OULUN YLIOPISTO
UNIVERSITY of OULU

Embedded Mobile Application For Controlling Acoustic Panels

University of Oulu
Faculty of Information Technology and Electrical Engineering / Degree program in information processing science
Master's Thesis
Juho Rantaharju
04.01.2020

Abstract

This thesis work is about acoustic panels and planning a software that would control these kinds of panels. The software is supposed to take information from the panels and then use that information for moving the acoustic panels to a desired location. The application is for mobile environment for both smart phones and tablets. This means that there are some constraints for the software such as scaling the panels so that all the panels can be used when moving the panels. This work introduces heuristic and design science theory and builds the application plan as an artifact from there onwards. The plan is based upon the original requirements for this application. This plan for the application meets the requirements set upon it by the customer.

The plan was created so that the basic functionalities that were discussed with the customer were satisfied. This included connection to panels, drawing a scaling panel view, moving panels, centring panels and so forth. The application was evaluated with two sets of heuristics. First one was the heuristics created by Nielsen 1995 and second heuristic was self-built. Nielsen's heuristics were meant for a more general usage while the set of heuristics that were self-build were meant for more general usage.

The heuristic evaluation provided results which were that the application needs at least more error prevention, documentation and a better way or representing panels actual physical location on the wall. Error prevention was a major issue in a case that one or more of the panels were broken and needed to be fixed. Documentation was more of an issue from the user's perspective in case some of the actions or error messages were such that the user did not understand them. Last issue of presenting the panel positions better in relation to the physical wall was an issue basically because the user needs to know where the panels are without too much difficulty. If the user is confused about panel location, they cannot be sure which panels to move.

These issues were discussed in the second iteration of the plan for this application. The second iteration was done in writing and a picture of the new user interface after the heuristic evaluation was done. This iteration discussed and solved these problems. For the limitations of this work there were issues with author doing the heuristic evaluation while not being an expert, implementation not being done in the scope of this work and implementation details not being discussed. For future research, the implementation should be done and the heuristics that were self-built need more though put into them.

Keywords
Acoustic panels, design science, evaluation, heuristics, mobile control application

Supervisor
Yliopistonlehtori Ari Vesanen

Contents

1. Introduction

The issue of this research is to build a plan for an application that moves acoustic panels in different positions to control acoustics in music halls. The reason why this plan is done is to understand more about mobile control software and its functionality. The contribution made is a plan for new kind of mobile control application that is meant for a relatively specific task. There have previously been build mobile control application but this plan for a mobile control application differentiates itself from others by the specific use of it. Also, there can still be more studies done in the field of mobile control applications since relatively to its importance there are only a few studies done.

This application plan will be developed as a design science work in which there is theory represented both for the construct which is the plan and for usability of the application in terms of heuristics. The application usability is one of the issues that is important because the application is going to be used by non-expert users in order to move acoustic panels in correct locations based upon wanted acoustics. The way this plan is done is based upon requirements analysis, system analysis, functional analysis, and design synthesis. The usability of the software is evaluated based on Nielsen’s (1995) heuristics and self-build heuristics for specifically mobile control applications. The specific heuristics are based upon previous work from i.e. Quinones & Rusu (2017) in building heuristics and they are used because Nielsen’s heuristics can be too general for mobile control application.

The research problem statement is how to make a mobile control application that is as usable by non-expert users as possible for controlling acoustic panels. This about making a plan that considers requirements that client has made, and theory discussed in the work. The approach for the research problem is answering research questions which are:

RQ1: How to create an efficient and usable mobile application for controlling acoustic panels?

RQ2: Based upon heuristics, what are the usability problems of this application?

RQ3: How, based upon heuristics, to improve the application operation for non-expert user?

Answering these research questions gives more understanding of mobile control software in general and for this specific case.

Usability is one of the important parts in this software due to the nature of the application. There can be as much as sixty panels that need to be controlled and the controlling is done via a mobile phone or tablet. The usability of this plan will be evaluated against heuristics. In Chapter 2 there will be theory on available heuristics and heuristics building. First there is theory in building heuristics and heuristic evaluation. After that Nielsen’s heuristics will be introduced. Then there will be self-build heuristics that are meant specifically for evaluation of mobile control application such as the one being planned in this thesis. After that there will be introduction to similar applications where heuristic evaluation was done. These heuristics will give an estimation of how the actual implementation of the plan would work in real life and based upon this evaluation the plan can be iterated.

Usability issues that come up with this application are related to scalability. Many panels will not fit on the screen of a tablet or mobile phone. This means that the application must scale based upon the number of panels. Initially panels can be made smaller for the screen but when number of panels becomes too high there must be grouping of the panels in different views. The screen of a mobile phone is even smaller than that of a tablet and therefore, scaling must be done even more aggressively for a mobile phone environment. There are two usability issues: first if there are too many panels shown in the screen the buttons can become so small that they are not intelligible or clickable and second if panels are divided into too many view groups there can be a situation where information is so divided that it will confuse the user.

For the constructing of the plan design science research approach will be used. This approach is discussed in depth in research methods Chapter 3. The basis of the theory is Hevner et al (2004) but other papers, that update that paper, will be discussed. First in the Chapter 3 research problem and research questions are introduced. There is substantial amount of theory for the artifact building. The artifact will be the plan for this mobile control application. This plan will be then evaluated based upon heuristics discussed in Chapter 2. In Chapter 4 there will be the construction of the artifact. The artifact building includes requirements analysis, functional analysis, system analysis and design synthesis.

In Chapter 5 there is evaluation of this artifact based upon heuristics and after that reiteration of the artifact is going to be discussed. The basic issues with heuristics evaluation are that when some problems are found the plan needs to be reiterated to solve these problems. Chapter 6 is discussion chapter in which the whole of the research is being discussed and combined to understand where it solves the problems that were found and what could be added. Also, limitations of the research will be discussed here. Chapter 7 is conclusions of the research and future research.

2. Heuristics and Artifact Evaluation

Usability is an important aspect for the application that will be planned in this work because of the application user being a non-technical user. The usability concern needs to be addressed and theory of usability evaluation needs to be considered. The application needs to work efficiently and with good usability. This section is an overview of usability and heuristics. This theory will be used for evaluation of the artifact.

According to the article Bevan, Carter & Harker (2015) usability is defined by the ability of being used and how the entity can be used for the purpose it is meant to be used. Quinones & Rusu (2017) say usability should be designed into the product and it can be inspected with usability inspections and with usability tests. One of the widely used methods for usability testing and inspecting is heuristic evaluation. In this method heuristic principles and usability heuristics are used for evaluating and considering how usable the software application is. According to Quinones & Rusu (2017) there have been several heuristics developed for usability by different authors. Mostly research has been developing new heuristics to solve these problems.

There is a standard of ISO 9241-11 that has been a good guide to usability and its definition and application in different fields according to Bevan et al (2015). On this standard the basic definition of usability is about the extent with which a product is and can be used by a user for achieving different and specified goals with effectiveness, efficiency and satisfaction in a specific context of a use of that product, software or application. There are different terms related to the usability of the product. When usability is evaluated these terms become very important to define. In the ISO 9241-11 standard these are defined as following: A user is a person who interacts with the product in place. Goal is defined as an intended outcome of the use. Effectiveness is defined as being the accuracy and completeness with which the user can achieve specified goals. Efficiency is about accuracy and completeness when achieving certain goals with a product. Satisfaction is freedom from discomfort and having positive attitudes when using the product for a goal achieved. Context of use means the users, tasks, equipment, and the environment both physical and social in which the product is being used in. These definitions are useful for the study of usability even though according to ISO 9241-11 there still is no generally accepted definition for usability because of the complexity of its nature that is hard to define and describe with one definition. (ISO 9241 -11, 1998)

For the usability inspection for this work heuristics will be used for evaluation of the artifact. The evaluation itself based upon these heuristics will be done by the author of this work. There will be no outside evaluation. After the heuristic evaluation is done one more iteration of the requirements and plan of the application will be done. There will be only one heuristic evaluation from which the results will be used for the iteration. The iteration will be separate subchapter in the plan.

2.1 Heuristics building

In this subchapter there is a look at how heuristics are developed and what are the advantages and disadvantages of using them. Also, evaluation of usability problem severity is discussed. This kind of severity discussion is important when looking at usability problems based upon heuristics because it is important to find out how much the usability problem effects the application that is being evaluated.

According to Anganes, Pfaff, Drury & Toole (2016) heuristic evaluation is one of the most popular inspection methods. There are many heuristics developed for a specific domain to evaluate specific and more general features of software. These heuristics should be well-designed, easy to use and they also should domain specific in a way that helps in identifying domain related usability problems. Methodology specified by Rusu et al (2011) has been used a lot of developing heuristics that are specific to application. Quinones et al (2018) is however critical of this methodology in that according to them this methodology has problems with explaining stages and creates confusion on how to iterate and apply it properly.

2.1.1 Heuristic development process

There are two stages for this kind of heuristic development process. First there is a necessity to extract information and second this information is transformed into heuristics according to Hermawati & Lawson (2016). According to Van Greunen, Yeratziotis & Pottas (2011) heuristics can be validated in two phases. Validation should be done with first a group of experts and second via case studies. However, when it comes to specific domains of applications there is no consensus about how to develop heuristics according to Sim, Read & Cockton (2009). Also, the same paper talks about how this leads to difficulty of understanding and using of heuristics and evaluating usability in a specific way.

According to Lechner, Fruhling, Petter & Siv (2013) there is no evidence that in development of heuristics any specific methodologies have been used and according to Inostroza et al (2016) the whole process of developing usability heuristics has not yet been formalized. There are systematic step by step processes but also there are informal processes following steps and performing activities in a non-systematic way. Mostly when working on new heuristics they are established as an extension of what has been done before such as Nielsen's (1995) heuristics. When these new sets are built studies usually will not document the steps taken to create the new heuristics according to Quinones & Rusu (2017). Also, most studies do not specify formal and informal processes they used for creating heuristics and the methodology is hidden.

According to Rusu, Ronzagliolo, Rusu & Collazos (2011) there is no totally accepted process that could be used for validation of the heuristics that has been developed. Also, according to Inostroza, Rusu, Roncagliolo, Rusu & Collazos (2016) there has been no formalization of a heuristic development method. This does not mean that there is absolutely no methodology but that there is no formal agreed upon methodology. There is no agreement of the process to formulate, specify, validate, and refine the usability heuristics proposed. Most of the heuristics being used now have been developed based on researchers experience and using methods that have previously been used for other purposes according to Jaferian, Hawkey, Sotirakopulos, Velez-Rojas & Beznovos (2014)

For example, Hub & Capkova (2010) propose a methodology for creating heuristics for specific domains. In it experts are for specification and revision of heuristics and end users for evaluation and revision of the heuristics. There are also other methodologies created in articles such as Yeratziotis, Pottas & Van Greunen (2012), Franklin et al (2014) and Lechner et al (2014). These methodologies were used as a basis for Quinones et al (2018) methodology. In the article Quinones et al (2018) develops a new methodology by making changes to previous methodologies. This is done by adding new steps, definitions, and diagrams. This should improve the specification of the methodology. As an example, experimental data shows that Nielsen's heuristics have problems in their definitions, and they cannot be used to evaluate all the features of a specific application according to Quinones et al (2014).

Most studies use steps to develop heuristics. Different approaches that are used for developing these heuristics for usability are: existing heuristics, methodologies, literature review, usability problems, mixing processes, guidelines and principles, interviews, and theories. According to Quinones & Rusu (2017) they are used to collect information about existing heuristics and pointing out limitations of using them in the context of evaluating usability within the specific problem domain. Researchers are exploring information and features within the domain that the new heuristic will be used and proposing a set of new heuristic.

There is a possibility with Nielsen's (1995) heuristics to assess general usability of an application or system. Issues can be assessed such as error prevention, user control and freedom, flexibility and efficiency of use and others. It should be considered what must be added to new heuristics to cover the gap between new heuristics and traditional heuristics. There are a few important issues for this: first same mistakes should not be made when developing new heuristics and second refining and improving existing heuristics is better than creating new sets and this should be done by adding and modifying certain existing elements. Another issue is providing and proposing a totally new set of heuristics when the problem domain needs it and cannot be worked on with existing set. (Quinones & Rusu, 2017).

According to Quinones & Rusu (2017) there has been a lot of development of heuristics for specific mobile domains because of the traditional heuristics do not do well in evaluating the specific features of mobile applications. New sets of heuristics have been developed for mobile applications. This means that specific heuristics can be used to evaluate different problem domains in mobile applications. It is important to know which heuristics are most suitable for mobile domains. There is a need to discuss how these heuristics make mobile applications better. Most heuristics for mobile domain are informal. The heuristics are mainly focused on existing heuristics, usability problems, literature reviews and guidelines

2.1.2 Advantages and disadvantages of heuristics

There are many advantages of using heuristics for evaluation of the product interface. The number one advantage is that it is less expensive when it comes to other methods in terms of time, number of usability experts and resource. A second advantage of the heuristic evaluation is that it does not require extensive planning to execute. A third advantage is that it can be also applied in the early stages of the software development process. This is all the way from prototype to executable systems. A fourth advantage is that heuristics evaluation can be used to find many problems both critical and less critical. A fifth and final advantage is that it does not require any users to execute. (Nielsen & Molich, 1990; Scholtz, 2009)

There are however some disadvantages for this kind of heuristic evaluation. First disadvantage is that the heuristics evaluators must have experience and adequate knowledge to perform this kind of evaluation for the user interface. Second issue is that the evaluators might not have enough understanding and knowledge of the product interface to know what tasks it is meant to perform and this way it can be hard to find usability problems with the product interface. Third problem is that based upon the previous one there can be a problem where evaluators cannot give good feedback for solving these problems. This is because heuristic evaluation does not offer a systematic way to generate solutions for the problems that it identifies. (Nielsen & Molich, 1990; Jaspers, 2009)

2.1.3 Evaluation of severity of usability problems

When heuristics are used to evaluate problems there is an inspection for frequency, severity, and criticality of the problems according to Nielsen and Molich (1990). Recommendation for this evaluation is that five different experts perform heuristics evaluation. For this each expert will separately find usability problems and prepare a list of usability problems based on the heuristics. After this expert work together to create a list of issues for usability problems. After this all evaluators individually estimate problems in terms of their frequency, severity, and criticality according to Jaspers (2009). Finally, a list of problems is collected, and a summary is provided about the problems in a single report. This report should have in it the problems that were found and collected and different suggestions for solutions to these problems that can help designers of the software.

For different usability problems there are three scales that need to evaluated according to Nielsen and Molic (1990). First there is a severity scale: this means that the severity of the usability problem must be identified. Problem can be identified that prevent the product interface from functioning properly. Second scale is frequency scale: this scale identifies the degree of how many times the usability issues occurs in the interface of the product. It identifies problems that are the most frequent, common, and recurrent for the product interface. Third scale is the criticality scale: for this scale according to Nielsen and Molic (1990) severity and frequency of the problems are summed up. In this way the problems that are most severe and most frequent can be identified and dealt with by development team.

This explanation of heuristics comes to be important when heuristics will be used to evaluate the plan that has been made for the acoustic control application. The plan is evaluated by heuristics. Partly these heuristics come from Nielsen’s heuristics that are the basis for many heuristic evaluations and partly form heuristics developed for this purpose of evaluating mobile applications. These heuristics are a bit more specific towards mobile control applications and can therefore be slightly better for evaluation of such an application. Next subchapter will introduce Nielsen’s heuristics and after that heuristics build for this specific thesis work will be introduced.

2.2 Nielsen’s heuristics

Jakob Nielsen has in the year 1995 defined ten principles for evaluating user interfaces and how well they work. They are called Nielsen’s heuristics. They are more like a rule of thumb rather then set in stone usability guidelines. The heuristics are listed below:

1. Visibility of System Status

2. Match between System and the Real World

3. User Control and Freedom

4. Consistency and Standards

5. Error Prevention

6. Recognition Rather Than Recall

7. Flexibility and Efficiency of Use

8. Flexibility and Minimalist Design

9. Help Users Recognize, Diagnose and Recover from Errors

10. Help and Documentation

From (Nielsen 1995).

First heuristic is visibility of system status. This means that the system should always keep the user informed about what is going on and it should give enough feedback at a reasonable time frame. The second heuristic is match between system and the real world. This means that system must speak the language of the user when it comes to the words, phrases and concepts given to the user. This also means following real world concepts and making the information appear natural and logical. Third heuristic is user control and freedom. Users often choose different functions in the application by mistake and there must be a clear exit to leave the unwanted state. The system needs to support undo and redo. Fourth heuristic is consistency and standards which means that the user should not have to wonder if different words, situations, or actions mean the same or different things in the system. There should be platform conventions to follow. (Nielsen 1995).

Fifth heuristic is error prevention. There needs to be clear error messages and elimination of errors that before they appear. This means that it is important to either eliminate the error conditions or check for errors and present the user of the software with a confirmation button where they have an option to choose whether to perform an action beforehand. Sixth heuristic is recognition rather than recall. This means that it is important to minimize the user's memory load in making objects, actions, and options visible. The user by themselves should not have to remember conditions and information from one part of the dialogue to another. Also, the instructions for a user must be visible and easy to retrieve whenever needed and appropriate. Seventh heuristic is about flexibility and efficiency of use. This means that there should be accelerators that are not seen by the user that can be used to speed up the system. The user interface should be tailored for both novice and experienced users. This means allowing the tailoring of frequent actions. (Nielsen 1995).

Eight heuristic is aesthetic and minimalist design. This means that the dialogues given to the user should not contain any information that is either irrelevant or rarely needed by the user. The more information there is in a dialogue it competes with relevant units of information and leads to diminishment of the relevant dialogue's relative visibility. Ninth heuristic is help user recognize, diagnose, and recover from errors. This means that error messages should be expressed in plain language instead of codes. They should indicate the problem and suggest solutions in constructive way. Tenth and final heuristic is help and documentation. It would be better for the system to work without documentation. This documentation and information in it should be easy to find and search and it should be focused on related user task. Documentation should also list concrete steps to be carried out. (Nielsen 1995).

2.3 Self-built heuristic requirements

Not only will Nielsen's heuristics be used for analysis of the artifact but there are some heuristics that are self-build but are based upon Nielsen's (1995) heuristics and build partly in a way that is described in literature in subchapter 2.1. They are meant to be more specific to the mobile platform. These will be used because Nielsen's (1995) heuristics are good but general. For the purpose of being more specific these heuristics will be used for artifact evaluation. They are:

1. Visibility of the status of the system to the user.
2. Learnability of the system and concise explanation of possible system usages.
3. Matching of application status and functions with real world physical world and interaction with it. Intended outcome is visible and intuitive.
4. Ability of the user to control physical devices with the application.
5. Preciseness of error messages and ability of user to understand error messages.
6. Efficiency of application usage.
7. Error recovery of the system. If no possible way of error recovery notifying the user precisely what went wrong.

These heuristics were built by going through the earlier combined theory of building on heuristics and a process of thinking about how the heuristics for a mobile control application should work and what should be thought about for this kind of heuristic evaluation. This building of heuristics was quite largely done by the point of view of the author and the thought process of the author. There are some heuristics here that are somewhat overlapping with Nielsen's heuristics such as number one and number three but these heuristics are meant to be specific for the mobile platform and as new heuristics they can overlap with the previous Nielsen's (1995) heuristics. The heuristics here were mostly done based upon figuring out what Nielsen's heuristic lack in terms of evaluation of mobile control applications and how these new heuristics could be used for further evaluation. Of course, this method of figuring out heuristics without using a very rigorous methodology is problematic but in the scope of this work this method was evaluated to be enough. However, there is a clear structure for these heuristics, and they are formally solid. For validation in the paper Yeratziotis & Pottas (2011) there were two phases of validation. These stages are validation with experts and validation by usage. Unfortunately, in the scope of this work validation was only done via the second stage which is the validation by using the heuristics and trying to figure out if the heuristics work in evaluation of the artifact.

