NTNU
Norwegian University of
Science and Technology

Implementation of a Mobile Application for Diagnosis of Cerebral Palsy

Christine Østraat Buen

Master of Science in Industrial Cybernetics
Submission date: June 2016
Supervisor: Ole Morten Aamo, ITK

Norwegian University of Science and Technology
Department of Engineering Cybernetics

Preface

This thesis concludes my master's degree in Industrial Cybernetics at Norwegian University of Science and Technology (NTNU) in Trondheim. The thesis was performed during spring semester 2016 at Department of Engineering Cybernetics, in collaboration with a research group at NTNU and St. Olavs Hospital, Trondheim University Hospital.

I chose this project because I think the work they are doing is very important. They are trying to assess the risk of a child developing cerebral palsy, at an earliest possible age. The earlier the child gets the diagnosis, the earlier treatment can begin, which will improve the child's mobility and reduce other symptoms for the rest of their lives.

The objective of this thesis was to develop a prototype for a mobile application. The app is written for Android devices and its main task is to record a video of a child and send it to St. Olavs Hospital. Physiotherapist at the hospital will analyse the video using the GMA method and look for abnormal movement patterns that may indicate risk of cerebral palsy.

I did not have experience with such comprehensive programming projects, so this was a challenging task to engage. I am however very interested in programming, and I was eager to learn. And so I did! Throughout this project I have gained lots of knowledge, and I am proud of the result.

I would like to thank my advisor from St. Olavs Hospital, Lars Adde, and my supervisor at NTNU, Ole Morten Aamo, for valuable help and comments throughout the process. I would also like to thank those who helped me with the programming and the report, you know who you are, and you are much appreciated.

Trondheim, 13 June 2016

Christine Østraat Buen

Summary

This master's thesis is executed in collaboration with a research project at St. Olavs Hospital and NTNU in Trondheim, concerning early assessment of infants' risk of developing cerebral palsy (CP). Infants with high risk of developing CP has a distinctive pattern of movement. The idea behind this master's project is to develop a method for parents to film their child's movements in safe surroundings with a smartphone, and send the video for assessment at St. Olavs Hospital.

The main objective of the thesis is to develop a prototype for a mobile application (app). It is designed as a "native" app, and programmed in Java for Android devices. The prototype's main task is to take a three-minute video and send it to St. Olavs Hospital. It uses an existing camera app on the phone and the user must check and approve the video according to some quality requirements before it is sent by email.

For safety reasons, the prototype requires that the user creates a profile in order to use the app. The username and password are stored on the device, inaccessible to other applications. A QR code that contains the child's unique information is issued when participating in the research project. The QR code must be scanned and the user must register the term date and a username before recording a video of the child.

The prototype also includes user guidance. It calculates when the video should be recorded from the child's term date and displays this in a timeline. A notification on the device will notify the user when it is time to record a video of a child, even if the prototype is inactive. An instruction video showing how the video should be recorded will be designed later, but a temporary version is created and included in the prototype. In addition, the app contains project information, frequently asked questions and contact information.

The project is completed according to the plan and the prototype has most of the features determined in the requirements specification. The marker indicating today's date in the graphical countdown and a notification to record a video a week in advance, are the exceptions. The prototype is tested by a representative of the research group, as well as a handful of others. Suggestions for improvements are proposed and several of these are implemented in the final version of the prototype. Not implemented suggestions are discussed in the report, and some are also proposed as a part of further development.

Sammendrag

Dette masterprosjektet er gjennomført i samarbeid med et forskningsprosjekt ved St. Olavs Hospital og NTNU i Trondheim, som omhandler tidlig vurdering av spedbarns risiko for å utvikle cerebral parese (CP). Spedbarn med høy risiko for å utvikle CP har et særegent bevegelsesmønster. Ideen bak dette masterprosjektet er å utvikle en metode hvor foreldre kan filme barnets bevegelsesmønster i trygge og rolige omgivelser med egen mobil, og sende denne til vurdering ved St. Olavs Hospital.

Hovedmålet med masteroppgaven er å utvikle en prototype til en mobilapplikasjon (app). Den ble utviklet som en "native" app, og programmert i Java for Android. Prototypens hovedoppgave er å ta en tre minutters video og sende den til St. Olavs Hospital. Den bruker et eksisterende kameraprogram på mobilen, og brukeren må sjekke og godkjenne videoen i henhold til noen kvalitetskrav før den sendes på e-post.

Av sikkerhetsmessige årsaker kreves det at brukeren oppretter en profil for å bruke prototypen. Brukernavnet og passordet blir lagret på enheten, utilgjengelig for andre programmer. En QR-kode som inneholder barnets unike informasjon blir utstedt ved deltagelse i forskningsprosjektet. QR-koden må skannes, og brukeren må registrere termindatoen og et brukernavn før video av barnet kan tas.

Prototypen består også av en veiledning til brukeren. Prototypen beregner når en video skal filmes basert på termindatoen, og viser dette i en tidslinje. En melding på enheten vil også varsle brukeren når videoen skal tas, selv om appen er inaktiv. En instruksjonsvideo som viser hvordan barnet skal filmes vil bli lagt til senere, men en midlertidig versjon er laget og lagt inn i prototypen. I tillegg kan man finne informasjon om prosjektet, ofte stilte spørsmål og kontaktinformasjon i prototypen.

Prosjektet er gjennomført i henhold til prosjektplanen og den ferdige prototypen har de fleste av funksjonene som var bestemt i kravspesifikasjonen. Varsling på enheten en uke før videoen skal tas og markøren i tidslinjen som viser dagens dato er ikke implementert. Prototypen er testet av en representant fra forskningsgruppen, samt en håndfull andre. Det er kommet forslag til forbedringer og flere av disse er implementert i gjeldende versjon av prototypen. Forslag som ikke er implementert er diskutert i rapporten, og noen er også foreslått som en del av det videre arbeidet.

Table of Contents

List of Figures

List of Tables

Abbreviations

API	application programming interface
APK	Android application package
app	application
CP	cerebral palsy
FMs	Fidgety General Movements
GMA	General Movement Assessment
HEMIT	Helse Midt-Norge IT
IDE	integrated development environment
JVM	Java virtual machine
MRI	magnetic resonance imaging
NTNU	Norwegian University of Science and Technology
OOP	object-oriented programming
OS	operating system
REC	Regional Committees for Medical and Health Research Ethics
TLS	Transport Layer Security
UI	user interface

Chapter 1

Introduction

In Norway, about 140 children are diagnosed with cerebral palsy each year (Hollung et al., 2015; Statistics Norway, 2016). Cerebral palsy is a collective term for a group of conditions with impaired motor function, which is caused by an injury to the immature brain (Andersen, 2016b). Physiatrics is the best treatment and it is important to start at the earliest possible age. Children are usually diagnosed between the age of one and two, and for this reason the treatment is often started too late.

For a long time, researchers have been working on developing methods for earlier diagnosis of cerebral palsy. In 1997, Prechtl et al. developed a method to assess the risk in children 10 to 15 weeks after the term date by observing their movement patterns (Einspieler and Prechtl, 2005). This has proven to be a reliable method. To make the method more objective, a research group in Trondheim has been working on developing computer-based methods to perform the analysis instead of human observation (Adde et al., 2010). The concept is to record a video of the child, and use an image recognition software to analyse the child's movements. The research group has conducted several qualitative studies of the method with good results (Valle et al., 2015), but more testing remains before the method may be used as a reliable tool to diagnose cerebral palsy.

To get an assessment, the parents must bring their child to the hospital where a physiotherapist records a video using a standard video camera. In Norway this may be a long trip, as many people live far from the nearest hospital. It is important for the child to be content and not distracted when filmed because this is when the particular movements may be observed. Recording the video may be difficult because the session can be stressful for both parents and child, and they may have to try again on a later occasion. Therefore, the research group wants to develop a mobile phone application (app) that enables the parents to record their child at home using a mobile phone. This ensures that the child is happy and satisfied during the recording, in addition to giving the physiotherapist more time to provide an assessment.

Developing this app is the main purpose of this master's thesis. Initially, the app will be used in a research project, but when it is fully developed and tested, it may be offered to all parents who have children with risk of developing cerebral palsy.

1.1 The Thesis Assignment

The assignment in this master's thesis is to develop a prototype for a mobile phone app. The prototype is made for a research group at Norwegian University of Science and Technology (NTNU) and St. Olavs Hospital, Trondheim University Hospital. In this thesis, Lars Adde, who henceforth is referred to as "the client", represents the research group and will also act as an advisor.

This thesis involves accounting for the requirement specifications of the app resolved in the preliminary project autumn 2015 (Buen, 2015), including changes applied afterwards and tying up any loose ends. The development platform must be decided and a development plan must be prepared for the various modules and testing of the prototype. The prototype must be developed according to the plan and well documented. The final product, along with the report, is to be delivered to the client for further use in the research project.

1.2 Main Objective

The programming of the prototype is the main objective of this master's thesis. The prototype's main purpose is to let the user record a video and send it to assessment at St. Olavs Hospital. A simple overview of the concept of this project is shown in figure 1.1. When the prototype is ready, it will be tested by a group of relevant users. The users of the app may be parents or physicians, and each user may register one or more children.

1.2.1 Limitations

Hospital Server

Helse Midt-Norge IT (HEMIT) will develop the server for receiving the videos from the final app, Therefore, the development of a server will not be a part of this project. The prototype will send the videos by email.

Feedback to the User

The solution for returning feedback to the user concerning the progress of the video assessment will not be a part of this project.

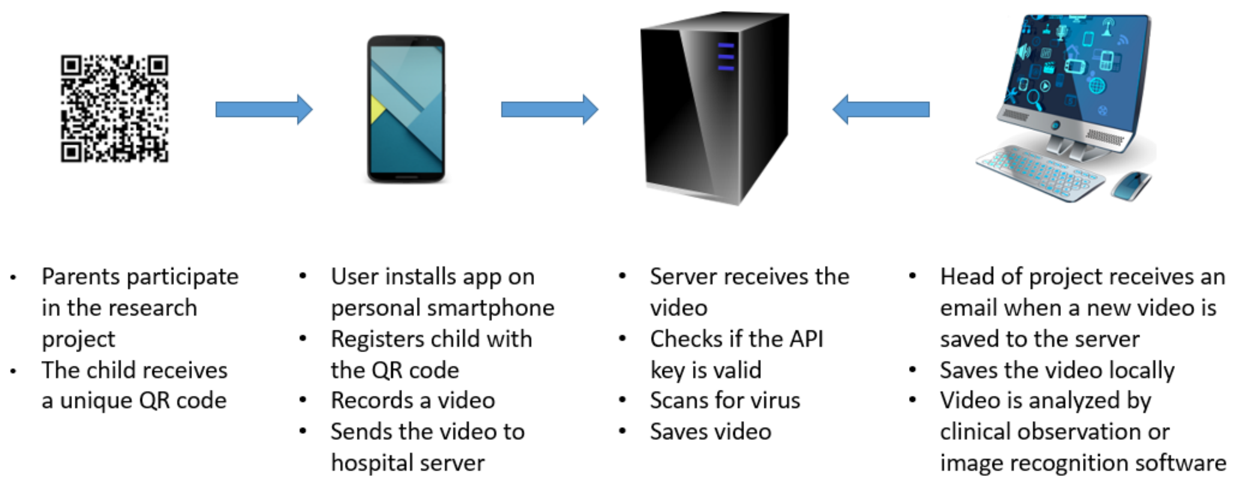


Figure 1.1: Concept of the app. The prototype will send the video directly to the head of project, not through the hospital server which will be used in the final app

Information to the User

The information to the user in the app is not a part of this project. This includes the instruction video, the about texts and frequently asked questions.

1.2.2 Prototype Requirements

Below follows a description of the main requirements for the prototype.

Register a User Profile

The user must be able to create a password protected profile before using the prototype. This is for local security of the app content and, in the final app, preventing others from sending unwanted content to the hospital server.

Register a Child with a QR code

The user must be able to register children in the prototype. A QR code will be issued from the hospital to prevent wrong registration and misspelling. Each child will get a unique QR code when participating in the research project.

Record a Video in the Right Timeframe

In the prototype, the user must get information on how and when to record a video, and be able to record a video 10 to 13 weeks after the child's term date. When outside that timeframe, it should not be possible to record a video.

Notifications on the Device

The prototype must give the user notifications on the device a week in advance and when it is time to record a video. A notification to send a video if one or more videos are recorded, but none are sent, must also be implemented.

Send a Video

The prototype must be able to attach the video to an email and use an existing email app to send it to the hospital.

1.3 Structure of the Report

The rest of the thesis is structured as follows: Chapter 2 describes the background of the project and the requirement specifications for the prototype. Method is presented in Chapter 3. Chapter 4 describes how the prototype was developed and tested, and the result of the questionnaire. Chapter 5 contains discussions on correlation between health and technology, the ethical aspect of mobile diagnostics and the result of the prototype testing and questionnaire. Conclusions and further work are presented in Chapter 6. The bibliography is listed at page 53. The .ZIP file delivered with the report contains the appendix.

