Sanni Partanen

User Requirements and User Interface Design for Virtual Patient Simulator

Helsinki Metropolia University of Applied Sciences

Bachelor of Engineering

Health Informatics

Thesis

1.12.2016

Tiivistelmä

Tekijä(t)	Sanni Partanen
Otsikko	Vaatimusmäärittely ja käyttöliittymäsuunnitelma virtuaalipotilassimulaattorille
Sivumäärä	36 sivua + 4 liitettä
Aika	1.12.2016
Tutkinto	Insinööri (AMK)
Koulutusohjelma	Hyvinvointiteknologia
Suuntautumisvaihtoehto	Hyvinvointiteknologia
Ohjaaja(t)	Sakari Lukkarinen, Lehtori, Metropolia Ammattikorkeakoulu Willem Gorter, Senior Clinical Specialist

Tässä työssä luotiin tilaajalle käyttäjävaatimukset sekä käyttöliittymäsuunnitelma virtuaaliselle potilassimulaattorille. Työn tilaaja kehittää ja valmistaa potilasvalvontamonitoreja sairaalakäyttöön. Virtuaalista potilassimulaattoria käytetään toistamaan aitoa potilasdataa potilasmonitorille samalla, kun käyttäjä voi tutkia dataa ja sen annotointia. Oikeaa potilasdataa tallennetaan ja annotoidaan samanaikaisesti, kun potilasmonitoria testataan sairaalaolosuhteissa.

Käytettävyystutkimus sekä fokusryhmähaastattelu tehtiin selvittämään loppukäyttäjien mieltymyksiä ja toiveita tulevalta ohjelmalta. Yhteensä viisi tilaajayrityksen mahdollisista loppukäyttäjistä osallistui selvitykseen. Loppukäyttäjät ovat sairaanhoitajataustaisia kliinikoita. Käytettävyystutkimuksessa pyrittiin paperiprototyypin avulla selvittämään, kuinka käytettävänä kliinikot kokivat käyttöliittymäsuunnitelman. Fokusryhmähaastatteluissa koitettiin syvemmin perehtyä kliikoiden vaatimuksiin ohjelmalta.

Käyttäjävaatimukset jaoteltiin selvityksen perusteella kahteen ryhmään: vaadittaviin ja toivottaviin ominaisuuksiin. Vaadittavat ominaisuudet määrittelevät ohjelman perustoiminnot, joita siltä odotetaan. Toivottavat ominaisuudet sen sijaan määrittelevät toimintoja, jotka hyödyttäisivät käyttäjiä ja lisäisivät ohjelman käyttömukavuutta. Kaikki vaatimukset on luokiteltu fokusryhmissä niiden tarpeellisuuden mukaisesti. Vaatimuksista ja käyttöliittymäsuunnitelmasta luotiin dokumentti, joka luovutettiin tilaajayrityksen käyttöön.

Virtuaalinen potilassimulaattori on osa työn tilaajan kehittämää laajempaa projektia, jonka tavoitteena on tallentaa ja toistaa aitoa potilasdataa. Virtuaalisen potilassimulaattorin lisäksi oleellinen osa projektia on annotaatiotyökalu, jolla annotoidaan nauhoitettavaa potilasdataa sairaalassa. Tämän insinöörityön kanssa samanaikaisesti tehtiin Tahiti Konttisen ja Pekka Poikolaisen insinöörityö annotaatiotyökalun käyttäjävaatimuksista. Yhteistyötä tehtiin insinöörityöntekijöiden kesken virtuaalipotilasprojektin tavoitteiden saavuttamiseksi.

Avainsanat	potilasmonitori, käytettävyystutkimus, GUIDe-prosessimalli

Author(s) Title Number of Pages Date	Sanni Partanen User Requirements and User Interface Design for Virtual Patient Simulator 36 pages + 4 appendices 1st December, 2016
Degree	Bachelor of Engineering
Degree Programme	Health Informatics
Specialisation option	Health Informatics
Instructor(s)	Sakari Lukkarinen, Senior Lecturer, Metropolia University of Applied Sciences Willem Gorter, Senior Clinical Specialist

In this thesis, user requirements and a user interface design for a virtual patient simulator were created for a company that manufactures patient monitoring devices. A virtual patient simulator is used for replaying recorded patient data to a patient monitor, while observing the data and its annotations. Real patient data is collected and annotated while testing the monitors in hospitals. The use of real patient data brings the in-house testing a step closer to the real hospital environment.

Usability study and focus group interviews were used to find out the end users' preferences and desires. In total five possible end users participated in the research. The end users are clinicians with backgrounds in nursing. The usability study was used to chart the clinicians initial thoughts on the new software and to test the usability of the user interface design. The focus groups were used for more thorough investigation on what the clinicians truly wanted in the software and what features would ease their work the most.

The user requirements were divided into two groups: mandatory and desirable requirements. The mandatory requirements describe what is essential for the software, and the desirables what would make the use more efficient and pleasant for the users. The requirements were prioritised in the focus groups, but the final decision on what to implement first was left for the company to decide. The requirements and the suggestions for user interface were gathered in to a document which was handed over to the company.

Virtual patient simulator software is a part of a larger project at the company. The project aims at collecting and playing back real patient data. In addition to the virtual patient simulator software, the project also consisted of the annotation software that will be used to annotate the recorded patient data at the hospital. Another thesis was carried out simultaneously with this thesis by Tahiti Konttinen and Pekka Poikolainen, and their thesis focused on the user requirements for the annotation software. The thesis workers cooperated with each other to benefit all parties.

Keywords	patient monitor, usability study, GUIDe process model

Contents

List of abbreviations

ECG	Electrocardiogram.
EEG	Electroencephalogram, a parameter describing the electrical activity of the brain.
IBP	Invasive Blood Pressure.
ISO	International Organisation for Standardisation.
NIBP	Non Invasive Blood Pressure.
SpO2	Measures the arterial oxygen saturation of hemoglobin at a peripheral measurement site, an estimate of oxygen in the blood.
UI	User Interface.
URS	User Requirement Specification.

1 Introduction

This thesis is a part of a larger project at the company, which aims at developing a virtual patient simulator for better and more realistic testing of the patient monitors. The finished virtual patient simulator will repeat real recorded patient data along with annotations. The main reason for developing a more realistic patient simulator is securing patient safety: the more realistic and varying the test data is, the more thorough the testing can be. It is therefore possible to find more defects and thus make patient monitors safer for patients.

This thesis was carried out for a health care device manufacturer. The company produce patient monitors which are one of the most critical medical devices in the modern society. Thus, they have to be tested thoroughly so they can be trusted to work accordingly in life-threatening situations. Currently, the company is conducting testing for the monitors in the company's in-house laboratory as well as in the hospital on real patient data. Unfortunately, testing can not be conducted in the hospital as much as the company would like to. Instead, while they are testing the monitor, they are also recording the patient's data. So far the company has collected hundreds of hours of Electrocardiogram (ECG) data from different patients in different situations. What makes the real data collected from patients so valuable for testing, is that it is unpredictable and contains different artefacts. Artefacts are disturbances in the parameters that can be caused by movement or different medical procedures. Artefacts can be so disparate that they are hard to produce by simulation. While recording the data is important, it is also necessary to annotate it on the spot. Annotation tells the testers which alarms were real and which were false and what action possibly caused the false alarm.

Currently, the customer has an outdated software with limited functionality that can play back recorded data to a patient monitor. While the data is valuable, the clinicians in charge of testing are not using the annotation documents to see what the files contain. They use the data files without knowing how long the files are or what events they contain, because the annotations are on a separate paper document, not with the recorded data.

The aim of the thesis is to define the user requirements and propose a design for the user interface of the virtual patient simulator. The GUIDe process model [1] and usability testing will be used for the design phase of the user interface. The end users will be interviewed using the focus group method [2]. Even though the data collecting software and possibly the simulator will use and show four parameters, this thesis will cover only ECG, which is one of the most commonly used parameters.

2 ECG

ECG is the measurable electrical activity of the heart. It is one of the five vital signs that are commonly monitored in the hospital on a patient monitor; it is usually the first one the clinicians use to assess the patient. ECG can be measured with three, six or twelve leads, with twelve leads being the conventional setup of ten electrodes. Initially ECG was used to study arrhythmias, but now it can also be used to measure the heart rate, the size and orientation of the heart, the presence of any damage to the heart's muscle cells or conduction system, the effects of cardiac drugs, the function of implanted pacemakers and even respiratory signals [3; 4].

