UNIVERSITEIT TWENTE.

A Creative Technology Bachelor thesis
Biomedical Signals and Systems, Research Group

Mobile pain screening with *FollowUp*

An app to support anaesthesiologist in the pain screening at outpatient clinics

Shanna de Lang

Supervisors:
Dr. ir. Jan R. Buitenweg
Dr. ir. Eric J. Faber
Mr. ir.ing. Richard G.A. Bults

Enschede, July 2016

Abstract

Chronic pain is difficult to treat. It is hard to measure pain objectively. Measuring the pain threshold of patients with chronic pain, can be part of the pain screening. NociTrack developed the *Ambustim*, a portable device which is able to measure the pain threshold through electronic Quantitative Sensory Testing (QST). However, there is no structured way to save the results of this measurement.

A prototype of an app that will support anaesthesiologists in this process, has been developed. The design process followed four design phases: 1) Ideation, in which there has been brainstormed about design question, 2) Specification, in which the ideas have been translated into project requirements through rapid-prototype testing, 4) Realization, in which the prototype has been realized in Android studio, 4) Evaluation, in which the prototype has been evaluated through user testing.

The prototype has proven to be effective. All users were able to perform tasks successful, within the given time constraints. Moreover, the usability of the interface scored above average on the System Usability Scale (SUS). Nevertheless, proper database structures, data security and unfinished functions need to be implemented before further development. Next to this, design issues regarding the usage of medical terms, more efficient and consistent spacing, need to be solved.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Current issues of chronic pain treatment

One out of five adults in the Netherlands is suffering from chronic pain [1]. According to different researches, an increased sensitivity is a key factor of chronic pain. Increased pain sensitivity is caused by a dysfunction in the nerve system: it responds too heavily to incoming pain stimuli. Medication exist, but because of negative side effects, it is not desirable to subscribe it to every patient with chronic pain symptoms.

Measuring pain sensitivity is possible through Quantitative Sensory Testing (QST). Hereby, pain stimuli, either mechanical or electrical, will be exposed to the body to inspect the functionality. Wilder-Smiths review of Sensory testing proves that QST can be a valuable clinical tool for assessing risk for the development of chronic pain [2]. Unfortunately, QST is not applied in daily clinical practise, because the equipment and the usability of the method do not suit the point-of-care.

The research department Bio Signals and Systems (BSS), at the University of Twente, developed the Ambustim to measure pain sensitivity everywhere in the hospital. This portable device, controlled by a tablet or smart-phone, measures the pain threshold through electrical stimuli on the skin. This makes the measurement process easier and faster. However, measuring takes time, execution is not flawless. Moreover, the results of the test have to be written down manually, which aspect results in unstructured documents of numbers. Therefore, there has been requested to develop a mobile application (app), to improve the efficiency and usability of the measurement process in the daily practice of anaesthesiologists. The app should be able to save this results and present it.

1.2 Project goal and research questions

This goal of this project is to design an app to support anaesthesiologists during a pain screening. This report elaborates on the design process of a prototype app, which allows anaesthesiologists and medical staff to use electronic QST in daily clinical practice. To reach this goal, three sub questions were formulated. These questions are:

1. What patient and pain related information do anaesthesiologists consider to be relevant and should be available and addable into the app, and when should this a be accessible?

2. What laws and regulations related to patients privacy and data collection are applicable?

3. How can an effective mobile application be realized in Android Studio within the legal constraints, which supports anaesthesiologist in the pain screening with the Ambustim as a clinical tool?

1.3 Methodology

The methodology used is based on the Creative Technology Design Process, as described by Mader [3]. This process consists of four phases: ideation, specification, realisation and evaluation. A schematic overview of this process can be found in figure 1.1. The report starts with a state-of-the-art review, in which the project background and related works are investigated through a review of literature. This chapter will elaborate the methods that has been applied throughout the four design phases.

The ideation phase identifies stakeholders and their needs through expert and user interviews. It describes the users in more detail trough personas. A persona is an imaginary individual, to represent a specific target group [4]. The leRouge method, which recognises different segments of user information: demographics, health-care specifics and technology, will be applied [5]. This use-centered method has been developed for the creation of personas regarding technological health-care products. Next to this, a description of the brainstorming process regarding the realisation of their needs, will be given. This part reviews the different creative ideas, to continues to the next phase with one or more draft concepts.

The next phase, specification, further develops a selection of creative ideas, followed by a review through rapid prototype testing. The outcomes of the interviews and user tests, result in a lift of project requirements. Next to this, use-case-scenarios are constructed. They provide a deeper understanding of the goals of the users, their expectations and problems users might encounter. This phase results in a final list of requirements and the final design.

In the next phase, the final design is realised. It has been required by the client to develop an mobile android application. A software platform had to be chosen to develop the app in. Android Studio as been chosen as a platform to develop a prototype. The were several reasons to choose this platform.

- The systems user interface has a high usability and is visually orientated (preview of the layout is always available).
- There is a lot of documentation available online.
- The available documentation available online is clearly formulated.
- It has been advised by other Creative Technology students who have experience with app development.
- Previous personal experience with Android Studio.

Finally, the prototype created in the previous phase, been evaluated by end-users and experts in the Evaluation phase. Three aspects of the user interface were evaluated: the effectiveness, efficiency and satisfaction. This phase will conclude with to what extend the project meets the requirements listed at the and of the specification phase.

The project report will finish with a conclusion. In the conclusion, the three research questions, formulated in the previous section, will be answered. Possible improvements regarding the prototype will be underlined and advice for further research will be given.

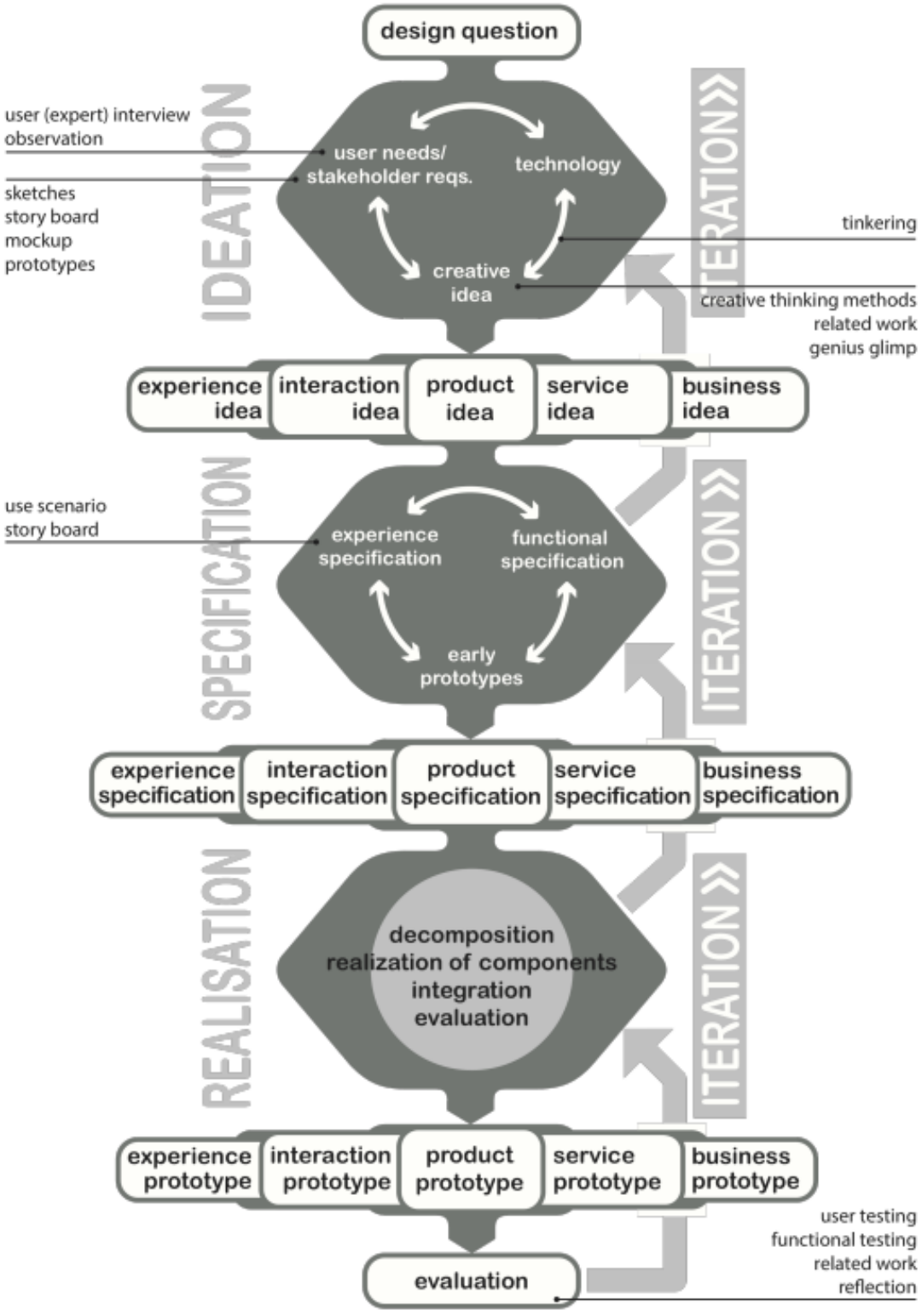


Figure 1.1: The design process for creative thinking, by Mader and Eggink.

Chapter 2

Background

This chapter includes a review of the state-of-the art in the field of development. First, there will be elaborated on chronic pain and the current treatment of patients suffering from it. Second, existing mobile applications related to supporting health-caregivers or specialist and pain management. At last, the second research questions regarding laws and regulation applicable for such apps, will be answered.

2.1 What is chronic pain?

Pain is a complex multidimensional sensory-phenomenon [1]. Pain and emotion can not be separated [6]. Pain can be categorized in acute pain and chronic pain, or nociceptive pain and neuropathic pain. Acute pain indicates possible danger or damage of the body, which provides us the sign to quickly distance ourselves from the source of danger. Chronic pain however, does not have a biological function [7]. Chronic pain is defined as pain which maintains three months or longer, or longer than expected regarding the disease or damage [1]. Nociceptive pain is caused by painful stimuli, such as physical tissue destruction or chemical, pressure or health processes, which are detected by neurons (nociceptive nerve fibers) [8]. Neuropathic pain is defined as pain initiated or caused by a primary lesion or dysfunction of the peripheral or central nerve system. It is associated with severe chronic medical conditions [6].

Chronic pain is a big health issue. It has great impact on the quality of life, pleasure, mood, and absenteeism of a patient. Compared to other chronic diseases, it occurs frequently. A proposal constructed by experts in cooperation with Pfizer, states that approximately one out of five adults in Europe suffers from chronic pain [9]. Only in the Netherlands over two million citizens experience pain daily, which is equal to the sum of the amount of diabetes, heart and cancer patients [1].

2.2 The clinical practice and implementation of QST

Treating pain is often complex. The *Regieraad Kwaliteit van Zorg (RKZ)*, a dutch advisory board, and Merskey explain that there is no official taxonomy of pain. Pain is not recognised as a disease "in its own right" and therefore not included in the International Classification of Diseases (ICD) [1] [10]. There are many different possible causes of pain. Patients with chronic pain symptoms are therefore diagnosed by different kinds of medical specialists [11]. During an interview with anaesthesiologist Krabbenbos he explains "Pain is a symptom. What is causing it, has to be understood before an adequate treatment can be applied." Next to this, there is a wide range of pain treatments. The RKZ underlines that this wide range of treatments, pharmacological as well as non-pharmacological [1]. However, many patients experience their treatment as inadequate [12].

Several bottlenecks regarding the clinical practice of pain medicine can be addressed. General standards and guidelines are missing. Breivik shows that only 60% had been asked to express

how much pain they were experiencing. 9% of them was asked to do this based on a pain scale. Moreover, only 23% of the chronic pain patients who visited a general practitioner, has been seeing a pain specialist [13]. There can be concluded that structured integrated care regarding chronic pain symptoms is missing.

An extensive QST can be a valuable clinical tool for monitoring and assessing risk for the development of chronic pain [2]. Implementing QST in clinical practice, provides an objective quantification of pain. Krabbenbos said: "At the moment, we build upon what a patient tells us. This is rather subjective. A patient might rate the pain with an 8 before treatment, and a 3 right after. As a patients get used to the treatment, peoples references change and they might say 8 again." Unfortunately, QST is not applied in the daily clinical practise, because it does not suit the point-of-care [2]. Currently, QST has to be conducted in a lab. The Ambustim has been developed to make QST easier and more available.

The procedure with the Ambusitim is as follows. During the measurement process, the patient will be exposed to electrical pain stimuli through two electrodes on the skin. As the patient is holding the button, a current will start to flow. The specialist can ask the patient to release the button at three different sensations. The first is as soon as the patient starts to feel the incoming stimuli, to measure the sense threshold. Second, to release once this feeling starts to become unpleasant, to measure the pain threshold. Third, to release when the pain becomes unbearable, to measure the pain tolerance. The second value, the pain threshold, is relevant for diagnosing an increased sensitivity for pain. The system will return the value of the current flowing at the time of release.

Currently, this value will have to be written down by the doctor on paper, which results in an unstructured document of numbers. This is a disadvantage for the implementation of the system in clinical practice. A mobile platform, such as a mobile application which conducts the tests and saves the results, provides the ease of use which is needed for implementation.

2.3 Related Work

This chapter will elaborate previous research and developments regarding mobile applications to support in the daily clinical practice, and related to pain management. Because the app provides medical support on electronic, mobile devices, this arouses interest in the following fields: Mobile clinical decision support systems (CDSS), e-health and m-health applications aimed on medical professionals and chronic pain related mobile applications.

Before the review continues, e-health, m-health and clinical decision support systems will be defined. According to the RVZ and The World Health Organisation, e-health is on the intersection of medical informatics, public health and business related applications, combined with the use of the Internet or related technologies [14] [9]. M-health or mobile Health falls within the category of e-health and is defined as medical and public health practice, supported by mobile devices, such as mobile phones, personal digital assistants (PDAs), and other wireless devices [9]. Mobile clinical decision support systems (CDSS), are mobile application which provide suggestions regarding the patients condition and possible treatment or medication.

According to a review of Jaspers, significant evidence can be found that CDSS affects health care providers performance positively [15]. In the review on mobile CDSS from Martinez-Prez et al, 172 android apps related to medical clinical decision support were found [16]. Most are focused on general medicine (47) and drugs information (16). Only 6 apps were related to neurology, but none to chronic pain specifically. Regarding the target users of CDSS, 48,4 % was aimed on medical specialists and 33,9 % on physicians. Jaspers mentions that most user interfaces of apps that target medical specialist focus on the information content, rather than the ease of use and user experience [15]. There has been recommended to avoid text-based interfaces and to take more advantage of the interaction that smart-phones can offer. Additionally, the required time of interaction with the system should be taken into account. Implementing this, and focusing on a

medical fields with little research, will enhance the effectiveness of mobile CDSS [16].

Regarding pain management applications, Rine, et al, state that the ability for (self-)monitoring is promising for implementation in the clinical practice [17]. However, current applications are mostly commercial apps, which lack validation through empirical testing. Apps are often simplistic and their effectiveness, regarding health outcomes, has not been tested [18]. Questions remain regarding the actual influence on the patients' situation. In a review concerning pain self-monitoring apps of Lalloo, et al, it was also concluded that self-management apps for patients lack the involvement of health-care professionals in their development [18]. More academic guidance is needed, but this requires commitment and time from not only the patient, but also medical staff [18]. This will increase the labour intensity, which will form a barrier for the implementation.

In the Google play store, there has been searched for other mobile applications regarding pain management or medical professionals as target users. The keywords to medical decision support systems, medical professionals and "pain management" have been used. A short analysis will be given for 6 of the apps that have been found. The first three apps are related to CDSS and the others are pain related (self-)monitoring apps.

- eH&P [19], an app to facilitate, organise and process patients clinical data and offers reference, productivity and decision making tools for medical professionals. However, the system does not have a focus on a specific medical fields, neither support for symptoms which are difficult to diagnose. The broad scope results in a lot of options and menu items.
- Best Practice [20], is a decision support tool for physicians, nurses and medical students. It includes over 1000 diagnostic tests and over 3500 medical images. Academic guidance or involvement in the development has not been mentioned, but they claim to be a trusted, authoritative clinical decision support tool. They do provide references within the app (see figure 2.1).
- MediDSS [21], is developed by the Finnish company Medical Publications Ltd, for assisting clinicians in the selection of medication. It is build upon 800 healthcare support rules and the description mentions the usage of academic references. See figure 2.1.
- Manage My Pain [22], a self monitoring system for patient suffering from chronic pain (See figure 2.2). It allows users to track, analyze and share their pain experiences, to help them cope with their situation. Their reports are designed with guidance of doctors. This reflects in the medical concepts, such as a pain rating on a scale from one to 10.
- My Pain Diary [23], a pain tracking app which allows patients to keep accurate records for their doctor. It has been created by patients with chronic pain after they have been diagnosed with the complex regional pain syndrome. This is chronic arm or leg pain developed after an injury or surgery. It is possible to track many different conditions and symptoms: the pain location, pictures of visible symptoms and even the weather.
- Catch My Pain [24], is a pain dairy and forum. It visualizes and monitors pain by detailed drawing and tracking other relevant information. It is linked to research at the University of Zurich and various medical experts. The description mentions that the app has won an award, but not when and which award.

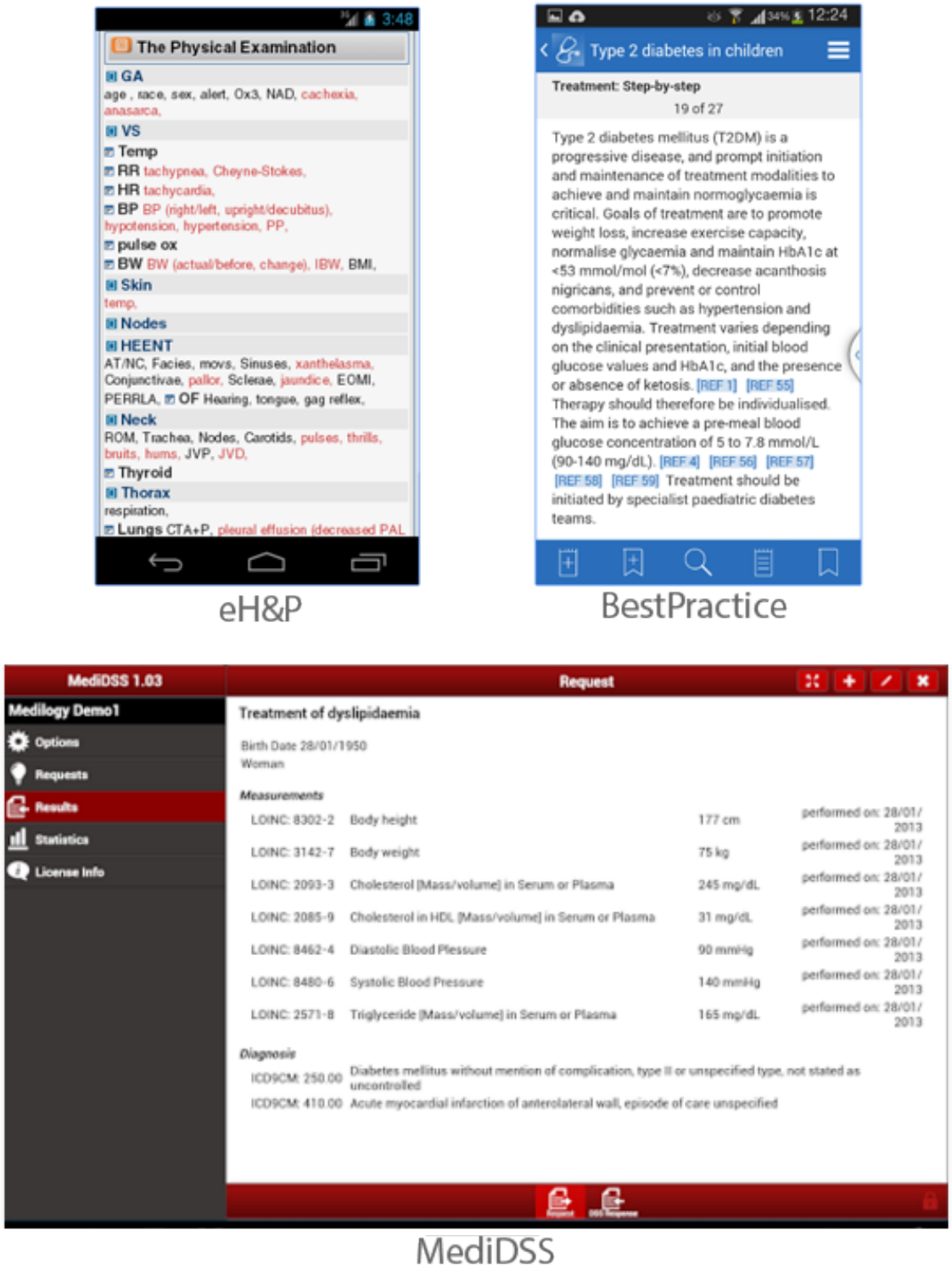


Figure 2.1: Screenshots of three app related to mobile clinical decision support.

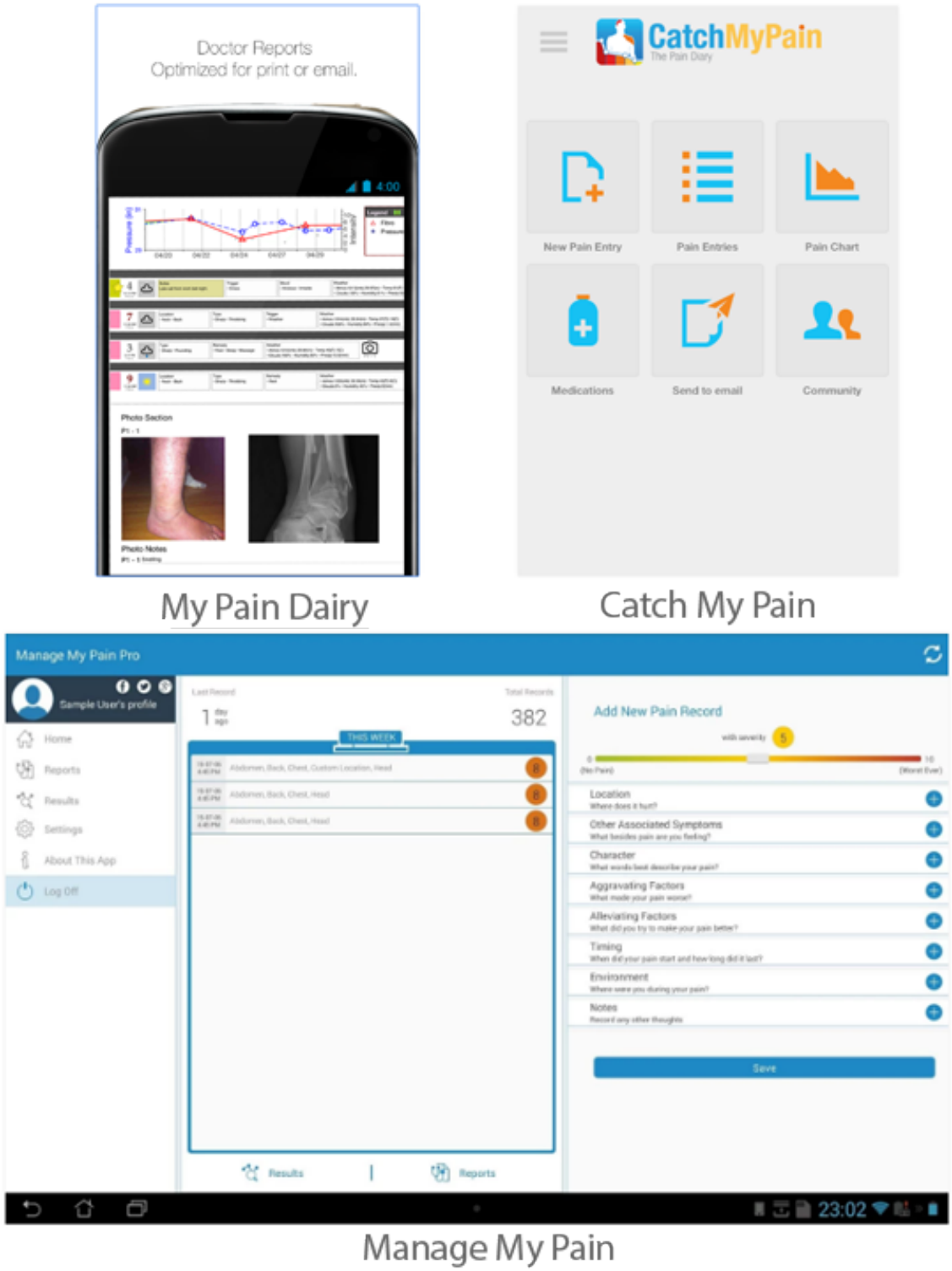


Figure 2.2: Screenshots of three pain tracking/monitoring apps

A number of design aspects can be reflected on regarding these apps. For example, the apps Catch my pain and Manage My Pain have the minimalistic user interfaces with clear images and graphs. Categorising the content and features, also provides clarity and calmness (see figure 2.3). Other apps, for example eH&P and Best Practice, use mainly text-based information. These examples are shown in figure 2.4. Compared to the interfaces shown in figure 2.3, there are harder to read and understand.

The design for this prototype should user clear categorisation and visual representation of the data, such as the examples in figure 2.3. Big text areas should be avoided. Regarding the privacy and security, more knowledge is needed. The reviewed apps do not specify how they have handled such legal issues.

Figure 2.3: Some possitive aspects of the reviewed apps.

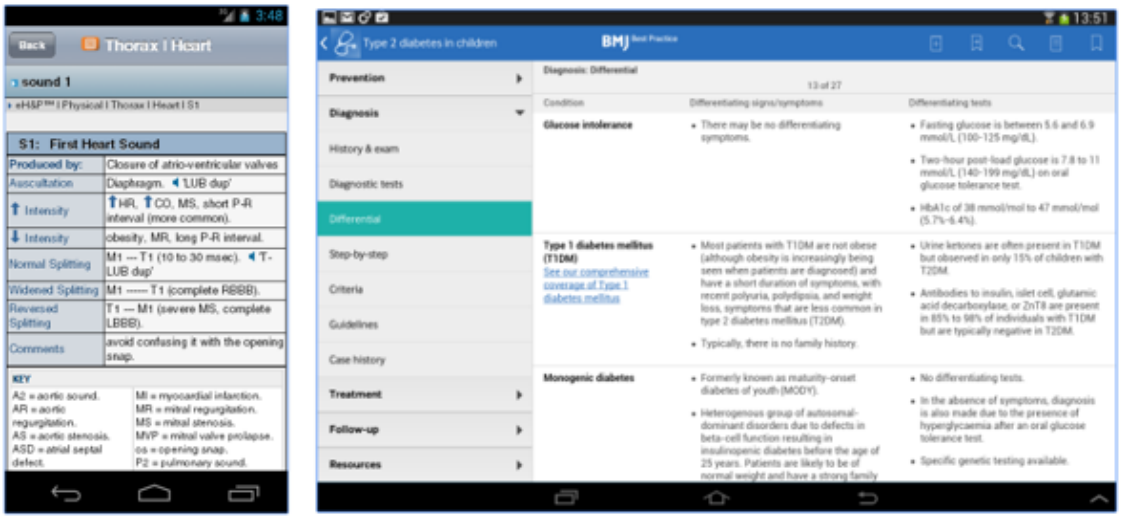


Figure 2.4: Some negative aspects of the reviewed apps.

2.4 Laws and regulations regarding patient privacy and security

The app will be collecting, storing and representing personal information. This section will address the laws and regulations applicable to patient privacy, data security and authorisation regarding the collection of and processing of such information. The necessary personal information to be collected will be considered, followed by which laws apply to them and if it is wise to reduce or avoid the collection of certain information. With this information the second research question, *What laws and regulations related to patients privacy and data collection are applicable?* will be answered.

In the Netherlands, the privacy law "Wet bescherming persoonsgegevens (WBP)", applies to all information regarding an identifiable, natural person. In other words, it applies to personal data [25]. Personal data, is defined as all information for purpose-orientated usage directed to an individual person [26]. With this knowledge, there can be concluded that this law will only apply if the data is linked to information which is able to identify the patient. This suggest that, if the health information is not linked to identifiable data, such as name, but non-identifiable data, the WBP might not be applicable.

The collection of personal data might be avoided by numbering patients randomly. Privacy laws do not apply to this type of data. However, it is needed to identify of the patients to some extent. The doctor needs some method to recognize if the data really corresponds the patient, in order to obtain valid results. To do this, an additional document, attached to the electronic health records for example, to translate is back into identifiable data will be needed. As a result of this, the data will be considered personal information again[27]. To conclude, because the app is processing health data which is related to an individual, it will always be personal data. This means that the WBB will apply. Companies and/or agencies which process personal information, should register this at the government department regarding authorisation of personal information [28].

The health data that will be collected, will influence the diagnose and treatment the patient. Relevant medical data has to be included to the patient's health record, because every practitioner is obliged to [29]. As a result of this, the same security and authorization regulation might be applicable. Therefore, these regulations have to be considered. For authentication, this means only the doctors and assistants directly involved are allowed access this data. According to de Bruijn and de Brie, the authority who collects the data, has to protect this data from getting lost and inspection by unauthorized parties, and secure accordingly [29]. They underline that a strong level of authorisation requires a physical authentication key in combination with a password or code.

Regarding authentication in the app, only the medical staff which has access to the patients health record, should be able to access the patient page on the app. It might still be desirable to save the data with a random number which can only be encrypted with patients health record, to add an extra security layer. current laws and regulation are already applied for securing the health records.

To conclude, the research question, *What laws and regulations related to patients privacy and data collection are applicable?*, will be answered. The personal data of patients will be collected. Next to this, relevant medical data will be obtained through the app. This means that the following laws and regulations be applicable on the app:

- Dutch privacy law regarding the protection of personal data, the WBP, is applicable. Which means that:
  - The patient has to be informed about the exact data the app stores.
  - The patient has the right see inspect, request to delete, change or supplement the data.

- The data collection has to be registered at the Dutch government department *Autoriteit persoonsgegevens* and clearly defined, with respect to justification of the collected data and its purpose, in proper language.
- The collected health information has to included in the (electronic) health record of the patient.
- To obtain a high security level, a password in combination with an unique and personal physical key should be used for authentication.

Chapter 3

Ideation

In this chapter, the ideation phase is described. First, a user analysis has been done. For this, the important stakeholders of the project are analysed, followed by a more detailed description of the actual users through personas. Moreover, the user needs are translated into requirements. Second, the brainstorm regarding three major design issues of this project, is reviewed.

3.1 User analysis

3.1.1 Users and Stakeholders

Three actors are interested in the development of this app: the pain specialist, the patient and the hospital. Figure 3.2 describes to what extend: they have interest in the app, they are affected by the app, they are a target user, they have the means resist against the implementation and they have the potential to help and support during the design and development process.

To determine which of these stakeholders should be focused on, their power and interest can be compared to each other. This will help to understand whose needs should be watched closely [30]. Moreover, priorities with respect to the stakeholders requirements can be set. The power/interest grid of the different stakeholders is shown in figure 3.1. More details about the stakeholder analysis can be found in Appendix A.

Figure 3.1: The power-interest grid of the stakeholders.

To summarise, of all the identified stakeholders, the anaesthesiologist scores high for all parameters. The patient scores second with respect to this parameters. Next to this, the power-interest ratio of the anaesthesiologist is also the highest of all stakeholders. There can be concluded that the anesthesiologist, is the target user, so their needs are the most relevant for the project.

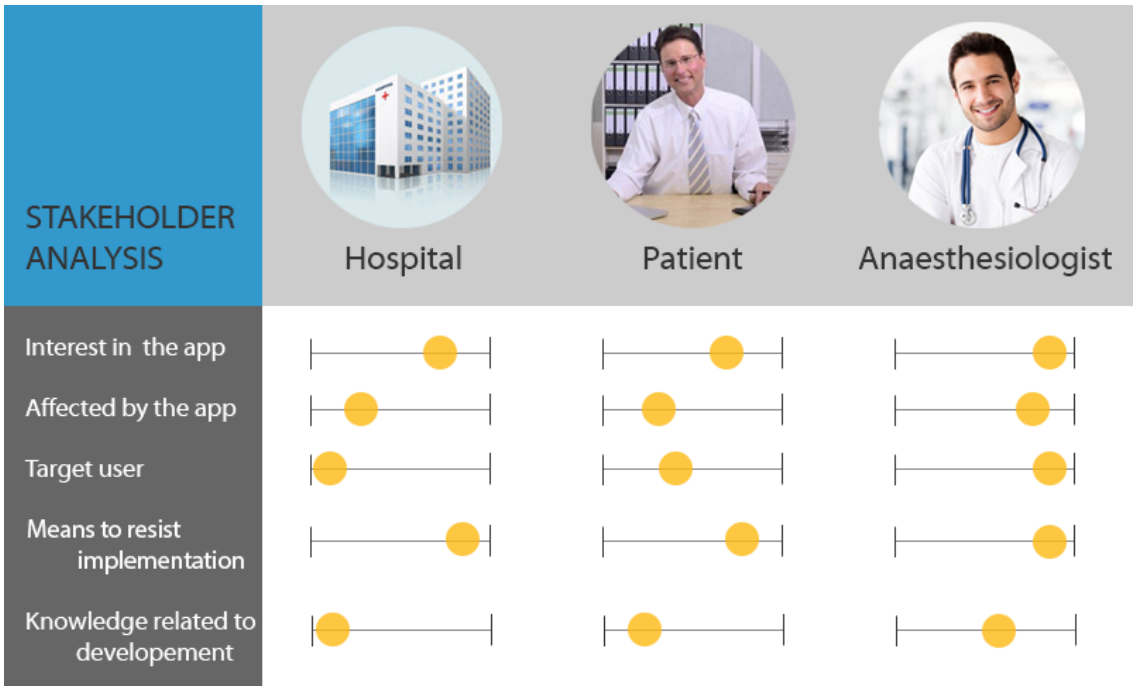


Figure 3.2: A schematic overview of the stakeholders description.

3.1.2 Personas

These personas have been extracted from the answers of an interview with the end user, in the categories based on the LeRouge user model [5]. The interview results and what has been extracted for the personas can be seen in Appendix B.

Erik Veldweg

Anaesthesio-logist

"The Ambistim can provide us with objective results. It would be nice to conduct the test as part of the routine."

AGE: 34

EDUCATION
Medicine, specialization is anaesthesiology.

PROFESSION
Anaesthesiologist in the hospital., where he spends two days in the pain outpatient clinic and 3 days at the operation theatre complex

Patients he receives at the outpatient clinic, all report to experience pain. This pain occurs mostly at places of a former injuries or peration, the knee or the back. He believes the majority of this patients suffer from neuropatic pain. Most patient had visited another doctor before, but couldn't get a complete diagnoses.

'Pain is a symptom, not a disease. You need to understand the symptoms, before a good diagnose can be made."

This makes every case different. Therefore, he works together with different specialists. Erik believes QST and the Ambustim can be a good clinical tool to monitor the pain. 'We have to go with what the patients is telling us. This system could provide more objective results. It would be nice to conduct the test on every new patient as part of the routine.' At the moment, he sees too many obstacles in the current system to implement it. He wants the usability to increase and the measurement process to be standardised, so results won't vary because of this aspect.

Tom Overbeek

The patient

"He has been examined by practitioners and physiotherapists. They advised him to visit the outpatient clinic."

AGE 43

EDUCATION
Business administration

PROFESSION
Tom is accountant for a local company that sells different plastic products. Ha has always been an active amateur volleyball player. When he was 29, he twisted his knee.

Next to volleyball, Tom likes to cycle with his wife in the weekends. He also enjoyed cycling to the office if it is not raining. Since the pain returned, he often takes the car. Also during volleyball and trips, he feels like his knee problems restrain him.

'He twisted his knee when he was 29. Since 5 years, he started to experience pain again."

Tom did not have much knee problems untill 5 year ago. Since then, it hurts every time he walks. Practitioners, physiotherapists and sport doctors were not able to help anymore and advised to visit an anaesthesiologist in a pain outpatient clinic. Here, at the outpatient clinic, he was also examined by a group of different specialists.

He believe the diagnoses process for this symptoms takes very long. He had to see multiple doctors, who do different tests. He spends a lot of time waiting for results, before he received treatment.

3.1.3 User requirements

In can be concluded from the previous section that the anaesthesiologist is the most imporatant stakeholder. Next to this, their user profile has been sketched. More detailed needs and expectations regarding an app that will support them in pain screening, have been identified through an interview with the anaesthesiologist I. Krabbenbost. A set of user requirements can be derived from this interview.

- The app is efficient: time of required interaction has been limited.
- The app advices a preferable measurement location
- The app saves the actual measurement location
- The app saves the threshold value generated by the Ambustims QST automatically
- The app saves data per patient
- The user is able to reject measurement values
- The app saves the pain score on a pain scale of 1 tot 10
- The app shows the development of this pain score
- The app present the results visually
- The app presents results in such a manner that it is meaningful for the health-care professionals
- The app provides a checklist of symptoms which are a signal of neuropathic pain
- The user can mark the patient's pain symptoms in this list
- The app is visually attractive
- All measurement values will be saved (so including rejected values) in a data file
- Data can be exported to electronic health records to prevent data lose in case of damage of the device or system
- It should only allow access to the patients data by authorised users
- The app displays only one piece of identifiable data, so the user can confirm that the app displays the correct page

3.2 Brainstorm

To implement all the different needs and goals of the stakeholders in the project, there has been brainstormed about different design aspects. Three main design challenges can be distinguished: authentication, data visualisation and the user interface of the app. This section will describe the brainstorm process of these aspects and the advantages and disadvantaged of the different ideas.