Chapter 2

Background

The primary objective of the research project is to assess the risk of a child developing cerebral palsy at an early age. The earlier the child gets the diagnosis, the higher are the prognosis for a better and easier life. This chapter will introduce the reader to the diagnosis cerebral palsy, the method used in the assessment of the risk, and the work of the research group, which this project is subject to. The last part of this chapter presents the requirement specifications resolved in the preliminary project from autumn of 2015.

2.1 Cerebral Palsy

Cerebral palsy (CP) is a collective term for a group of disorders caused by an injury to the immature brain, which occurred sometime between the first trimester of pregnancy and up to 2 years of age (Andersen, 2016b). CP may affect motor function, posture, balance and more. There are several reasons why a child may develop CP. In foetal life, the brain injury may be caused by infections in the mother, drug intoxication or hormonal disorders with impaired function of the placenta (Gjerstad, 2016). Abuse of alcohol or tobacco during pregnancy may also increase the risk of CP. During labour, cerebral haemorrhage, growth retardation or hypoxia are some reasons for injury (Andersen, 2016b). This is often caused by complications during birth. After birth, brain injury may result from infections in the central nervous system, poisoning or accidents.

The most common treatment for CP is physiatrics due to the impairment of the motor functions (Gjerstad, 2016). Daily exercise is also very important for both children and adults with CP. This is to maintain mobility, prevent stiffening, disorder of posture and pain, and to maintain physical fitness and health. Other treatments can be the use of orthosis to replace or restore an impaired function, surgery to treat disorder of posture and pain, and drug treatment to reduce the symptoms of muscle spasm (Solheim, 2016).

The diagnosis CP is often given by clinical observation when the child is about 1 to 2 years old, which is when the child is old enough to develop clinical signs typical of CP. When physicians or physiotherapists do a clinical assessment they look for delayed motor development, a change in muscle tone, persisting immature movements after six months of age, and more (Andersen, 2016a). Another way to assess the risk of CP is to do an imaging of the brain using magnetic resonance imaging (MRI). MRI can sometimes show the extent of the damage, where the damage is located and when it occurred, while other times MRI gives no answers (Andersen, 2016a).

A more recent way to assess the risk of CP is by looking at the movement pattern of children 10 to 15 weeks after term date. This observation method is called General Movement Assessment and is further described in Chapter 2.2.

2.2 General Movement Assessment

General Movement Assessment (GMA) is a clinical method for assessing the risk of a child developing CP, which was developed by Prechtl et al. in 1997 (Einspieler et al., 1997). It is based on observations of Fidgety General Movements (FMs) in infants, which may be present 9 to 20 weeks after the term date (not birth date). General Movements are a part of the spontaneous movements in children from foetal life until the child is approximately six months old. FMs are small circular movements with moderate speed, but with variable acceleration of the neck, trunk and limbs in all directions. FMs are present when the child is awake and not restless, crying or distracted (Einspieler and Prechtl, 2005).

The GMA method involves a trained observer contemplating whether FMs are normal or abnormal. FMs are normal when they are present and observable. Abnormal FMs is absent or exaggerated, in which they resemble normal FMs, but the speed of the movements is different. Such exaggerated FMs are very rare. Absence of FMs, that is FMs not observed at all, is more common. Absence of FMs is a strong signal for later neurological impairments, especially CP (Einspieler and Prechtl, 2005).

One of the advantages of the GMA method is that it gives an indication of the risk at an early age. Therefore, measures are taken long before the more visible features of CP are developed and while the child's brain is responsive and the plasticity is greatest. The measures will not stop the progression of CP, as it is a permanent brain injury, but they may help improve the child's motor, social and cognitive functions. Early diagnosis can also reduce the psychological stress for the parents. By informing them that their child has a high risk of developing CP, they can get the help they need quickly. If the child is in the high-risk group, but the probability of developing CP is small, then the parents will not have to worry needlessly. Another advantage of this method is that it is non-intrusive, making the process more comfortable for the child and parents and less costly for the hospital.

A disadvantage of this method is that a considerable amount of training is required for the physiotherapists to maintain the knowledge. For hospitals with few cases of CP each year, the clinical observer will not be able to maintain the same level of knowledge as a colleague at a hospital with a larger population base. This can lead to a higher misjudgement rate or higher course expenses because the knowledge must be maintained.

2.3 The Work of the Research Group

Centre for Early Brain Development at NTNU conducts research on how to achieve optimal brain development in children with a brain injury (CEBRA, 2015a). They investigate causes, preventative measures and treatment that can reduce the injury and the consequences of the damage. Their research is divided into four topics, where one is Cerebral Palsy Group (CEBRA, 2015b). There are many projects within this group, and one of them is the project that this master's thesis is a part of: Identification of fidgety movements and prediction of CP in infants by the use of computer-based video analysis.

The research group working on this project consists of physicians and researchers from NTNU and St. Olavs Hospital in Trondheim. They have been working on improving and developing new methods for assessing the risk of CP at an early age, since 2002. At first they used Prechtl's method and clinical assessment, but have gradually increased their work with computer-based movement analysis. Because both clinical and computer-based assessment use video recordings of the child, they have developed a standard equipment package used to film the child, so every video will have the same basis for assessment.

The first computer-based video analysis was published in 2009 by Lars Adde et al. It involves looking at how much each pixel changes from one frame to the next, throughout the video (Adde et al., 2009). Several values are calculated, including "Quantity of motion", which is the sum of all pixels that changes divided by the total number of pixels in the image, and "Centroid of motion", which is the centre of all active pixels in the video. Lastly, a predicting variable for CP is calculated.

Hodjat Rahmati et al. published the second method in 2015. It involves motion segmentation of the child's arms, legs, head and trunk in the video, and looking at the movements of the different body parts separately (Rahmati et al., 2015). The calculated data provides an assessment of the FMs and with that, also an assessment of the risk of developing CP.

The research group is cooperating with hospitals in China, Belgium, India, Turkey, and USA, using the analysis method from 2009.

2.4 Similar Projects Worldwide

The research group in Trondheim competes with other research communities worldwide that also work on assessing the risk of developing CP. They are however using other kinds of data calculations. Some methods use electromagnetic sensors, accelerometers or other markers attached to the body of the child (Marcroft et al., 2014). Others use 3D motion capture. The methods developed at St. Olavs Hospital and NTNU are non-invasive and requires only a standard video camera. Thus it is inexpensive.

When it comes to working with app development associated with CP, there are similar projects in progress. According to the website of the World Cerebral Palsy Day, Christa Einspieler is part of a project in which health personnel in developing countries will use an app to record videos of infants and send them to trained observers for clinical assessment (World Cerebral Palsy Day, 2015). In Melbourne, an ongoing project is about developing an app for parents of high-risk children to use, such that they can film their child at home and send the video to clinical assessment (Cerebral Palsy Alliance, 2014; Spittle, 2015). This indicates that this is a very exciting and topical field of study.

2.5 The Requirement Specification

This chapter will present an overview of the most important requirement specifications determined in the preliminary project, concerning the prototype. The full specification is documented in the report "Development of requirements for a mobile application for the diagnosis of cerebral palsy" (Buen, 2015). This overview also contains the modifications applied after the preliminary report was completed. The prototype may be used on mobile phones or tablets, hereafter referred to as "the device" for simplicity.

2.5.1 Overview of the App

The current name of the app is PROMISE. The name was given by the research group, and is an acronym for personalized motor disability risk assessment. This name will be used in the prototype.

Figure 2.1 shows a simple representation of all of the features in the app. It describes two different start-ups. The first time the app is used on a device, the start-up will be different from the regular start-up. The difference is that the user must create a profile and register a child. The next time the user enters the app the main action will be to log in, but creating a new profile will be an additional option. This applies for registering a new child as well.

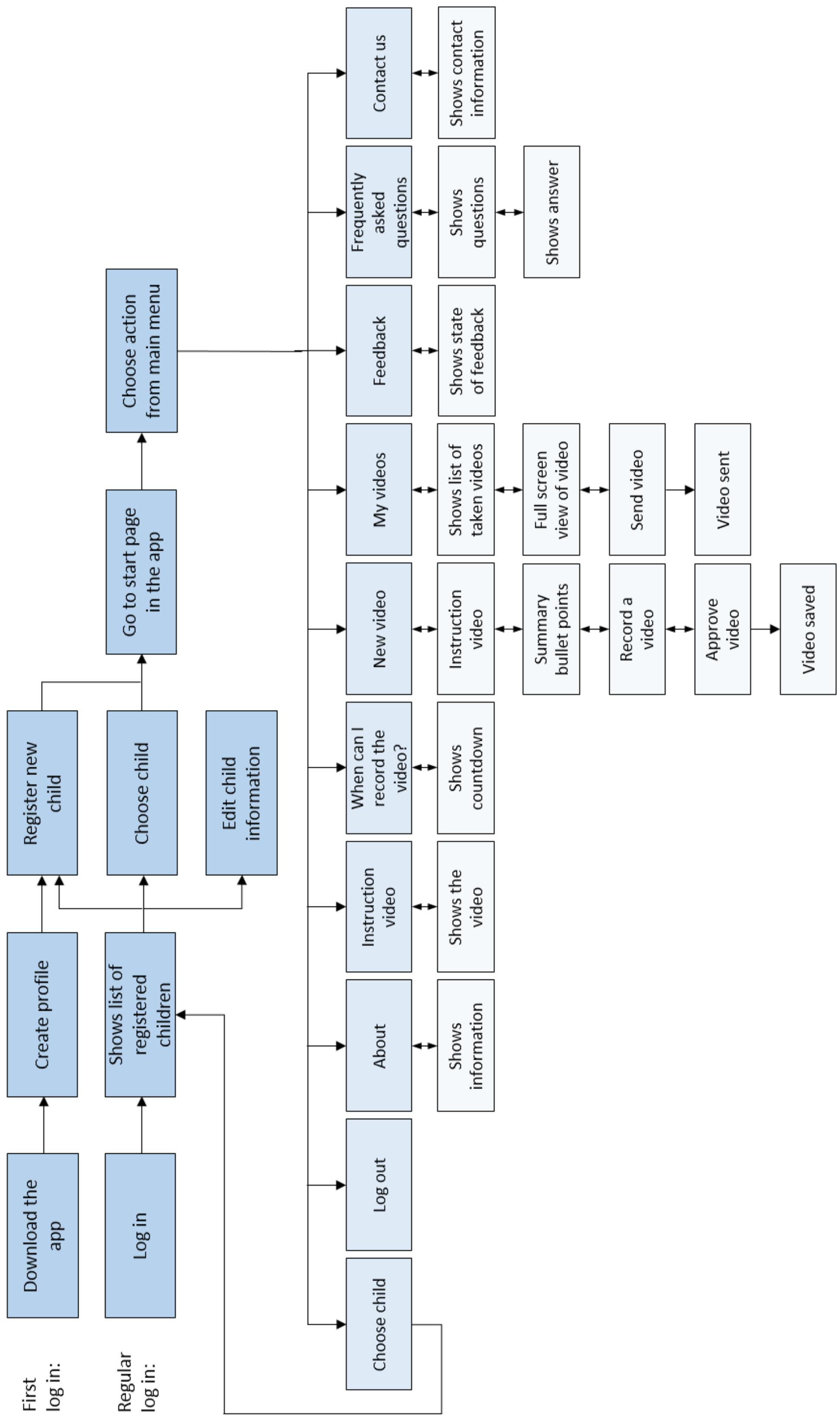


Figure 2.1: Simple representation of the features in the app

In order to make the app available to non-native speaking Norwegians and collaborating hospitals in other countries, the content of the final app must be written in both Norwegian and English. However, the prototype will only be written in English due to limited time. The language must be concise and easy to understand for both parents and physicians, and discouraging formulations must be avoided.

2.5.2 Device Permissions

The app must have permissions to use the camera on the device, and to read and write to the folder where the pictures are stored on the device. It also needs permission to allow notifications on the device.

2.5.3 Device Notifications

The app must be able to give notifications to the user on the device, even if the app is inactive. Figure 2.2 shows an example of a notification in the pull-down menu. The notifications on the device will be:

- Record a video of the child in one week from now
- Record a video within the next three weeks from now
- Reminder to send a video if one or more videos are recorded, but non are sent

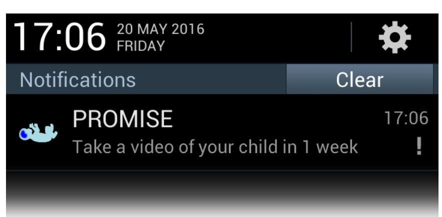


Figure 2.2: Example of how a notification in the pull-down menu may look like

Figure 2.3: Menu button

2.5.4 Main Menu

If the button in the top left corner is pressed, the main menu will be sliding in from the left and cover the existing frame. The button will look like figure 2.3. If the user presses outside the menu, it will slide left and hide. The menu will exist of the following items:

- Instruction video
- When can I record the video?
- New video
- My videos
- Feedback
- Frequently asked questions
- Contact us
- About
- Choose child
- Log out

2.5.5 User Registration

User Profile

The user must create a personal password protected profile to use the app. This will be done by entering a self-made username and password. The password must be entered twice to avoid spelling mistakes. The prototype will not have any password requirements, but a username can only be used once. The profile must be saved locally in a secure, private folder on the device. To use the app, the user must log in each time and log out when finished.

