and password written below:
Emai l: designer.yoon.lee@gmail.com
Password : dldbs

Easy ——— **Okay** ——— **Difficult**

[✓ 2] 1 0

Why?

4



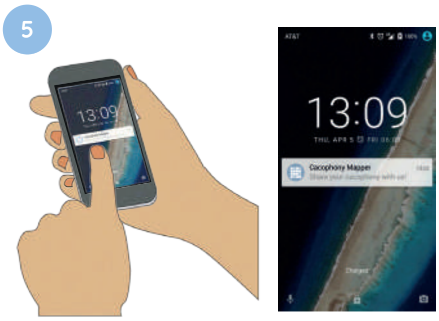
Please **start HB button** to connect the device to the Bluetooth. Assign the time slot **from now to 12 minutes** later and press **ttart rec button** and enjoy your work! :)

Easy ——— **Okay** ——— **Difficult**

2 [✓ 1] 0

Why?

5



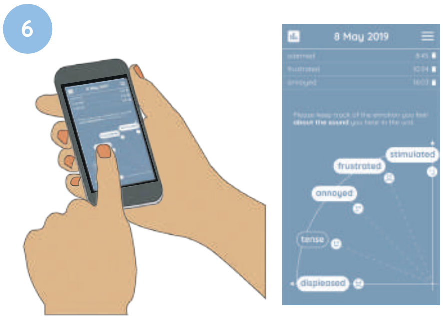
When there is a notification, please start the application by swiping the notification from the left to the rigt and report your emotion.

Easy ——— **Okay** ——— **Difficult**

[✓ 2] 1 0

Why?

6



Please press the emotion button which was relevant to your emotion related to your recent sound experience.

Easy ——— **Okay** ——— **Difficult**

[✓ 2] 1 0

Why?

The SUS(System Usability Scale) Survey for Cacophony Mapper application

1. Cacophony Mapper application was manageable to learn.

Strongly disagree ○ 1 ○ 2 ○ 3 ○ 4 [✓ 5] Strongly agree

Why do you think so?

2. Cacophony Mapper application was clearly structured.

Strongly disagree ○ 1 ○ 2 ○ 3 [✓ 4] ○ 5 Strongly agree

Why do you think so?

3. Cacophony Mapper application was complicated to use.

Strongly disagree ○ 1 ○ 2 ○ 3 [✓ 4] ○ 5 Strongly agree

Why do you think so?

It took me some time to get a Bluetooth connection for heart rate tracker.

4. Cacophony Mapper interface was straghtforward to use.

Strongly disagree ○ 1 ○ 2 ○ 3 ○ 4 [✓ 5] Strongly agree

Why do you think so?

5. I think it will be practical to used Cacophony Mapper during work in a medical environment.

Strongly disagree ○ 1 ○ 2 ○ 3 [✓ 4] ○ 5 Strongly agree

Why do you think so?

Notifications without vibration was not easy to catch.

Comments? I think respondent's personal situation can affect to general emotion report even though that is not related to sound experience.

User test data: Alex

15:39:59	8	39	20	0.644106388	0.529583812	0.486123085	incidental	incidental	
15:40:09	147	21	135	0.850320756	0.74557966	0.652014613	incidental	alarms	
15:40:19	117	135	122	6.582211971	5.469937325	4.860274792	incidental	alarms	
15:40:29	499	493	505	1.458024025	1.273670793	1.180526137	incidental	alarms	
15:40:39	554	533	520	1.176363468	1.047368526	0.942393959	incidental	alarms	
15:40:49	133	128	137	4.106258869	2.228001118	2.034805298	incidental	alarms	
15:40:59	129	128	535	4.591194153	2.443080664	2.379304409	incidental	alarms	
15:41:09	150	25	17	1.705943704	1.50355494	1.231989026	incidental	alarms	
15:41:19	18	21	19	3.56516242	1.723113775	1.622363448	incidental	alarms	
15:41:29	17	554	18	1.429094315	0.83224678	0.754692078	incidental	alarms	
15:41:39	12	52	22	0.785875738	0.635855138	0.611743569	incidental	alarms	
15:41:49	127	159	150	3.22524786	2.465080261	2.105055332	incidental	alarms	
15:41:59	159	147	158	1.371004581	1.291425467	0.976236224	incidental	alarms	
15:42:09	22	25	133	7.856760502	6.864361286	6.343356609	incidental	alarms	
15:42:19	156	159	18	2.803440094	1.492396116	1.418008924	incidental	alarms	
15:42:29	21	20	48	3.771864176	2.307833195	1.827970743	incidental	alarms	
15:42:39	580	524	125	3.482909679	1.499684334	1.484639406	incidental	alarms	tense
15:42:49	135	149	148	1.19209671	1.131442308	1.110735178	incidental	alarms	
15:42:59	33	152	228	1.62851119	0.860989034	0.694765806	conversation	alarms	
15:43:09	130	129	195	8.403878212	7.079025269	2.119298697	conversation	alarms	
15:43:19	19	21	38	1.467500687	0.70647794	0.523199558	conversation	alarms	
15:43:29	555	18	5	0.64147234	0.544946015	0.460469186	conversation	alarms	
15:43:39	62	233	234	5.182752132	5.126153469	4.658127308	conversation	alarms	
15:43:49	12	19	160	0.927852988	0.765834391	0.679667771	conversation	alarms	
15:43:59	21	25	382	4.410992622	4.356499672	1.724694848	conversation	alarms	
15:44:09	498	501	130	5.031188011	3.394296408	3.284371376	conversation	alarms	
15:44:19	43	50	44	1.183899522	1.072505713	0.983790219	conversation	alarms	
15:44:29	41	39	133	4.704407215	3.654771328	3.458458662	conversation	alarms	
15:44:39	44	48	47	18.07993317	17.20371056	14.6455555	conversation	alarms	
15:44:49	31	32	486	4.456507206	4.183105946	3.207586288	conversation	alarms	
15:44:59	64	162	61	1.789117455	1.464941144	1.153492808	conversation	alarms	
15:45:10	17	23	25	1.650973797	1.355594158	0.870652497	conversation	alarms	
15:45:20	6	28	19	0.616817892	0.56064719	0.461586595	conversation	incidental	
15:45:30	137	136	152	8.398532867	4.987271786	4.382613659	conversation	alarms	
15:45:40	39	20	13	2.887423515	2.511431932	2.166568995	conversation	alarms	
15:45:50	49	46	54	6.325631142	5.379469395	4.926165581	conversation	alarms	
15:46:00	19	140	15	0.624008477	0.555241644	0.545738697	machinery	alarms	neutral
15:46:10	42	43	15	1.675107121	1.314034462	1.162965298	machinery	alarms	
15:46:20	18	130	17	1.775048733	1.482331872	1.049422622	machinery	alarms	
15:46:30	18	25	476	0.604535758	0.580966711	0.508769512	machinery	alarms	
15:46:40	17	25	501	0.953009903	0.852466166	0.819085479	machinery	alarms	
15:46:50	18	131	125	2.445971966	1.784727454	1.45101738	machinery	alarms	
15:47:00	132	158	125	1.264959574	1.236393571	1.102118254	machinery	alarms	
15:47:10	9	35	11	0.528651774	0.488947213	0.399754435	machinery	incidental	
15:47:20	122	16	23	0.917823017	0.729592025	0.648360014	machinery	alarms	
15:47:30	533	514	19	1.798423052	1.781312227	1.694318771	machinery	alarms	
15:47:40	46	129	132	4.429954529	3.993202686	3.572512865	machinery	alarms	
15:47:50	21	127	133	0.952254772	0.887187839	0.784747362	machinery	alarms	neutral
15:48:00	20	21	27	1.057020307	0.658643484	0.533652604	machinery	alarms	
15:48:10	22	39	47	3.88098526	3.417440891	3.361878872	machinery	alarms	
15:48:20	17	16	20	0.97189647	0.746565938	0.50773561	machinery	alarms	
15:48:30	33	31	36	8.989224434	7.620128632	5.083240032	machinery	alarms	
15:48:40	152	126	17	3.234216452	1.745088577	1.532834291	machinery	alarms	neutral
15:48:50	14	27	23	1.509068847	0.955012321	0.881629407	machinery	alarms	
15:49:00	26	20	27	0.721761763	0.678753912	0.628189802	alarms	alarms	
15:49:10	46	43	41	5.472846031	2.885528326	1.660222054	alarms	alarms	
15:49:20	503	541	187	3.318225622	2.716632366	2.330497026	alarms	alarms	displeased
15:49:30	48	140	418	36.3119812	35.67761993	32.44536209	alarms	alarms	
15:49:40	324	325	232	9.851565361	5.417140961	4.082657337	alarms	alarms	tense