At every beat, the heart depolarises to trigger a contraction to pump blood. This transmits an electrical activity throughout the body that can be picked up on the skin with an electrode. Through the electrodes the signal can be displayed graphically on a monitor or a screen. The conventional setup of the ECG is the 12-lead setup, where the electrodes go one to each limb and six to the chest. The electrodes on the limbs can go anywhere within the limb, but the chest electrodes have predefined and specific places near the fourth rib. The limb leads are used to get a vertical plane on the heart and the chest leads to get a horizontal plane. When combined these planes make a three-dimensional electrical picture of the heart. [5]

The different signals received from any one combination of electrodes are named leads. As in the conventional 12-lead setup, there are twelve signal leads and ten electrode leads. From the limb leads, right leg's electrode acts as the ground, so that leaves only three electrodes to construct the leads from. Lead I is constructed by comparing the left arm as positive and the right arm as negative, while the zero point is at the center of the lead. Lead II compares the left leg's negative to the right arms positive. Lead III connects the left leg's positive the left arm's negative. Each lead portrays the heart's electrical activity from a certain point of view to collectively create the vertical plane. The limb leads are also used to create augmented unipolar leads named aVR, aVL and aVF, which help to increase the signal strength of the heart and give extra views from different angles on it. The augmented unipolar leads are constructed by bringing two of the three electrodes tied together to ground. The last six leads are unipolar and come from the electrodes attached to the chest. They are named V1, V2, V3, V4, V5 and V6 from left to right. [5]

Artefact is a disturbance in the signal. Common ECG artefacts include: Motion artefact, caused by the patient moving and the skin stretching; muscle artefact, generated by skeletal muscles; electrostatic artefact, caused by electrostatically charged person moving near the patient or the ECG device; poor contact artefact, caused by dried gel or poor adhesion in the electrode; electromagnetic interference, generated by items like cell phones or radios; implanted stimulators, like pacemakers. Generally low number of artefacts are mostly harmless, but at worst they can trigger false alarms in the patient monitor. False alarms instead can lead to desensitisation of the caregivers, along with slowing response times and decrease in the quality of care for the patient [6]. Through testing the monitors with real data, the algorithms of the monitor that are responsible of raising an alarm can be taught to not react to certain type of artefacts. But it is very important that the data is annotated, so the tester can know when an alarm is, in fact, a false one. [7]

3 Patient Monitor Testing

This chapter presents what patient monitors are and what is needed for testing them. The virtual patient simulator project is also introduced in this chapter.

3.1 Patient Monitors

Patient monitors are used to continuously monitor patients' physiological parameters or to repeatedly perform medical tests. The most commonly monitored parameters are heart rate, heart rhythm, arterial blood pressure, respiratory rate and blood oxygen saturation. For effective patient care, bedside monitors are used – increasingly by non-invasive sensors – to collect, display and store physiological data. Patient monitoring are used routinely on general hospital wards, for diagnostic purposes in the emergency room, or for therapeutic purposes in the operating room to help make accurate and prompt decisions. Nowadays, bedside monitors can even be used in some situations by patients in their own homes. [8, 561]

Patient monitors are often considered as something that watch for and warn about serious or life-threatening events in patients. But they can do more than just alert caregivers of potential life-threatening events, since they can also control connected life support devices using the physiological input data and provide guidance for care. [8, 561-562]

Patient monitors are medical devices that consist of one or more sensors, processing components, a display device and communication links for displaying or recording the results elsewhere in the hospital through the monitoring network [9]. The monitor is usually located near the patient's bed, since the sensors are connected to it by wires. There it can also provide the caregivers with the up-to-date information they need.

There are at least three situations where patients need physiological monitoring:

1. Patients with compromised physiological regulatory systems, like suppressed respiratory system by drug overdose or anesthesia.

2. Patients with a suspected life-threatening conditions that could suddenly change to life-threatening, such as patients with an indication of acute heart attack; patients who have recently had open-heart surgery; fetus during labor and delivery.
3. Patients in a critical physiological state, such as patients with septic shock or multiple trauma. [8, 564]

Patient monitors have alarms for the different arrhythmias of the heart. Arrhythmia is used to describe an abnormal heart rhythm. Some arrhythmias can be totally harmless, but some can be serious, such as asystole, extreme bradycardia, extreme tachycardia, ventricular tachycardia and ventricular fibrillation/tachycardia are life-threatening alarms in most patient monitors. [6]

3.2 Patient & Vital Sign Simulators

Patient simulators are used in medical training and clinical testing. Simulators are often used to explore situations which would otherwise be too dangerous, time-consuming or rare to experience in real life. There are different kinds of patient simulators ranging from virtual patients to life sized manikin patients to mere box-looking vital sign simulators. Each of these simulators has their own place in making healthcare safer, but for this thesis the focus is on virtual patient simulators and vital sign simulators.

Virtual patient simulators are used as hands-on practice for healthcare trainees. The simulators use virtual patients that are computer generated and are able to demonstrate realistic clinical cases. The simulators can be used to provide varied and standardised training to the trainees. Simulations are also beneficial for patient safety, because the students do not have to practise on real patients. If necessary, they can practice through repetition in safe boundaries before interacting with real patients. [10, 16]

Vital sign simulators are used to simulate different kinds of patient signals one or multiple at a time. Simulated vital signs include ECG, respiration, temperature, IBP, NIBP, cardiac output and SpO2. These are used to test and verify basic operations of patient monitoring devices with realistic, but simulated patient data. Some simulators can even include artefacts in the signals if wanted, but usually the signals are very homogeneous and even. [11]

3.3 Testing

Testing is done to find errors or unexpected behaviour; to raise the quality and reliability of the software. The found errors are usually called defects. One important aspect of testing is to constantly check that the output matches the intended. [12]

Testing is one of the most important steps in profitability. Companies that carefully test their products make better profit than companies that poorly test their products. The difference in profitability is emphasised when you also consider the customer dissatisfaction and caused bad reputation. Properly done testing can save money early on, since fixing a mistake while designing the program costs one to two percent of what it would cost after publishing the product. [12]

When choosing what data to use to test a system, one needs to remember that testing is done to see if the system works and to check that it does not break. This is extremely important when dealing with medical devices, which are responsible for patients' lives. Usually three types of test data are used, as seen in Figure 1: normal data values, extreme data values and abnormal data values.

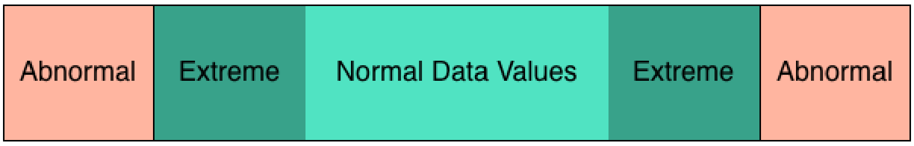


Figure 1: Figure describing test data values. [13]

Normal data values are data that would normally be entered into the system. Extreme data values are still normal data, but the values are at the absolute limits of the normal range. Abnormal data values are data that should not normally be accepted by the system. Abnormal values should be rejected by the system and they're used to see that the system does not break from invalid input. [13]

When testing patient monitors, normal data values are standard heart rate while the extreme values are the different conditions and arrhythmias of the heart. Abnormal data values can be data that a human can not possibly produce. Normal data and some of the extreme data is easily fabricated by vital sign simulators, but the possibility for extreme data values is endless. This is where virtual patient simulator's recorded patient data would come in handy.

Regression testing is a type of software testing that is done to confirm that the system still performs correctly after a change is made. The most important feature of regression testing is to verify that already fixed defects do not resurface and no new defects appear. [12]

The company's most widely used testing methods are manual and automated regression testing and exploratory testing. While engineers focus on manual- and automated testing, the clinicians work on exploratory testing. Exploratory testing is exploring different options and features of the system to see that everything works and trying to find bugs. Exploratory testing is simultaneously learning, test design and test execution; the tester actively tries to create better and more accurate tests based on what they have learned of the software's capabilities. This way of testing takes advantage of the clinicians' varying medical backgrounds. True exploratory testing is completely non-scripted, but for efficiency reasons test sessions can be set themes or certain aspects of the software to concentrate on. [14]

The clinicians are doing testing at the in-house laboratory and in hospitals. In-house laboratory testing is done with simulators, vital signs measured from oneself and using the recorded ECG data. In the hospital, the clinicians record and annotate data simultaneously while testing. If it were possible, the company would like the clinicians to do testing with live data in the hospital constantly. As a good alternative, the clinicians should be able to replay the hundreds of hours of collected data using the virtual patient simulator for more efficient testing in the laboratory. With recorded data, verifying and demonstrating defects is easier.

3.4 Virtual Patient Simulator Project

The customer company has been collecting data actively from different kind of patients in different hospitals, resulting in hundreds of hours worth of data featuring different parameters, such as ECG and pulse oxygen saturation. When recording the data, a clinician has annotated important events, such as patient standing up and different procedures, to a paper document. The annotations are linked to the data files by time of occurrence and they are not currently exploited in the clinicians' daily work.

The stakeholders of the project consist of the thesis workers; clinicians who are the end users; project manager; senior clinicians, who instructed the thesis workers; and engineers who will implement the final product and use it in software testing. Weekly meetings were held within the framework of the thesis with the stakeholders. The meetings discussed the progress of this thesis, and Tahiti Konttinen and Pekka Poikolainen's thesis on patient data annotation tool.

Virtual patient simulator will be a computer software capable of playing recorded patient data, such as ECG. The data can then be fed into the patient monitor under testing at the customer company's in-house laboratory. While replaying the data, the clinician should be able to see the graphs and annotations in the software.

The software that the company currently has in use is outdated and its functionality does not fulfil the users' requirements. The existing features and requirements will be used as a basis in the virtual patient simulator.

4 Usability

In this chapter usability is defined as a term. After that, usability testing is introduced.

4.1 Definition of Usability

When talking about usability, people usually talk about the ease of use or user friendliness of the product. These definitions are easy to understand, but they are not precise definitions. These terms focus mainly on the user's needs of quick learnability and ease of use, while neglecting the need for efficiency in achieving the user's goal. Usability is just one attribute of the system's acceptability. [15]

Usability relies on research related to human-computer interaction and research done in the field of cognitive psychology. The term usability is often seen as an interchangeable term with human-computer interaction. Although usability is usually associated with human-computer interaction, it can also be used to define anything from a mobile phone to a faucet. The field of usability engineering aims at making the user and apparatus interaction more efficient and pleasant. [15]

The concept of usability has been studied widely and there is not just one definition of usability. The two most extensively used definitions of usability are Jakob Nielsen's definition and ISO 9241-11. These definitions will be introduced later on in the thesis.