Register a Child

The child will get a unique QR code from St. Olavs Hospital with a child ID number, hospital code, country code and an API key when signing up for the project. The API key will not be used in the prototype, but in the final app. When the user sends the video to the hospital server, it will check whether the video has a valid API key. This is the only way the transmit will gain access to the server.

When registering a new child, the QR code must be scanned to register the information in the code. The user must also create a username for the child, e.g. the child's first name, and enter the term date, which will be used to calculate when to record the video. All details will be saved to the child's profile in a secure location on the device. To summarize, the information saved on a child's profile will be

- Child ID number (from QR code)
- Hospital code (from QR code)
- Country code (from QR code)
- API key (from QR code)
- Username
- Term date

List of Registered Children

After a successful login, a list of registered children on current profile is shown. If there are no registered children, the text "No registered children" is shown. The list will also be visible when "Choose child" is selected in the main menu. When a child is selected from the list, the user will have a choice between choosing this child and enter the main app or edit selected child's data. The user will also have the option to register a new child.

Edit Child

When the "Edit child" option is chosen, the details from the QR code, username and term date must be shown. Only username and term date should be editable. The details from the QR code should be permanent.

The user will get three options; cancel the edit, save changes and delete the child. If the user chooses to delete the child, the child's details will be deleted from the database after the user confirm the deletion in a dialogue window. However, the videos recorded with the app should still be saved in a folder on the device, accessible from other apps.

2.5.6 Video Recording

When Can the User Record the Video?

The app will contain a countdown to show the user when the video must be recorded. The time-frame in which the user can record a video is three weeks. This is 10 to 13 weeks after term date.

The 13-week limit provides a 2-week period for an additional video if needed.

The countdown will also contain a timeline where the user can see the term date, today's date and the dates when the video must be recorded. When it is time to record a video, a button which will take the user directly to the new video action will appear.

Record a Video

There are two ways to use the camera on a device. The app can use the existing camera app on the device (Android Developers, 2015c) or implement a solution within the app itself (Android Developers, 2015b). This project requires a specialised camera function. Therefore, the second solution is preferable. To ensure a correct distance from the camera to the child, the camera needs a frame on the screen to target the child. It is also preferable with a large font of the timer on the screen or a 3-minute countdown, because that is how long the video should be to get a good assessment.

The video will be saved locally on the device in a folder created by the app called "PROMISE". The file should be named: "[child's username]_[year]_[month]_[day]_[hour]:[minute].mp4". After recording a video, the user must approve the quality according to a few criteria.

List of Recorded Videos

A list of recorded videos of selected child will be visible in the app when "My videos" are chosen in the main menu. The user will get two options after selecting a video. Either the user can show a full screen preview of the video or choose to send it.

Send a Video

When the user chooses to send a video in "My videos", the app must open an existing email app. This requires that the user has already configured a private email on the device. The video must automatically be included as an attachment in the email, and the user must enter the recipient's address. This will most likely be the client's email address when testing the prototype.

2.5.7 Information to the User

The app will contain important information to the user. As described in Chapter 1.2.1, this information will not be made in this thesis. The prototype will only contain illustrating text and a short instruction video. This information will be altered later.

Instruction Video

One option in the main menu will be to view the instruction video. When this is selected, a thumbnail of the video with a play button will appear. If the user presses “Play”, the video will open in a full screen view. The user will be able to pause, rewind, fast forward and exit the preview.

Frequently Asked Questions

A list of frequently asked questions will appear if this alternative is selected in the main menu. If the user presses the plus sign to the left of each question, the answer will appear beneath. The plus sign will simultaneously convert into a minus sign, which will hide the answer when pressed.

About the App and the Project

The app will contain information pages about the app and the project. One is in the start-up page where the user can read about the app and how to use it. Another will be in the main menu where the user can read more about the research project and the risk assessment.

Contact Us

In the main menu there will be a contact information page where the user can find phone number, email address and possibly a home page for the research project. This information will be obtained by the research group.

Feedback

The user can follow the status of the video by choosing “Feedback” in the main menu. This page will show the user if a video is sent, if the video is registered and approved for analysis, and if the video is analysed and a letter is sent to the parents with the results. The solution for how to achieve feedback is not a part of this thesis, therefore it will be presented only as a demonstration of how it may look like in the final app.

Chapter 3

Method

This chapter presents the platform that the prototype is developed for, and the development tools and programming language used in the prototype development. It will also introduce the testing of the prototype and the questionnaire, which will be given to a random selection of parents with preterm born infants at St. Olavs Hospital.

3.1 Prototype Development

3.1.1 Platform

The prototype is developed as a *native app* for Android devices. Native implies that the app is written in a programming language supported by a specific mobile operating system (OS), as opposed to *web apps* that are written in a standard web language and can run on several platforms (Marszałek, 2016). A third option is *hybrid apps*, which is a web app hidden behind a native app shell. Advantages and disadvantages of native, web and hybrid apps are presented in table 3.1.

As the app requires customised camera features, a web app is inadequate. The choice between hybrid and native was based on my previous programming experience and the fact that hybrid apps will not have the same familiar user experience as native apps. I do not have much experience with web languages, but I learned the basics of Android and Java programming last semester. Therefore, the prototype was made as a native app for Android devices, with Java as the programming language.

3.1.2 Android Studio

The development software used in this project is Android Studio. When Android Studio 1.0 was released in December 2014, it replaced Eclipse Android Development Tools as Google's primary

Native apps	Advantages:	- Easy access to the device's functionalities - Fast performance - Familiar user experience - Internet access not required - Distributed in app stores
	Disadvantages:	- One specific platform - Costlier to develop
Web apps	Advantages:	- Cross-platform - Faster and less costly to develop
	Disadvantages:	- Lack of native features - Not distributed in app stores - Slower performance than native
Hybrid apps	Advantages:	- Cross-platform - Access to some native features - Faster and less costly to develop - May be distributed in app stores
	Disadvantages:	- Lack of some native features - Slower performance than native

Table 3.1: Advantages and disadvantages of native, web and hybrid apps

integrated development environment (IDE) for Android app development (Android Developers, 2016a). Since then, there have been various fixes and enhancements. The latest version, Android Studio 2.0, was released in April 2016, but Android Studio 1.5.1, released in December 2015, was used throughout this project.

Android Studio is based on IntelliJ IDEA, which is a Java IDE. It provides intelligent code editing capabilities such as advanced code completion, refactoring, and code analysis, and it allows you to compile and run your program in a single step. The IDE also has a drag-and-drop environment such that you can design the graphical interface visually, in addition to the text editor. Android Studio also provides different user interface previews, such as various devices, screen sizes and API levels which is very useful when designing the layouts (Eason, 2014).

The Android build system is based on Gradle (Android Developers, 2015a). It is an automated build toolkit for building, testing, running and packaging the app. With Gradles you can generate multiple Android application packages (APKs) with different package and build configurations from a single module (Thornsby, 2015). This means creating apps which can support a wide range of devices or create different versions of an app, such as a free version and a professional version. Another advantageous feature is that Gradle automatically generates a test directory and a test APK

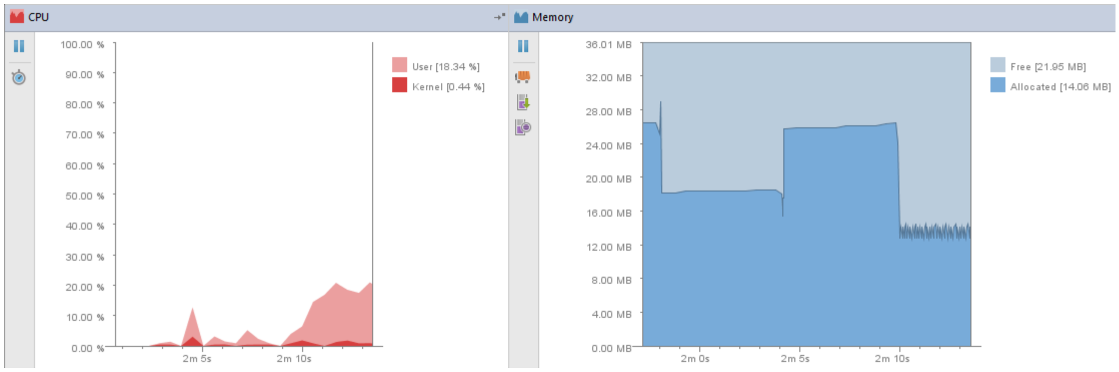


Figure 3.1: CPU and memory monitors in Android Studio

from the project's test sources and can run tests during the build process.

Android Studio supplies project templates which can help you get started with a new project or you can import Google code samples (Eason, 2014). It also provides a CPU and Memory monitor which can help you with performance analysis by showing the CPU and memory usage of the app over time, see figure 3.1.

The programming language used in Android Studio is Java.

3.1.3 Java

In the mid-1950s, the first modern computer programming language, FORTRAN, was developed (Barry, 2011). About 20 years later, Dennis Ritchie at AT&T Bell Labs develops the C programming language. They are both based on the idea that you issue direct, imperative commands to the computer ("Do this, and then do that"). In the 1970s, the first object-oriented languages were developed, but not well received, until C++ was created in 1986 which combined the old C language terminology with the object-oriented structure. Finally, in 1995, James Gosling of Sun Microsystems developed Java programming language. It is based on the look of C++, but most of the old "Do this" features are gone and the language has a pure object-oriented philosophy.

In 2010, Oracle purchased Sun Microsystems and they are further developing the language today. The development is done through an open community called Java Community Process. Still, Java is not open source code because Oracle owns the trademark and reserves the right to lay down the user terms and approve all changes and releases (Rossen, 2014).

Java is an object-oriented language. This implies that the focus is on organizing the data and the commands come later (Barry, 2011). The first step is to identify the objects and how they relate to each other (Rouse, 2008). Once an object has been identified, it is generalized as a class of objects which defines the kind of data it contains and any logic sequences that can manipulate it called methods. One of the benefits of object-oriented programming (OOP) is called inheritance, which makes it possible to define subclasses of data objects that share some or all of the main class characteristics, in other words the subclass inherits features from the main class. This property reduces development time and ensures a more accurate coding.

One of the main advantages with Java is portability (Barry, 2011). This means that the language is not specific for one kind of OS or one kind of processor. The compiler takes the Java source code and translate it into Java bytecode. The device itself must have a Java virtual machine (JVM) to translate Java bytecode into a language specific for its OS, and to run the program (Rossen, 2014).

3.1.4 XML

XML is a markup language and it is short for Extensible Markup Language. It was developed by the XML Working Group in 1996, led by Jon Bosak at Sun Microsystem (Bosak, 2007). XML describes a class of data objects called XML documents (Cowan et al., 2008). The documents are made up of entities, which are storage units containing either parsed or unparsed data. It is composed of declarations, elements, comments, character references, and processing instructions, all of which are indicated in the document by explicit markup. The markup encodes a description of the document's storage layout and logical structure.

In Android Studio, XML files are used to design the screen layout of each activity in the app. The layout is set by the Activity class using the `setContentView(View)` method in the beginning of the `OnCreate(Bundle)` method. When designing the layout in the text editor, Android Studio provides a preview window in which you can see the result of the XML code while writing. When designing the layout in design mode, Android Studio will convert the graphical interface into source code.

3.1.5 GitHub

Git is a free and open source version control system, and is the core technology that GitHub is built on top of (GitHub Help, 2013). GitHub is a program that is tracking changes in files such that specific versions may be recalled later.

By pushing committed changes to the online repository, GitHub was used to save backups of the source code in this project. It is possible to see the difference between two commits, or an earlier version and current local version. It is also possible to pull changes to the local copy if you wish to restore an earlier version or if there are other contributors working on the same project and you want the source code to be up to date.

3.1.6 User Interface

It is important to create the user interface (UI) of the app as simple as possible. Advices on optimising the UI emphasise making the app “simple” and “easy to learn” (Cline, 2015; Williams, 2015). The intention of this project is to create a solution to replace the current practice. To make sure the new solution is preferred, it is important to spend time and resources on making it as user-friendly as possible.

When developing a new app, one of the most important things to consider is to follow the standards (Android Developers, 2012a). Users like to recognise features and layouts. It is also important not to mix elements from different OS’s. This makes it easier for the developer who will not have to develop new solutions each time. Standard Android buttons, menus, text fields etc. are used in the prototype.

Another important tool for creating a simple UI is to keep the UI graphical whenever possible (Cline, 2015). Examples of graphical UIs in the prototype are: the main menu will include icons as well as text, the countdown to the next video will be displayed graphical and an instruction video will show the user how to record a video, instead of a textual description.

It should constantly be assumed that people make mistakes (Williams, 2015). In the prototype, users could have unintentionally deleted a profile of a child or accidentally sent a video of poor quality. Examples of the measures taken to prevent this are, respectively, a dialogue box in which the user must confirm the deletion of a child profile and a checklist with quality criteria for the video before the “Send” button is activated.

3.2 Prototype Testing

In this project I made two prototype tests; a usability test (Appendix C) and a technical test (Appendix B). The first test will be carried through without any guidance. This is because it is interesting to see how intuitive the prototype is and to see whether a user without knowledge of the project will understand how to use the prototype. This may also give an indication of what should be included in the instruction video.

The second test is to investigate whether the prototype does what it is supposed to do. The test form for the technical test consists of 86 numbered questions, and is to be completed in the given order. This will take the user through the entire prototype and test all the functions. Any errors will be written down.