The first heuristics is relatively simple and like first heuristic of Nielsen. However, this heuristic when using as a standalone is important due to what it reveals from the application. It is made for understanding how well the application shows what is going on in the application itself and the thing it is meant to control. The second heuristics was built because especially in mobile applications it is important that the application can be learned without too much difficulty and that the application is concise and easy to use. This kind of learnability and ease of use means that the user will want to use the application for the task it is meant in the long run. The application needs to be intuitive to use. If the application is difficult to use and learnability is poor the users will move to another application that does the same task. In a mobile application there is no space for long explanations or complexity.

The third heuristics is for understanding how well the application corresponds to what is happening in the real world. The physical outcome when using the application must be the same as intended by the user. For example, in a smart home this means that if the user raises temperature of the home in the application the temperature should rise in the smart home. The thought process of this heuristic is that this kind of application needs to efficiently control the real-world outcome and the outcome must strictly correspond to the user intentions when using the application. The fourth heuristic is about the ability to control the physical devices with the application. The application should give as much control that is required for the outcome and it could give more control to expert users. In the third heuristic the outcome of controlling should be the same in the application and on the physical world. However, this fourth heuristic is about the ability of controlling all the devices needed. In a smart home this means that the application can control all the devices that it needs to control to make the smart home environment work. There is a difference between outcome and the ability to influence that outcome for this heuristic.

The fifth heuristic was developed because it was thought to be important for users that there are error messages that are simple to understand. This is very important because users need to know what goes wrong in the physical devices or application connection to these devices. If users do not get error messages it is impossible for users to know if something is wrong and they cannot then fix these errors. The sixth heuristic is about how efficient the application usage is. This heuristic was thought of because in this kind of mobile control application there is not much space for making the application too complicated to use. This is because of the lack of space in the user interface and because of the nature of these kinds of applications. There is a need to make sure that the application can guide the user towards what it is meant to be used for without any side-tracks. The application needs to be efficient in working for the intended outcome. The seventh heuristic was built for the purpose of evaluating how error recovery works. The application should recover from errors by itself if possible and if not possible, the user should be able to work on error recovery in error conditions. According to the author, using these heuristics should be sufficient to find problems in the artifact when evaluating it.

When working with heuristics Nielsen Molic (1990) scale can be used to estimate how bad the usability problem is. The scale is:

Severity: How bad a usability problem is?

Frequency: How often does it occur?

Criticality: How critical are the problems of the application?

As previously described this scale is used to estimate the severity and impact of the problems with usability. First there is an estimation of severity of the error and then frequency meaning how often does it occur. These first two points are then combined to estimate the criticality of the error. However, because the artifact is a plan in this case the second issue of frequency is not applicable. This is because how often the error occurs cannot be more than one.

2.4 Examples of similar artifacts

In this part examples of related artifacts and their evaluation is presented. The artifacts are not the same as the artifact being built, but they are similar enough that something can be learned from these artifacts in relation to the artifact that will be build and evaluated.

First artifact coming from Mowad, Fathy & Hafez (2014) is a smart home control and monitor system that uses a mobile phone. This control system is using an embedded micro-web server that has IP connectivity in order to access devices and appliances remotely with an android smart phone device. There is no requirement of dedicated server for this application. The application that was build was based on RESTful based web services for flexibility and functionality. Also, there is a micro web server that was based on Arduino ethernet, hardware interface modules and smart phone application. In this system there can be changes made to the architecture in order to accommodate different scenarios in the application. One of the things done in the artifact was software for home gateway that was divided into two parts server application software and microcontroller firmware. Another issue was smartphone application and features that provided the functionality of remote connection to home gateway, device control, device monitoring and managing schedule. (Mowad et al 2014)

Another example is from Piyare (2013) a ubiquitous home control and monitoring system. This is also based on flexible home control and embedded micro-web server. The connections are based on an IP connectivity for all access and control of the devices and appliances. There is an android application used for this purpose. In this application there are no requirements of dedicated server PC for similar systems. It offers a novel communication protocol for monitoring and controlling the home environment. This article tries to demonstrate how effective this ubiquitous solution for the smart home control system is. (Piyare 2013)

The proposed system was intended to be standalone, flexible, and low-cost home controlling and monitoring system. It uses RESTful based web services which provides an interoperable application layer. The system is made of micro web server which is based on Arduino ethernet with hardware interface modules and android compatible smart phone application. The way it works is so that different devices and appliances to be controlled will be able to be controlled in a smooth and flexible fashion. Whenever there is addition into of a new appliance into the webserver there will be a new thread dedicated to that application in the smart phone application. One of the purposes is that there is no need for expensive hardware to implement the control application. The system allows owners of smart homes to monitor and control their home systems remotely via a Wi-Fi, 3g/4g enabled smart phone that has java support in it. There is a graphical user interface for the home control system. (Piyare 2013)

For this application to control a smart home remotely there are three layers. The first is the home environment. The second is the home gateway and third is remote environment. The third remote environment is for users to access their systems via a smart phone and who are also authorized to have this access. The first home environment consists of home gateway and hardware interface module. The home gateway is meant to work so that it provides data translation services between internets. The main component of this home gateway is a web server that is based on Arduino ethernet. The server then manages, controls and monitors system components. This then enables hardware interface modules to be successful in implementing hardware interface modules and execute assigned tasks using actuators and report the server triggered events that are done via sensors. In this system the hardware interface modules are interfaced with sensors and the actuators via wires. This means it has the capability to control energy management systems such as lightings, power plugs, heating, ventilation, and air conditioning. It also can control systems such as security systems like door locks and gates. In the home environment it also supports systems such as temperature, humidity and current. (Piyare 2013)

There is another example from Inal (2018) where an application interface for a large mobile doctor appointment application in Turkey was evaluated. This evaluation was based on Nielsen’s heuristics and had 40 information system engineers take part in it. Also, there was a system usability scale used for a detailed analysis of the data. There were problems related to usability according to the study. Some of the major ones were error prevention and user control and freedom. The application did well on the heuristics of consistency and standards. According to the evaluation these problems were identified either as minor or major usability problems. There is a lot of effort in Turkey that is put on mobile systems for government provided services. Therefore, it is important for Turkey to get information about how usable they are for a specific area in which they are used. In this work the part about the system usability scale is ignored because it does not directly relate to the topic of this thesis work. (Inal 2018.)

Mobile systems are according to the above-mentioned study Inal (2018) very important technologies today in many areas from payments to commerce and government systems. Many governments want to use mobile applications for interaction with their citizens for ease of use and they want to provide a better service and quality information for their citizens. There are a lot of advantages in providing services through mobile platforms such as flexibility of accessing services and reaching a large portion of the country’s population. One of the applications of mobile services is the usage of them in health care.

There have been a lot of health care applications but one of the issues in them has been usability. Mobile devices have a lot of issues with things like screen size, limited capacity, bandwidth, connection, interface, and context of use. This means that usability testing in mobile applications is very important. In this study aim was to provide a usability examination of the largest e-government application in Turkey which is making medical appointments to public hospitals trough the centralized doctor appointment system (CDAS). The heuristics were used by system engineers to evaluate the usability of the application. The study is aimed to be a reference for future building of government run mobile applications. (Inal 2018.)

Research questions of Inal (2018) were:

1: What are the usability problems identified in the heuristic evaluation?

2: What is the severity rate of violations according to Nielsen's heuristics?

3: What is the usability score of the mobile health application according to the system usability scale (SUS) evaluation?

These research questions will be used as one reference for the research questions formed in this thesis work. However, the third research question is not applicable for this work since the system usability scale will not be used in this work at all for evaluation of the artifact. Therefore, that question is going to be ignored.

The application in Inal (2018) had similar problems to other mobile applications. There were problems with the design of different applications for monitoring health records. For this kind of application strong error prevention is very important. In these mobile health applications, the basic needs of the patients should be simple do deal with while having options for advanced users. Inal (2018) found that there is also need for more customization in the application used in Turkey. For this kind of application also different kinds of users must be considered. This application was found to be easy to use by different kinds of users though. There was however feedback from older users to streamline the application and make it even simpler to use. Small amount of difficulty of usage for this kind of application can lead to significant loss of productivity. Therefore, it was found that there is a need for this application to increase efficiency and performance for the users. From the point of view of heuristics evaluation of this Turkish healthcare application most critical failures were in heuristics of "help and documentation", "help users recognize, diagnose and recover from errors" and "error prevention". However, this application was still considered easy to use by the users based upon their feedback. There was however recognized that help features were important for the users and needed improvement. In this study the recognized problems were considered helpful for improving the effectiveness, efficiency, and satisfaction of the users towards the application. (Inal 2018)

This study was a good example of how to use heuristics for evaluation of a mobile application. There was a lot of good information here and especially the issue that was found in help features and error recovery was a good point later evaluation of the artifact that is built in this work. Similar problems can be found later in the artifact of this work.

3. Research Methods

In this chapter there is a definition and description of the research problem and research questions. These questions are solved by building the artefact which is a plan for the mobile control application for acoustic panels. In the Chapter 4 of this thesis the application is designed. The research questions here are going to be discussed in Chapter 5. Another issue discussed in this chapter are the research guidelines. The research is done based upon design science theory that is explained further here. This theory is used for building the plan and its iterations.

The evaluation is done by the author himself and the iteration based upon this for the design science design cycle will be done once after the heuristic evaluation is done. The cycles based upon this chapter will be used for the iteration.

3.1 Research problem and questions

The problem statement for this research is how to make a mobile control application that is easily usable by non-expert user for controlling acoustic panels. This problem is answered by a plan for the application that includes requirements analysis, system analysis, functional analysis, and design synthesis. The plan is then evaluated using Nielsen's (1995) heuristics and heuristics specifically made for mobile control application that were listed in the previous chapter. These heuristics are made so that it considers the requirements of the client and theory discussed for this work. We approach the research problem by trying to answer following research questions:

RQ1: How to create an efficient and usable mobile application for controlling acoustic panels?

RQ2: Based upon heuristics, what are the usability problems of this application?

RQ3: How, based upon heuristics, to improve the application operation for non-expert users?

With these research questions the idea is to plan the application in such a way that when built it would be usable and efficient. The usability is evaluated by looking at the plan through the eyes of heuristic evaluator. The evaluator should be able to determine how critical the problems are in the plan using the Nielsen, Molic (1990) scale. When the usability problems are identified they will be discussed in the discussion part.

The aim of these research questions is to find as much as possible about the usability of the application and to make sure that it can be used later for the purpose of controlling acoustic panels without problems and with ease. The usability of the application is important also for the fact that the users are not themselves necessarily ICT experts but working on the music field and they must be able to use the application with ease. One of the issues for the application is the number of panels that needs to be controlled and the amount positions each panel can have. To represent panels in the artefact there is a need to concise but also have usability on a level that individual panels can be easily used when operating the artefact. If there are large amounts of panels shown in the user interface, it can be hard for the user to operate singular panel at the time and there might be a need to separate panels on multiple views in order to make this work.

3.2 Research guidelines

Evaluation of the artifact is one of the most important parts of the design science research. There are good guides and papers written about how to do this evaluation. In this research the guidelines come from multiple different papers. The basic papers are Hevner, Ram & March (2004) and Hevner (2007). The rest of the papers used here are additions or comments on this paper. The other papers are working around Hevner et al (2004) and adding to the knowledge base of that paper.

It is important to understand and evaluate the artifact to understand how well it solves the research problem. Also, it is important to learn new information from the artifact. There must be a research contribution given by the artifact. The artifact needs to incorporate previous knowledge and create new knowledge for the research field. The artifact build here will solve the problem of how to build an application for controlling acoustic panels via mobile environments. There has been previous mobile application for controlling different physical environments and these have been looked at in the previous chapter. These previous works give important knowledge for this application.

According to ISR (2002) point of information systems research is to create knowledge that furthers productive application of information technology in organizations. Also, according to Zmud (1997) information systems research is done to develop and communicate knowledge in management of information technology and usage of information technology in managerial and organizational purposes. The paper of Hevner et al (2004) talks about how to conduct, evaluate, and present a research for design science research in this field. According to March and Smith (1995) there are two research paths in information systems research, which are behavioral science and design science. Design science according to the paper is about creating artifacts and extending boundaries of human and organizational abilities. The paper of Peffers, Tuunanen, Rothenberger & Chatterjee (2008) observes that design science has been slowly adopted into information systems research but so far the methodology has not been developed well enough.

According to Hevner et al (2004) some of the most important things in design science are understanding, executing, and evaluating research. There needs to be knowledge and understanding of the problem itself and how its solution works. According Hevner et al (2004) design science is both a process and a product. Design science should produce an innovative new product that solves the problem efficiently and easily. The evaluation process itself is there for providing feedback and understanding of the problem. Hevner et al (2004) talks about how artifacts can be many different kinds such as constructs, models, methods, and instantiations. These kinds of artifacts are built and evaluated in order to solve problems. On the other hand, according to Hevner & Chatterjee (2010) design science is about solving unsolved problems in a new and unique way. The serve the purpose the solutions are meant for they should be effective and efficient.

Problems in design science according to Hevner et al (2004) are: “unstable requirements and constraints based upon ill-defined environment contexts, complex interactions among subcomponents of the problem and solution, inherent flexibility to change design processes as well as design artifacts, critical dependence upon human cognitive abilities to product effective solutions and a critical dependence upon human social abilities to produce effective solutions.” The problems must be considered when developing the artifact. The requirements for the application must be known and stable and the environmental context needs to be taken into consideration. There is a need to know and plan along the lines of how the interactions of subcomponents to the problem work. This is in order to know how to build the application in the best possible manner. The requirements should be defined so that there is no need to change them during the development process. Also, the human cognitive abilities must be considered when working on the problem in the artifact and the failure of these abilities should be considered. These kinds of problems mean that the design science research is constantly changing and in a state of revolution as Hevner et al (2004) puts it.

3.2.1 Guidelines for artifact development

Next, I will paraphrase from Hevner et al (2004) guidelines for artifact development. First guideline is about designing the artifact to be used. According to Orlikowski and Lacono (2001) artifact is the core subject matter in the information systems research field. This first guideline says that in design science research the artifact should in form of construct, model, method or instantiation but it is rare in design science that the artifact is going to be a full-grown information system that will be used in practice according to Hevner et al (2004). Instead of this according to two studies of Denning (1997) and Tsichritzis (1998) artifacts are innovations that define ideas, practices, technical capabilities and product through a process of analysis, design, implementation and use is information systems. Constructs have a significant impact on the way in which tasks and problems are perceived according to Boland (2002). Models and representations can be done for problem domains. The artifact developed in this work is a plan for mobile control application. The plan can be used to implement an application that can be used for acoustic panel control. The artifact being a plan is not a finished application that could be tested on the panels themselves but a plan for how to build such an application. One important thing of the plan is to evaluate its usability so that when the application is built it will be easily usable by a non-technical user.

According to Hevner et al (2004) the second issue is problem relevance. The problem for which the research is done should be important and relevant business problem. Design science approaches this goal by building innovative artifacts that are meant to change the phenomenon when it occurs. Venkatesh (2000) challenges researchers to design artefact that can overcome acceptance problems. According to Hevner et al (2004) a combination of technology-based artefact, organization-based artefacts and people-based artefacts is important for addressing problem relevance issues. Problem is about difference between the goal of the system and current state of the system. According to Simon (1996) problem solving can be defined as a process of actions to reduce and eliminate these differences. According to Hevner et al (2004) problem relevance is about working with planning, managing, designing, implementing, operating, and evaluating technologies that will further develop knowledge in the field of information systems research. The problem in this work is relevant due to the nature of the artefact. There are mobile control applications build before, but they do not directly deal with acoustic panel control. The problem of this artefact is how to make this kind of application plan that is workable in the future when the application is built.

The third guideline from Hevner et al (2004) is about evaluation. This means evaluation of utility, quality, and efficacy. These things should be evaluated and demonstrated in a rigorous way and for this rigorous evaluation methods must be used. For the evaluation, the business environment establishes the requirements for the evaluation. Evaluation needs appropriate metrics and gathering and analysis of data according to Hevner et al (2004). The evaluation can be done in terms of functionality, completeness, reliability, usability, fit with organizations and other possible metrics. For design phase evaluation is important because it can provide feedback for design and make next iteration of the application better. The methods of evaluation must be according to Hevner et al (2004) such that they match with designed artifact and selected metrics. Design by nature is innovative and human perceptions change it therefore it is important within constraints and challenges to vary design process. Evaluation for this work will be done based on Nielsen's heuristics and set of heuristics built in the Chapter 2 for this specific purpose. Evaluation based upon these heuristics will give information on utility, quality and efficacy. These heuristics evaluation will be used as metrics for this plan.

The fourth Hevner et al (2004) guideline is about research contributions. The artifact that is build should provide some clear contributions to design science research from the artifact being build, its design foundations and methods. There are three types of research contributions that are based on novelty, generality and significance of the artifact. One of the contributions can be the artifact itself. The artifact can either extend the knowledge base or use existing knowledge in new and innovative way. Second important contribution can be according to Hevner et al (2004) foundations. There can be constructs, models, methods or instantiations created that extend or improve design science knowledge base. Third way of research contributions in design science can come from evaluation methods and new evaluation metrics. These research contributions can be assessed based on representational fidelity and implementation ability according to Hevner et al (2004). Research contribution of this work is the artifact itself. It is not an artifact that extends the knowledge base a lot, but it is using existing knowledge in a new and innovative way. There are also self-built heuristics that are used to evaluate the artifact. These heuristics make one contribution to the knowledge base even though they are not included in the artifact itself.

Fifth guideline Hevner et al (2004) is research rigor. Rigor in research is about how the research is done or conducted. In design science there needs to be rigorous methods for both construction and evaluation of the artifact. There is a problem with rigor in that when an artefact is evaluated with too much mathematical formalism the environment in which the artefact should be used may defy this excessive formalism. According to Hevner et al (2004) rigor must be assessed in terms of applicability and generalizability. Rigor might diminish relevance. According to Applegate (1999) it should be possible and necessary for information systems artefacts to be both rigorous and relevant. Rigor is defined by effective use of knowledge base, theoretical foundations, and research methodologies. Research rigor will be considered when working with the artefact. However, in this kind of work too much rigor can lead to results that are not applicable for the actual user of the artefact. The reason why the artefact is evaluated both based on self-build heuristics and Nielsen's heuristics is that Nielsen's heuristics are a standard in the field and using them will add rigor to the research. For construction, the rigor comes from following standard procedures in building the plan which is the artefact.

According to Hevner et al (2004) sixth guideline is design as a search process. Design science is completely iterative. It can be hard to track the process of searching for best or optimal design solution for realistic information systems. Heuristic strategies can produce good and feasible artifact designs. In design science it is common to simplify the problem by introducing the relevant means, ends and laws. Means are decision variables that constitute implementable design solutions. Ends are represented in utility function and constraints. Laws are being represented by values of constants in utility functions and constraints. The problem can also be broken into parts. According to Hevner et al (2004) progress is then made iteratively as the problem scope is expanded. When these means, ends and laws are made more realistic the artifact becomes more relevant and valuable. However according to Vessey & Glass (1998) due to the nature of information system design problems it might not be possible to determine or describe explicitly these means, ends or laws. For satisfactory solutions heuristic strategies can be used for creation, utilization, and assessment. Using heuristics however does open the question of how good solution is defined in information systems. According to Hevner et al (2004) one approach when using heuristics is to show that the solution is close to optimal solution. Another way is to compare the solution to expert designers' solutions for similar problems. For this issue of search process heuristics are used to evaluate the plan because it is hard to say exactly what are the means, ends and laws for this project. These can be defined partly during the process and partly during the end but before beginning building the artifact it can be hard to realize what are the means, ends and laws.