4.1.1 Jakob Nielsen's Definition of Usability

Jacob Nielsen [16] defines usability as part of a bigger concept of system acceptability which is pictured in Figure 2 on the next page. System acceptability answers the question of whether the system is good enough to satisfy the users' and other stakeholders' needs and requirements. It is a combination of social acceptability and practical acceptability. These are equally important aspects, since if the system is deemed offensive, it might not matter how practically acceptable or easy to use the system is. In practical acceptability, the system is affected by various categories, such as usefulness, cost, compatibility, reliability, etc. Usefulness contains utility and usability. Utility refers to the question whether the functionality of the system in principle can do what is needed; usability refers to the question of how well users are able to use that functionality. If the utility of the system is low, the usefulness will also be low. [15]

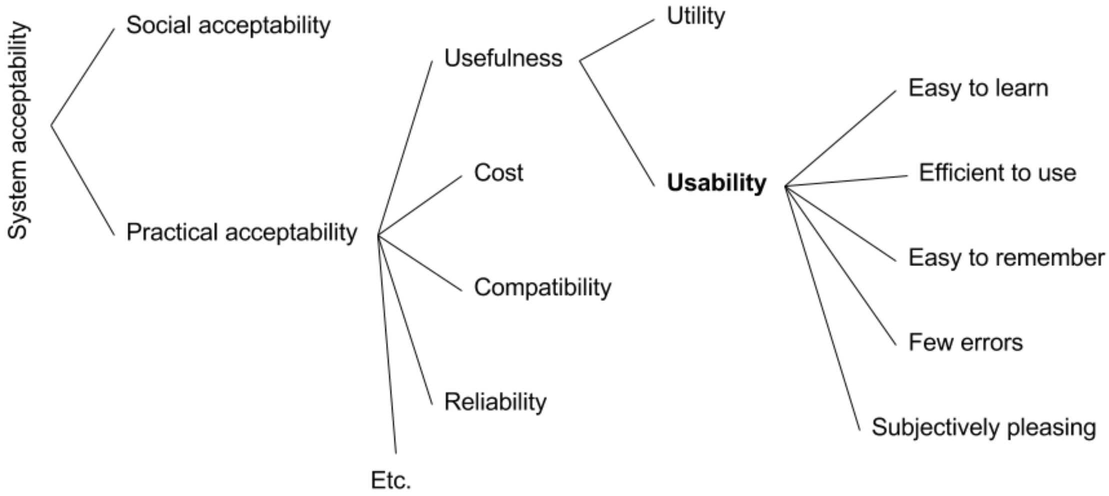


Figure 2: Usability as an attribute of system acceptability. [16, 25]

Usability is not just a single, one-dimensional property of a User Interface (UI). To define the abstract concept of “usability”, Jakob Nielsen describes it as an association of five attributes: learnability, efficiency, memorability, error and satisfaction. By these more precise and measurable components usability can be systematically approached, improved, evaluated and measured.

According to Nielsen, learnability is the most fundamental attribute of usability. Learning to use a system is after all the first experience most people have with a new system. Thus, the system should be easy to learn and understand; users should be able to discover how to do tasks easily by using the system. When analysing learnability, one should remember that users have a tendency of jumping right in and starting to use the system, without mastering to use it completely.

After the users have learned to use the system, the next thing that needs to be evaluated is efficiency. The system should be efficient to use, so the users can achieve a high level of productivity.

Since all users will not use the system daily, it should also be memorable. When a user returns to the system after a while, they are still able to use it without having to relearn everything.

When using the system, the users should make as few errors as possible. Error is an action that does not accomplish the desired goal. Some errors can be corrected immediately by the user, getting only somewhat slowed down by them – these mainly affect the user's efficiency. Some errors on the other hand can be fatal in nature; these errors can lead to faulty work or end up destroying the user's work. Fatal errors should not occur at all.

The final usability attribute is subjective satisfaction. Satisfaction refers to how pleasant the system is to use. This can be an especially important attribute for systems that are used in a non-work related environment, such as games. The entertainment value is more important for these systems than the speed of getting tasks done.

4.1.2 ISO 9241-11

ISO 9241 "Ergonomics of human-system interaction — Part 11: Usability: Definitions and concepts" [17] focuses on designing and evaluating usability for systems, products and services. The ISO defines usability as "extent to which a system, product or service can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use". This definition of usability puts emphasis on the specific circumstances in which a system is used. The ISO's definition can easily be incorporated to user and business requirements [18]. Like Nielsen's definition, the standard also lists learnability, efficiency, memorability and errors as important attributes when evaluating usability. It also adds attributes such as accessibility for users with wide range of capabilities; enabling maintenance tasks to be completed; and productivity.

4.2 Usability Testing

According to Steve Krug [19], the basic idea of usability testing is to watch people trying to use your product and note where they run into trouble. Then the idea is to fix it and test again. Usability testing is an important part of software development, since it is hard to look at your own work objectively.

Usability testing is an iterative process and multiple rounds of testing are better than testing once with a huge group. More users can find more defects in a single test, but the worst defects usually keep the users from finding others. Few users may not find as many defects in a single test, but if tested a second time they will find defects they could not have spotted on the first test. As the defects are fixed between tests, the more thoroughly the software can be tested.

Krug states that testing should be done early on and often enough to get the most out of it. It is easier to fix problems and implement better solutions early on than later on in the software production. It is recommended to test on users that are most likely to use the software, but if such users can not be found, Krug recommends testing on anyone you can and evaluating the results accordingly.

The results of the tests should be reviewed as soon as possible with the development team. It is important to review everyone's observations from the test to decide which problems need fixing and in what priority order. It is also good to discuss on how to fix the problems, especially if they are more complex. Sometimes just watching someone else using the software can help to inspire entirely new solutions to problems that were already recognised.

4.2.1 Conducting a Usability Study

Usability study is a series of tests and tasks conducted to test the usability of a product. A usability test is one session with a user. Carolyn Snyder [20] lists essential activities for conducting a usability study in her book about paper prototyping:

- Determining the goals for testing.
- Defining the users.
- Creating tasks around the things the users do.
- Creating the prototype needed to perform the tasks (if needed).
- Holding internal walkthroughs to prepare for testing.
- Conducting several usability tests and refining the prototype after each test.

- Establishing priority for the found issues.
- Planning additional changes to the interface in the short term.
- Communicating your findings to others who were not directionally involved.

When starting a usability study, one needs to have a clear goal of what they want to achieve. After determining the goal, it is good to identify the target audience. The target audience can consist of one or more user groups, depending on what is being tested. If there are more than one user group, each group should be given tasks to suit their different usage patterns. Typically tasks are made to represent the most common user goals, e.g. opening a file, or to suit the goals that the owner of the product deems important, e.g. replaying a file to a patient monitor. Usually users will have five to ten tasks in a 90-minute testing session. It is important that the tasks have clear success criteria and start state. It is also recommended that the tasks would be of interest to the users – a certain freedom within the limits can help to keep the users more intrigued. [21; 22]

Before conducting the first usability test, it is important to pilot test the equipment and materials with a volunteer user. Running the pilot test one or two days prior to the actual test, leaves time to fix technical problems, alter tasks or other materials if necessary. The pilot test also works as good practice for the facilitator and note-takers. [21]

A usability test is usually begun by introductions and an explanation of the goals of the test session for the user. If there are other people present in the session, they should be explicitly mentioned, e.g. note-takers. It is important to let the users know how valuable they are and that the system is under testing, not them. If testing is done with a prototype, it is good to acknowledge the unfinished nature of it and explain how the design will still evolve. Before starting with the tasks the facilitator should reward the users and get their signatures in informed consent forms. After introducing the system under testing and how the facilitator would like the users to behave during the testing session, it is time to begin with the tasks. The facilitator reads the user one task at a time and allows them to complete the task without interruption. To prevent bias, the facilitator should stick to the same script with all the users. [20]

It is good to record the user’s actions when testing, either by computer or video camera depending on the testing facilities. If recording the testing session is not possible, it is good to have people from the development team present in the test as observers or have someone take thorough notes on the process that will be later discussed with the team.

5 Methods

This chapter introduces the methods used in this project. The chapter will start by defining GUIDe model and then move on to paper prototype, focus groups and user requirements.

5.1 GUIDe model

The GUIDe model was developed by Sari A.Laakso and Kari-Pekka Laakso to solve UI-related problems in software development models [1]. It emphasizes the fact that the UI should be thought of before the software is built, since it can be harder and more expensive to make changes in the end of the development process. GUIDe tries to guarantee that the users of the software can achieve the needed goals reasonably and that there is no unneeded features. The name of the model comes from the letters in the activities that make up the model as explained below.

Goal: In the beginning of the process you need to think of goal-based use cases from the perspective of the user of your software. Use cases are descriptions of how the software ought to behave. Developers come up with the use cases and not the customer, even though use cases have to be approved by the customer. The use cases should not be too specific. These are important since they make up for everything that will be implemented later on to the UI.