It is preferable to include both parents and physicians in the control group because both groups will be potential users of the final app. Physiotherapists in the research group at St. Olavs Hospital will be the most important participants in this test.

3.3 Parental Survey

To gain knowledge of what parents think about the project idea, I made a simple questionnaire in collaboration with the client. We got a verbal approval of the survey by the Regional Committees for Medical and Health Research Ethics (REC) as long as certain arrangements were made. The questionnaire was anonymous and contained five questions. REC was clear on the fact that the parents would have to initiate their participation in the survey. Therefore, they were given the form on paper at the hospital. They would then decide whether or not to participate in the survey. The questionnaire is presented in Appendix D.

Chapter 4

Results

This chapter presents the prototype of the app, which is the main object of this master's thesis. The result is presented as screen shots of the pages in the prototype with a description of how it works. Mainly, the Support Library in Android Studio is used, but when extra libraries or classes are imported or created, this is specified. After the prototype development is presented, the results from the testing of the prototype and the parental survey are described.

4.1 Prototype Development

4.1.1 Compatible Android Versions

The prototype supports Android application programming interface (API) level 16 and up to API level 22. The prototype was made using a test phone with Android 4.1.1, but it is also tested on Android 5.1. By May 2, 2016 the percentage of Android users compatible with the prototype was 88.5 %, see figure 4.1.

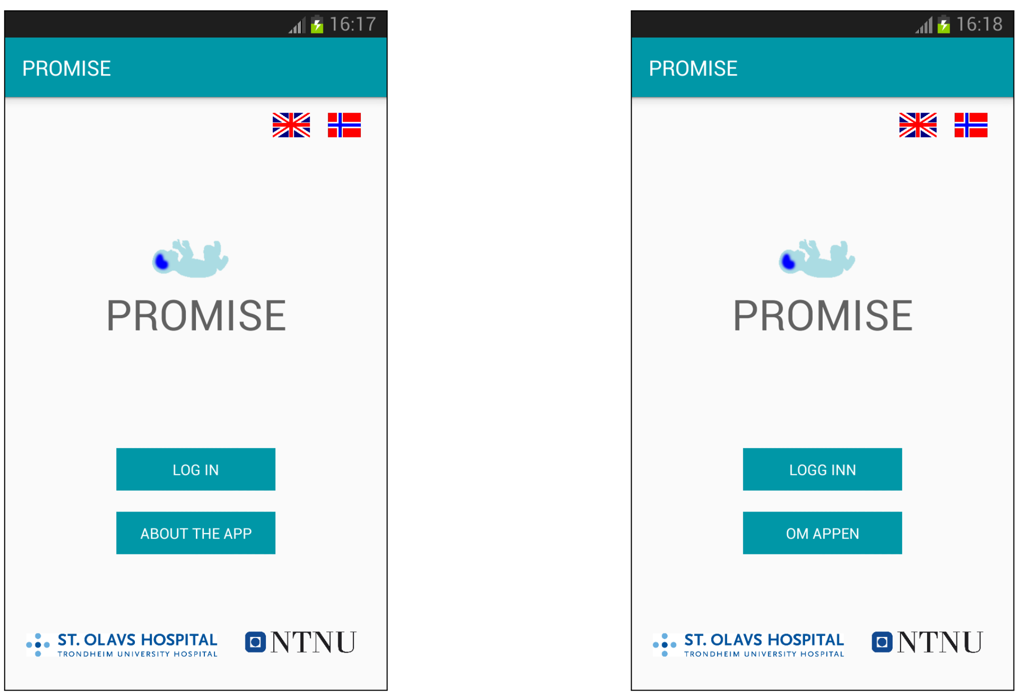
After Android 6.0 (API level 23) was released in October 2015, considerable changes were made to the handling of device permission, as discussed later in Chapter 4.1.3. Therefore, the prototype is not compatible with devices with this operation system or newer.

4.1.2 Literary Language

As described in Chapter 2.5.1, the content in the prototype is only written in English. However, the first page gives a demonstration on how the user can choose between English and Norwegian, see figure 4.2. By clicking on the flags in the top right corner, the button text changes between English and Norwegian.

Version	Codename	API	Distribution
2.2	Froyo	8	0.1%
2.3.3 - 2.3.7	Gingerbread	10	2.2%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	2.0%
4.1.x	Jelly Bean	16	7.2%
4.2.x		17	10.0%
4.3		18	2.9%
4.4	KitKat	19	32.5%
5.0	Lollipop	21	16.2%
5.1		22	19.4%
6.0	Marshmallow	23	7.5%

Figure 4.1: Distribution of devices running the different Android versions, by May 2, 2016 (Android Developers, 2016b). The yellow marked percentages indicate the number of compatible users for the prototype, which is 88.5 % in total



(a) English **(b)** Norwegian

Figure 4.2: The first page in the prototype in (a) English and (b) Norwegian. The user chooses between the two languages by clicking on the flags in the top right corner

4.1.3 Device Permissions

A key point in the security architecture in Android is that by default no app has permission to perform any operations that would adversely impact other apps, the operating system, or the user (Android Developers, 2012b). To use protected features of the device, such as the camera module or read external storage, `<uses-permission>` tags must be included in the app's manifest. The permissions are divided into two groups; normal and dangerous permissions. The system will automatically grant a normal permission, such as `SET_ALARM`, but for a dangerous permission, such as `READ_CONTACTS`, the user is asked to explicitly grant the permission.

As from Android 6.0 (API level 23), the app requests permissions from the user at run-time, meaning that the user can revoke the permissions at any time. In this case, the app needs to check whether it has the permissions every time it runs. For older Android versions, which this prototype is made for, the system asks the user to grant the permissions when installing the app. To revoke the permissions, the user has to uninstall the app.

A count done by Android from May 2, 2016 shows that only 7.5 % of the Android devices is running Android 6.0 (Android Developers, 2016b). Thus, the prototype is still usable for most of the Android users. However, for further developing of the app this is something that must be considered.

In the prototype, app permissions are included in the AndroidManifest.xml file. The request for permission to use the camera on the device is written as:

```
<uses-permission android:name="android.permission.CAMERA" />
```

This format is used for the other permission requests as well. The permissions required by the prototype are:

- `CAMERA`
- `RECORD_AUDIO`
- `WRITE_EXTERNAL_STORAGE`
- `READ_EXTERNAL_STORAGE`
- `WAKE_LOCK`
- `RECEIVE_BOOT_COMPLETED`

4.1.4 Data Storage

There are several ways to store app data in Android, such as shared preferences and internal and external storage (Android Developers, 2016c). SQLite database is used in the prototype to store structured data in a private database on the device, inaccessible to other apps. The data is stored in tables, and the database may consist of as many tables as needed. In this case four tables are used. User profiles, children, videos and current values are stored in separate tables.

Java classes were made for Profile, Child, Video and CurrentValue. Creating objects of these classes simplified the management of the database. A java class was implemented for managing the database. This class is called MyDBHandler and executes different SQLite queries to perform different tasks, such as read from or write to the database. Each `query()` method returns a cursor which points to the rows found by the query (Android Developers, 2016c). To write to and read from the database, the methods `getWritableDatabase()` and `getReadableDatabase()` are called and the cursor is used to navigate to the part of the database where the changes should be made.

4.1.5 Device Notifications

To create a simple notification, the procedure in the sample project "Basic Notification" in Android Studio can be used. It is based on the NotificationCompat.Builder class which is implemented in the standard Support Library in Android Studio. This produces a notification in the pull-down menu when a button is pressed. To implement notifications triggered at a certain time you must use an alarm. Alarms are based on the AlarmManager class which allows the app to run at a scheduled time (Android Developers, 2011a). There are different types of alarms (Android Developers, 2013b). In this project we need to wake up the device if it is in sleep mode and give a notification at a specified time, so the `RTC_WAKEUP` is best suited. We need a BroadcastReceiver class to listen for intents broadcasted by the AlarmManager, which creates a new intent to start a Service class to handle the wake up and display the notification. A Service is a component that performs operations in the background and without a user interface (Android Developers, 2011c). The BroadcastReceiver should in this case be the subclass WakefulBroadcastReceiver, with a `WAKE_LOCK` permission which will ensure that the device does not go back to sleep before the Service class is called (Android Developers, 2013c). When the Service is finished, it must release the wake lock to allow the device to go back to sleep.

When using a scheduled alarm, it is also beneficial to implement a BroadcastReceiver which listens for intents from the operational system such as `TIMEZONE_CHANGED`. If this intent is received, the existing alarm will be cancelled and reconfigured. This will prevent the user from getting alarms in the middle of the night during travels.

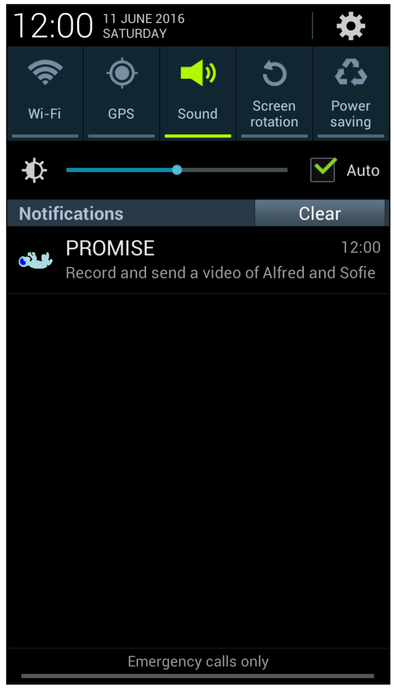


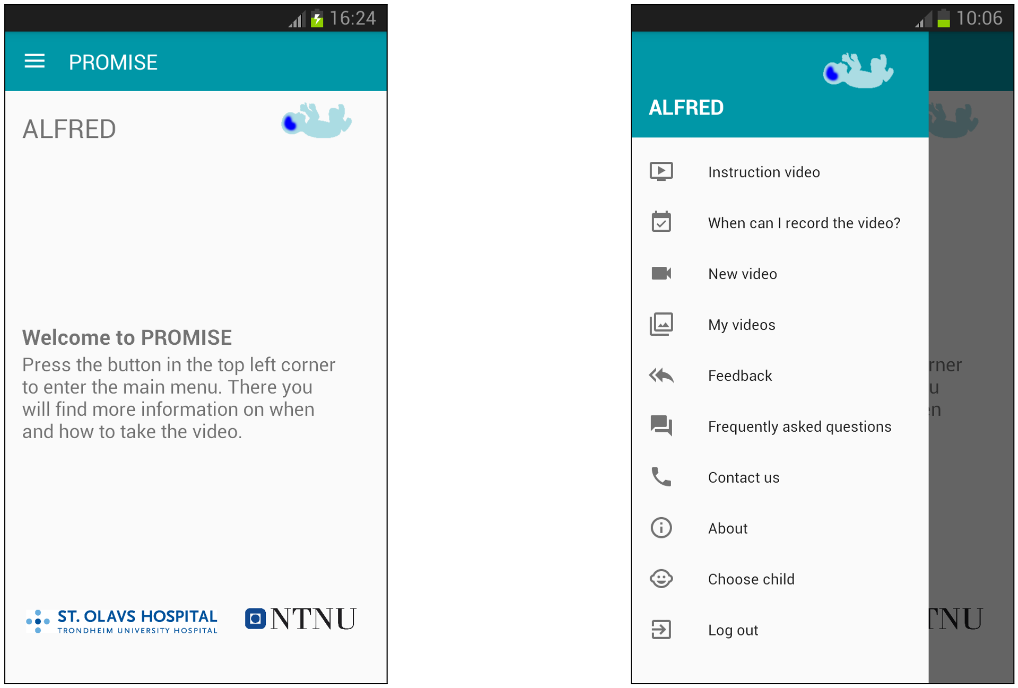
Figure 4.3: A notification on the device created by the prototype

In the prototype, the alarm is scheduled to send an intent every day at 12:00. The service class will wake up the device, and a method is implemented to check whether or not the app should send a notification. This method will read from the Child table in the SQL Database and get the term date of each registered child. If it is time to record a video, it will check if a video is already sent of this child, and if not the child's username will be added to a list. When the method has checked all registered children, it will return the list and the service class will display a notification which includes the names in the list, see figure 4.3. If no children are within the timeframe for a video or if a video has been sent of all the children, no notification will be sent.

This notification combines two of the notifications presented in Chapter 2.5.3. Within the timeframe it will remind the user every day at noon to record a video until a video is sent. The prototype will open with the login page when the user presses the notification. The notification a week in advance is not implemented due to limited time.