User test data: Seanne

16:07:57	5	6	2	0.710522115	0.38446638	0.21181196	incidental	incidental	
16:08:07	20	7	8	0.614666522	0.446586609	0.230622366	incidental	incidental	
16:08:17	294	279	48	2.520742178	1.796815515	1.766399622	incidental	alarms	
16:08:27	28	20	3	0.502557516	0.399103969	0.200112805	incidental	alarms	
16:08:37	483	496	479	2.770599127	1.472480297	1.317313075	incidental	alarms	
16:08:47	17	156	538	1.037716389	0.900786817	0.862035334	incidental	alarms	
16:08:57	426	506	503	12.87022114	9.396623611	8.88446331	incidental	alarms	
16:09:07	27	48	72	0.971475065	0.959652245	0.83101064	incidental	alarms	
16:09:17	120	20	22	0.872095048	0.683949709	0.682326436	incidental	alarms	
16:09:27	544	9	481	0.772643209	0.589879036	0.555261612	incidental	conversation	
16:09:38	17	481	483	1.896683812	1.046649814	0.902090669	incidental	alarms	
16:09:48	8	17	19	5.272541523	4.872628212	2.357375145	incidental	alarms	
16:09:58	132	130	133	1.713200688	1.30224371	1.295361042	incidental	alarms	
16:10:08	127	126	122	1.185773849	1.14768517	0.844431102	incidental	alarms	
16:10:18	13	11	119	0.404566944	0.383023739	0.359775096	incidental	incidental	
16:10:28	20	106	524	1.354585528	1.105025768	0.968459606	incidental	alarms	
16:10:38	8	12	19	0.509446263	0.425008386	0.318956614	incidental	incidental	
16:10:48	542	511	518	7.420064926	6.505860329	6.165812969	incidental	alarms	
16:10:58	570	569	540	10.40882301	9.861982346	9.4626894	conversation	alarms	
16:11:08	32	28	18	0.681998551	0.580636263	0.541672051	conversation	alarms	neutral
16:11:18	15	24	21	0.289766639	0.283656776	0.279452324	conversation	incidental	
16:11:28	23	21	487	1.564345717	1.18693006	1.133638263	conversation	alarms	
16:11:38	40	52	105	4.217610359	3.89155364	2.871868372	conversation	alarms	
16:11:48	489	487	40	2.055076361	1.845409274	1.835873842	conversation	alarms	
16:11:58	77	64	137	1.019643545	0.842453599	0.776765645	conversation	alarms	
16:12:08	61	38	277	4.702656746	3.706007004	3.543186665	conversation	alarms	
16:12:18	17	136	131	0.663666546	0.602090955	0.545530736	conversation	alarms	
16:12:28	541	556	549	2.203088999	2.124807835	2.085826635	conversation	alarms	
16:12:38	493	495	42	1.822015047	1.413558722	1.295358062	conversation	alarms	
16:12:48	497	48	494	2.958487034	2.785494328	2.444543123	conversation	alarms	
16:12:58	545	548	46	2.178141356	1.705423117	1.656334281	conversation	alarms	
16:13:08	159	20	140	6.431181431	4.953906059	2.434844255	conversation	alarms	
16:13:18	17	22	18	1.072820306	0.443038553	0.407504082	conversation	incidental	
16:13:28	39	36	40	1.810913801	1.525228739	1.178532958	conversation	alarms	
16:13:38	19	48	20	3.536713123	2.250235081	1.585165262	conversation	alarms	
16:13:48	17	5	8	1.168688059	1.112052441	1.072661877	conversation	incidental	
16:13:58	488	485	25	0.617590249	0.602681339	0.585592806	machinery	alarms	
16:14:08	22	23	24	0.836657047	0.734859228	0.478975773	machinery	alarms	
16:14:18	18	14	15	0.506452143	0.410884231	0.337433875	machinery	incidental	
16:14:28	20	22	9	0.840863526	0.383311301	0.283714145	machinery	incidental	
16:14:38	158	117	128	1.092973113	0.985507011	0.765605927	machinery	alarms	
16:14:48	151	15	12	0.731207788	0.27691409	0.26621449	machinery	alarms	
16:14:58	548	346	130	0.304116726	0.298543245	0.275202662	machinery	alarms	
16:15:08	120	117	1761	0.324523389	0.31837526	0.289710939	machinery	alarms	
16:15:18	18	503	512	1.063646674	0.740127385	0.722709715	machinery	alarms	
16:15:28	46	49	44	2.448371887	2.439112425	2.314001322	machinery	alarms	
16:15:38	544	542	558	13.55915833	10.4619627	9.281758308	machinery	alarms	
16:15:49	41	20	6	1.022966504	0.541732252	0.432336271	machinery	alarms	
16:15:59	37	48	121	1.122999787	0.789976895	0.727570772	machinery	alarms	
16:16:09	12	120	18	0.380976975	0.195934013	0.172100037	machinery	incidental	
16:16:19	46	47	49	8.318790436	5.412504196	3.104804516	machinery	alarms	
16:16:29	119	120	12	1.583693027	0.931433201	0.802872837	machinery	alarms	
16:16:39	20	150	159	0.818787873	0.557632267	0.545074165	machinery	alarms	
16:16:49	22	33	41	0.556367457	0.3789379	0.374347687	machinery	alarms	
16:16:59	18	16	44	2.989885092	0.889253795	0.783872426	alarms	alarms	neutral
16:17:09	22	485	17	0.555345595	0.424865514	0.366340011	alarms	alarms	
16:17:19	74	71	154	2.860734701	2.535277843	2.400329113	alarms	alarms	
16:17:29	160	18	144	1.358982205	1.010075212	0.912889004	alarms	alarms	
16:17:39	122	119	123	10.73904991	10.46014214	5.199195385	alarms	alarms	
16:17:49	189	192	571	43.70874023	13.57853222	3.447214127	alarms	alarms	
16:17:59	45	46	138	33.79218674	13.26746082	10.60404491	alarms	alarms	
16:18:09	323	322	321	30.4916172	20.15363503	18.0724144	alarms	alarms	
16:18:19	327	328	323	52.7885704	30.98361778	19.71361542	alarms	alarms	
16:18:29	92	279	464	5.219455242	4.342640877	3.122792006	alarms	alarms	
16:18:39	137	296	493	2.697107077	2.487110853	2.120351076	alarms	alarms	
16:18:49	32	52	59	0.44848451	0.340228647	0.309576213	alarms	alarms	
16:18:59	18	24	479	0.86117059	0.338442057	0.271438807	alarms	alarms	
16:19:09	17	20	130	4.65100956	3.322273254	2.697648048	alarms	alarms	
16:19:19	20	19	261	1.290632486	0.807500601	0.645923078	alarms	alarms	
16:19:29	156	19	46	0.293792069	0.28677088	0.280494869	alarms	alarms	
16:19:39	135	131	140	1.888917923	1.728644967	0.893991351	alarms	alarms	
16:19:49	17	18	28	1.058460832	0.527031839	0.412888795	alarms	alarms	
16:19:59	484	481	498	1.551034093	1.548577309	1.44999826	alarms	alarms	
16:20:09	22	17	19	6.529929161	3.020483017	1.814458489	alarms	alarms	stimulated

User test data: Rob

16:40:47	10	8	17	1.039099932	0.982508481	0.742913127	incidental	incidental	
16:40:57	24	26	20	0.597828984	0.420042962	0.375925213	incidental	alarms	
16:41:07	154	169	155	2.09198451	1.621603608	1.618193984	incidental	alarms	
16:41:17	513	505	485	7.740833282	7.589360237	7.34264946	incidental	alarms	
16:41:27	544	540	542	13.37444115	13.34817982	12.28671932	incidental	alarms	
16:41:37	53	18	51	2.07562542	0.900625646	0.796648741	incidental	alarms	
16:41:47	506	481	490	5.118053913	4.638739586	3.971023798	incidental	alarms	
16:41:57	158	47	49	4.941292286	4.792693138	4.550886154	incidental	alarms	
16:42:07	213	522	91	1.44577086	1.25209403	1.231671929	incidental	alarms	
16:42:17	119	33	114	14.63299561	6.066511631	5.645432949	incidental	alarms	
16:42:27	552	577	512	1.18301487	1.161760092	1.09439671	incidental	alarms	
16:42:37	554	59	7	1.169650674	0.897233069	0.878208041	incidental	alarms	
16:42:47	33	34	520	5.164369106	4.326026917	3.890940666	incidental	alarms	
16:42:57	541	526	539	6.508298397	5.754269123	5.532942295	incidental	alarms	
16:43:07	148	147	60	1.438809514	1.057294369	0.677373827	incidental	alarms	
16:43:17	518	542	148	11.18504143	7.719101429	7.59879446	incidental	alarms	
16:43:27	505	513	515	4.314263344	3.869976282	3.861080647	incidental	alarms	
16:43:37	31	7	20	1.364399076	0.792173564	0.631704152	incidental	incidental	
16:43:47	524	127	143	1.579311728	1.573803306	1.379780293	conversation	alarms	
16:43:57	220	41	21	3.114866257	1.62343514	1.501739621	conversation	alarms	
16:44:07	138	68	115	16.65231514	11.49840641	10.73727512	conversation	alarms	
16:44:17	74	33	16	3.471644878	1.793573022	1.715986729	conversation	alarms	
16:44:27	45	132	133	2.168279886	1.703951359	1.491841912	conversation	alarms	
16:44:37	216	36	33	8.010576248	5.788788795	5.404756546	conversation	alarms	
16:44:47	38	17	33	2.10831666	1.460656762	1.07184279	conversation	alarms	
16:44:57	18	35	16	3.192421675	2.487936735	2.197903633	conversation	alarms	
16:45:07	132	234	182	3.579767227	3.376803398	3.171082973	conversation	alarms	
16:45:17	464	19	505	4.562903881	3.651084423	3.628459215	conversation	alarms	
16:45:28	157	159	153	8.243818283	4.493390083	3.034391165	conversation	alarms	
16:45:38	60	33	61	3.446494579	2.492540836	2.211471319	conversation	alarms	
16:45:48	457	492	455	5.494787216	5.396153927	4.695379257	conversation	alarms	
16:45:58	36	31	40	9.650292397	2.56801796	2.463037491	conversation	alarms	
16:46:08	119	132	123	1.530821085	1.503254771	1.032045245	conversation	alarms	
16:46:18	268	267	530	4.447056293	3.627594233	3.614217043	conversation	alarms	
16:46:28	481	82	80	1.45530355	1.281589746	1.280394554	conversation	alarms	
16:46:38	10	19	45	1.384601593	1.085710883	0.931094766	conversation	incidental	
16:46:48	495	506	494	4.967416763	3.030070543	2.93023181	machinery	alarms	
16:46:58	145	89	125	2.030191183	1.818526506	1.682952166	machinery	alarms	
16:47:08	41	126	17	1.387679577	0.526874006	0.472448468	machinery	alarms	
16:47:18	13	12	27	0.847348154	0.721876264	0.61306107	machinery	incidental	
16:47:28	482	152	479	0.974067211	0.649145424	0.641013205	machinery	alarms	
16:47:38	13	505	518	4.059233665	3.272021532	3.204139948	machinery	alarms	
16:47:48	62	493	509	7.315855503	5.156857491	5.14608717	machinery	alarms	
16:47:58	145	148	13	2.073961496	1.219843149	0.720811009	machinery	alarms	
16:48:08	29	34	11	1.37118113	1.175948381	0.76844883	machinery	alarms	
16:48:18	101	16	117	1.229141712	0.839885294	0.834397018	machinery	alarms	
16:48:28	17	13	0	0.52560699	0.470985949	0.372783601	machinery	incidental	
16:48:38	26	5	514	1.036781311	0.902969897	0.850231826	machinery	conversation	
16:48:48	149	52	146	3.579552889	1.519678354	1.462223172	machinery	alarms	
16:48:58	179	145	111	1.20058763	1.138506413	1.100787878	machinery	alarms	
16:49:08	60	5	26	0.987124622	0.625522673	0.491485	machinery	conversation	
16:49:18	19	59	55	1.309285045	0.81005758	0.677002668	machinery	alarms	
16:49:28	18	64	236	1.117798209	0.796342432	0.780175149	machinery	alarms	
16:49:38	46	51	36	2.256728888	1.87263298	1.871433496	machinery	alarms	
16:49:48	63	27	144	1.074630618	0.735072374	0.709842741	alarms	alarms	
16:49:58	190	187	192	14.01190758	9.375844955	6.73725462	alarms	alarms	
16:50:08	28	39	49	2.557298899	2.413362741	1.667062759	alarms	alarms	
16:50:18	323	322	324	51.12166977	17.29683495	16.79649162	alarms	alarms	
16:50:28	323	322	321	42.66131973	12.62232685	9.018177032	alarms	alarms	
16:50:38	324	232	325	26.75146294	22.65579796	17.38445854	alarms	alarms	
16:50:48	94	93	96	11.16823769	4.36613512	2.763538122	alarms	alarms	
16:50:58	10	14	16	0.771783412	0.471904904	0.442848235	alarms	incidental	
16:51:08	45	44	47	3.361966848	2.544906378	2.436533213	alarms	alarms	
16:51:18	483	59	480	0.600999832	0.565864682	0.427426606	alarms	alarms	
16:51:29	12	10	19	0.858640671	0.790932059	0.752977967	alarms	incidental	
16:51:39	42	43	47	3.044989109	2.926728964	2.712777376	alarms	alarms	
16:51:49	47	50	36	14.47799301	10.80557251	2.591726542	alarms	alarms	
16:51:59	45	48	47	17.35769272	13.05684566	10.20236397	alarms	alarms	
16:52:09	29	107	34	1.334489226	0.613675833	0.543630958	alarms	alarms	
16:52:19	32	477	516	7.847962379	6.610364914	6.157445431	alarms	alarms	
16:52:29	16	2	3	0.541903794	0.391672611	0.384153426	alarms	incidental	annoyed