Seventh guideline in Hevner et al (2004) is communication of research. There is a need to communicate design science research results to both technical and managerial audiences. For technological audience there needs to be enough detail for them to implement the artifact in an organizational context. This then enables researchers to get benefits from the cumulative knowledge built while practitioners can use the product for their own means. This also means that results can be repeatable. According to Hevner et al (2004) managerial audiences need enough detail to decide if the organizational resources should be used for constructing and using the artifact. According to Zmud (1997) for managerial audience the presentation of the artifact should not be about the inherent nature of the artifact but on knowledge required to use the artifact for organizational or individual gain. Therefore, the emphasis should be on the problem relevance and the effectiveness of the solution to the problem. According to Hevner et al (2004) it also can be necessary to describe the artifact in some technical detail for managers to understand the artifact and appreciate it. This kind of description can be done concise and organized appendices. The artifact in this work will be communicated through this thesis work. Therefore, the artifact is made so that it is understandable for both non-technical and technical audiences. The basics of everything will be made simple enough but at the same time there are some technical details. Also, this work is detailed enough to make business decisions based upon it. There will be technical detail and explanation of this detail by words.

3.2.2 Design Science cycles

The paper of Hevner (2007) analyses cycles of activities that are important design science research. This is done based upon Iivari (2007). There are three cycles for design science. First is relevance cycle which inputs requirements and introduces the research artefact into field testing. Second is rigor cycle that introduces grounding theories and methods. Rigor cycle also introduces experience and expertise that comes from the foundation knowledge base. Rigor cycle also adds the new knowledge created back into the knowledge base. Third cycle is design cycle that supports research activity. In this cycle construction and evaluation of artefact and processes related to it are proposed. These research cycles help to process the artefact and situate it in time in a relevant way. Figure 1 shows the cycles.

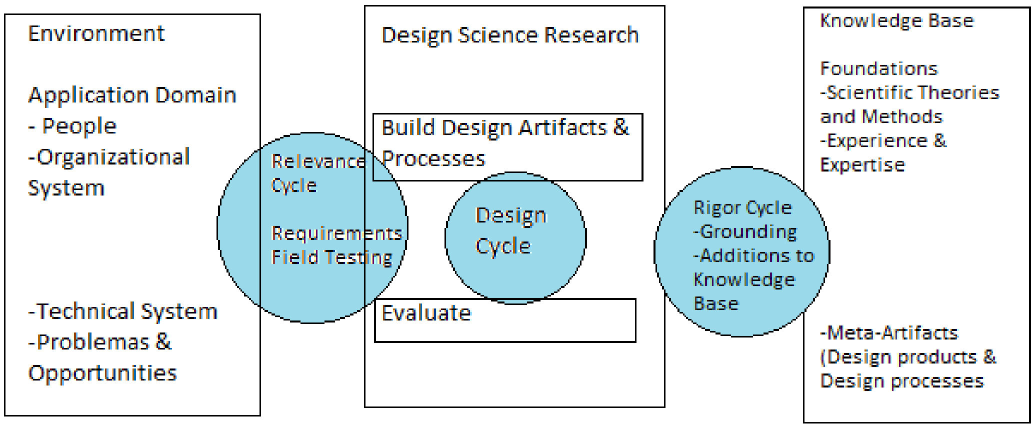

Figure 1. Design Science Research Cycle. From Hevner (2007)

In design science research there needs to be clear definitions, ontologies, boundaries, guidelines, and deliverables for the design itself and for the execution of said design according to Hevner (2007). To be useful design science research must be about identifying and representing opportunities and problems in an actual environment in which the application is to be used. There is also a possibility that the research is about potentiality which means that there is potential area for building an artifact that has not been previously explored but can be examined for a new artifact. Design science itself has to do with improving the environment by building new and innovative artifacts and developing new processes for the purpose of building these artifacts as in Simon (1996). Also, according to Hevner & Chatterjee (2010) design science research contribution is not only about the application but also its environment which works as a kind of acceptance criterion. Hevner (2007) also says that there are two ways to do. One is about identifying new opportunities for artifacts and working on them. The other is about using previous knowledge and making the artifact better for working based upon previous research with more rigor and practice.

According to Cole, Purao, Rossi & Sein (2005) and Jarvinen (2007) relevance cycle is the one that initiates design science research and provides requirements and defines an acceptance criterion for the research. The acceptance criterion is used for evaluation of the research in the end. In the end the output of the research is returned to the field for study and evaluation. According to Hevner (2007) the results of this field testing are then used to determine if research project needs additional iterations for the artifact. The new artifact can have problems with functionality or other qualities such as performance or usability. Also, the original requirements can be lacking. According to Hevner (2007) feedback for the next iteration can lead to restatement of the research requirements.

The rigor cycle according to Hevner (2007) draws from knowledge base which has scientific theories and methods that are the foundations of design science rigor. There is also additional knowledge which are experiences and expertise defining application domains and existing artifacts and process from the application domain. For this cycle it is important according to Hevner & Chatterjee (2010) to note that design science becomes research when it increases knowledge in the knowledge base. Further artifact building also requires that the knowledge base be broadened with further research. According to Hevner (2007) this cycle uses past knowledge to ensure innovation in the design science research. Therefore, it is important for researchers to know and research the knowledge base to guarantee that the research they are doing provides research contributions. It is this rigor that makes design science different from just building IT systems. This means that researchers must be skilled in selection and application of right theories and methods when constructing and evaluating artifacts. Additions for the knowledge base based upon this design science research according to Hevner (2007) include theories and methods, meta-artifacts and experiences gained during the performance of the research and field testing. According to Hevner & Chatterjee (2010) another contribution is additions and extensions to original theories and methods that are in the problem domain. Also, it is important to advance knowledge instead of just rehash old solutions.

Last cycle in Hevner (2007) is design cycle. This design cycle is the heart of a design science research project. This cycle operates between construction of artifact, evaluation of the artifact and feedback with further iterations of the artifact. According to Simon (1996) this cycle works with generating design alternatives and evaluation of those alternatives to requirements all the time until satisfactory design is achieved. This cycle draws from both other cycles. The requirements are input from relevance cycle and evaluation of theories and methods come from the rigor cycle. According to Hevner (2007) in the design cycle all the hard work of design science is done. There is a need to understand dependencies of the design cycle on the other cycles while at the same time making the actual execution independent research. When working on the design cycle it becomes important to maintain a balance between both constructing and evaluating the artifact. Both activities must be based in both relevance and rigor. For example, if there is a strong argument for the construction of the artifact but the evaluation is weak the research is not done properly. Hevner (2007) says that artifacts must be rigorously and completely tested in laboratory and experimental situations before they are ready for field testing. This means that there needs to be multiple iterations of the design cycle before releasing the artifact to relevance and rigor cycle. According to Hevner & Chatterjee (2010) most of the hard work in design science is done in this cycle and it is important to evaluate and get feedback from the artifact building but there also must be a balance in this cycle.

According to Hevner (2007) design science research is pragmatic in nature due to its emphasis on relevance. It makes a clear contribution to the application environment. It can be noted that practical utility is not what defines good design science research. In design science research there is a synergy with relevance and rigor cycles and contributions to these cycles. This synergy is what defines good design science research. Applicability of the research in building a practical artifact does not alone make good design science research according to Hevner (2007)

3.2.3 Objectives of design science research

Peffers et al (2008) develops that kind of framework and methodology by talking about principles, practices and procedures that could adopted to design science research. The paper talks about three different objectives that it meets. These things are consistency with prior literature, providing a nominal process for design science research and providing a mental model for the presentation and evaluation of design science research. There are also many different types of design science processes that have been suggested by different authors. There are methodologies that are clearly designed to be methodologies used by developers and not by researchers such as Coopers (1990). The paper of Peffers et al (2008) on the other hand wants to develop the research methodology more. There are also design methods that are developed to address the problems of requirements engineering for different situations in a human-centred design cycle that point to the information systems field from a functional viewpoint such as the work by Hickey & Davis (2004). There is also the work of Iivari, Hirschheim & Klein (1998) that considered the differences between the information systems development methodologies and different kinds of methods and the needs that come up from method development in this kind of issues. There is problem however with all these methods described in that they are all for specific situations instead of a more general approach that can be used on all the information systems research. The paper of Peffers et al (2008) is designed as a more general approach on development to get to a point where a design science process for general usage in information systems field is developed.

There are in design science research a lot of ideas about how to do research in the field according to Peffers et al (2008). This research is done in a way that the process models cannot really be used in a broader context but could be used only on specific contexts. However, these research methods and methodologies and processes can be used as a guideline for developing a more general approach to design science research in information systems research. There is a framework, or an abstract model developed by Nunamaker, Chen & Purding (1991) that connects different aspects of design science research, but this paper also leaves the processes and methodologies for conducting such a research for researchers' inference. Other theories such as Wall, Widmeyer & Sawy (1992) develop a high-level abstraction for researchers from which researchers can infer a more specific process. Others such as Hevner et al (2004) and March & Smith (1995) have published guidelines for design science research influence methodologies to be chosen by information systems researchers for their design science research. There is an issue however with complete methodologies for design science research that have not been developed yet. Other such as Archer (1984) have provided ways to get design science research incorporated with a more consensus process. Also, Adam & Courtney (2004) have proposed extensions to the model that were talked about before to make it more complete. This extension was about inclusion of action research and grounded theory to the design science research focus on information systems research. Cole et al 2005 also proposed a way to integrate action research and design science research. However, all these do not yet have a complete general research methodology, processes, or methods for all kinds of research in the field. There must be more general approach to the research done and that was proposed in Peffers et al 2008.

There are six steps identified in Peffers et al (2008). First is problem identification and motivation. For the problem to be identified and motivation to be derived there needs to be a clear research problem and the value of the solution needs also to be clear in order to find the value of the research problem for design science research. A good way to work with the solution is to divide it into parts to better understand the problem. In this way complexity of the problem and its solution can be more easily dealt with. Second step according to Peffers et al (2008) is to define what are the objectives of the solution. This should be done with the problem definition and previous knowledge when it can be used. The third step according to Peffers et al (2008) is about design and development. In this phase the artifact is created. The artifacts functionality that is desired or architecture must be known when creating the artifact.

The fourth step according to Peffers et al (2008) is about demonstration of the artifact in that it should solve the problem it was meant to solve. The demonstration should show that the artifact solves one or more of the problems it was supposed to solve. The fifth step is evaluating the artifact that was built with design science research methods. What should be done is observe and measure how well the artifact that was built solves the problem for which it was built for. This can be done by comparing objectives for the artifact building to the solution of the actual artifact. Important things to know are artifacts performance in relation to artifact objectives, performance measures, satisfaction surveys, client feedback, simulations, and system performance. In Peffers et al (2008) the sixth step is communication which is important for design science research. There is a need to communicate how the artifacts solves problems, the importance of the solution, the artifact that was built and what it does, what is the artifact utility and novelty. Also, the rigor of the artifacts design should be communicated and its effectiveness for researchers and other issues that are relevant from the point of view of the design science research.

This work follows more Hevner (2004) guidelines then these steps but they are also an important reference point for this thesis work. They are considered as an extra guideline of how to work with design science research.

A paper from Gregor & Hevner (2010) provides further instructions on how to do design science research. According to it there are three different design science research issues. First is the nature of the artifacts/problems that are studied in ICT disciplines. Second one is the nature of the research's approaches used and the third is the nature of research contribution made. These issues are all independent from each other and they can improve the understanding of the goals and processes in design science research field and knowledge base. (Gregor & Hevner 2010).

According to Gregor & Hevner (2010) it is also important to understand the nature of the artifacts, problems, and objectives. There is need for treating different artifacts as special and infer from that point of view the contribution that they make. Also, it is important to note that abstract results can be treated as an artifact type. These artifacts can all take a different meaning based upon their type and it is important to note that the goals in design science research can be varied. Artifacts should be considered in a more expansive way. There can be two kinds of methodologies which are hard methodologies and flexible methodologies. For this work a more flexible methodology is used. This is because the work is a socio-technical system. (Gregor & Hevner 2010).

4. Constructing the Artifact

The process of developing the artefact follows processes of design science research as explained in Chapter 3. The idea is to first establish the relevance of the software with requirements and understanding of the application domain. First there needs to be an understanding of the application domain by understanding people that use the application, organizational style, technical systems and problems and opportunities. After this the design of the artefact is done by introducing the artifact and building it. Rigor cycle is last for grounding the application in theory and scientific practice.

The artefact in this case is the application plan that is built for acoustic panel control. This plan is then evaluated based upon heuristics mentioned in Chapter 2. The design science approach is used for this artefact building. As based upon Hevner et al (2004) guidelines and elaborations for it the artefact is built after which it is evaluated, and research contributions are explained. The artifact will be evaluated for utility, quality and efficacy. Also, research rigor must be explained. After evaluation all of this is discussed in Chapter 6.

In Hevner (2010) there are questions which need to be answered for a good design science project. The first question is about research problem and this was answered before in Chapter 3. Next question is what artefact is and how is it represented. For this question the answer is the artefact is the plan for the acoustic panel control application. The plan is represented through analysis of the application and how it functions and going through requirements for the application.

Next question in Hevner (2010) is the design process that is used to build the artefact. In this case this process includes plan in basic text form and diagrams to explain how the application is supposed to function once built. However, the design process does not include actual programming or making a functional application. The process is about working on a throughout plan as an artefact. Another issue is evaluation. This is done based upon Nielsen's (1995) heuristics and self-built heuristics. The artifact will be evaluated for quality and usability by these heuristics in Chapter 5.

First in this artefact building process there is an application overview. The application overview is based upon requirements of the client. After that there is are specifications for the application. Then functionality of the application is discussed via use cases and functional analysis and system analysis is done. The design will then be concluded with a design synthesis chapter. The iteration of the plan is done after Chapter 5 heuristic evaluation has been done in Chapter 5.

4.1 Overview of the application

For building the artifact there is a need to understand how mobile development works and how is it different from other fields. Basic planning of the artifact is done through analyzing the requirements for the artifact. The plan itself for the software is the artifact and the requirements analysis are part of building the artifact. Other important parts are system analysis, functional analysis and design synthesis. Ones the plan is done it is evaluated based upon Nielsen's heuristics and self-built heuristics that were introduced in Chapter 2.

For the planning of the application there was a basic user interface model given by the client. Part of the problem with the interface is about scalability and how well scalability can be worked with for the plan of the application. This was because the application was meant for tablets and smart phones that do not have a very big screen. This means that some scalable parts of the interface might become too small for the user to efficiently use.

The issue of scalability comes up when there are a lot of panels to work with. For example, the issue can be that there are too many panels shown in the user interface at the same time and they can become unusable because of how many panels there actually are to control in the application. The original plan for the interface that was suggested by the client might be unusable when there actually are a lot of panels to use. Figure 2 shows how the user interface should work.

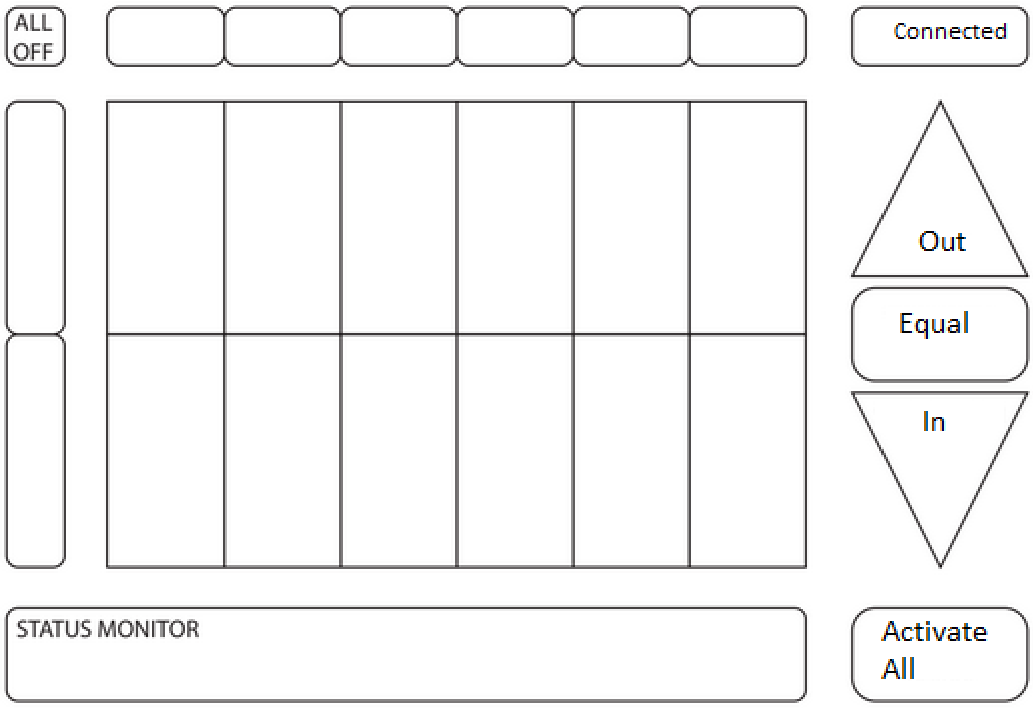

Figure 2. The User Interface.

As Figure 2 shows there some buttons that are usable all the time and these are not hard to work with. The problem with this user interface can be the requirement of scalable panel vision which takes most of the space here. If there are a lot of panels this interface can become unusable because of how small the panels shown are to click especially when this interface is used on the phone or on tablet where the screen is not that big to begin with. It can be that there is a need for planning the user interface again to accommodate this concern but in the beginning the interface will be done with the original user interface requirements in mind. There is also an original requirement of an emergence stop button that is not shown here. The user interface however in this way is already quite jammed so figuring out the emergency stop button might be difficult. However, it should be workable so most problem with this interface is the scalability problem and how to design the interface with scalability and good usability at the same time.

The way to use the actual system of panels response to commands through the user interface is done via Wi-Fi. This requirement can change though and there can be additional requirements for it if necessary, depending on how the panels will be build when they are ready. In the structure of the program there is information that the panels give. There can be multiple different positions of the panels all the way up to seven positions. The panels will send data to the application about their current positions in the panel wall and the number of panels there are and how they are configured.

The plan for the application must consider what is discussed above and work through those requirements. This means that the plan build works with actual requirements for the application that were given. The plan must take all these things into account. Also, the plan must solve the problem of multiple panels that become too numerous to show on the screen at the same time.

In the application development after the plan is done the first thing was to design the basic interface as according to what is above mentioned. The next step is to plan the functionality of the application. Whatever issues there will be in constructing the application will be discusses. There is a need to make sure that the application fulfils the requirements it has, and it has the scalability to be used. One of the requirements was that it should be able to show all the acoustic panels on the initial user interface, but this might be completely in practical depending on the number of panels there actually are. With large amounts of panels, it can become hard or almost impossible to select an individual panel or be able to control one panel in an easy fashion. This is a problem that needs some thought put into it because of the initial requirements but also because of the way this should work in moving the panels from one position to another.