User **I**nterface **D**esign: The order and what needs to be designed is defined together with the customer. The designer begins with the first goal and then creates the functionality and solutions to support it. When designing the UI, one should try to avoid adding extra features that are not needed specifically to achieve the goal. After the first goal is achieved, they add solutions for the other goals based on the use cases and the workflow. When the design is ready, the designer makes an UI specification that shows how the users can achieve the defined goals step by step. If a prototype is made, it can be tested with the end users to get comments. The feedback from the users can be used to fix the design to be more usable and user friendly.

Implementation: The UI specification is given as an input to the implementation specification to lower the risk of making a wrong kind of UI. This also guarantees that the programmers can focus on implementation without having to worry about the UI and usability at the same time. During the implementation phase the design can change due to compromises caused by coding restrictions. If the same people who designed the UI design also the implementation changes, there should not be any new usability issues.

The GUIDe model was chosen for the project since it focuses on the usability and UI aspect of the development of the software. It should also enable an as efficient implementation phase as possible to ensure that the end users will get the software that they need.

5.2 Paper Prototype

Paper prototype is made out of paper to visualise what a product would look like and how it would function. Paper prototype can be made of any kind of human-computer interface, e.g. web site, software or handheld device.

Paper prototyping can increase the creativity of the development group, since they can try different solutions on key problems with low effort. The solutions can then be tested further in a usability study to determine what solutions work best for the users. Testing early can help save money and time, since it can prevent major changes late in the development process. Iteration on a paper prototype can be fast as the prototype can be altered right on the spot if the user runs into problems or the facilitator notices some mistakes. Testing in the early stages will steer the development process in the right direction. Since anyone who can hold a pen can utilise paper prototyping, there is no need to waste time in trying to program a minimum UI. When choosing to program a prototype some functionalities may need to be left out or simplified, causing extra lines of code to be written to go around the functionalities. Thus, the code will probably be discarded later on when the product's implementation starts. [23]

When using only paper and pens, the tools that you can easily find even at home, it is quite difficult to make a prototype look polished or finished. This can be seen as a positive matter, though. Unfinished designs can encourage the users to be more creatively engaged in their feedback and even feel more included in the process. Even though the prototype does not have to look finished, it still needs to look realistic enough to elicit feedback for the issues that you're most worried about. There is even proof that paper prototypes seem to find as many and as severe problems as you would get when testing the real thing [20]. The effort put into paper prototyping needs to be weighed against the benefit that is gotten out of it. A sketched prototype will help to avoid the temptation of spending too much time on perfecting the visual appearance too early on in the development process, as changes are still most likely to happen. It has been noticed that neat looking prototypes can encourage users to give low-level feedback about the visual design [20], which is not useful until features are actually implemented. When something appears to be finished, minor flaws will stick out and catch the user's attention and might thus interfere with the more important issues.

A typical usability test with a paper prototype is similar to any other usability test, it just needs a proper introduction and clear instructions for the user. In the beginning it can feel unfamiliar for the user because of its unpolished nature and different operating system, but they will soon adapt to it. Talking is allowed as it can actually give a better insight for the facilitator about what the user is thinking.

5.3 Focus Group

Focus group is a way of qualitative data collection and it is used to understand and explain what your audience wants, needs and likes [24]. Much of focus group's value comes from participants reacting to each other's opinions, the group dynamic thus having an effect on the results. The results of the focus group are not statistically valid and should be used more as suggestive direction for the research.

The standard size for a focus group is five to ten people, but the preferred size is six to eight. If there are less participants, e.g. two to five, it is called a mini group. Mini groups are favoured in situations where potential participants are harder to reach and when the topic requires greater depth of discussion. Generally two or more focus groups are carried out as part of a given study to provide comparisons between groups. The group discussions should be facilitated by a trained moderator to ensure the discussion is focused, equally balanced and keeps on topic. Sessions normally last one to two hours and often include interactive techniques and informative materials to spark conversation. Interactive techniques can include role playing scenarios or writing opinions on post-it notes and later discussing them with the group. [2]

Focus groups are good for exploring the reasons behind attitudes and behaviour. A common mistake is to think of focus groups as a way of usability testing, even when focus groups can not really tell whether people can actually use the product or not [19]. Focus groups can give important insight when designing or implementing a product. The most common uses are brainstorming new ideas, testing new concepts, advertising or marketing promotions. Focus groups can also be conducted online. [24]

5.4 User Requirements

The user requirements define what the system is used for. A requirement is used to describe a quality that the system must satisfy. User requirements specification (URS) is a documentation specifying all the requirements that the users expect from the software, including any background material needed to make it understandable. The most common way to portray the requirements is in plain text with the help of Figures. The nature of the requirements should be: flawless, since a flaw in the requirements will affect later implementation; uncontradicted, since contradictions can be reduced by carefully compiled requirements; realistic, since they need to be feasible; necessary and preferably prioritised; verifiable; unambiguous; traceable [25]. [26]

There is no specified correct way to write the requirements; most organisations have their own way of devising them. Relatively broad requirements can be broken down into multiple more specific requirements to spell out the implications of the original. It is also possible to sort the requirements into different levels of specification, e.g. high-level to specify a standard and detailed-level to specify what the standard will include. [26]

The user requirements for the virtual patient simulator will be compiled from discussions and interviews with the users and the stakeholders of the software. The requirements will also be prioritised with the users and divided into mandatory requirements and desirable requirements. The division into mandatory and desirable features is done, because the company wants the first version of the software implemented as fast as possible and all features cannot be implemented immediately. After the first version, the software’s development will continue and other features will be implemented to it if needed.

6 Results

This chapter goes through how the methods were used in this thesis and what results were achieved. First off the UI is presented along with the usability study and the paper prototype. This is followed by the focus group interviews and the user requirements.

6.1 User Interface

The UI design process was started according to the GUIDe process model, by coming up with four use cases for the UI. The use cases were decided on and prioritised in a stakeholder meeting early on in the project:

1. Data visualisation.
2. Annotation.
3. Data input.
4. Data output.

After the defining the use cases, work was begun on sketching how the recorded data would look when visualised in the UI. During the sketching phase the pre-existing playback software and collecting software were used as basis for the design and feedback was discussed in the weekly stakeholder meeting. After the data visualisation was completed, the work proceeded on to sketching how the annotation of the data would look in the UI. Following that, the input and output options were sketched into the paper prototype.

6.1.1 Usability Study

During the project and in between use cases a usability study was conducted on three separate occasions. Five clinicians were consulted for approximately 15 minutes per usability test to get the end user’s feedback on the UI design. The aforementioned list of essential activities for conducting a usability study was followed during the testing together with the GUIDe model. The use cases were used as a basis for the tasks used in the testing. First the user was asked to open up a data file from the start screen of the software. When they had accomplished the first task, they were asked to replay the file and stop it. Lastly they were asked to add an annotation to the ECG graph.

In the beginning of the test, the testing procedure was explained as well as why usability testing is done and why it is necessary. Then the participants were presented with the paper prototype and asked to perform the tasks one at a time while thinking aloud their decisions. After the tasks, they were advised to give feedback on what they felt was good and what was still missing from the prototype. Notes were made of all the usability test sessions.

After each test session, the feedback was gone through and some of the suggestions were implemented into the prototype. Some good ideas that arose from the usability test sessions include:

- Real time clock that shows the time of the recording.
- Minutes and second shown in the time.
- Fast forward and rewind.
- Pressing stop takes back to the beginning while playing the file.
- Choosing a part of the file to play on loop.
- Choosing a part to annotate by clicking the starting point and then another click for ending the selection.
- Color coding different alarms and events.
- Wide area of selection.
- Hotkeys and keyboard shortcuts.
- Playlist.
- Tabs that allow multiple files being open at the same time for comparison.

Some of the ideas did not make it into requirements, since they were thought to be more advanced features, such as hotkeys. Looping was left out because it was thought to be unnecessary if the files can be cut into shorter bits.

6.1.2 Paper Prototype

During the UI design phase iteration a paper prototype was made of the software, which can be seen in Figure 3. The design is a suggestion of how the software should look like and how the different requirements and features should be implemented.

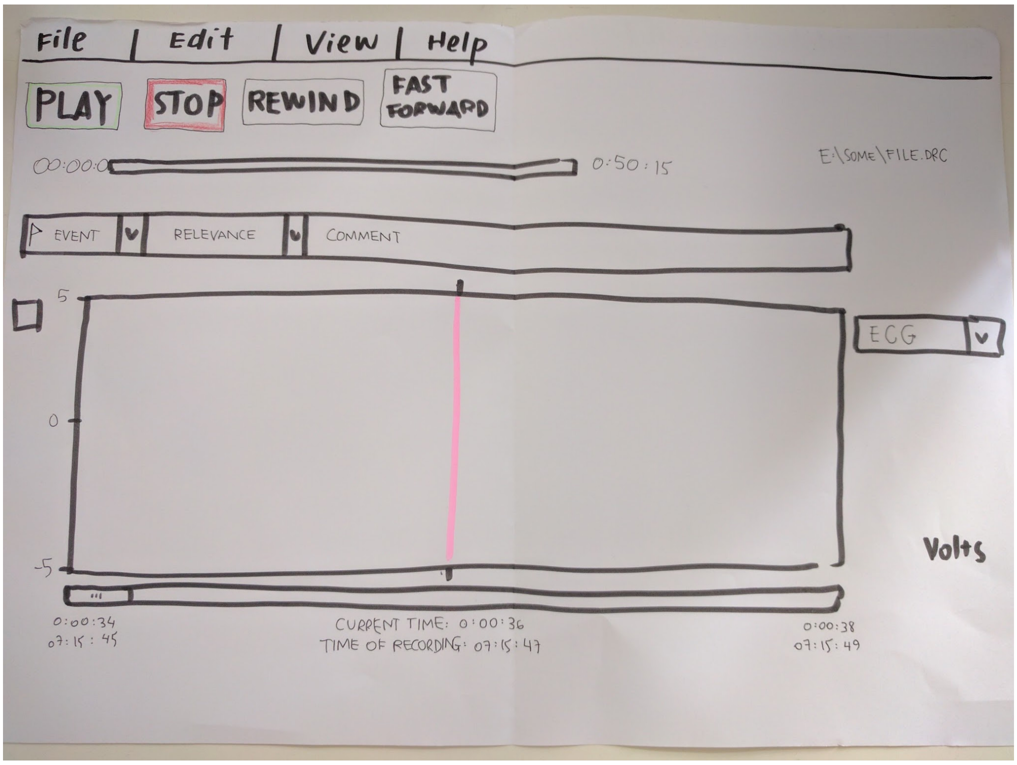


Figure 3: Picture of the paper prototype.