3.2.1 Authentication: log-in procedure

The stored data of the patient is personal information. Therefore, it should not be available for unauthorised parties. It should especially be protected against unauthorized individuals who open the app, no matter their intentions. To secure the information against this, there has been brainstormed about several log-in procedures. The procedures and there pros and cons are described in this section.

1. Log-in with a bar-code or QR-code

A bar-code or QR code will be printed and included in the electronic health record of the patient. This code will the a physical key which allows access to the page of a specific patient. At first, the doctor (who is already registered in the system) will log-in trough a log-in name and the corresponding password. To see the data of a certain patient, the code will have to be scanned, using the camera on the tablet.

Pros	Cons
No personal data needed from the patient to log-in.	The electronic health records needs to be available every time.
Dont have to remember passwords or codes for every individual patient.	Registration procedure takes time: the physical code needs to be created, then send to electronic health records and/or be printed.
The procedure is quick.	
It is easy to log-in	
A physical key in combination with a password has a high security level	

2. Log-in with the patient card, ID-card or insurance card

A card with a certain code which correspond to either a patient card number, ID-card number or insurance card number. This, in combination with a log-in name and a password for the doctor, allows access to the patients information.

Pros	Cons
The patient's health record is not needed to log-in.	If the patient forgets the card, there cant be logged in.
A second password provides extra security.	It is less secure than the log-in procedure which combines a password with a physical key.

3. Scanning a card with a chip.
Several cards work with a chip. One of these can be used as the physical key to log-in to the system. This physical key, in combination with the correct user identification (log-in name and password), allows access to the patient data. Another option is that the doctors will log-in with their card and the patient with a password.

Pros	Cons
Secure: physical key in combination with a password.	If the patient/doctor forgets the card, there cannot be logged in. If the doctor has to remember or write the password for every down. Harder to share patient with other doctors.

4. Log-in with only password.
The password and log-in name will allow access to the system. The doctor only has access to the patient he/she registered (unless permission is given). After log-in, there can be chosen from a list of patient, which can be searched for with either the name or patient number.

Pros	Cons
It is easy to log-in It is easy to register The system can be accessed without patient or the patients health record The doctor does not have to remember passwords for every individual patient. Easy to share patient profile with multiple users.	Not as secure as with physical keys. You need identifiable data from the patient.

5. Log-in with password and patient code
The password and log-in name per doctor allows access to the system. The page of a certain patient can only be access with the patient number as a password. Only the doctor who registered the patient has access to it (unless permission is given).

Pros	Cons
It is easy to register	You need identifiable data from the patient.
The system can be accessed without patient or the patients health record	Not as secure as a log-in procedure with physical keys.
Dont have to remember passwords for every individual patient.	
Easy to share patient profile with multiple users.	
More secure than a password for only the doctor	

Log-in method 1 and method 5 are the best options, with respect to the amount of pros and cons. The first idea, which uses a bar code or a QR code, has been selected for further development. This method is more secured and faster that method 5. Nevertheless, it is equally easy to use.

3.2.2 Data visualisation

The section *User requirements*, addresses what medical data anaesthesiologists want to include in the app. These have to be visualised in clear overviews. Several data visualisation are drawn, to visualize the possible presentations of this data.

Figure 3.3 shows different ways to visualize the pain threshold or pain scale individually. Visualisations 1 and 3 are only suitable for the pain scale, because only one value is shown. Therefor these visualisations are not suitable for visualizing the development of the pain threshold.

However, the pain score and the pain threshold are different aspect of a pain screening. Rating pane of a scale from 1 tot 10 represents the patients' personal experience of their pain, which is subjective. The pain threshold is a measurable phenomenon, which is more objective than a pain score. It is interesting to see the development of both values relative to each other. If a patient's pain threshold improves, but the patient does not experience less pain, other medication may be prescribed. In order to relate mutations of the pain score or pain threshold, some indication of new medication and treatment is desirable. Figure 3.4 shows data visualisations which present the pain score and threshold in one graph. Indications of a new medication or treatment, further referred to as *interventions* are also included in figure 3.5 and figure 3.6

To indicate a location on the body, a model of a human body could be used. A sketch of 2 different models can be seen in figure 3.7. Model 1 in this figure, is a more realistic model then model 2. For indicating both the location of the pain (figure 3.8) and the measurement location on the model (figure 3.9), several methods have been considered. Option 1 indicates the measurement location by a selection which is specified by the grid lines on the body. Option 2 indicated the location with a "target" or "aim" icon. Option 3 indicated the measurement location with real-sized representation of the electrodes placed on the skin.

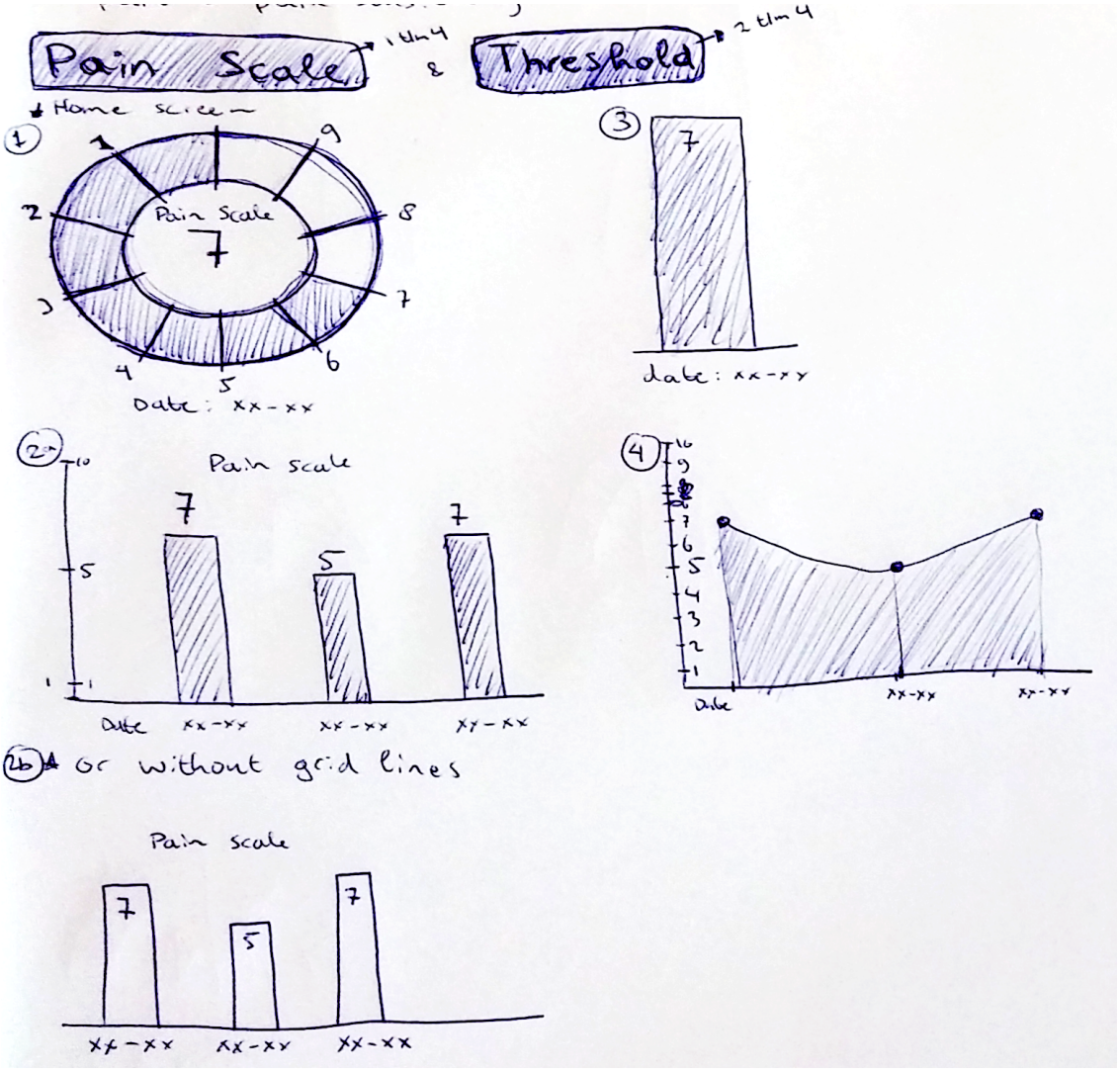


Figure 3.3: Sketches of data visualisations to present the pain scale or the pain threshold individually.

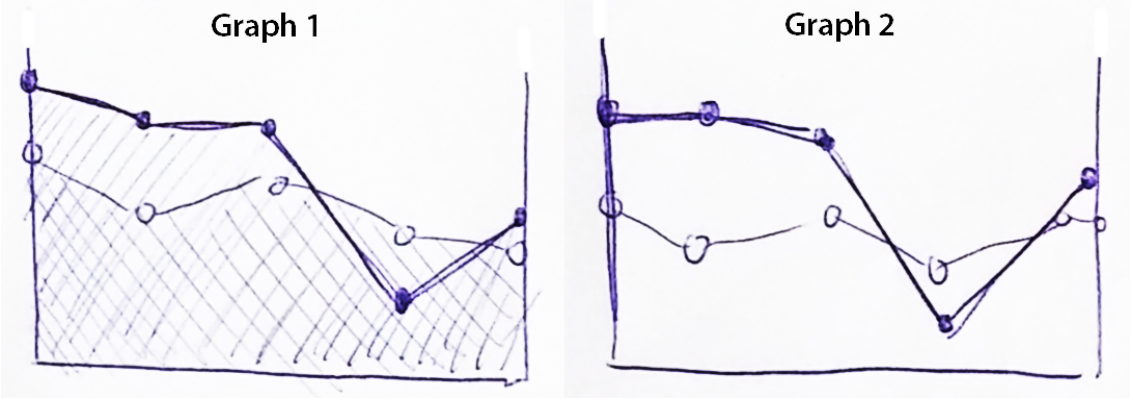


Figure 3.4: Two different designs of a line graph which represents the development of the pain score and pain threshold, in relation to each other.

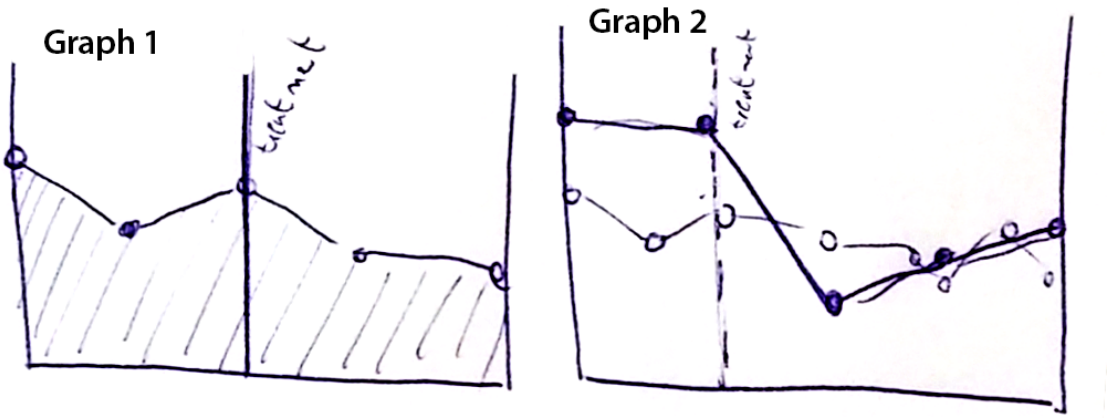


Figure 3.5: Two examples of how interventions, such as new medication or treatment, can be visualised in relation to the pain score or threshold.

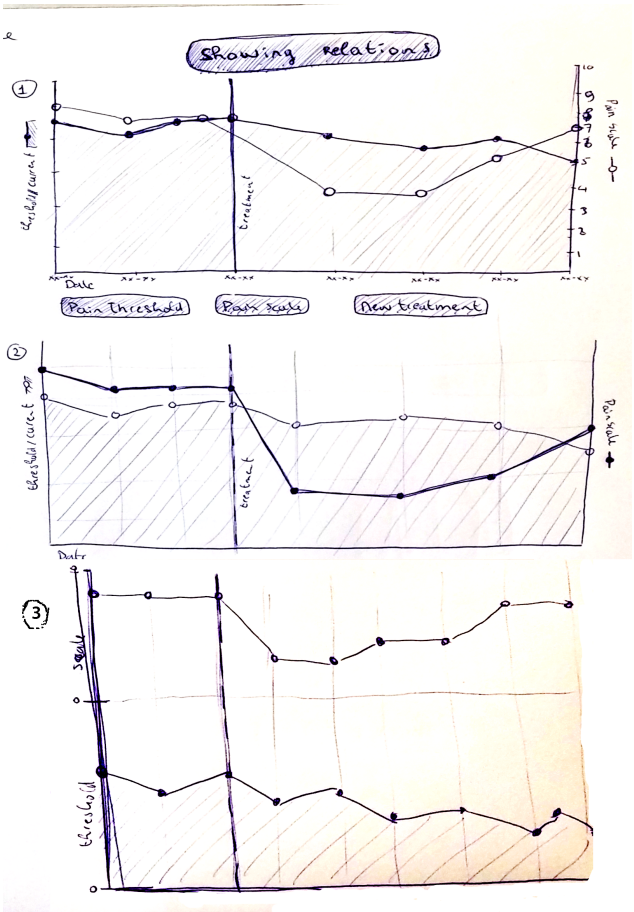


Figure 3.6: Three graphs which elaborate on visualising the relation between the development of the pain score, pain threshold and interventions.

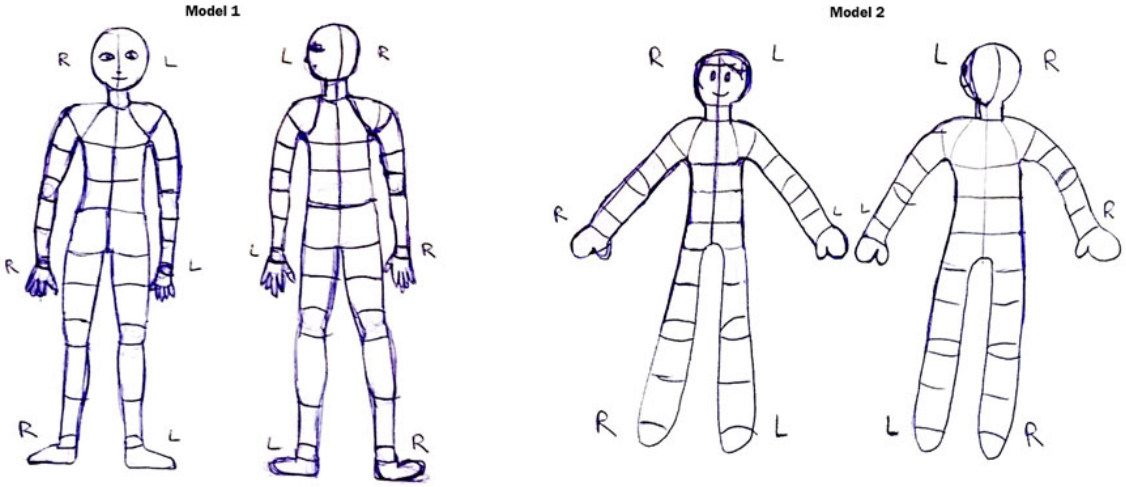


Figure 3.7: Two different visualisation of the human body.

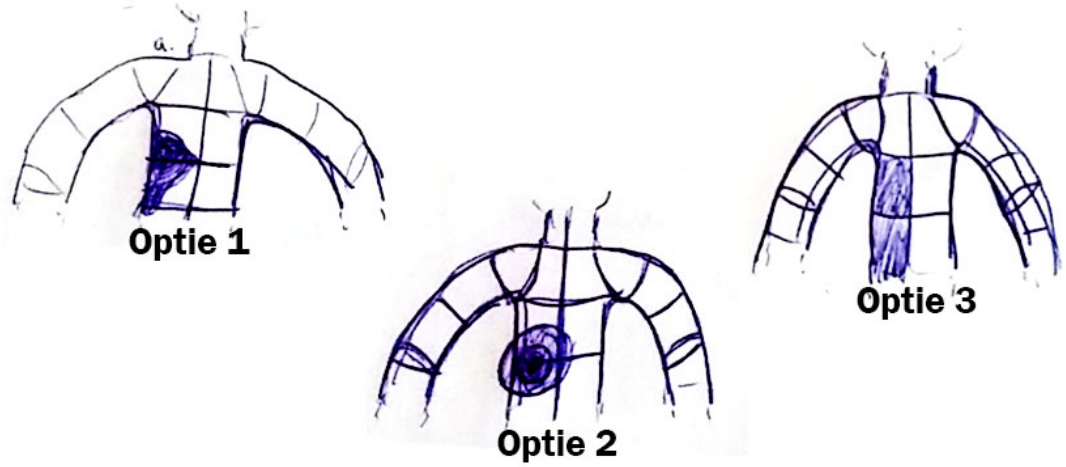


Figure 3.8: Indicating the location of the pain.

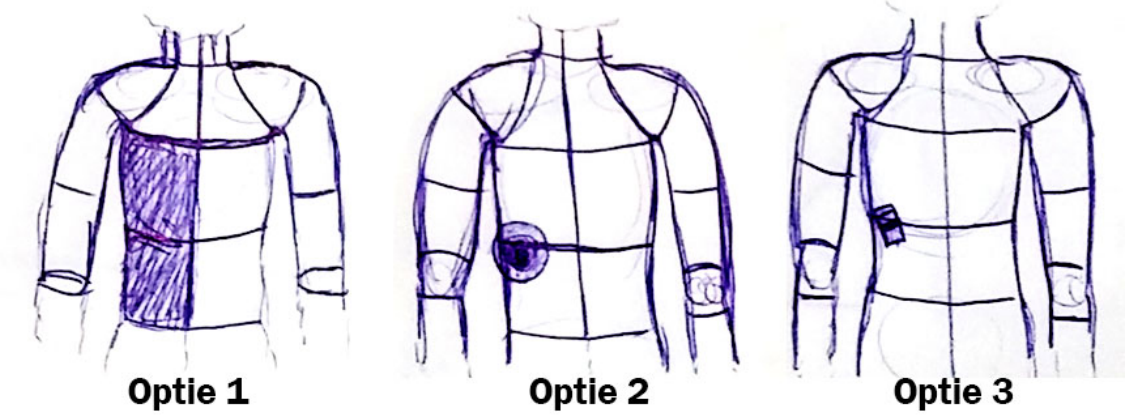


Figure 3.9: Indicating the location of the pain.

Anaesthesiologist Krabbenbos has been asked to evaluate these visualisations. Images of the possible presentation of data were shown and there has been asked which visualisation he preferred and which visualisation his patient would prefer. The result of this conversation can be found in Appendix C.

Based on his answers, the following can be concluded. The model to represent the human body should be improved. Body parts such as the bottom of the feet and more detailed hands should be included. The current model that is used in the hospital to indicate the pain will not be used. This model has been received through a screen-shot, but the image was incomplete. The target method will be used for the final design to indicate the pain area, because is easy to use and accurate. Next to this, line graphs were considered to be the most clear visualisations. Line graphs are especially a better representation for the development of the pain threshold, because the actual values are less relevant. A graph which shows development of the pain scare pain threshold related to interventions, provides the most relevant information. Next to this, it provided a complete overview in one visualisation. Therefore, this visualisation best fits on the home page.

3.2.3 User interface design

To design the first version of the app interface, what the app should display and what actions should be included were considered. On the one hand, the app has to inform the user about the status of the patient and the results. On the other, the app is intended to add new information and measurements. After logging-in, the user can accomplish two things: gather information, such as to which patient the page belongs to and the patient' s status, and choose to perform actions, such as adding values or request more detailed information.

This resulted in the design drawing, which can be seen in figure 3.10. In the drawing, arrows which start at the button and point to the action that follows refer to a click action. Again, there had been started with a home screen, followed by the first screen and a menu bar on the left side. The menu options lead to either an event to add results of to display them respectively.

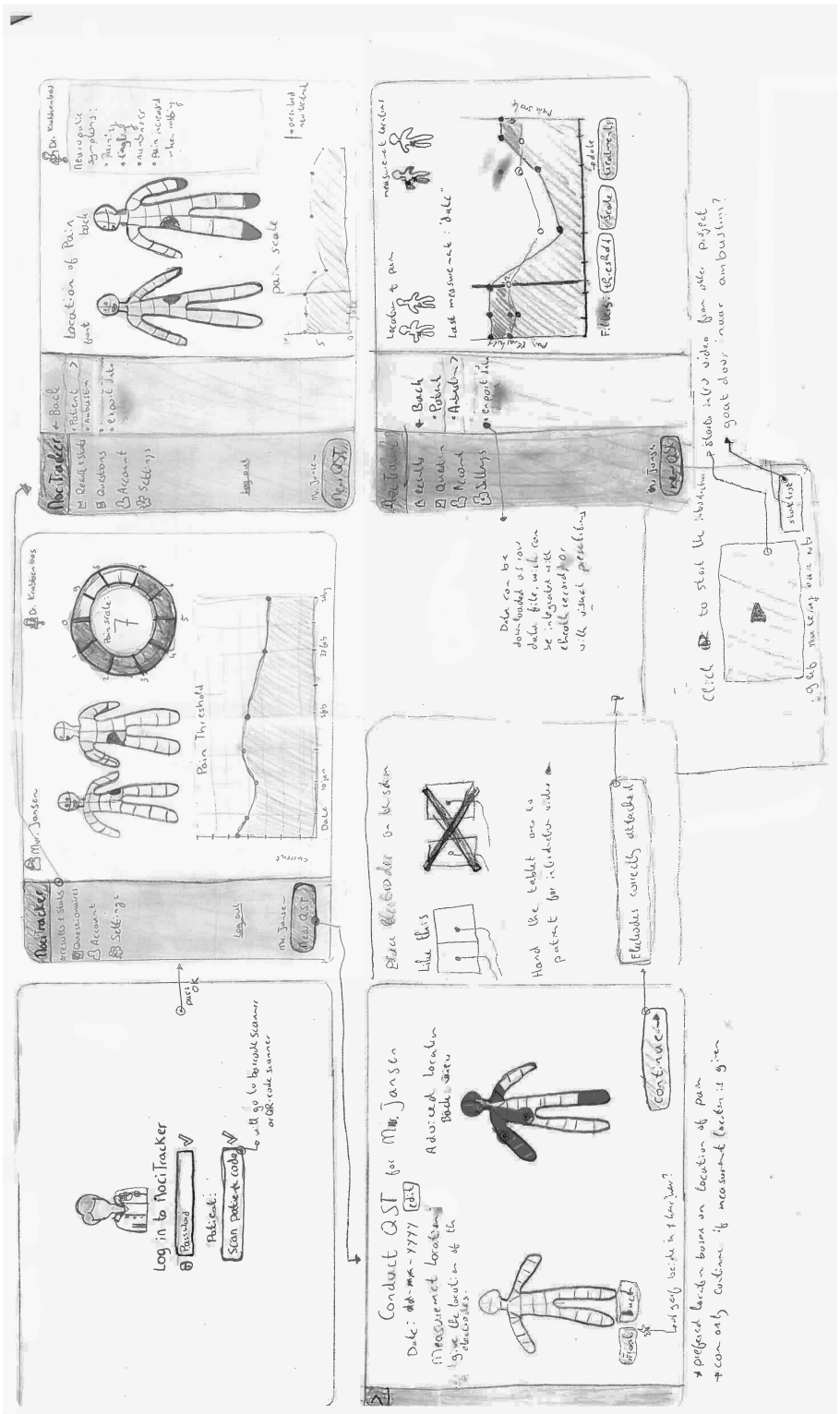


Figure 3.10: User interface design drawing, including the flow of actions.

Chapter 4

Specification

This chapter specifies the design choices of the previous chapter and narrow them down to one final design. First, to provide a clear outlook on the scope of the project and the goals of the main user, a use case scenario has been constructed. Second, the interface design presented at the end of the previous chapter, *ideation*, has been evaluated through rapid prototype testing. Finally, this will lead to specified system requirement, which are listed at the end of this chapter.

4.1 The use case scenario

This use case scenario described the main success story regarding the usage of the mobile application. The use case scenario consists of multiple use cases. Every individual use case, represents a single user-goal. Each use case will tackle the flow of action that needs to be performed, in order to achieve this goal.

Use case scenario: Track the pain development of a patient

User story

Anaesthesiologist X is involved in the treatment of patient Y. Patient Y has been diagnosed with chronic pain for 3 year and is seeing anaesthesiologist X ever since. Anaesthesiologist X is using the app 'FollowUp' to monitor the pain development of his patients. Patient Y has an appointment today with anaesthesiologist X, to check the development of her pain and how she reacts to the medication.

Primary actors

- Aneasthesiologist X
- Patient Y
- The Ambustim
- FollowUp, the app

Use cases

Basic flow of events

1. Anaesthesiologist X logs-in to the app and the page of patient Y
   (a) Anaesthesiologist X inserts log-in name and password

   (b) Log-in name and password are validated
   (c) FollowUp continues to QR code scanner page
   (d) Anaesthesiologist X scans the code of patient
   (e) Patient code is validated
   (f) FollowUp continues to the patient (home) page
2. Anaesthesiologist X checks the status of the patient
   (a) He checks the last pain score presented in FollowUp
   (b) He checks the last pain threshold presented in FollowUp
   (c) He checks the pain location presented in FollowUp
   (d) He checks the pain symptoms presented in FollowUp
3. Anaesthesiologist X adds a new pain score
   (a) Anaesthesiologist X asks patient Y to rate the pain on a scale from 1 to 10
   (b) Patient Y answers with '7'
   (c) Anaesthesiologist X goes to section in FollowUp to add the pain score
   (d) Adds 7 in FollowUp
   (e) FollowUp adds it to the data file
   (f) FollowUp presents a confirmation message
4. Measure and save a new value for the pain threshold
   (a) Anaesthesiologist X goes to the section in FollowUp to add the pain scale and starts a new measurement
   (b) FollowUp displays an advised location to place the electrodes
   (c) Anaesthesiologist X places the electrodes on the skin of patient X
   (d) Anaesthesiologist X specifies this location in the app and presses next
   (e) FollowUp adds the location to the data file and continues to the next page
   (f) Anaesthesiologist X instructs patient Y during the measurement
   (g) Patient Y presses and releases the button on the Ambustim
   (h) The value calculated by the Ambustim will be transferred to FollowUp
   (i) Anaesthesiologist X accept this value
   (j) FollowUp adds the threshold value to the data file and gives a confirmation message
5. Log-in to the page of another patient
   (a) Anaesthesiologist X opens menu and selects the option to log a new patient in
   (b) FollowUp continues to QR-scanner page
   (c) Anaesthesiologist X scans the QR-code of another patient
   (d) FollowUp validates the code of this patient and continues to the patient page

Alternative flows and error handling

1. Invalid doctor log-in name or password
2. FollowUp displays an error message
3. Anaesthesiologist X checks more details about the pain measurements

4. Invalid value for the pain score
   (a) FollowUp displays an error message
5. Invalid value for the pain threshold
   (a) Anaesthesiologist X rejects the threshold value received by the Ambustim
   (b) FollowUp saves the threshold value as a rejected value and will not display it in data visualisations
   (c) FollowUp gives a confirmation message
   (d) Anaesthesiologist X may restart the measurement
6. Anaesthesiologist X adds symptoms of the pain for a patient
   (a) Anaesthesiologist X opens the menu and goes to the section which gives the pain symptom questionnaire
   (b) FollowUp up displays the DN4 questionnaire
   (c) Anaesthesiologist X and patient Y answer the questions of the DN4 questionnaire
   (d) FollowUp saves the answers

4.2 Rapid prototyping

A mock-up of the app has been used to perform user tests with expert users. The end-user will not take part in this test, because of the limited time he is available. The user test, will evaluate the usability. The usability of the user interface design will be evaluated based on how usability is defined by Bevan [31]. He described that the effectiveness, efficiency and satisfaction as critical aspects of usability. Usability testing should cover these three aspects [32]. The user test consists of separate tasks. The tasks are based on the use cases of the previous sections. A more detailed description of the test method can be found in Appendix D.

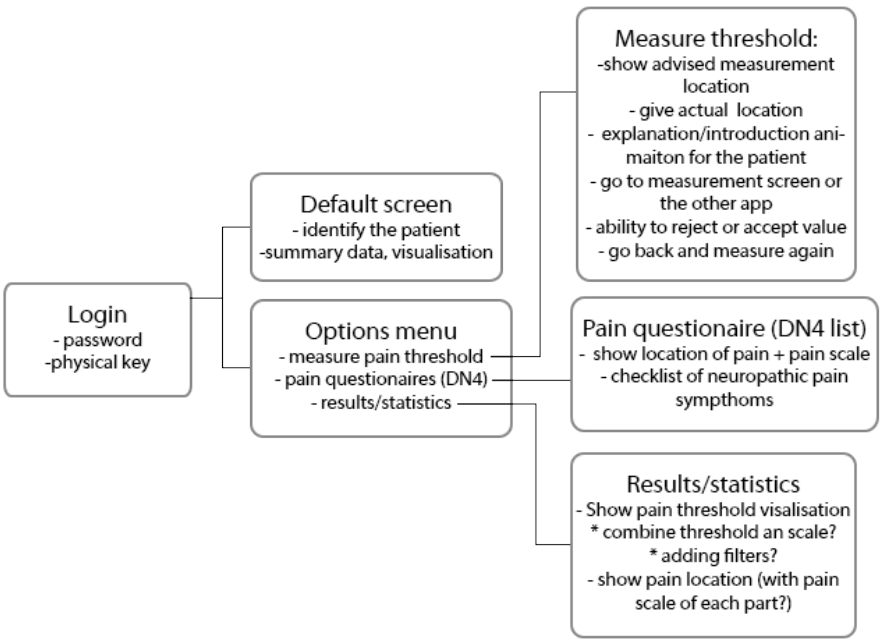


Figure 4.1: A simple system diagram which shows the transitions of the mock-up app.

Invision, an online software tool to create app mock-ups, has been used to create a rapid prototype. The mock up design is based on the design drawing in figure 3.10, at the end of the previous chapter. A simple system diagram can be found in 4.1, to to provide an outlook of two flows of actions and data transition in the app. A part of the mock-up design can be seen in figure 4.4, to show how the design drawings have been realized. This figure shows two actions: logging-in and adding performing a new measurement with the Ambustim.

The effectiveness of the app has been measured by the amount of successfully completed tasks. Every Participant had to complete 5 tasks, such as logging-in to the system or adding a new pain score. The success rate per tasks can be found in figure 4.2. As this figure shows, the participants had difficulties finding the exact threshold value and the latest pain score. This is causes by the fact that not all information that was asked for is included in the home page. Even though the participants were free navigate through the app, most did not check the other pages. The efficiency

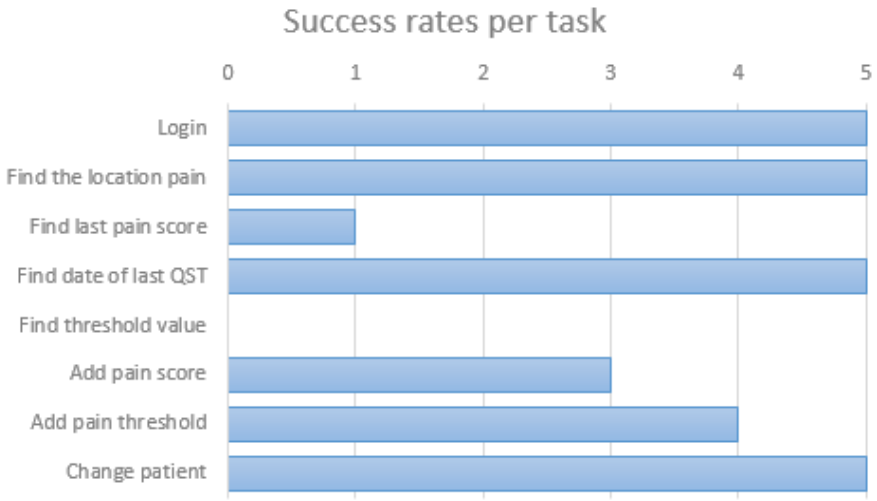


Figure 4.2: Effectiveness of the app, measures in the amount of successfully completed tasks.

has been measured as the time that participants need to complete the tasks. Figure 4.3 shows how much time every participant needed to complete an individual task. This figure also shows how much time an advanced user needs to complete the same task. The advanced user is familiar with the system and knows the shortest path to complete the tasks. Comparing this to the time that participants need, bottlenecks and confusing pathways can be distinguished. By comparing the average time of the participants with the advanced user in figure 4.3, there can be concluded that the the paths which lead to adding pain scores and gathering information was unclear for most users. These flow of actions should be improved.

The usability of the app is below average. As the System Usability Score (SUS) has a logarithmic scale in the range of 1 to 100. Scores below 68 are considered as below average [33]. On Average, participants rated the overall usability of the app with a score of 59,1 on the SUS An exact overview of all the results can be found in Appendix E.

To summarise, the usability of the app needs to be improved. The efficiency and effectiveness are low. There can be concluded that the biggest need for improvements are related to finding general information, such as the pain scale and the pain, and adding new information. The first page should contain more information. Enlarging the data visualisation and especially the font size of the labels, will make the visualisation clearer and easier to read. The user satisfaction if below average. Clearer pathways, more confirmation and error messages should be added to decrease confusions.

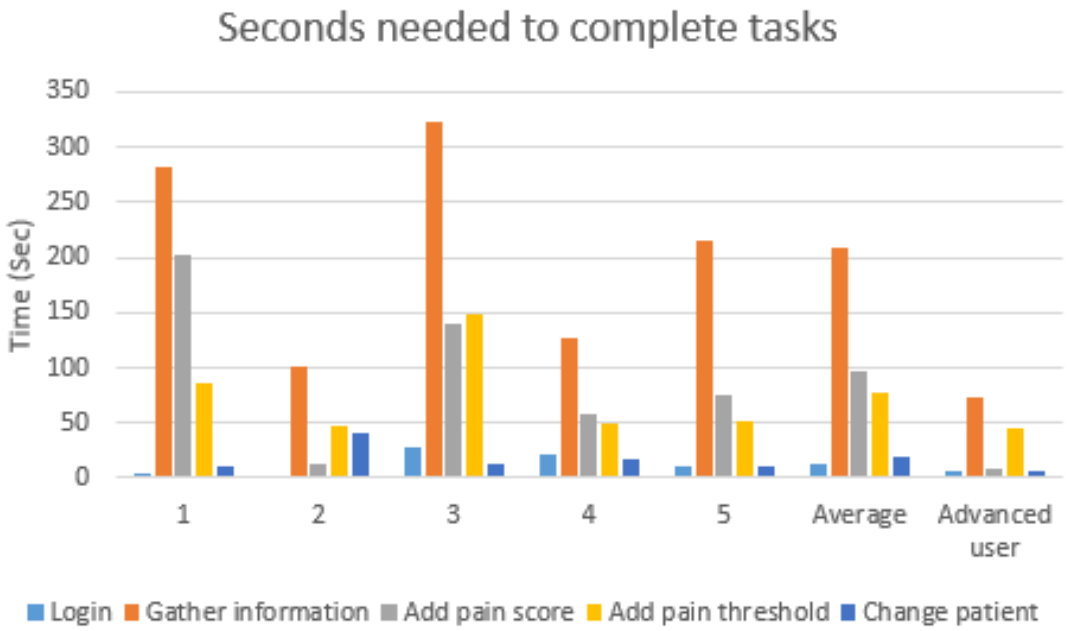


Figure 4.3: The efficiency of the app, measured in the amount of time needed to complete a task.