User test data: Yiling

17:07:59	130	128	115	1.892530441	1.887761712	1.160543799	incidental	alarms	
17:08:09	8	10	1	14.47416401	13.03718376	11.41836548	incidental	incidental	
17:08:19	45	60	30	1.843403578	0.892732263	0.656602204	incidental	alarms	
17:08:29	564	537	512	4.499670029	4.161001682	4.047255516	incidental	alarms	
17:08:39	32	481	537	4.941087723	4.837652683	4.40790844	incidental	alarms	
17:08:49	21	44	51	2.524539471	2.470428705	2.144984484	incidental	alarms	
17:08:59	32	30	33	0.659386456	0.633183956	0.427133858	incidental	alarms	
17:09:09	152	153	210	1.580917358	0.828518093	0.613653362	incidental	alarms	
17:09:19	19	13	15	1.102884531	0.792247772	0.761434138	incidental	incidental	
17:09:29	531	526	540	1.315104723	1.281718254	1.059751749	incidental	alarms	
17:09:39	535	503	6	0.88737905	0.85753274	0.837998211	incidental	alarms	
17:09:49	14	39	20	0.576659441	0.489192367	0.423495114	incidental	incidental	
17:09:59	34	149	152	2.739526749	2.692025661	2.542077541	incidental	alarms	
17:10:09	33	20	60	0.546090305	0.44175306	0.42544055	incidental	alarms	
17:10:19	581	60	63	1.531219125	1.257000566	0.950802982	incidental	alarms	
17:10:29	533	532	463	3.638239145	2.315394163	2.274621964	incidental	alarms	
17:10:39	57	34	59	1.538600326	1.469774723	1.275994897	incidental	alarms	
17:10:49	58	56	535	2.168540716	1.630045056	1.187678337	incidental	alarms	
17:10:59	46	18	48	3.755710125	0.843919337	0.742275596	conversation	alarms	
17:11:09	21	18	17	1.937965035	1.390839219	0.99963057	conversation	incidental	neutral
17:11:19	25	27	5	1.321648359	0.743083715	0.722719371	conversation	alarms	
17:11:29	56	28	206	4.934378147	4.771194458	4.543140888	conversation	alarms	
17:11:39	492	487	498	14.86143494	14.0588398	13.76727104	conversation	alarms	
17:11:50	59	61	62	3.773186445	2.108029604	2.048760414	conversation	alarms	
17:12:00	48	34	29	1.63340795	0.96076107	0.948511243	conversation	alarms	
17:12:10	21	18	19	6.610150337	3.775693417	2.861665249	conversation	alarms	
17:12:20	55	37	56	10.30368233	6.7374053	6.126133442	conversation	alarms	
17:12:30	284	282	155	34.15539932	26.80799294	13.12618446	conversation	alarms	
17:12:40	317	312	287	7.513030529	6.557031631	6.187628746	conversation	alarms	
17:12:50	484	285	560	1.381744742	1.182373881	1.167956591	conversation	alarms	
17:13:00	118	115	121	28.22658157	27.96336555	9.983579636	conversation	alarms	
17:13:10	66	25	46	1.006198764	1.002270341	0.949438274	conversation	alarms	
17:13:20	555	523	532	2.457521915	2.391184092	1.807694077	conversation	alarms	
17:13:30	58	111	57	11.69158363	8.796308517	8.334608078	conversation	alarms	
17:13:40	515	510	296	2.896110058	2.278457403	1.730339766	conversation	alarms	
17:13:50	19	152	18	0.8618294	0.645956814	0.538463235	conversation	alarms	
17:14:00	138	132	285	2.674177408	2.597175598	1.875570774	machinery	alarms	
17:14:10	59	54	31	1.449941754	0.6231336	0.599691093	machinery	alarms	
17:14:20	38	35	24	1.090775728	0.932018042	0.668730021	machinery	alarms	
17:14:30	33	109	518	1.52170372	1.477239966	1.34579587	machinery	alarms	
17:14:40	149	152	484	1.504201889	1.112612963	1.010590672	machinery	alarms	
17:14:50	131	133	135	5.324565411	3.477408648	2.141494989	machinery	alarms	
17:15:00	132	128	131	3.120678186	1.83014524	1.83001864	machinery	alarms	
17:15:10	18	34	55	0.924144506	0.696073115	0.654142678	machinery	alarms	
17:15:20	109	114	154	2.670768499	2.331732035	1.644517779	machinery	alarms	
17:15:30	112	114	33	0.86798197	0.70795536	0.566696942	machinery	alarms	
17:15:40	25	38	140	0.763505936	0.705826402	0.70350188	machinery	alarms	
17:15:50	560	116	33	0.786450148	0.753486097	0.639213204	machinery	alarms	
17:16:00	17	16	9	1.080584168	1.029021144	0.532033205	machinery	alarms	
17:16:10	60	24	21	0.819250882	0.500545084	0.498998195	machinery	alarms	
17:16:20	50	45	31	2.172754049	1.850085735	1.305486083	machinery	alarms	
17:16:30	214	267	496	3.492395878	3.327813625	3.184565306	machinery	alarms	
17:16:40	291	287	281	4.489722252	3.185323238	2.429413557	machinery	alarms	
17:16:50	19	17	15	6.279038906	2.531110048	1.82383728	machinery	alarms	
17:17:00	9	15	18	0.569962382	0.553806663	0.517280757	alarms	incidental	
17:17:10	187	185	186	14.16517353	8.190429688	4.94155407	alarms	alarms	
17:17:21	21	5	150	2.055325747	0.57019949	0.437132299	alarms	incidental	
17:17:31	45	93	326	38.4779892	29.77163315	16.26302338	alarms	alarms	annoyed
17:17:41	328	321	91	21.0336647	20.98144913	19.54422188	alarms	alarms	
17:17:51	326	323	93	108.581604	60.45914459	19.34014702	alarms	alarms	
17:18:01	93	89	88	13.85219574	8.014172554	4.928872108	alarms	alarms	
17:18:11	34	23	25	1.139199853	0.611036658	0.471668422	alarms	alarms	
17:18:21	39	36	41	4.296550751	2.712813854	1.552251101	alarms	alarms	
17:18:31	212	311	113	0.682120204	0.642637312	0.605268657	alarms	alarms	
17:18:41	151	154	160	1.98104322	0.992358565	0.742482007	alarms	alarms	
17:18:51	20	59	17	1.02164638	0.805145741	0.787052155	alarms	alarms	
17:19:01	16	33	14	1.119623542	0.395195872	0.348778397	alarms	incidental	
17:19:11	35	48	46	14.59531403	13.8795929	11.33552361	alarms	alarms	
17:19:21	141	130	102	2.547394037	2.275119543	1.877234936	alarms	alarms	
17:19:31	135	216	155	1.034382105	1.008874178	0.958984256	alarms	alarms	
17:19:41	34	29	27	0.930357873	0.841998875	0.547120035	alarms	alarms	
17:19:51	18	21	17	8.081004143	7.637934685	4.078391552	alarms	alarms	