In the top left corner of the paper prototype is where the buttons are located. The buttons should contain at least options to play and stop the execution, but other options can be added next to them when needed (like rewind and fast forward). Play button should be made to alter between play and pause, depending on which state is not active at the moment. Since the clinicians especially asked for colours in the software, it is recommended that green is used for the play button and red for the stop button. Under the buttons is a bar that shows how much of the file is played and how much is still left. How much time has passed from the beginning of the file and how much is still left, are located at either end of the bar.

Next to the buttons in the right corner is where the file information is presented. The information should contain at least the file name and if the data is collected from a neonate (newborn). If the file path and size are deemed necessary, they should be shown here.

Under the buttons is the annotation bar. The annotation bar has to contain the same options for annotations as the annotation software. In the first dropdown the user can choose if the annotated part is an alarm or an artefact. The colour of the flag should be the same as the severity of the alarm. Artefacts should be marked on a colour deviating from the alarms. The colours used for alarms should be similar to the ones used in the patient monitor. The second dropdown is for choosing if the alarm was relevant, irrelevant, false or n/a. The last part of the bar is for adding a description of the event if needed.

The parameters are located under the annotation bar. The checkbox on the left allows the user to choose one or multiple parameters at a time, if there is a need to annotate or edit them separately. The large box contains the graph of the parameter in it is entirety. When a part of the graph is annotated, it should be the same colour as in the annotation bar and the annotation text should be visible on the top of the box. The dropdown on the right lets the user choose what parameter they want to observe from the recorded parameters. The unit of the parameter is located under the dropdown.

At the bottom of the software there should be two sets of numbers: one to describe the time the recording was collected at the hospital and another to describe the file's full length. There should also be a scroll bar for allowing observation of different parts of the graph. The times should change in relation to the scroll bar.

After conducting the usability study on the paper prototype, wireframes describing how the software should look like were drawn up on the computer. These wireframes of the UI can be found in Appendix 4.

6.2 Focus Group

The focus group interviews were held in spring 2016 on two separate days, to gather as many participants as possible. Two clinicians attended the first interview and three in the second. Almost all end users of the virtual patient simulator were heard during the interview process. The interviews lasted almost an hour each.

6.2.1 Interview Questions and Feature Prioritising

Focus group interviews were organised to find out about the expectations of the users about the software. The interviews were held to get a clearer picture of what features the users valued. The interviews consisted of a mini focus group and feature prioritising as an interactive technique. The findings from these data collection methods were analysed and used to prepare the user requirements.

The focus groups were questioned about the current work processes and how they could be improved. These questions were to give insight on how the testing is conducted at the moment and if the participants had ideas for future improvements. After these questions, the interview proceeded to questions about the new software e.g. which search criteria would be useful, what background information is needed for the files and what parameters would the participants like to have next. The complete list of questions can be found from Appendix 1. The questions were not asked in any predetermined order, but more according to where the conversation was going. This way the participants could continue their conversation more naturally without having to pause. After all the questions were asked, the participants were asked if they had any questions they would like to ask or if they had some thoughts they would still like to share about the subject.

After the questions the participants were given the task of sorting the features according to what they felt their priority was. Each possible feature was written on a separate card. All of the cards were given at once for the participants, so they could evaluate their values in relation to other cards. The features chosen for the cards originated from meetings with the stakeholders and from the users during paper prototype usability tests. If features that were not on the cards emerged during the interviews, new cards were added to the prioritisation.

When it looked like the participants were ready, the prioritisation order was discussed with the participants. After they had made a clear list of which the highest valued features were and which the lowest, a photograph was taken to archive the results. Then they were asked to choose five features that they would like to have implemented to the first version of the software. The results were also photographed for later analysis.

6.2.2 Interviews

When asked about the current testing process at the in-house laboratory, the clinicians all described it to be mostly freestyle exploratory testing, with sometimes the engineers setting up themes for testing or telling what does not need to be tested and which defects are already known. Apparently everybody has their own style of testing. Some clinicians like to use simulators and parameters from self to test the monitors generally, while others prefer to focus on one specific thing at a time - sometimes they have even tried walking around with the patient monitor connected to see how it reacts. The existing software is used rarely, since the clinicians are not sure how long or interesting the files are until they play them.

All the clinicians felt that clinical testing should be kept mostly exploratory. However, having dedicated themes for testing, such as a specific parameter or a feature to test, was also seen beneficial. They also expressed concern that the new system should show clearly how long the files are and what kind of conditions are in it. It is more preferred that the files are long and have interesting variations in the parameters, but it would also be good if there were some files that were short and had a lot of events going on. According to some of the clinicians, setting up the testing takes a long time, since the clinicians have to first search for a free simulator, then connect all the wires and leads – sometimes this preparation can take even half of the allocated time. Thus, it would be preferred that the virtual patient simulator could replay more parameters than ECG.

Since the current software is restricted to limited parameters at the moment, the clinicians were asked what parameters they would like to have added to the software next and what the most interesting parameters are testing-wise. Both groups mentioned the relevance of SpO2 to ECG data. Respiratory impedance and EEG caused especial interest in the clinicians, since they can not be simulated that well. Other parameters that the groups mentioned were: invasive pressures, such as blood pressure; spirometry and CO2. One of the clinicians said “an ideal future scenario would be that you go to the hospital and you record parameters, and when you come back here you can just play all of the parameters. It would be nice to see how ECG reflects on different parameters”. They all agreed that the more parameters you can play back, the better and more realistic it would be.

After discussion about different parameters, the focus groups were asked about what metadata would be needed of the recording. Does it matter which ward it was recorded at? What gender the patient was? How old they were? According to the clinicians it does not matter what ward the patient was at or what gender they were when conducting testing. The only information the clinicians felt that had some importance, was knowing if the patient was neonatal, child or adult. This information was deemed vital since the heart rate is quite different between these groups. This question in particular was shared with the other thesis workers, Tahiti Konttinen and Pekka Poikolainen, since they were making the user requirements for the annotation tool to be used while recording the data at the hospital.

When the clinicians discussed how the data files were managed, it became clear that the files were accessible only locally on two computers that reside in the in-house laboratory. For now that is okay, since the files are not editable nor contain annotations. In the future, the files should be shared at least between the two computers, so if someone adds annotations or edits the files into more usable lengths, everyone has access them.

After talking about the file system, the clinicians were asked about the search criteria they would want to use to sort the files. The most desirable criteria to search the files by was the length of the file and different alarms, such as arrhythmia or ventricular tachycardia for ECG. As mentioned before, the file should state if the data is from a neonate, a child or an adult. The clinicians also wished that the name of the file would state when it was recorded.

Defects are handled at the moment by taking a photograph or a video of the patient monitor and then attaching that to the report in the internal system. The first group of clinicians felt that it would be a good idea to annotate the defect into the file, so the engineers could easily find it there through the search, assuming the files would be shareable. The other group felt that this would be a bad practice, since it is not the data that is under testing, but the device. There was also doubts, if the defect annotations would ever get deleted after fixing. The second group’s idea instead was that the file name and the exact time of the defect could be attached to the report.

One of the stakeholders felt that audio alarms could be useful in testing, so the testers would not miss any forthcoming annotations in the data. The clinicians felt that this could be useful, but there is already too much noise while testing. If there would be audio alarms, they felt like there should be an option to silence them if needed.

Lastly, the clinicians were asked if they had any expectations or wishes for the new version. Below are all the ideas that were expressed:

- Virtual patient monitor that you could test the recorded data on.
- Quick transition to the next file.
- A box that states what kind of alarm it is supposed to be according to the annotation.
- Different colored alarms and use of colours in general.
- Simple and intuitive to use, since clinicians are not the most technically gifted.
- Relevance of the alarms (Relevant, Irrelevant, False, N/A).

All of the ideas were noted when designing the final version of the UI, except the virtual patient monitor that could be its own project in the future.

6.2.3 Feature Prioritisation

Both group's prioritisations were rated on a scale from one to five, where the most needed got five points and the least needed got one. Since the groups were of a different size and neither could prioritise the features into a straight line, because some features were of the same importance, the rating had to be adapted to the situation. The first group's result can be seen in Figure 4.