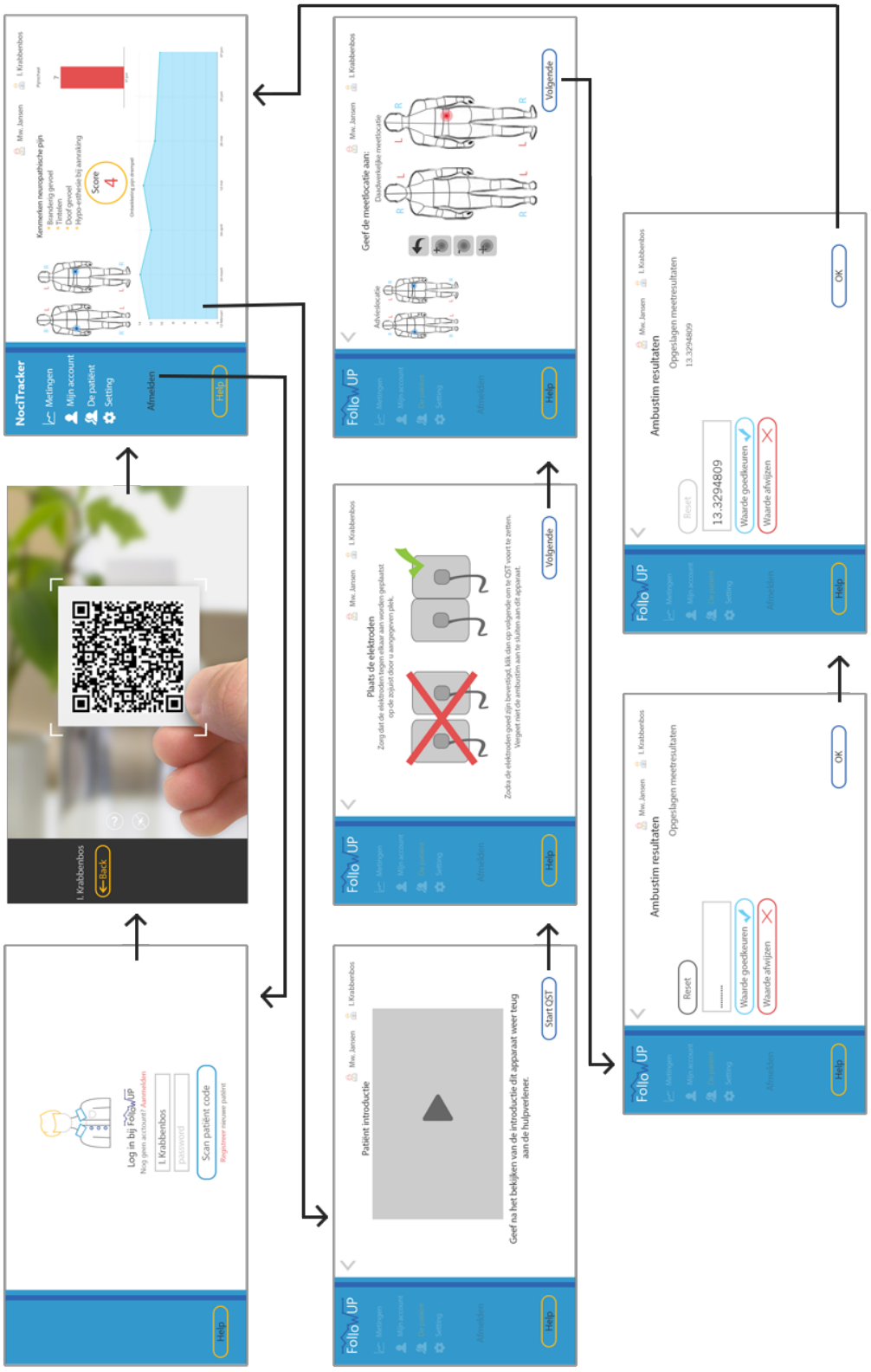


Figure 4.4: Mock-up interface. Two flows of actions are visualised: loging-in to the system and adding the pain threshold.

4.3 System requirements

A preliminary list of requirements van be found in the ideation phase. Through the previous sections, the stakeholders' needs and expectations, laws and regulations were determined and ideas were evaluated. In order to create the final design of the product, this chapter will provide a list of system requirements. They have been categorized in functional and non-functional requirements. The functional requirements state what the system and the user should be able to accomplish, while the non-functional requirements specify how they should be accomplished [34]. The system requirements are listed below in table 4.2 and table 4.1 The requirements have been prioritises based the Moscow method [35]. Their level of priority has also been specified in the table as high, medium or low.

Functional Requirement	Priority
It must be an android app	High
It must be designed for a tablet	High
It must support android versions 4.04 and higher	High
It must retrieved the value of the pain threshold from the app which measures this through the Ambustim	High
It must save the date and time of user actions	High
It must not save identifiable data of the patient	High
It should be possible to save the measurement location during the QST with the Ambustim	Medium
It should provide an advised measurement location for QST with the Ambustim	Medium
It should be possible to reject results of the pain threshold gained with the Ambustim	Medium
The rejected results of the pain threshold should not be presented in the graphs	Medium
It must be possible to log-in	High
It should be possible to make an account	Medium
It should be possible to register multiple patient on one doctor's account	Medium
It could be possible to add interventions for when a new treatment starts, when the treatments stops and other interventions	Low
It could guide the doctor in performing the QST with the Ambustim	Low
It could have an introduction for the patient about QST with the Ambustim	Low

Table 4.1: List of the functional system requirements.

Non-functional Requirement	Priority
It must be effective: users must be able to understand 80% of the functions during their first use	High
It must present the value for the pain threshold over time	High
It must present the value for the pain score over time	High
Logging-in must work with a name and password, in combination with a physical key	High
It must be visually attractive	High
The rejected results of the pain threshold should be saved with the reason for rejection	Medium
It should save all data in a data file	Medium
It should take maximum 2 minutes to log in to the system (from opening the app until access to the patients page)	Medium
It should take maximum 5 minutes of interaction with the system to gather all information about the patient's status	Medium
It should take maximum 3 minutes to add new information to the app (excluding the QST measurement process itself)	Medium
It could be possible to tick the symptoms related to neurotic pain of in a checklist which corresponds to the DN4 questionnaire	Low
It could introduce the patient through an introduction animation or video	Low
It could include a page which informs the doctor how to connect the device and place the electrodes through text and images	Low

Table 4.2: List of the non-functional system requirements.

4.4 Final design

This section presents the final user interface design. The user's needs and goals have been translated into system requirements in the previous section. Next to this, a draft of the user interface design has been evaluated. With this, a final design drawing has been constructed. The most important difference compared to the draft version (see figure 3.10) is an additional menu page to directs the user to the pages in which new information can be added. Next to this, more information is included in the home page.

The design drawings can be found on the next pages, in figure 4.5 and figure 4.6 This design drawing will be implemented in Android Studio in the next chapter, *Realization*.

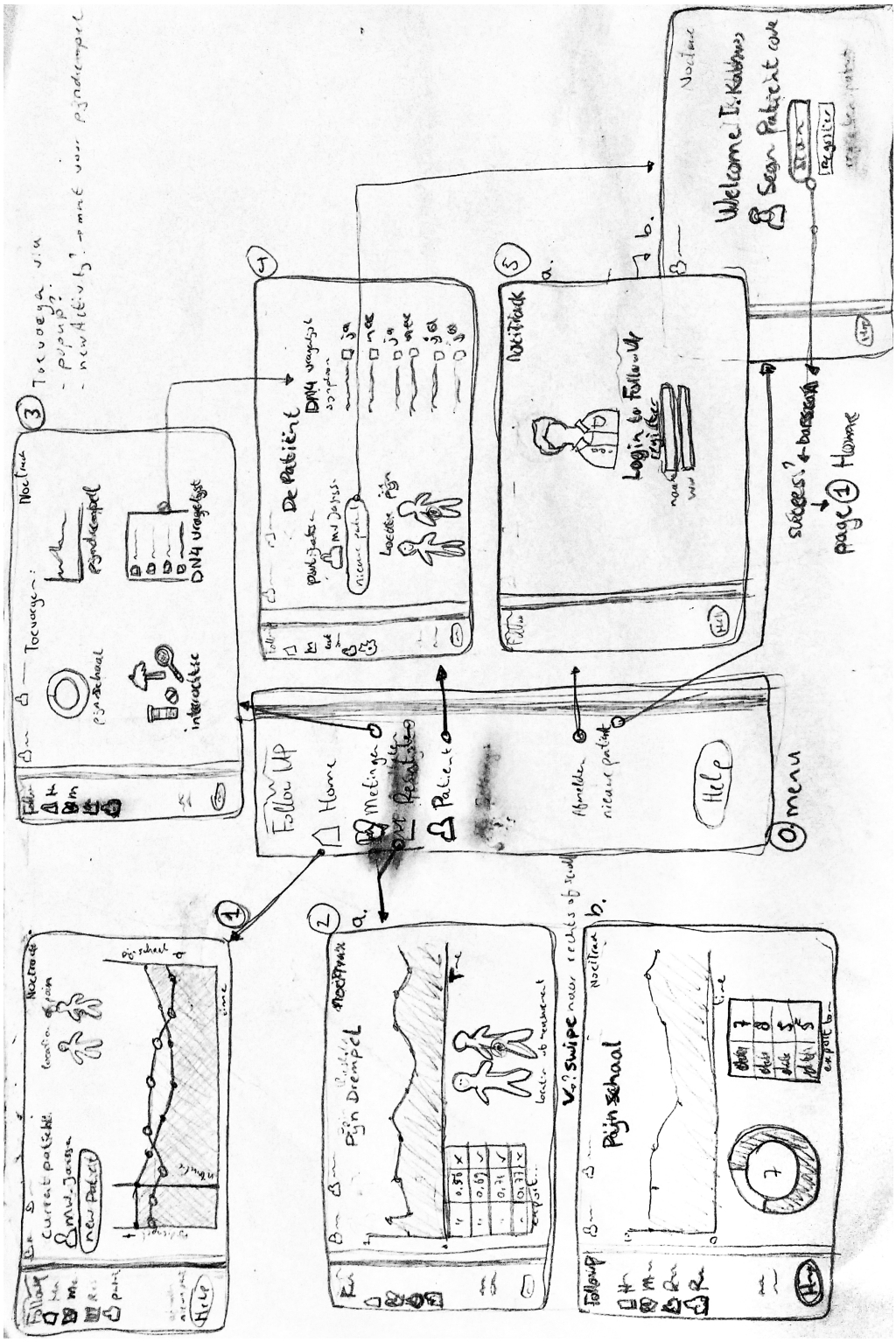


Figure 4.5: Design drawing of the final user interface, showing a system overview.

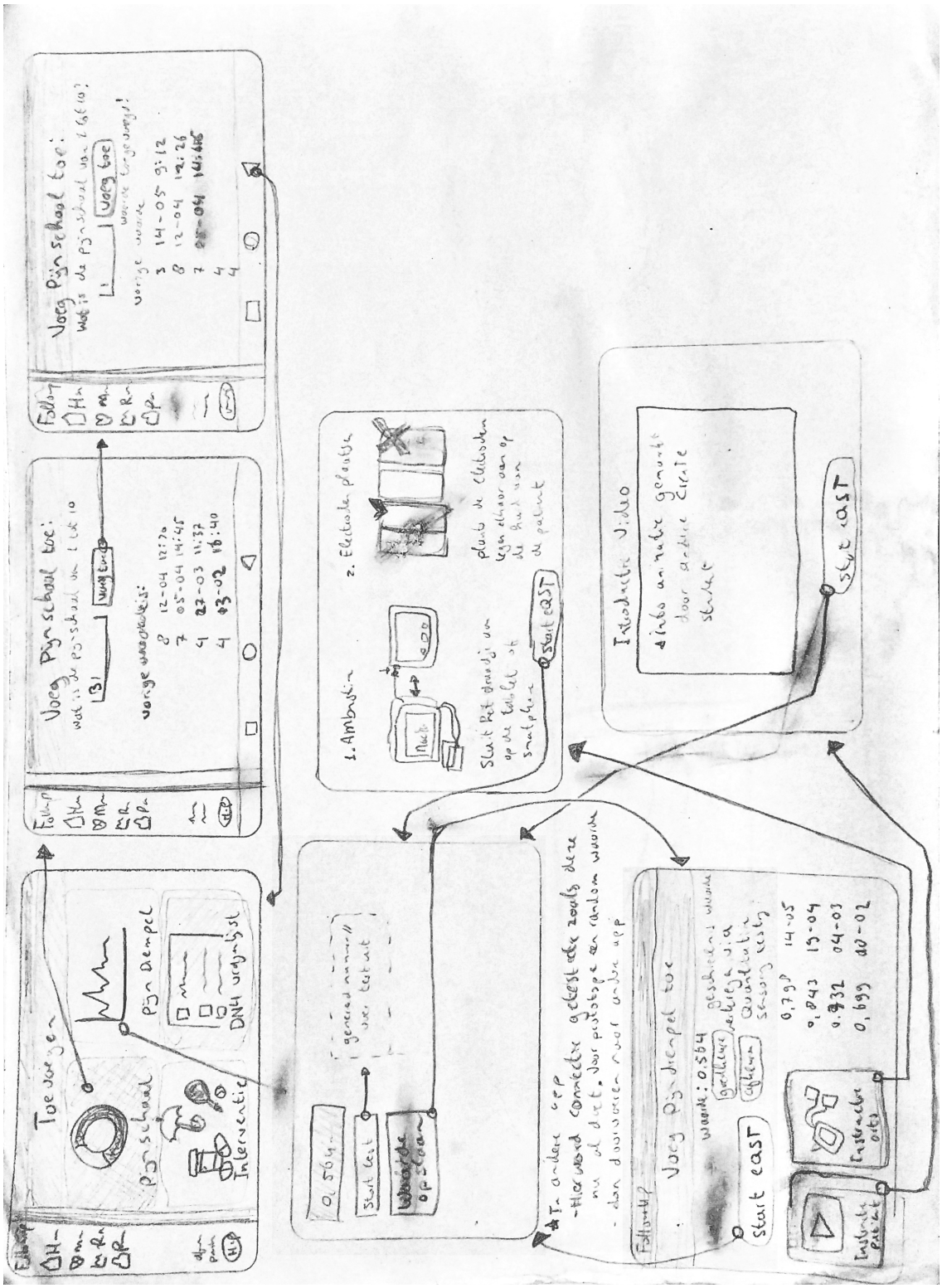


Figure 4.6: Part 2 of the final user interface design drawing, showing pages to add a new pain score and a new pain threshold.

Chapter 5

Realization

This chapter describes how the project requirements are translated into concrete design aspects. A prototype of an app to support anaesthesiologists in pain screening of chronic pain patient has been developed in Android studio. This chapter elaborates on the design and system structure of this app prototype. There will only be elaborated on design and content. For details regarding the codes of front-end pages (xml files) and back-end (Java classes) of the prototype, see the Appendix F.

5.1 Interaction design

There are several design guidelines and heuristics related to enhancing the usability of user interfaces. Usability can be described as a concept regarding subjective user preferences, , assessing how much users like the system, and objective performance, how capable user are at using it [36]. Nielsen and Molich defined 10 usability criteria as described in table 5.1. These design heuristics will be considered during development.

Simple and Natural Dialogue	Information should appear in a natural and logical order. Dialogues should provide relevant information only.
Speak the User's Language	Dialogues should be in clear words and phrases which are familiar to regular users.
Minimize the User's Memory Load	Repeat information, rather than letting the users recall part of the dialogue.
Be Consistent	Users should not have to wonder whether the same words, situation of action mean the same thing. Use existing standards where possible.
Provide Feedback	Always provide clear information regarding the system status withing reasonable time.
Provide Clearly Marked Exits	Users should always be able to exit their current state or return to the previous.
Provide Shortcuts	Explanatory actions will be cumbersome for experienced users. Create shortcuts makes the system cater both experienced and inexperienced users
Provide Good Error Messages	Error messages provide exact information regarding the cause of the problem and suggest a solution, but never criticise the user.
Error Prevention	Careful design will prevent errors from occurring in the first place.

Table 5.1: The 10 UI design heuristics of Nielson and Molich.

The Android developer website shows many relevant guidelines and tips regarding the material design of android applications. They cover 6 aspects of material design: animation, style, layout, components, patterns and usability [37]. The aspect which were most relevant for this design of this prototype are the style, layout and usability. These address concepts such as the use of primary and accent colours for the layout or feedback messages. Moreover, size and spacing standards are specified here.

To realize the design aspects in Android studio, some knowledge about programming layout design had to be gained. Therefore, The course *'Android Development for Beginners'* available on the online educational platform Udicity, were followed. It provided a good foundation regarding the layout structure, placement and positioning of view items [38].

5.2 System features

In the app, the user should be able to accomplish the following things:

- Login to the system
- Switch between different patients registered in the system
- See the previous measurement such as the pain score and results of the QST
- Perform a QST with the Ambustim and the complementary app of NociTrack
- Add other pain related information, such as the pain score and the symptoms

To translate these goals into system features and user interfaces, a system diagram has been made. This block diagram is shown in figure 5.1. It provides a complete overview of the app's functions and pages. The blocks represent pages and the content of them, represent the content of the page respectively. The relation between the pages is visualised with a line that connects the blocks. The actions users can take are presented as dotted rectangles in figure 5.1. The following sections will elaborate on the different actions.

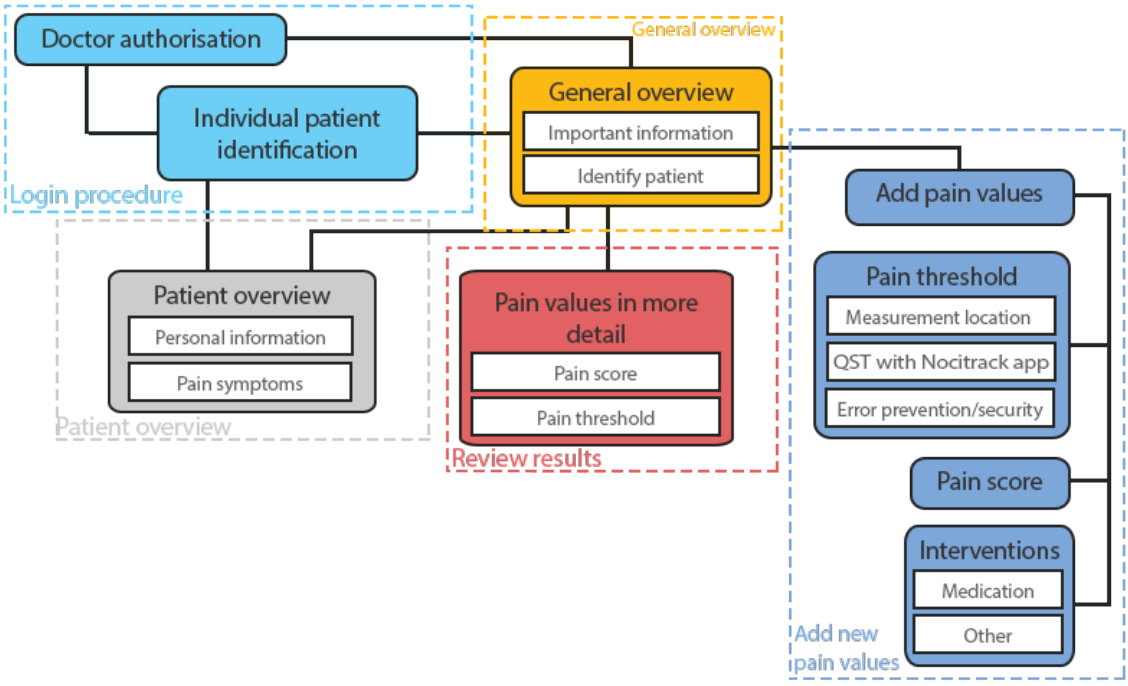


Figure 5.1: Block diagram of all the pages and their relation to each other.

5.2.1 General overview

The home page, will be the main page of this system. At this page, the most relevant information regarding the patients' condition is shown. Moreover, all possible goals that users may have, are accessible from here through a side menu. Screen number 1 in figure 4.5 shows the design drawing of this page.

Figure 5.2 shows how the design drawing has been realized in Android studio. It includes the most relevant information: the name of the patient, a graph with the development of the pain threshold and pain scale and the location of the pain. The menu can be opened by clicking on the menu icon in the upper left corner on the page.

To prevent possible errors, the following is implemented in the design. The user and patient name are shown in the left upper corner. This is to prevent users from continuing while the wrong user is logged in or another patient's data is displayed. In case the wrong patient is logged in, the user can either press the button below the patient name or select the item, *Nieuwe patint*, in the menu to scan the QR-code again.

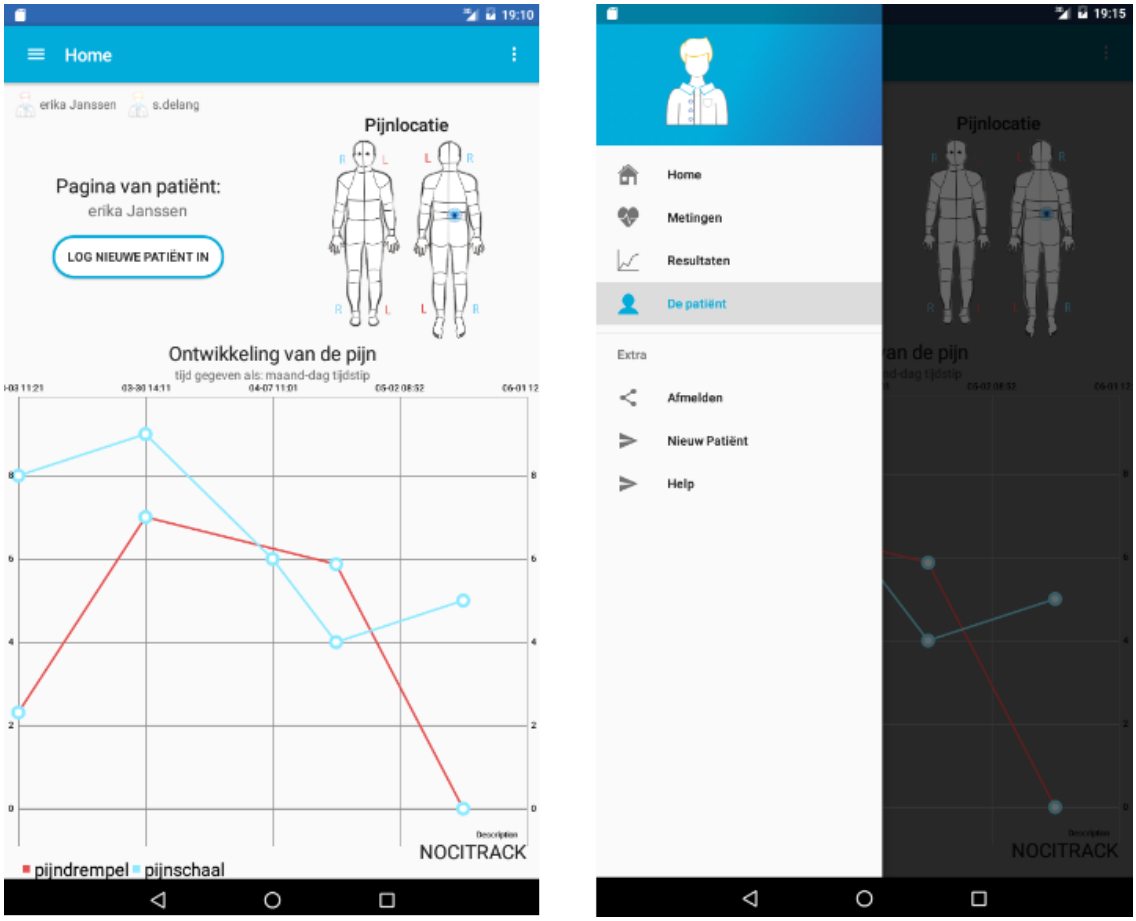


Figure 5.2: Screenshot of the home page.

5.2.2 Add new pain values

Users are able to add new pain related information into the app. The side menu item *Metingen* of the home page will enable the user to choose between 4 different options: adding a new pain threshold, adding a new pain score, add new intervention or continue to the DN4 pain questionnaire

Figure 5.3: Screenshot of the 'add measurements' option menu.

to mark/review the specific pain symptoms. This has been realized in Android Studio, as is shown in figure 5.3.

The options are represented by four image buttons. To create four equal sized click-able areas, multiple *linear layout* views are combined. *OnClickListners* methods have been set for these areas, so an action will follow after the user has clicked it. See Appendix F for more details on how this *OnClickListners* has been constructed.

Feedback is provided regarding the current status and the actions that follow, by adding a title to the page and to the image buttons. The back button can be used to return to the previous page, In case the user entered this page by accident. Android devices contain a back-button by default. Therefore, an extra back button has not been implemented in the layout itself.

The next sections describe the actions that are needed to add the pain threshold and the pain score. The DN4 pain questionnaire is a part of the patient activity, so that will be discussed in section 5.2.4, *Patient overview*. The intervention option is not functional. This option had a lower priority and has therefore been postponed because of time constrains.

Add the pain score

This section elaborates on the the actions needed to add the the pain score. A new pain score can be added, by clicking on the yellow image button in the upper right corner. This will open the page, shown in figure 5.4.

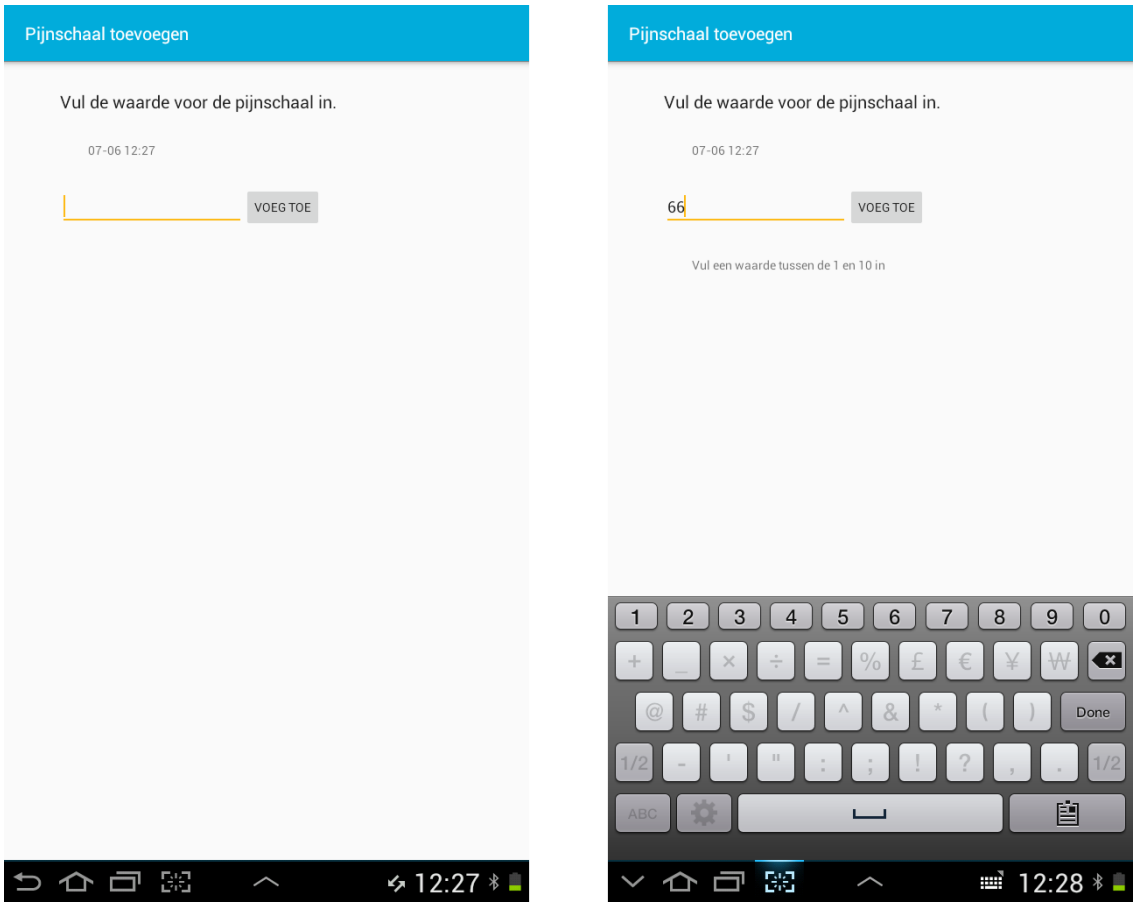


Figure 5.4: Screenshot of page to add a new pain score.

The page contains an area to add text. Since the pain score is a value on the scale from 1 tot 10, the text input has been constrained to numbers withing this range. The button below this area should be clicked to add the value. Feedback will be provided when the user tries to add a value which differs from the range. When succeeded, a confirmation message will appear.

Add the pain threshold

NociTrack developed an app to control the Ambustim and calculate the threshold accordingly. Therefore, the pain threshold will be retrieved from this app. However, the app from NociTrack was not available during the development of the prototype. Another app, which generates random numbers, has been created to replace it. The structure of the app that generates random numbers, can be found in Appendix G. The design drawing of the different steps needed for a new threshold measurement are shown in figure 4.6, in section 4.4.

The first step for adding the pain threshold is specifying where the measurement took place. This page can be found in figure 5.5. Here, the user can add the location at which the electrodes are placed. The app even provides an advised location to place them. However, this aspect of the prototype is not functional. This action is replaced by static images.

After this has been specified, the pain threshold measurement can continue. The user is able to enter the NociTrack app, perform the QST with the Ambustim, and return to the app to save the threshold value. The threshold that has been received, is displayed. The user is able to reject or accept this value. A screenshot of this page can be found in figure 5.6.

To prevent mistakes of the user, multiple confirmations are implemented. First, the threshold

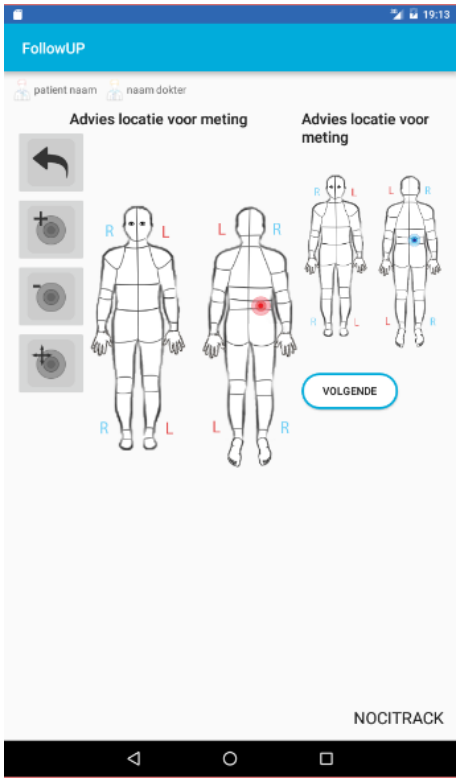


Figure 5.5: The page in which the measurement location has to be specified.

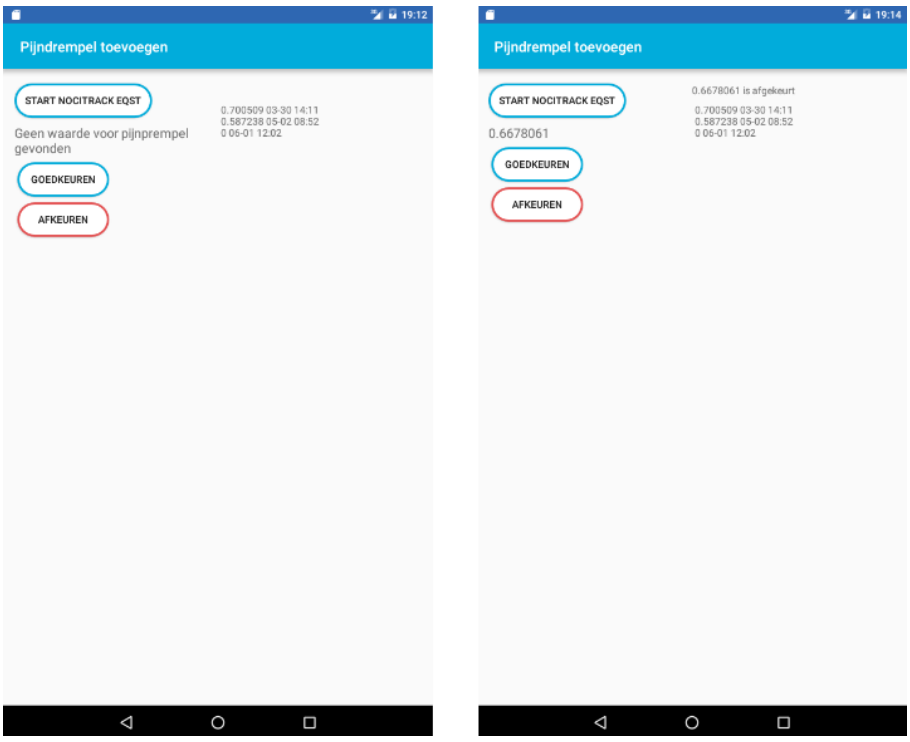


Figure 5.6: Page to save the pain threshold.

value will be send to the app, the first confirmation. Than, the user has to review this value again in the app to either accept or reject it, the second confirmations. The last confirmation, is given after accepting or rejecting, by providing a success message.

5.2.3 Review results

The home page already allows user to see some information, but more details may be desirable. Therefore, an extra page to review the pain score and pain threshold results in more detail has been considered. Because all the information could not be visualised clearly when included in one page, it has been separated into two categories. The interface is shown in figure 5.7.

To realize this in Android studio, one activity and two fragments are created. Fragment represent one action of portion of an activity [39]. Multiple fragment can be combined in one activity, for example to switch between multiple layouts. In this case, the fragments include most of the visible content. More details on what is included in these fragments can be found in Appendix F. There can be switched between the fragments with a swipe motion.

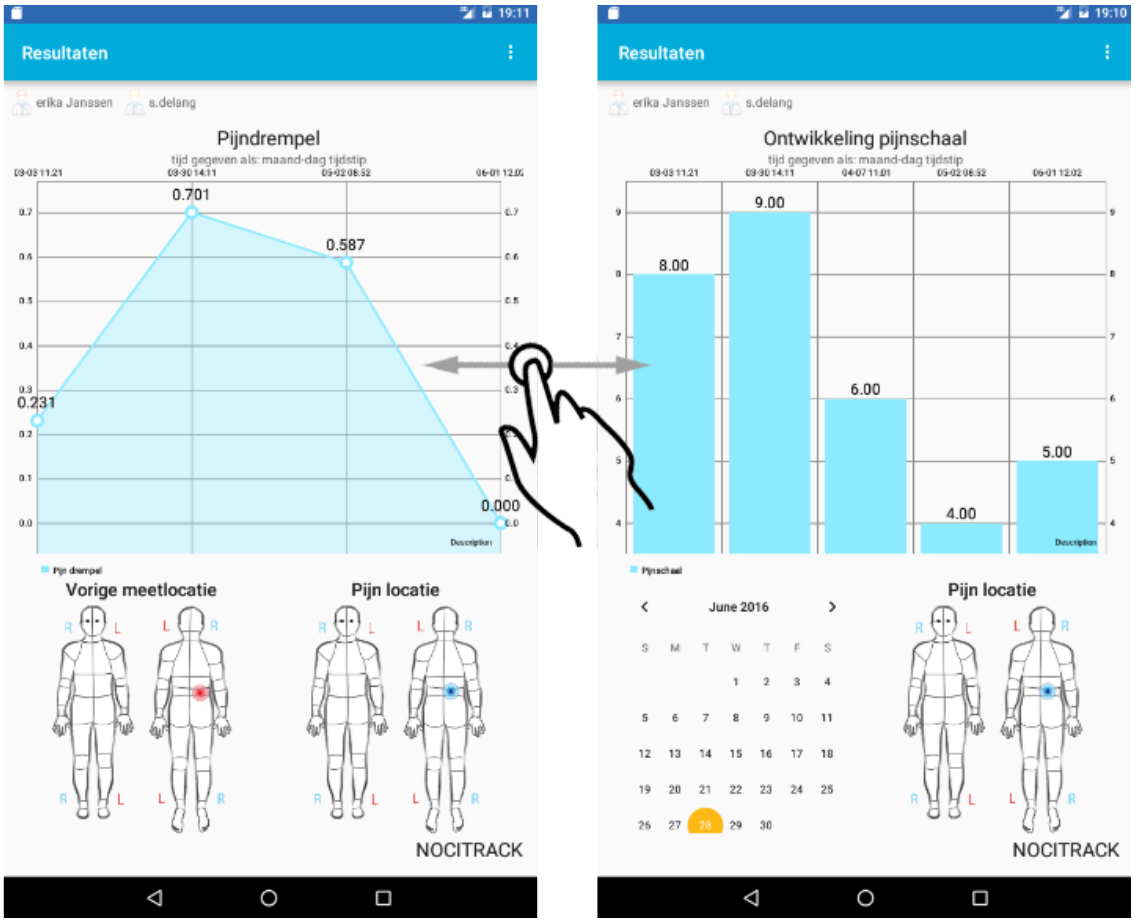


Figure 5.7: Screenshots of the two results fragments.

5.2.4 Patient overview

The patient page provides an overview of more detailed patient information and is shown in figure 5.8. It includes the name of patient and detailed regarding the pain symptoms. The pain symptoms, are specified according to the DN4 questionnaire. The questionnaire items are accompanied by check-boxes. These boxes can be dis- or enabled to answer the question stated

above it. Because this option has a low priority, these boxes are not fully operational. Even though check-box state will not yet be saved, it does show how this option will be added in the future.

Hospitals may use different pain questionnaires, because there is no standard questionnaire for this field. Nevertheless, the project supervisor and anaesthesiologist Krabbenbos advised to use this questionnaire.