User test data: Danielle

9:16:16	21	41	52	7.151490211	2.696174145	2.694789648	incidental	alarms	
9:16:26	482	479	528	11.69486809	8.466977119	8.18756485	incidental	alarms	
9:16:36	27	50	6	3.748390913	1.653293133	1.299674153	incidental	alarms	
9:16:46	27	21	9	3.690021992	1.711241484	1.43593514	incidental	alarms	
9:16:56	28	12	55	3.491794348	2.818813562	1.292420149	incidental	conversation	
9:17:06	12	9	28	4.835211754	4.16134882	3.147669315	incidental	conversation	
9:17:16	27	28	168	4.23562336	2.718127966	1.630605102	incidental	alarms	
9:17:26	9	8	5	1.879603267	1.383043528	1.272710681	incidental	incidental	
9:17:36	134	139	188	2.879995584	2.100272655	1.794801593	incidental	alarms	
9:17:46	12	13	9	2.807193279	2.272709131	2.003452778	incidental	incidental	
9:17:56	28	514	25	3.699324846	2.390648603	1.377858877	incidental	alarms	
9:18:06	27	12	39	1.851634502	1.356700778	0.839499593	incidental	incidental	
9:18:16	143	49	54	4.915223598	4.474447727	3.903252125	incidental	alarms	
9:18:26	530	536	204	14.83256435	13.02470112	8.226765633	incidental	alarms	
9:18:36	28	27	9	3.195938826	1.428242445	1.125320911	incidental	alarms	
9:18:46	519	537	512	11.30263805	9.977215767	8.189624786	incidental	alarms	annoyed
9:18:56	520	509	566	8.580681801	7.507294178	7.255396843	incidental	alarms	
9:19:06	27	9	8	1.6654284	0.870800734	0.795195639	incidental	incidental	
9:19:16	28	484	45	4.535767555	3.724573374	3.509334803	conversation	alarms	
9:19:26	239	220	238	5.089596272	2.54197526	2.098855257	conversation	alarms	
9:19:37	140	136	91	22.62044334	21.28461647	21.10818672	conversation	alarms	annoyed
9:19:47	72	27	75	3.109105349	2.962091923	2.503932476	conversation	alarms	
9:19:57	145	144	27	3.715475082	3.640892029	2.553212881	conversation	alarms	
9:20:07	251	233	245	12.419137	7.990366936	3.109497786	conversation	alarms	
9:20:17	27	44	152	4.778178215	1.546033859	1.335282922	conversation	alarms	annoyed
9:20:27	53	27	55	9.021807671	4.30816555	4.215033531	conversation	alarms	
9:20:37	143	184	161	6.345106125	5.323157787	4.861945629	conversation	alarms	
9:20:47	20	19	43	7.577735424	7.523971081	6.884532928	conversation	alarms	
9:20:57	154	151	23	12.93583393	10.03177929	10.01042271	conversation	alarms	frustrated
9:21:07	158	27	41	4.832979679	4.447251797	3.749598503	conversation	alarms	
9:21:17	28	156	154	3.806804419	3.52761054	3.487945795	conversation	alarms	
9:21:27	27	177	24	9.0704422	6.537706852	5.312616348	conversation	alarms	
9:21:37	12	131	113	2.733172178	2.338883638	2.053529263	conversation	alarms	
9:21:47	11	26	13	5.031493664	2.06901741	1.945884824	conversation	alarms	
9:21:57	28	133	162	4.315606594	3.694514036	2.994884014	conversation	alarms	
9:22:07	28	45	21	6.98544693	3.490190029	2.547591686	conversation	alarms	
9:22:17	485	495	502	8.426734924	7.754645348	7.670830727	machinery	alarms	
9:22:27	28	152	143	5.35162735	2.826239347	2.304220915	machinery	alarms	
9:22:37	9	12	1	1.730967641	1.150119901	1.021242619	machinery	incidental	
9:22:47	12	11	20	1.623417974	1.274803519	1.173481226	machinery	incidental	
9:22:57	25	23	136	2.642693996	2.514294624	2.048959017	machinery	alarms	
9:23:07	490	493	481	15.02823257	10.89540672	8.769422531	machinery	alarms	
9:23:17	54	50	45	17.39298058	16.67319298	13.28390694	machinery	alarms	
9:23:27	17	150	23	1.772077799	1.713576555	1.503858447	machinery	alarms	
9:23:37	28	27	17	4.319963455	2.006504536	1.951328754	machinery	alarms	
9:23:47	25	126	23	2.973407269	1.5728122	1.335409522	machinery	alarms	
9:23:57	27	10	14	4.628973484	2.337071657	1.705862999	machinery	conversation	
9:24:07	23	508	511	2.615906954	1.148960233	0.923203588	machinery	alarms	
9:24:17	28	130	188	4.083840847	3.069315672	2.831234932	machinery	alarms	
9:24:27	28	27	10	4.054014683	2.284908056	1.950602651	machinery	alarms	
9:24:37	12	25	15	1.059217334	1.020497918	0.910541177	machinery	incidental	displeased
9:24:47	503	24	484	1.29420197	1.234768391	1.026834369	machinery	alarms	
9:24:57	22	23	28	6.802853584	4.788938046	1.880402088	machinery	conversation	
9:25:07	52	28	39	6.390545845	5.622183323	5.287916183	machinery	alarms	
9:25:17	27	10	12	5.016571999	3.487641335	2.576686144	alarms	conversation	
9:25:27	189	25	21	2.548378706	1.74114275	1.524577379	alarms	alarms	
9:25:38	33	22	42	2.47625351	2.156744003	2.084729671	alarms	alarms	
9:25:48	93	326	232	12.6768856	11.38009548	10.77188492	alarms	alarms	
9:25:58	325	46	93	28.05281448	16.8348465	15.22848225	alarms	alarms	annoyed
9:26:08	93	92	232	15.26310158	9.712119102	6.359476089	alarms	alarms	displeased
9:26:18	94	91	96	23.6879921	14.61385059	9.964966774	alarms	alarms	
9:26:28	12	28	9	1.595017433	1.548710585	1.081302524	alarms	alarms	
9:26:38	27	46	31	5.54929781	3.00070262	2.209462166	alarms	alarms	
9:26:48	27	161	13	3.174592733	1.077640295	0.81621176	alarms	alarms	
9:26:58	28	25	18	2.535887241	1.154134035	0.87116617alarms	alarms		