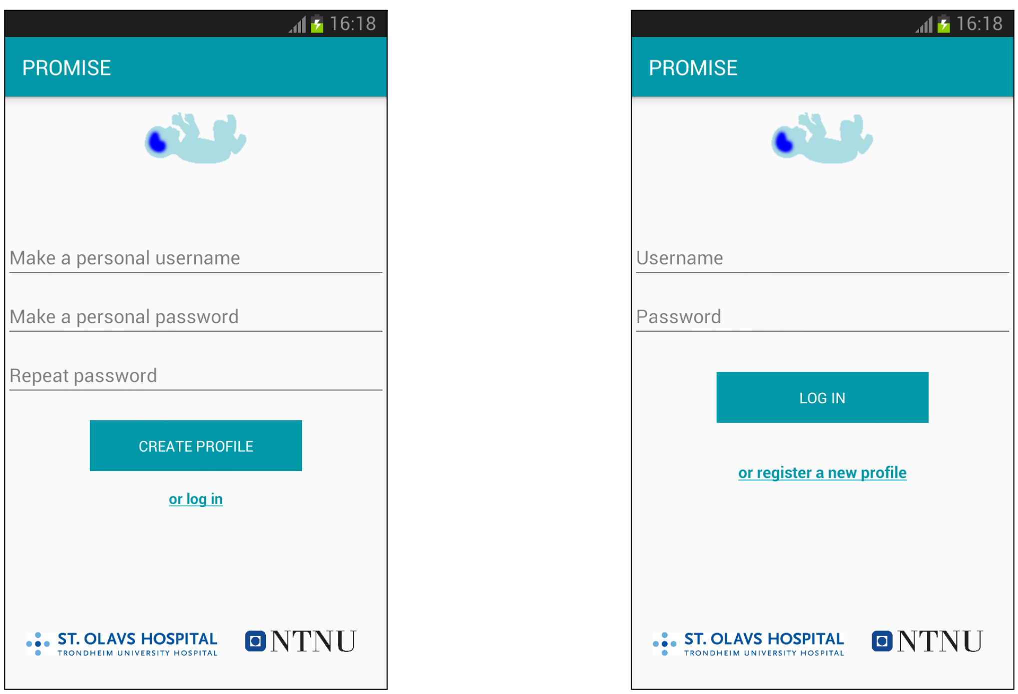
4.1.6 Main Menu

When the user presses the menu button in the top left corner of the page in figure 4.4a, a menu will slide inn from the left and reveal the options in the app, see figure 4.4b. The menu is a Navigation Drawer consisting of the navigation options for the app. The DrawerLayout class from the Support Library is used.



(a) **(b)**

Figure 4.4: (a) shows the menu button in the top left corner and (b) shows the menu options



(a) First login **(b)** Regular login

Figure 4.5: The login page in the app. (a) is the main action the first time the user opens the app. When the user has created a profile, (b) will be the main action when opening the app

4.1.7 User Registration

User Profile

When the user first opens the app and presses the “Log in” button on the first page of the app, a profile must be created, see figure 4.5a. The user must create a username and a password. The password must be entered twice. The app checks whether the passwords entered are alike or not, and if all fields are filled in. Then the profile is saved in the SQL database in the Profile table. If not, the user will receive a message on the screen describing the error.

When a profile is created, the main login screen will be the page shown in figure 4.5b. The username and password entered by the user will be checked against the Profile table in the SQL Database. If correctly entered, the user will proceed in the app. The user will also have the option of creating a new profile and that will again take the user to the page in figure 4.5a. The prototype does not have a limit for number of profiles.

Register a Child

As described in Chapter 2.5.5, each child participating in the research project will get a unique QR code. The format of the information in the QR code will be in JSON, see figure 4.6. JSON stands for JavaScript Object Notation (JSON.org, 2003). This is a simple text format which is easy for humans to read and write, and easy for machines to parse and generate. It is also programming language independent. The library used to implement the QR code scanner in the prototype is called ZXing. It is an open-source, multi-format 1D/2D barcode image processing library implemented in Java (Owen, 2016).

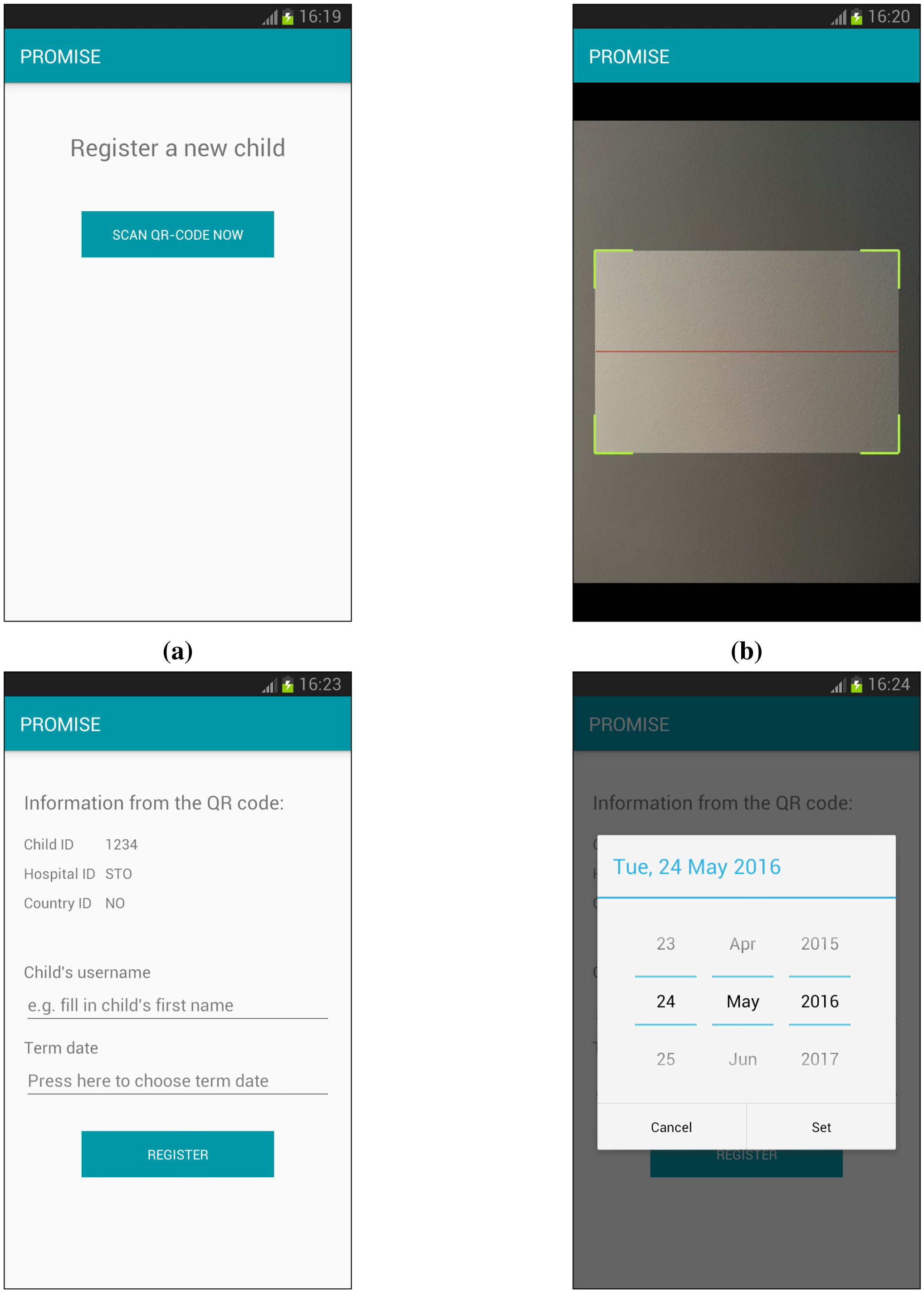
The steps in registering a new child in the prototype is displayed in figure 4.7. When the button in 4.7a is pressed, the app will use the camera to scan for a QR code as shown in figure 4.7b. When a valid QR code is discovered, the information is read and displayed in the next page, see figure 4.7c. The user must then create a username for the child and enter the term date. The term date is chosen by pressing the term date text field and scrolling the day, month and year bars which will appear, see figure 4.7d. When the “Register” button is pressed, the app will check the inputs and if

```
{
"CHILD_ID": "1234",
"HOSPITAL_ID": "STO",
"COUNTRY": "NO",
"API_KEY": "1234abcdEFGH#!"
}
```

(a)

(b)

Figure 4.6: QR code generator example: (a) is the input in JSON format and (b) is the output QR code



(a) (b) (c) (d)

Figure 4.7: These are the steps in registering a new child in the prototype: When the button in (a) is pressed, the app will use the camera to scan for a QR code (b), and if the QR code is valid, the information is read and displayed in (c). The user must then create a username for the child and choose the term date (d).

everything is correct the information will be stored in the SQL Database in the Child table and the user will enter the main page (figure 4.4a).

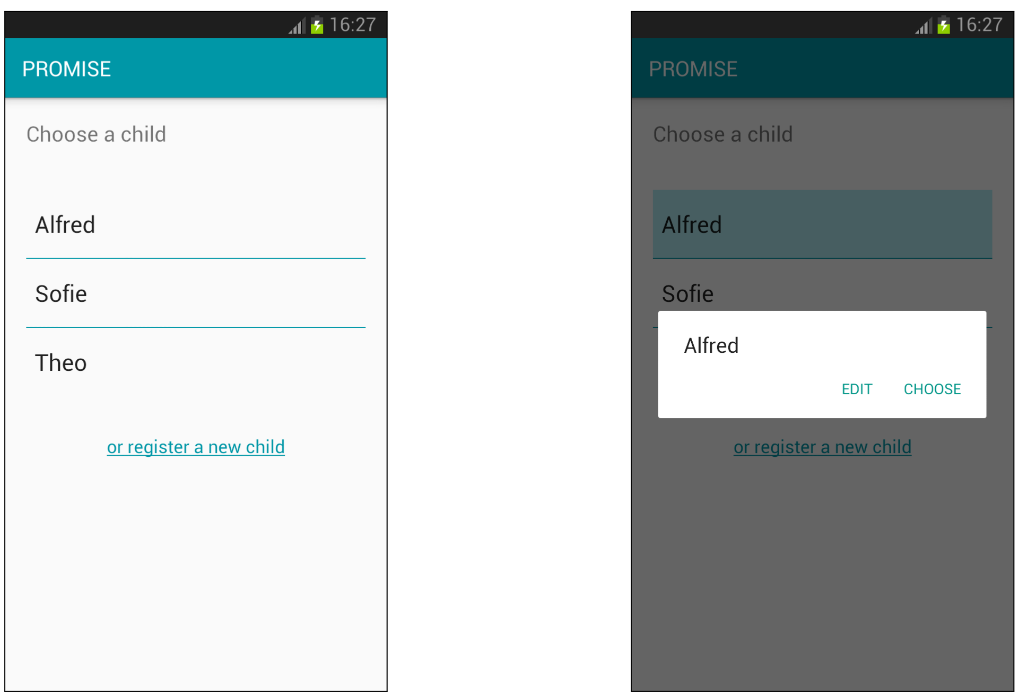
Each QR code can only be used once per device. The child's username however may be reused, but only once per profile. To ensure this, the profile name is saved with the child's details in the Child table in the SQL Database, in addition to the details listed in Chapter 2.5.5.

List of Registered Children

After a user has logged in, a list of registered children will appear, see figure 4.8a. The app reads from the SQL Database in the Child table and search for all children registered on current logged in profile. The children's usernames are displayed in a ListView. This list will also appear if the user selects "Choose child" in the main menu.

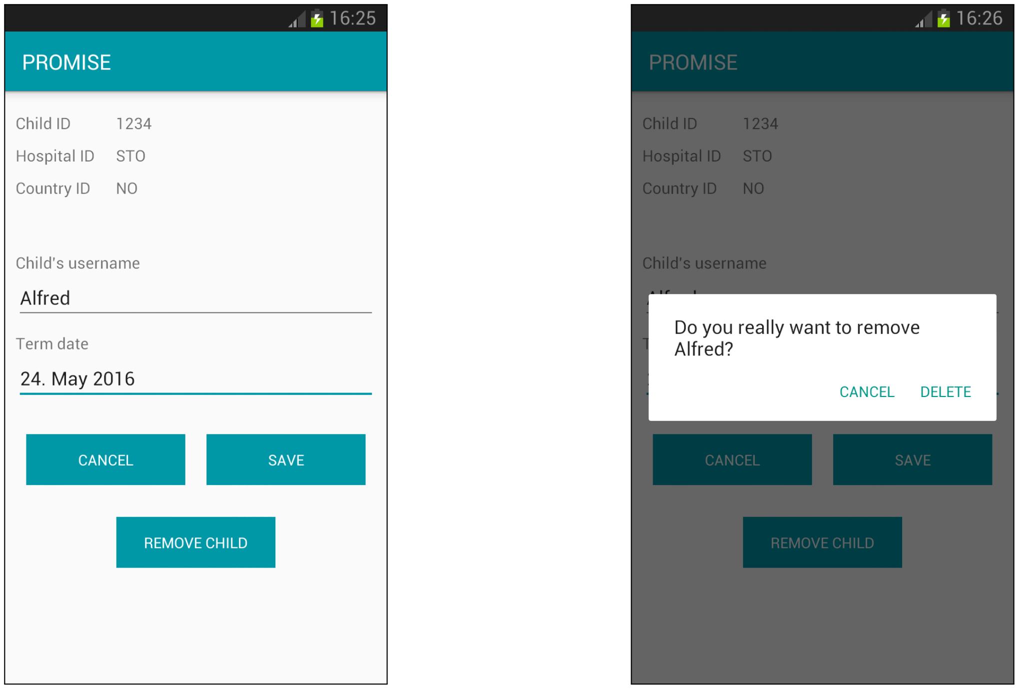
Edit Child

When the user selects "Edit" in figure 4.8b, a screen with the information on selected child stored in the Child table in the SQL Database is displayed, see figure 4.9a. Only username and term date are editable. If the user chooses "Cancel" or "Save", the changes will, respectively, be disregarded or updated in the Child table and the user is sent back to the list of children. If the user chooses to