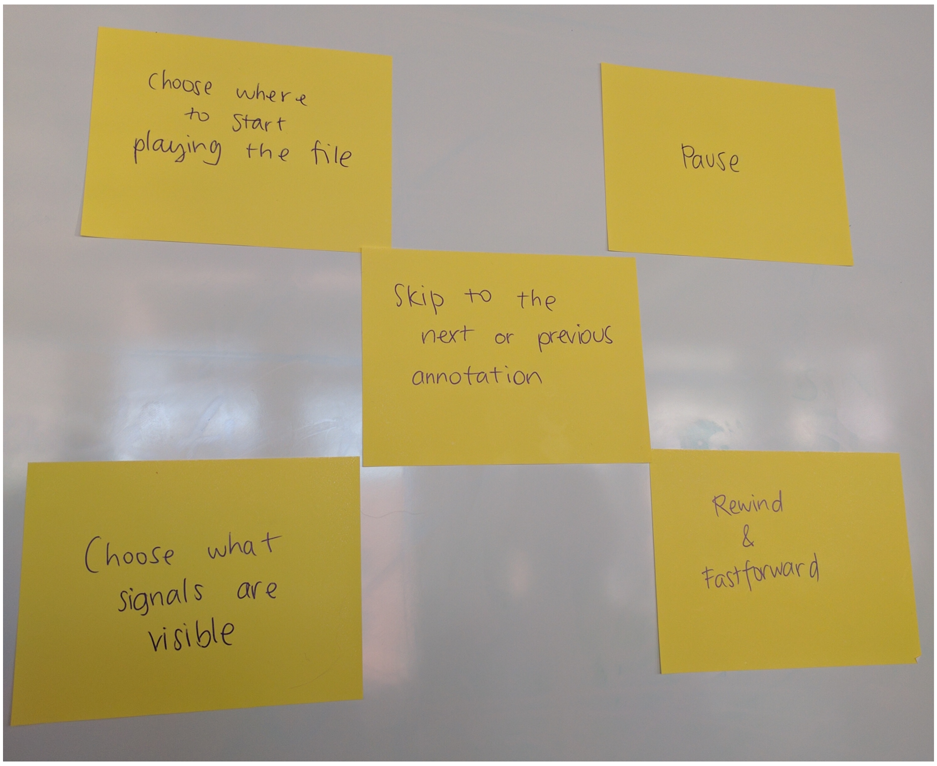

Choose where to start playing the file
Pause
Skip to the next or previous annotation
Choose what signals are visible
Rewind & Fastforward

Figure 4: Picture of the first group's top 5.

The results and scores of the prioritisation can be found in Appendix 3.

6.3 Mandatory Requirements

The mandatory requirements were created based on discussions at the stakeholder meetings. These requirements should be implemented to the first version of the software and they are listed in priority order in Appendix 2. These features are the basic features of the software.

Since the current software can replay recorded data to the patient monitor and read the data files that the collecting software makes, it is only reasonable that the first mandatory requirement is to fulfil the same requirements set for the existing software (URS_1). The only difference being that the virtual patient simulator should open one file at a time (URS_2). The new software should allow users to do more than play files, so opening and playing multiple files in a row could turn out to be bothersome. Unless of course multiple tabs or playlist features are supported, then it would be feasible to open and play multiple files consecutively.

To improve the clarity of the data files and help the clinicians see when alarms or artefacts happen, they should be able to see the annotations together with the parameter's graph (URS_3). The annotations and the recorded data are on different files, but they should be joined in the virtual patient simulator, possibly using a database linking the two files together. This linking should be done by the timestamps. If feasible, the graph of the parameter should move at the same time as the replayed data file.

When recording the data in the hospital, the clinicians usually record at least four different parameters, but the data file usually consists of multiple different parameters that were monitored and can be collected from the same electrodes. Thus, similarly to the collecting software's options to choose which parameters are visible on the screen, the clinicians should be able to do that in the virtual patient simulator (URS_4). In the virtual patient simulator it should be possible to see four different parameters one below another and to choose the ones that are visible from the available parameters, e.g. one should be able to see all the ECG leads separately.

To make good use of the already recorded data which is not electronically annotated, the software should also support adding annotations to it (URS_5). This will also be a useful feature if someone has forgotten to annotate the file. The annotation options should be the same as in the annotation tool. After adding annotations, it is important that the user can save their changes to the file (URS_6).

6.4 Desirable Requirements

The desirable requirements were created based on discussions with the stakeholders and clinicians' feedback during the usability study. They were not classified as mandatory requirements since the software can work fine without these features, but they will add to the usability of the software. The customer company will be developing a first version of the software in the near future. The stakeholders have to decide which features to include in this version and which they will leave for later implementation. To ease the decision making, the clinicians as the end users were asked to prioritise the desirable features. The features were numbered according to their desirability from 7 to 19 – Mandatory requirements being one to six.

The most desirable feature conducted by the research was being able to search for files (URS_7). Based on the interviews, the clinicians would like to search the files based on what alarms are present in the file. They would also like to see before opening the file how long it is and if the data belongs to a neonate, a child or an adult patient. As an addition to the search feature, there should be a possibility to add tags to the data files (URS_15). Adding tags would help the clinicians by making useful files more easier to find. If tagging is implemented, rules should be made to guide the use.

The recorded data can range in length from few minutes to several hours. Especially in the longer files, the clinicians have had to wait and see if there was something worthwhile. Therefore, it should be possible to choose where to start executing the file (URS_8). The clinicians should be able to start playing the file straight from where something happens, if they want to. Another feature to solve this problem was the ability to cut the file into pieces (URS_12). This way, some of the ordinary uninteresting data can be left out of the file to make it more efficient for testing. If the files were to be cut, there should be a different folder where they should be saved without altering the original files.

The virtual patient simulator should also have features to pause, rewind or fast forward the file (URS_9-10). In addition, the software should have a feature to skip to the beginning of the next or previous annotated part (URS_11). When skipping to a part of the file, there should be at least a few seconds of buffer for the monitor to catch up. It is also highly possible that for rewinding or skipping to work, the monitor has to recalculate the ECG. For efficiency, the software should trigger the recalculation automatically when necessary. If it is not feasible, then the software should inform the user to do it manually. Since patient monitors produce data constantly and a stop in data could mean an electrode has fallen off the patient, the simulator should keep playing a small part of the file over and over, when pause or user action is needed.

Even though the virtual patient simulator should be able to open only one file at a time for observation, it could open files on multiple tabs (URS_13). Still, only one file should be able to be replayed to the patient monitor at a time. One of the clinicians’ wishes for the software was for the files to change smoothly. Thus, there should be a possibility to make a playlist consisting of multiple files to automatically play in a row (URS_18). The playlist should open to its own window in the UI.

When examining the data files, the user sometimes needs more information from the parameter's graph. There should be a feature to zoom in on the graph to see changes more accurately (URS_14). To help more accurate interpretation, there should be tooltips to show e.g. the exact time and units of the graph or the time difference when a larger area is chosen (URS_16). Tooltips should appear when the cursor is hovered on top of the graph.

When replaying the file to the patient monitor, the graphs should move accordingly. To help the clinicians notice the annotations easier, the annotations should be shown more boldly when the annotated part of the file is under replay. While replaying the file, the clinician should be able to see clearly what annotation is currently replaying (URS_17).

During one of the stakeholder meetings it became clear that there should be a feature for demonstration purposes – the random function (URS_19). The random function would play random files in random order to show what the patient monitor can do. It was discussed that real data would look better on screen than simulated data. The random function could be a part of the playlist feature. This feature was prioritised at the very bottom, since it is not that useful in testing.

7 Summary and Discussion

The customer company has hours of recorded patient data and they are recording more every time they do testing in a hospital setting. At the in-house laboratory, they have a software and needed settings to replay ECG from the recordings to a patient monitor. The current software is outdated with limited functionality and does not have any metadata for the recording – no name, no length and nothing about the content. The annotations for the files are on paper documents far from the in-house laboratory, so the clinicians do not use them. The clinicians have to use the data without knowing what it contains, if they even choose to use it.

This thesis was done to kickstart a project on improving the use of real patient data on testing patient monitors. Under this project, two theses were begun simultaneously: this thesis to come up with user requirements and user interface design for the virtual patient simulator and another thesis by Tahiti Konttinen and Pekka Poikolainen to think of user requirements for the annotation tool [27]. The thesis workers cooperated during the project since it was vital that the annotations produced by the annotation software were compatible with the virtual patient simulator and both softwares had the same options for annotations. The research on the clinicians' preferences was also shared between the thesis workers, to guarantee the best possible user requirements for the softwares.

The GUIDe model was used during the thesis work, since it starts software development from the UI. The model's advantage is that it emphasises usability, which made it a good fit for the project. According to the model, the goals and users of the project were defined first. After that, the UI designing was begun by following the instructions on how to make a usability study. To carry out the usability tests, a paper prototype was made of the UI. The usability study and the paper prototype were good methods for figuring out how the UI should look like and what it should contain. Their iterative work process helped in figuring out which the best options for the UI were.

The usability study was conducted and tested only on the known end users. Since the software is for in-house use, this was deemed sufficient. The results of the usability tests were implemented to the paper prototype and later on to the UI design that can be found in Appendix 4. This alone still does not guarantee the usability of the final software, so the usability study should be continued in the next phases of the project to guarantee the final product’s usability after implementation. Since the UI is bound to change during the implementation phase, it is good to test that the end users can still understand and use it efficiently.