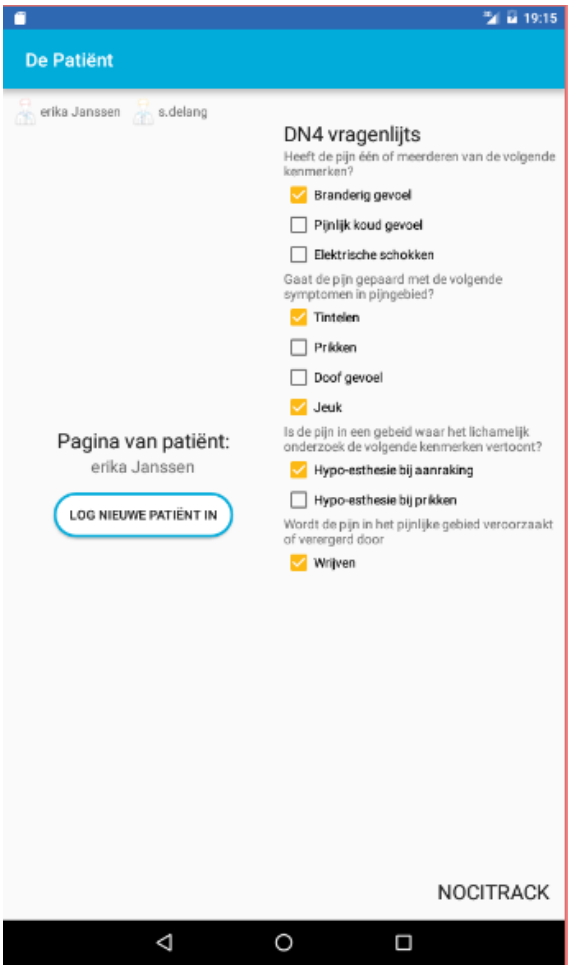


Figure 5.8: Screenshot of the patient page.

5.2.5 Log-in procedure

Multiple doctor account may be created. However, they do not have the authorisation to review information of all the patients which are registered. Therefore, the system needs a log-in procedure to identify the doctor and the patient separately. The procedure includes two steps: the doctor logs-in with an log-in name combined with a password, followed by scanning the patient's QR-code to request a certain patient page. Only when the doctor and patient code match correctly, there can be continued.

This has been realized in android studio with two separate activities (see figure 5.9). The first activity, is the doctor log-in. The second is the patient verification. For this prototype, the log-in name and password are not specified. Nevertheless, empty field will not be accepted. When either the log-in name or password do not contain input, an error message will appear. If the log-in name and password input are valid, there will be continued to the next activity. For the patient verification activity, the QR-code has to be scanned. This has been implemented through

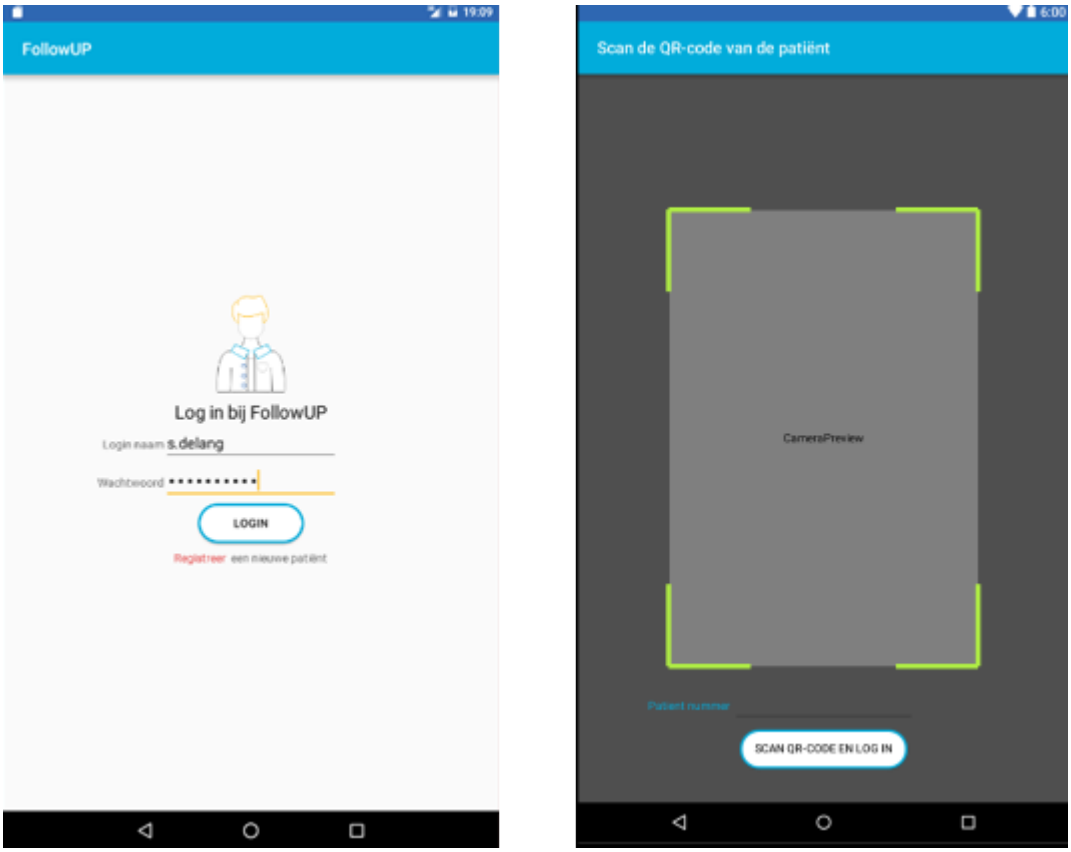


Figure 5.9: the log-in procedure realized in Android studio.

a bar-code and QR-code scanner library for Android Studio. The procedure is easy and fast: the activity start the camera and will scan the image for a QR-code. The prototype now also included a field to enter the patient number to continue. This has been done to allow testing in case a QR-code is not available.

Error handling has been considered by providing feedback messages to confirm actions. For example, after the doctor logged in successfully, their name will appear in the next activity. Also for the patient log-in, the phrase "login successful" will appear before continuing to the next activity. Next to this, separate error messages and advice for invalid log-in names or password will be provided.

Chapter 6

Evaluation

In this chapter, the prototype will be evaluated with respect to the system requirements (see chapter 4.3). The evaluation involves experts and end-users. They were asked to perform 5 different tasks. These tasks can be found in table 6.1. The usability of the prototype will be evaluated by measuring the effectiveness, efficiency and user satisfaction. The effectiveness will be measures by the amount of successfully completed tasks. The efficiency will be measured as the time users need to perform a task. To measure the user satisfaction, the System Usability Score (SUS) will be calculated. SUS is a score from 1 to 100 which is determined by a list of 10 statements which need to be rated on a likert scale [33]. Nielson and Molich have proven that usability testing can best be performed with 3 to 5 participants [36]. 4 user (2 experts and 2 end-users) participated in this test. This chapter will conclude with to what extend the prototype meets the project requirements.

Task 1	Log in, until the patients page
Task 2	To get familiar with the patient's status, gather the following information:
Task 2a	the location of the pain
Task 2b	the last given pain score
Task 2c	the last date on which a QST measurement took place
Task 2d	the exact value for the pain threshold from the measurement
Task 3	Add the new pain score (value of 7)
Task 4	Add the new pain threshold by performing a new QST measurement
Task 5	Log-in to the page of the next patient

Table 6.1: List of tasks during the user evaluation

More details about the methodology of the user test can be found in Appendix I. The following paragraphs will summarize the results of the user evaluation. The exact results of the user test can be found in Appendix J.

For effectiveness, the user interface scores rather high. In figure 6.1, the amount of successes for every task can be seen. As the figure shows, almost every task has been completed successfully by every participant. Only task 2d, scores lower compared to the other tasks. This seemed to be caused by a design mistake. The app contains multiple graphs which shows the pain threshold. One of this graphs shows the incorrect value for the pain threshold. Reconsidering this inconvenience, the success rate of task 2b scores a 100% (see figure 6.2).

In the non-functional requirements, some efficiency standers were specified. They state that logging in should not take longer than 2 minutes, gathering a general overview of the patient situation should take maximum 5 minutes and adding new data maximum 3 minutes per item. Figure 6.3 shows that all participants were able to complete the tasks within the given time constraints specified in the requirements.

To determine what flow of actions can still be improved in efficiency, the time per task of the

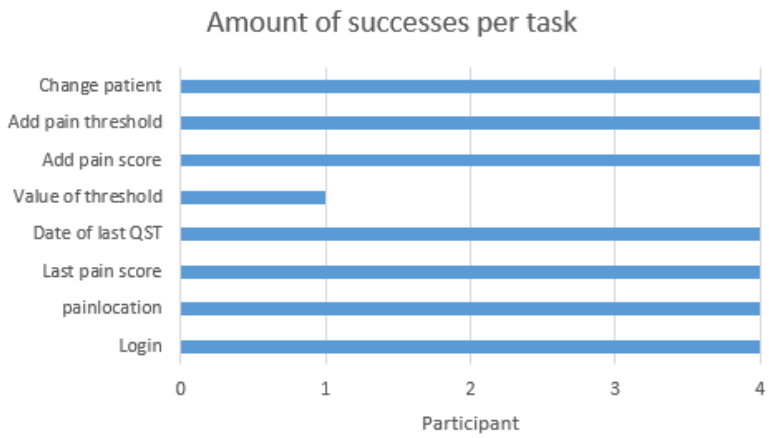


Figure 6.1: Effectiveness: the amount of successes per task

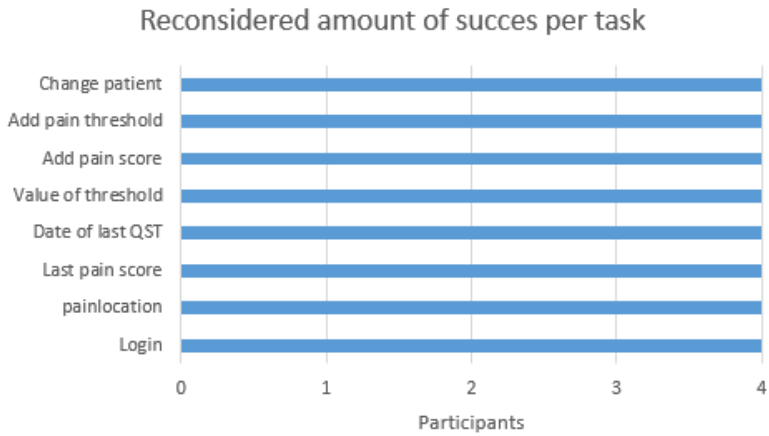


Figure 6.2: Effectiveness, reconsidering task 2d

participants are compared to that of an advanced user. The advanced user is someone who is familiar with the app and knows the shortest path to complete a task. These results can be found in figure 6.4.

These graphs shows that participants need 57 seconds to add a new pain score, which is 518,2% more than the advances user (11 seconds). This indicated this task is relatively hard to complete. For some users, it took some time to find the menu and the correct option which allows new values to be added. Moreover, one stated that the titles of the menu items do not explain their content clearly. Added to this, 2 participants clicked on another menu item first. It can be concluded that it is not clear where to add new values. Next to this, all participants needed to be reminded of whether they had to add the pain score or the threshold.

Adding the pain threshold took the participants 250% longer (80 seconds) than the advances user (32 seconds). Most participants were not confident about their actions. One stated that is was unclear what was expected when the app asked for the location of measurement. Another was not sure when to continuing to the following step. Participants were confused when they pressed accept/reject after receiving the value. It was not clear whether this step had been completed successfully.

Regarding the user satisfaction, the participant rated the app with a SUS of 85. This is an increase of 37,1 point compared to the rapid prototype test (with a score of 59,1). As is explained

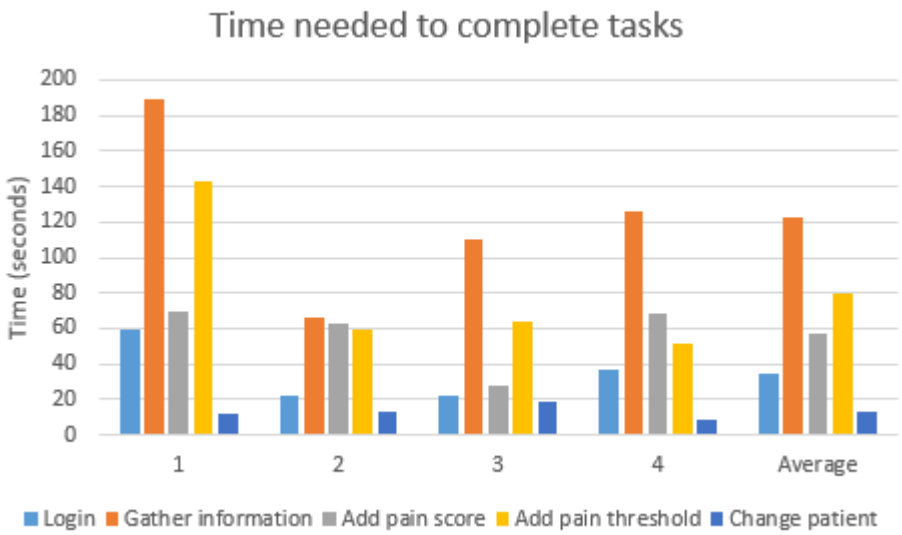


Figure 6.3: Efficiency scores per task

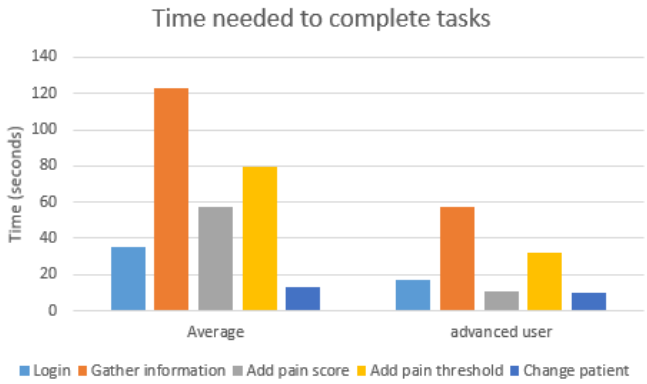


Figure 6.4: Average time per task of participants compared to that of an advanced user

in section 4.2 and Appendix D, the average SUS score is 68. With a score of 85, the satisfaction is above average [33].

The user tests tackled a few requirements. Because all tasks have been completed successfully and withing the given time constrains, there can be concluded that the app is both effective and efficient. Finally, table 6.3 and table 6.2 provides all the project requirements and the evaluation of them. With this there can be concluded to what extend the prototype meets the project requirements.

Functional Requirement	Priority	Success	Explanation
It must be an android app	High	Yes	The app has been developed in android.
It must be designed for a tablet	High	Yes	It has been designed for a tabled and runs on a tablet.
It must support android versions 4.04 and higher	High	Yes	The tablet used for perform user tests runs on android version 4.04.
It must retrieved the value of the pain threshold from the app which measures this through the Ambustim	High	Yes	The prototype is able to receive a value from another application on the device.
It must save the date and time of user actions	High	Yes	It saves the date and time when either the pain threshold of pain score is being saves
It must not save identifiable data of the patient	High	Yes	The prototype does not save identifiable data. The identifiable data it shows has been received from the QR-code and is not saved the application itself.
It must be possible to log-in	High	Yes	The prototype requires access through a log-in procedure to enter it. However, no real user database has been implemented.
It should be possible to save the measurement location during the QST with the Ambustim	Medium	No	The prototype does include images which represent this option, but it is not possible to save the real location.
It should provide an advised measurement location for QST with the Ambustim	Medium	No	The prototype does provide an advices measurement location. However, this is just an image so it is not based on real data
It should be possible to reject results of the pain threshold gained with the Ambustim	Medium	Yes	It is possible to either accept or reject the received values for the pain thresholds.
The rejected results of the pain threshold should not be presented in the graphs	Medium	Yes	The rejected values are not visible in any graph within the app.
It should be possible to make an account	Medium	No	This has not been implemented in the prototype.
It should be possible to register multiple patient on one doctor's account	Medium	Yes	The prototype provides a link to create a new bar-code for a new patient. However, different doctor accounts have not been implemented in the app.
It could be possible to add interventions for when a new treatment starts, when the treatments stops and other interventions	Low	No	The app contains a button to add an intervention. However, it is not functional.
It could guide the doctor in performing the QST with the Ambustim	Low	No	This has not been implemented in the prototype.
It could have an introduction for the patient about QST with the Ambustim	Low	No	This has not been implemented in the prototype.

Table 6.2: Evaluation of the functional requirements

Non-functional Requirement	Priority	Completed?	Explanation
It must be effective: users must be able to understand 80% of the functions during their first use	High	Yes	The success rate varies from 93,8% to 100%.
It must present the value for the pain threshold over time	High	Yes	A line graph which shows the development of the threshold over time is present in the prototype.
It must present the value for the pain score over time	High	Yes	A line graph and a bar graph show the development of the pain score over time, is present in the prototype.
Logging-in must work with a name and password, in combination with a physical key	High	Yes	Logging-in works with a name and password, in combination with a unique QR-code which for every patient. However, the name and password are not yet linked to a user profile.
It must be visually attractive	High	Yes	The participant all agreed or strongly agreed with the statements " I think that I would like to use the system frequently".
The rejected results of the pain threshold should be saved with the reason for rejection	Medium	No	This function has not been implemented in the prototype. However, the action has been described in the design drawing and the mock-up app.
It should save all data in a data file	Medium	Yes	It saves the pain score, pain threshold and the times of measurement in an local MySQL data file.
It should take maximum 2 minutes to log in to the system (from opening the app until access to the patients page)	Medium	Yes	Every participant of the user test was able to log in within 2 minutes.
It should take maximum 5 minutes of interaction with the system to gather all information about the patient's status	Medium	Yes	Every participant was able to find 4 different aspects regarding the patient's situation within 5 minutes.
It should take maximum 3 minutes to add new information to the app (excluding the QST measurement process itself)	Medium	Yes	Every participant of the user test was able to add new information to the app in within 3 minutes.
It could be possible to tick the symptoms related to neurotic pain of in a checklist which corresponds to the DN4 questionnaire	Low	Yes	The patient page of the app includes the DN4 questions with check-boxes. However, the tick action of these boxes will not be saved.
It could introduce the patient through an introduction animation or video	Low	No	This has not been implemented in the prototype.
It could include a page which informs the doctor how to connect the device and place the electrodes through text and images	Low	No	This has not been implemented in the prototype, but it is included in the final design drawings.

Table 6.3: Evaluation of the non-functional requirements

Chapter 7

Conclusion

Treating pain is often complex. The medical field of anaesthesiology lacks a clear clinical pathway and it is hard to measure pain objectively. NociTrack developed the Ambustim, a tool to measure the pain threshold of the patient. This portable device is controlled by a mobile application, which can be installed on a smart-phone or tablet. The mobility enhances the point-of-care approach of QST. The current problem on the system, is the lack in structured documentation of the measurement results. A prototype of an app to support anaesthesiologists during a pain screening and to follow the development of their patients pain, has been developed. To do this, three research questions are asked regarding 1) laws and regulations applicable, 2) the needs of anaesthesiologists with respect to monitoring pain development 3) realisation of an effective mobile application to support anaesthesiologist within the legal constrains.

Legal issues regarding patient privacy and security are reviewed. As personal health information of patients will be saved, this has to meet certain security standards. First of all, the data collection has to be registered a the Dutch government department regarding authorisation of personal data (*Autoriteit Persoonsgegevens*). Second, the Dutch law with respect to handling and saving personal data, *Wet bescherming persoonsgegevens* applies to all patient related data and should therefore be followed. Finally, patients need to be properly informed about their rights. For example, they have the right to refuse that this app will be used in their treatment.

In order to support the anaesthesiologist in pain screenings, they need to gather and review the following information:

- The development of the pain threshold over time.
- The development of the pain scale over time.
- The location of the pain.
- An advice regarding the measurement location of the QST.
- The specific symptoms of the pain (optional).

The prototype of the app has shown to be effective. It includes all data which has been marked, by an anaesthesiologist, as relevant for pain treatment. It allows the user to perform all action named above, except for the last one. Additionally, these actions can be completed successful within the specified time constrains. The user interface design has been proven to be user friendly. It scored above average on the System Usability Scale (SUS).

Chapter 8

Recommendations

First, the prototype should be improved in order to meet all the system requirements. Some concepts included in the app are not functional, such as specifying the patients pain symptoms, adding interventions related to treatments and medication, registering actual users, etc. These should be finished. Second, Flaws regarding the layout design need to be tackled. For example, the labels of the x-axes on the home page, are not completely visible. Implementing default padding size should be considered. Moreover, responsive layouts should be created, to make the app adapt to different screen sizes and position. Additionally, proper data structures are not within the scope of this project, so further research should elaborate on this. The usability can be further improved by enhancing the navigation. There can be considered to use more fragment layout, so the menu can be opened from multiple pages. Even though Android has a default back button, it is desirable to implement these in the system as well. Because the default back button will return to the previous *app state*, instead of the previous *page*, the page will not be refreshed. There should be considered whether this effect is desirable or not. Correct usage of medical terms should be validated. For example, "pain scale" has been used incorrectly and should be replaced by "pain score". After the app is improved, the system should be tested on the actual effectiveness in clinical practice. Tests should be performed in real case scenarios.

More research should be done regarding the monitoring of external factors. Anaesthesiologist Krabbenbos indicated that the correlation between these interventions and the pain development can provide new insights. For example, there can be elaborated on different visualisations. They can be visualised as described in chapter 5, but there can also be considered to visualized them in a time line.

Better integration with the patients' electronic health record (EHR), should be considered. The ideal situation would be, to automatically save the results on in the EHR as well. Communication with servers can be considered to accomplish this. Servers can also allow better transition of data, as a patient is treated by multiple doctors.

Finally, further research regarding the patients' perspective should be considered. Until now, the focus has been constraint to the needs of the anaesthesiologists. Nevertheless, it is interesting to elaborate on the patient aspect on the system. For more consistency in measurements, there can be considered to provide patient instructions regarding the measurement procedure. Moreover, patient might desire to review their development at home as well. There can be considered to extend the system with an additional patient platform. This platform, might even allow patient to monitor their pain themselves.

Bibliography

[1] R. K. van Zorg, "Chronische pijn," *Den Haag, november*, 2011. 1, 5

[2] O. H. Wilder-Smith, "Chronic pain and surgery: a review of new insights from sensory testing," *Journal of pain & palliative care pharmacotherapy*, vol. 25, no. 2, pp. 146–159, 2011. 1, 6

[3] A. Mader and W. Eggink, "A design process for creative technology," The Design Society, 2014. 2

[4] L. Nielsen, *Personas-user focused design*, vol. 15. Springer Science & Business Media, 2012. 2

[5] C. LeRouge, J. Ma, S. Sneha, and K. Tolle, "User profiles and personas in the design and development of consumer health technologies," *International journal of medical informatics*, vol. 82, no. 11, pp. e251–e268, 2013. 2, 14

[6] E. Kerstman, S. Ahn, S. Battu, S. Tariq, and M. Grabois, "Neuropathic pain," *Handb Clin Neurol*, vol. 110, pp. 175–187, 2013. 5

[7] C. Clauss and W. Clauss, *Humanbiologie kompakt*. Springer-Verlag, 2010. 5

[8] V. Thei and R. L. Fainsinger, "pain," in *Palliative care: Core skills and clinical competencies*, ch. 7, Elsevier Health Sciences, second edition ed., April 2011. 5

[9] M. Kay, J. Santos, and M. Takane, "mhealth: New horizons for health through mobile technologies," *World Health Organization*, vol. 64, no. 7, pp. 66–71, 2011. 5, 6

[10] H. E. Merskey, "Classification of chronic pain: Descriptions of chronic pain syndromes and definitions of pain terms.," *Pain*, 1986. 5

[11] L. M. E. Frantsve and R. D. Kerns, "Patient–provider interactions in the management of chronic pain: current findings within the context of shared medical decision making," *Pain Medicine*, vol. 8, no. 1, pp. 25–35, 2007. 5

[12] F. J. P. M. Huygen, M. A. F. J. van de Laar, K. C. J. Vos, M. M. C. Kuin, R. van der Heide, and D. Kuijpens, "Pain proposal–noodzakelijke verbetering van zorg en aanpak van chronische pijn," *the Netherlands, April*, 2011. 5

[13] H. Breivik, B. Collett, V. Ventafridda, R. Cohen, and D. Gallacher, "Survey of chronic pain in europe: prevalence, impact on daily life, and treatment," *European journal of pain*, vol. 10, no. 4, pp. 287–287, 2006. 6

[14] A. Van Rijen, M. de Lint, and L. Ottes, *Inzicht in e-health*. Raad voor de Volksgezondheid en Zorg Den Haag, 2002. 6

[15] M. W. Jaspers, M. Smeulers, H. Vermeulen, and L. W. Peute, "Effects of clinical decision-support systems on practitioner performance and patient outcomes: a synthesis of high-quality systematic review findings," *Journal of the American Medical Informatics Association*, vol. 18, no. 3, pp. 327–334, 2011. 6

[16] B. Martínez-Pérez, I. de la Torre-Díez, M. López-Coronado, B. Sainz-De-Abajo, M. Robles, and J. M. García-Gómez, "Mobile clinical decision support systems and applications: a literature and commercial review," *Journal of medical systems*, vol. 38, no. 1, pp. 1–10, 2014. 6, 7

[17] C. Rini, D. A. Williams, J. E. Broderick, and F. J. Keefe, "Meeting them where they are: using the internet to deliver behavioral medicine interventions for pain," *Translational behavioral medicine*, vol. 2, no. 1, pp. 82–92, 2012. 7

[18] C. Lalloo, L. A. Jibb, J. Rivera, A. Agarwal, and J. N. Stinson, "theresa pain app for that: Review of patient-targeted smartphone applications for pain management," *The Clinical journal of pain*, vol. 31, no. 6, pp. 557–563, 2015. 7

[19] S. Inc., "eh&p - android apps on google play." https://play.google.com/store/apps/details?id=com.scymed.android.ehnp, November 2015. (Accessed on 07/07/2016). 7

[20] BMJ, "Best practice - android apps on google play." https://play.google.com/store/apps/details?id=com.bmjgroup.bestpractice, September 2015. (Accessed on 07/09/2016). 7

[21] "Medidss - android apps on google play." https://play.google.com/store/apps/details?id=com.Medilogy.MediDSS. (Accessed on 07/09/2016). 7

[22] ManagingLife, "Manage my pain pro - android apps on google play." https://play.google.com/store/apps/details?id=com.lcs.mmp.full, June 2016. (Accessed on 07/09/2016). 7

[23] L. DamoLab, "My pain diary - android apps on google play." https://play.google.com/store/apps/details?id=com.damonlynn.mypaindiary, May 2016. (Accessed on 07/09/2016). 7

[24] "Pain diary & forum catchmypain - android apps on google play." https://play.google.com/store/apps/details?id=com.sanovation.catchmypain.phone. (Accessed on 07/10/2016). 7

[25] A. Nieuwenhuis *et al.*, "Wet bescherming persoonsgegevens," *Tijdschrift voor bestuurswetenschappen en publiek recht*, pp. 311–316, 2002. 11

[26] "Artikel 1 sub a wbp — autoriteit persoonsgegevens." https://autoriteitpersoonsgegevens.nl/nl/over-privacy/wetten/wbp-naslag/hoofdstuk-1-algemene-bepalingen-art-1-tm-5/artikel-1-sub-wbp. (Accessed on 07/08/2016). 11

[27] "Data gebruiken en delen voor medisch- wetenschappelijk onderzoek vormt privacywetgeving een bedreiging? mr. dr. corrette ploem 11-9-2014. - documents." http://docslide.nl/documents/data-gebruiken-en-delen-voor-medisch-wetenschappelijk-onderzoek-vormt-privacywetgeving.html. (Accessed on 07/08/2016). 11

[28] "Melden verwerking persoonsgegevens — autoriteit persoonsgegevens." https://autoriteitpersoonsgegevens.nl/nl/melden/melden-verwerking-persoonsgegevens. (Accessed on 07/08/2016). 11

[29] R. Adri de Bruijn, P. Advisory, and C. de Bie MSc, "Beveiligingeisen ten aanzien van identificatie en authenticatie voor toegang zorgconsument tot het elektronisch patiëntendossier (epd)," 2008. 11

[30] R. Thompson, "Stakeholder analysis - project management tools from mindtools.com." https://www.mindtools.com/pages/article/newPPM_07.htm, 2002. (Accessed on 07/11/2016). 13

[31] K. S. Park and C. H. Lim, "A structured methodology for comparative evaluation of user interface designs using usability criteria and measures," *International journal of industrial ergonomics*, vol. 23, no. 5, pp. 379–389, 1999. 27

[32] J. Brooke *et al.*, "Sus-a quick and dirty usability scale," *Usability evaluation in industry*, vol. 189, no. 194, pp. 4–7, 1996. 27

[33] J. Sauro, "Measuring usability with the system usability scale (sus): Measuringu." `http://www.measuringu.com/sus.php`, February 2011. (Accessed on 07/11/2016). 28, 45, 47

[34] "Functional requirements vs non functional requirements." `http://reqtest.com/requirements-blog/functional-vs-non-functional-requirements/`. (Accessed on 07/08/2016). 31

[35] K. Waters, "Prioritization using moscow," *Agile Planning*, vol. 12, 2009. 31

[36] R. Molich and J. Nielsen, "Improving a human-computer dialogue," *Communications of the ACM*, vol. 33, no. 3, pp. 338–348, 1990. 35, 45

[37] "Design — android developers." `https://developer.android.com/design/index.html`. (Accessed on 07/12/2016). 36

[38] K. Chawla, K. Kuan, and L. Fujiwara, "Android development for beginners: How to make apps — udacity." `https://www.udacity.com/course/android-development-for-beginners--ud837`. (Accessed on 07/08/2016). 36

[39] "Fragments — android developers." `https://developer.android.com/guide/components/fragments.html`. (Accessed on 07/08/2016). 41

Appendix A
Stakeholder analysis

Network analysis

Politic-administrative Actors	Actors with civil servant functions	Actors with social function	Actors with technical/scientific function	Individual citizens (ordered and ordered)
-	-	Hospital	Pain specialist	Patients with chronic pain symptoms

Stakeholder selection

Pain specialist

Interest: (++) Has interest in the project because it will support the pain specialist in the clinical process of pain screening and diagnosing chronic pain. The app will make is easier and quicker to perform an eQST test, because it does not have to be done in the lab anymore. Next to this, the app will be able to collect this data, together with questionnaires about (chronic) pain. This combined data collection and presentation will make it able to spot chronic pain patients sooner, so adequate treatment can be applied.
Effect standard: (++) Interest will served met with the solution of the problem.
Target standard: (++) actor is the main target group/user group of the product.
Resistance standard: (+) pain specialist will be able to block the plan, by rejecting to application during the design process. Possible reasons to do this can be: they don't believe it will fit the clinical process, they believe it is not save to use due to lack of proper data security or because they are afraid it violates their patient's privacy.
Process standard: (++) Pain specialist's opinion about useful data collection and data presentation are very important for the development process. They will be able to enrich the process with their knowledge of the daily clinical process of pain screening and the scope of it. Their needs are the foundation of the system and user requirements. They opinion about useful data collection and data presentation are very important for the development process.

Patients with chronic pain symptoms

Interest: (+) Patients will be interested because the app will help doctors recognise chronic pain sooner, so adequate they can receive adequate treatment. They won't have to go to the lab for the screening, it can be done easily and fast during a regular appointment.
Effect standard: (-) Patients interest are in an indirect relation with the problem the product is solving. The products supports the solution of the problem they are suffering from.
Target standard: (-) Patients are not the main target or user group. However, they are indirect users because it is their personal data that will be saved in the application, and they have the right to see the presentation of they own data.
Resistance standard: (+) Patients have the right to resist the use of the product for their treatment. Reasons for this might be that they are afraid that their data is not sored safely or their privacy is violated. In this case, the pain specialist will not be able to use the application.

Process standard: (--) Patients do not have the knowledge of clinical process of treatment. Their opinion about system related with the system or user requirements are not important, because they are not direct users of the product.

Hospitals

Interest: (+/-) Product is interesting for the hospital because it will help their staff with structuring their work. This will be for the better with respect to its patients. Application of eQST system during daily clinical process will also result in less use of the lab for this test. This might be cost reducing, because the lab can be used for other purposes instead.
Effect standard: (-) hospitals interests are not influenced by the direct problem or solution with respect to the product.
Target standard: (--) Hospitals itself are not a user of the project.
Resistance standard: (+) Hospitals might forbid they staff to work with the system.
Process standard: (--) Hospital as an institution cannot provide knowledge or information which can be used for the product development process.

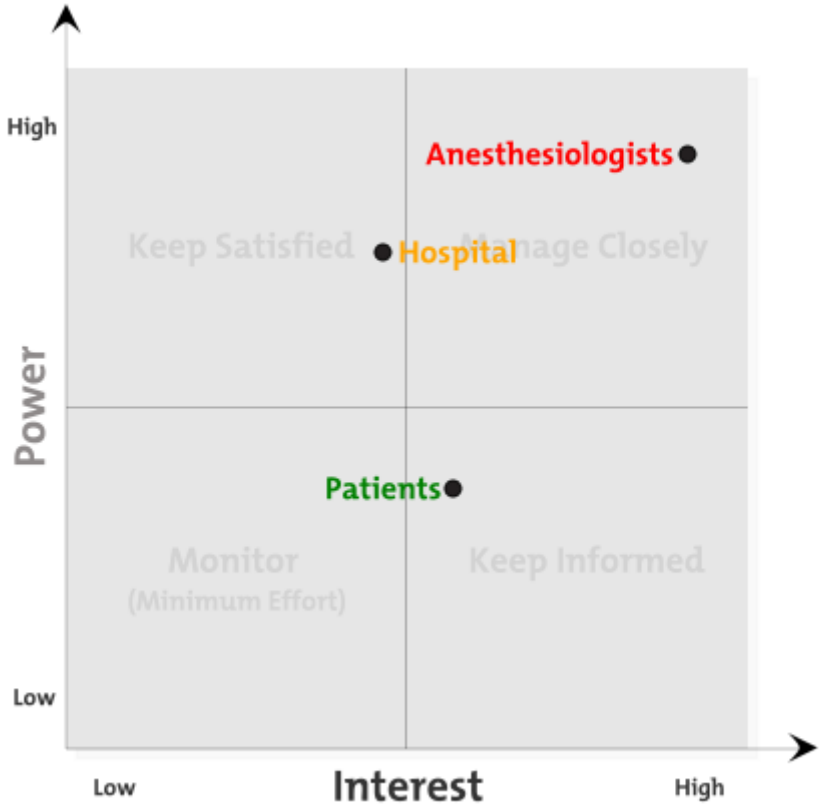


Figure 2. Stakeholder power/interest grid.

References:

1. “Stappenplan voor stakeholdersanalyse” [online], *Externe analyse,* chapt. 4.2.1., Available: http://www.coutinho.nl/fileadmin/documenten/ondernemen-en-innoveren/4.2.1_Stappenplan_stakeholdersanalyse.pdf
2. Mindtools, “Stakeholder analysis – Winning support for your projects” [online], available: https://www.mindtools.com/pages/article/newPPM_07.htm

Appendix B. LeRouge method: translate interview into personas

LeRouge classification	Interview segment	Questions to guide the interview
Demographics	Algemeen	• Wat is uw naam? • Wat is uw geboortedatum? • Welke studies heeft u gevolgd? • Wat is de reden dat u zicht bent gaan specialiseren in patiënten met chronische pijn?
Healthcare specifics	De patiënt	• Om welke redenen komen patiënten bij u langs? Pijn symptomen, neurostimulatie • Hoeveel mensen behandeld u per dag? 5 nieuwe, controle telefoon • Hoeveel komen met pijnklachten? Hoeveel daarvan lijden (uiteindelijke) aan chronische pijn? Iedereen komt met pijn klachten. Acute pijn: hernia bvb, functionele pijn, gaan meestal. Merendeel is wel grootse deel chronisch. Hernia uitzondering. Hernia patiënten: neurologen. • Bij hoeveel is dit veroorzaakt door verhoogde pijngevoeligheid (neurotische pijn)? • Wat is de leeftijdscategorie van deze patiënten? Piek pensioen leeftijd, mensen die hier lopen bij bijvoorbeeld neuroloog. Acute en hernia zijn jonger. Mensen die gewoon pijn hebben zijn ouder
Healthcare specifics	Klinische proces	• Hebben patiënten al bij andere dokter/specialist gelopen voordat u ze gaat behandelen? Behandelingen: medicatie tot ruggenprikken enzo, implanteren dingen in neurosysteem. • Kunt u stapsgewijs de huidige gang van zaken uitleggen vanaf het moment dat u een nieuwe patiënt krijgt? • Hoeveel tijd kost het gemiddeld van het eerste bezoek tot aan de diagnose? • De ambities gebruikt u nu alleen voor onderzoek. Wat is hier de reden van? • (Waarom) wilt u dit ook graag gaan gebruiken in het klinische proces? • Waar past dit in/tussen de stappen van het klinische proces zoals u die zojuist beschreef?
Healthcare specifics	Diagnose proces	• Op basis van welke gegevens stelt u een diagnose? • Hoe verkrijgt u deze nu? • Zou u deze ook allemaal terug willen zien in de app or zou u meer of minder willen zien? • Hoe worden deze gegevens nu voor u gepresenteerd?

Appendix B. LeRouge method: translate interview into personas

		• Hoe zou het voor u overzichtelijker kunnen worden zodat u de situatie van een patiënt makkelijker kan bepalen?
Technology	Technology	• Maken jullie gebruik van elektronische dossier? • Welke eisen stelt u aan deze app met betrekking tot: efficiëntie (snelheid & handigheid), functies, testuitslagen/resultaten. • Wat weet u over de opslag van patiëntgegeven op mobile apparaten? • Staat het ziekhuis toe dat u met een tablet door het ziekenhuis loopt? • Heeft u tablet en/of smartphone? Persoonlijk gebruik of voor uw werk? Wordt er vanuit u beroep verwacht dat u een smartphone heeft? • Gebruikt u deze werk gerelateerde dingen? Wat voor? Welke apps gebruikt u hiervoor?