4.2 Business requirements

The background for this product is that music hall wants to build an acoustic wall that moves depending on the acoustic requirements. The technical issue with this wall is that it should be easily movable with a mobile application. This application communicates with the wall via Wi-Fi. There are similar applications that already exists for controlling smart houses and doctor appointments as an example. The new issue of this application is not as much about how it works but about what it is used for which is the acoustic panels. This application can then be used in multiple locations for similar panel control when these kinds of movable acoustic panels become popular. The reason they can become popular is because they offer a novel solution to use in controlling acoustic panels. There can be panels that would be movable by hand, but these would be uncomfortable to set in the right place due to the number of panels there can be.

For business objectives the idea of building more of these kinds of panels in different locations can be profitable if they work right. The application is part of the profitability of the acoustic environment, but it is not the most important part. It is important because it allows for easy control of the acoustic panels but in a way, it is only a small part of this acoustic environment. For the project to be successful the panel control application must of course work in controlling the panels. The application must be able to represent the panels in such a way that any person without much prior knowledge can use it to control the panels. The scalability problem is there for the application and it should be worked around. Success is defined as an ability of a relatively non-technical person to use the application in such a way that it works for intended purpose. Another success metric is that the application can work efficiently with panels in moving them without problems to the right positions and show panel positions to the user in an easy to understand way. Also, the application must be technically well done so that it does not have issues with connections and such things.

The purpose of this software is not to have long term commercial application but satisfy the need of a single client. The application can be used by others, but it is not a complicated software that could be used by itself to commercial usage. The application is very much about solving a problem of a single client in a special environment. Therefore, long-term commercial usage for the software is not applicable. The target customer is the user who uses the software in a single location for controlling specific acoustic panels. There are no diverse stakeholders for this application. This application is both similar and unlike for example smart home control applications. Smart home control application can be used for multiple usages and customized for different environment. The environment for this application is specific and therefore usage in other environments is not an option.

Risks for this development are things like acoustic walls not being ready for usage when the software is ready or the application having problems with scalability. Changes in the panel composition can be a risk also due to changes that can occur in the panel environment. However, these changes should not be too big since the environment is very specific. Another risk could potentially be that there can be a situation where the application should diversify to control other aspects of music hall such as lighting or sound mixing. However, sound mixing as an example in a big hall can be so complicated that a simple mobile control application cannot do it. At least from personal experience sound mixing even in a smaller environment is complicated depending on how many musicians there are. Assumptions for this work are that there are no changes in the environment and no additional requirements for the control applications. The project is depended on the environment not having big changes and in that the contractor will get their side of the acoustic panels working on time for the application to be used.

4.3 Requirements analysis

In this section requirements for the software will be discussed and analyzed as a part of planning for the application. There are multiple requirements that this application for controlling the acoustic panels must fulfil and these requirements will be analyzed one by one in this part.

Requirements:

1. Mobile application for tablet and smart phone
2. Cross platform application
3. Panels easily visible and usable
4. All off button: Turns all panels off
5. Seven positions inwardly and outwardly plus center position
6. Status monitor on user interface
7. Possible to connect and disconnect to panels
8. Individual and multiple panels can be selected

9. Button that will activate all panels at the same time
10. Three buttons that move panels inward, outward and equal
11. Wi-Fi connection from application to panels

4.3.1 Requirement specification

The first requirement means that there needs to be a possibility for the application to scale for both small and bigger space. This is doable but scaling must be considered. The user interface for this application can have a problem with it being too big to be efficiently used in a mobile smart phone environment. This can mean that the application functionality must be some scaled down for mobile phones or it can even mean that the panel views of the application must be increased in number for a mobile phone environment. There can possibly be multiple views from which the application works in order to include all the functionality that the application needs. The introduced user interface can be fine for tablets but too big for mobile phones. This is the biggest problem for usage in both mobile phone and tablet environments. One of the ways to make it usable on mobile phone environment is that the status monitor can be skipped, and status shown in individual panels. Also, some not so often used functionality like connection, disconnection, all off and all on can be put inside a context menu so that the application will not be too cluttered.

The number two requirement is about the platform that will be used for this application. The customer wants the language to be either C++, C# or Java. The cross-platform development can introduce constraints for the application and some cross-platform tools must be used. However, for this work there is no need to determine which of these tools should be used. Therefore, this aspect of the plan is left for the implementation phase to decide.

The third requirement is about the visibility of the panels in the application. Depending on the number of panels this can be challenging since it is possible that even for a tablet this cannot be made to work due to too many panels. One issue is also with the mobile phone. The challenges this causes on the tablet environment are also critical. There can be so many panels that need to be worked with that the application cannot show all of them in a single screen. This can mean that the panels have to be cut into segments for the user interface to work and it can also mean that certain actions such as centering the panels easily with one button push to the whole physical panel environment cannot be done because all the panels cannot be selected at the same time.

The fourth requirement is about the all off button in the user interface. This button is meant simply to turn off all the panels and return them to the zero position which means centered. This button sends a signal through Wi-Fi to the first panel and the first panels microchip takes care of the rest. Even with a user interface that is divided into segments this button can be used for all panels in all segments without much of a problem.

The fifth requirement is that there is multiple position, in this case seven, in which the panels can be. This just means that the panels are moved upon these positions based upon how the user interacts with the user interface. These panels positions must be clearly shown in the user interface perhaps in the panel buttons themselves. The panels are moved inward or outward based upon the user interaction with the user interface.

Sixth requirement is that there is a status monitor for the panels in the user interface. This requirement was unclear about whether it means that there is status monitor that monitors the panels position status or simply the connected or not connected status of the application. In the first case the monitor can have way too much information and this kind of monitoring is not practical and could be done in the panels themselves. The second way is doable of course and does not require a lot of effort. It just means that either the panels are connected or not and this is indicated with the status monitor. However, this might be unnecessary since this kind of status could be shown in the connected button and this means that the status monitor should be done, if possible, in such a way that it shows information not shown in the panel view.

The seventh requirement is straight forward since it is only about the ability to connect or disconnect all the panels. When the panels are disconnected the status of connected button will show red and when they are connected green. There is no requirement of any position the panels would move when disconnected which is of course is the way it should be since if the panels would move every time disconnected that could cause a lot of problems for example if the Wi-Fi connection goes down for a reason not connected to the user. Also, it would cause problems if the application is for some reason closed during a concert. The panels therefore must stay where they are and wait for the application to reconnect. The button for connection can be green when connected and red when disconnected. This way it is easy to recognize whether the panels are connected to the application.

Requirement number eight is also straight forward. It basically means that panels can be selected in groups and individually. There should be buttons that will allow selecting or deselecting either vertically or horizontally certain group of panels. The panels should also be able to be selected or deselected individually. This requirement is not hard to fulfil except when there are multiple views of panels that can be selected. Of course, then the selection will happen in only the panels that are visible in the view.

The ninth requirement is to have a button that will select all the panels. There is a bit of problem here with how many panels should be selected since all can mean all the physical wall with physical panels or all the panels that are visible in the user interface. There needs to be thought put here about which way this would be. Either the whole physical wall is selected or only the panels that are visible in the user interface. Another way to solve this is to have two options. This means there would be an option to select the whole physical wall or only all the panels that are visible at the current moment in the user interface.

Number ten is having the buttons that move the panels inward or outward or makes the panel wall equal. Moving inward and outward is plain and simple until of course the panel hits the physical boundary where the physical system does not allow it to be moved anymore however this can be dealt with some kind of information given that the panel has hit this max or min point. The problem of this requirement comes from the setting in which panels are set equal. There are multiple ways to solve this. For example, it could be that this button moves or the panels to the center but there is also a possibility that the button would move all the panels to be in the position in which most panels are currently. Another way which is not that useful would be to just make them equal in position in a random position. It must be discussed whether equal means all in the zero position or all in the position on which most panels are.

The eleventh and the last issue is the Wi-Fi connection which is used to get information from the panels and send information from the application to the panels. There is a basic structure of the information from Wi-Fi connection that comes to the application. Based upon this there will be information send to the wall that will be used to control the wall via the application.

From the point of view of these requirements they are relatively clear. One of the requirements that is unclear is how to scale the panels if there is too much data to fit the screen. This is not addressed in the original requirements and it should be made clear. A solution is needed for the cases when there are so many panels the application can become unusable for the small screen of mobile phones or even tablets. Another issue concerns the phone vs table size. The smart phones can fit a lot less information on their screen then tablets that are bigger. Because of this scalability must designed so that it works for both environments. Also, there is one button, which is the button that makes the wall equal, that is not well defined. There can in this button a meaning of centering the wall or another way it could work is the make all the panels the same inward outward position as where the most panels are.

4.3.2 Use cases

This subchapter discusses the use cases for the application. Figure 3 shows use case diagram that is meant to make it clear how the application functions from the user's perspective and what are the basic use cases.

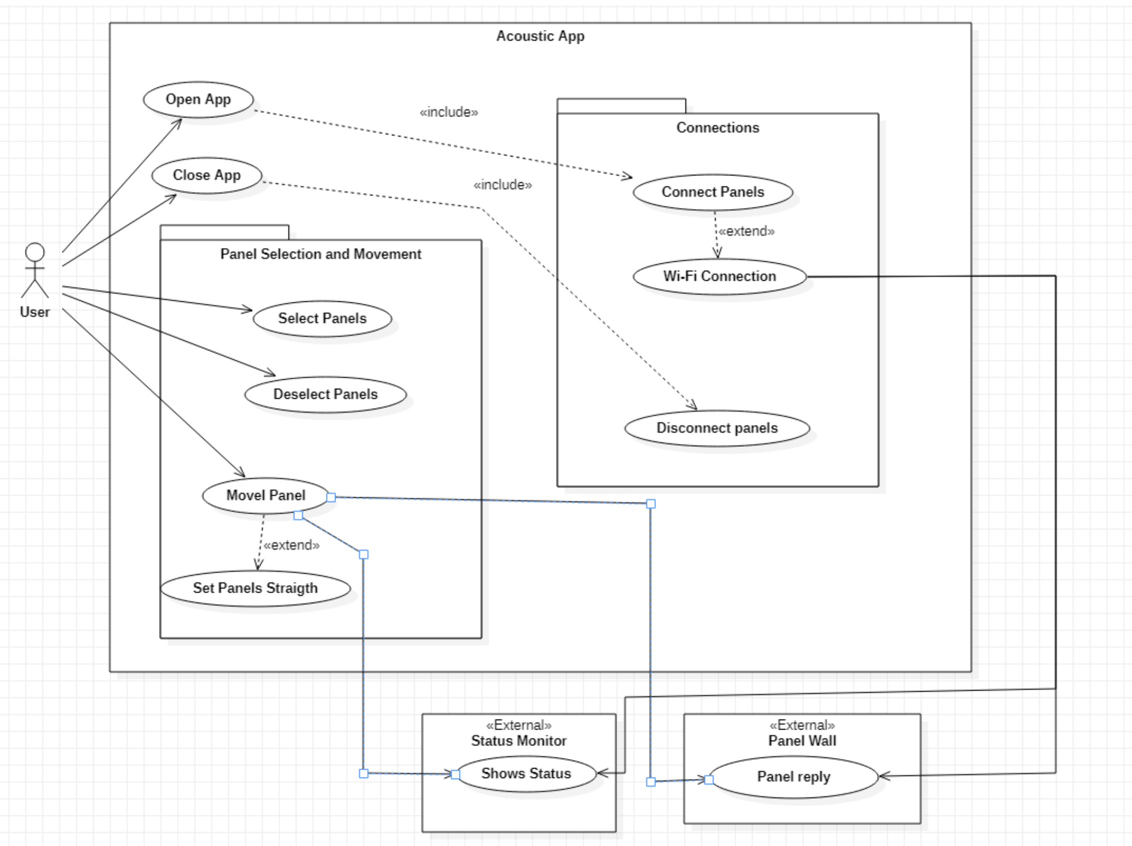

Figure 3. Use cases

This use case diagram shows the basic use cases for this program. The first use cases are opening app and closing app. These are self-explanatory. The use case of opening app will go to connect the panels use case which is in turn makes a Wi-Fi connection that is connected to external panel wall. This panel wall will then connect to the acoustic application and send data to it about the number of panels, established connections and panel positions. The other use case of close app will open the disconnect panels use case that will simply disconnect the panels and then the close app use case will close the application.

In panel selection and movement section there are multiple four use cases. First two are select and deselect panels in the UI. From there now of selecting and deselecting panels it is not necessary to connect immediately to the external panels. They can simply show the selected or deselected panels in green for selected and no colour for deselected panels. After when the panels are moved this information goes to the external physical panel wall. The move panel use case is when the wall panel is called, and the panels move. This is when the panel selection earlier will send information to the panel wall. Another use case specified earlier as a requirement is set panels straight. The move panels extend the set panels straight because it is still move action but this time all the panels that are not in a middle position will move to middle position. The status monitor is described here as external to the application because it does not include application moving or sending any signals to the wall, but it simply gets wall status and prints in the status monitor section of the UI.

4.4 Functional analysis

The main function of this system is to move the panels on the acoustic panel wall to desired configuration. The first function of this application is to connect to the wall. This is done by connecting to the wall when the application is launched. After connection to the wall is established the next issue with the application is to show that application is connected in the user interface. After this the application divides the panels in to view groups based on how many there are. There can be one or more view groups. The amount of these view groups will differ between mobile phone and tablet user interfaces. This is because mobile phones are smaller. When the application gets data of how many panels there are it divides them and after that it will draw the matrix for the panels. The panel amounts in a user interface can be hardcoded into the application or another way is to let user decide on how many panels they want to see.

The issue after this division and drawing of the panel matrix is basically about selecting and moving the panels. There should be a button for moving all the panels at once or centering them. The panels could be moved from their position to back or foward. In what way the panels can be moved will depend on the configuration of the wall. The panel movement is done to several positions. There also needs to be a way to select all panels or turn them all off. This function of course should only select or deselect all the panels in the current view. If panels from other view groups are also selected or deselected with all on or all off buttons it can confuse the user on how many panels will move when all of them are moved. One of the movement buttons should be center all the panels because there can be a need to start reconfiguring the panels and if there is no way to center all the panels at once centering them one by one is too tedious task. This center all should work for all view groups at once so that there will not be needed to move from view group to another in order to centre all the panels.

Final functions for this are to turn off the application and disconnect. Based upon the requirements disconnect should also be possible either when the application is open or when the application closes. Closing the application must leave the panels in the positions they are when the application is disconnected or closed. There is no need to move the panels.

These functions should be good enough for working with the panels. There is not much need for extra functions at least in the initial state because this application is quite simple. The application simplicity comes from the fact that it functions basically for only one simple task of moving acoustic panels. Since the application does not need to perform any other function there is no need to make it too complicated. The idea is to keep the application simple and at the same time make sure it performs its function as well as possible.

One of the important topics in this acoustic application is about how the panel wall view is scaled so that the panels can either be all shown at once if there are not too many of them or they can be all shown in groups if there are too many of them. This is done in the application by taking a percentage of the total size of the container that has the panels. The idea is that vertically there are at most four sets of panels which is 25 percent of the vertical size while horizontally there are at most five panels which is 20 percent of the container size. However, a minimum size for the panel in view must be set so that the panels are not going to get too small. This minimum size can be set during the implementation phase. The minimum size is hard coded into the application because of differences in size of the two environments of mobile and tablet. However, if there is a case that the phone screen is too small and could only show very few panels it might be necessary to make this whole application to only be functional in a tablet environment. This is a decision that must be made in the implementation face of the application. The original idea was to make it work both in tablet and mobile phone environments but judging from the user interface Figure 2 there can be a possibility that the tablet environment is the only suitable environment for this application. That is left for the implementer to decide.

4.5 System analysis

This subchapter is about using activity, class, and sequence diagrams to describe the functionality of the acoustic panel application. The class diagram is the main way the system is described while the other diagrams support this description and make it clearer.

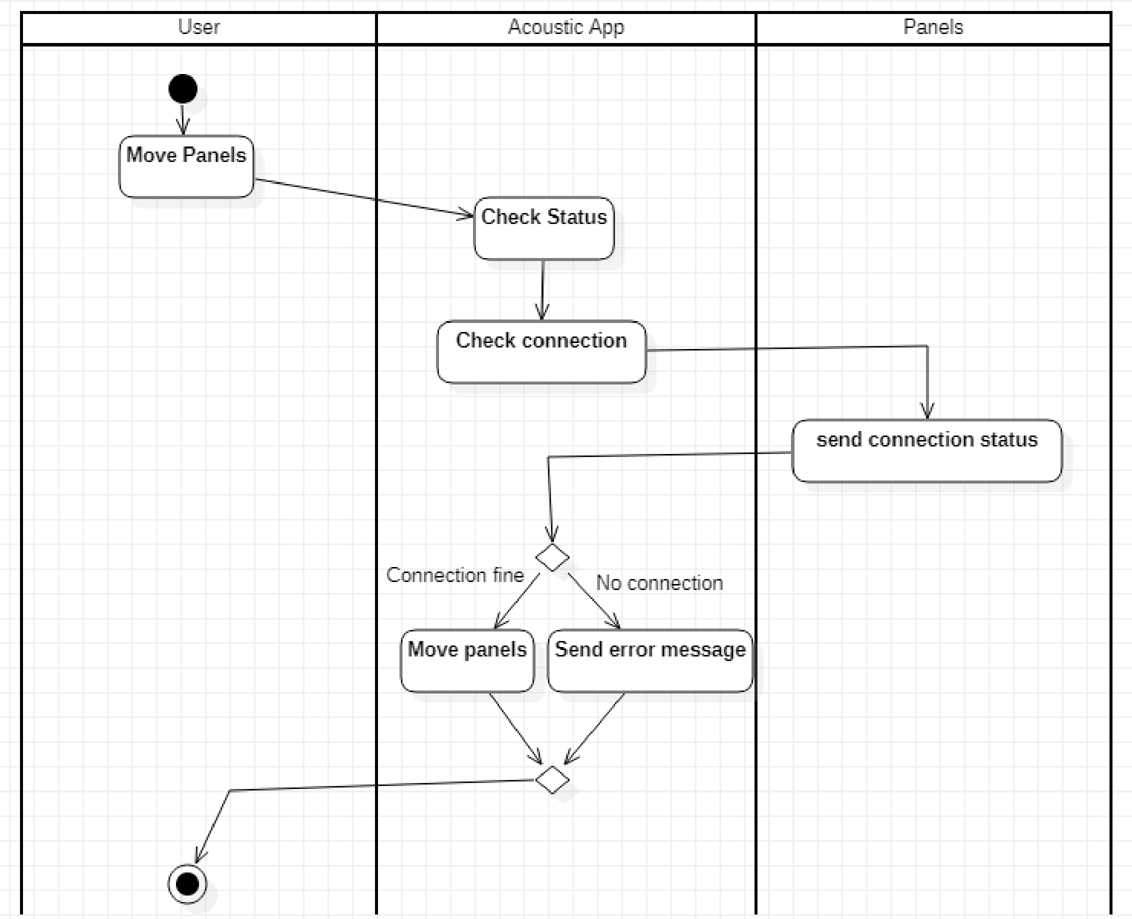

Figure 4. Panel movement

This activity diagram in Figure 4 shows how the panel movement is done by the acoustic app. The user first sends a signal to move the panels after which the acoustic application checks status and connection of the application to the wall. The wall sends connection status back to the acoustic application which if connection is fine moves panels or if there is no connection sends an error message. After this the activity is done.