User test data: Veronika

10:32:18	11	27	7	2.294693947	1.308060408	1.165325284	incidental	incidental	
10:32:28	8	44	13	1.148837805	1.056872964	0.764154613	incidental	alarms	
10:32:38	411	415	377	5.520960808	5.406888962	5.054074287	incidental	alarms	
10:32:48	10	503	20	1.620798707	1.217840672	0.918102205	incidental	alarms	
10:32:58	51	54	33	19.56132698	11.39912987	9.058878899	incidental	alarms	
10:33:08	56	128	148	1.977231503	1.533670425	1.476028681	incidental	alarms	
10:33:18	516	521	554	4.619310856	4.604483128	4.390069485	incidental	alarms	
10:33:28	28	50	145	1.054684162	0.597533882	0.451067865	incidental	alarms	
10:33:38	46	29	55	1.976608157	1.133196831	1.129597902	incidental	alarms	
10:33:48	144	155	510	3.733062029	2.26845932	2.052835226	incidental	alarms	
10:33:58	28	534	51	1.861937642	1.528154016	0.999179065	incidental	alarms	
10:34:08	20	7	10	3.508394003	3.008552074	2.930915356	incidental	conversation	
10:34:18	38	43	26	3.569845676	2.493527174	1.86967957	incidental	alarms	
10:34:28	19	13	17	1.237397671	1.055128813	0.899583817	incidental	incidental	
10:34:38	576	577	21	3.037503719	2.049071789	1.814663291	incidental	alarms	
10:34:48	577	576	579	10.59157276	7.799260616	5.518237114	incidental	alarms	
10:34:58	27	22	14	2.773247719	1.819146395	1.255747437	incidental	alarms	
10:35:08	12	7	9	1.890351057	1.817890882	1.201205969	incidental	incidental	
10:35:18	27	14	8	1.140398979	0.99508971	0.670914412	conversation	incidental	
10:35:28	150	77	143	14.56927299	8.228412628	7.403042793	conversation	alarms	
10:35:38	8	15	22	0.694181263	0.666387439	0.664834738	conversation	alarms	
10:35:48	125	132	126	3.705796719	3.58012557	2.983273745	conversation	alarms	
10:35:58	34	36	37	2.66730094	2.007698298	1.034283519	conversation	alarms	
10:36:08	52	9	28	2.844129562	1.158126712	1.025991917	conversation	conversation	
10:36:18	212	44	209	7.532952309	7.169885635	6.140591145	conversation	alarms	
10:36:28	19	56	27	4.454545498	2.939962626	2.378005981	conversation	alarms	
10:36:38	67	55	17	3.75669241	2.175770521	2.072430849	conversation	alarms	
10:36:48	305	114	149	2.345376968	2.209614038	2.151815176	conversation	alarms	
10:36:58	35	54	33	3.958182573	2.233396053	1.661250234	conversation	alarms	
10:37:08	112	123	69	4.761347771	4.521755219	2.933573246	conversation	alarms	
10:37:18	505	407	472	2.394695044	1.36916399	1.301686406	conversation	alarms	
10:37:29	28	27	12	1.191939712	0.671868503	0.506473601	conversation	incidental	
10:37:39	141	372	146	6.257899761	5.942346573	5.800124168	conversation	alarms	
10:37:49	54	27	12	1.435435295	1.17553103	1.16431427	conversation	alarms	
10:37:59	32	137	14	0.778958619	0.690298557	0.594677866	conversation	alarms	
10:38:09	129	134	135	4.224490643	3.754508257	3.384399891	conversation	alarms	
10:38:19	28	9	33	0.914806545	0.905131757	0.715242624	machinery	conversation	
10:38:29	484	27	28	2.883055687	2.434592247	2.254121304	machinery	alarms	tense
10:38:39	27	23	24	3.595589638	2.730292082	1.885015965	machinery	alarms	
10:38:49	21	26	12	2.410000563	1.412234783	1.136987209	machinery	alarms	
10:38:59	131	53	129	8.022500038	4.093475819	3.588572741	machinery	alarms	
10:39:09	128	56	21	2.772661924	2.552744389	1.967232585	machinery	alarms	
10:39:19	28	20	22	2.064037561	1.43258667	1.313506722	machinery	alarms	
10:39:29	28	10	54	1.462924361	1.452179909	1.3332026	machinery	conversation	
10:39:39	516	486	485	7.150791645	6.08809185	5.812129498	machinery	alarms	
10:39:49	9	17	21	1.205497742	0.981280565	0.867879152	machinery	incidental	
10:39:59	27	127	29	2.167774677	1.633837461	1.284159422	machinery	alarms	
10:40:09	24	26	52	1.465933561	0.947442949	0.904983342	machinery	alarms	
10:40:19	43	27	14	2.248245478	2.157669306	1.138263106	machinery	incidental	
10:40:29	24	27	134	1.974739313	1.908869267	1.027631998	machinery	alarms	annoyed
10:40:39	497	22	134	3.770386934	2.519325733	1.829385757	machinery	alarms	
10:40:49	27	28	11	2.055547476	1.777517438	1.709682584	machinery	alarms	
10:40:59	21	20	135	2.056849003	1.80619967	1.781591654	machinery	alarms	
10:41:09	10	8	6	1.597311258	1.551157355	1.162927866	machinery	incidental	
10:41:19	191	570	571	18.62853622	9.561905861	6.333662987	alarms	alarms	
10:41:29	28	24	32	3.129999876	2.237655401	1.133799434	alarms	alarms	
10:41:39	488	485	27	3.898231268	3.632342577	3.030939817	alarms	alarms	
10:41:49	232	92	231	22.61315727	18.33107948	12.34101868	alarms	alarms	
10:41:59	323	230	94	54.44478607	32.63043976	21.67127991	alarms	alarms	
10:42:09	463	91	92	11.01971054	8.352343559	3.330014706	alarms	alarms	
10:42:19	456	505	466	14.69853783	11.63384342	11.57468128	alarms	alarms	
10:42:29	490	493	492	4.898583889	3.240541935	3.147239447	alarms	alarms	
10:42:39	21	491	490	1.995904326	1.951113582	1.910012841	alarms	alarms	
10:42:49	28	27	22	3.666051865	2.646187305	1.476629615	alarms	alarms	
10:42:59	20	23	13	1.034578443	0.856300771	0.813100934	alarms	alarms	

User test data: Sebastiaan

11:25:36	28	55	27	1.969115615	0.96558249	0.852519095	incidental	alarms	
11:25:46	150	24	9	1.083293557	0.797544658	0.779554784	incidental	alarms	
11:25:56	13	10	55	1.642445207	1.271178722	0.692914367	incidental	incidental	
11:26:06	23	16	45	1.471314669	1.17630589	1.175912619	incidental	alarms	
11:26:16	518	8	527	2.126106739	2.025937319	1.866433859	incidental	conversation	
11:26:26	405	436	419	6.452401638	5.874213219	5.731959343	incidental	alarms	
11:26:36	58	18	35	11.48550892	11.34888268	5.59655714	incidental	alarms	
11:26:46	453	456	461	2.32869792	2.127718925	2.049471855	incidental	alarms	neutral
11:26:56	57	59	56	11.52961349	10.70764542	8.709519386	incidental	alarms	
11:27:06	12	28	7	4.126995087	2.08877039	1.658205867	incidental	alarms	
11:27:16	28	10	27	2.42629838	1.798405766	1.202665925	incidental	conversationtense	
11:27:26	25	28	57	1.60232532	1.316610694	1.270137072	incidental	alarms	
11:27:36	69	47	33	2.95507884	2.58637619	2.145574808	incidental	alarms	
11:27:46	12	13	27	1.514156699	0.93178153	0.871504366	incidental	incidental	
11:27:56	8	28	4	1.179131389	0.995558918	0.639388502	incidental	alarms	
11:28:06	581	66	28	6.356960297	2.897024155	2.589231253	incidental	alarms	
11:28:16	579	64	63	11.54938316	9.506243706	6.628759861	incidental	alarms	annoyed
11:28:26	27	20	14	2.3043468	1.05183959	0.895087123	incidental	incidental	
11:28:36	27	28	19	2.328874588	1.650809407	1.040549159	conversation	alarms	
11:28:46	28	21	56	1.378231287	0.791463077	0.623678684	conversation	alarms	displeased
11:28:56	78	146	40	10.5734396	6.748380661	6.364908218	conversation	alarms	
11:29:06	28	10	5	2.391965151	0.994561851	0.840453506	conversation	conversation	
11:29:16	63	62	126	8.254466057	7.274656773	3.203687906	conversation	alarms	
11:29:26	28	27	34	2.856121778	2.427476406	1.825103641	conversation	alarms	
11:29:36	28	9	21	2.884320021	1.230260849	0.929153979	conversation	incidental	
11:29:46	43	46	22	6.469465733	6.010754585	4.727680683	conversation	alarms	
11:29:56	38	43	41	6.253027916	3.661488056	3.647302866	conversation	alarms	
11:30:06	27	54	57	3.528403282	2.249177456	1.817506671	conversation	alarms	neutral
11:30:16	11	28	27	5.370411396	2.179083586	2.030194998	conversation	conversation	
11:30:26	53	12	63	15.89331055	1.966136575	1.282641292	conversation	conversation	
11:30:37	28	153	151	3.195996523	2.174797773	1.970234871	conversation	alarms	
11:30:47	27	152	399	3.083113909	2.606058836	2.307239771	conversation	alarms	
11:30:57	13	28	17	2.015722036	2.702246904	1.986137748	conversation	alarms	
11:31:07	28	112	30	3.332910776	2.305878639	2.01207757	conversation	alarms	
11:31:17	27	17	26	3.077638388	1.084090948	1.050943375	conversation	alarms	
11:31:27	28	20	16	3.309455156	1.112711668	0.801709235	conversation	alarms	neutral
11:31:37	27	71	74	4.305193424	3.982532501	3.721191168	machinery	alarms	
11:31:47	21	54	24	1.141684055	0.816473663	0.727447629	machinery	alarms	
11:31:57	27	151	152	3.296149254	2.892079353	2.689720154	machinery	alarms	
11:32:07	27	480	485	2.384147167	1.47986269	1.294992089	machinery	alarms	
11:32:17	136	132	20	2.529531717	2.182869673	1.419336438	machinery	alarms	
11:32:27	138	129	115	9.554092407	4.907641411	4.631075859	machinery	alarms	
11:32:37	73	11	72	2.276298761	2.185675859	1.921110988	machinery	conversation	
11:32:47	27	26	10	2.718696594	1.457232952	1.149860859	machinery	alarms	
11:32:57	129	54	22	2.023449421	1.746567488	1.443922639	machinery	alarms	
11:33:07	154	14	16	2.058388948	1.799431205	1.677916884	machinery	conversation	
11:33:17	27	10	151	3.752159119	1.928512931	1.87612009	machinery	conversation	
11:33:27	28	41	126	4.618263721	4.542482853	4.404816151	machinery	alarms	neutral
11:33:37	28	9	130	3.381670237	0.790322185	0.726473331	machinery	conversation	
11:33:47	12	17	40	4.70091486	3.033494473	2.020224094	machinery	conversation	
11:33:57	58	115	61	3.9207623	1.72622931	1.629024029	machinery	alarms	
11:34:07	27	55	58	2.656564474	2.546852827	1.564499021	machinery	alarms	
11:34:17	122	154	16	1.566611767	1.519802809	1.206100464	machinery	alarms	
11:34:27	28	125	27	2.851987123	2.626901627	2.219040632	machinery	alarms	
11:34:37	28	25	8	2.491541386	1.343864918	1.239791512	alarms	alarms	neutral
11:34:47	189	192	570	63.17152023	18.07365799	4.755170345	alarms	alarms	
11:34:57	25	15	22	1.228029132	0.759709477	0.747937441	alarms	alarms	stimulated
11:35:07	27	151	130	3.59883213	2.339906454	2.218961	alarms	alarms	
11:35:17	231	325	92	13.85807323	13.01589966	9.820515633	alarms	alarms	annoyed
11:35:27	328	321	330	11.19163704	9.75357151	6.940268517	alarms	alarms	
11:35:37	93	464	28	4.722037792	2.948964834	2.645795107	alarms	alarms	tense
11:35:47	518	529	490	10.32171535	9.806639671	6.019711018	alarms	alarms	
11:35:57	135	129	495	2.496124268	2.153523445	1.873429179	alarms	alarms	
11:36:07	489	487	488	2.231730223	2.174782038	2.016093493	alarms	alarms	
11:36:17	27	11	18	3.075688362	1.105739713	0.886654019	alarms	incidental	
11:36:27	20	22	1	1.023683429	0.995732427	0.729698062	alarms	alarms	
11:36:38	28	17	22	2.703814268	1.627921939	1.418863297	alarms	alarms	annoyed
11:36:48	46	50	41	58.60517883	30.10993385	22.87420082	alarms	alarms	
11:36:58	22	16	26	0.861316621	0.852836967	0.815629721	alarms	alarms	
11:37:08	28	17	63	4.332481861	2.10238719	1.671923995	alarms	alarms	
11:37:18	28	13	21	1.889196634	1.651843548	1.268960834	alarms	incidental	neutral
11:37:28	26	24	25	1.001429677	0.823279381	0.728381813	alarms	alarms	