The focus group interviews were done for five clinicians who do most of the in-house testing at the company. The interviews worked well and it was good to hear the clinicians discussing ideas and concerns amongst themselves. The interviews and the feature prioritisation gave a good insight on what the clinicians were expecting from the software. All the user requirements can be seen in Appendix 2. A great deal of good development ideas arose from the interviews and they were recorded in this thesis under chapter 6.

Challenges in the thesis included a lack of experience in conducting the interviews and the usability study. But, since the thesis is a learning process, it was great to try new techniques. In the future when conducting a similar study, recording all conversations and test sessions with the end users is recommended, since notes are not as trustworthy and can be misplaced.

Even though the user requirements and the user interface design are done, there is still more work to be done before moving on to the next phase of the GUIDe model. The system requirements need to be compiled before the project is ready for implementation. The stakeholders will also have to decide what is to be implemented to the virtual patient simulator software. The first version of the software is planned to be finished in the near future.

Bibliography

1 Laakso, Sari A , Laakso, Kari-Pekka. Ensuring quality of the user interface with the GUIDe process model. 2004;Available from: http://www.cs.helsinki.fi/u/salaakso/papers/GUIDe.pdf.

2 Focus groups, mini groups, and 1:1 interviews;. Available from: https://www.gov.uk/service-manual/user-centred-design/user-research/focus-groups-mini-groups-interviews.html [cited September 16, 2016].

3 Majd, AlGhatrif, Joseph, Lindsay. A brief review: history to understand fundamentals of electrocardiography. Journal of Community Hospital Internal Medicine Perspectives, 2(1). 2012;Available from: http://doi.org/10.3402/jchimp.v2i1.14383.

4 Electrocardiography; 2016. Available from: https://en.wikipedia.org/wiki/Electrocardiography [cited September 16, 2016].

5 Karius, Diane R . ECG Primer : The Leads of a 12-lead ECG;. Available from: https://courses.kcumb.edu/physio/ecg%20primer/ecgleads.htm [cited September 16, 2016].

6 Aboukhalil, Anton, Nielsen, Larry, Saeed, Mohammed, Mark, Roger G , Clifford, Gari D . Reducing False Alarm Rates for Critical Arrhythmias Using the Arterial Blood Pressure Waveform. Journal of Biomedical Informatics, 41(3), 442–451. 2008;Available from: http://doi.org/10.1016/j.jbi.2008.03.003.

7 Physio Control. Minimizing ECG Artifact. 2015;.

8 Shortliffe, Edward H , Cimino, James J . Biomedical Informatics - Computer Applications in Health Care and Biomedicine. Springer; 2014.

9 Monitoring (medicine); 2016. Available from: https://en.wikipedia.org/wiki/Monitoring_(medicine) [cited September 16, 2016].

10 Heitz, Alexandre. Improving Clinical Education Through the Use of Virtual Patient-based Computer Simulations. 2013;Available from: http://hdl.handle.net/10092/8193.

11 Fluke Corporation. Prosim 6/8 Vital Signs Simulator. Users Manual. 2011;.

12 Kasurinen, Jussi Pekka. Ohjelmistotestauksen Käsikirja. Docendo; 2013.

13 Copley S. Testing the New System;. Available from: http://www.igcseict.info/theory/8/test/ [cited September 16, 2016].

14 Bach J. Exploratory Testing Explained. 2003;Available from: http://www.satisfice.com/articles/et-article.pdf.

15 Sinkkonen, Irmeli, Kuoppala, Hannu, Parkkinen, Jarmo, Vastamäki, Raino. Psychology of Usability. Edita Publishing Oy; 2006.

16 Nielsen, Jakob. Usability Engineering. Academic Press; 1993.

17 Bevan, Nigel, Carter, James, Harker, Susan. ISO 9241-11 revised: What have we learnt about usability since 1998? 2015;Available from: http://www.nigelbevan.com/papers/ISO%209241-11%20revised-%20What%20have%20we%20learnt%20about%20usability%20since%201998.pdf.

18 International Organization for Standardization. ISO/DIS 9241-11(en). Ergonomics of human-system interaction — Part 11: Usability: Definitions and concepts. 2015;.

19 Krug, Steve. Don’t Make Me Think! Berkeley: Peachpit; 2006.

20 Snyder, Carolyn. Paper Prototyping - The Fast and Easy Way to Design and Refine User Interfaces. Morgan Kaufmann Publishers; 2003.

21 Running a Usability Test;. Available from: https://www.usability.gov/how-to-and-tools/methods/running-usability-tests.html [cited September 16, 2016].

22 Usability Testing;. Available from: http://www.usabilityfirst.com/usability-methods/usability-testing/ [cited September 16, 2016].

23 Sinkkonen, Irmeli, Nuutila, Esko, Törmä, Seppo. Helppokäyttöisen verkkopalvelun suunnittelu. Tietosanoma Oy; 2009.

24 Edmunds H. Focus Group Principles; 2001. Available from: http://web.archive.org/web/20041026140011/http: //www.marketingpower.com/content1293.php [cited September 16, 2016].

25 Taina J. Vaatimusmäärittely; 2005. Available from: https://www.cs.helsinki.fi/u/taina/ohtu/k-2005/2luku4.pdf [cited September 16, 2016].

26 Withall, Stephen. Software requirement patterns. Microsoft Press; 2007.

27 Tahiti Konttinen, Pekka Poikolainen. User requirements for patient data annotation tool; 2016. Available from: http://urn.fi/URN:NBN:fi:amk-2016083113912.

1 Interview questions

Questions:

How is the testing done? What is the typical work process like?

How could the process be improved?

What parameters are usually recorded in the hospital?

What parameters would you like to have for testing? Or what are the most interesting parameters?

How is the file structure managed at the moment? Are the files shared?

On what criteria would you like to search the files? Number of artefacts, different conditions and alarms?

How should defects be handled? Annotated to the files?

What info is needed on a defect?

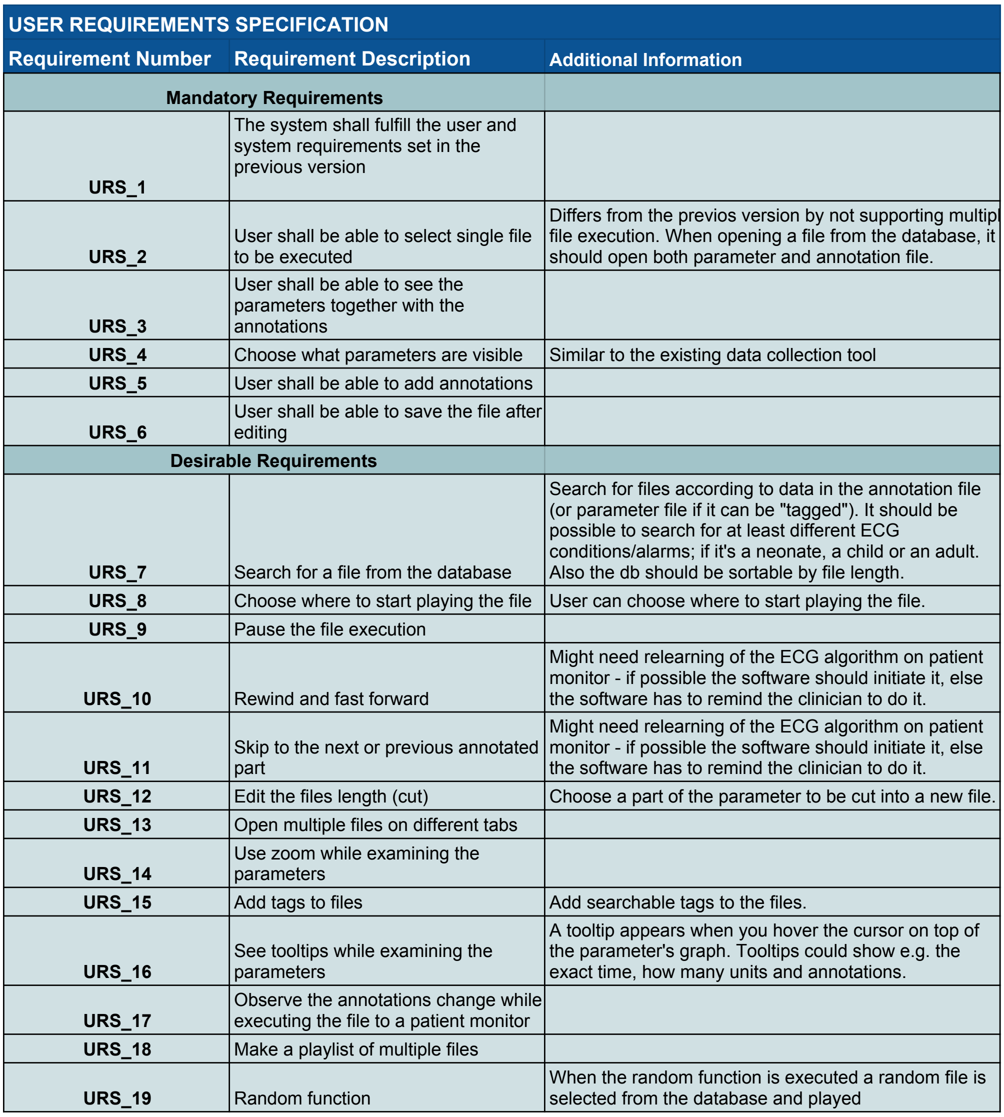
Should the virtual patient simulator have audio alarms?

Do you have any expectations for the new system?