Appendix B. LeRouge method: translate interview into personas

LeRouge classification	Interview segment	Quotes/answers to interview	Translation to persona: anaesthesiologist	Translation to persona: patient suffering from chronic pain
Demographics	Algemeen	Pijn specialisten zijn ook Anesthesiologen Eerst dokter en dan specialisatie, laatste jaar specialiseren in onder andere pijn Anesthesist, combinatie met werken op de ok. “Alle dokter zijn pijnbestrijders, iedereen schrijft pijnmedicatie voor. Dat doen wij niet alleen.” Ze hebben 7 pijn bestrijders, 2 dagen pijn en de rest ok. Dag pijn voor specialist bestaat uit: 1 dagdeel poli, en 1 dagdeel behandelingen.	He followed medicine, with specialisation for doctor and later anaesthesiologist. He work in the outpatient clinic for 2 days of the week, the other days at the operation theatre complex.	
Healthcare specifics	De patiënt	Zien mensen die vanuit 1e lijn worden doorgestuurd (dokter/huisarts), of 2e lijn (ziekenhuis). Zijn een van de grootste kliniek qua patiëntaantal (hier meer behandelmethoden), dus krijgen relatief veel vanuit deze stroom. Mensen met chronisch pijn syndroom: komen hier op eigen verzoek, die al 20 jaar pijn hebben en al van alles hebben geprobeerd. Hernia: komen acute pijn, maar komen hier pas als de pijn aanhoudt. Neuromodulatie en neurostimulatie ook een grote groep patiënten. Merendeel is wel patiënten heeft idd last van chronische pijn. Hernia is eigenlijk een uitzondering. Alle mensen hebben pijnklachten. Maar varieert stek want pijn is een symptoom, met vele verschillend mogelijk oorzaken. Kan oncologische pijn zijn (kanker) of acute pijn, bvb hernia die niet over gaat en alles wat ertussen in zit. Ruime schatting: 60 procent met zenuwpijn/kenmerken van neuropatische pijn. Leeftijd: Gemiddeld heb je niet zo veel aan. Hangt af van wat je aanbiedt op de praktijk en de verwijzing stroom (ketenzorg) hier op praktijk ontvangt een neuroloog herniapatenten, dus pijnpoli ook. Mensen die ouder zijn hebben natuurlijk wel meer sneller pijnklachten. Zonder	At the outpatient clinic, the main complains of the patients he receives is pain: acute (because of demages tissue) as well as chronic pain. The chronic pain patients have seen general practitioners, but they couldn’t help. They still experience pain on for example, places of former injuries or operation, knee or back pain. He believes the majority of this patients suffer from neuropatic pain.	Twisted knee when he was 20. Had a knee operation so knee should be working ok. However, since 5 years, he started to experience pain again, which wouldn’t go always after exercises and rest. Practitioners, physiotherapists and sport doctors were not able to help anymore and advised to visit an anaesthesiologist in a pain outpatient clinic.

Appendix B. LeRouge method: translate interview into personas

		acute pijn en hernia, zijn de meeste patiënten rond 65. Maar in ander praktijk kan je heel ander antwoord krijgen.		
Healthcare specifics	Klinische proces	Als dokter diagnose niet rond krijgt, kunnen ze hier aanvullende diagnose doen, maar kan ook zijn dat ze eerst langs andere specialist bvb neuroloog of orthopeed. Dan probeer je tot diagnose te komen en een behandel traject te komen. Mensen chronisch pijn syndroom: verwachten een andere vorm van zorg dan mensen die zijn doorgestuurd. Ze hebben multidisciplinaire spreekuren, meerdere specialisten: revalidatie, neuroloog, fysiotherapeut, psychiater. Doorgestuurd patiënten: pijn hier en hier na bvb operatie duurt langer dan verwacht, dan is er een duidelijke probleem en makkelijke cases eigenlijk. 7 pijn bestrijders, 2 dagen pijn en de rest ok. Dag pijn voor specialist bestaat uit: 1 dagdeel poli, en 1 dagdeel behandelingen. Poli: Nieuwe patiënten [4-5], controle patiënten [5], (evaluatie via gesprek of telefoon [20]). Mensen van 65 of ouder liever niet via de telefoon. Moeilijk te zeggen hoe het “algemene proces in zijn werk gaat”. Zo’n test als die van de ambustim wordt zeker herhaalt. Dit is het monitoren van de pijn. Strikt indicatie gebied voor relevantie en toepassing van een Nocitrack/neuromodulatie.	Most patient had visited another doctor before, but couldn’t get a complete diagnoses. Therefore, he works together with different specialists to understand the symptoms.	At the outpatient clinic, he was also examined by a group of different specialists.
Healthcare specifics	Diagnose proces	Chronisch pijn syndroom: neuroloog, dan kunnen ze zomaar een diagnose en behandeling voor geven die zal helpen. Je moet eerst doorgronden wat achter de symptomen zit voor een passende behandeling, en dat doen ze vaak met multidisciplinair team (met anesthesioloog, fysiotherapeut, oncoloog, psychiater). Ligt ook aan de classificatie van pijn die wordt gebruikt, waar je je patiënt indeelt. Verschillende pijn: oncologisch- niet	Pain is a symptom, not a disease, you need to understand the cause before a good diagnose can be make. This makes every case different from the other.	He believe the diagnoses process for this symptoms takes very long. He had to see multiple doctors, who do different tests. He spends a lot of time waiting for results, before he received treatment.

Appendix B. LeRouge method: translate interview into personas

		oncologisch, Acture pijn –chronische pijn, neuropatische pijn-nonstructieve pijn. Maar kan ook een combinatie zijn van. Pijn specialisten zijn beter op de hoogte welke medicatie voor welke problemen het beste werk en hoe ze dat moeten voorschijven. Pijn specialisten hier dus toegevoegde waarden. Moeilijk te zeggen welke indicatoren (altijd) relevant zijn voor de diagnose. Relevante gegevens waar ze mee werken: - Kwaliteit van leven. Zijn vragenlijsten over. Kan belangrijk zijn: kleine daling pijnscore kan grote invloed hebben hierop. Kan overwegen vragenlijsten te implementeren, maar zal hier niet zo snel voor kiezen, omdat het vaak niet relevant zal zijn. Wel relevent kan zijn de indicatoren of symptomen waaraan ze herkennen of mensen zenuwpijn hebben. -Pijnschaal. Verandering van deze score is waar behandeling sterk van af hangt. -Behoefte aan methode om meer subjectief vaste te kunnen stellen hoeveel pijn iemand ervaart. Ze varen erg op wat de patiënt zegt. Vanwege gewenning verschuift het referentiekader en is de score hierdoor erg subjectief. Dit wil je eigenlijk niet. Overzicht in resultaten en gegevens in een app: -Visualisatie van resultaten belangrijk, zeker voor patiënten die langer in behandeling zijn. Verloop laten zien in grafieken kan zeker de efficiëntie van de behandeling beter weergeven. In kliniek is dit zeker belangrijk, zowel relevant voor de patiënt en voor de dokter. -getallen zijn soms nodig voor statistieken etc. dus ook handig om deze te kunnen exporteren o.i.d. -Indicatoren waaraan je kan afleiden dat een patient lijdt aan zenuwpijn.	Believes all doctors are pain relievers, but as anaesthesiologist you have more knowledge about what treatment suit which problems best.	

Appendix B. LeRouge method: translate interview into personas

Technology	Technology	Ambistim/nocitrack: Ze hebben aantal liggen op de praktijk. Imre kent het apparaat goed. Doen ze onderzoek mee. Heeft nog geen CEE keurmerk, dus mag nog niet gebruikt worden in klinische praktijk. Lopen nu meerde studies met de Ambistim, zijn op weg naar implementatie in kliniek. Het beeld daarvan: voor elke nieuwe patiënt meting gedaan door doktersassistent of pijnconsulent. Ethisch mag dit nog niet voordat het door die CEE procedure is. Nu alleen in onderzoek verband. Onderzoek wordt wel gedaan met patiënten uit de kliniek. Gebruik: Ze trainen patiënten eerst, 3x proberen hoe het voelt. Daarna instructie: laat los waarneem het pijn doet. Waarde heeft pijndrempel aan. Test moet wel zo betrouwbaar mogelijk zijn. Dus zou zinvol zijn dat als een patiënt het gebruikt, en een algemene instructie is. Meting kan sterk variëren: Heeft te maken met wie meet en de instructies, maar ook de ruimte waar je in zit. Eerste metingen zijn nog niet goed Functies die app moet hebben: -ingeven van meetlocatie -ingeven van voorkeurslocatie -trainingssessie/instructies inbouwen -meerwaarde waar -directe opslag en expoteren naar een databestand Hoe waarde creëren klinische praktijk: 1. Met behulp van app meet proces standaardiseren: iedereen zelfde instructies. (bvb instructie via de app zodat iedereen dezelfde instructies krijgt). Dit scheelt ook heel veel tijd. 2. Implementatie is makkelijker te realiseren als de patiënt zelfstandig het proces door kan lopen, vanwege verlaging van de arbeidsintensiteit.	The believes QST and the ambistim can be a good clinical tool to monitor the pain. ‘We have to go with what the patients is telling us. This system could provide more objective results. It would be nice to conduct the test on every new patient as part of the routine.’ At the moment, he sees too many obstacles in the current system to implement it. He wants the usability to increase and the measurement process to be standardised, so results won’t vary because of this aspect.	

Appendix B. LeRouge method: translate interview into personas

		3. Vergroten van gebruikersgemak, dat is zeker belangrijk voor gebruik in kliniek. Opslag data meerwaarde, omdat het nu van papier naar computer moet worden gestuurd. 4. Nocitrack geeft meer objectieve waarde van een pijnschaal/pijnscore. Kliniek maakt gebruik van een elektronisch dossier, hoort hier ook thuis. Voorwaarden van beveiligingen etc. Kijken naar richtlijnen van *Good clinical practice.* Gegevens in tablet o.i.d. moet ook ergens anders worden opgeslagen (afgesloten ruimte). In onderzoeken moeten gegevens niet direct herleidbaar zijn, maar in de kliniek is dit minder relevant.		

Belangrijk dat patiet goed ge:istureerd worden. 3 xmetn

Functies: waar meet je, voorkeurslocatie. Kunnen meten dat je een trainingen hebt. Niet alle dat is valid, me moet iets kunnen kiezen welke je wil gebruiken.

Onderzoeken: wie meet maakt heel erg uit, vanwege instructies en pschychologisch: mensen, ruimte. Moet een standaard omgeving voor komen. Meetsessie standaardiseren voor gehele kliniek via de app zou wenselijk zijn: instucties via app zou echt meerwaarde hebben. Gebruiksvriendeijker.

De data opslaan is niet het belangrijkste, wel meting process.

Nog niet te gebruiken in clinische praktijk. Mag nog niet, zijn er wel mee bezig. Medisch ethisch mag dat nog niet. Moeten meerdere studies zijn die laten zien dat het echt werkt voordat het kan. Medische personeel is heel drunk, nocitrack doen bij iedereen met een medwerken duurt lang en is arbeidsintenvief. Doorloop waarmee patient dat zelfstandig zou kunnen doen heeft zeker meerwaarde. Zeker meerwaarde om deze gegevens samen op te slaan en te presenteren, dat ook exporteren naar een digitaal databestand: increase gebruikersgemak. Metingen zijn altijd niet gelijk goed.

WEkren met electronische dossiers voor patienten. Exporteren data: naar bvb medisch dossier.

Visiele presentatie is relevant zeker als je mensen langer behandeld, zodat je effectieviteit van behandeling kan zien, ook voor uitleg naar patient. Zowel voor dokter als patent: willen zien wat er gebeurt.

Patienten worden zeker vaken in tijd gemeten. App kan dan goed monitoren, juist bij deze patiënten.

Appendix B. LeRouge method: translate interview into personas

Varen heel veel op wat patient zegt, pijnschaal (1 tot 10). Maar is wel erg subjectied. Hangt nu te veel van patienten af. Chronisch pijn patienten: neurostimulatie gedaan, werkt eerst, maar lichte pijn blijft, en referentiekader geschuift. Aparaat die vast kan stelen hoeveel mijn iemand heeft: nicitrack heeft die potentie zeker.

Neurstimulatie, bij verscillende patienten gebruikt, wel met strike indicatiegebeid.

pijn indicatoren via vragenlijsten die standaar worden afgenomen. Resultaten nodig naast nocitrack: Pijnschaal,

Zelf een lijst met vragen? kenmerken van neuropatische pijn erin zettne/aankruizen, Hieraan te herkennen dat mensen zenuwpijn hebben.

Meerderheid _60 procent met kenmerken van neuropatische pijn

Good clinical practische.

Moet andere vorm van opslag zijn dan tables. Resultaten niet lijden naar naam, nummering zou wel kunnen, doen ze nu ook. Naam doen ze nu wel, omdat het in onderzoeksverband plaatvind.

Appendix B LeRouge method: translate interview into personas

DR. ERIC VELDWEG – THE ANAESTHESIOLOGIST

AGE 36

EDUCATION Eric started with medicine. After he chose to specialize himself in the anaesthesiology.

PROFESSION Eric works as an anaesthesiologist in the hospital. During the week, he spends two days working in the pain outpatient clinic. The other days, he works at the operation theatre complex.

"Pain is a symptom, not a disease. You need to understand the symptoms, before a good diagnose can be made."

Patients he receives at the outpatient clinic, all report to experience pain. This can be acute pain (because of damaged tissue) or chronic pain. The patients might have received treatment from other doctors before, but still experience pain. This pain occurs mostly at places of a former injuries or operation, the knee or the back. He believes the majority of this patients suffer from neuropatic pain.

Most patient had visited another doctor before, but couldn't get a complete diagnoses. Pain is a symptom, not a disease. You need to understand the symptoms, before a good diagnose can be made. This makes every case different. Therefore, he works together with different specialists. His motto as a pain specialist is:

"All doctors are pain relievers, but as anaesthesiologist you have more knowledge about what treatment suits different problems best."

Eric believes QST and the Ambustim can be a good clinical tool to monitor the pain. 'We have to go with what the patients is telling us. This system could provide more objective results. It would be nice to conduct the test on every new patient as part of the routine.'

At the moment, he sees too many obstacles in the current system to implement it. He wants the usability to increase and the measurement process to be standardised, so results won't vary because of this aspect.

"The Ambistim can provide us with objective results. It would be nice to conduct the test as part of the routine."

Appendix C

Telephone conversation I. Krabbenbos - logging in to the system

Idea

A bar- or QR-code per patient will be put into the electronic health record (EHR). This, in combination with a password and login will allow access to the system

Problem perceived by I. Krabbenbos during the last conversation

The health record is only available on a computer, so this means that the app can only be used if there is a computer in the room. While visiting patients which are lying in a hospital bed, a computer might not be available. This will decrease the mobility which is such an advantage of used a tablet.

Results

During regular appointment, the EHR will be accessible. While walking around, you may not always have it.

Right now, the doctors have card with which they can login to the system.

Sometimes, people have a barcode sticker on this card, or the case around it. There are used for example for blood analysis. This barcode allows them access to certain rooms, systems of test results. The medical staff receives this barcode sticker after they followed a training. This way, only people with knowledge of the system will be able to use it, and it is a unique login code

Every staff member of the hospital needs this card, so everyone has one.

Conclusion

Working with barcode system is already implemented in the hospital.

Having a unique physical key/code for ever patient to login to the system is not very convenient, because the EHR or the patient might not be available when one needs access to the system. Since doctors will always carry their member card, it would be more convenient to attach this code here. Now, a physical key will still be used to secure authentication, but it will be connected to the healthcare provider and not the patient.

Appendix D

Lofi prototype, user test

The target group

Participant will be above 18 years old. They are either graduates or students of Technical Medicine (Technische geneeskunde) or currently working in a hospital. They need to have a Dutch nationality.

This mock-up test will not be for the end user. A different, more quality aimed, session will be planned to evaluate the design and to answer specific design questions. This is so the end user will be bothered by determining basic usability flaws which can also be spot by expert users.

Setting

Participant will conduct the test individually. They will perform the test on the screen of a laptop. The test will be held in a place which will represent the daily live environment. Therefore, the participant should know the place and feel comfortable in it. This place does not necessarily have to be quit, there can be other people present. However, during the test, the participant should not be interrupted and except for the one observer, others are not allowed to watch the participant or the screen.

Setup

The participant will enter and be asked to sign the consent form. After, the participant will be told by the observer that he/she will be testing an app which will be applied in the pain outpatient clinic. The participant will be asked to read the description of NociTRACK, which is available on the NociTRACK website, so they can imagine how the system works. A screenshot of the webpage that will be shown, can be seen in figure 1.

Appendix D

Figure 1. Screenshot of the NociTRACK website, with an explanation if the device.

Appendix D

When they are done reading, the purpose of the app will be further explained:

"Zoals je net hebt gelezen, kan met de NociTrack de pijn gevoeligheid worden gemeten door middel van het apparaat en een smartphone of tablet. Deze methode heet 'quantitative sensory tesing', ook wel afgekort QST genoemd. Dit wordt vaak gedaan bij chronische pijn patienten, waarbij er misschien sprake is van zenuwpijn, of neuropathische pijn. Dit houdt in dat de patiënt over-gevoelig is voor pijn.

Het meten van de pijn gevoeligheid gaat als volgt: er worden electroden op de huid aangebracht. Bij het inhouden van de knop op de NociTrack, gaat er een oplopende hoeveelheid stroom doorheen lopen. Dit geeft eerst een tintelend gevoel. Zodra dit gevoel onprettig wordt, moet de patiënt de knop weer los laten. De waarde die het apparaat dan aangeeft in de pijn drempel.

Deze waarde voor de pijn drempel wordt alleen nog niet automatisch opgeslagen. De app die je voor je gaat krijgen, zal deze waarde opslaan en presenteren aan de arts. Daarnaast kunnen er ook andere gegevens die te maken hebben met de pijnklachten in worden opgeslagen. Dit zijn onder andere het gebied waar de pijn zich voordoet, de pijn schaal, dit is een waarde van 1 tot 10 waarin de patient de sterkte van de pijn aangeeft, en de DN4 vragenlijst test aan de hand van de symptomen of het om zenuwpijn gaan."

The participant will be given the time to asked questions about (chronic)pain or the device before the user test will start. Question about the app will not be answered.

If the participant does not have questions anymore. There will be told that the test will start soon, but you will he/she will be informed about login procedure, which works with a QR code per patient which works in combination with a login name and password, in order to secure the personal data of the patient. Next to this, the participant should know that it is not possible to add text and that therefore example text is already given for every text field inside the app.

Now, the user test will start. The user will be given a user scenario containing 4 different use cases which include 5 tasks in total. Every task is represented with a bullet point. There tasks will be read out loud by the participant. The scenario can be found in the next page.

When the last case has been completed, the participant will have to complete the System Usability Scale questionnaire, which can be found at the end of this section. This will be the last part of the user test.

Finally, the participant will be thanked for the participation and be given the offered to receive a copy of the results when finished.

Appendix D

User scenario:

You are an anaesthesiologist who works at the outpatient clinic. You deal with several chronic pain patients every day. You use NociTRACK and the complementary app for every patient, to save and review relevant information about the patient pain symptoms.

Use case 1: recap of information

Your first patient today is Miss Inge Jansen. She has been a patient at the outpatient clinic for several months. However, you see many different patients and you want to check her status on the app.

1. Log in to the system
2. Navigate through the system freely. Find the information below.
    a. In what area of the body is her pain located?
    b. What was the last given pain scale?
    c. When was the last measurement of the pain threshold?
    d. What was the exact value of the pain threshold the last time you measured?
    e. Are the pain symptoms characterized by electric shocks?

Use case 2: New pain rate

You patient, Miss Inge Jansen has entered the room. You ask her how she is doing. She says she still has a lot of pain. When you ask her to rate her pain, she says 8.

3. Add this new pain scale

Use case 3: New QST

Now, you want to conduct a new test to measure the pain threshold to monitor how this develops.

4. Add the new pain threshold, which will be done in combination with the NociTrack

Use case 4: Next patient

You the appointment with Inge Jansen has ended and a new patient enters the room. You will need the data from the other patient.

5. Log a new patient in

Appendix D

	Strongly disagree				Strongly agree
1. I think that I would like to use this system frequently	1	2	3	4	5
2. I found the system unnecessarily complex	1	2	3	4	5
3. I thought the system was easy to use	1	2	3	4	5
4. I think that I would need the support of a technical person to be able to use this system	1	2	3	4	5
5. I found the various functions in this system were well integrated	1	2	3	4	5
6. I thought there was too much inconsistency in this system	1	2	3	4	5
7. I would imagine that most people would learn to use this system very quickly	1	2	3	4	5
8. I found the system very cumbersome to use	1	2	3	4	5
9. I felt very confident using the system	1	2	3	4	5
10. I needed to learn a lot of things before I could get going with this system	1	2	3	4	5

Appendix D
Method

During the test, the following user interface concept will be analysed: efficiency, effectiveness and usability. The efficiency will measures by keeping the time per task, to measure the time needed to complete a task. To evaluate the effectiveness a binary score of 1 and 0 will be used: a 1 when the task has been completed successfully and a 0 if the participant failed to complete the task successfully. Additional notes will be made to determine what is causing confusion, frustration and where the bottleneck in the systems are. To evaluate the usability, the System Usability Scale (SUS) will be used. This is questionnaire, which consist of 10 statements which will have to be rated using a liked-scale. SUS scores have a range of 0 to 100. The SUS score will be calculated the following:

- Subtract 1 from the likert scale response for the odd numbered statements
- For even numbered statements, subtract the response from 5
- Add all values up
- Multiply by 2.5, to convert the range of possible values from 0 to 100

The SUS score can be best interpreted after is has been normalized. The graph below shows the scores through percentile ranks and a letter-grades scale. A SUS score above 68 are considered above average.

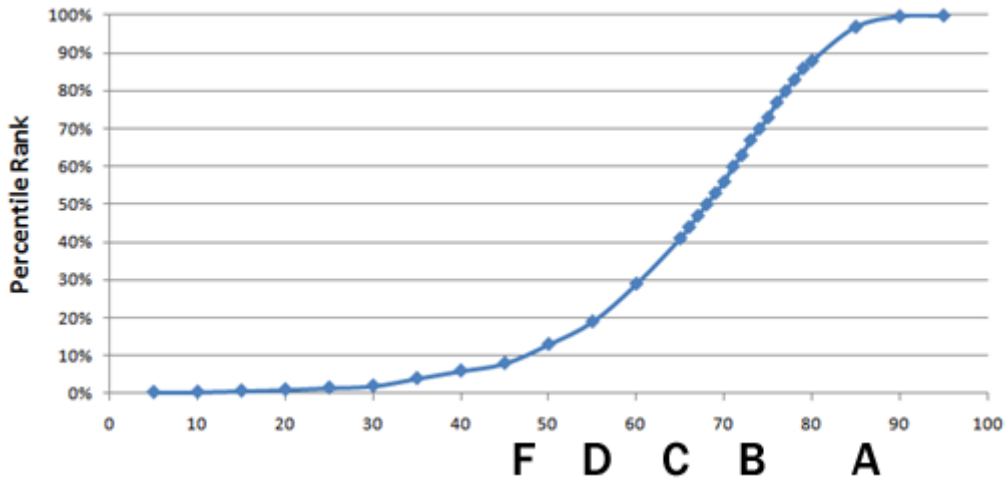


Figure 2. Interpreted SUS scores: normalizing through percentile ranking and letter-grade scale

A score of 74 or higher, will be strived for the final prototype. For this low-fi prototype, a score of 64 or higher will be considered successful. The time per task and the click per task will be used to determine which aspect of the user interface or user interaction will have to be improved.

The case tackle different user requirement. Below a list can be found in which the use cases are linked to different user requirments

Case 1: recap of information Task 1 Task 2e Task 2c Task 2a	 -doctors should be able to login to get access to the data -it present the accepted QST results visually -Graphs axes have to be readable - the actual measurement location will have to be saved

Appendix D

Task 2b Task 2d Task 2	- it shows the pain scale visually - it provides a checklist of symptoms which are a signal of neuropathic pain in - it present results in such a manner that it is meaningful for the user
Case 2: New pain rate, task 3	- it is possible to save the pain rate on a pain scale of 1 tot 10
Case 3: New QST, task 4	- it provides the user with information about the preferable measurement location - the actual measurement location will have to be saved - it saves the results generated by the Ambustim's QST - it saves this results per patient - it is possible to reject measurement values if a reason will been added
Case 4: Next patient, task 5	- doctors have access to the results of multiple patients - it saves this results per patient - only one piece of identifiable data will be saved for the doctor to recognise the patient in the system with which the user can distinguish whether the app shows the correct data

Appendix D

Results

participant	Effectiviteit score (uit 9)	SUS	effecientie score task 1	effecientie score task 2	effecientie score task 3	effecientie score task 4	effecientie score task 5
1	6	60	4	281	202	87	11
2	8	45,5	3	102	13	48	40
3	4	77,5	27	324	140	149	13
4	5	57,5	21	127	58	50	18
5	6	55	11	215	75	51	10
Average	5,8	59,1	13,2	209,8	97,6	77	18,4

Table 1. Scores per participant and averages

Effectiveness:

As can be seen in table one, the participant were able to complete about 64%of the tasks on average. The lowest amount of successfully completed tasks was 4 out of 9, which is 44%. The highest amount of successes was 8 out of 9, which is 89%.

task	success (out of 5)
1	5
2a	5
2b	1
2c	5
2d	0
2e	2
3	3
4	4
5	5

Table 2. Successes per task

Participants had the most difficulties with task 2d, which was finding the exact value of the pain threshold the last time it had been measured. Task 2b (finding the last pain scale), 2e (finding the pain symptoms) and task 3 also scores low in effectiveness.

These task, which score low in effectiveness, tackle the requirements:
- It shows the pain scale visually
- It present results in such a manner that it is meaningful for the user
- It provides a checklist of symptoms which are a signal of neuropathic pain in
- It is possible to save the pain rate on a pain scale of 1 tot 10

The tasks 1, 2a, 2c, 4 and 5, which scored high in effectiveness tackled the requirements
-Login as a user
-Adding new value for the pain threshold (conducting the test with the device)
- Graph axes have to be readable
- It provides the user with information about the preferable measurement location
- The actual measurement location will have to be saved
- It saves the results generated by the Ambustim's QST automatically
- It saves this results per patient
- It is possible to reject measurement values if a reason will been added

Appendix D

- Only one piece of identifiable data will be saved for the doctor to recognise the patient in the system with which the user can distinguish whether the app shows the correct data
- Doctors have access to the results of multiple patients

Usability:
Sus scores below 64, are considered to be below average and therefore the target usability score for this prototype. The average SUS usability score of this prototype is 59,1 which is 4,9 points lower the target. This means the usability of this prototype is low. Looking at the individual usability scores, given by the participants individually, only 1 had given the prototype a score above 64.

Efficiency:
No targets were set for the efficiency. Therefore, the observer/designer of the app, completed the task. This will represent the time in which it is possible to complete the tasks if you are an advanced user of the system. This value will be a reference to see whether the tasks or flow of actions might cause confusions and need to be clearer.
Allow that participants need 2,5 as much time, because it is their first time. That is 150% more than the ideal case.

	efficiëntie score task 1	efficiëntie score task 2	efficiëntie score task 3	efficiëntie score task 4	efficiëntie score task 5
gemiddelde:	13,2	209,8	97,6	77	18,4
advanced user:	6	74	8	44	6
Efficiency (%)	120	183,51	1120	75	207

Table 3. Efficiency scores. The time difference (in %) from the “ideal situation”.

Only task 1 and task 4 have meet this requirement. This means the other tasks are not efficient enough. Especially task 4. It exceeds the limit of 150% with a value of 1120%, which is almost 8 times as much.

Discussion

Task 3 and 4 are similar: they both consist of adding a new value. Task 3 is the pain scale, task 4 is adding the pain threshold. Results of task 4 are probably more positive, both on efficiency and effectiveness. The difference in effectiveness might be caused by the fact that most of the participant noticed during task 3, how this task had to be completed. Next to this, if a participant was not able to add the pain scale in task 3, it is less likely that task 4 will be completed successful, because this means the participant did not made the link between clicking on the graph and adding a new value. Same holds for task 2b, if the participant was not able to find the value of the pain sale, it will probably not be able to add a new value. The difference in efficiency, an 1120% difference between ideal and the participant’s time of task 3 and the 75% of task 4, is also caused by this effect.

It seemed that many did not made the link between clicking on the graph and adding new values. Instinctively, almost all of them started with clicking on the graphs, in the exploration case. The menu item name “metingen (measurements)”, which actually leads to more visualisations of the data, clearly did not trigger the participants to click it at this stage.

Appendix D

The opposite happened for adding new values for the pain scale and threshold. A lot of participant kept clicking the menu bar (hoping for new items to appear?), when they were asked to add new data.

Another source of confusion, what the QR code scanning. This might be caused because this functionality could not be implemented in the mock-up software. That is why this caused some confusion when the participants had to login a new patient. In the patient page, there is a button “Scan patient code” to scan a new QR-code to log a new patient in. This option allows the user to login a new patient, without completely login out and in again by password. 3 out of the 5 participants however, did chose to completely logout and login again. This method is effective as well, but not as efficient. Because most participants chose this methods above the shorter one, there can be concluded that this flow of action after clicking “scan patient code” to login a new patient, was not clear enough.

Two participants (3 and 4) admitted they were not able to find one or more values during task 2. Participant 3 also stopped task 3 before it was competed.

Other returning remark on the design and usage were given by the participants. One was that the graphs show the most important information, was rather small. Especially the axes were hard to read. Another comment from one of the participants was that the first page should contain more information. The information you have to look for in task 2, should be quicker and easier to find. A doctor has not time to switch between different pages. One thought the front side and the back side of model of the human body was hard to distinguish, but it can be resolved by adding a face/eyes. One patient was wondering if there would be a list of all the patients available for the doctor, which is a function which is not implemented. This participant thinks it would be relevant and useful to have such a list, to switch between patients quickly.

Conclusion

The usability of the app is below average. From the efficiency scores and effectiveness scores found there can be concluded that the biggest need for improvements are related to following tasks:

- Finding general information such as, 1) the pain scale, 2) the pain threshold, 3) the body area the pain is located, 4) symptoms of neuropathic pain.
- Adding the pain scale
- Adding the pain threshold

These cases tackle the following requirements:

1. It shows the pain scale visually
2. It present results in such a manner that it is meaningful for the user
3. It provides a checklist of symptoms which are a signal of neuropathic pain in
4. It is possible to save the pain rate on a pain scale of 1 tot 10

The visualisation can be made clearer by enlarging them and especially enlarging the font size of the labels of the axes.

The first page should contain more information. A ‘home’ page can be constructed, which shows the most important graphs with the most important information.

Appendix D

The method to add new values should be cleared. There can be either a page or menu items which specifically indicates that a new value can be added if one will click there. Another option would be to add buttons below a graph to add a new value. Or a description which explains where to click to add new values clearly.

The login procedure might needs to be reconsidered. A list of patients to choose from and swhich might be more convenient. There should be though about whether and how the QR-code can still be applied with this method.

References

http://www.measuringu.com/sus.php

Appendix E

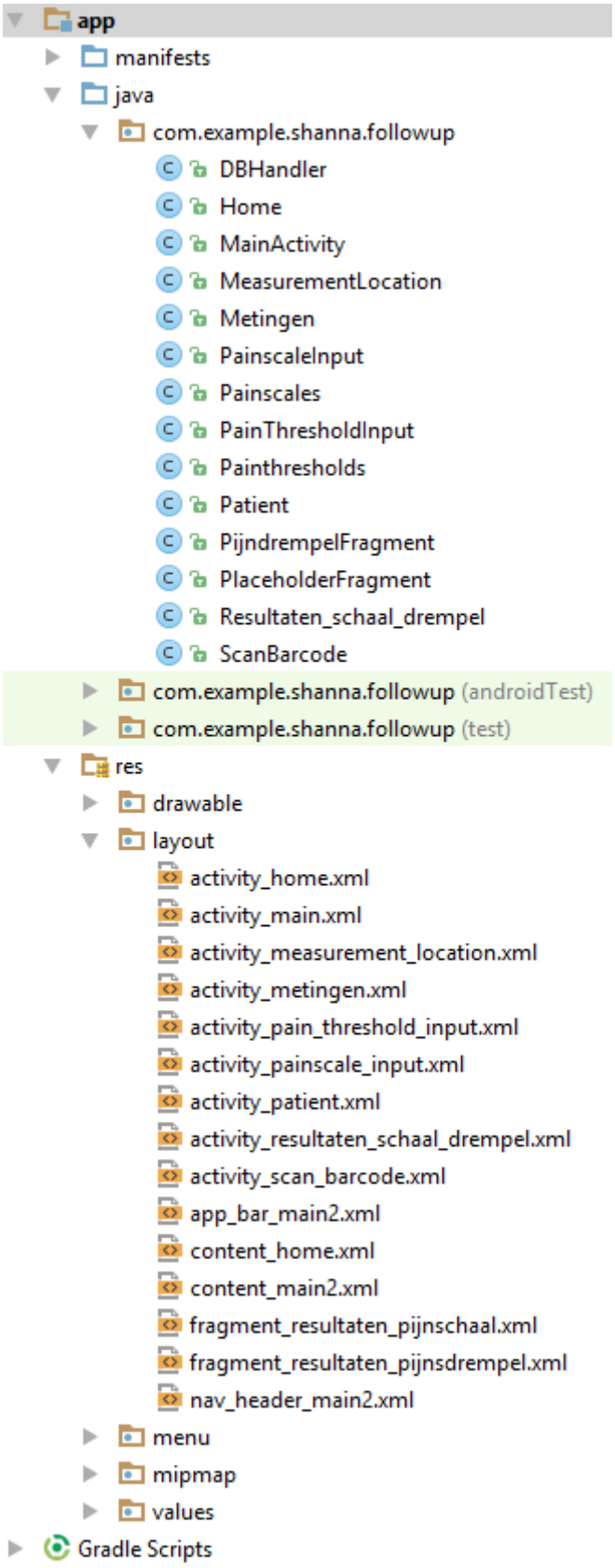
participant	Effectiviteit score (uit 9)	SUS	efficientie score task 1	efficientie score task 2	efficientie score task 3	efficientie score task 4	efficientie score task 5
1	6	60	4	281	202	87	11
2	8	45,5	3	102	13	48	40
3	4	77,5	27	324	140	149	13
4	5	57,5	21	127	58	50	18
5	6	55	11	215	75	51	10
Average	5,8	59,1	13,2	209,8	97,6	77	18,4
Advanced user			6	74	8	44	6

task	succes (out of 5)
Login	5
Find the locatio	5
Find last pain sc	1
Find date of las	5
Find threshold	0
Find pain sympt	2
Add pain score	3
Add pain thresh	4
Change patient	5

participant	Login	Gather information	Add pain score	Add pain threshold	Change patient
1	4	281	202	87	11
2	3	102	13	48	40
3	27	324	140	149	13
4	21	127	58	50	18
5	11	215	75	51	10
Average	13,2	209,8	97,6	77	18,4
Advanced use	6	74	8	44	6

Appendix F

Final tablet lay-out



Appendix F

Inhoud

Appendix F

The Layout (xml) files

Activity_main.xml

Layout of the login page

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.shanna.followup.MainActivity">

    <EditText
        android:id="@+id/inputLoginName"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true"
        android:ems="10" />

    <TextView
        android:id="@+id/loginMessage"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_above="@id/inputLoginName"
        android:layout_centerHorizontal="true"
        android:text="Log in bij FollowUP"
        android:textAppearance="@style/TextAppearance.AppCompat.Large" />

    <TextView
        android:id="@+id/loginNaam"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBaseline="@+id/inputLoginName"
        android:layout_below="@+id/loginMessage"
        android:layout_toLeftOf="@+id/inputLoginName"
        android:text="Login naam" />

    <TextView
        android:id="@+id/warningLoginName"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_toRightOf="@+id/inputLoginName"
        android:layout_alignBottom="@+id/inputLoginName"
        android:textColor="@color/colorAccentRouge" />

    <EditText
        android:id="@+id/inputLoginPassword"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/inputLoginName"
        android:layout_centerHorizontal="true"
        android:ems="10" />

    <TextView
        android:id="@+id/wachtwoord"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
```

Appendix F

```
        android:layout_alignBaseline="@+id/inputLoginPassword"
        android:layout_below="@+id/inputLoginName"
        android:layout_toLeftOf="@+id/inputLoginPassword"
        android:text="Wachtwoord" />

    <TextView
        android:id="@+id/warningLoginPassword"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_toRightOf="@+id/inputLoginPassword"
        android:layout_alignBottom="@+id/inputLoginPassword"
        android:textColor="@color/colorAccentRouge"/>

    <ImageView
        android:id="@+id/loginIcon"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_above="@id/loginMessage"
        android:layout_centerHorizontal="true"
        android:src="@drawable/app_icon" />

    <TextView
        android:id="@+id/succesMessage"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:textAppearance="@style/TextAppearance.AppCompat.Medium" />

    <Button
        android:id="@+id/loginButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/inputLoginPassword"
        android:layout_centerHorizontal="true"
        android:layout_margin="@dimen/text_spacing"
        android:background="@drawable/buttonshape_blue"
        android:ems="8"
        android:onClick="logUserIn"
        android:text="Login" />

    <RelativeLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/loginButton"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="@dimen/text_spacing">

        <TextView
            android:id="@+id/register"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Registreer "
            android:textColor="@color/colorAccentRouge" />

        <TextView
            android:id="@+id/eenNieuwePatient"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_toRightOf="@+id/register"
            android:text=" een nieuwe patiënt" />
```

Appendix F

```
    </RelativeLayout>
</RelativeLayout>
```

Activity_scan_barcode.xml

Page which camera activity and QR-code scan library.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.shanna.followup.ScanBarcode">


    <me.dm7.barcodescanner.zxing.ZXingScannerView
        android:id="@+id/barcodeScanner"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

    <TextView
        android:id="@+id/loginConfirm"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="@dimen/text_spacing"
        android:layout_centerHorizontal="true"
        android:layout_alignParentTop="true"
        android:textAppearance="@style/Base.TextAppearance.AppCompat.Large"
        android:textColor="@color/colorAccent"
        android:text=""/>

    <TextView
        android:id="@+id/patientNummer"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_above="@+id/scanCodeButton"
        android:layout_toLeftOf="@+id/inputPatientNumber"
        android:layout_alignBaseline="@+id/inputPatientNumber"
        android:layout_centerHorizontal="true"
        android:layout_margin="4dp"
        android:text="Patient nummer"
        android:textColor="@color/colorPrimary"/>
    <EditText
        android:id="@+id/inputPatientNumber"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_above="@id/scanCodeButton"
        android:ems="10"/>
    <Button
        android:id="@+id/scanCodeButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_above="@+id/warningPatientNumber"
        android:text="Scan QR-code en log in"
        android:layout_margin= "8dp"
        android:ems="12"
        android:onClick="accessToPatientPage"
        android:background="@drawable/buttonshape_blue"/>
```

Appendix F

```
    <TextView
        android:id="@+id/warningPatientNumber"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_alignParentBottom="true"
        android:layout_margin="8dp"
        android:textColor="@color/colorAccent"
        />

</RelativeLayout>
```

Activity_home.xml

The main page: enables the menu bar to open. Include 2 other layout files for further content.