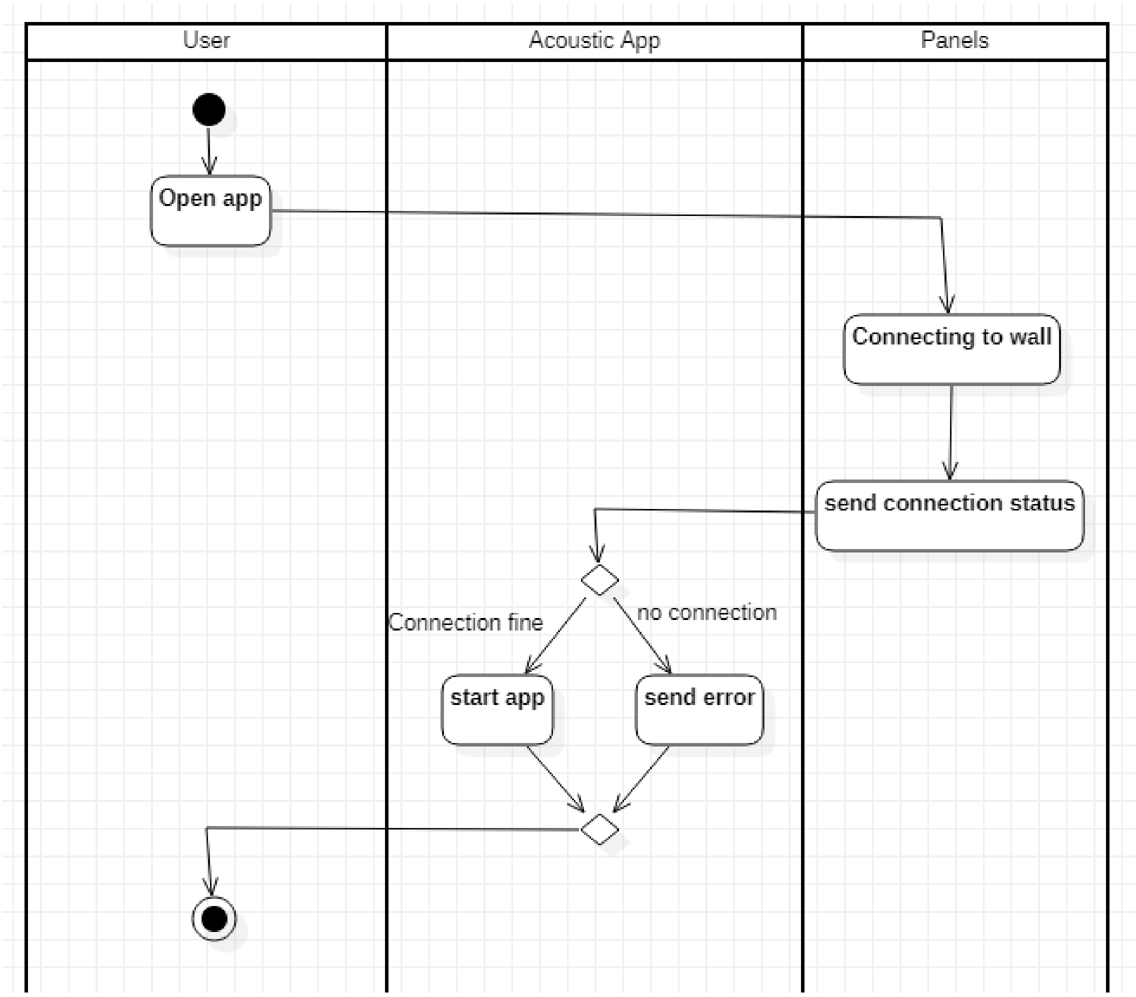

Figure 5. Open application

This diagram in Figure 5 shows how the application is opened. First the user sends a signal the application to open the app. The application starts opening but before opening it sends a signal to Wi-Fi connection which tries to connect to the wall. Connection status is sent from the wall to the application. After this if the connection is fine the application will start. If there is no connection the application sends an error message to the user through the status monitor.

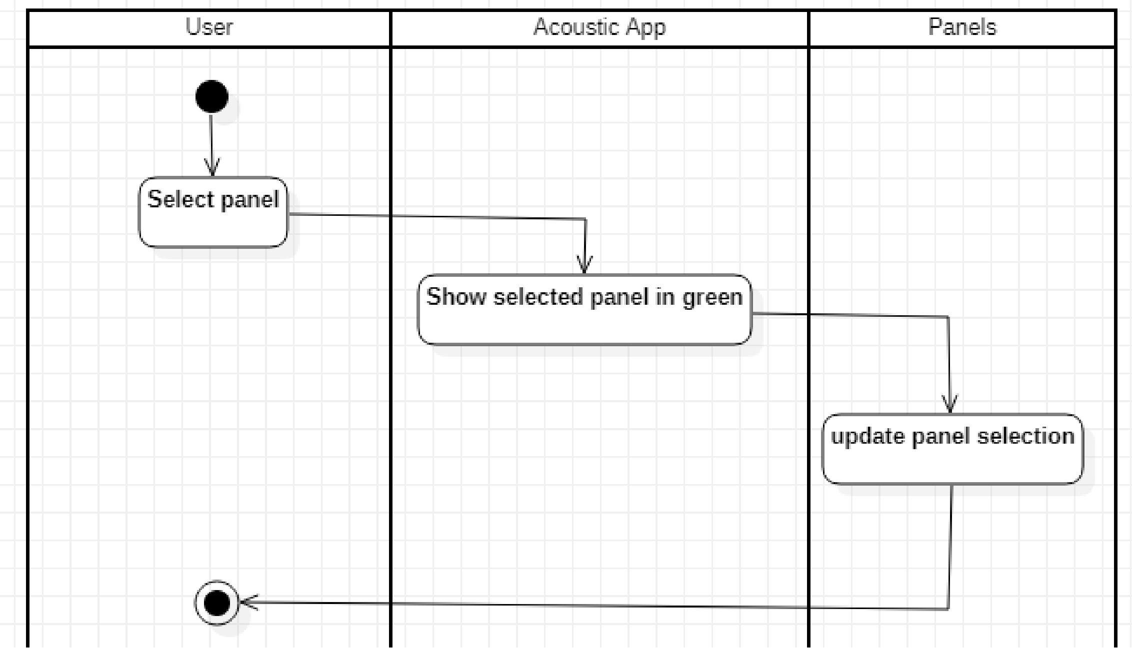

Figure 6. Selecting panels

This is a simple diagram in Figure 6 showing what happens when the user selects a panel. First user selects a panel after which the acoustic application shows panel selection in green. After this the selection the panel selection is updated to the panels and after this the selection process ends.

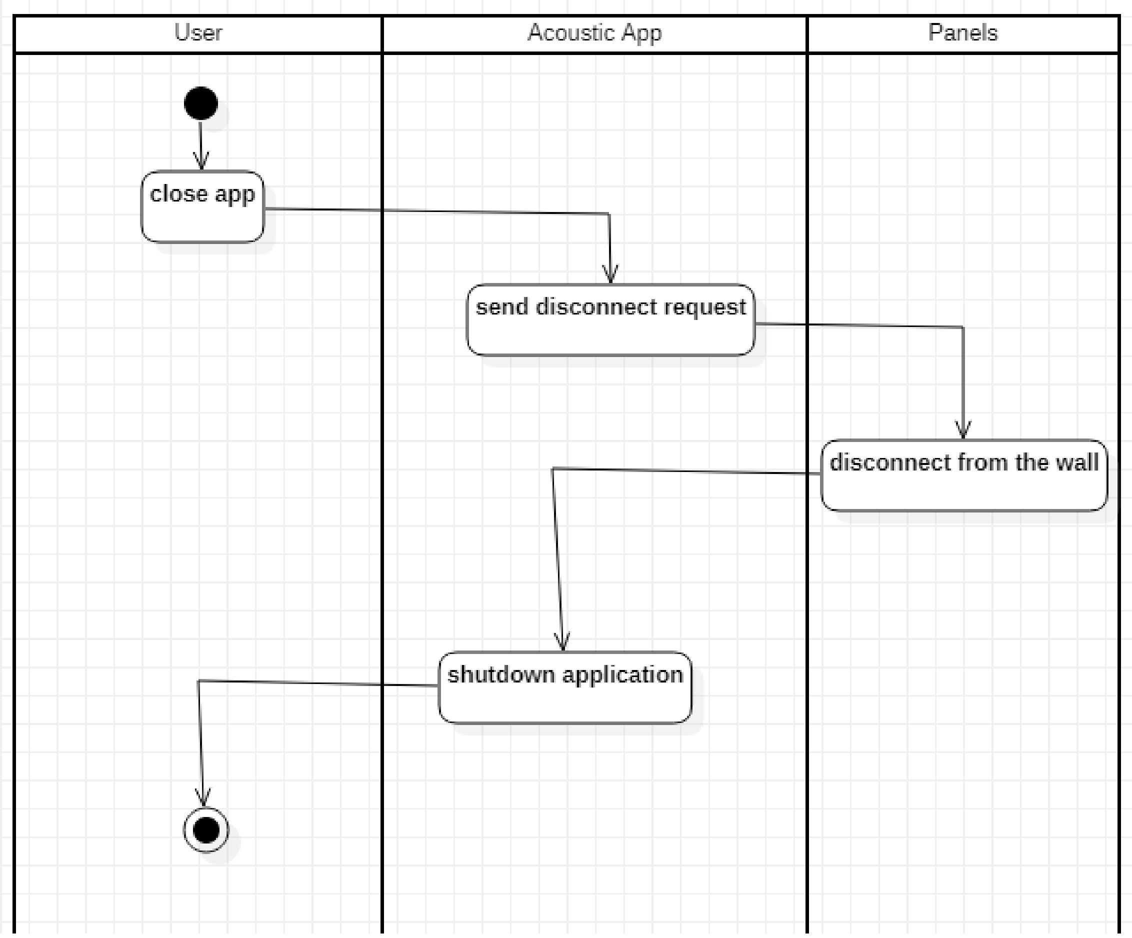

Figure 7. Close application

This Figure 7 diagram is about closing the application. First user decides to shut down the application. The user sends a shutdown command to the application and the application sends disconnect signal to the panels. The panels disconnect and the application shuts down.

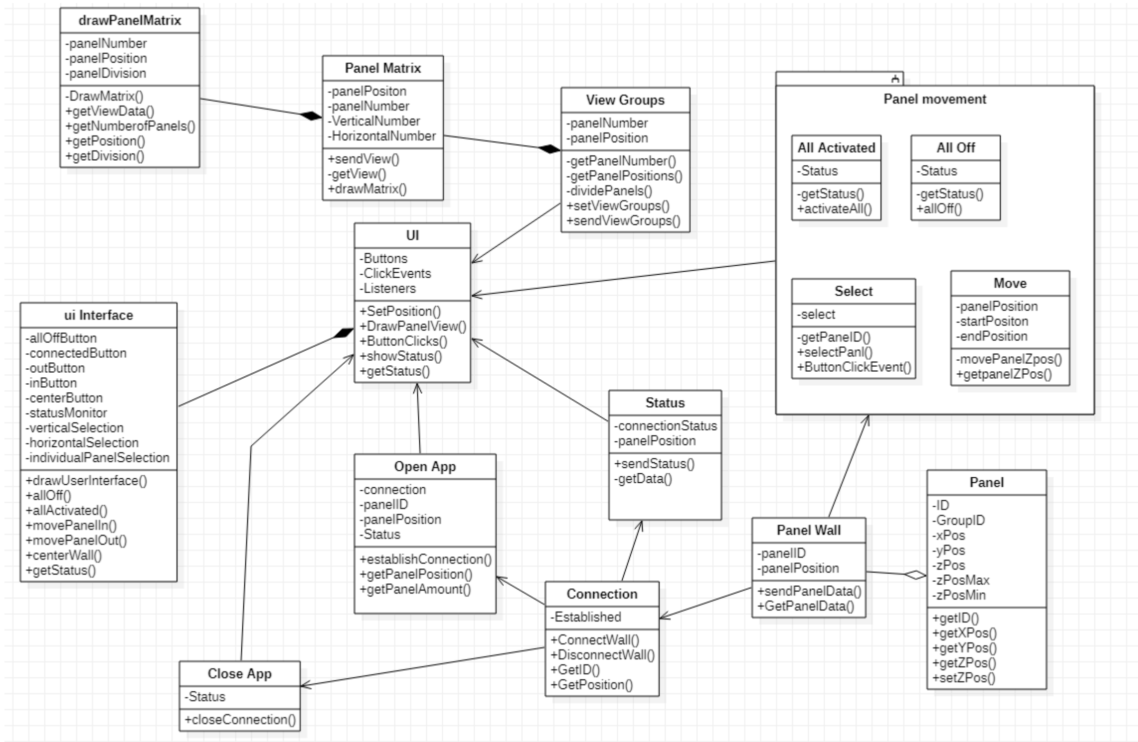

Figure 8. Class diagram

This class diagram in Figure 8 includes fifteen classes. These classes are planned around the use cases represented in the use case diagram and with functionalities that were thought of as important for the application build in the future. There is one interface class which means that the UI class must implement the interface class called UI interface. This class has multiple attributes and operations to be used in drawing the user interface. Operations that must be implemented by the UI class are about getting panel status, moving panels and activation of those panels. The UI class which implements the user interface is the central class in this application. It sets panel positions, draws panels views, implements button clicks, shows status of the wall, and moves panels around. The UI class is meant to be the central class because of the way the application functions. The functionality of the application comes from the user interface. UI class gets information from all the other classes and when the user of the application gives commands to the user interface the other classes do the heavy lifting.

Panel matrix class draws the panel view based upon information provided to it by the view groups class. The relationship to the view groups class is such that the view groups class gets the information about the panels from the UI class and divides panels into groups. This information on the panels and the division of the panels is send to panel matrix class. This class has a composition relationship with the drawPanelMatrix class that draws the matrix itself. The panel matrix class holds the information for drawing the panel view but then the class of drawPanelMatrix does the drawing for it. This separation is done so that it would be easier to understand how the panel matrix drawing works.

There are also two classes for the basic issue of open app and close app. The open app class will establish connection and get positions of the panels for the application. This class of open app has relationship with connection class which connects to the panel wall and gets positions of the panels in the beginning. The panel position is based upon where the panels are now when establishing connection. The panels can be in different positions based upon previous use. Centring them when application closes is not practical. The close app class will simply close this connection. The connection class is also in relationship with this class. The connection class has a simple method of disconnecting from the wall and the close app class uses it to close the connection.

Connection class is there to connect to the panel wall and get information from the panel wall. Panel wall on the other hand connects to the panel class. This class is the one that establishes connection to the actual wall. This panel class will send Id’s and positions of panels to the panel wall class that then sends the data forward. The status class will get data from the panel wall in order to send status to the UI class. Under the panel view in the user interface there is a status monitor text view that will show the status of the wall and this status class sends this status data to the user interface in order for it to use it to show the status in the status monitor text view. Panel movement is shown here in the diagram in its own package, so it is easier to see what is going on. There are inside classes All Activated, All Off, Select and Move. These classes are used for movement. The All Activated class is meant to be used by All Activated button while the All Off class is used by the All Off button. Select class is used for panel selection and Move class is used for moving the panels to the right positions.

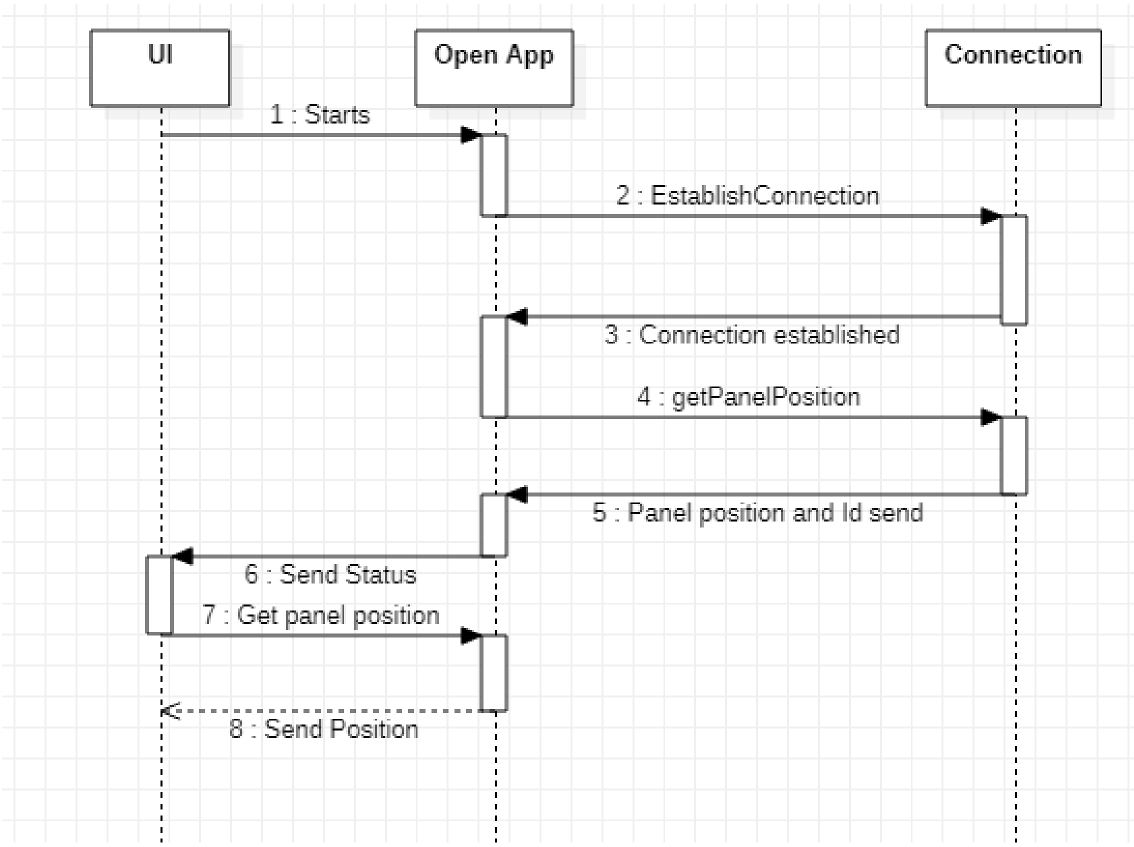

Figure 9. Open application sequence

This sequence diagram of Figure 9 shows how the application is opened and how it connects to the panel wall. In this diagram in order to start the application the UI class sends a message to Open App class which in turn sends a message to Connection class. The connection class establishes connection after which Open App class asks for the panel positions. The positions are after that send to the Open App class that will send status of the system to the UI class. After this UI asks for panel positions from the Open App which returns the panel positions. After this the application is opened and the UI will next start drawing the panel matrix which is shown in the next panel drawing sequence diagram.