User test data: Grace

11:59:39	28	27	9	4.623202801	2.426186562	1.078794718	incidental	alarms	
11:59:50	27	24	17	7.094645023	3.410418987	1.574731112	incidental	alarms	
12:00:00	52	136	153	6.153242588	1.650621057	1.537567973	incidental	alarms	
12:00:10	26	24	147	1.720538497	1.568169475	0.865384281	incidental	alarms	
12:00:20	56	51	26	3.003758907	2.502877474	1.643029213	incidental	alarms	
12:00:30	28	19	25	4.17519331	3.298220873	2.283214569	incidental	conversation	
12:00:40	22	25	20	3.1704247	1.586277127	1.251597881	incidental	alarms	
12:00:50	27	22	28	4.608717918	3.244668484	2.961169958	incidental	conversation	
12:01:00	23	28	25	2.182429314	2.154643774	1.992760062	incidental	alarms	
12:01:10	21	25	22	1.919779301	1.330961347	1.186939716	incidental	alarms	
12:01:20	27	10	45	7.359247684	2.281397581	2.134804964	incidental	conversation	
12:01:30	27	28	24	5.957940102	4.857526302	1.836008549	incidental	alarms	
12:01:40	53	54	61	13.33977127	11.89666843	7.421809673	incidental	alarms	
12:01:50	27	28	138	3.677411079	2.854299068	1.196129084	incidental	alarms	
12:02:00	21	28	20	2.441614866	1.58179009	1.084333539	incidental	alarms	
12:02:10	28	21	19	6.474571228	2.861078024	2.834282637	incidental	alarms	
12:02:20	28	203	45	7.768414021	3.959387541	3.095061064	incidental	alarms	
12:02:30	14	23	21	2.429658651	1.61804986	1.437155366	incidental	alarms	
12:02:40	27	24	26	5.499542236	2.563435316	2.514909029	conversation	conversation	
12:02:50	28	44	48	4.839356899	2.237853527	1.319298625	conversation	alarms	
12:03:00	49	128	50	11.87108421	10.7442131	8.419054031	conversation	alarms	
12:03:10	23	22	20	3.433101177	2.112426996	2.04229331	conversation	alarms	
12:03:20	28	54	30	4.907268047	3.335788012	2.74038291	conversation	alarms	
12:03:30	232	53	194	4.528813362	4.229128838	3.905090809	conversation	alarms	
12:03:40	58	57	40	8.287996292	7.892627716	3.754794359	conversation	alarms	
12:03:50	27	22	59	4.607179165	1.908158422	0.603257179	conversation	alarms	
12:04:00	58	137	55	4.16057682	3.644209862	3.15717721	conversation	alarms	
12:04:10	41	40	24	6.083827972	3.689831257	3.67198801	conversation	alarms	
12:04:20	44	40	45	9.784633636	4.929528236	4.852638721	conversation	alarms	
12:04:30	111	38	73	2.058978558	2.005007029	1.64692235	conversation	alarms	
12:04:40	28	19	30	5.395042419	1.810998917	1.302332163	conversation	conversation	
12:04:50	541	529	552	17.2049408	13.07628059	12.96273422	conversation	alarms	
12:05:01	27	21	20	2.784080744	1.395780683	1.323904514	conversation	alarms	
12:05:11	146	145	25	3.148075104	2.793838024	1.542563319	conversation	alarms	
12:05:21	28	53	56	5.673161507	5.567042351	3.4982059	conversation	alarms	
12:05:31	28	127	168	5.609119892	1.591896653	1.403025985	conversation	alarms	
12:05:41	28	21	19	3.03504014	2.029938936	0.95227164	machinery	alarms	neutral
12:05:51	135	27	128	8.071238518	7.314929008	3.466858149	machinery	alarms	
12:06:01	28	137	27	5.373339653	1.982198954	1.782373071	machinery	alarms	
12:06:11	21	19	28	2.181253672	1.113038778	0.738007307	machinery	alarms	
12:06:21	27	135	128	4.173387527	2.944423199	1.755921245	machinery	alarms	
12:06:31	27	16	51	5.847138882	1.135391474	0.780976355	machinery	alarms	
12:06:41	13	149	21	1.972770452	0.98650521	0.928998828	machinery	alarms	
12:06:51	136	27	130	5.366831779	4.095824718	3.854221106	machinery	alarms	
12:07:01	27	28	13	5.370240211	0.933895469	0.823529541	machinery	alarms	
12:07:11	27	19	29	5.371603489	1.497075796	1.439094186	machinery	conversation	
12:07:21	27	29	14	5.507139206	1.78002429	1.589178205	machinery	alarms	
12:07:31	27	22	33	4.221342087	1.783385873	1.399047375	machinery	conversation	
12:07:41	27	129	53	3.729072571	3.649280071	3.058070183	machinery	alarms	
12:07:51	27	129	132	4.369183064	2.629248142	2.236401558	machinery	alarms	
12:08:01	27	20	21	5.912745953	2.582180977	1.833705664	machinery	alarms	
12:08:11	133	53	21	3.094628096	2.745390177	2.60572052	machinery	alarms	
12:08:21	27	497	28	5.293276787	3.235704899	3.10655117	machinery	alarms	
12:08:31	9	27	14	7.266902924	4.628377438	2.63356328	machinery	alarms	
12:08:41	21	22	20	3.30622077	2.706644297	1.049539447	alarms	alarms	neutral
12:08:51	28	30	56	4.57575655	1.155439138	1.133819699	alarms	alarms	
12:09:01	9	19	21	2.832947493	1.86837554	1.745624661	alarms	alarms	
12:09:11	232	231	234	37.37086868	13.0040369	4.840088367	alarms	alarms	
12:09:21	46	45	231	28.89851761	20.03741074	12.21646881	alarms	alarms	
12:09:31	231	328	47	17.29243279	12.07922649	11.34845066	alarms	alarms	
12:09:41	91	87	89	16.13611031	7.662304401	7.047790527	alarms	alarms	
12:09:51	27	28	9	4.057697773	3.879599333	1.087979198	alarms	alarms	
12:10:01	27	56	28	4.371813774	3.534749746	3.121341228	alarms	alarms	
12:10:11	23	5	7	1.37717855	1.14567852	1.029925346	alarms	incidental	
12:10:21	27	125	503	4.893764496	1.307442546	1.264758229	alarms	alarms	
12:10:31	28	27	24	5.428726196	4.070519447	1.40096736	alarms	conversation	
12:10:41	19	48	15	1.417472363	1.332935929	1.122065544	alarms	alarms	
12:10:51	48	47	45	46.22812653	33.09703064	32.45941544	alarms	alarms	
12:11:01	507	53	27	14.86945724	8.799114227	6.372742176	alarms	alarms	
12:11:11	134	136	302	3.781892538	2.86131072	2.816670656	alarms	alarms	
12:11:21	28	22	508	7.127122402	2.285135269	1.682128429	alarms	alarms	
12:11:31	28	25	11	5.143651962	1.54765594	1.225855112	alarms	alarms	annoyed