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.v4.widget.DrawerLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/drawer_layout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fitsSystemWindows="true"
    tools:openDrawer="start">

    <include
        layout="@layout/app_bar_main2"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

    <include
        layout="@layout/content_home"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginTop="@dimen/activity_horizontal_margin" />

    <android.support.design.widget.NavigationView
        android:id="@+id/nav_view"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_gravity="start"
        android:fitsSystemWindows="true"
        app:headerLayout="@layout/nav_header_main2"
        app:menu="@menu/activity_main2_drawer" />

</android.support.v4.widget.DrawerLayout>
```

Content_home.xml

The content of the home page behind the drawer menu.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:paddingTop="@dimen/userbar_padding_top">

    <LinearLayout
```

Appendix F

```
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:paddingTop="@dimen/userbar_padding_top"
        android:layout_weight="2">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_gravity="center"
            android:layout_weight="1"
            android:gravity="center"
            android:orientation="vertical">

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_above="@+id/homePatientTitle"
                android:text="Pagina van patiënt:"

android:textAppearance="@style/TextAppearance.AppCompat.Large" />

            <TextView
                android:id="@+id/homePatientTitle"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_centerVertical="true"
                android:text="null"

android:textAppearance="@style/TextAppearance.AppCompat.Medium"
                android:textColor="@color/material_drawer_secondary_text"
/>

            <Button
                android:id="@+id/loginNewPatient"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_below="@+id/homePatientTitle"

android:layout_marginTop="@dimen/nav_header_vertical_spacing"
                android:background="@drawable/buttonshape_blue"
                android:ems="12"
                android:text="Log nieuwe patiënt in" />

        </LinearLayout>

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:orientation="vertical">

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_gravity="center"
                android:text="Pijnlocatie"

android:textAppearance="@style/TextAppearance.AppCompat.Title" />

            <ImageView
                android:id="@+id/img_pijnLocatie"
```

Appendix F

```
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:layout_gravity="center"
                android:src="@drawable/dummies" />
        </LinearLayout>
    </LinearLayout>

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="1">
        <TextView
            android:id="@+id/section_label"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_centerHorizontal="true"
            android:textAppearance="@style/TextAppearance.AppCompat.Large"
            android:text="Ontwikkeling van de pijn"/>
        <TextView
            android:id="@+id/section_ondertitel"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_centerHorizontal="true"
            android:layout_below="@id/section_label"
            android:textAppearance="@style/TextAppearance.AppCompat.Small"
            android:text="tijd gegeven als: maand-dag tijdstip"/>

        <com.github.mikephil.charting.charts.LineChart
            android:id="@+id/chart"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_below="@id/section_ondertitel"/>
    </RelativeLayout>
</LinearLayout>
```

Activity_metingen.xml

Layout views with a text and an image. In Java class on click listeners make the layouts clickable to go to other actions.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <include
        layout="@layout/content_main2"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />


    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="false"
        android:layout_alignParentRight="false"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginBottom="@dimen/text_spacing"
        android:layout_marginTop="42dp"
```

Appendix F

```
        android:text="Voeg meting toe"
        android:textAppearance="?android:attr/textAppearanceLarge" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_below="@+id/textView2"
        android:layout_marginBottom="@dimen/userbar_padding_top"
        android:orientation="vertical">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="0dip"
            android:layout_weight="1">
            <RelativeLayout
                android:id="@+id/layout_pijndrempel_toevoegen"
                android:layout_width="0dip"
                android:layout_height="match_parent"
                android:padding="@dimen/text_spacing"
                android:layout_weight="1">
                <TextView
                    android:id="@+id/pijndrempel_toevoegen"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:layout_centerHorizontal="true"

android:textAppearance="@style/TextAppearance.AppCompat.Title"
                    android:text="Pijndrempel"/>
                <ImageView
                    android:id="@+id/pijndrempel_menu"
                    android:layout_below="@+id/pijndrempel_toevoegen"
                    android:layout_width="match_parent"
                    android:layout_height="match_parent"
                    android:src="@drawable/pijndrempel"/>
            </RelativeLayout>
            <RelativeLayout
                android:id="@+id/layout_pijnschaal_toevoegen"
                android:layout_width="0dip"
                android:layout_height="match_parent"
                android:padding="@dimen/text_spacing"
                android:layout_weight="1">
                <TextView
                    android:id="@+id/pijnschaal_toevoegen"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:layout_centerHorizontal="true"

android:textAppearance="@style/TextAppearance.AppCompat.Title"
                    android:text="Pijnschaal"/>
                <ImageView
                    android:layout_width="match_parent"
                    android:layout_height="match_parent"
                    android:layout_below="@+id/pijnschaal_toevoegen"
                    android:src="@drawable/pijnschaal"/>
            </RelativeLayout>
        </LinearLayout>
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="0dip"
            android:layout_weight="1">
            <RelativeLayout
```

Appendix F

```
                android:id="@+id/layout_vragenlijst"
                android:layout_width="0dip"
                android:layout_height="match_parent"
                android:padding="@dimen/text_spacing"
                android:layout_weight="1">
                <TextView
                    android:id="@+id/vragenlijst"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:layout_centerHorizontal="true"

android:textAppearance="@style/TextAppearance.AppCompat.Title"
                    android:text="DN4 vragenlijst"/>
                <ImageView
                    android:layout_width="match_parent"
                    android:layout_height="match_parent"
                    android:layout_below="@+id/vragenlijst"
                    android:src="@drawable/pijnvragenlijst"/>
            </RelativeLayout>
            <RelativeLayout
                android:id="@+id/layout_interventies_toevoegen"
                android:layout_width="0dip"
                android:layout_height="match_parent"
                android:layout_weight="1"
                android:padding="@dimen/text_spacing">
                <TextView
                    android:id="@+id/interventies_toevoegen"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:layout_centerHorizontal="true"

android:textAppearance="@style/TextAppearance.AppCompat.Title"
                    android:text="Interventie"/>
                <ImageView
                    android:layout_width="match_parent"
                    android:layout_height="match_parent"
                    android:layout_below="@id/interventies_toevoegen"
                    android:src="@drawable/interventies"/>
            </RelativeLayout>
        </LinearLayout>
    </LinearLayout>
</RelativeLayout>
```

Activity_measurement_location.xml

Page in which the location of measurement can be saved before the QST will start. Page is not functional, it only includes images to represent the functionality.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.shanna.followup.MeasurementLocation">

    <include
        layout="@layout/content_main2"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

    <LinearLayout
```

Appendix F

```
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:paddingBottom="@dimen/userbar_padding_top"
        android:paddingLeft="@dimen/activity_vertical_margin"
        android:paddingRight="@dimen/activity_vertical_margin"
        android:paddingTop="@dimen/userbar_padding_top">

        <RelativeLayout
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:layout_weight="1">
            <TextView
                android:id="@+id/txt_meetlocatie"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_centerHorizontal="true"
                android:text="Advies locatie voor meting"

android:textAppearance="@style/TextAppearance.AppCompat.Title" />
            <LinearLayout
                android:id="@+id/buttons"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_below="@+id/txt_meetlocatie"
                android:orientation="vertical">
            <ImageButton
                android:id="@+id/btn_undo"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:src="@drawable/btn_undo" />

            <ImageButton
                android:id="@+id/btn_add"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_below="@+id/btn_undo"
                android:src="@drawable/btn_add" />

            <ImageButton
                android:id="@+id/btn_remove"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_below="@+id/btn_add"
                android:src="@drawable/btn_remove" />

            <ImageButton
                android:id="@+id/btn_move"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_below="@+id/btn_remove"
                android:src="@drawable/btn_move" />
            </LinearLayout>


            <ImageView
                android:id="@+id/img_aangevenMeetlocatie"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_below="@+id/txt_meetlocatie"
                android:layout_toRightOf="@+id/buttons"
                android:src="@drawable/dummies_meetlocatie" />
```

Appendix F

```
        </RelativeLayout>

        <RelativeLayout
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:layout_weight="2">

            <TextView
                android:id="@+id/txt_advieslocatie"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Advies locatie voor meting"

android:textAppearance="@style/TextAppearance.AppCompat.Title" />

            <ImageView
                android:id="@+id/img_advieslocatie"
                android:layout_width="match_parent"
                android:layout_height="300dp"
                android:layout_below="@id/txt_advieslocatie"
                android:src="@drawable/dummies" />

            <Button
                android:id="@+id/btn_next"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_below="@+id/img_advieslocatie"
                android:background="@drawable/buttonshape_blue"
                android:ems="8"
                android:text="Volgende" />

        </RelativeLayout>
    </LinearLayout>
</RelativeLayout>
```

Activity_pain_threshold_input.xml

Buttons to go to the QST app and to accept or reject this value. Added values printed as text as feedback.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_vertical_margin"
    android:paddingRight="@dimen/activity_vertical_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.shanna.followup.PainThresholdInput">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="1">

        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="horizontal">
```

Appendix F

```
            <Button
                android:id="@+id/btn_StartQST"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_marginBottom="@dimen/text_spacing"
                android:layout_marginTop="@dimen/text_spacing"
                android:background="@drawable/buttonshape_blue"
                android:ems="12"
                android:text="Start NociTRACK eQST" />

            <TextView
                android:id="@+id/txt_receivedPijndrempel"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_below="@id/btn_StartQST"
                android:layout_marginBottom="@dimen/text_spacing"
                android:layout_marginTop="@dimen/text_spacing"
                android:text="Geen waarde voor pijnprempel gevonden"

android:textAppearance="@style/TextAppearance.AppCompat.Medium" />

            <Button
                android:id="@+id/btn_goedkeurenPijndrempel"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_below="@id/txt_receivedPijndrempel"
                android:layout_margin="@dimen/text_spacing"
                android:background="@drawable/buttonshape_blue"
                android:ems="8"
                android:text="Goedkeuren" />

            <Button
                android:id="@+id/btn_afkeurenPijndrempel"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_below="@+id/btn_goedkeurenPijndrempel"
                android:layout_margin="@dimen/text_spacing"
                android:background="@drawable/buttonshape_rouge"
                android:ems="8"
                android:text="Afkeuren" />


        </RelativeLayout>
    </RelativeLayout>

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="1">

        <TextView
            android:id="@+id/confirmation"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginBottom="@dimen/text_spacing"
            android:layout_marginTop="@dimen/text_spacing" />

        <TextView
            android:id="@+id/printData"
            android:layout_width="wrap_content"
```

Appendix F

```
            android:layout_height="wrap_content"
            android:layout_below="@id/confirmation"
            android:layout_margin="@dimen/text_spacing" />

        <!-- <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentBottom="true"
            android:layout_alignParentRight="true"
            android:background="@drawable/buttonshape_blue"
            android:layout_margin="@dimen/nav_header_height"
            android:text="OK" /> -->
    </RelativeLayout>
</LinearLayout>
```

Activity_painscale_input.xml

Page with input text to add a value within the range of 1 to 10.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.shanna.followup.PainscaleInput">

    <TextView
        android:id="@+id/title_pijnschaal_toevoegen"
        android:layout_marginTop="@dimen/activity_vertical_margin"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textAppearance="@style/TextAppearance.AppCompat.Title"
        android:text="Vul de waarde voor de pijnschaal in."/>

    <TextView
        android:id="@+id/dateTime"
        android:layout_below="@+id/title_pijnschaal_toevoegen"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="32dp"
        android:text="date and time" />

    <Button
        android:id="@+id/addButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/dateTime"
        android:layout_toEndOf="@+id/editText"
        android:layout_toRightOf="@+id/editText"
        android:onClick="addButtonClicked"
        android:text="Voeg toe" />

    <EditText
        android:id="@+id/editText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
```

Appendix F

```
        android:layout_below="@+id/dateTime"
        android:ems="10"
        android:maxLength="10"
        android:layout_marginBottom="@dimen/activity_horizontal_margin"
        android:inputType="number" />

    <TextView
        android:id="@+id/confirmation"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/addButton"
        android:layout_margin="32dp"
        android:text="" />

</RelativeLayout>
```

Activity_resultaten_schaal_drempel.xml

Activity in which 2 fragments are included. Swipe action allows switching between fragments

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.design.widget.CoordinatorLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main_content"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fitsSystemWindows="true"
    tools:context="com.example.shanna.followup.Resultaten_schaal_drempel">

    <include
        layout="@layout/content_main2"
        android:layout_width="match_parent"
        android:layout_height="match_parent"/>

    <android.support.v4.view.ViewPager
        android:id="@+id/container"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:layout_behavior="@string/appbar_scrolling_view_behavior" />

</android.support.design.widget.CoordinatorLayout>
```

Fragment_resultaten_pijnschaal.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:paddingBottom="@dimen/userbar_padding_top"
    android:paddingLeft="@dimen/activity_vertical_margin"
    android:paddingRight="@dimen/activity_vertical_margin"
    android:paddingTop="@dimen/userbar_padding_top"
    tools:context="com.example.shanna.followup.PlaceholderFragment">

    <TextView
        android:id="@+id/section_label"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
```

Appendix F

```
        android:textAppearance="@style/TextAppearance.AppCompat.Large"
        android:text="Ontwikkeling pijnschaal"/>
    <TextView
        android:id="@+id/section_ondertitel"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:textAppearance="@style/TextAppearance.AppCompat.Small"
        android:text="tijd gegeven als: maand-dag tijdstip"/>

    <FrameLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="2">

        <com.github.mikephil.charting.charts.BarChart
            android:id="@+id/pijnschaalBarChart"
            android:layout_width="match_parent"
            android:layout_height="match_parent"

android:layout_below="@id/editText"></com.github.mikephil.charting.charts.B
arChart>

    </FrameLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="3"
        android:orientation="horizontal">
        <CalendarView
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_weight="1"></CalendarView>


        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_weight="1">

            <TextView
                android:id="@+id/title_pijnlocatie"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_alignParentTop="true"
                android:layout_centerInParent="true"
                android:text="Pijn locatie"

android:textAppearance="@style/TextAppearance.AppCompat.Title" />

            <ImageView
                android:id="@+id/img_meetlocatie"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:layout_below="@+id/title_pijnlocatie"
                android:src="@drawable/dummies" />
        </RelativeLayout>

    </LinearLayout>
```

Appendix F

```
</LinearLayout>
```

Fragment_resultaten_pijndrempel.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:paddingBottom="@dimen/userbar_padding_top"
    android:paddingLeft="@dimen/activity_vertical_margin"
    android:paddingRight="@dimen/activity_vertical_margin"
    android:paddingTop="@dimen/userbar_padding_top"
    tools:context="com.example.shanna.followup.PlaceholderFragment">

    <TextView
        android:id="@+id/section_label"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:textAppearance="@style/TextAppearance.AppCompat.Large"
        android:text="Pijndrempel"/>
    <TextView
        android:id="@+id/section_ondertitel"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:textAppearance="@style/TextAppearance.AppCompat.Small"
        android:text="tijd gegeven als: maand-dag tijdstip"/>

    <FrameLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="2">

        <com.github.mikephil.charting.charts.LineChart
            android:id="@+id/pijndrempelLineChart"
            android:layout_width="match_parent"
            android:layout_height="match_parent"

android:layout_below="@id/editText"></com.github.mikephil.charting.charts.L
ineChart>

    </FrameLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="3"
        android:orientation="horizontal">
        <RelativeLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1">
        <TextView
            android:id="@+id/title_vorigeMeetlocatie"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_centerHorizontal="true"
            android:textAppearance="@style/TextAppearance.AppCompat.Title"
            android:text="Vorige meetlocatie"
```

Appendix F

```
            />
        <ImageView
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_below="@+id/title_vorigeMeetlocatie"
            android:src="@drawable/dummies_meetlocatie"/>
        </RelativeLayout>
        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_weight="1">
            <TextView
                android:id="@+id/title_pijnlocatie"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_alignParentTop="true"
                android:layout_centerInParent="true"
                android:text="Pijn locatie"

android:textAppearance="@style/TextAppearance.AppCompat.Title" />

            <ImageView
                android:id="@+id/img_meetlocatie"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:layout_below="@+id/title_pijnlocatie"
                android:src="@drawable/dummies" />
        </RelativeLayout>
    </LinearLayout>
</LinearLayout>
```

Activity_patient.xml

Patient overview: includes name of the patient and the symptoms of the pain, which can be ticked off from the list with checkboxes.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.shanna.followup.Patient">

    <include
        layout="@layout/content_main2"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:paddingBottom="@dimen/userbar_padding_top">

        <LinearLayout
            android:layout_width="0dp"
            android:layout_height="match_parent"
            android:layout_gravity="center"
            android:layout_weight="1"
            android:gravity="center"
            android:orientation="vertical">
```

Appendix F

```
            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_above="@+id/homePatientTitle"
                android:text="Pagina van patiënt:"

android:textAppearance="@style/TextAppearance.AppCompat.Large" />

            <TextView
                android:id="@+id/dePatientTitle"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_centerVertical="true"
                android:text="null"

android:textAppearance="@style/TextAppearance.AppCompat.Medium"
                android:textColor="@color/material_drawer_secondary_text"
/>

            <Button
                android:id="@+id/loginNewPatient"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_below="@+id/homePatientTitle"

android:layout_marginTop="@dimen/nav_header_vertical_spacing"
                android:background="@drawable/buttonshape_blue"
                android:ems="12"
                android:text="Log nieuwe patiënt in" />
        </LinearLayout>

        <LinearLayout
            android:layout_width="0dp"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:layout_marginTop="@dimen/userbar_icon_size"
            android:orientation="vertical">

            <TextView
                android:id="@+id/title_dn4"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"

android:textAppearance="@style/TextAppearance.AppCompat.Large"
                android:text="DN4 vragenlijts" />
            <TextView
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:text="Heeft de pijn één of meerderen van de
volgende kenmerken?"/>

            <CheckBox
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"

android:textAppearance="@style/TextAppearance.AppCompat.Small"
                android:text="Branderig gevoel"
                android:id="@+id/checkBox1a" />
            <CheckBox
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
```

Appendix F

```
android:textAppearance="@style/TextAppearance.AppCompat.Small"
                android:text="Pijnlijk koud gevoel"
                android:id="@+id/checkBox1b" />
            <CheckBox
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"

android:textAppearance="@style/TextAppearance.AppCompat.Small"
                android:text="Elektrische schokken"
                android:id="@+id/checkBox1c" />
            <TextView
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:text="Gaat de pijn gepaard met de volgende
symptomen in pijngebied?"/>

            <CheckBox
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"

android:textAppearance="@style/TextAppearance.AppCompat.Small"
                android:text="Tintelen"
                android:id="@+id/checkBox2a" />
            <CheckBox
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"

android:textAppearance="@style/TextAppearance.AppCompat.Small"
                android:text="Prikken"
                android:id="@+id/checkBox2b" />
            <CheckBox
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"

android:textAppearance="@style/TextAppearance.AppCompat.Small"
                android:text="Doof gevoel"
                android:id="@+id/checkBox2c" />
            <CheckBox
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"

android:textAppearance="@style/TextAppearance.AppCompat.Small"
                android:text="Jeuk"
                android:id="@+id/checkBox2d" />
            <TextView
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:text="Is de pijn in een gebeid waar het lichamelijk
onderzoek de volgende kenmerken vertoont?"/>

            <CheckBox
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"

android:textAppearance="@style/TextAppearance.AppCompat.Small"
                android:text="Hypo-esthesie bij aanraking"
                android:id="@+id/checkBox3a" />
            <CheckBox
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
```

Appendix F

```
android:textAppearance="@style/TextAppearance.AppCompat.Small"
                android:text="Hypo-esthesie bij prikken"
                android:id="@+id/checkBox3b" />
            <TextView
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:text="Wordt de pijn in het pijnlijke gebied
veroorzaakt of verergerd door"/>

            <CheckBox
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"

android:textAppearance="@style/TextAppearance.AppCompat.Small"
                android:text="Wrijven"
                android:id="@+id/checkBox4" />

        </LinearLayout>
    </LinearLayout>
</RelativeLayout>
```

Content_main2.xml

Page which is included in most layout files. It shows the user and patient which are logged in.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/appbar_padding_top"
    android:paddingRight="@dimen/activity_vertical_margin"
    android:paddingTop="@dimen/appbar_padding_top"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:context=".Home"
    tools:showIn="@layout/app_bar_main2">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <ImageView
            android:id="@+id/icon_patient"
            android:layout_width="@dimen/userbar_icon_size"
            android:layout_height="@dimen/userbar_icon_size"
            android:src="@drawable/patient_icon"/>
        <TextView
            android:id="@+id/bar_patientName"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_toRightOf="@id/icon_patient"
            android:layout_centerVertical="true"
            android:text="patient naam"/>
        <ImageView
            android:id="@+id/icon_doctor"
            android:layout_width="@dimen/userbar_icon_size"
            android:layout_height="@dimen/userbar_icon_size"
            android:layout_marginLeft="@dimen/appbar_padding_top"
```

Appendix F

```
            android:src="@drawable/app_icon"
            android:layout_toRightOf="@id/bar_patientName"/>
        <TextView
            android:id="@+id/bar_doctorName"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_toRightOf="@+id/icon_doctor"
            android:layout_centerVertical="true"
            android:text="naam dokter"
            />
    </RelativeLayout>

    <TextView
        android:id="@+id/nociTRACK"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentBottom="true"
        android:text="NOCITRACK"
        android:textAppearance="@style/TextAppearance.AppCompat.Large" />
</RelativeLayout>
```

Nav_header_main2.xml

Specifies header of the menu (part above the text)

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="@dimen/nav_header_height"
    android:background="@drawable/side_nav_bar"
    android:gravity="bottom"
    android:orientation="vertical"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:theme="@style/ThemeOverlay.AppCompat.Dark">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:paddingTop="@dimen/nav_header_vertical_spacing"
        android:src="@drawable/app_icon" />

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:paddingTop="@dimen/nav_header_vertical_spacing"
        android:text="@string/app_name"
        android:textAppearance="@style/TextAppearance.AppCompat.Body1" />

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="android.studio@android.com" />

</LinearLayout>
```

Appendix F

Java Classes

MainActivity.java

```
package com.example.shanna.followup;

import android.content.Intent;
import android.net.Uri;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.text.InputType;
import android.text.Layout;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.RelativeLayout;
import android.widget.TextView;

import java.util.Objects;

public class MainActivity extends AppCompatActivity {
    EditText inputLoginName;
    EditText inputLoginPassword;
    TextView registerPatient;
    TextView loginNameWarning;
    TextView passwordWarning;
    TextView loginConfirm;

    String QRgenerateURL;
    Intent goToScanner;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        QRgenerateURL = "http://www.qrstuff.com/";
        goToScanner = new Intent(this, ScanBarcode.class);

        inputLoginName = (EditText)findViewById(R.id.inputLoginName);
        inputLoginPassword = (EditText)
findViewById(R.id.inputLoginPassword);
        inputLoginPassword.setInputType(InputType.TYPE_CLASS_TEXT |
InputType.TYPE_TEXT_VARIATION_PASSWORD);

        registerPatient = (TextView) findViewById(R.id.register);
        registerPatient.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Uri uri = Uri.parse(QRgenerateURL);
                Intent intent = new Intent(Intent.ACTION_VIEW, uri);
                startActivity(intent);
            }
        });

        loginNameWarning = (TextView)findViewById(R.id.warningLoginName);
        passwordWarning =
(TextView)findViewById(R.id.warningLoginPassword);
        loginConfirm = (TextView)findViewById(R.id.succesMessage);
    }
```

Appendix F

```
    public void logUserIn(View view){
        String inputPassword = inputLoginPassword.getText().toString();
        String inputName = inputLoginName.getText().toString();
        loginNameWarning.setText("");
        passwordWarning.setText("");
        if(!inputName.equals("") && !inputPassword.equals("")){
            loginConfirm.setText("Login Succesvol!");
            goToScanner.putExtra("userName", inputName);
            startActivity(goToScanner);
                    }
        else {
            if (inputName.equals("")){
                loginNameWarning.setText("Vul een geldig gebruikersnaam
is");
            }
            if (inputPassword.equals("")){
                passwordWarning.setText("Dit wachtwoord is ongeldig");
            }
            loginConfirm.setText("Is niet in staat in te loggen. Ben u
aangemeld als gebruiker?");
        }
    }
}
```

ScanBarcode

```
package com.example.shanna.followup;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;

import com.google.zxing.Result;

import me.dm7.barcodescanner.zxing.ZXingScannerView;

public class ScanBarcode extends AppCompatActivity implements
ZXingScannerView.ResultHandler {
    TextView nameUser;
    TextView warning;
    EditText patientNumber;
    public String userName;
    Intent data;
    private ZXingScannerView barcodescanner;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_scan_barcode);
        nameUser = (TextView)findViewById(R.id.loginConfirm);
        warning = (TextView)findViewById(R.id.warningPatientNumber);
        patientNumber = (EditText)findViewById(R.id.inputPatientNumber);
        barcodescanner =
(ZXingScannerView)findViewById(R.id.barcodeScanner);

        data = getIntent();
        userName = data.getStringExtra("userName");
```

Appendix F

```
        nameUser.setText("Welcome " + userName);
    }

    @Override
    public void onResume(){ //handles camera action
        super.onResume();
        barcodescanner.setResultHandler(this);
        barcodescanner.startCamera();
    }

    @Override
    public void onPause(){
        super.onPause();
        barcodescanner.stopCamera();
    }

    public void accessToPatientPage(View view){
        String inputPassword = patientNumber.getText().toString();
        String loginName = userName;
        Log.i("user name", loginName);
        if(inputPassword.equals("") ^ inputPassword.equals(" ") ){
            warning.setText("Vul eerst het patiëntnummer in");
        }
        else {
            warning.setText("Succes");
            Intent goToHome = new Intent(this,Home.class);
            goToHome.putExtra("patientName", inputPassword);
            goToHome.putExtra("userName", loginName);
            startActivity(goToHome);
        }
    }

    @Override
    public void handleResult(Result result) {
        barcodescanner.stopCamera();
        Log.i("barcode", result.getText());
        if (result.getText().length() > 0){
            Intent goToHome = new Intent(this,Home.class);
            goToHome.putExtra("patientName", result.getText());
            goToHome.putExtra("userName", userName);
            startActivity(goToHome);
        }
    }
}
```

Home.java

```
package com.example.shanna.followup;

import android.content.Intent;
import android.graphics.Color;
import android.os.Bundle;
import android.support.design.widget.NavigationView;
import android.support.v4.view.GravityCompat;
import android.support.v4.widget.DrawerLayout;
import android.support.v7.app.ActionBarDrawerToggle;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.util.Log;
import android.view.Menu;
import android.view.MenuItem;
```

Appendix F

```
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

import com.github.mikephil.charting.charts.LineChart;
import com.github.mikephil.charting.data.Entry;
import com.github.mikephil.charting.data.LineData;
import com.github.mikephil.charting.data.LineDataSet;
import com.github.mikephil.charting.interfaces.datasets.ILineDataSet;
import com.github.mikephil.charting.utils.ColorTemplate;

import java.util.ArrayList;
import java.util.Collections;
import java.util.List;

//TODO grafiek maken met beide waarden

public class Home extends AppCompatActivity
        implements NavigationView.OnNavigationItemSelectedListener {
    //variables for content_home
    TextView homeText;
    Button patientLogin;
    public String patientNumber;
    Intent data;
    Intent backToScanner;
    LineChart combinedLineChart;
    DBHandler dbHandler;

    //variables for content_main2
    TextView patientName;
    TextView doctorName;
    public String userName;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_home);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);

         //content_main2
        data = getIntent();
        userName = data.getStringExtra("userName");
        patientNumber = data.getStringExtra("patientName");
        homeText = (TextView) findViewById(R.id.homePatientTitle);
        homeText.setText(patientNumber);

        setUserBar();

        //content_home

        backToScanner = new Intent(this, ScanBarcode.class);

        dbHandler = new DBHandler(this, null, null, 0);
        combinedLineChart = (LineChart) findViewById(R.id.chart);
        patientLogin = (Button) findViewById(R.id.loginNewPatient);
        patientLogin.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                backToScanner();
            }
```

Appendix F

```
        });


        //line chart
        getData();

        combinedLineChart.invalidate();

        //menu drawer
        DrawerLayout drawer = (DrawerLayout)
findViewById(R.id.drawer_layout);
        ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
                this, drawer, toolbar, R.string.navigation_drawer_open,
R.string.navigation_drawer_close);
        drawer.setDrawerListener(toggle);
        toggle.syncState();

        NavigationView navigationView = (NavigationView)
findViewById(R.id.nav_view);
        navigationView.setNavigationItemSelectedListener(this);
    }

    private void setUserBar() {
        //content_main2
        patientName = (TextView) findViewById(R.id.bar_patientName);
        patientName.setText(patientNumber);
        doctorName = (TextView) findViewById(R.id.bar_doctorName);
        doctorName.setText(userName);
    }

    private void getData() {

                //get x values
        List<String> pijnschaalX = dbHandler.pijnschaalXData();
        List<String> pijndrempelX = dbHandler.thresholdXData();
        ArrayList<String> xVals = new ArrayList<>();
        for (int i =0 ; i<pijnschaalX.size(); i++ ){
            xVals.add(pijnschaalX.get(i));
        }
        for (int i =0 ; i<pijndrempelX.size(); i++ ){
            xVals.add(pijndrempelX.get(i));
        }

        Collections.sort(xVals);

        //get y data of pijnschaal
        List<Float> pijnschaalYquery = dbHandler.pijnschaalYData();

        ArrayList<Entry> pijnschaalData = new ArrayList<Entry>();
        for (int i = 0; i < pijnschaalYquery.size(); i++) {

            pijnschaalData.add(new Entry(pijnschaalYquery.get(i),
xVals.indexOf(pijnschaalX.get(i))));
            String mystring = pijnschaalData.get(i).toString();
            Log.i("TAG",mystring);
        }

        List<Float> pijndrempelYquery = dbHandler.thresholdYData();
        ArrayList<Entry> pijndrempelData = new ArrayList<Entry>();
        for (int i = 0; i < pijndrempelYquery.size(); i++) {
            pijndrempelData.add(new Entry((pijndrempelYquery.get(i)*10),
```

Appendix F

```
xVals.indexOf(pijndrempelX.get(i))));
            String mystring = pijndrempelData.get(i).toString();
            Log.i("TAG2",mystring);
        }

        //insert the data into the graph
        ArrayList<ILineDataSet> lineDataSets = new ArrayList<>();
        LineDataSet dataSetPijnschaal = new LineDataSet(pijnschaalData,
"pijnschaal");
        LineDataSet dataSetPijndrempel = new LineDataSet(pijndrempelData,
"pijndrempel");
        lineDataSets.add(dataSetPijndrempel);
        lineDataSets.add(dataSetPijnschaal);

        dataSetPijndrempel.setDrawValues(false);
        dataSetPijndrempel.setColor(Color.rgb(228,79,80));
        dataSetPijnschaal.setDrawValues(false);

        combinedLineChart.setData(new LineData(xVals, lineDataSets));
        combinedLineChart.getLegend().setTextSize(20);
        combinedLineChart.invalidate();
    }

    private void backToScanner() {
        patientNumber = null;
        backToScanner.putExtra("userName", userName);
        startActivity(backToScanner);
    }

    @Override
    public void onBackPressed() {
        DrawerLayout drawer = (DrawerLayout)
findViewById(R.id.drawer_layout);
        if (drawer.isDrawerOpen(GravityCompat.START)) {
            drawer.closeDrawer(GravityCompat.START);
        } else {
            super.onBackPressed();
        }
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is
present.
        getMenuInflater().inflate(R.menu.main2, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so long
        // as you specify a parent activity in AndroidManifest.xml.
        int id = item.getItemId();

        //noinspection SimplifiableIfStatement
        if (id == R.id.action_settings) {
            return true;
        }

        return super.onOptionsItemSelected(item);
```

Appendix F

```
    }

    @SuppressWarnings("StatementWithEmptyBody")
    @Override
    public boolean onNavigationItemSelected(MenuItem item) {
        // Handle navigation view item clicks here.
        int id = item.getItemId();

        if (id == R.id.homePage) {
            Intent intent = new Intent(this, Home.class);
            intent.putExtra("userName", userName);
            intent.putExtra("patientName", patientNumber);
            startActivity(intent);
        } else if (id == R.id.metingenPage) {
            Intent intent = new Intent(this, Metingen.class);
            intent.putExtra("userName", userName);
            intent.putExtra("patientName", patientNumber);
            startActivity(intent);
        } else if (id == R.id.resultatenPage) {
            Intent intent = new Intent(this,
Resultaten_schaal_drempel.class);
            intent.putExtra("userName", userName);
            intent.putExtra("patientName", patientNumber);
            startActivity(intent);

        } else if (id == R.id.patientPage) {
            Intent intent = new Intent(this, Patient.class);
            intent.putExtra("userName", userName);
            intent.putExtra("patientName", patientNumber);
            startActivity(intent);

        } else if (id == R.id.nav_logout) {
            Intent intent = new Intent(this, MainActivity.class);
            startActivity(intent);

        } else if (id == R.id.newPatientPage) {
            Intent intent = new Intent(this, ScanBarcode.class);
            intent.putExtra("userName", userName);
            startActivity(intent);

        } else if (id == R.id.helpPage) {

        }

        DrawerLayout drawer = (DrawerLayout)
findViewById(R.id.drawer_layout);
        drawer.closeDrawer(GravityCompat.START);
        return true;
    }
}
```

Metingen.java

```
package com.example.shanna.followup;

import android.content.DialogInterface;
import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.LinearLayout;
```

Appendix F

```
import android.widget.RelativeLayout;
import android.widget.TextView;

//TODO menu item in de toolbar in plaats van de terug knop (die is
overbodig bij android applicaties)
//TODO icon in de menubar verbeteren


public class Metingen extends AppCompatActivity {
    int requestCode = 1;
    RelativeLayout addPijnschaal;
    RelativeLayout addPijndrempel;
    RelativeLayout vragenlijst;

    //variables for content_main2
    Intent data;
    TextView patientName;
    TextView doctorName;
    public String userName;
    public String patientNumber;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_metingen);
        addPijnschaal = (RelativeLayout)
findViewById(R.id.layout_pijnschaal_toevoegen);
        addPijndrempel = (RelativeLayout)
findViewById(R.id.layout_pijndrempel_toevoegen);
        vragenlijst = (RelativeLayout)
findViewById(R.id.layout_vragenlijst);

        addPijnschaal.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                addPijnschaal(v);
            }
        });

        addPijndrempel.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                addPijndrempel(v);
            }
        });

        vragenlijst.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                vragenlijst(v);
            }
        });

        data = getIntent();
        patientName = (TextView) findViewById(R.id.bar_patientName);
        doctorName = (TextView) findViewById(R.id.bar_doctorName);
        setUserBar();

    }

    private void setUserBar() {
```

Appendix F

```
        //content_main2
        userName = data.getStringExtra("userName");
        patientNumber = data.getStringExtra("patientName");
        patientName.setText(patientNumber);
        doctorName.setText(userName);
    }

    public void addPijnschaal(View view) { //go to activity to add the pain
scale value
        Intent intent = new Intent(this, PainscaleInput.class);
        startActivityForResult(intent, requestCode);
    }

    public void addPijndrempel(View view) { //go to activity to add the
pain scale value
        Intent intent = new Intent(this, MeasurementLocation.class);
        startActivity(intent);
    }

    public void vragenlijst(View view) { //go to activity to add the pain
scale value
        Intent intent = new Intent(this, Patient.class);
        intent.putExtra("patientName", patientNumber);
        intent.putExtra("userName", userName);
        startActivity(intent);
    }

}
```