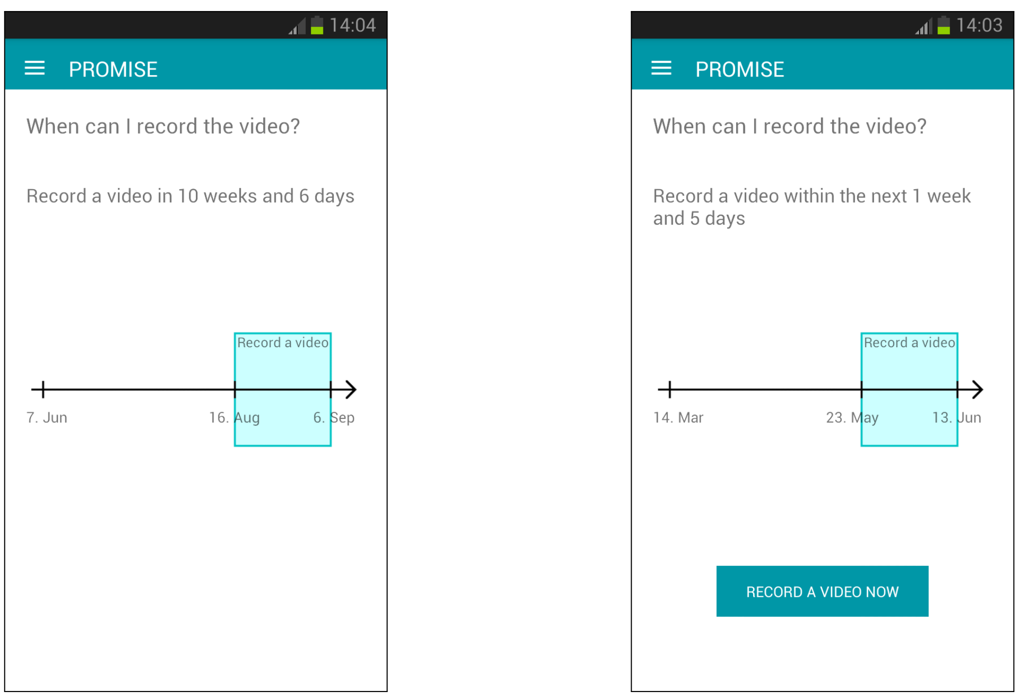
(a) (b)

Figure 4.8: (a) shows the list of registered children and (b) shows the options after selecting a child



(a) (b)

Figure 4.9: Edit a child: (a) displays the information on selected child saved in the Child table. Username and term date are editable. (b) shown the dialogue window when the user chooses to remove child



(a) (b)

Figure 4.10: (a) shows the weeks and days until it is time to record a video. If it is time to record a video the time left is displayed (b). The timeline shows a graphical presentation. The first date is the term date

remove the child, a dialogue window will appear and the user must choose between "Delete" or "Cancel", see figure 4.9b. This is to prevent the user from deleting children unintentionally.

4.1.8 Video Recording

When Can The User Record the Video?

The countdown is calculated from the term date entered when registering a child. The video must be recorded between 10 and 13 weeks after the term date. The number of weeks and days until it is time to record a video is written above the timeline, see figure 4.10a. If it is time to record a video, the time left of the 3-week period will be displayed and a "Record a video now" button will appear, see figure 4.10b.

The graphical presentation of the countdown consists of a timeline showing the term date and the two dates representing 10 and 13 weeks after the term date. The period in between these two dates is marked with a blue area, showing the user clearly when to record the video. The timeline in the prototype is incomplete. There should have been a active marker indicating today's date moving towards the blue area, but there was not enough time to implement this.

Record a Video

The prototype is using the existing camera app on the device. There was not enough time to achieve an implemented camera function within the app itself. The final solution is working well, but it has a few inconveniences. The font of the timer on the camera is too small and the user gets an extra action since the video must be saved before returning to the app, see figure 4.12a.

The camera intent is one of the common intents in Android (Android Developers, 2013a). An intent is used to start an activity by describing the wanted action in an intent object. The camera intent has two actions; `ACTION_IMAGE_CAPTURE` and `ACTION_VIDEO_CAPTURE`. The second action is used in the prototype. The `onActivityResult()` method is used to handle the result of the camera action. This method provides a result code, telling the prototype whether the camera action was successful or not. This will determine the next action in the prototype.

The steps in the process of recording a new video is shown in figure 4.11 and 4.12. First, the instruction video is shown, see figure 4.11a. Secondly, a brief summary of the most important topics in the instruction video is repeated in a check list, see figure 4.11b. When all the boxes are checked the "Continue" button becomes active, see figure 4.11c and when pressed, the user accesses the existing camera app, see figure 4.11d. Before returning to the app after recording the video, the user must press "Save" in the existing camera app, see figure 4.12a. Then, the video must be reviewed and the user must consider whether the video fulfil certain criteria, see figure 4.12b.

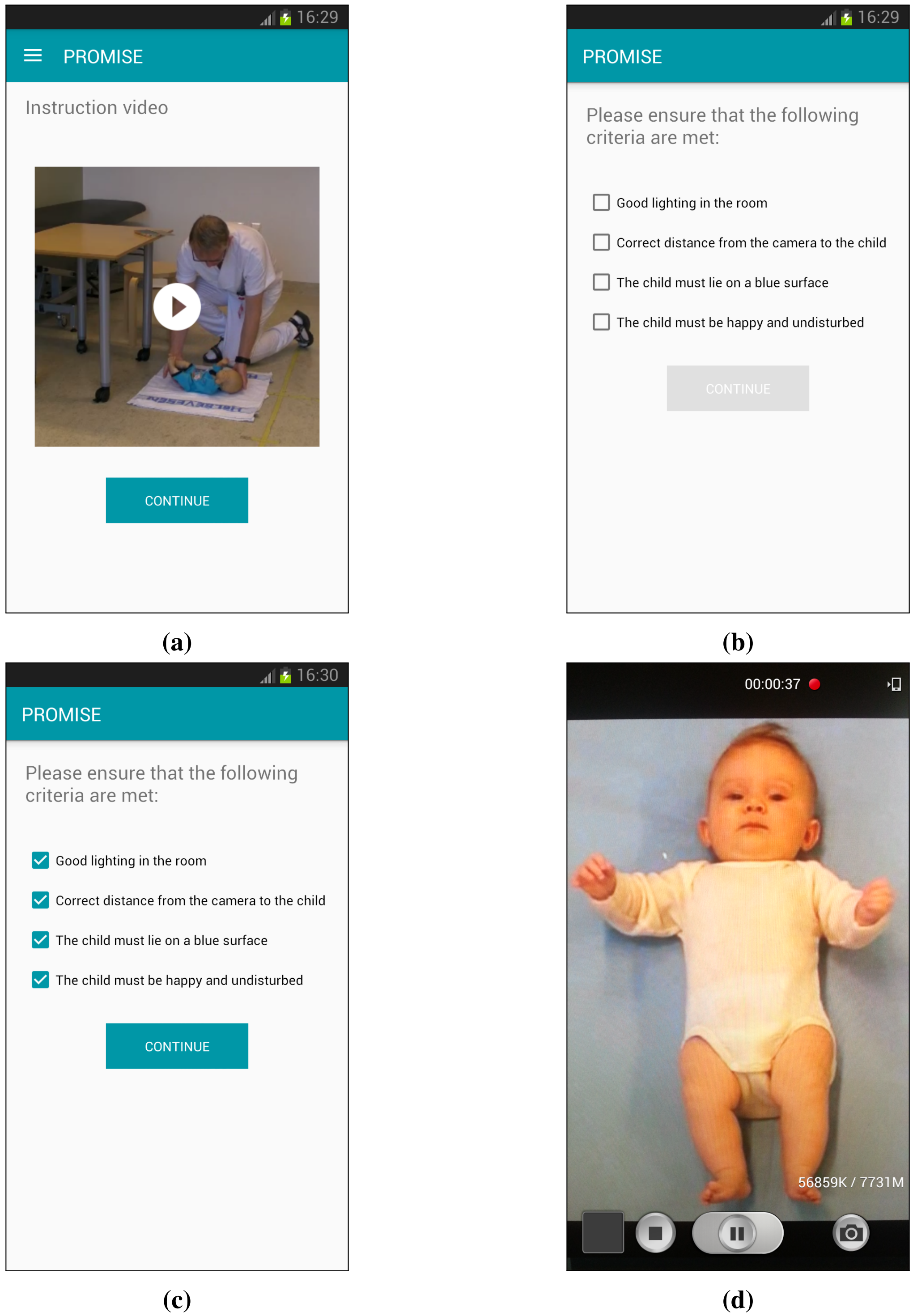


Figure 4.11: Part 1 of the "New video" procedure: First, the instruction video is shown (a), secondly, a brief summary of the most important topics in the instruction video is repeated in a check list (b). When all boxes are checked, the continue button becomes active (c) and the existing camera app is accessed (d)

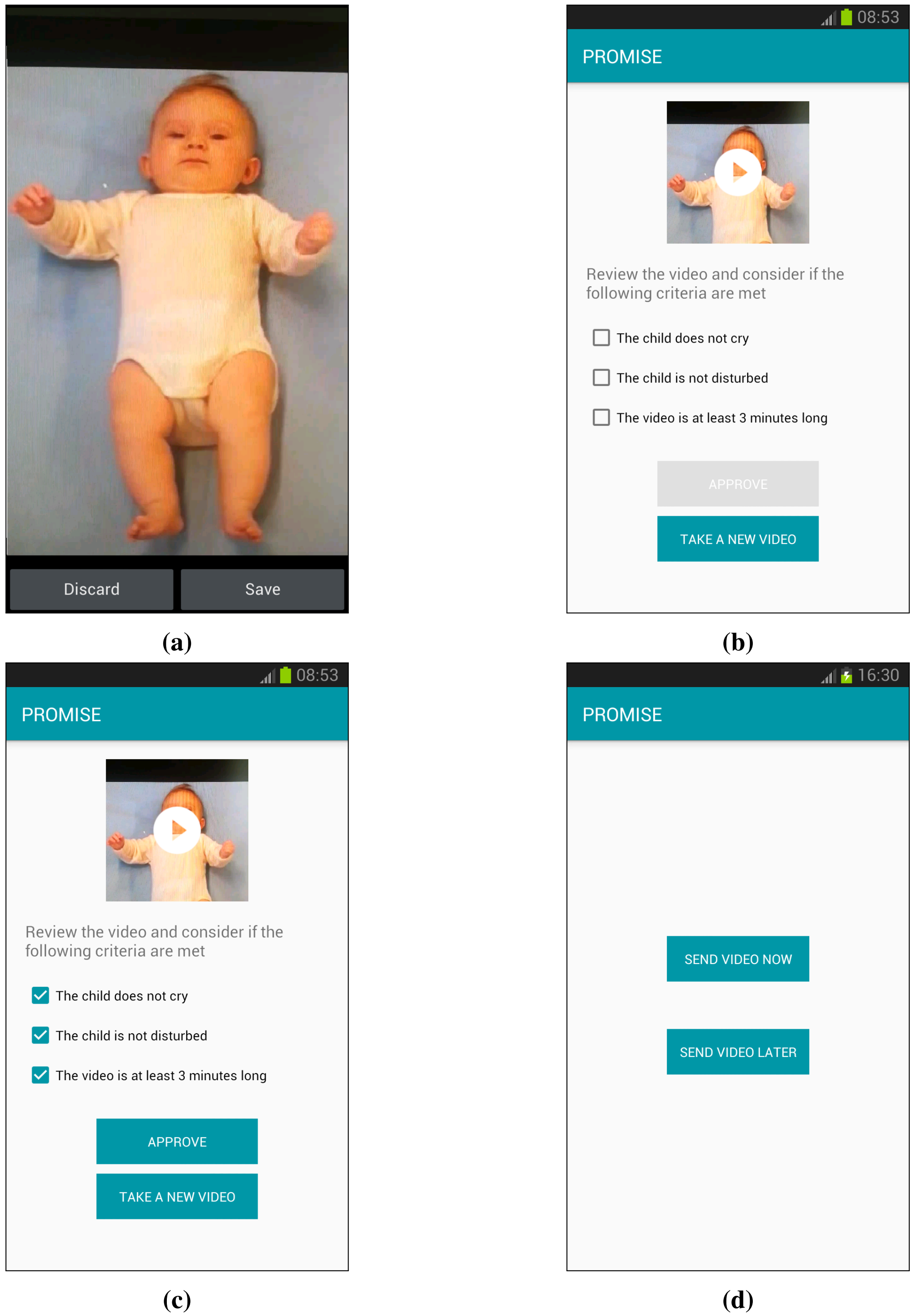
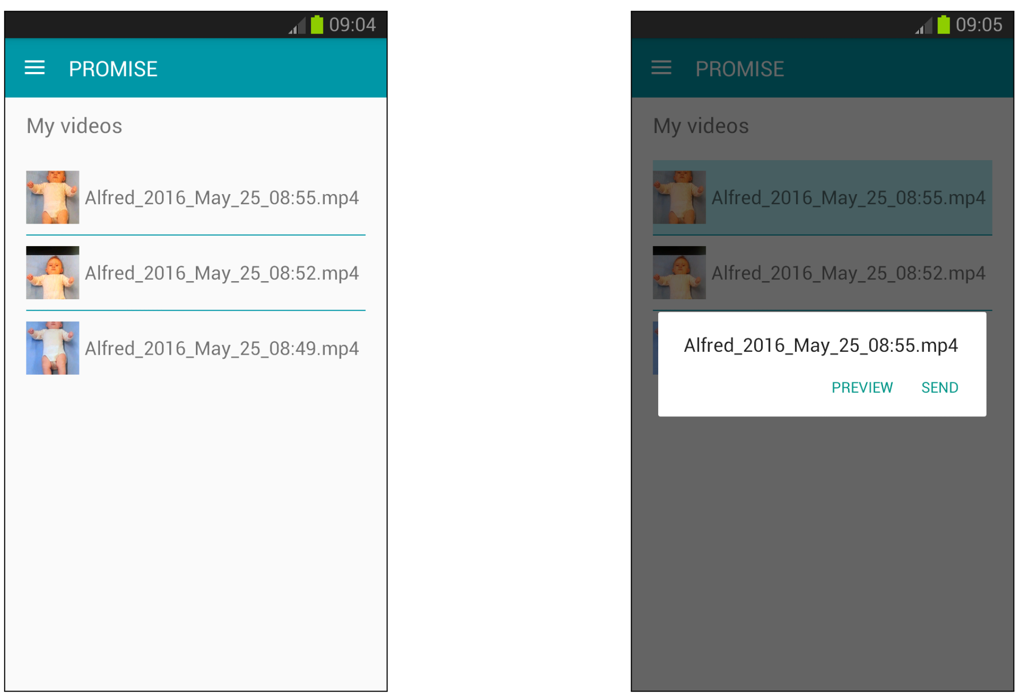