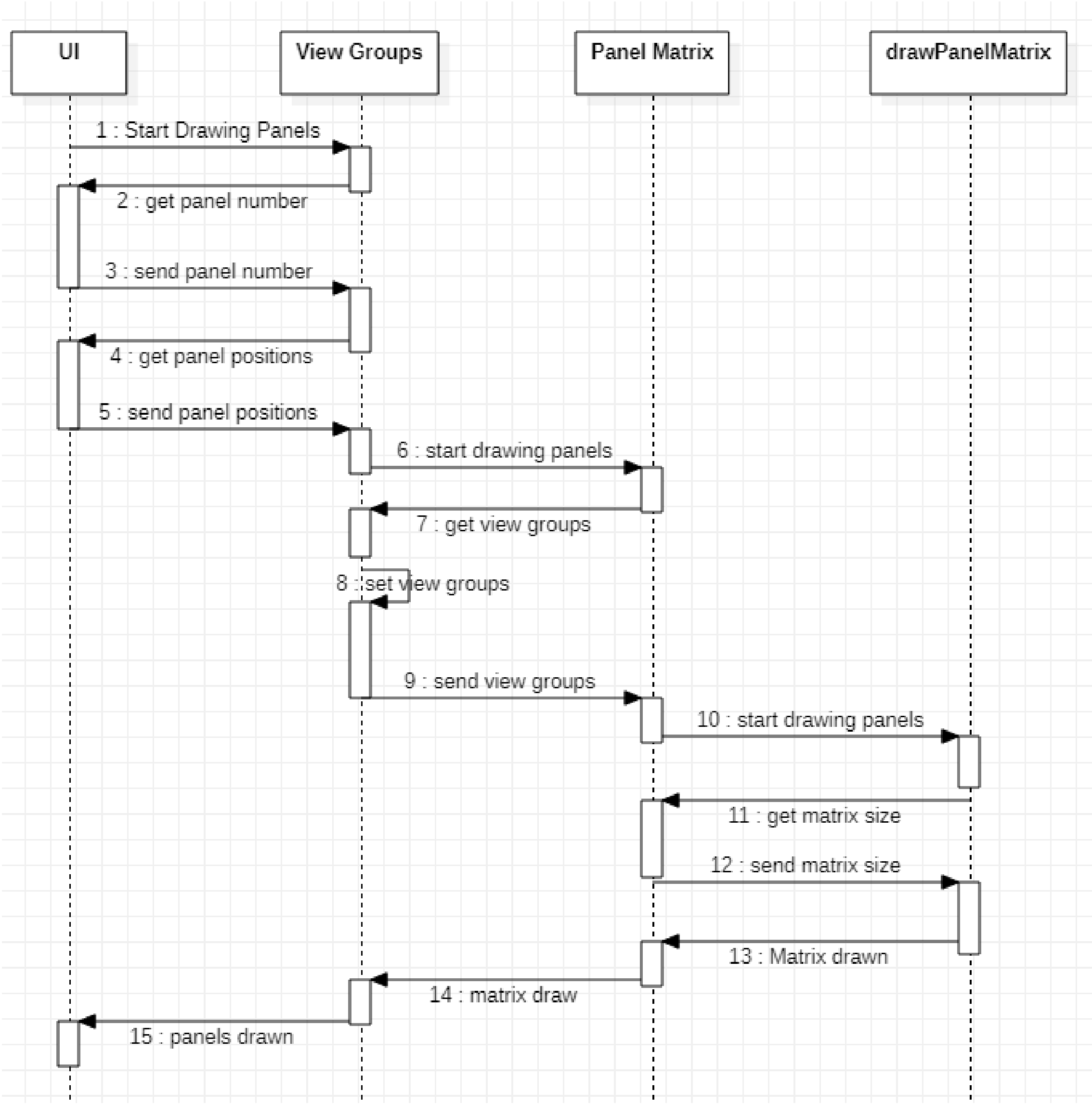

Figure 10. Draw matrix

This diagram in Figure 10 contains four classes that interact in sequence. The first is UI that draws the user interface it sends a message to view groups that then answers with a request for the number and positions of the panels. When view groups get this information, it will send a message to panel matrix for drawing panels. After this it communicates with itself about how to set the view groups. After that it sends a view group information to the panel matrix. Panel matrix sends a request to drawPanelMatrix to start drawing the panels the drawPanelMatrix class then the drawPanelMatrix class asks for the matrix size and after that it draws the matrix. It sends a message to panel matrix that the matrix has been drawn and this information is sending all the way back to the UI class.

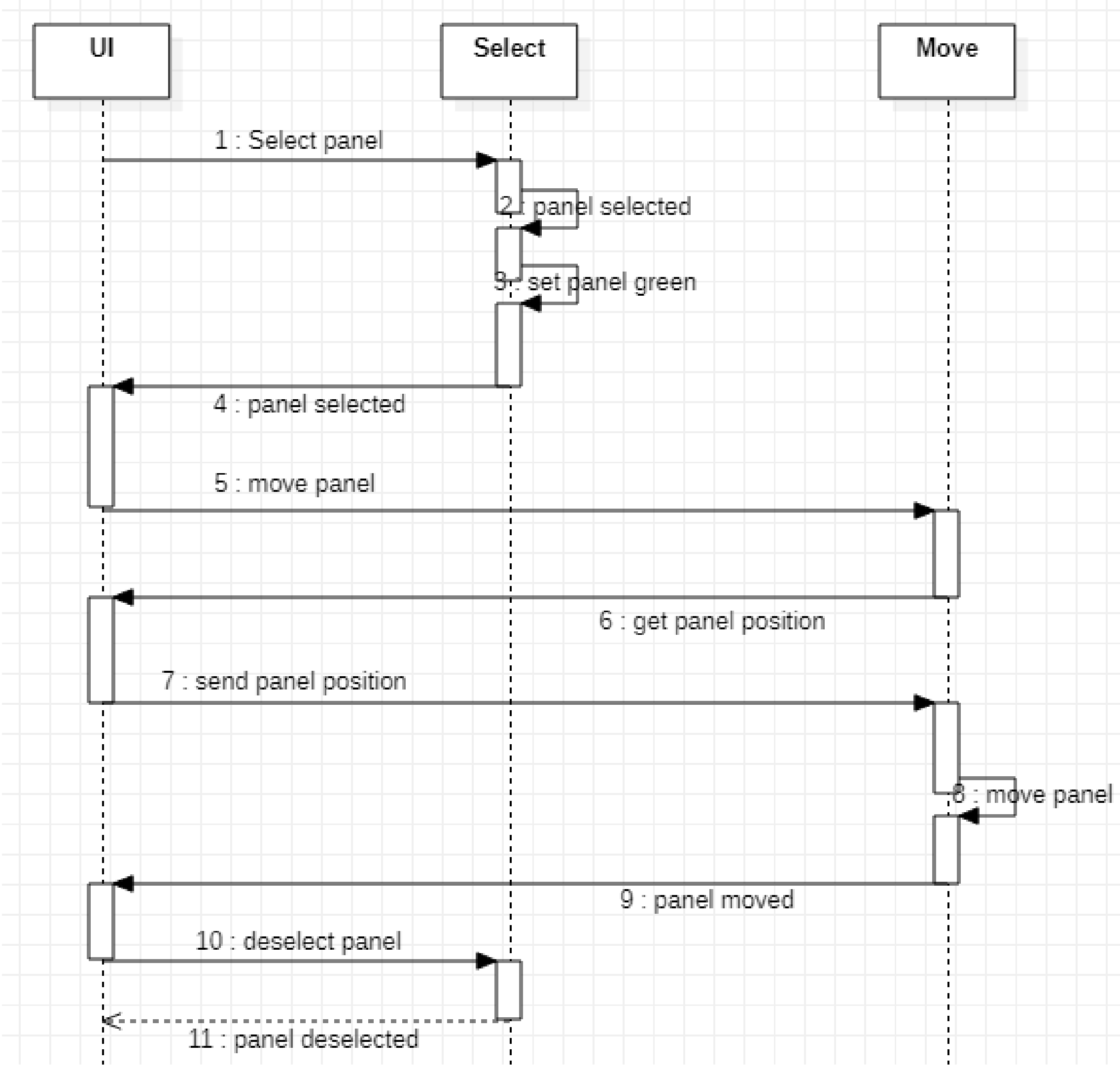

Figure 11. Panel movement

This diagram in Figure 11 shows the panel movement sequence. The UI first selects a panel and the Select class then selects the panel and sets the panel colour green. After that it sends the UI message that the panel has been selected. The UI sends the Move class a request to move the panel after which the Move class requests for the panel position. After this the Move class moves the panel and sends a message to the UI class that the panel has been moved. The UI deselects the panel and the Select class sends a message that the panel has been deselected.

4.6 Design synthesis

This plan was done according to guidelines of design science by Hevner and others. They have published a few articles about this subject. From the research cycles of the design there are three cycles that need to be worked with. First is cycle being relevance cycle in which it is estimated what are the requirements and how field testing is done for the design science. The issue with this plan is that it cannot be field tested since it is plan and not actually an application. However, when it comes to requirements which were introduced earlier one of the issues is that this design needs to be made so that in the application domain and environment the application works with the kinds of users there are. The users of this application are not technical users and therefore the application needs to be planned so that it can be used by non-expert users. However, in today's environment it can be certain that most of the users know how to use mobile and tablet applications without too much of a problem.

The technical requirements for this plan are quite simple. The application needs to work with regular mobile and tablet environments, and it needs to be able to connect to the acoustic panel wall via Wi-Fi. For the application, the technical environment is therefore relatively simple, and the biggest issue is to get the application work in such a way that the panels are easy to select and use. This requires dividing panels into view groups if there are too many of them. The application was planned for a system that has around sixty panels which would mean that in the mobile environment it would be practically impossible to keep all the panels in the same view. This problem is solved with view groups and division of these panels. Organizational viewpoint is not that important for this application because the application is not complicated and will not be used in a complicated organizational setting. Comparing it to many other systems that can be quite large this application should be simple enough to be used by a regular user. The relevance cycle also requires that problems and opportunities be discussed. The problems with this application are making sure that Wi-Fi connection to the panels sends the right data. Only the first panel sends data and all the other panels are connected to it. This means that the requirement is just to make sure this first panel sends the right data. As discussed, many times another problem is how to divide the panels in to view groups in such a way that the wall is still usable and in fact it should become easier to use.

Opportunities for this plan of an application are twofold. First opportunities come from the actual application itself. This kind of application for controlling acoustic panels when done right can be used of course for other places where similar panel structures are built, and these panel walls need to be controlled. This opportunity for using the application in other settings is important but at the same time these panel walls cost a relatively large amount of money and that can mean that building of them can be limited. Second issue is about mobile control applications in general. This application plan can be also used as a reference point for building other similar applications. There are already as introduced in Chapter 2 some mobile control application research. However, from looking at the previous research a lot of these applications are for different environments such as smart home control systems. This plan can be used as a reference to systems that are similar such as for example solar panel control. Solar panels sound like a much bigger issue but when thinking about them controlling them to different positions and shutdown and so forth is relatively like controlling acoustic panels. Also, in Chapter 2 there was introduction of a doctor appointment system. This was introduced because these kinds of appointment systems also have a relatively simple interface but of course for these appointments more rigor is needed for how the appointments are done. However, user interfaces that are relatively simple to use can look upon this plan and use it as one of their reference points.

For the rigor cycle the grounding of this plan in the theory was done via a few examples of similar applications. Otherwise the theory is mostly about heuristics and design science. There was a problem in finding similar applications and articles about them. However, the plan was grounded on working with heuristics as a reference point to make the application as usable as possible. This kind of heuristic evaluation is done in the next Chapter 5. The plan also adds to the knowledge base via making an artifact of a control application. Since there are not that many studies in this kind of field the knowledge base is increased.

Earlier in the functional analysis basic functions of the system were described. In this analysis it becomes clear that the design should be kept simple and the functions of it should be simple enough but at the same perform the function they are supposed to perform. Application simplicity is because of the way the application functions for a single task of moving panels. Because it is focused on single task and does not perform complicated tasks keeping it simple works best. Also, the users of the application most likely know a lot about acoustics of performing stage, but they might not be proficient in using complicated application. This demands for simplicity. This design has taken that into account and worked so that this will be fulfilled as a requirement.

For this design, the idea is that the user interface is the central class. The user interface is built so that when the application is opened it will call the open app class which will establish connection and through the connection the panel data is send to the user interface that calls the view groups. The view groups will then establish grouping for the panel that scales between how many panels there are. This class has composition class that draws panel matrix and panels. The class that gets status is independent from opening the app and the status is shown in the user interface status monitor view. Application also closes so that the user interface calls close app class that through the connection class will disconnect from the panels. Moving of the panels is done in its own grouping so that the user interface calls it and its selection and moving of the panels is done. Also, all panels can be put off or activated depending of users will.

This design is made like this so that the central control is in the user interface. The application is relatively simple since it is only there for controlling the panels and there is no other purpose for it. This means that it should be good to also keep the structure simple for the application in order to avoid unnecessary complication. The software function is done in the user interface and the other classes support the user interface. There also are not that many things that the application does. Basically, the application moves panels, connects to them, and draws a panel view so that they can be moved. Technical details also do not need unnecessary complications. With the application being relatively simple the design of the application is also made simple. A few activity diagrams and a use case diagram are provided so that it can be seen how the application works.

5. Heuristic Evaluation of the Artifact

In this chapter heuristics will be used to evaluate the plan that was made in the previous Chapter 4. These heuristics include Nielsen's (1995) heuristics and self-built heuristics. The evaluation is first done for Nielsen's heuristics and a second with the self-built heuristic. The point of the evaluation is to find out how well does the planned mobile application work for solving the problems that were posed in the research questions. These questions put a lot of emphasis on the usability of the application from the point of view of a non-technical user. After the evaluation with heuristics is done these questions will be answered. In the discussion chapter there will be more information about how to fix the problems that come apparent when answering these questions.

For criticality of the errors the Nielsen Molic (1990) scale is used. The scale includes three dimensions. First is severity which describes how bad the usability problem is. This means that the usability problem makes the application unusable. Second dimension is frequency. However, due to the artifact being a plan for an application and not the application itself this dimension is omitted. This is because if there is an error the frequency of it is always one in the plan. Third dimension is criticality. This means how big the problem is for the application. Does it crash the application or totally shut it down and so on? The dimensions will be evaluated in three different levels. First is low, which means the problem is not important, second is mid-level, which means the problem is important and needs to be fixed, and third is high, which means the problem absolutely has to be fixed before the application is used.

5.1 Nielsen's heuristics-based evaluation

There are ten heuristics in the Nielsen 1995 heuristics list. These ten are:

1. Visibility of System Status

2. Match between System and the Real World

3. User Control and Freedom

4. Consistency and Standards

5. Error Prevention

6. Recognition Rather Than Recall

7. Flexibility and Efficiency of Use

8. Flexibility and Minimalist Design

9. Help Users Recognize, Diagnose and Recover from Errors

10. Help and Documentation

The first heuristic is about visibility of the system status. This in the plan for the application is shown in the status monitor. The status monitor is showing the status of the system. The selection of the panels is done so that green shows the selection of a panel and gray or no color shows that the panels are not selected. To make this system status better the panels that are not working could be shown in red and panel positions instead of being shown in the status monitor could also be shown in the actual panels as a number for example. This usability problem is mid-level severe, and criticality can also be estimated to be on the mid-level.

The second heuristic is match between the real world and the system. This match for this application comes from the information that the physical panel wall sends to the application. There should be a way for the application to get the right information and any errors with the panels that they might have. One issue is with the scaling. Since in bigger panel walls it is impossible to represent the whole wall in the user interface there needs to be some scaling of the application in a way that it represents the real physical wall but at the same time is usable from the users perspective. This scaling was discussed in the plan. The representation of the wall in the user interface cannot be the same as the actual wall. In the scaling discussed earlier there was an idea that the wall would be represented at the most in 20 panel configurations. However, this might not be practical if the wall is built in such a way that the panels are not actually in a rectangular formation. There needs to be more discussion on the scaling and view groups that could base the views also on the actual positions of the panels in their physical environment. This problem is on the mid-level in severity and criticality.

Third heuristic is user control and freedom. The user must have the freedom to control the panels in any way they want restricted only by physical possibilities. In this way the plan does solve this problem and does give the user freedom to select and move panels in any way they want based upon physical limitations. There are not that many functions in this application so the correction of mistakes can be simply done through the original interface. For example, if a panel is accidentally moved to a wrong position this panel can simply be selected and moved back to right position. Also, if a situation comes where panel positions are totally wrong there is a button to center all the panels and start all over. Specific buttons for undo and redo type of functions are not necessary since this application is relatively simple. In case there is need panels can be turned off or connection can be redone. Also, status monitor shows the status of the application in a visible way so redoing panel movements is relatively easy.

Fourth heuristic is about consistency and standards. In the plan no standards were used but consistency is there in the user interface. There are relatively few functions in this application, so consistency is not that hard to come by. The consistency is in the fact that panel buttons look the same and the application does same function for every single panel when a button is pushed. The simplicity of the application and the fact that it has a single view for a user makes consistency not a problem. Also, the functions of the application should not be hard to use by the user since they are not complicated, and they clearly state the function they are for.

Fifth heuristic is error prevention. Now the plan has no discussion related to this heuristic. There is no error prevention possible. This is partly because the application is simple but also, because it was thought that such issues could be figured out in the implementation phase. However at least basic errors such as disconnection from all or single panels or there being something wrong with the panel should have been discussed. In the next iteration this is discussed. Also, it can be necessary for there to be an emergency stop button in case the panel movements start working in the wrong way. This is discussed in the second iteration. In the Nielsen Molic (1990) scale the lack of error prevention in the plan would be bad because if the wall has a physical error but it is not reported to the user it can cause damage. Also, when evaluating criticality of it I would say this is at least mid-level. Also, in severity this can be considered a mid-level problem. This means that this problem with the plan needs to be fixed in the second iteration.

Sixth heuristic is about recognition rather than recall. This means that the user should be informed by the system enough so that she/he would not have to remember how the application works. Recognition in this case is easy in all the application functionalities such as moving the panel/panels or centering them or selecting them. One issue as discussed in the plan is that the status monitor on the downside can be too small to be used to inform the user about current panel positions. Also, if panel positions were reported there this could be confusing. Therefore, it is better to give the user the positions of the panels in number from -3 to 3, with 0 being center, in the panel view. This means that each panel would show where it is in the panel representation of the user interface. Otherwise, all the functionality of the application is simple and there is no need for extra recognition by the user.

Seventh heuristic is flexibility and efficiency of use. This heuristic explanation is that there should be additional functionality for expert users and in this case this heuristic is not applicable for this work. This is because the functions of this application are limited and both an expert and novice user would be doing the exact same things with the application. There is no need to separate novice users from expert users by giving expert users the ability to tailor the application. The only issue is that maybe there could be some tailoring is in the panel view which could be set to use panels. This could be considered for all users however, but it might be quite unnecessary. This idea will however be discussed in the reiteration of the plan.

Eight heuristic is about flexibility and minimalist design. This means that all the actions and dialogues with users should have a minimalist quality. The dialogues are almost totally absent in the plan represented. This is a problem but from the point of view of severity and criticality it is low. There could be consideration for some but since the application is so simple dialogue might not be necessary beyond the status monitor dialogue on the downside of the application. The plan uses very minimalist design choices with only having necessary functions and no kind of aesthetic.

Ninth heuristic is help users recognize, diagnose, and recover from errors. There are always errors that can happen. One of them is moving the panel to wrong positions. However, in this kind of simple application panels can be immediately moved back to original position so there is no need for error recovery here. One error can be a physical error in which a panel does not actually work. This can be shown in red in the application and this error then must be physically solved in the panel wall itself. There is no way to solve this kind of physical problem in the application. However, there is one situation in which if the connection to the panel is incomplete and a panel is showing red the application could in the status monitor show a need to restart the connection to try to first solve the problem. This kind of problem with not representing errors in physical panels is severe in mid-level and critical on a high level. High level criticality is because if the application does not give any recognition of physical problems with the panel wall this can lead to damaged wall or wrong acoustic configuration due to one or more of the panels not actually moving to the right position.