MeasurementLocation.java

```
package com.example.shanna.followup;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class MeasurementLocation extends AppCompatActivity {
    Button nextButton;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_measurement_location);

        nextButton = (Button) findViewById(R.id.btn_next);
        nextButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                goToAddThreshold();
            }
        });
    }

    private void goToAddThreshold() {
        Intent intent = new Intent(this, PainThresholdInput.class);
        startActivity(intent);
    }
}
```

Appendix F

PainThresholdInput.java

```
package com.example.shanna.followup;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

import java.text.SimpleDateFormat;
import java.util.Calendar;

public class PainThresholdInput extends AppCompatActivity {
    Button startQST;
    Button goedkeuren;
    Button afkeuren;
    TextView confirmInput;
    TextView printData;
    TextView receivedThreshold;

    float requestThreshold;
    public static final int REQUEST_CODE_THRESHOLD = 1;

    DBHandler dbHandler;
    Calendar c;
    SimpleDateFormat df;
    public float newResult;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_pain_threshold_input);

        confirmInput = (TextView) findViewById(R.id.confirmation);
        printData = (TextView) findViewById(R.id.printData);
        receivedThreshold=
(TextView)findViewById(R.id.txt_receivedPijndrempel);

        dbHandler = new DBHandler(this, null, null, 0);
        c = Calendar.getInstance();
        df = new SimpleDateFormat("MM-dd HH:mm");

        startQST = (Button) findViewById(R.id.btn_StartQST);
        startQST.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                goToOtherapp();
            }
        });
        goedkeuren = (Button) findViewById(R.id.btn_goedkeurenPijndrempel);
        goedkeuren.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                addThreshold();
                printDataBase();
            }
        });
        afkeuren= (Button)findViewById(R.id.btn_afkeurenPijndrempel);
        afkeuren.setOnClickListener(new View.OnClickListener() {
```

Appendix F

```
            @Override
            public void onClick(View v) {
                confirmInput.setText(newResult + " is afgekeurt");
            }
        });

        //printDataBase();//method for see the database on the screen
        Runnable r = new Runnable() {
            @Override
            public void run() {
                printDataBase();//method for see the database on the screen
            }
        };

        Thread t = new Thread(r);
        t.start();
    }

    private void printDataBase() {
        final String dbString = dbHandler.databaseToSting();

        //TODO aangezien deze methode vanuit andere threads aangeroepen
wordt, moeten we zorgen dat de UI update
        //TODO wel op de main thread afgehandeld wordt.
        runOnUiThread(new Runnable() {
            @Override
            public void run() {
                printData.setText(dbString);
            }
        });
    }

    private void goToOtherapp() {
        Intent intent = new Intent("HelloWorld_MakeValue");
        intent.putExtra("thresholdRequest", requestThreshold); //TODO Do I
really need to put this extra?

        startActivityForResult(intent, REQUEST_CODE_THRESHOLD);
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
        if (requestCode == REQUEST_CODE_THRESHOLD) {
            if (resultCode == RESULT_OK) {
                c = Calendar.getInstance();
                df = new SimpleDateFormat("MM-dd HH:mm");
                newResult = data.getFloatExtra("ThresholdResult", -1);
                String myString = Float.toString(newResult);
                Log.i("Drempel:", myString);
                //confirmInput.setText("Succesfully added:  " + myString);
                //addThreshold();
                showThreshold();
            }
        }
        super.onActivityResult(requestCode, resultCode, data);
    }

    private void showThreshold() {
        String mystring = Float.toString(newResult);
        receivedThreshold.setText(mystring);
```

Appendix F

```
    }

    private void addThreshold() {
        Runnable r = new Runnable() {
            @Override
            public void run() {
                final Painthresholds painthresholds = new
Painthresholds(newResult);
                painthresholds.setThreshold_date(df.format(c.getTime()));
                dbHandler.addThreshold(painthresholds);
                printDataBase();

                runOnUiThread(new Runnable() {
                    @Override
                    public void run() {
                        //"Reset" edit text field
                        String mysting =
Float.toString(painthresholds.getThreshold_value());
                        confirmInput.setText("Pijndrempel waarde: " +
mysting + " is succesvol toegevoegd!");
                    }
                });
            }
        };

        Thread t = new Thread(r);
        t.start();
    }
}
```

PainscaleInput.java

```
package com.example.shanna.followup;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;

import com.github.mikephil.charting.charts.BarChart;
import com.github.mikephil.charting.data.BarData;
import com.github.mikephil.charting.data.BarDataSet;
import com.github.mikephil.charting.data.BarEntry;

import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.List;

public class PainscaleInput extends AppCompatActivity {

    EditText inputPainscale;
    TextView confirmInput;
    TextView dateTime;
    DBHandler dbHandler;
    Calendar c;
    SimpleDateFormat df;

    @Override
```

Appendix F

```
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_painscale_input);

        c = Calendar.getInstance();
        dbHandler = new DBHandler(this, null, null, 0);

        inputPainscale = (EditText) findViewById(R.id.editText);
        confirmInput = (TextView) findViewById(R.id.confirmation);
        dateTime = (TextView) findViewById(R.id.dateTime);

        df = new SimpleDateFormat("MM-dd HH:mm");
        String formattedDate = df.format(c.getTime());

        dateTime.setText(formattedDate);
    }

    public void addButtonClicked(View view) {
        //runnable made with Rick Gemert
        int painscale =
Integer.parseInt(inputPainscale.getText().toString());
        if (painscale > 10) {
            confirmInput.setText("Vul een waarde tussen de 1 en 10 in");
        } else {
            Runnable r = new Runnable() {
                @Override
                public void run() {
                    //add user input object (object is a row in database)
to the database
                    final Painscales painscales = new
Painscales(Integer.parseInt(inputPainscale.getText().toString()));
                    painscales.set_date(df.format(c.getTime()));
                    dbHandler.addPainscale(painscales);

                    runOnUiThread(new Runnable() {
                        @Override
                        public void run() {
                            //"Reset" edit text field
                            inputPainscale.setText("");
                            String mysting =
Integer.toString(painscales.get_value());
                            confirmInput.setText("Pijnschaal waarde " +
mysting + ", succesvol toegevoegd!");
                            finish();
                        }
                    });
                }
            };

            Thread t = new Thread(r);
            t.start();
            //whenever user clicks the add button, take the input, add it
to the database and print it on the screen;
        }
    }
}
```

Resultaten_schaal_drempel.java

```
package com.example.shanna.followup;
```

Appendix F

```
import android.content.Intent;
import android.support.design.widget.FloatingActionButton;
import android.support.design.widget.Snackbar;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;

import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentManager;
import android.support.v4.app.FragmentPagerAdapter;
import android.support.v4.view.ViewPager;
import android.os.Bundle;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;

import android.widget.TextView;

import com.github.mikephil.charting.charts.BarChart;
import com.github.mikephil.charting.data.BarData;
import com.github.mikephil.charting.data.BarDataSet;
import com.github.mikephil.charting.data.BarEntry;

import java.util.ArrayList;
import java.util.List;

//TODO email ding weghalen, of vervangen door nocitrack logo

public class Resultaten_schaal_drempel extends AppCompatActivity {

    /**
     * The {@link android.support.v4.view.PagerAdapter} that will provide
     * fragments for each of the sections. We use a
     * {@link FragmentPagerAdapter} derivative, which will keep every
     * loaded fragment in memory. If this becomes too memory intensive, it
     * may be best to switch to a
     * {@link android.support.v4.app.FragmentStatePagerAdapter}.
     */
    private SectionsPagerAdapter mSectionsPagerAdapter;

    /**
     * The {@link ViewPager} that will host the section contents.
     */
    private ViewPager mViewPager;

    //variables for content main2
    Intent data;
    TextView patientName;
    TextView doctorName;
    public String userName;
    public String patientNumber;


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_resultaten_schaal_drempel);

        // Create the adapter that will return a fragment for each of the
```

Appendix F

```
three
        // primary sections of the activity.
        mSectionsPagerAdapter = new
SectionsPagerAdapter(getSupportFragmentManager());

        // Set up the ViewPager with the sections adapter.
        mViewPager = (ViewPager) findViewById(R.id.container);
        mViewPager.setAdapter(mSectionsPagerAdapter);

        data = getIntent();
        patientName = (TextView) findViewById(R.id.bar_patientName);
        doctorName = (TextView) findViewById(R.id.bar_doctorName);
        setUserBar();
    }

    private void setUserBar() {
        //content main2
        userName = data.getStringExtra("userName");
        patientNumber = data.getStringExtra("patientName");
        patientName.setText(patientNumber);
        doctorName.setText(userName);
    }


    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is
present.
        getMenuInflater().inflate(R.menu.menu_resultaten_schaal_drempel,
menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so long
        // as you specify a parent activity in AndroidManifest.xml.
        int id = item.getItemId();

        //noinspection SimplifiableIfStatement
        if (id == R.id.action_settings) {
            return true;
        }

        return super.onOptionsItemSelected(item);
    }

    /**
     * A {@link FragmentPagerAdapter} that returns a fragment corresponding
to
     * one of the sections/tabs/pages.
     */
    public class SectionsPagerAdapter extends FragmentPagerAdapter {

        public SectionsPagerAdapter(FragmentManager fm) {
            super(fm);
        }

        //@Override
        public Fragment getItem(int position) {
```

Appendix F

```
            // getItem is called to instantiate the fragment for the given
page.
            // Return a PlaceholderFragment (defined as a static inner
class below).
        //    return PlaceholderFragment.newInstance(position + 1);
        //}

        //@Override


        //@Override
        //public CharSequence getPageTitle(int position) {
            switch (position) {
                case 0:
                    return PlaceholderFragment.newInstance(position + 1);
                case 1:
                    return PijndrempelFragment.newInstance(position);
            }
            return null;
        }

        public int getCount() {
            // Show 2 total pages.
            return 2;
        }
    }
}
```

PlaceholderFragment.java

```
package com.example.shanna.followup;

import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import com.github.mikephil.charting.charts.BarChart;
import com.github.mikephil.charting.data.BarData;
import com.github.mikephil.charting.data.BarDataSet;
import com.github.mikephil.charting.data.BarEntry;

import java.util.ArrayList;
import java.util.List;

/**
 * Created by Shanna on 21-6-2016.
 */

/**
 * A placeholder fragment containing a simple view.
 */

    //TODO layout afmaken
    //TODO die "Hello World section 1/2 weghalen
    //TODO titel goedzetten

public class PlaceholderFragment extends Fragment {
```

Appendix F

```
    /**
     * The fragment argument representing the section number for this
     * fragment.
     */
    private static final String ARG_SECTION_NUMBER = "section_number";

    public PlaceholderFragment() {
    }

    /**
     * Returns a new instance of this fragment for the given section
     * number.
     */
    public static PlaceholderFragment newInstance(int sectionNumber) {
        PlaceholderFragment fragment = new PlaceholderFragment();
        Bundle args = new Bundle();
        args.putInt(ARG_SECTION_NUMBER, sectionNumber);
        fragment.setArguments(args);
        return fragment;
    }

    BarChart pijnschaalBarChart;
    DBHandler dbHandler;

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        View rootView =
inflater.inflate(R.layout.fragment_resultaten_pijnschaal, container,
false);

        dbHandler = new DBHandler(getContext(), null, null, 0);
        pijnschaalBarChart = (BarChart)
rootView.findViewById(R.id.pijnschaalBarChart);
        getData();

        return rootView;
    }

    private void getData() {

        ArrayList<BarEntry> yVals = new ArrayList<BarEntry>();
        //logdata
        String mySting = yVals.toString();
        String dbString =
Integer.toString(dbHandler.pijnschaalYData().size());
        Log.i("AddData", mySting); //deze print hij wel, maar die in de for
loop niet. Size is leeg
        Log.i("dbObject", dbString); //waarschijnlijk kent bij db.handler
al niet?
        List<Float> Yquery = dbHandler.pijnschaalYData();
        List<String> Xquery = dbHandler.pijnschaalXData();

        for (int i = 0; i < Yquery.size(); i++) {
            //logdata
            String iString = Integer.toString(i);
            String dbObject = Yquery.get(i).toString();
            Log.i("AddData", iString);
            Log.i("object", dbObject);

            yVals.add(new BarEntry(Yquery.get(i), i));
```

Appendix F

```
            //logdata
            String myVals = yVals.toString();
            Log.i("TAG", myVals);
        }

        ArrayList<String> xVals = new ArrayList<String>();
        for (int i = 0; i < Xquery.size(); i++)
            xVals.add(Xquery.get(i));

        BarDataSet dataSet = new BarDataSet(yVals, "Pijnschaal");
        final BarData data = new BarData(xVals, dataSet);

        pijnschaalBarChart.setData(data);
        pijnschaalBarChart.invalidate();
    }
}
```

PijndrempelFragment.java

```
package com.example.shanna.followup;

import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import com.github.mikephil.charting.charts.LineChart;
import com.github.mikephil.charting.data.Entry;
import com.github.mikephil.charting.data.LineData;
import com.github.mikephil.charting.data.LineDataSet;
import java.util.ArrayList;
import java.util.List;

public class PijndrempelFragment extends Fragment {
    /**
     * The fragment argument representing the section number for this
     * fragment.
     */
    private static final String ARG_SECTION_NUMBER = "section_number";

    public PijndrempelFragment() {
    }

    /**
     * Returns a new instance of this fragment for the given section
     * number.
     */
    public static PijndrempelFragment newInstance(int sectionNumber) {
        PijndrempelFragment fragment = new PijndrempelFragment();
        Bundle args = new Bundle();
        args.putInt(ARG_SECTION_NUMBER, sectionNumber);
        fragment.setArguments(args);
        return fragment;
    }

    LineChart pijndrempelLineChart;
    DBHandler dbHandler;

    @Override
```

Appendix F

```
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        View rootView =
inflater.inflate(R.layout.fragment_resultaten_pijnsdrempel, container,
false);

        dbHandler = new DBHandler(getContext(), null, null, 0);
        pijndrempelLineChart = (LineChart)
rootView.findViewById(R.id.pijndrempelLineChart);
        getData();

        return rootView;
    }

    private void getData() {
        ArrayList<Entry> yVals = new ArrayList<Entry>();
        //logdata
        String mySting = yVals.toString();
        String dbString =
Float.toString(dbHandler.thresholdYData().size());
        Log.i("AddData2", mySting);
        Log.i("dbObject2", dbString);

        List<Float> Yquery = dbHandler.thresholdYData();
        List<String> Xquery = dbHandler.thresholdXData();
        Log.i("size", String.valueOf(Yquery.size()));

        for (int i = 0; i < Yquery.size(); i++) {
            //logdata
            String iString = Float.toString(i);
            String dbObject = Yquery.get(i).toString();
            Log.i("AddData", iString);
            Log.i("object", dbObject);

            yVals.add(new Entry(Yquery.get(i), i));

            //logdata
            String myVals = yVals.toString();
            Log.i("TAG1", myVals);
        }

        ArrayList<String> xVals = new ArrayList<String>();
        for (int i = 0; i < Xquery.size(); i++) {
            xVals.add(Xquery.get(i));
            String myVals = Xquery.get(i);
            Log.i("TAG2", myVals);
        }

        LineDataSet dataSet = new LineDataSet(yVals, "Pijn drempel");
        dataSet.setDrawFilled(true);
        final LineData data = new LineData(xVals, dataSet);
        pijndrempelLineChart.setData(data);
        pijndrempelLineChart.invalidate();
        }
}
```

Patient.java

```
package com.example.shanna.followup;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
```

Appendix F

```
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

public class Patient extends AppCompatActivity {
    //variables for content_main2
    Intent data;
    Intent backToScanner;
    Button patientLogin;
    TextView patientName;
    TextView doctorName;
    TextView patient;
    public String userName;
    public String patientNumber;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_patient);

        data = getIntent();
        patientNumber = data.getStringExtra("patientName");
        userName = data.getStringExtra("userName");
        patientName = (TextView) findViewById(R.id.bar_patientName);
        doctorName = (TextView) findViewById(R.id.bar_doctorName);
        setUserBar();

        patient = (TextView)findViewById(R.id.dePatientTitle);
        patient.setText(patientNumber);
        backToScanner = new Intent(this, ScanBarcode.class);
        patientLogin = (Button)findViewById(R.id.loginNewPatient);
        patientLogin.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                backToScanner();
            }
        });
    }

    private void backToScanner() {
        patientNumber = null;
        backToScanner.putExtra("userName", userName);
        startActivity(backToScanner);
    }

    private void setUserBar() {
        //content main2;
        patientName.setText(patientNumber);
        doctorName.setText(userName);
    }
}
```

Database classes:

DBHandler.java

```
package com.example.shanna.followup;

/**
 * Created by Shanna on 21-6-2016.
```

Appendix F

```
 */

import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.database.Cursor;
import android.content.Context;
import android.content.ContentValues;
import android.util.Log;
import java.util.ArrayList;

public class DBHandler extends SQLiteOpenHelper{

    private static final int DATABASE_VERSION = 1; //this is the first
datbase, so first version

    private static final String DATABASE_NAME = "Patient_nummer.db";

    //name of table and columns
    public static final String TABLE_NAME_PIJNSCHAAL= "Pijnschaal";
    public static final String TABLE_PAINTHRESHOLD = "Painthreshold";


    //field of TABLE_NAME_PIJNSCHAAL
    public static final String PIJNSCHAAL_ID = "_id";
    public static final String PIJNSCHAAL_VALUE = "value";
    public static final String PIJNSCHAAL_DATE  = "date";

    //fields of TABLE_PAINTHRESHOLD;
    public static final String PAINTHRESHOLD_ID = "_id";
    public static final String PAINTHRESHOLD_VALUE ="threshold_value";
    public static final String PAINTHRESHOLD_DATE  = "threshold_date";

    // a constructor, to set path to database file and background
information
    public DBHandler(Context context, String name,
SQLiteDatabase.CursorFactory factory, int version) {
        super(context, DATABASE_NAME, factory, DATABASE_VERSION);
    }


    @Override
    public void onCreate(SQLiteDatabase db) {
        //what to do with the database when activity is opened, which is
creating a table
        //format rows of Pijnschaal data table
        String scaleQuery = "CREATE TABLE " + TABLE_NAME_PIJNSCHAAL + "( "
+
                PIJNSCHAAL_ID + " INTEGER PRIMARY KEY AUTOINCREMENT," +
                PIJNSCHAAL_VALUE + " INTEGER, " +
                PIJNSCHAAL_DATE + " TEXT " +
                ");";
        db.execSQL(scaleQuery);

        //format rows of Pijndrempel data table
        String thresholdQuery = "CREATE TABLE " + TABLE_PAINTHRESHOLD + "(
" +
                PAINTHRESHOLD_ID + " INTEGER PRIMARY KEY AUTOINCREMENT," +
                PAINTHRESHOLD_VALUE + " FLOAT, " +
                PAINTHRESHOLD_DATE + " TEXT " +
                ");";
        db.execSQL(thresholdQuery);
```

Appendix F

```
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int
newVersion) {
        //what to do if the database is upgraded, which is deleting the old
table and make a new one
        db.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME_PIJNSCHAAL);
        onCreate(db); //create the new updated table
    }

    //adding new row to the databse
    public void addPainscale(Painscales painscales){
        //pass an object into the painscals table
        ContentValues values = new ContentValues();//making list of values
        values.put(PIJNSCHAAL_VALUE, painscales.get_value()); //put item
value of the painscale in the column
        values.put(PIJNSCHAAL_DATE, painscales.get_date());
        SQLiteDatabase db = getWritableDatabase();//key to the databse
        db.insert(TABLE_NAME_PIJNSCHAAL, null, values);
        db.close(); //end of the row
    }

    public void addThreshold(Painthresholds painthresholds){
        //pass an object into the threshold table
        ContentValues values = new ContentValues();//making list of values
        values.put(PAINTHRESHOLD_VALUE,
painthresholds.getThreshold_value()); //put item value of the painscale in
the column
        values.put(PAINTHRESHOLD_DATE, painthresholds.getThreshold_date());
        SQLiteDatabase db = getWritableDatabase();//key to the databse
        db.insert(TABLE_PAINTHRESHOLD, null, values);
        db.close(); //end of the row
    }

    //set data for pain scale graph
    //set values for the x axe to date and time
    public ArrayList<String> pijnschaalXData(){
        ArrayList<String> xNewData = new ArrayList<String>();
        SQLiteDatabase db = getWritableDatabase();
        String query = "SELECT " + PIJNSCHAAL_DATE + " FROM " +
TABLE_NAME_PIJNSCHAAL;
        Cursor cursor = db.rawQuery(query, null);
        int colidx = cursor.getColumnIndex(PIJNSCHAAL_DATE );
        Log.i("bla", colidx+"");
        for (cursor.moveToFirst();
!cursor.isAfterLast();cursor.moveToNext()){
            String xval = cursor.getString(colidx);
            xNewData.add(xval);
        }

        cursor.close();
        return xNewData;
    }

    public ArrayList<Float> pijnschaalYData(){
        //set values for the y axe to the value of pijnschaal
        ArrayList<Float> yNewData = new ArrayList<Float>();
        String mySting = yNewData.toString();
        SQLiteDatabase db = getWritableDatabase();
        String query = "SELECT " + PIJNSCHAAL_VALUE + " FROM " +
```

Appendix F

```
TABLE_NAME_PIJNSCHAAL;
        Cursor cursor = db.rawQuery(query, null);
        for (cursor.moveToFirst();
!cursor.isAfterLast();cursor.moveToNext()){
            Float yval = cursor.getFloat(cursor.getColumnIndex("value"));
            yNewData.add(yval);
        }
        cursor.close();
        return yNewData;
    }

    //set data for pain threshold graph
    //set values for the x axe to date and time
    public ArrayList<String> thresholdXData(){
        ArrayList<String> xNewData = new ArrayList<String>();
        SQLiteDatabase db = getWritableDatabase();
        String query = "SELECT " + PAINTHRESHOLD_DATE + " FROM " +
TABLE_PAINTHRESHOLD;
        Cursor cursor = db.rawQuery(query, null);
        int colidx = cursor.getColumnIndex(PAINTHRESHOLD_DATE);
        Log.i("bla2", colidx+"");
        for (cursor.moveToFirst();
!cursor.isAfterLast();cursor.moveToNext()){
            String xval = cursor.getString(colidx);
            xNewData.add(xval);
        }

        cursor.close();
        return xNewData;
    }
    //set value for y axe to painthreshold value
    public ArrayList<Float> thresholdYData(){
        Log.i("queryData2", "running" ); //can't see this in logcat. Does
this mean it empty? Or can log not be used in a public float?
        ArrayList<Float> yNewData = new ArrayList<Float>();
        String mySting = yNewData.toString();
        //Log.i("FloatArray", "running" );
        SQLiteDatabase db = getWritableDatabase();
        String query = "SELECT " + PAINTHRESHOLD_VALUE + " FROM " +
TABLE_PAINTHRESHOLD;
        Cursor cursor = db.rawQuery(query, null);
        for (cursor.moveToFirst();
!cursor.isAfterLast();cursor.moveToNext()){
            Float yval =
cursor.getFloat(cursor.getColumnIndex(PAINTHRESHOLD_VALUE));
            yNewData.add(yval);
        }
        cursor.close();
        return yNewData;
    }

    //print database from threshold as a string
    public String databaseToSting(){
        String dbString = "";
        SQLiteDatabase db = getWritableDatabase(); //key to database
        String query = "SELECT * FROM " + TABLE_PAINTHRESHOLD + " WHERE 1
";

        //cursor paints to location in your results
        Cursor c = db.rawQuery(query, null);
        c.moveToFirst();
```

Appendix F

```
        while (c.moveToNext()){ //loops through database in order to print
everything
            if(c.getString(c.getColumnIndex("threshold_value"))!= null){
                dbString +=
c.getString(c.getColumnIndex("threshold_value")); //retrieve item
                dbString += " ";
                dbString +=
c.getString(c.getColumnIndex("threshold_date"));
                dbString += "\n"; //hop to new line
            }
        }

        c.close();
        close();
        return dbString;
    }
}
```

Painscales.java

```
package com.example.shanna.followup;

/**
 * Created by Shanna on 21-6-2016.
 */
public class Painscales {
    //each row is a java object
    //we have different collumns of differerent properties (here ID and
value)
    // the object/row need this properties

    private int _id ; //propperies
    private int _value;
    private String _date;

    //empty constructor, in case you want to make an object, but add the
properties later on
    public Painscales(){
    }

    //constructor to allow user to type the value, and it will have this
value an id
    public Painscales(int value){
        this._value = value;
    }

    //table needs to work with the object Painscale

    public void set_id(int _id) {
        this._id = _id;
    }

    public void set_value(int _value) {
        this._value = _value;
    }

    public void set_date(String _date){this._date = _date;}

    //getter to retrieve (print out value for example)
    public int get_id() {
        return _id;
```

Appendix F

```
    }

    public int get_value() {
        return _value;
    }

    public String get_date(){return _date;}
}
```

Painthresholds.java

```
package com.example.shanna.followup;

/**
 * Created by Shanna on 22-6-2016.
 */
public class Painthresholds {
    //each row is a java object
    //we have different collumns of differerent properties (here ID and
value)
    // the object/row need this properties

    private int _id ; //propperies
    private float threshold_value;
    private String threshold_date;

    //empty constructor, in case you want to make an object, but add the
properties later on
    public Painthresholds(){
    }

    //constructor to allow user to type the value, and it will have this
value an id
    public Painthresholds(float value){
        this.threshold_value = value;
    }

    //table needs to work with the object

    public void set_id(int _id) {
        this._id = _id;
    }

    public void setThreshold_value(float threshold_value) {
        this.threshold_value = threshold_value;
    }

    public void setThreshold_date(String
threshold_date){this.threshold_date = threshold_date;}

    //getter to retrieve (print out value for example)
    public int get_id() {
        return _id;
    }

    public float getThreshold_value() {
        return threshold_value;
    }

    public String getThreshold_date(){return threshold_date;}
}
```

Appendix G

Hello World | My first app!

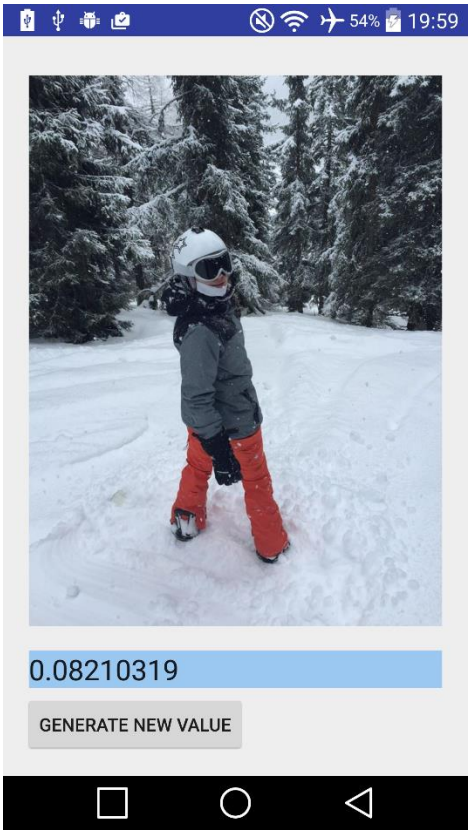


Figure1. First number generator

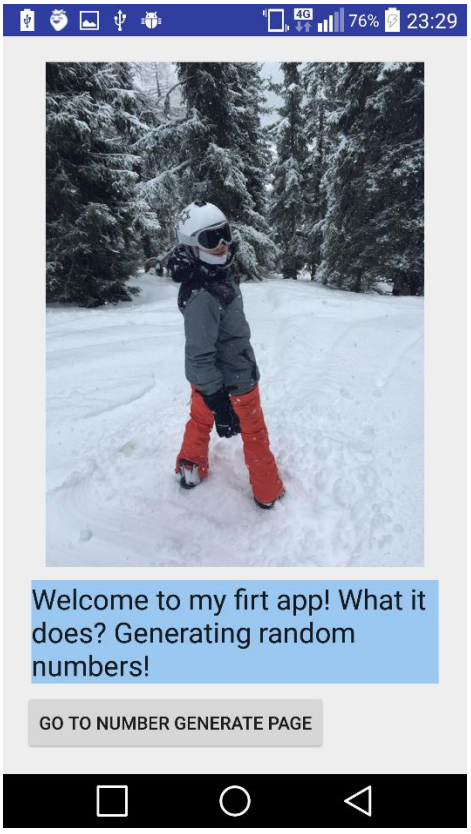


Figure 2. Multiple activities

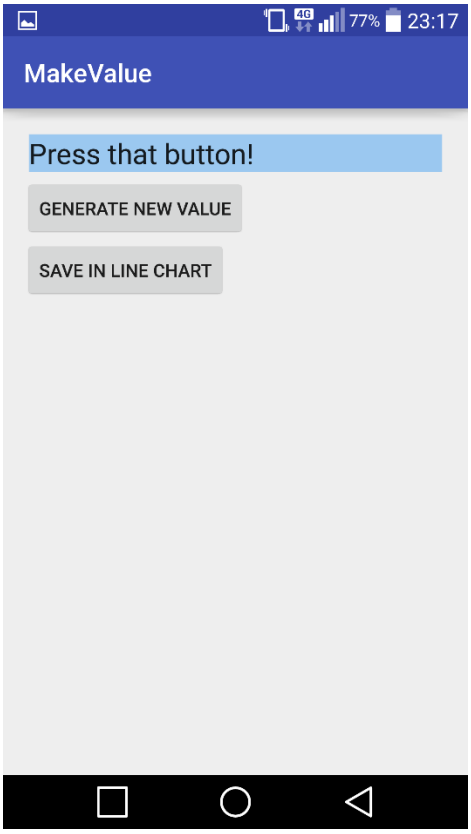


Figure 3. Another activity

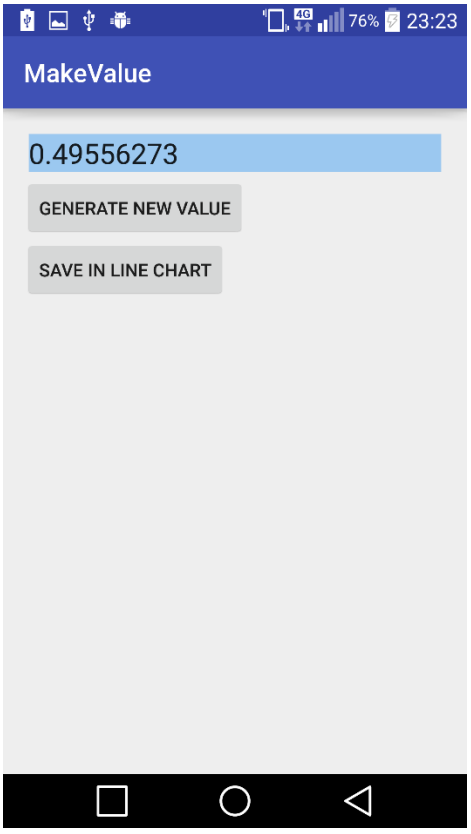


Figure 4. Action of the other activity

Appendix G

Introduction

The first app I made (see figure 1) was an app which displayed a picture and generated a random number.

I added the picture and the text field with a tutorial on Udacity.com, which learned be a little bit about layouts and how to position things.

Because the final app I want to make, will retrieve a value from another app, I wanted to use this app to generate this number so I could retrieve this one. With this, I can already test the functionality of adding retrieved values into a database and displaying these values.

I also wanted to add another activity into it, because my final app will have multiple pages as well (see figure 2 to 4). Next to this, the tutorial I followed was also using two different activities to switch from one app to the other, so why not implement this into these prototypes already.

This document will give code of the app you see in figure 2 to 4. This app will be used in the final prototype to retrieve the random number, add it to a database and display this in graphs.

Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.design.widget.CoordinatorLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fitsSystemWindows="true"
    tools:context="com.example.shanna.helloworld.MainActivity">

    <android.support.design.widget.AppBarLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:theme="@style/AppTheme.AppBarOverlay">

    </android.support.design.widget.AppBarLayout>

    <include layout="@layout/content_main" />

</android.support.design.widget.CoordinatorLayout>
```

Content_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:orientation="vertical"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:context="com.example.shanna.helloworld.MainActivity"
    tools:showIn="@layout/activity_main">

    <ImageView
```

Appendix G

```
        android:contentDescription="@string/app_name"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:id="@+id/image"
        android:src="@drawable/wintersport2"
        android:layout_margin="4dp"/>


    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textAppearance="?android:textAppearanceLarge"
        android:background="#9bc8f0"
        android:text="Welcome to my firt app! What it does? Generating
random numbers!"
        android:id="@+id/Lorem_textView"
        android:layout_margin="6dp"
        />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Go to number generate page"
        android:id="@+id/button"
        android:onClick="MakeValue"/>

</LinearLayout>
```

Appendix G

MainActivity.java

```
package com.example.shanna.helloworld;

import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.support.design.widget.FloatingActionButton;
import android.support.design.widget.Snackbar;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.view.View;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.TextView;

import com.google.android.gms.appindexing.Action;
import com.google.android.gms.appindexing.AppIndex;
import com.google.android.gms.common.api.GoogleApiClient;

import org.w3c.dom.Text;

import java.util.Random;

public class MainActivity extends AppCompatActivity {

    /**
     * ATTENTION: This was auto-generated to implement the App Indexing
API.
     * See https://g.co/AppIndexing/AndroidStudio for more information.
     */
    private GoogleApiClient client;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // ATTENTION: This was auto-generated to implement the App Indexing
API.
        // See https://g.co/AppIndexing/AndroidStudio for more information.
        client = new
GoogleApiClient.Builder(this).addApi(AppIndex.API).build();
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is
present.
        getMenuInflater().inflate(R.menu.menu_main, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so long
        // as you specify a parent activity in AndroidManifest.xml.
        int id = item.getItemId();

        //noinspection SimplifiableIfStatement
```

Appendix G

```
        if (id == R.id.action_settings) {
            return true;
        }

        return super.onOptionsItemSelected(item);
    }

    public void MakeValue(View view) {
        Intent intent1 = new Intent(this,MakeValue.class);
        startActivity(intent1);
    }

    @Override
    public void onStart() {
        super.onStart();

        // ATTENTION: This was auto-generated to implement the App Indexing
API.
        // See https://g.co/AppIndexing/AndroidStudio for more information.
        client.connect();
        Action viewAction = Action.newAction(
                Action.TYPE_VIEW, // TODO: choose an action type.
                "Main Page", // TODO: Define a title for the content shown.
                // TODO: If you have web page content that matches this app
activity's content,
                // make sure this auto-generated web page URL is correct.
                // Otherwise, set the URL to null.
                Uri.parse("http://host/path"),
                // TODO: Make sure this auto-generated app URL is correct.
                Uri.parse("android-
app://com.example.shanna.helloworld/http/host/path")
        );
        AppIndex.AppIndexApi.start(client, viewAction);
    }

    @Override
    public void onStop() {
        super.onStop();

        // ATTENTION: This was auto-generated to implement the App Indexing
API.
        // See https://g.co/AppIndexing/AndroidStudio for more information.
        Action viewAction = Action.newAction(
                Action.TYPE_VIEW, // TODO: choose an action type.
                "Main Page", // TODO: Define a title for the content shown.
                // TODO: If you have web page content that matches this app
activity's content,
                // make sure this auto-generated web page URL is correct.
                // Otherwise, set the URL to null.
                Uri.parse("http://host/path"),
                // TODO: Make sure this auto-generated app URL is correct.
                Uri.parse("android-
app://com.example.shanna.helloworld/http/host/path")
        );
        AppIndex.AppIndexApi.end(client, viewAction);
        client.disconnect();
    }
}
```

Appendix G

activity_make_value.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.shanna.helloworld.MakeValue">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textAppearance="?android:textAppearanceLarge"
        android:background="#9bc8f0"
        android:text="Press that button!"
        android:id="@+id/Lorem_textView"
        android:layout_margin="4dp"
        />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Generate new value"
        android:id="@+id/button"
        android:layout_below="@+id/Lorem_textView"
        android:onClick="generate"/>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/buttonTwo"
        android:layout_below="@id/button"
        android:text="Save in Line Chart"
        android:onClick="giveIntent"/>


</RelativeLayout>
```

Appendix G

MakeValue.java

```
package com.example.shanna.helloworld;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

import java.util.Random;

public class MakeValue extends AppCompatActivity {

    public float number;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_make_value);


    }

    public void generate(View view) {
        Random randNumber = new Random();
        number = randNumber.nextFloat();
        TextView myText = (TextView) findViewById(R.id.Lorem_textView);
        String myString = String.valueOf(number);
        myText.setText(myString);

        //Intent intent = new Intent();
        //intent.putExtra("ThresholdResult", number);
       // setResult(RESULT_OK, intent);
       // finish(); //return to last activity, which can also be the main
screen, which is anointing cause you can'


    }

    public void giveIntent(View view){
        //getIntent().putExtra("requestThreshold", 1); only set result and
finish if there is there is the intent from the other app
        Intent intent = new Intent();
        intent.putExtra("ThresholdResult", number);
        setResult(RESULT_OK, intent);
        finish();
    }

}
```

Appendix H

First Line Chart Generator

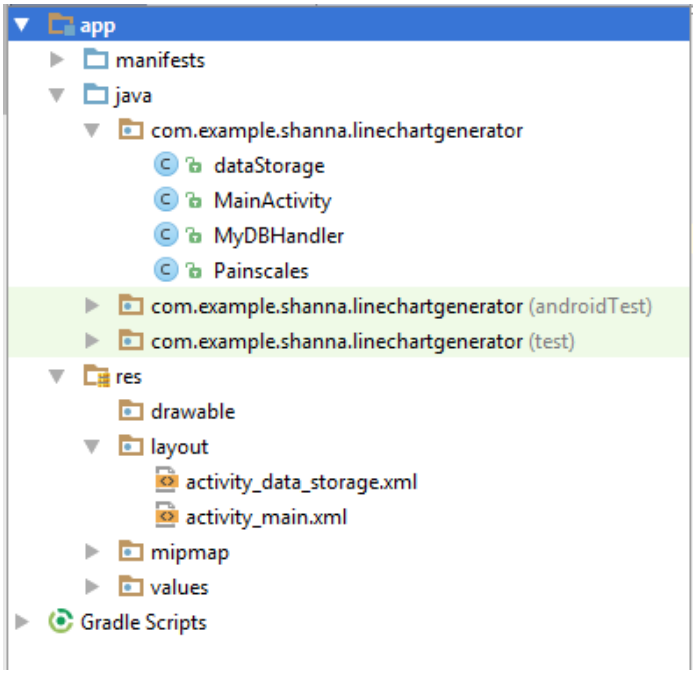


Figure 1. File structure in Android studio

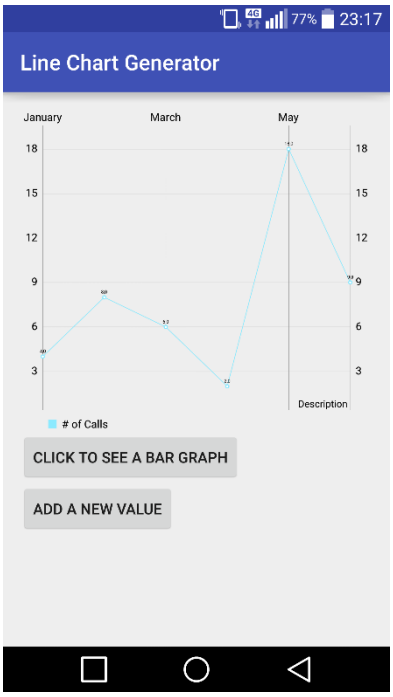


Figure 2. Main Activity

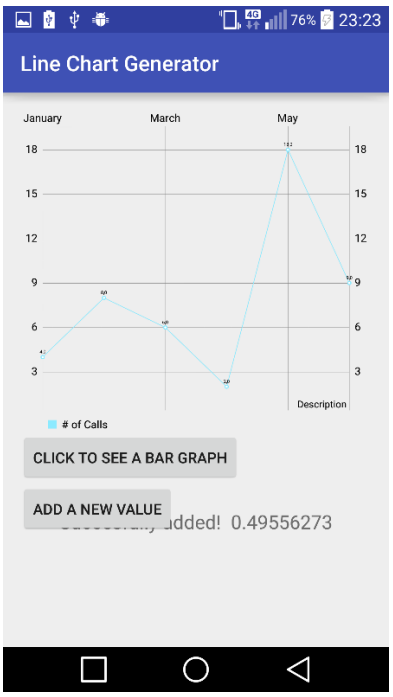


Figure 3. Main Activity, action

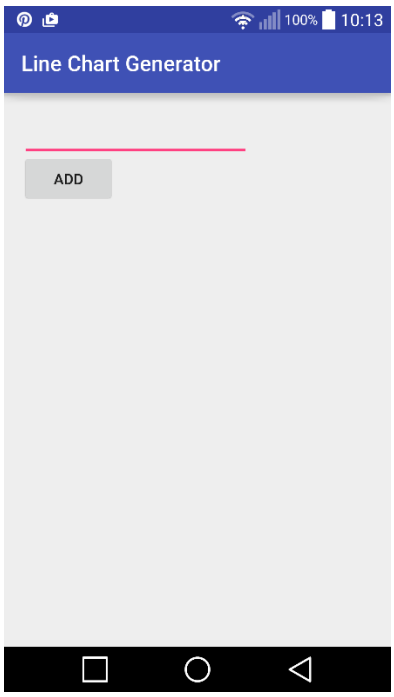


Figure 4. Activity One, data storage

Appendix H

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.shanna.linechartgenerator.MainActivity">

    <com.github.mikephil.charting.charts.LineChart
        android:id="@+id/chart"
        android:layout_width="match_parent"
        android:layout_height="300dp" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Click to see a bar graph"
        android:id="@+id/ButtonActivityOne"
        android:layout_below="@+id/chart"
        android:onClick="goToActivityOne"/>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Add a new value"
        android:id="@+id/ButtonOtherApp"
        android:layout_below="@+id/ButtonActivityOne"
        android:onClick="goToOtherApp"/>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/succes"
        android:text=" "
        android:textAppearance="@style/TextAppearance.AppCompat.Medium"
        android:layout_alignBottom="@+id/ButtonOtherApp"
        android:layout_centerHorizontal="true" />

</RelativeLayout>
```

Appendix H

MainActivity.java

```
package com.example.shanna.linechartgenerator;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;

import com.github.mikephil.charting.charts.LineChart;
import com.github.mikephil.charting.data.Entry;
import com.github.mikephil.charting.data.LineData;
import com.github.mikephil.charting.data.LineDataSet;

import java.util.ArrayList;

public class MainActivity extends AppCompatActivity {

    public static final int REQUEST_CODE_THRESHOLD = 1;
    LineChart lineChart;
    float requestThreshold;
    TextView succesLabel;
    public float newResult;

    @Override
    protected void onCreate(Bundle savedInstanceState) {


        setContentView(R.layout.activity_main);
        lineChart = (LineChart) findViewById(R.id.chart);

        super.onCreate(savedInstanceState);

        // creating list of entry
        ArrayList<Entry> entries = new ArrayList<>();
        entries.add(new Entry(4f, 0));
        entries.add(new Entry(8f, 1));
        entries.add(new Entry(6f, 2));
        entries.add(new Entry(2f, 3));
        entries.add(new Entry(18f, 4));
        entries.add(new Entry(9f, 5));

        LineDataSet dataset = new LineDataSet(entries, "# of Calls");

        // creating labels
        ArrayList<String> labels = new ArrayList<String>();
        labels.add("January");
        labels.add("February");
        labels.add("March");
        labels.add("April");
        labels.add("May");
        labels.add("June");

        LineData data = new LineData(labels, dataset);
        lineChart.setData(data); // set the data and list of lables into
chart
    }

    public void goToActivityOne(View view){
```

Appendix H

```
        Intent intent = new Intent(this,dataStorage.class);
        startActivity(intent); //go to other activity in app
    }

    public void goToOtherApp(View view){
        Intent intent2 = new Intent("HelloWorld_MakeValue");
        intent2.putExtra("thresholdRequest", requestThreshold);

        //startActivity(intent2); //To go to activity in the other app
        startActivityForResult(intent2,   REQUEST_CODE_THRESHOLD);
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
        if(requestCode==REQUEST_CODE_THRESHOLD){
            if(resultCode==RESULT_OK){
                succesLabel = (TextView) findViewById(R.id.succes);
                newResult = data.getFloatExtra("ThresholdResult", -1);
                succesLabel.setText("Succesfully added!  "
                        + newResult
                );
            }
        }
        super.onActivityResult(requestCode, resultCode, data);
    }
}
```

Appendix H

activity_data_storage.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.shanna.linechartgenerator.dataStorage">

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Add"
        android:onClick="addButtonClicked"
        android:id="@+id/addButton"
        android:layout_marginTop="40dp"
        />

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:inputType="number"
        android:ems="10"
        android:id="@+id/editText"
        android:layout_alignParentTop="true"
        android:layout_alignParentStart="true" />

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/printDatabase"
        android:layout_below="@id/addButton"
        android:layout_margin="32dp"
        android:text=""
        />

</RelativeLayout>
```

Appendix H

dataStorage.java

```
package com.example.shanna.linechartgenerator;

import android.database.sqlite.SQLiteDatabase;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;

public class dataStorage extends AppCompatActivity {

    EditText inputPainscale;
    TextView viewDatabase;
    MyDBHandler dbHandler;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_data_storage);

        inputPainscale = (EditText) findViewById(R.id.editText);
        viewDatabase = (TextView) findViewById(R.id.printDatabase);

        //set db handler objec
        dbHandler = new MyDBHandler(this,null, null, 1); //has 4 parameters
in the constructor in MyDBHandler.class

        printDataBase();//method for see the database on the screen

    }

    //for clickevent: update
    public void addButtonClicked(View view){
        Painscales painscales = new
Painscales(Integer.parseInt(inputPainscale.getText().toString()));
        dbHandler.addPainscale(painscales);
        printDataBase();
        //whenever user clicks the add button, take the input, add it to
the database and print it on the screen;
    }

    public void printDataBase(){
        String dbString = dbHandler.databaseToSting();
        viewDatabase.setText(dbString);
        //inputPainscale.seText(""); //refresh input
    }

}
```

Error in dataStorage.java: help by Rick van Gemert

App crashed when a data item was being added. After, the UI of the data storage activity was black.

```
//set db handler objec
dbHandler = new MyDBHandler(this, null, null, 1); //has 4 parameters in the
constructor in MyDBHandler.class

//TODO database werk kan lang duren, vandaar dat dit op een aparte thread
moet gebeuren zodat de UI niet vast loopt
```

Appendix H

```
public void addButtonClicked(View view){
    //TODO zie bovenstaande comment
    Runnable r = new Runnable() {
        @Override
        public void run() {
            Painscales painscales = new
Painscales(Integer.parseInt(inputPainscale.getText().toString()));
            dbHandler.addPainscale(painscales);
            printDataBase();

            runOnUiThread(new Runnable() {
                @Override
                public void run() {
                    //Reset de waarde in het invoerveld
                    inputPainscale.setText("");
                }
            });
        }
    };

public void printDataBase(){
        final String dbString = dbHandler.databaseToSting();

        //TODO aangezien deze methode vanuit andere threads aangeroepen
wordt, moeten we zorgen dat de UI update
        //TODO wel op de main thread afgehandeld wordt.
        runOnUiThread(new Runnable() {
            @Override
            public void run() {
                viewDatabase.setText(dbString);
            }
        });
//        Log.d(TAG, "printDataBase: " + dbString);
        //inputPainscale.seText(""); //refresh input
    }

Runnable r = new Runnable() {
    @Override
    public void run() {
        printDataBase();//method for see the database on the screen
    }
};

Thread t = new Thread(r);
t.start();
```

Appendix H

MyDBHandler.java

```
package com.example.shanna.linechartgenerator;

/**
 * Created by Shanna on 1-6-2016.
 * class to work with the database
 * with the help of the youtube tutorial, "Android App Development for
Beginners - 51, 52 & 53 " from thenewboston
 */
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.database.Cursor;
import android.content.Context;
import android.content.ContentValues;

import java.security.PublicKey;

//class for working with the database: e.g adding data, upgrading database,
etc.
public class MyDBHandler extends SQLiteOpenHelper{
    //variable to name the table and data an to name the columns

    private static final int DATABASE_VERSION = 1; //this is the first
datbase, so first version

    private static final String DATABASE_NAME ="Pijnschaal_DataFile"; //we
will store data on a file on the device. This gives this file a name

    //name of table and columns
    public static final String TABLE_NAME= "Pijnschaal";
    public static final String COLUMN_ID = "_id";
    public static final String COLUMN_VALUE = "value";

    //to use this on yout phone you need:
    // a constructor, to set path to database file and background
information
    public MyDBHandler(Context context, String name,
SQLiteDatabase.CursorFactory factory, int version) {
        super(context, DATABASE_NAME, factory, DATABASE_VERSION);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        //what to do with the database when activity is opened, which is
creating a table
        String query = "CREATE TABLE " + TABLE_NAME + "( " +
                COLUMN_ID + " INTEGER PRIMARY KEY AUTOINCREMENT," +
                COLUMN_VALUE + " INTEGER " +
                ");";
        db.execSQL(query);
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int
newVersion) {
        //what to do if the database is upgraded, which is deleting the old
table and make a new one
        db.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME);
        onCreate(db); //create the new updated table
```

Appendix H

```
    }

    //adding new row to the databse
    public void addPainscale(Painscales painscales){
        //pass an object into the database
        ContentValues values = new ContentValues();//making list of values
        values.put(COLUMN_VALUE, painscales.get_value()); //put item value
of the painscale in the column
        SQLiteDatabase db = getWritableDatabase();//key to the databse
        db.insert(TABLE_NAME, null, values);
        db.close(); //end of the row
    }

    //print database as a string
    public String databaseToSting(){
        String dbString = "";
        SQLiteDatabase db = getWritableDatabase(); //key to database
        String query = "SELECT * FROM " + TABLE_NAME + " WHERE 1 ";

        //cursor paints to location in your results
        Cursor c = db.rawQuery(query, null);
        c.moveToFirst();

        while (!c.isAfterLast()){ //loops through database in order to
print everything
            if(c.getString(c.getColumnIndex("value"))!= null){
                dbString += c.getString(c.getColumnIndex("value"));
//retrieve item
                dbString += "\n"; //hop to new line
            }
        }
```

Error in MyDBHandler.java, help by Rick van Gemert

There was an infinite loop for printing the database after adding a value to it.

```
Cursor c = db.rawQuery(query, null);
c.moveToFirst();

//TODO deze moveToNext zorgt er voor dat alle waardes in de cursor
doorlopen worden
//TODO isAfterLast() die je eerst gebruikte gaf een eindeloze loop omdat
hij dus steeds bij
//TODO de eerste entry bleef.
while (c.moveToNext()){ //loops through database in order to print
everything
    if(c.getString(c.getColumnIndex("value"))!= null){
        dbString += c.getString(c.getColumnIndex("value")); //retrieve item
        dbString += "\n"; //hop to new line
    }
```

Appendix H

Painscales.java

```
package com.example.shanna.linechartgenerator;

/**
 * Created by Shanna on 1-6-2016.
 * class to deal with the user input
 * with the help of the youtube tutorial, "Android App Development for
Beginners - 49 and 50, from thenewboston
 */
public class Painscales {
    //each row is a java object
    //we have different collumns of differerent properties (here ID and
value)
    // the object/row need this properties

    private int _id ; //propperies
    private int _value;

    //empty constructor, in case you want to make an object, but add the
value later on
    public Painscales(){
    }

//constructor to allow user to type the value, and it will have this value
an id
    public Painscales(int value){
        this._value = value;
    }

 //table needs to work with the object Painscale

    public void set_id(int _id) {
        this._id = _id;
    }

    public void set_value(int _value) {
        this._value = _value;
    }

    //getter to retrieve (print out value for example)
    public int get_id() {
        return _id;
    }

    public int get_value() {
        return _value;
    }
}
```

Appendix I

Final prototype | User evaluation

User test with experts and end-users

The target group

This user test will be conducted by hospital staff or employees of the Bio Signals and Systems research group of the University of Twente. The University staff needs to have some background knowledge about NociTRACK and the Ambustim before participating.

Setup

The participant will enter and be asked to sign the informed consent form. If the participant is willing to participant, the observant will inform the participant about the goal of the user test. The observer will explain that he/she will be testing an app which should applied in the anaesthesiology and especially pain outpatient clinics. The participants will be ask if they are familiar with the following:

Quantitative Sentor Testing as a method in chronic pain screenings

NociTrack and/or the Ambistim and how it performs an eQST

The definition of the medical terms pain threshold and pain scale and the DN4 questionnaire

If the patient if not yet familiar with one of more of these concepts, they will be explained. The following will be used to inform the participant:

Met de NociTrack kan de pijn gevoeligheid worden gemeten door middel van het apparaat en een smartphone of tablet. Deze methode heet 'quantitative sensory tesing', ook wel afgekort QST genoemd. Dit wordt vaak gedaan bij chronische pijn patienten, waarbij er misschien sprake is van zenuwpijn, of neuropathische pijn. Dit houdt in dat de patiënt over-gevoelig is voor pijn.

Het meten van de pijn gevoeligheid gaat als volgt: er worden electroden op de huid aangebracht. Bij het inhouden van de knop op de NociTrack, gaat er een oplopende hoeveelheid stroom doorheen lopen. Dit geeft eerst een tintelend gevoel. Zodra dit gevoel onprettig wordt, moet de patiënt de knop weer los laten. De waarde die het apparaat dan aangeeft in de pijn drempel. Deze waarde voor de pijn drempel wordt alleen nog niet automatisch opgeslagen.

De app die je voor je gaat krijgen, zal deze waarde opslaan en presenteren aan de arts. Daarnaast kunnen er ook andere gegevens die te maken hebben met de pijnklachten in worden opgeslagen. Dit zijn onder andere het gebied waar de pijn zich voordoet, de pijn schaal, dit is een waarde van 1 tot 10 waarin de patient de sterkte van de pijn aangeeft, en de DN4 vragenlijst test aan de hand van de symptomen of het om zenuwpijn gaan.

After, the participants will get the opportunity to ask questions about chronic pain, the Ambustim and the goals of the app. Questions related to the content of the app, will not be answered. When the participant is familiar with all these concepts and has no further questions, the user test can continue without further explanation. The user will be given a user scenario containing 4 different use cases which include 5 tasks in total. Every task is represented with a bullet point. There tasks will be read out loud by the participant. The scenario can be found in the next page. When the last case has been completed, the participant will have to complete the System Usability Scale questionnaire, which can be found at the end of this section. This will be the last part of the user test. Finally, the participant will be thanked for the participation and be given the offered to receive a copy of the results when finished.

Appendix I

User scenario:

Je werkt op de pijnpolikliniek als anesthesioloog. Je ontvangt hier dagelijks mensen met pijn klachten en chronische pijn. Jullie gebruiken hier de Ambustim en de bijbehorende apps voor het verkrijgen van verschillende resultaten, waaronder de pijndrempel en de pijn schaal. De app help om te ontwikkeling van de pijn te volgen.

User case 1: opdoen van informatie

Uw eerst patiënt vandaag is mevrouw Erika Janssen. Ze is hier al een tijdje bij jullie op de kliniek en er zijn al meerdere tests bij haar gedaan met dit systeem. Voordat ze komt wil je alvast even haar haar gegevens bekijken om straks op de hoogte te zijn.

1. Log in, in het system met jouw naam en de inlog gegevens van de patiënt.
2. Navigeer vrij door het systeem. Vind ondertussen de volgende informatie:
    a. Waar zit het pijngebied van Erika?
    b. Welke waarde op de pijnschaal heeft ze bij haar laatste bezoek aangegeven?
    c. Wanneer was de laatste keer geweest dat haar pijndrempel gemeten is?
    d. Wat is de exacte waarde van de pijndrempel toen geweest?

Use case 2: Nieuwe waarde op de pijnschaal

Erika is zojuist door u binnen gelaten. U gaat een gesprek met haar aan over hoe het nu met haar gaat en of ze haar pijn aan kan geven volgens de waarden van de pijnschaal. Ze zegt 7.

3. Voeg deze waarde toe in de app.

Use case 3: Nieuwe QST afnemen

Nu wilt u met haar een QST afnemen door middel van de ambustim. *Deze is nu niet aanwezig, daarom zal er tijdens de "test" een random nummer worden gegeven.*

4. Voor de test 2 keer uit, waarbij de eerste waarde wordt afgewezen en de tweede waarde wel goed is.

Use case 4: Volgende patiënt

Het bezoek loopt tot een einde en dit was alles wat u vandaag voor Erika Janssen kon doen. U laat haar uit en vraagt de volgende patiënt binnen te komen. Dit is meneer Timmerman.

5. Login naar de pagina van meneer Timmerman.

After these 5 tasks, the participant will reflect on their experience with the ap through a questionnaire. This is the SUS questionnaire, which consist of 10 statements which they have to rate from a scale from 1 (strongly disagree) to 5 (strongly agree).

Appendix J

participant	Effectiviteit score (uit 8)	SUS	efficientie score task 1	efficientie score task 2	efficientie score task 3	efficientie score task 4	efficientie score task 5	efficientie is in seconde
1	8	85	59	189	70	143	12	
2	6,5	97,5	22	66	63	59	13	
3	7,5	72,5	22	110	28	64	19	
4	7,5	85	37	126	68	52	9	
Average	7,375	85	35	122,75	57,25	79,5	13,25	
advanced user			17	57	11	32	10	
percentage			205,9	215,8	250	518,2	130	One

task	successes
Login	4
painlocation	4
Last pain score	4
Date of last QST	4
Value of thresh	1
Add pain score	4
Add pain thresh	4
Change patient	4

participant	Login	Gather information	Add pain score	Add pain threshold	Change patient
1	59	189	70	143	12
2	22	66	63	59	13
3	22	110	28	64	19
4	37	126	68	52	9
Average	35	122,75	57,25	79,5	13,25
advanced user	17	57	11	32	10

Appendix J

task	successes (fault in
Login	4
painlocation	4
Last pain scor	4
Date of last Q	4
Value of thres	4
Add pain scor	4
Add pain thres	4
Change patien	4

Appendix J

Tasks

Participant	gender	age	succes t1	time t1	succes t2a-d*	time t2a-2d	succes t3	time t3	succes t4	time t4	succes t5	time t5
1	male		1	59	4	189	1	70	1	143	1	12
2	female		1	22	3,5	66	1	63	1	59	1	13
3	male	21-25	1	22	3,5	110	1	28	1	64	1	19
4	male		1	37	3,5	126	1	68	1	52	1	9

* = fault in graph: 0,5 point for reading correct but answers is wrong because of this mistake

System Usability Score (SUS)

** = for odd numbers a score closer to 5*4 = 20 is desireble, for even numbers a score closer to 1*4= 4

Statement #	statement	p1	p2	p3	p4	Total **
1	I think that	4	5	4	4	17
2	I found the	2	1	2	2	7
3	I thought t	4	5	4	5	18
4	I think that	1	1	2	1	5
5	I found the	5	5	3	5	18
6	I thought t	1	1	3	4	9
7	I would im	4	5	5	5	19
8	I found the	2	1	2	1	6
9	I felt very	4	4	3	4	15
10	I needed t	1	1	1	1	4
score odd		16	19	14	18	
score even		18	20	15	16	
SUS		85	97,5	72,5	85	

Appendix J

succes rate
100%
93,8% / 100%
93,8% / 100%
93,8% / 100%

Appendix K

Ethical reflection on mobile application to support pain specialist in pain screening

Shanna de Lang, Creative Technology | s1470930

PROJECT DESCRIPTION

One out of five adults in the Netherlands is suffering from chronic pain (Kay, 2011). Chronic pain has a negative effect on the personal circumstances of a patient as well as on society as a whole (Raad kwaliteit van Zorg, 2011). How and why this pain occurs, is mostly unclear. According to different researches, an increased sensitivity might be a key factor. Increased sensitivity is caused by a dysfunction in the nerve system: it responds too heavily on incoming pain stimuli. Medication exist to decrease this over-sensitivity, but because of negative side effect, it is undesirable to subscribe it to every patient with chronic pain. A good approach is to diagnose patients with complex pain problematics by multidisciplinary teams (Frantsve, 2007). This team consists of different medical specialists, such as physiotherapists, anaesthesiologist and clinical psychologist. However, these examinations are based on the patient's perspective on its own pain. This is a rather subjective measure.

Measuring pain sensitivity is possible through 'quantitative sensory testing' (QST). Wilder-Smith proves that an extensive QST can be a valuable clinical tool for monitoring and assessing risk for the development of chronic pain (Wilder-Smith, 2011). Implementing QST in clinical practice, provides the opportunity to quantify the pain more objectively. For this method, a patient will be exposed to mechanical or electrical stimuli to stimulate pain. However the QST protocol is common in scientific research, it is not yet implemented in the daily clinical practice. The research group BSS at the University of Twente developed the Ambustim. This is a portable electronic QST system, which uses a tablet or smartphone to conduct the test. This makes the QST protocol more accessible for clinical purposes and to suit the daily practice of measuring and diagnosing chronic pain better. Currently, this value is not saved automatically, but has to be written down manually. This often results in an unorganized document containing numbers.

An android app, for a tablet, has to be developed, which will make the electronical QST protocol of the Ambustim more convenient and easier to use in the daily practice of anaesthesiologists. The results of the QST protocol have to be organized and clearly presented. In the following chapters, there will be elaborated on ethical problems concerning this project.

THE STAKEHOLDERS

Three main stakeholders can be recognised. This are the pain specialists/medical professionals, patients which chronic pain and the hospital. To identify the important stakeholders, an external stakeholder analysis as described by Ten Berge will be applied (Ten Berge, 2004). The stakeholders can be described by five parameters: 1) to what extend they will benefit from the project and therefore be interested in the development, 2) to what extend they are affected by it, 3) to what extend they will be part of the user group, 4) what means they have to resist the implementation of the project, 5) to what extend they process meaningful assets or knowledge to support in the development process. These parameters were determined for every target user and can be seen in table one. The following paragraph will go into this in more detail.

	Pain specialist	patient	hospital
Interest in app	++	+	+/-

Appendix K

Affected by app	++	+	-
Is a target user	++	-	--
Means to resist	+	+	++
Can help & support process	++	-	--

Tabel 1. Stakeholder description

The app will be developed for pain specialist and medical staff related to anaesthesiology. The pain specialist or other healthcare provider, will have the most interest in the app, because it will enhance their practical performance. The app will become a part of the daily clinical routine, therefore the affectedness is high. Implementation of it depends on their opinion and trust in this system. Without this, it is very unlikely that the system will be used in the clinic.

Patient will not directly be affected or interested in the application, because they will not be using the app interface themselves. The healthcare professionals will be dealing with this instead. Indirectly however, the patient will be affected the most: their data will be collected and their diagnoses, treatment or medication will depend on this. Next to this, patients have the right to refuse the usage of this applications to collect and store their personal data (Dutch government: Wet bescherming persoonsgegevens, 2014). This can be a barrier for implementation.

Even though Hospitals will not be affected by the app directly, they will be interested in the app, as it improves their staff's performance. Indirectly, the professional status of the hospital will be affected by the performance of the app. If the app results in better conditions for either the medical staff or the patients, this will have a positive influence on their status. The hospital has the highest position and power status to either implement or refuse the system.

From the stakeholder analysis there can be concluded that the patients and pain specialists/healthcare professionals are the stakeholders which will be affected the most by this application. In the next section, there will be elaborated on ethical issued related to these two stakeholders.

PATIENT PRICAVY

The app will allow medical professionals to save health related data on a mobile device. To make decisions, the performing caregiver, needs to relate this information to an individual patient. Therefore, there has to be dealt with private, personal data, even if the app itself does not contain identifiable data such as name, residence, age or gender. Private data has to be protected by the instance colleting it (Ploem, 1995).

Concerning the privacy of personal information, there are two categories in which this can be defined (Mittelstadt, 2011). First, there is data privacy, regarding the right to access, control or even involve third parties. Second, personal privacy, related to the right to keep information to yourself.

Unauthorised parties or individuals might be interested in access to health data to use them for their own benefits. For example, insurance companies may use it to estimate risks, employers may use it to hire or fire employees (Welch, 2001). Relatives or friends, might be able to harm the patient either physically or socially. They might restrain them in their needs, bully or even abuse them. For example, relatives may use the patient's condition of get medication for their own, personal benefits (Peleg, 2008).

The app has to secure accessibility in terms of authentication, as well as the identification of the patient. Unauthorised parties should not be able to enter the app. Next to this, the patient should not be easily identifiable in the app itself, in case unauthorised parties get in contact with an already unlocked app status.

Appendix K

POWER RELATIONSHIP BETWEEN DOCTOR AND PATIENT

Patient might not feel comfortable with this app or a mobile device being involved in their treatment. However, healthcare professionals are considered to be specialists and therefore they have more knowledge than their patients (Illich, 174). Doctors need this power to fulfil their obligations towards patients, the community and themselves (Goodyear-Smith, 2001). This results in a power inequality between patient and doctors. Therefore, patients might not feel in the position to refuse the usage of this app during their treatment.

According to Dutch and European laws, patients have the right to refuse the collection of their data (Ploem 1995). They also have the right to review their data and request to edit or delete it (Dutch government: Wet bescherming persoongegevens, 2014). The healthcare professional has to inform the patient about their rights, in an understandable manner (Doppegieter, 2004). So before the app will be used for the patient's treatment, the patient should be properly informed.

The accessibility to the patient's personal health data, can also result in a different power inequality. The treatment and medication are based on the results collected and presented through the app. A doctor might abuse the ability to access and add data, to falsify results, to influence the provided treatment and medication. Therefore, the medical professionals should be limited in the ability to change or delete data. So even if data will be changed or deleted, it is still visible in the system.

IMPACT ON THE HEALTHCARE PROFESSIONALS

The app will be part of the daily routine of healthcare professionals, but it will also have influence on the long term. Tiwari discusses the effect of monitoring systems on the job perspective of medical professionals (Tiwari, 2010). Monitoring systems might replace tasks of the daily routine and will probably replace more and more tasks in the future. This may reduce the amount of available jobs. Gerdes claims that ehealth can have a negative influence on the skills of medical professionals (Gerdes, 2008). Autonomous categorization and monitoring, might lead to stunted skill growth or the disability to recognise unmonitored symptoms. However, this app is not a self-monitoring system. It does not work autonomously, so it still requires professional skills and user input. Therefore these issued have a negligible influence on the healthcare professionals.

HEALTHCARE QUALITY

Storing and presenting data in an app can enhance the usability and therefore the daily performance of healthcare professionals. Documenting medical data on electronic media has several advantages over paper, such as reducing problem related to medication prescription, accessibility and storage (Ozair, 2005). Next to this, appointments offer little time to interact with the patient, which makes it difficult to remember the complete situation of patient. This app can provide this overview briefly before the patient comes in, which give the opportunity for more personal contact with patients.

Nevertheless, the app requires interaction of the healthcare professional. This means that parts of the medical consolation, the healthcare professional's attention will be directed towards the screen. Patients might feel there is more interest in their virtual representation, rather than in them personally (Bernat, 2013). Therefore, the interaction time between the app and the performing caregiver and the app, should be minimized.

Appendix K

Systems which are made for specialists, will probably contain terms which are known in the field. On the one hand, this is needed for the system to be respected. Otherwise, the system might be considered unprofessional or for amateurs. On the other hand, it might be dangers to just assume the user is known with terms such as pain scale, threshold, or Quantitative sensor testing (QST). Is these terms are interpreted or used differently, data might be added incorrectly or incorrect diagnoses will be formed. In the treatment and medication is based on incorrect data, this can have dangerous consequences for the patient's health. Therefore, healthcare professionals should be properly informed and educated before they will start to work with the system. Next to this, the app should provide guidance during the process, especially during the first use.

DISCUSSION

In the previous sections, the ethical issues concerning the implementation of this app for patients and healthcare professional has been identified. The following paragraphs will wrap up the previously addressed considerations related to these issues. Additionally, the several design choices to avoid these problems as much as possible, will be discussed.

To protect the patients in terms of data and personal privacy, the app has to secure authentication to the correct parties, as well as secure the identification of the patient. Protecting accessibility by unauthorised parties, can be accomplished through a login system. This can be a login name in combination with a unique password for the medical professionals to access to the system. However, a combining this with an additional physical key would be more secure. Only medical caregivers who are directly involved in the patient's treatment, have access to the electronic health record (EHR) and therefore the patient's personal health data. The physical key can either be something the patient is carrying or be a bar/QR code included in the EHR. This, to prevent all medical staff from using the same login name and password and it will automatically assure access by authorised only. A bar/QR code seems more convenient, because is also available when the patient is not present or forgets the key. By not using identifiable data, such as name or gender, it will not be able to directly identify and therefore relate the data to the patient. However, the patient needs to be identifiable by the medical professional, to use the system correctly. Therefore, the patient's number can be stored in the app instead.

The patients' autonomy has to be guaranteed as well. This will be the task, and legal obligation, of the medical professional to inform the patient about their right. The app itself will not be able to guarantee this, but might be able to support it. For example, a reminded of their right can be added to the patient introduction part in the app.

To prevent medical professional from falsifying patient data. The app can set a timespan for editing or adapting information. Next to this, the data should be saved as log data. This holds that not just results and values will be saved, but also mutations in activities and when they took place. This will ensure that all information is visible and the actions can be verified by others if needed.

Self-monitoring systems raise concern regarding the job perspective and skills of medical professionals in the future. Because this system is not autonomous, so professionals are still responsible in the monitoring and decision making process, there has been concluded that these effects are negligible. However, clear evidence for the impact on skills or job perspective of non-autonomous monitoring and decision making application is missing. More research has to be done to formulate a founded conclusion with respect to these issues.

Implementing an app to support in the daily clinical practice, holds that the doctors attention will be directed towards a screen during part of the patient's consultation. To retain personal contact between patient and medical professional as much as possible, the time that has to be spend on interacting with the system should be minimized. This can be accomplished by direct links to add data and with clear, complete overviews of the patient's status on a minimum amount of pages. Here, a balance has to be found in compactness and clarity.

Appendix K

Next to this, a balance has to be found in the amount of interaction time and quality of usage. Multiple opportunities to review and either verify or falsify the obtained data might be desirable hence they can prevent frequent or likely mistakes.

Quality of usage also has to be assured through properly preparing and educating medical professionals before they will use the system. Even though the system might aims to be self-explanatory, introduction and guidance has to be provided. This can done by providing an additional user manual which explains how to use the app and the complementary hardware which will be needed for the pain screening. Nest to this, the app should contain a guided user introduction, which starts automatically during the first usage but will also be available afterwards.

CONCLUDING REFELCTION

Writing this paper has given various insight in the user perspectives and possible issues. The relationship between patient and doctor will always raise questions regarding the abuse of power and right. Nowadays, the knowledge gap between patients and doctors will be mainly concern medical perspective. The increasing implementation of technological application in healthcare, can create another gap concerning technical knowledge. This problem will probably occur more frequently for elderly or low-educated patients.

It was very useful to think about the possible ways of abuse or to harm the patient. It addresses many issues one may not consider, especially if the field or environment of implementation is unknown. These findings resulted in a constrained framework in which this system has to fit. This provides much clarity during the design process.

The only inconvenience of this ethical reflection was due to the timeframe. While writing this report, the app design has already been finished and the prototype is almost fully realised. Even though preparation with respect to this report were made before, finishing this report did not lead to many new insights. Even if it did, it would already be too late to implement this in the design or prototype. However, it can be included as consideration for future development or improvements.

Appendix K

REFERENCES

Bernat, J. L. (2013). Ethical and quality pitfalls in electronic health records.Neurology, 80(11), 1057-1061.

Doppegieter, R. M. S. (2004). *Arts en patiëntenrechten.* Koninklijke Nederlandsche Maatschappij tot Bevordering der Geneeskunst (KNMG).

Frantsve, L. M. E., & Kerns, R. D. (2007). Patient–provider interactions in the management of chronic pain: current findings within the context of shared medical decision making. *Pain Medicine*, *8*(1), 25-35.

Gerdes, A. (2008). The clash between standardisation and engagement.*Journal of Information, Communication and Ethics in Society*, *6*(1), 46-59.

Goodyear-Smith, F., & Buetow, S. (2001). Power issues in the doctor-patient relationship. *Health Care Analysis*, *9*(4), 449-462.

Illich, I. (1974). Medical nemesis. *The Lancet*, *303*(7863), 918-921.

Kay, M., Santos, J., & Takane, M. (2011). mHealth: New horizons for health through mobile technologies. *World Health Organization*, 66-71.

Mittelstadt, B., Fairweather, N. B., McBride, N., & Shaw, M. (2011). Ethical issues of personal health monitoring: a literature review. In *ETHICOMP 2011 Conference Proceedings. Presented at the ETHICOMP.*

Ozair, F. F., Jamshed, N., Sharma, A., & Aggarwal, P. (2015). Ethical issues in electronic health records: A general overview. Perspectives in clinical research, 6(2), 73.

Peleg, M., Beimel, D., Dori, D., & Denekamp, Y. (2008). Situation-based access control: Privacy management via modeling of patient data access scenarios. *Journal of biomedical informatics*, *41*(6), 1028-1040

Ploem, M. C. (1995). Patiëntenrechten in Europees perspectief. *Tijdschrift voor Gezondheidsrecht*, *19*(8), 138-147.

Raad Kwaliteit van Zorg, (2011). Chronische pijn. *Den Haag: Regieraad. Geraadpleegd op http://www. regieraad. nl/fileadmin/www. regieraad.nl /publiek/Downloads/Regie_op_kwaliteit/Advies_Chronische_pijn. pdf.*

T, P., WARREN, J., DAY, K. J. & MCDONALD, B. 2010. Some non-technologyimplications for wider application of robots assisting older people. Health Care and Informatics Review Online, 14.

Ten Berge, L. J. A. M., Oteman, M. P., & Kooten, J. P. (2004). Externa analyse - Stappenplan Stakeholderanalyse. *Inleiding organisatiekunde.* Coutinho.

Welch, C. A. (2001). Sacred secrets—the privacy of medical records. *New England Journal of Medicine*, *345*(5), 371-372.

Wilder-Smith, O. H. (2011). Chronic pain and surgery: a review of new insights from sensory testing. *Journal of pain & palliative care pharmacotherapy*, *25*(2), 146-159.

Dutch government, ‘’Artikel 1 sub a WBP’’, Autoriteit persoonsgegevens, 2014