User test data: G.Young

14:20:06	67	27	19	7.540002346	6.797169685	4.774970055	incidental	alarms	
14:20:16	28	9	27	3.746375561	1.423963189	1.030683041	incidental	conversation	
14:20:26	11	18	27	2.572280407	2.555030346	1.985975385	incidental	alarms	
14:20:36	27	12	149	3.094005346	1.33594954	1.163872957	incidental	conversation	
14:20:46	27	28	1	6.982021809	1.277415037	1.232492089	incidental	alarms	
14:20:56	145	25	23	1.474550605	1.313052893	1.151423693	incidental	alarms	
14:21:06	14	43	11	0.910595953	0.760030389	0.698426902	incidental	alarms	
14:21:16	27	26	24	5.603274345	2.860610723	1.747789741	incidental	conversation	
14:21:26	28	53	50	5.087116718	3.275455952	2.189687729	incidental	alarms	
14:21:36	44	46	41	7.94879961	4.669539928	4.141410351	incidental	alarms	
14:21:46	14	16	501	1.648228049	1.117802024	0.948689699	incidental	alarms	
14:21:56	143	21	20	2.46069932	2.130955458	1.570138693	incidental	alarms	
14:22:07	28	21	9	4.282043934	2.245159864	2.123203278	incidental	alarms	
14:22:17	27	29	26	2.802129269	1.345947981	1.243124127	incidental	alarms	
14:22:27	27	29	28	3.77788353	1.30975461	0.80465734	incidental	alarms	
14:22:37	515	575	540	10.93678284	4.823992252	4.630722523	incidental	alarms	
14:22:47	151	150	53	28.11200523	11.50099087	11.04461765	incidental	alarms	
14:22:57	28	27	30	3.810489178	1.321841955	1.225100756	incidental	conversation	
14:23:07	28	21	25	3.588833332	3.297245979	1.623665452	conversation	alarms	
14:23:17	28	5	31	4.337031364	1.542915463	1.365833402	conversation	conversation	
14:23:27	28	10	25	3.826195002	1.292457223	1.056930423	conversation	conversation	
14:23:37	27	139	39	6.210716724	4.731399536	3.01632905	conversation	alarms	
14:23:47	54	163	161	15.39936161	12.98440456	10.37187481	conversation	alarms	
14:23:57	27	39	56	4.574797153	1.054690123	0.826096416	conversation	alarms	
14:24:07	28	30	7	1.046790719	1.044104934	1.027809262	conversation	alarms	
14:24:17	29	26	27	3.560585737	2.895953417	2.722746849	conversation	alarms	
14:24:27	42	145	34	7.475886822	6.122803211	5.115092278	conversation	alarms	
14:24:37	27	49	48	5.124500275	4.331405163	4.283764362	conversation	alarms	
14:24:47	551	554	527	5.142971992	4.753363132	4.356653214	conversation	alarms	
14:24:57	52	51	24	7.251577377	6.546813011	2.458556652	conversation	alarms	
14:25:07	55	27	14	3.130869627	2.569172144	1.187230706	conversation	alarms	
14:25:17	141	151	28	6.358185768	5.438684464	3.437332392	conversation	alarms	
14:25:27	27	145	147	6.514137268	3.42183423	3.269988537	conversation	alarms	
14:25:37	12	9	131	3.33723855	3.078137398	2.723371983	conversation	conversation	
14:25:47	117	115	124	14.22468281	12.14705563	9.012540817	conversation	alarms	
14:25:57	156	151	73	4.37443161	4.15144825	3.061911821	conversation	alarms	
14:26:07	27	12	9	2.698652506	1.761046767	0.991763651	machinery	conversation	
14:26:17	143	147	149	2.456121922	2.376675844	2.282536983	machinery	alarms	
14:26:27	152	151	150	5.746739388	4.245556831	2.853684425	machinery	alarms	
14:26:37	28	21	34	4.751237392	0.969172716	0.962122858	machinery	conversation	
14:26:47	136	133	138	5.149169922	4.903017044	4.793756485	machinery	alarms	
14:26:57	28	23	119	5.076906681	2.424602509	1.735145688	machinery	alarms	
14:27:07	25	51	28	1.921126485	1.898246646	1.811532497	machinery	alarms	
14:27:17	28	27	21	4.865671635	3.02214098	1.460920215	machinery	alarms	neutral
14:27:27	27	151	149	5.026008129	3.743047714	2.436625957	machinery	alarms	
14:27:37	27	128	151	5.550057888	2.681246758	1.777306557	machinery	alarms	
14:27:47	134	138	136	7.474253178	6.976080418	5.371792316	machinery	alarms	
14:27:57	28	21	15	4.47353363	1.330658317	0.850514829	machinery	alarms	
14:28:08	27	26	55	2.151736975	1.644249439	1.49890089	machinery	alarms	
14:28:18	28	120	13	3.414346457	1.948920846	0.725260794	machinery	alarms	
14:28:28	19	150	21	1.608497143	1.490173936	0.926643193	machinery	alarms	
14:28:38	27	138	140	3.939167976	1.987244129	1.635384798	machinery	alarms	
14:28:48	27	11	28	3.367008448	1.446322799	1.245889306	machinery	conversation	
14:28:58	27	13	15	2.18998456	0.940152109	0.77273941	machinery	incidental	
14:29:08	501	53	549	9.223461151	8.79083252	8.472768784	alarms	alarms	
14:29:18	489	28	493	4.883031368	4.495944977	2.865291834	alarms	alarms	neutral
14:29:28	189	27	185	9.122590065	3.791307449	2.567850351	alarms	alarms	displeased
14:29:38	142	512	229	16.28835106	15.96070004	12.4248333	alarms	alarms	
14:29:48	323	91	321	31.82233238	11.99202633	11.57143116	alarms	alarms	tense
14:29:58	91	322	324	18.06955338	9.634752274	9.087903023	alarms	alarms	
14:30:08	93	95	27	17.91080284	4.426354408	3.258761644	alarms	alarms	tense
14:30:18	27	20	29	6.169540405	1.720032692	1.588703871	alarms	conversation	frustrated
14:30:28	56	82	24	5.73029089	1.951060176	1.811835051	alarms	alarms	
14:30:38	28	27	9	4.102149487	3.575804234	2.356558561	alarms	alarms	
14:30:48	27	28	12	4.723105431	2.164969921	1.434097767	alarms	alarms	neutral
14:30:58	17	10	54	1.297272444	1.237224579	1.066417575	alarms	incidental	
14:31:08	37	8	10	3.180339813	2.068088055	1.736163735	alarms	conversation	
14:31:18	43	46	47	18.67030144	15.32499886	11.50626659	alarms	alarms	annoyed
14:31:28	28	148	13	3.252566814	1.61956501	1.449410439	alarms	alarms	tense
14:31:38	9	19	5	2.238597631	1.305418134	1.151694894	alarms	incidental	frustrated
14:31:48	27	29	31	5.278033733	1.612689018	1.419395566	alarms	conversation	
14:31:58	13	22	87	1.560978413	1.514557958	1.456640124	alarms	alarms	neutral

User test data: Jooyoung

14:20:06	67	27	19	7.540002346	6.797169685	4.774970055	incidental	alarms	
14:20:16	28	9	27	3.746375561	1.423963189	1.030683041	incidental	conversation	
14:20:26	11	18	27	2.572280407	2.555030346	1.985975385	incidental	alarms	
14:20:36	27	12	149	3.094005346	1.33594954	1.163872957	incidental	conversation	
14:20:46	27	28	1	6.982021809	1.277415037	1.232492089	incidental	alarms	
14:20:56	145	25	23	1.474550605	1.313052893	1.151423693	incidental	alarms	
14:21:06	14	43	11	0.910595953	0.760030389	0.698426902	incidental	alarms	
14:21:16	27	26	24	5.603274345	2.860610723	1.747789741	incidental	conversation	
14:21:26	28	53	50	5.087116718	3.275455952	2.189687729	incidental	alarms	
14:21:36	44	46	41	7.94879961	4.669539928	4.141410351	incidental	alarms	
14:21:46	14	16	501	1.648228049	1.117802024	0.948689699	incidental	alarms	
14:21:56	143	21	20	2.46069932	2.130955458	1.570138693	incidental	alarms	
14:22:07	28	21	9	4.282043934	2.245159864	2.123203278	incidental	alarms	
14:22:17	27	29	26	2.802129269	1.345947981	1.243124127	incidental	alarms	
14:22:27	27	29	28	3.77788353	1.30975461	0.80465734	incidental	alarms	
14:22:37	515	575	540	10.93678284	4.823992252	4.630722523	incidental	alarms	
14:22:47	151	150	53	28.11200523	11.50099087	11.04461765	incidental	alarms	
14:22:57	28	27	30	3.810489178	1.321841955	1.225100756	incidental	conversation	
14:23:07	28	21	25	3.588833332	3.297245979	1.623665452	conversation	alarms	
14:23:17	28	5	31	4.337031364	1.542915463	1.365833402	conversation	conversation	
14:23:27	28	10	25	3.826195002	1.292457223	1.056930423	conversation	conversation	
14:23:37	27	139	39	6.210716724	4.731399536	3.01632905	conversation	alarms	
14:23:47	54	163	161	15.39936161	12.98440456	10.37187481	conversation	alarms	
14:23:57	27	39	56	4.574797153	1.054690123	0.826096416	conversation	alarms	
14:24:07	28	30	7	1.046790719	1.044104934	1.027809262	conversation	alarms	
14:24:17	29	26	27	3.560585737	2.895953417	2.722746849	conversation	alarms	
14:24:27	42	145	34	7.475886822	6.122803211	5.115092278	conversation	alarms	
14:24:37	27	49	48	5.124500275	4.331405163	4.283764362	conversation	alarms	
14:24:47	551	554	527	5.142971992	4.753363132	4.356653214	conversation	alarms	
14:24:57	52	51	24	7.251577377	6.546813011	2.458556652	conversation	alarms	
14:25:07	55	27	14	3.130869627	2.569172144	1.187230706	conversation	alarms	
14:25:17	141	151	28	6.358185768	5.438684464	3.437332392	conversation	alarms	
14:25:27	27	145	147	6.514137268	3.42183423	3.269988537	conversation	alarms	
14:25:37	12	9	131	3.33723855	3.078137398	2.723371983	conversation	conversation	
14:25:47	117	115	124	14.22468281	12.14705563	9.012540817	conversation	alarms	
14:25:57	156	151	73	4.37443161	4.15144825	3.061911821	conversation	alarms	
14:26:07	27	12	9	2.698652506	1.761046767	0.991763651	machinery	conversation	
14:26:17	143	147	149	2.456121922	2.376675844	2.282536983	machinery	alarms	
14:26:27	152	151	150	5.746739388	4.245556831	2.853684425	machinery	alarms	
14:26:37	28	21	34	4.751237392	0.969172716	0.962122858	machinery	conversation	
14:26:47	136	133	138	5.149169922	4.903017044	4.793756485	machinery	alarms	
14:26:57	28	23	119	5.076906681	2.424602509	1.735145688	machinery	alarms	
14:27:07	25	51	28	1.921126485	1.898246646	1.811532497	machinery	alarms	
14:27:17	28	27	21	4.865671635	3.02214098	1.460920215	machinery	alarms	neutral
14:27:27	27	151	149	5.026008129	3.743047714	2.436625957	machinery	alarms	
14:27:37	27	128	151	5.550057888	2.681246758	1.777306557	machinery	alarms	
14:27:47	134	138	136	7.474253178	6.976080418	5.371792316	machinery	alarms	
14:27:57	28	21	15	4.47353363	1.330658317	0.850514829	machinery	alarms	
14:28:08	27	26	55	2.151736975	1.644249439	1.49890089	machinery	alarms	
14:28:18	28	120	13	3.414346457	1.948920846	0.725260794	machinery	alarms	
14:28:28	19	150	21	1.608497143	1.490173936	0.926643193	machinery	alarms	
14:28:38	27	138	140	3.939167976	1.987244129	1.635384798	machinery	alarms	
14:28:48	27	11	28	3.367008448	1.446322799	1.245889306	machinery	conversation	
14:28:58	27	13	15	2.18998456	0.940152109	0.77273941	machinery	incidental	
14:29:08	501	53	549	9.223461151	8.79083252	8.472768784	alarms	alarms	
14:29:18	489	28	493	4.883031368	4.495944977	2.865291834	alarms	alarms	neutral
14:29:28	189	27	185	9.122590065	3.791307449	2.567850351	alarms	alarms	displeased
14:29:38	142	512	229	16.28835106	15.96070004	12.4248333	alarms	alarms	
14:29:48	323	91	321	31.82233238	11.99202633	11.57143116	alarms	alarms	tense
14:29:58	91	322	324	18.06955338	9.634752274	9.087903023	alarms	alarms	
14:30:08	93	95	27	17.91080284	4.426354408	3.258761644	alarms	alarms	tense
14:30:18	27	20	29	6.169540405	1.720032692	1.588703871	alarms	conversation	frustrated
14:30:28	56	82	24	5.73029089	1.951060176	1.811835051	alarms	alarms	
14:30:38	28	27	9	4.102149487	3.575804234	2.356558561	alarms	alarms	
14:30:48	27	28	12	4.723105431	2.164969921	1.434097767	alarms	alarms	neutral
14:30:58	17	10	54	1.297272444	1.237224579	1.066417575	alarms	incidental	
14:31:08	37	8	10	3.180339813	2.068088055	1.736163735	alarms	conversation	
14:31:18	43	46	47	18.67030144	15.32499886	11.50626659	alarms	alarms	annoyed
14:31:28	28	148	13	3.252566814	1.61956501	1.449410439	alarms	alarms	tense
14:31:38	9	19	5	2.238597631	1.305418134	1.151694894	alarms	incidental	frustrated
14:31:48	27	29	31	5.278033733	1.612689018	1.419395566	alarms	conversation	
14:31:58	13	22	87	1.560978413	1.514557958	1.456640124	alarms	alarms	neutral