Figure 4.12: Part 2 of the "New video" procedure: Before returning to the app after recording a video, the user must save the video in the existing camera app (a). Then, review the video and consider if the video fulfils certain criteria (b). When all boxes are checked, the approve button becomes active (c). Finally, the user must choose to send the video now or later (d)

When all boxes are checked, the “Approve” button becomes active, see figure 4.12c. Finally, the user must choose to send the video now or later, see figure 4.12d.

List of Recorded Videos

A list of recorded videos of the current child appears when the user chooses “My videos” in the main menu, see figure 4.13a. The file names are fetched from the Video table in the SQL Database and a thumbnail picture of the video appears next to the title. When the user selects a video, a dialogue window with the two options “Preview” and “Send” appears, see figure 4.13b. If “Preview” is chosen the selected video will be shown in full screen. A menu will be visible if the user presses the screen and it will disappear again after a couple of seconds if not used, see figure 4.15b. The menu is made using the MediaController class from Support Library in Android Studio. This controller contains buttons for play, pause, rewind, fast forward and a progress slider and it handles synchronizing the controls with the state of the MediaPlayer (Android Developers, 2011b).



(a) **(b)**

Figure 4.13: The list of recorded videos of current child is shown in (a) and the dialogue window after selecting a child is shown in (b)

Send a Video

The prototype will send the video by email. The email intent is also one of the common intents in Android (Android Developers, 2013a). It has three different actions depending on what type of email to send. The prototype will send emails with an attachment and to one address only. For this, `ACTION_SEND` is the best option.

The most common email providers, such as Gmail and Outlook, are encrypting incoming and outgoing emails using Transport Layer Security (TLS) if they can (Long, 2014). To encrypt an email, both sides of the exchange must support encryption. 100 % of all emails sent between Gmail and Outlook, and within each provider, are encrypted (Google Transparency Report, 2016). Some of the smaller providers may not support encryption and emails sent through those providers will not be encrypted. To check the safety of an email provider, go to `https://g.co/saferemail`.

The process of sending a video in the prototype is presented in figure 4.14. When choosing "Send" in figure 4.14a, a list of existing email apps on the device is presented, see figure 4.14b. The user must choose the preferred app and enter the recipient address when the compose email page opens, see figure 4.14c. After the video is sent, it will appear in submitted videos under "Feedback", see figure 4.14d. Since there was no time to implement an `onActivityResult()` method in this activity, the video will be marked as submitted in the app, even though the email app operation is cancelled.

4.1.9 Information to the User

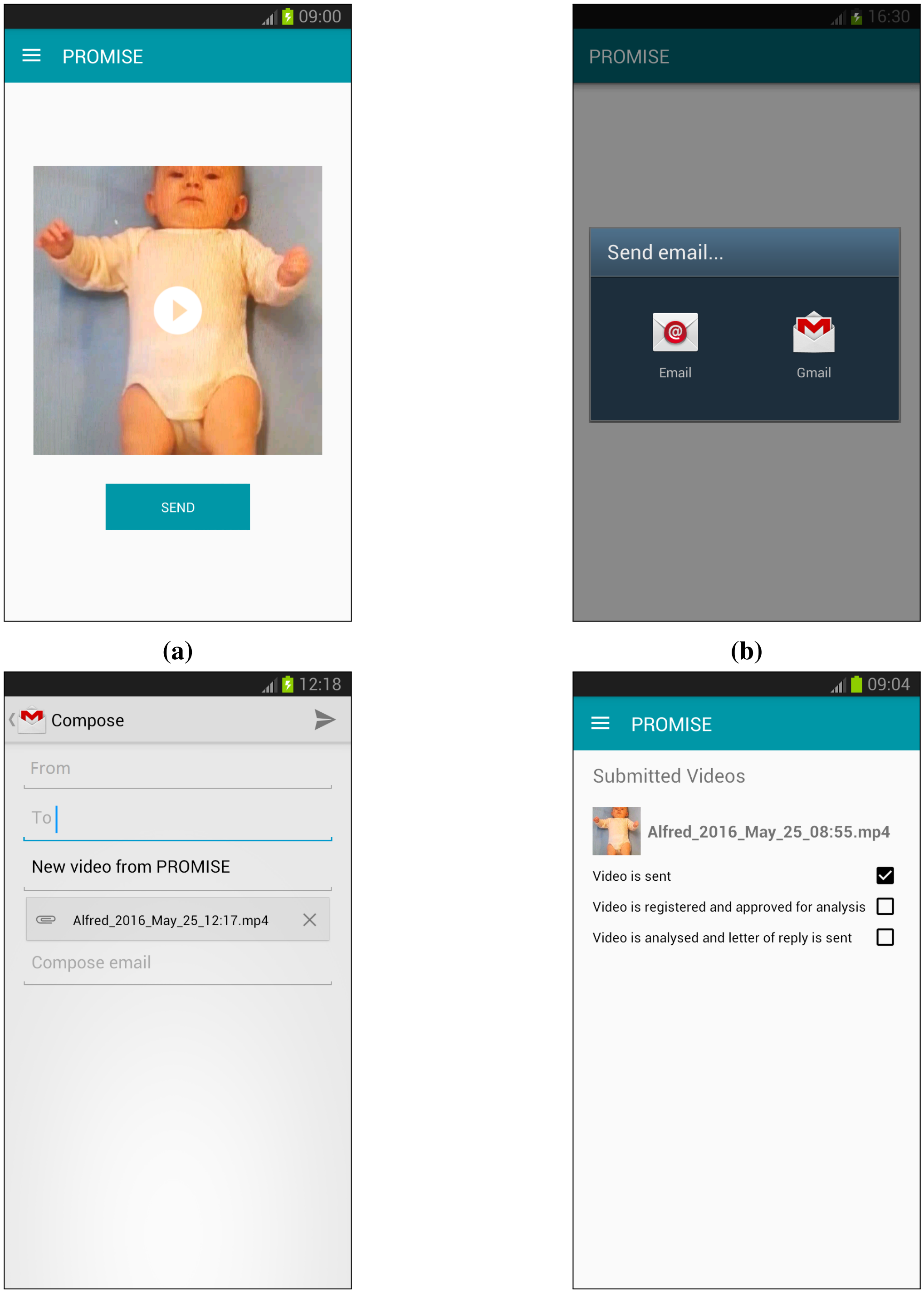
The pages in the prototype presented in this chapter are only illustrational pages as described earlier in Chapter 2.5.7. Not all pages are illustrated with screen shots in the report.

About and Contact Us

The "About" page is shown in two places in the prototype; in the main menu and at the start page of the app. The "Contact us" page is in the main menu. These pages are only presented with a heading because the text will be altered later when the research group have conducted the contents. Thus, screen shots are not included in the report.

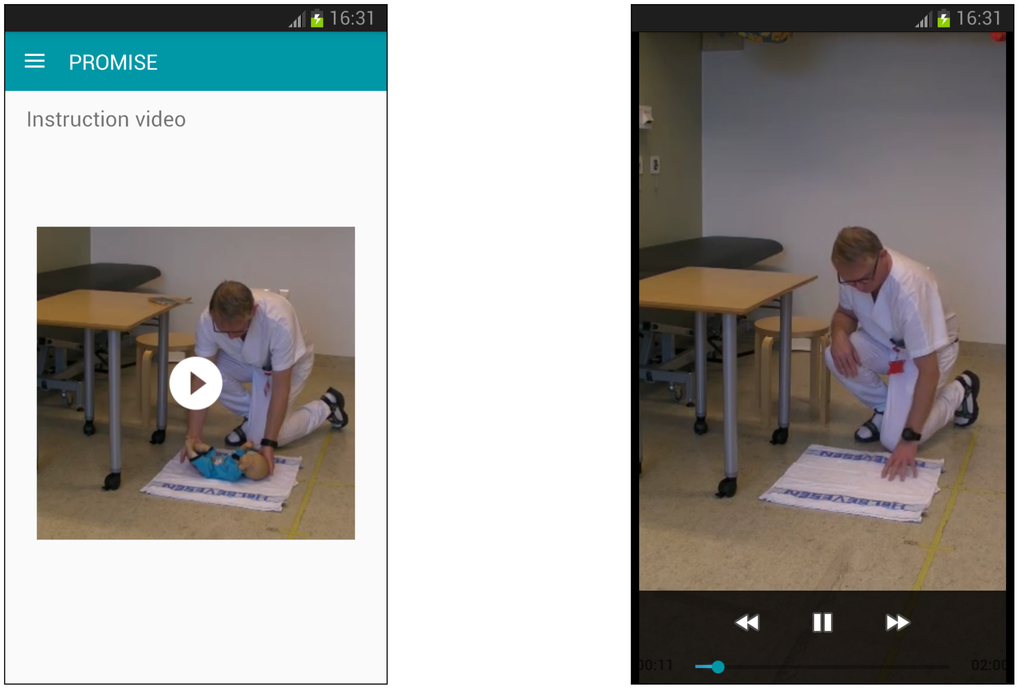
Feedback

The "Feedback" page is shown in figure 4.14d. It will appear after the user sends a video and if "Feedback" is chosen in the main menu. Only the "Video is sent" check box will be checked in the prototype because the feedback solution is not a part of this master's thesis as discussed earlier.



(a) **(b)** **(c)** **(d)**

Figure 4.14: When choosing "Send" in (a), a list of existing email apps is presented (b). The user must choose the relevant app and when the "Compose email" page opens (c), enter recipient address and press send. After the video is sent, it will appear in submitted videos under "Feedback" (d)



(a) **(b)**

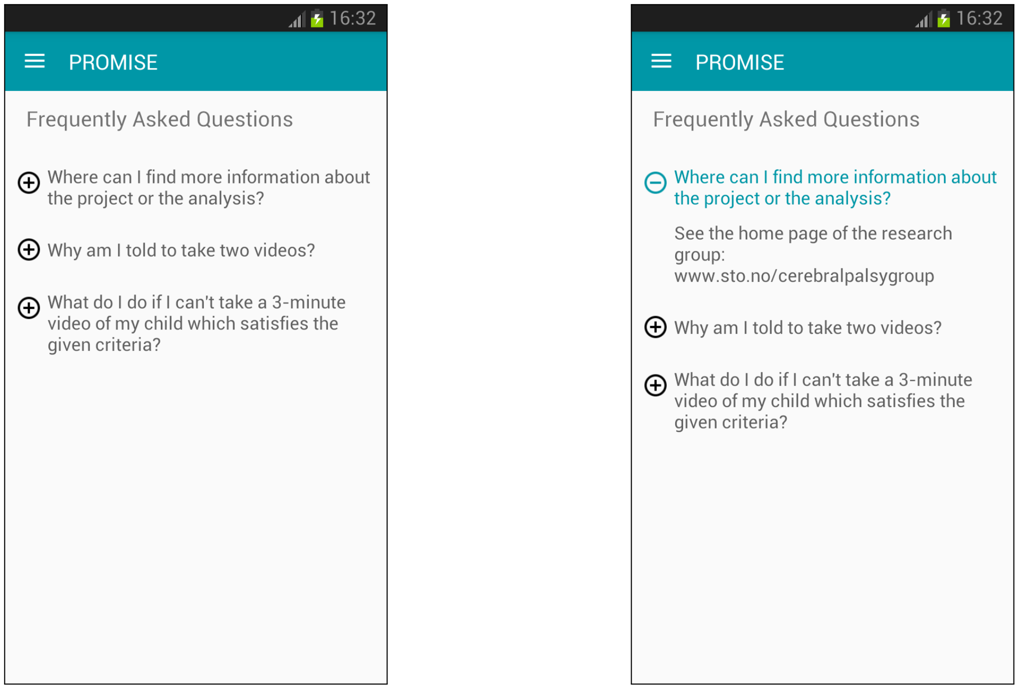
Figure 4.15: (a) shows a thumbnail of the instruction video with a play button. When the play button is pressed, the video opens in full screen (b) with a media controller menu.