The tenth heuristic is help and documentation. Once it is shown to the user how the application works there should not be need for documentation due to the simplicity of the application. However, in the plan it could be mentioned that some basic documentation could be created for the user for example in the case of when a new user must use the application without prior training. This situation might come to reality for example when original user is sick and somebody else must use the application. This could mean that the user now using the application does not have training for it. The application is simple and intuitive to use but in this case some help might be had with a small documentation of the application for example in a pdf file. On the Nielsen Molic (1990) scale this problem can be estimated to be low in the levels of criticality and severity.

5.2 Self-Built heuristics-based evaluation

The self-built heuristic are as follows:

1. Visibility of the status of the system to the user.
2. Learnability of the system and concise explanation of possible system usages.
3. Matching of application status and functions with real world physical world and interaction with it. Intended outcome is visible and intuitive.
4. Ability of the user to control physical devices with the application.
5. Preciseness of error messages and ability of user to understand error messages.
6. Efficiency of application usage.
7. Error recovery of the system. If no possible way of error recovery notifying the user precisely what went wrong.

In these self-build heuristics, the first heuristic is like the Nielsen (1995) heuristic first point. The system status and panel button statuses were discussed in the last subchapter. The status of the system should be all the time visible to the user. This is done by the status monitor and as discussed before the panel buttons with their markings. There is really no need for additional ways to show system status.

On the second heuristic of learnability of the system and concise explanation of possible system usages this application is not hard to learn. The buttons have a text in them that explains what they do. The only problem in this perspective can be knowing which panels the panel buttons are for. If the panel wall is just a wall with panels exactly next to each other in rows it is easy to see where the panels are. However, these panels being acoustic means that they can be separated in different groups across the music hall. Because acoustic panels are not all in nice rows. This is one problem that was not addressed enough in the plan and should be solved. This on the level of Nielsen Molic (1990) scale is a mid-level problem in terms of severity and criticality.

The third heuristic in this application means that when user gives a command in the application it the real-world physical wall should respond to it. If the user wants to move a panel forward the panel must move forward in the physical panel wall. The outcome also needs to be shown in the system. The panel movement can be shown in the panel button by showing the number where the panel is. When the panel moves the corresponding number also changes. This way the user can be sure that the panel moved in the physical wall. With this kind of numbers shown as to the location of the panels it can be seen by the user that the panel has moved on the wall as intended.

The fourth heuristic is about users having as much control over the physical acoustic wall as possible with the application. There is an idea here that expert users can be given more control. However, with this application plan the expert user will have as much control over the application as the novice user simply because this application that is planned is a straight-forward application and the expert user does not really need more control compared to the novice user.

Based upon the fifth heuristic at this point in the plan for the application errors were not considered at all. This means that in the next iteration of the application users need to be somehow notified of errors. One error notification of course is the status monitor. The plan for the application did include this monitor that can be used for possible error messages. However, this status monitor is not good for showing individual panel disconnection or other errors since if there are more errors to be shown they can become confusing when shown in the status monitor. Better way to show individual panel connection and other errors is to show them in the panel view by for example showing a disconnected or broken panel in red. Additional information about these errors could be shown then in the status monitor for more expert users to decipher. This problem with the heuristic can be said to be mid-level in severity and criticality.

For the sixth heuristic the way the plan describes the application this heuristic is fulfilled by the application since it has been made as simple as possible with only the functions that are needed for the task that it has. There is no cosmetic dimension except for the user to be able to understand what is being done and where in the panel wall. Cosmetic improvements can of course be made but they are not in the current plan. The application plan was made as efficient as possible and mostly only thinking about usage for the panel wall control.

For the seventh heuristic there is something that still needs to be worked at on the next iteration of the plan. Some sort of error recovery process needs to be described since certainly there are errors that happen sometimes if not with the application itself at least with the wall. A part of the wall might break or become disconnected for some reason and if possible, there should be a way to find out why and fix the problem. This is one of the things that will be discussed in the next iteration of the plan. This problem has a severity of mid-level but criticality of mid-level also.

5.3 Iteration of the plan based upon heuristic evaluation

In this subchapter iteration of the plan will be done. It is done based upon the previous usability heuristics. This subchapter will only describe those aspects of the plan that need to be modified. Also, there are additions to the plan. There is no need to discuss those issues in the plan that are not problems based upon heuristics evaluation. This is done in order to keep this subchapter short and understandable. Idea of this subchapter is based upon Hevner (2007) iterations of the design cycle. The design cycle is one of the three cycles in design science and it encompasses building the artifact and then evaluating the artifact. After that, a new iteration of the artifact is done.

From the Nielsen's heuristics (1995) the first heuristic was about visibility of system status. From this evaluation it was noted that if an individual panel is not connected or not working this panel could be shown in red on the panel view. Also, the positions of the panels should show in individual panel button to make it clear which panel is in what position in the wall. This being a mid-level problem is because it can make it hard for user to use the application if the panel positions are unclear. Showing the panel positions only on the status monitor can be confusing. Status monitor is used for other information such as more specific information about why the panel is shown red and, if possible, what is wrong with the panel. There can be a problem with showing what is wrong with the panel if the panel cannot send this information to the application. This problem is best solved by showing the status of the panel in the panel button itself to make it clear what is the status and position of the panel.

Second issue that was not discussed in the original plan is the issue of how the physical panel wall can be built in such a way that dividing panels into 20 panel groups is not going to represent the physical panel wall in a way that is not confusing. The issue is that there can be a panel wall that is straight up easy to divide into basic groupings of something like 20 panel with four rows vertically and five rows horizontally. However, there can be multiple or individual panels on the physical wall that are hard to divide like that straight up. There should be a way in this case to divide the panels in such a way that it works with the actual wall configuration. However based upon previous plan there is a situation in which the information coming from the panels might not be able to differentiate panels that are individually built in such a way that grouping them in the way the plan describes is not practical. This was mid-level problem in severity because it can become hard for user to use the application when these panels are not represented in understandable way. Because of the information that comes from the panels however, this is an issue of future development.

As discussed earlier there is a need for way to inform the user of the application about possible errors with the connection of the application and the panels in the wall. There is a possibility that one or more panels have something wrong with connection to the application in which case this has to be shown in the panel view in order for the user for example can disconnect and reconnect to the wall in order to establish connection to that panel. There is another issue also. This issue is that there can be something physically wrong with one or more panels. In this case if possible, the application should show what is wrong with the panel and if not possible to show what is wrong there should be at least information shown that this panel is broken. The status monitor on the downside shows what is wrong with the panel if information is available. Also, emergency button must be implemented since when changing configurations of the panel wall something can go wrong and the panel wall movement must be suddenly stopped. The same issue of error prevention also came up in the self-built heuristics. The solution presented there is the same as for the Nielsen's heuristic evaluation. This problem was estimated to be mid-level problem because error prevention is important. In the original plan there was no error prevention available so this must be implemented as said above.

There was an issue found in the heuristics that was discussed a bit before which is that the status monitor can be too small to show all the information about panel positions and errors in the panels. Therefore, a good way to show panel positions is in the panel buttons themselves. This is also less confusing for the user. The status monitor should show information about the panel wall status in what comes to connection to panels and it should show additional information about possible errors. This way the status monitor can be used for additional information while the panel buttons can show basic information about the panels. For example, when a panel turns red because of disconnection the status monitor can show the reason for the disconnection if possible but the panel itself shows red to mark that there is something wrong.

There was an issue with giving expert users additional information and more freedom to use the application. However, this application is relatively simple and therefore expert users or regular users are practically on the same level of usage. The only thing that could be done in this perspective is to give expert users in the status monitor extra information about the panel positions and possible errors. However, when it comes to giving expert users additional shortcuts and customization this is not needed for this application.

Another issue is about dialogues in the application. These are mostly not applicable due to the nature of the application. From the previous plan the user interface is intuitive and there is no need for additional information. However, as mentioned before the status monitor can be used give extra information about the panels when needed. However, additional dialogues in other ways are not necessary. Another issue that came up was user manual for the application. There can be some documentation done on a pdf file about basic functionality and the meaning of messages in the status monitor. This documentation can include text and pictures to show the user how to use the application. This should be done so that if a new user that has not used this application before has to use it for a reason. However, it must be noted that this is quite intuitive user interface and there most likely is not much need for this documentation. Most issue can be deciphering of the error messages in the status monitor.

It has already been said that if there is an error with a panel it should be shown in red and the status monitor should give information about what is wrong. Based upon ninth heuristic this is also an issue. If there is something wrong with a panel in a worst-case scenario the panel can break when trying to move it. Therefore, this is a problem that gets the level high in criticality. This problem is solved by showing the panel in red to sign a problem and giving possible information about the problem in the status monitor. Also, it is important to stop the user from trying to move a panel that is broken. The broken panel must unavailable for moving from the application.

On the self-built heuristic evaluation there was an interesting problem found which is that the panels can be in the wall in such a way that the panel view on the application is not representing panels in the correct way. When looking at how acoustic walls are built, they mostly make a flat wall that fits this application well but at the same time big music halls can have individual or multiple panels in locations that are not as intuitive. For example, a few panels can in the front or back in such a way that the grid created by the application makes it hard to know the location of the panel in the actual physical hall. This problem is solved by getting additional information from the panels which is their coordinate positions on the wall. This position information is then displayed on the panel button view in such a way that it is understandable to the user.

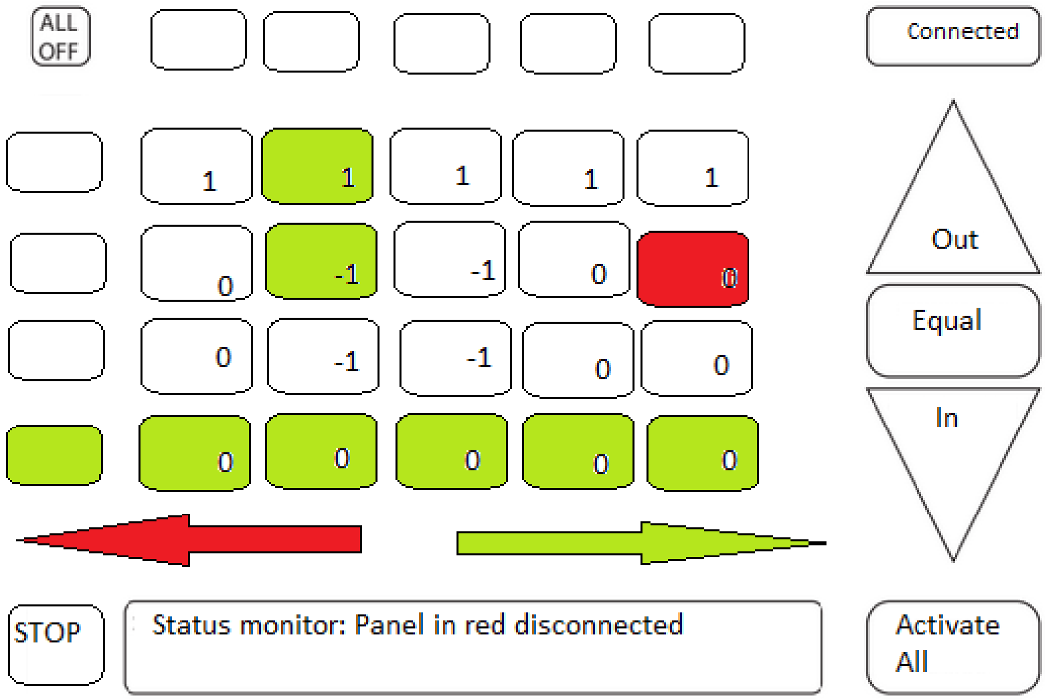

Figure 12. New user interface

Figure 12 shows a design for a new user interface for this application. In green are the panels selected by the user. The buttons with no numbers are all for selecting a complete row or column of panels. The down left side green button is selecting all the panels in that row. There are also individual panels selected. The panel buttons have numbers in them that indicate the positions of the panels. The panel button in red indicates that this panel has a problem and the status monitor identifies the problem. The arrow buttons allow movement between views. The arrow in red means that there are no more panel views in that direction and green shows that there are more panels. STOP button is the emergency button discussed earlier. Pressing this button will stop all the panel movement. Panels are divided into 20 panels per view at most.

This new plan only shows the new functionality that the original plan does not have. There is no need to present more in here. To summarize the new plan there is a need for error prevention. With the heuristic evaluation this was the most important part that was missing from original plan. The panels that have some sort of error should be shown in red and the information about the problem should be summarized in the status monitor on the downside. Error prevention is important so that the user can identify errors and fix them without damaging the wall more. This is important so that further damage can be avoided. Also, there should be an emergency stop button in a case were error comes when executing orders from the panel view. For example, if a panel does not move to the right location but gets stuck trying to move there is a need for this kind of emergency button. Another issue was showing the panel status. This is done by showing the location and position of the panels in the panel buttons. Also, status of the whole system is shown in the status monitor. Another issue is about documentation. Even though the application is relatively intuitive there is a need for instructions to be done in a separate pdf file just in case there is a need for this kind information for example for an inexperienced user. This pdf file should show basic usage information and, it should show how to decipher error messages in the status monitor. Last issue was about how to represent the panel wall when the panel position on the physical wall is hard to make visible on the panel view in the application. This problem is solved by giving coordinates of panel position to the user on the panel buttons or another way to solve it is to establish view groups in such a way that it considers panel coordinates. This way the user can know where the panel is without a problem. This of course also means that there is a need in the pdf documentation to explain how the panel position coordinates work. All in all, this new plan should solve the problems that were found in the heuristic evaluation.

5.4 Synthesis of heuristic evaluation

Heuristics evaluation was done in order to find problems with the original plan. There were a few problems found and most critical and severe of them was the lack of error prevention in the original plan. There were other problems also that were discussed in the new iteration of the plan. The heuristic evaluation was done so that the Hevner (2007) guidelines on design cycles could be fulfilled. The design cycle went in such a way that first there is a design of application based upon other important cycles which are relevance and rigor cycles. After the design of the application was done the design cycle heuristic evaluation was done and after that new iteration of the plan was done. The plan iteration was done so that only the new areas of the plan were discussed.

There were problems identified and they were fixed with the new plan. Now there is no need for additional cycles at this time and therefore the plan done so far is where the artifact building ends. Heuristic evaluation has produced results and these results were used for iteration. Some of the heuristics between self-built heuristics and Nielsen's (1995) heuristics were overlapping but this is to be expected. The heuristics were for different purposes but of course some of them must be like the ones previously done in order for them to cover as much territory as possible. The issue is that the heuristics should be such that they can be used by themselves for evaluation of applications and this means that some overlap needs to be there so that these heuristics can be used independently. Here are the heuristics for a close look:

1. Visibility of the status of the system to the user.
2. Learnability of the system and concise explanation of possible system usages.
3. Matching of application status and functions with real world physical world and interaction with it. Intended outcome is visible and intuitive.
4. Ability of the user to control physical devices with the application.
5. Preciseness of error messages and ability of user to understand error messages.
6. Efficiency of application usage.
7. Error recovery of the system. If no possible way of error recovery notifying the user precisely what went wrong.

The heuristics could have been also a bit clearer and maybe there could have been more of them to be used for the evaluation. The overlapping with Nielsen's heuristic is not a problem however because these new heuristics were also meant to be used independently from Nielsen. If overlap were eliminated these new heuristics would only be an addition to Nielsen and could not be used by themselves. These heuristics could have been more precise and, there could have been more of them. When considering that some overlap will occur between these and Nielsen it would have not been bad to overlap a bit more to get more precise heuristics.

So, the explanations of the heuristics could have been more precise and to the point. For second iteration of the heuristics the explanations of the heuristics should be more precise, and some heuristics should be added by modelling the Nielsen's (1995) heuristics so that these heuristics could be used more independently from Nielsen. Dependency on Nielsen to get enough information is not good if using these heuristics by themselves. However, Otherwise the heuristics worked well for the evaluation of the artifact of this thesis. Problems with the application were found and they were corrected in the second iteration. There is a need for improvement but that is for future work.

6. Discussion

In this work there has been discussion about the heuristics and design science research. The idea was to work on a design science research artifact that was the plan for an acoustic panel application. The panels are physical objects that send information to the application that will decipher this information and make controlling of this physical wall possible. The plan that was made was then evaluated. The evaluation did identify problems with the plan, and it was reiterated. Also, in this part of the work the answers to the research questions are going to be discussed.

There are a few results of this work. First is the heuristics that were used. There were two sets. The first set was Nielsen's heuristics. These heuristics are used a lot, but they have a problem in that they are often too general for the different kinds of application. Even when this is true, they are very useful as a basis for this kind of evaluation. Another set of heuristics were self-build which means that the author wrote the heuristics on Chapter 2 in order to make them more specific for evaluation. In the evaluation phase they were used to identify problems with the artifact but while using them a few problems arise. Number one problem was that some of them were very similar to Nielsen's heuristics and as such there was duplication of evaluation. Another issue was that there was some duplication also, between different heuristics in the self-built heuristics. For example, the third heuristic is meant to evaluate how well the application matches with physical panel wall and the fourth heuristic was about the user ability to control the physical devices with the application. There was a slight differentiation in these in such a way that the third one was supposed to be exactly about the physical wall application match but the fourth one was about giving the user as much control of the physical devices as possible. However, these could have been combined. The idea was that there could be more precise evaluation done but after the evaluation was done it was obvious, they could have been combined.

Another issue with the heuristics was that they were specifically meant to evaluate an actual application and with it the physical panel wall interaction. The plan as an artifact however is different from actual application since there is no way to test a plan in actual physical panel wall. The self-build heuristics should have been built so that they would have been more about evaluating a plan rather than an actual application. Otherwise the evaluation worked well and some problems with the plan were found and, the plan was reiterated based upon these problems. However, it is better next time when building heuristics to consider the demands of the actual artifact instead of just thinking about the heuristics in a more general way.

There were also in the theory for the heuristics in Chapter 2 some similar application examples. First examples from Mowad et al (2014) and Pivare (2013) where about how to build this kind of mobile control application for smart phones. They were used in Chapter 4 as basis for the plan, but they had some technical details that were not included in the plan. This is because implementation was not the point of the plan, but the plan was made without implementation details. This meant that parts of these examples were unusable for the plan. However, reading these articles did give some basis for how to work with mobile control applications which the acoustic panel application basically is. There is not much need for these two example applications to be discussed due to the nature of them.

However, the application evaluation of Inal (2018) is much more interesting from the point of view of this work. The application was not a mobile control application like the acoustic application is but there were heuristics used in order to evaluate a mobile doctor appointment system. Most of the errors in that system that were found were about error prevention and freedom. This article was used as an example of how to perform heuristic evaluation on a system. In this article experts from the field of information systems engineering were doing an application evaluation. Unlike Inal (2008) in this acoustic application work the author did the evaluation by himself without outside help so there was a difference there. In Inal (2018) one of the issues was how easy the application was to use, and the results suggested that usability was very important for users. Mostly users wanted an application with ease-of-use but were happy when there was an application in the first place rather than no application. Most of the usability problems in the Turkish mobile health application were found to be about error prevention and diagnosis, and documentation and help function of the application. For this acoustic application plan the evaluation had similar results with problems with documentation and error prevention. There were of course other problems as well. This article of Inal (2018) was mostly used for studying how a heuristic evaluation of a mobile application was done. It was one of the studies that also was used to determine the self-built heuristics for the evaluation done in this work.