Feature sorting

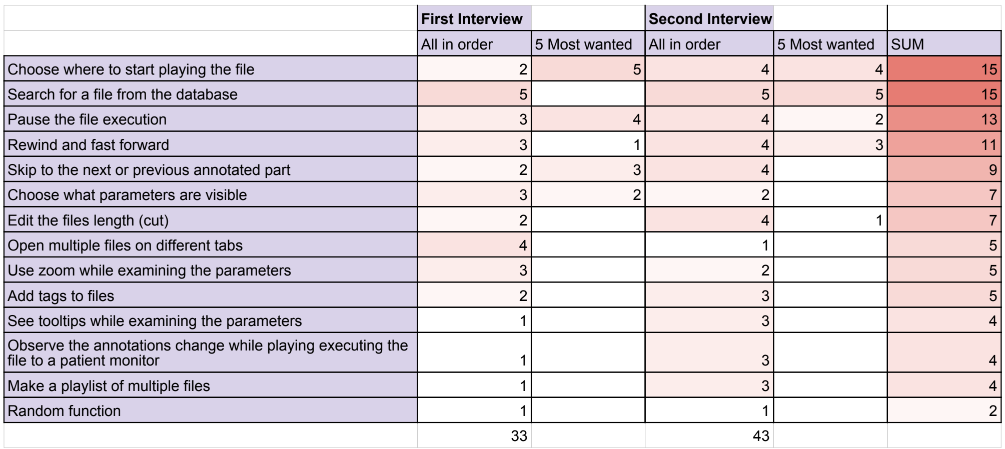
First task: Sort the notes on the level of need, on a scale of very useful to not really. Second task: Choose the five features that you would like to have in the first version of the software.

2 User requirements



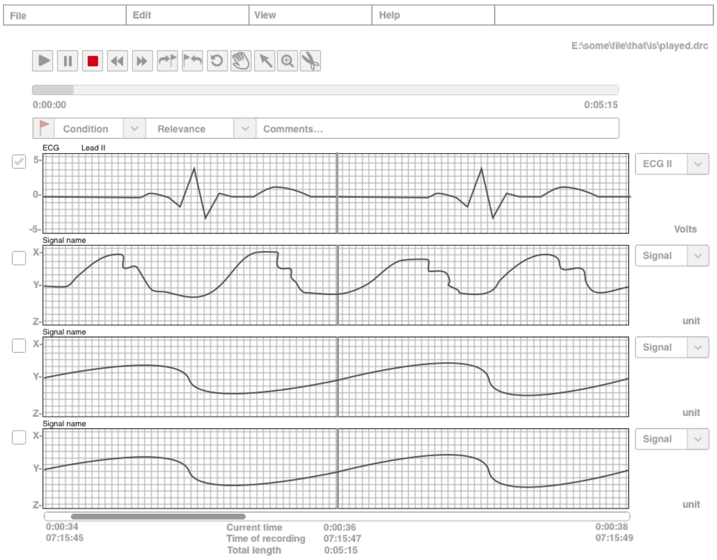
USER REQUIREMENTS SPECIFICATION		
Requirement Number	**Requirement Description**	**Additional Information**
Mandatory Requirements		
URS_1	The system shall fulfill the user and system requirements set in the previous version	
URS_2	User shall be able to select single file to be executed	Differs from the previos version by not supporting multipl file execution. When opening a file from the database, it should open both parameter and annotation file.
URS_3	User shall be able to see the parameters together with the annotations	
URS_4	Choose what parameters are visible	Similar to the existing data collection tool
URS_5	User shall be able to add annotations	
URS_6	User shall be able to save the file after editing	
Desirable Requirements		
URS_7	Search for a file from the database	Search for files according to data in the annotation file (or parameter file if it can be "tagged"). It should be possible to search for at least different ECG conditions/alarms; if it's a neonate, a child or an adult. Also the db should be sortable by file length.
URS_8	Choose where to start playing the file	User can choose where to start playing the file.
URS_9	Pause the file execution	
URS_10	Rewind and fast forward	Might need relearning of the ECG algorithm on patient monitor - if possible the software should initiate it, else the software has to remind the clinician to do it.
URS_11	Skip to the next or previous annotated part	Might need relearning of the ECG algorithm on patient monitor - if possible the software should initiate it, else the software has to remind the clinician to do it.
URS_12	Edit the files length (cut)	Choose a part of the parameter to be cut into a new file.
URS_13	Open multiple files on different tabs	
URS_14	Use zoom while examining the parameters	
URS_15	Add tags to files	Add searchable tags to the files.
URS_16	See tooltips while examining the parameters	A tooltip appears when you hover the cursor on top of the parameter's graph. Tooltips could show e.g. the exact time, how many units and annotations.
URS_17	Observe the annotations change while executing the file to a patient monitor	
URS_18	Make a playlist of multiple files	
URS_19	Random function	When the random function is executed a random file is selected from the database and played

3 Feature prioritisation



	First Interview		Second Interview		
	All in order	5 Most wanted	All in order	5 Most wanted	SUM
Choose where to start playing the file	2	5	4	4	15
Search for a file from the database	5		5	5	15
Pause the file execution	3	4	4	2	13
Rewind and fast forward	3	1	4	3	11
Skip to the next or previous annotated part	2	3	4		9
Choose what parameters are visible	3	2	2		7
Edit the files length (cut)	2		4	1	7
Open multiple files on different tabs	4		1		5
Use zoom while examining the parameters	3		2		5
Add tags to files	2		3		5
See tooltips while examining the parameters	1		3		4
Observe the annotations change while playing executing the file to a patient monitor	1		3		4
Make a playlist of multiple files	1		3		4
Random function	1		1		2
	33		43		

4 User Interface pictures



E:\some\file\that\is\played.drc

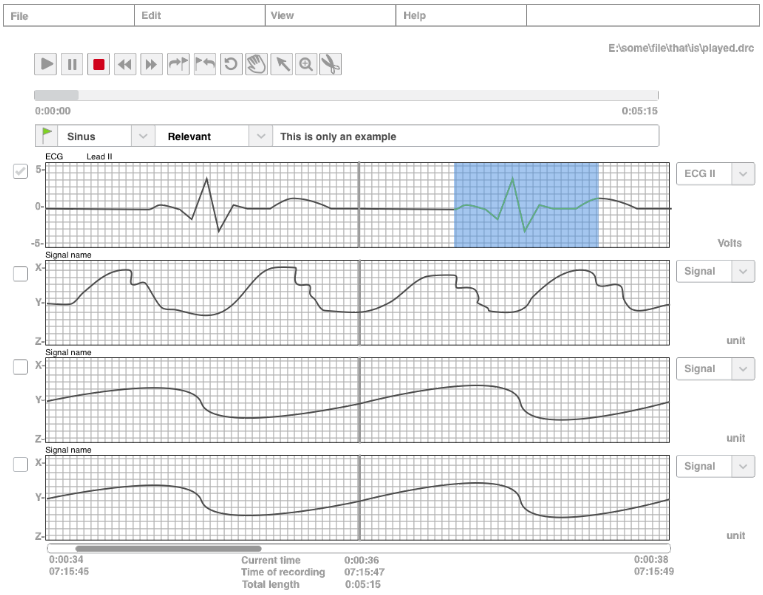
0:00:00 0:05:15

Condition | Relevance | Comments...

0:00:34
07:15:45

Current time 0:00:36
Time of recording 07:15:47
Total length 0:05:15

0:00:38
07:15:49



E:\some\file\that\is\played.drc

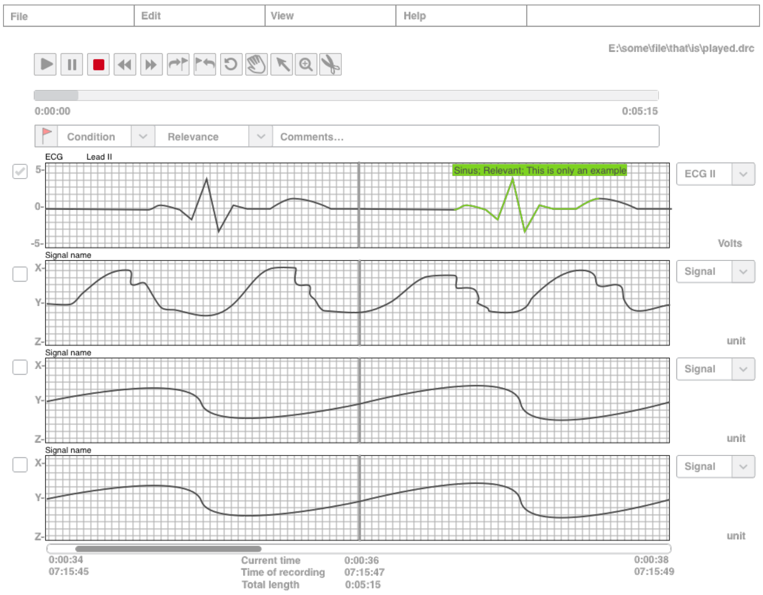
0:00:00 0:05:15

Sinus | Relevant | This is only an example

0:00:34
07:15:45

Current time 0:00:36
Time of recording 07:15:47
Total length 0:05:15

0:00:38
07:15:49



E:\some\file\that\is\played.drc

0:00:00 0:05:15

Sinus; Relevant; This is only an example

0:00:34 07:15:45	Current time Time of recording Total length	0:00:36 07:15:47 0:05:15	0:00:38 07:15:49