User test data: Jiwon

15:34:25	27	28	29	4.649060249	3.5375247	1.218516946	incidental	alarms	
15:34:35	25	19	7	2.3171525	1.434974551	1.1033988	incidental	alarms	
15:34:45	28	54	52	9.03886795	3.467441797	2.6831007	incidental	alarms	
15:34:55	28	25	13	2.932265043	1.768224478	1.088409185	incidental	alarms	
15:35:05	57	47	25	3.322274208	1.996452928	1.842850566	incidental	alarms	
15:35:15	27	56	60	5.955713749	3.307775259	1.475275397	incidental	alarms	
15:35:25	27	26	29	3.998114586	1.858543515	1.144407749	incidental	conversation	
15:35:35	48	57	59	3.004626751	2.824011803	2.805366755	incidental	alarms	
15:35:45	28	13	32	8.383062363	1.956995487	1.780521393	incidental	conversation	
15:35:55	28	27	54	5.997546673	2.373152494	1.949213386	incidental	conversation	
15:36:05	56	22	550	5.444364548	2.541113615	1.955647111	incidental	alarms	
15:36:15	27	544	29	6.433924675	3.270435095	1.539159417	incidental	alarms	
15:36:25	58	50	26	6.076436043	5.921990871	5.845228195	incidental	alarms	
15:36:35	9	135	18	1.000345826	0.955043793	0.919612169	incidental	alarms	
15:36:45	25	11	17	3.071307898	2.411989927	1.38168931	incidental	incidental	
15:36:55	27	248	191	5.247439861	3.736461878	3.528650045	incidental	alarms	
15:37:05	55	28	537	8.267560959	4.657581329	3.78694129	incidental	alarms	
15:37:15	27	12	29	5.759988785	1.704980254	1.097588181	incidental	conversation	
15:37:25	28	27	30	5.115778446	1.901258707	1.00969243	conversation	conversation	
15:37:35	43	45	42	27.57291222	13.56744957	11.17013359	conversation	alarms	neutral
15:37:45	28	47	128	10.11931705	9.106186867	7.741375923	conversation	alarms	
15:37:55	28	286	445	4.585098267	3.60614419	3.481886625	conversation	alarms	
15:38:06	28	32	16	6.063032627	2.08506918	2.057082653	conversation	alarms	
15:38:16	57	56	506	10.86387348	5.238829136	5.091310024	conversation	alarms	
15:38:26	38	43	13	7.207251549	2.237887859	1.883550525	conversation	alarms	
15:38:36	28	17	55	3.028325558	2.236908197	1.73830092	conversation	alarms	
15:38:46	60	142	55	13.47841549	6.184933662	3.770647764	conversation	alarms	
15:38:56	496	83	51	7.904444695	6.382527828	4.774605274	conversation	alarms	
15:39:06	44	43	28	17.81204224	10.48321533	9.190882683	conversation	alarms	
15:39:16	28	474	56	6.711685658	3.99195528	3.600019693	conversation	alarms	
15:39:26	25	141	9	1.25385654	1.188026309	1.124331951	conversation	alarms	
15:39:36	528	552	511	17.94819069	17.11728096	16.60152245	conversation	alarms	
15:39:46	119	122	117	16.9025116	10.84471416	5.798255444	conversation	alarms	
15:39:56	27	56	26	2.114571571	1.904428005	1.383525729	conversation	alarms	
15:40:06	130	51	160	3.12309742	2.79638505	2.541218758	conversation	alarms	
15:40:16	25	23	10	1.849452376	1.601099133	1.479453564	conversation	incidental	
15:40:26	25	13	16	1.686682463	1.497665644	1.385080218	machinery	incidental	neutral
15:40:36	122	127	130	4.781165123	4.170338154	3.406019211	machinery	alarms	
15:40:46	27	133	10	5.337959766	3.382359982	2.214487076	machinery	alarms	
15:40:56	28	30	13	4.319637775	1.067523003	0.751780272	machinery	alarms	
15:41:06	28	14	19	4.624732018	1.944813967	1.380652785	machinery	incidental	
15:41:16	27	57	165	7.22336483	2.407237768	2.385436535	machinery	alarms	
15:41:26	53	63	56	31.24959946	18.62298012	17.62074089	machinery	alarms	
15:41:36	129	27	11	8.761685371	6.636906624	3.185451984	machinery	alarms	
15:41:46	28	30	20	20.37532616	4.585150242	2.501611233	machinery	alarms	
15:41:56	28	169	12	2.873136282	2.111442804	2.06038332	machinery	alarms	
15:42:06	13	21	15	2.72927022	2.121870279	1.985887527	machinery	alarms	
15:42:16	130	487	484	5.87142992	4.82656765	4.530397892	machinery	alarms	
15:42:26	26	18	120	1.59770298	1.450454116	1.211845756	machinery	alarms	
15:42:36	27	124	28	6.60850811	1.716716766	1.634809613	machinery	alarms	
15:42:46	27	28	29	5.448685169	3.718815327	1.521103024	machinery	conversation	
15:42:56	28	27	498	6.086047649	1.938821077	1.772669911	machinery	alarms	
15:43:06	28	493	482	5.296457767	5.084425926	4.822442532	machinery	alarms	
15:43:16	27	28	487	5.019150734	3.654251337	2.316306591	machinery	alarms	
15:43:26	27	12	55	6.546087265	1.410562396	1.064283252	alarms	conversation	
15:43:36	28	27	14	6.258899212	1.610511065	1.367621899	alarms	alarms	
15:43:46	11	51	15	2.104042053	1.791398883	1.774894357	alarms	alarms	neutral
15:43:56	232	325	229	8.907125473	8.043885231	6.641419411	alarms	alarms	
15:44:06	231	94	91	18.65148354	9.791207314	9.710584641	alarms	alarms	
15:44:16	94	325	326	24.42816734	23.08727264	10.76447296	alarms	alarms	
15:44:26	92	91	90	21.23665619	14.96143913	7.315200329	alarms	alarms	
15:44:36	27	11	22	5.549755573	1.556075692	1.19172585	alarms	conversation	
15:44:47	28	27	11	4.871792316	4.405148506	1.601672292	alarms	alarms	
15:44:57	27	28	8	4.279597759	3.785203218	1.028474808	alarms	alarms	
15:45:07	28	9	25	4.821256161	1.543714523	1.338441849	alarms	conversation	
15:45:17	8	10	143	2.656351328	2.121887922	1.456040859	alarms	conversation	
15:45:27	28	9	10	5.880917072	1.795799613	1.231263876	alarms	conversation	
15:45:37	45	41	43	30.01897812	9.818226814	8.804474831	alarms	alarms	
15:45:47	27	53	524	6.979133606	4.921055794	3.606314898	alarms	alarms	
15:45:57	27	28	29	5.821588039	1.737487912	1.147939801	alarms	conversation	
15:46:07	28	27	490	4.448826313	3.800175905	1.440690994	alarms	alarms	
15:46:17	27	28	34	6.827897072	3.88398838	1.06381762	alarms	conversation	