For developing the plan as an artifact design science principle were used which are described in Chapter 3. The main article used for these principles was Hevner et al (2004). First principle in this article was about designing the artifact. In this work the artifact designed was a plan for an acoustic application so this work would be construct in the design science point of view. For the problem relevance-based finding information on mobile control applications was a hard project because there were not that many studies done about them. This was especially true when using databases to find articles about heuristic evaluation of this kind of work. This means that the problem is very relevant in the current situation and more evaluations of this kind of application must be done. Third principle was about evaluation and this was done with two sets of heuristics by the author. This provided good results about how the application could be made better and in the re-iteration phase new plan was made for these details.

The fourth guideline is about research contributions. There are some contributions of this work to software engineering design science. First is the artifact that was built creates a contribution. Second contribution comes from the fact that according to literature review this type of mobile control applications and evaluation of them with heuristics have not been done in literature that much. So, this work will represent one contribution to both building a plan for the application and using heuristics to evaluate it. Third contribution of this work is the new self-build heuristics that were used. There were problems with these heuristics, and they could have been done better but they can be used as one basis for evaluation of mobile control applications. These are all important contributions to knowledge base and create clear research contributions.

The research rigor is the next guideline. Rigor comes from using rigorous methods for constructing and evaluation of artifact. It must be said that for the plan the methods used to make it were not rigorous enough. There were some basic guidelines used for how to plan for a software, but more rigor would have been better. The evaluation with heuristics was done rigorously. Each heuristic was used for evaluation and based upon that usability problems in the plan were found. Second iteration of the plan discussed these problems. This process was much more rigorous compared to making the first plan. Also, this process has both generalizability and applicability for later projects. The plan did not use as much knowledge base or foundations in research when compared to heuristic process and this could be said to be a problem with rigor in this research.

The sixth guideline which is about design science as a search process was filled here by iterating the plan after the heuristic evaluation. In this way the original plan was a kind of first step and the heuristic evaluation was used to search for problems in it. The next step of iteration of a plan based upon results of this heuristic evaluation was the goal of the search process and in this way the design science as a search process guideline was fulfilled.

The last guideline is about communication of research which is of course done with this work. On this work there is not much need to change the communication between managerial and technical audience since it is done as a thesis project and it is meant to communicate at the same to all the possible audiences. This work should not be hard to understand by either audience. There is detail but at the same time the details are not too technical since there are no implementation details done in this work. The problem is relevant since there is not much research on similar topics and the solution after second iteration is effective to solve the problems.

The research problem presented in Chapter 3 was how to make a mobile control application that is easy to use by non-expert users when it comes to controlling acoustic panels. This research problem is answered in the artifact. The artifact provides a user interface for the application that is simple to use and intuitive. All the buttons except for the panels selection clearly state for what it is used. The only confusing thing for a non-expert user can be in the error prevention. When the panel that has error is shown in red it is quite clear to see where the panel with the error is in the physical wall and the status monitor will have to provide an error message to tell what the error is. This error message should be made as clear as possible but there can be a need for technical details that are hard for non-expert user to understand.

The research questions for this work are as follows:

RQ1: How to create an efficient and usable mobile application for controlling acoustic panels?

RQ2: Based upon heuristics, what are the usability problems of this application?

RQ3: How, based upon heuristics, to improve the application operation for non-expert users?

The RQ1 which is about how to create this kind of application is answered in the plan. The plan explains in detail the requirements for the application and shows in Figure 2 the detailed user interface that needs to be implemented. The application is created by first making sure the user interface has all the information needed for the requirements of the application. There are in the plan steps taken to create this application and usability of it was discussed in Chapter 5 while making the heuristic evaluation. The point is to learn how to make this application efficient and usable. Efficiency comes from the fact that this is a simple application for basically one task. The usability is discussed in the heuristic evaluation. The re-iteration based upon heuristics did show the need to add some error prevention methods and documentation to the application, but simplicity was the main point of the application to create an efficient and usable application.

The RQ2 about usability problems of the application based upon heuristic evaluation is important. The heuristic evaluation found problems especially with error prevention, documentation and positioning the panels so that the physical configuration of the panels and the application panel view correspond to each other. The problem of error prevention was solved by adding a panel colour of red when there is a problem with a panel and status monitor on the downside must show details of the error if possible, such as no connection. Another problem was lack of documentation and dialogue in the application and this was solved by adding a pdf documentation of functionalities and possible error message that the status monitor provides. This documentation should help identify problems and for a new user to give basic instructions on how to use the application. However, as said before, the application was designed so that it would be simple and intuitive and documentation need would be as low as possible. Third issue of panel positions and corresponding panel positions of the actual physical wall and the application was solved in a way that panel wall should send information about all the panel positions to the application and the application would give this information to the user in the panel button. It is true that there already are in the panel wall a way to send some panel position information such as x position and y position but there could be added information such as is the panel on left wall or right wall or ceiling. This is left for the implementation to decide.

The RQ3 is about how based upon heuristics evaluation application operation could be improved for non-expert users. This is answered in the Chapter 5 where the evaluation is done. To summarize the application should have error prevention, documentation, and better panel position information. There were other issues also discussed but these were the main ones from the usability point of view. Error prevention is important to all the users to make sure they notice problems with the wall or the application. Documentation should be done to help new users use the application and explain to users' error messages and how to deal with them. Also, clarity in panel positions is important as mentioned before.

In Hevner (2007) there are design cycles introduced. These are important from the perspective of this research since the report follows these cycles. The first one is relevance cycle. This means application domain of the work with people and organization systems and technical systems and problems and opportunities. From the application domain side organizational system is not that applicable for this work but people who use this application are. The idea of this application was to make it as user friendly and efficient as possible. The organizational system was not considered because the application is not a big one that needs this context and because the organizational system was not known. Technical system perspective was also not totally considered since the plan required no implementation of the application. This means that technical details are left to those who implement it. However, some details from a technical perspective were considered and the environment where the application works from technical perspective was discussed but this discussion did not go deep into implementation details. Problems of the design were found in the heuristic evaluation phase. Opportunities are from the actual implementation of the application. There is also in the relevance cycle the idea that the artifact should be field tested with acceptance criterion. However, because the artifact here is a plan for an application this kind of field testing cannot be done.

The rigor cycle encompasses foundations that are theories, methods, experiences, and expertise and design products and processes. This cycle draws from the knowledge base when it comes to foundations. The theories that were used here were about design science and about heuristics. This knowledge base was used to create the plan and create heuristics to evaluate the plan and then reiterate this plan. This means that the for the rigor cycle there was a good amount of theory used and it was reiterated to fit this project.

The last issue of design science was the design cycle that was applied in this project by making the original plan and evaluating it with heuristics. After this the plan was reiterated. This cycle was done once through and the results were written into the heuristic Chapter 5 of iteration of the plan. The rest of the information about design science research written in Chapter 3 were applied depending on how well it fit this research.

When constructing the original artifact for this work it was done by following basic design research guidelines. The first issue was to design the user interface which luckily came from the original order from a company that needed this application. This need was later cancelled but the original requirements were still there. The plan then identified requirements, use cases and functionalities. The basics of this was that there were a few technical requirements for the plan such as working through Wi-Fi or the way the basic interface was built. This plan was done according to original requirements of the application. The usage of the application was described with a use case diagram. The function of the application was described through activity, class, and sequence diagrams after which the application was analysed, and its functionality described. There was no need to provide implementation details since the artifact is the plan not the implementation of the application.

Next issue was evaluating this plan based upon heuristics. One of the issues with this is that the author did the evaluation. It might have been better if done by another person. However, there were problems found in the heuristic evaluation and after this evaluation a second iteration of the plan was made. This time no diagrams and such were used mostly because there was no need for it. The writing of the new plan was enough. The new plan had only added functionality and therefore the plan in Chapter 4 is still the basis even after the heuristic evaluation, but some functionalities were added to it in the iteration subchapter of Chapter 5. When implementing the application these additions must be considered.

The limitations of this work are discussed next. One limitation is that there was no prototype or application implementation which means that the application could not be tested with a real physical panel wall. This means that the application can have issues when it is implemented which were not realized in the plan. One of the issues that might become important is scalability, which might change, in the actual implementation of the application. It is not certain that the way in which the application was planned is the best in actual usage.

Another limitation is related to self-built heuristics since these were not built by an expert in the field of heuristics and it can be that they did not take into account all the possibilities that might exist in the usability of the artifact. Also, the heuristics testing was done by the author who is not an expert in this kind of testing even having worked in some university student projects that use them. This means that an expert might have found some more faults and problems with the artifact.

Another limitation comes from the application being relatively simple and straightforward. This is a limitation because if the application had been larger with more functions it might satisfy issues such as rigor and relevance cycle better.

Also, there is a limitation with the implementation details. It was decided that these details do not belong in the artifact due to it being a plan that could be implemented in many ways. This is a good thing but at the same time if one implementation way was chosen there might have been more details to help with this implementation.

Another limitation is that iteration for the application was done only once based upon heuristics. The issue with this is that if there were more iterations the plan could have become more robust. However, only one iteration was done due to limited space and time. Multiple iterations might have been better, but the scope of this project it was not feasible.

One last limitation is the tools used to make the plan. This means written text and diagrams. If diagrams were more detailed there could be more information about this plan. Also, text was chosen as the way to communicate details about iterating the artifact. This communication did use one figure but on the scope of this work it was decided that additional figures would not be feasible.

7. Conclusions and Future Research

This work was done to find out how to best plan an acoustic application for the controlling of acoustic walls. These walls work in such a way that they can be configured in different positions depending on the needed acoustics. This application is meant to be used in controlling these panels. The plan for this application was the artifact for this thesis work. The application was originally ordered by a company that later had to pull out from the project and that is why implementation of the application was not done.

Main issues for this was to make a plan that could be used to create a working application. The application must be relatively simple to use for non-expert users while at the same time having enough functionality to do the job it was meant to do. This plan was kept simple for this reason. There was no need for too many functionalities in the application. One of the main issues in planning the application was scaling of the panels. This was decided to be done so that one panel view would have four vertical and five horizontal panels, so that there would be twenty panels shown at the same time. The panels would be divided to view groups based upon this limitation. If panel amount is less, than twenty then only that amount would be shown. Another issue is that there should be minimum panel size in case the application was to be used in mobile phone as an example. If the panel view becomes too small, usage can become impossible and this why there is a minimum panel size.

Other issues with the plan were the data coming from panels, moving panels, and connecting to panels. These were all solved. The heuristic evaluation was done to find out how well the application met heuristic principles. The issues found were about error prevention, documentation, and dialogue inside the application. These were solved by introducing error prevention methods, documentation method and some basic dialogue in the application.

Overall, the plan now meets the requirements that were given in the beginning. Future research based upon this artifact is implementing the artifact. The plan can be used as a basis for this implementation but there is also a need for implementation details to be added to the plan. Another, future topic based upon this is improving the heuristics that were self-built because though they identified problems in the research, they could be more extensive and specific to mobile application. Of course, in the field of mobile control applications there is still a lot of need for different kinds of research projects and evaluation of applications. It was hard to find any research that used heuristics to evaluate mobile control applications and therefore this is one of the future research topics.

References

Adams, L. Courtney, J. (2004) Achieving Relevance in IS Research via the DAGS Framework. Proceedings of the Thirty-Seventh Annual Hawaii International Conference on System Sciences. IEEE Computer Society Press.

Anganes, A., Pfaff, M., Drury, J., Toole, C. (2016). The Heuristic Quality Scale. *Interact. Comput.* 28(5), 584-597

Applegate, L. (1999). Rigor and Relevance in MIS Research – Introduction. *MIS Quarterly.* 23: 1, 1-2

Archer, L. B. (1984) Systemic Method for Designers. Development in Design Methodology. 57-82

Bevan, N., Carter, J., Harker., S. (2015). ISO 9241-11 revised: What have we learnt about usability since 1998. *Human-Computer Interaction: Design and Evaluation, Lecture Notes in Computer Science.* Springer 143-151.

Boland, R. (2002). Design in the Punctuation of Managed Action. *Managing ad Designing: Creating a Vocabulary for Management Education and Research. Frontiers of Management Workshop.* Available online at: http://design.cwru.edu

Choi, W., Kannan, J., Babic, D., (2018). A Scalable, Flow-and-Context-Sensitive Taint Analysis of Android Applications.

Cole, R. Purao, S. Rossi, M. Sein, M. K. (2005). Being Proactive: Where Actuin Research Meets Design Research. Twenty- Sixth International Conference on Information Systems. 325-336

Cooper, R. G., (1990). Stage-Gate Systems – A New Tool for Managing New Products. Business Horizons. 33, 3, 44-54

Dennings, P. (1997). A New Social Contract for Research. *Communications of the ACM.* 40:2, 132-134

Franklin, F., Breyer, F., Kelner, J. (2014). Heuristicas de Usabilidade Para Sistemas Colaborativos Remotos de Realidade Aumentada. *Proceedings of XVI Symposioum on Virtal and Augmented Reality.* 53-62

Gregor, S., Hevner, A. (2010). Introduction to Special issues in Design Science. Inf Syst E-Bus Manage 9, 1-9

Hermawati, S., Lawson, G. (2016). Establishing Usability Heuristics for Heuristics Evaluation in a Specific Domain: Is There a Consensus? *Applied Ergonomics.* 56, 34-51.

Hevner, A., Ram, S., March, S. (2004). Design Science in Information System Research. MIS Quartely. 28 (1), 75-105.

Hevner, A. (2007). A Three Cycle View of Design Science Research. Scandinavian Journal of Information Systems. 19 (2), 4.

Hevner, A., Chatterjee, S. (2010). Design Science Research in Information Systems. Integrated Series in Information Systems. 22.

Hickey, A. M. and Davis, A. M. (2004). A Unified Model of Requirements Elicitation. Journal of Management Information Systems. 20, 4, 65-84

Hub, M., Capkova, V., Heuristic Evaluation of Usability of Public Administration Portal, DEO. *Applied Computer Science, Proceeding of International Conference on Applied Computer Science, ACS.*

Iivari, J. Hirschheim, R. Klein, H. K. (1998). A Paradigmatic Analysis Contrasting Information Systems Development Approaches and Methodologies. Information Systems Research. 9, 2, 164-193

Iivari, J. (2007). A paradigmatic Analysis of Information Systems as a Design Science. Scandinavian Journal of Information Systems. 19(2)

Inal, Y. (2018). Heuristic-based User Interface Evaluation of the Mobile Centralized Doctor Appointment System – A Case Study. The Electronic Library. 37 (1).

Inostroza, R., Rusu, C., Roncagliolo, S., Rusu, V., Collazos. (2016). Developing SMASH: A Set of Smartphone's Usability Heuristics. *Comput. Stand. Interfaces.* 43, 40-52.

ISO 9241-11. (1998). Ergonomic Requirements for Office Work With Visual Display Terminals – Part 11. *Guidance on Usability, International Organization of Standardization, Geneva.*

ISR. (2002). Editorial Statement and Policy. *Information Systems Research.* 13:4

Jaferian, P., Hawkey, K., Sotirakopulos, M., Velez-Rojas, M., Beznosov, K. (2014). Heuristics for Evaluating it Security Management Tools. *Human computer Interaction.* 29 (4), 311-350

Jaspers, M. (2009). A Comparison of Usability Methods for Testing Interactive Health Technologies: Methodological Aspects and Empirical Evidence. *International Journal of Medicine Information* 78 (5), 340-353.

Järvinen, P. (2007). "Action Research is Similar to Design Science". *Quality & Quantity.* 41, 37-54

March, S. Smith, G. (1995) Design and Natural Science Research on Information Technology. Decision Support Systems. 15, 4, 251-266

Mowad, M., Fathy, A., Hafez. (2014). Smart Home Automated Control System Using Android Application and Microcontroller. *International Journal of Scientific & Engineering Research.* 5 (5), 935-939

Nielsen, J., Molich, R., (1990). Heuristic evaluation of user interfaces. *Proceedings of ACM CHI'90 Conference.* 249-256.

Nielsen, J, (1995). 10 Usability Heuristics for User Interface Design. From http://courses.ischool.utexas.edu Retrieved: 30.01.2020

Nunamaker, J. F. Chen, M. Purding, T. D. M. (1991). Systems Development in Information Systems Research. Journal of Management Information Systems. 7, 3, 89-106

Lechner, B. Fruhling, S., Petter, S., Siy, H. (2013). The chicken and the Pig: User Involvement in Developing Usability Heuristics. *Proceedings of the Nineteenth Americas Conference on Information Systems.*

Orlikowski, W., Lacono, C. (2001). Research Commentary: Desperately Seeking the IT in IT Research – A Call to Theorizing the IT Artifact. *Information Systems Research* 12:21 121-134

Peffers, K., Tuunanen, R., Rothenberger, M., Chatterjee, S. (2008). A Design Science Research Methodology for Information Systems Research. Journal of Management Information Systems. 24 (3), 45-77.

Piyare, R., Lee, S. (2013). Smart Home-Control and Monitoring System Using Smart Phone. Proceedings, The 1st International Conference on Convergence and It's Application. 24. 83-86.

Piyare, R. (2013) Internet of Things: Ubiquitous Home Control and Monitoring System Using Android Based Smart Phone. International Journal of Internet of Things. 2(1), 5-11

Razak, M., Anuar, N., Salleh, R., Firdaus, A., Faiz, M., Alamri, H. (2019). "Less Give More": Evaluate and Zoning Android Applications. Measurement. 133, 396-411.

Rusu, C., Ronzagliolo, S., Rusu, V., Collazos. (2011). A Methodology to Establish Usability Heuristics. *Proceedings of the Fourth International Conference on Advances in Computer-Human Interactions ACHI2011.* 59-62

Simon, H. (1996) The Sciences of Artificial. MIT press, Cambridge, MA.

Sim, G., Read, J., Cockton, G. (2009). "Evidence Based Design of Heuristics for Computer Assisted Assessment." *Human-Computer Interaction INTERACT,* 204-216.

Scholtz, J. (2004). Usability Evaluation. *National Institute of Standards and Technology.*

Quinones, D., Rusu, C. (2017). How to Develop Usability Heuristics: A Systematic Literature Review. Computer Standards & Interfaces. 53, 89-122

Quinones, D., Rusu, S., Roncagliolo, V. (2014). Redefining Usability Heuristics for Transactional Web Applications. *Proceedings of the 11th International Conference on Information Technology.* 260-265.

Quinones, D., Rusu, C., Rusu, V. (2018). A Methodology to Develop Usability/User Experience Heuristics. Computer Standards & Interfaces. 59, 109-129

Tsichritzis, D. (1998). The Dynamics of Innovation. *Beyond Calculation: The Next Fifty Years of Computing. Copernicus Books, New York.* 259-265

Van Greunen, D., Yeratziotis, A., Pottas, D. (2011). A Three-phase Process to Develop Heuristics. *Proceeding of the 13th ZA-WWW Conference, Johannesburg.*

Venkatesh, V. (2000). "Determinants of Perceived Ease of Use: Integrating Control, Intrinsic Motivation, and Emotion into the Technology Acceptance Model." *Information Systems Research.* 11 (4), 342-365

Vessey, I., Glass, R. (1998). Strong Vs. Weak Approaches to Systems Development. *Communications of the ACM.* 41:4, 99-102

Walls, J. Widmeyer, G. El Sawy, O. (1992) Building an Information System Design Theory for Vigilant EIS. Information Systems Research. 3, 1, 36-59

Yeratziotis, D., Pottas, D. Van Greunen. (2012). A Usable Security Heuristic Evaluation for the Online Health Social Networking Paradigm. Int. J. Hum. Comput. Interaction 28 (10), 678-694

Zmud, R. (1997). "Editor's Comments." *MIS Quartely.* 21:2