Instruction Video

The “Instruction video” page is shown in two different actions in the main menu. It is displayed when the user chooses “Instruction video” and when the user chooses “New video” within the correct timeframe for a video. Both times a thumbnail image of the video is shown with a play button, see figure 4.15. The video in the prototype is only a short and simple demonstration video made for the prototype. If the user presses play, the video will be shown in full screen, and if the user presses the screen once, a media controller menu will appear, see figure 4.15b. The menu is described earlier in Chapter 4.1.8.

Frequently Asked Questions

When the user presses “Frequently asked questions” from the main menu, a page on how this may look like in the app appears, see figure 4.16. If the user presses the plus button next to each question, the corresponding answer will appear and the other answers will hide. At the same time, the button changes from plus to minus.



(a) **(b)**

Figure 4.16: The list of frequently asked questions is shown in (a). (b) shows the answer to question one after the plus sign has been pressed

4.1.10 Publication of the Prototype

For several reasons the prototype is not published at Google Play store. The prototype will only be tested by a handful of people, thus a simple download from a computer to the device is sufficient. Neither does the research group want any users to download the app and think that the app is active. A third reason is that they do not want to let the competition know the progress of this project. Therefore, the source code will only be attached to this report, see Appendix A. The final version of the app will be published in Google Play Store, and App store if the app is made for both Android and iOS. This will make it easy to access for many people, and possibly worldwide.

4.2 Prototype Testing

The prototype was tested by the client and a handful of other people. If there had been more time it would have been tested on parents as well. The feedback from the two tests described in Chapter 3.2 are presented in the following chapters. The comments are listed in tables along with information on whether or not the prototype has been altered. The feedback from the client was the most important feedback and is presented in a separate table. He summarizes the prototype's user experience in these words:

> "Clear and intuitive app. Seems thoroughly prepared and with professional and high quality. Easy to follow the workflow." (Lars Adde)

4.2.1 Usability Tests

The usability test was divided in two. The first part was with no guidance. The participants navigated through the app and wrote down their first impression. In the second part, the participants answered nine questions about the user interface of the prototype, such as text size, buttons, colours and so on. See the usability test form in Appendix C. They were also asked to think of important or less important improvements. The users did not have any objections to the current sizes or colours.

The two sub-tests resulted in some suggestions for improving the app. The comments from the client and the other test persons are presented in table 4.1 and 4.2, respectively. Some of the suggestions are further discussed in Chapter 5.4 or included in further development in Chapter 6.2. Except from these comments, the users were pleased with the first impression.

Changes made	Feedback
yes	Include a dialogue box for user to confirm deletion of a child profile
yes	The keyboard cannot be covering the text field where the user enters the child's username in "Edit child" page
yes	The icon next to "New video" in the main menu should be changed from a camera to a video icon
yes	The project logo (the blue child in figure 4.2) should be included on the first page of the app
no	The pause function in the camera should possibly be removed in further development of the app
no	The NTNU logo (see figure 4.2) is not visible on all screen formats. This should be fixed
no	The font of the timer in camera should be bigger (see figure 4.11d)
no	An explicit back button should be included in the app, in addition to the back button on the device
no	The solution for choosing a child and show menu in the app should be re-evaluated in further development
no	The file names in the list of recorded videos should perhaps include number of weeks after term date the video was recorded

Table 4.1: Result of the usability test from the client

Changes made	Feedback
yes	Change name of “Next Video” in the app to “When can I record the video?”
yes	Change the order of the items in the main menu, prioritised by the importance of the content
yes	The user should not be able to change the status of the check boxes in feedback
no	The Feedback function in the main menu should be renamed. It may be mistaken for a function where the user should give feedback about the app
no	The app should not log out the user automatically when closed. It should remember the logged in user for a certain amount of time
no	When pressing the back button in combination with the prototype being forwarded to the existing camera app, the prototype may shut down or place two frame layouts on top of each other.

Table 4.2: Result of the usability test from the other test persons

4.2.2 Technical Test

The functionality of the prototype was also tested, see the technical test form in Appendix B. This form was used multiple times during the development to test the app module separately and together. The notifications in the prototype was not implemented by the time the app was tested by the client and others, but it was tested later on by me. Most actions in the technical test form is working as planned in the final prototype. The marker indicating today’s date in the graphical countdown and a notification to record a video a week in advance, are the exceptions. The client was satisfied with the prototype and content with the progress of the project.

4.3 Parental Survey

The questionnaire described in Chapter 3.3 was answered by eight people. It was given to randomly chosen parents with premature infants born at St. Olavs Hospital. The parents were invited to answer separately to see if there were a difference between male and female, and to increase the participant group. The reference group is too small to make a statistical conclusion, but the answers were still evaluated.

As table 4.3 to 4.6 shows, there are no clear conclusions to whether the parents find this project meaningful or not. Also, there are no coherence between age and opinion, or gender and opinion.

Half of the participant group think this might be a good idea and the other half are sceptical or unsure. A comment area was also included in the survey and three people left comments which are very interesting. All three responded negatively to the idea and they explained why. They were afraid the video recording at home would replace the physiotherapist visits and concerned whether the video recording is as good as physical observation. It is important to emphasise that these parents did not get any information about the project before they participated in the survey and the information presented there was very limited. The result of this survey is discussed further in Chapter 5.3.

Below 20	20 - 29	30 - 39	Above 40
0	5	3	0

Table 4.3: Parental survey, result of question 1: The age of the participants

Female	Male
5	3

Table 4.4: Parental survey, result of question 2: The gender of the participants

Yes	No
8	0

Table 4.5: Parental survey, result of question 3: The number of participants where at least one of the parents have a smartphone

Don't have smartphone	Not desirable	Not too desirable	I don't know	Somewhat desirable	Very desirable
0	0	3	1	2	2

Table 4.6: Parental survey, result of question 4: The parents' thoughts about the idea of recording the video at home

Chapter 5

Discussion

This chapter contains discussions regarding the technology development within health care, and the ethical aspects of fully automated mobile diagnosis. It also contains discussions regarding the result of the parental survey and the improvement suggestions that were a result of testing the prototype.

5.1 Health and Technology

Technology is becoming increasingly important in the health sector, especially in aids that will make everyday life easier for people. There is already an app for "just about everything". There are diabetes diaries, sleep monitoring and assistance to quit smoking. Pregnant women can monitor the growth of their child with the Symfyse-Fundus app, which is also developed at NTNU and St. Olavs Hospital. There are also apps to help people remember to take medications and activity apps for children with disabilities, where daily training is made into a game.

People with chronic diseases can do their own monitoring with simple measuring devices synchronised with an app. This can lead to fewer visits to the doctor, more accurate medicating, and the doctors have more accurate data to work from.

More and more information about diseases and symptoms is available on-line. When people get sick these days, they often search on-line, before going to the doctor. If people diagnose them self with information from the internet, the risk of misdiagnosis is very high. At worst, patients who is in need of treatment postpone a visit to the doctor, and this can lead to serious health consequences. Otherwise, self-diagnostics can lead to hysteria. If the symptoms are interpreted wrong, people might think they are more ill than they are.

When it comes to diagnostic, it is difficult to assess the development. There are e.g. apps with algorithms that can assess whether a mole may be malignant or not. In such cases it is important to ask; can algorithms be better and more reliable than judgement? Software analysis of FMs, which may be used in this project, is a similar scenario. There is a lot of research and testing needed before we can rely completely on such methods. It is fair to believe that people think the combination of algorithms and physicians are the most reliable.

Another important aspect of health apps is: who owns the data in an app? Health information is some of the most sensitive information we have. It is very important that they are treated confidentially and according to applicable laws. Lately, we have heard in the news that sensitive information from fitness apps are sold by the owner firm and that the apps were collecting data about the user even when the app was inactive (Forbrukerrådet, 2016). This quote from Snowden makes people think about the importance of privacy:

> "Arguing that you don't care about the right to privacy because you have nothing to hide is no different than saying you don't care about free speech because you have nothing to say" (Edward Snowden, Reddit, May 21, 2015)

In Norway, there is a great deal of disagreement about whether the health apps must be approved by the government or not and if so, how this should be done (Solli and Eriksen, 2015; Åse, 2015). It would be unfortunate if an approved app turned out to be unsafe to use.

Several health care centres provide apps where patients can set up appointments with their primary doctor and get prescriptions, etc. (HelseRespons, 2015). It is important that these apps are safe to use and that the ownership of the data stored in the app belongs to the user and not the developer firm. When you set up a doctor's appointment, it is customary to say what it concerns and this can be very private information. This information could be used by commercial companies in targeted advertising or other unwanted events.

5.2 Mobile Diagnostics

In the future, fully automated mobile diagnosis of CP may be a possibility. This requires that the app utilizes an algorithm to analyse the video on the user's phone and provides a risk assessment right there and then. This would be a simple and cheap solution, but the ethical aspect of this is very difficult.

Say that the app can be downloaded throughout the world and all you need is a smartphone. In 2015, approximately 68 % of adults in advanced economies countries had a smartphone and 37 % in developing countries, which is an increase of 16 % in 2 years (Poushter, 2016). This number will probably increase. This means that many people will be able to conduct such an analysis, without necessarily being offered medical assistance, kinds of treatment or follow-up programs. All children are entitled to live adequate lives, even though they have disabilities. In developed countries the opportunity for support will be greater.

According to UNICEF, children with disabilities are almost 17 times more likely to be institutionalised compared to children who are not disabled (UNICEF, 2013). Although they have as much right to stay with their family, many of the children live in institutions, nursing homes or other residential institutions their whole life. Some countries even encourage parents to place their children in institutional care immediately after birth or as soon as they are diagnosed (UNICEF, 2013).

To stay confined and be robbed of rights has a bad impact on these children's lives and development (UNICEF, 2013). It is easy to imagine what might happen if parents themselves could diagnose their children by using an app, without knowing the children's rights and without being offered assistance. Therefore, such an app should be carefully controlled by the authorities in each country and not offered without mandatory medical assistance.

5.3 Result of Parental Survey

Parents with new-borns, and especially when born preterm, are very vulnerable. They will naturally be very protective of their children. Those who responded to this survey were in this position. They had received little information about the CP project, and the little information on the form was not enough to cover all the questions a parent might have about such a project. Therefore, it was expected that many would be sceptical of the idea to film their child at home.

The idea for the project is not to replace meetings with a physiotherapist. By recording the video at home, the physiotherapists will not have to spend the entire session trying to get a good video recording of the child. The physiotherapist can instead spend time assessing the child.

Unfortunately, few people responded to the survey. Therefore, the answers cannot be used to confirm or set aside that the parents welcome this idea. However, it is interesting to realize how this idea may seem unpleasant for the parents. This can improve the preparations before initiating the research project. It was the comments of those who were sceptical towards the idea that gave the most informative results.

It might be a good idea to perform a bigger survey where the parents get more information to possibly get a more representative feedback. It is important that the parents are well informed and feel protected before executing this research project.

5.4 Result of Testing the Prototype

When the client tested the prototype for the first time, it was mostly clear and intuitive. However, a couple of things were difficult to grasp in the beginning. One of the things that was difficult to understand was that you had to register a child at first login before you got access to the main menu.

This is implemented as planned in the preliminary project, but the question is whether there is a better solution for this. The client said that this was fine after understanding the concept, but he was confused by the layout when he first tried to locate the menu. When the project involves the app developers, this should be discussed. One solution could be to enter the home page of the app (see Figure 4.4a) after login and that a child must be chosen from a list when entering the pages; “When can I record the video?”, “New video”, “My video” and “Feedback”. This again can be very bothersome. It is important to find a good balance between simple and intuitive.

The client believes it will be an advantage for physiotherapists using the app if the number of weeks after term date the video was recorded, was included in the file name or listed beside the file name in “My Videos”. This may be convenient for a physiotherapist, but can be confusing for parents who do not have any relation to these number of weeks. Maybe they will think it is the child’s age and that there is something wrong with the app. This must be considered in further development of the app. It is important that both groups of users are taken into account.

Another subject that came up during the testing was the pause function in the camera. The client thinks that when the user is recording a video, it should not be possible to pause the recording. For the clinical assessment this may be possible to handle, but for software analysis it will not. The quality of the recording,however may not be good if the video is divided in 2-3 sequences. It may imply that the child is disturbed during the recording, or restless, hungry or sleepy. This will complicate the assessment regardless of the number of sequences in the video. Therefore, a video in only one sequence might be a quality assurance to receive a good recording for the assessment.

Chapter 6

Conclusions and Further Development

The main objective of this thesis was to develop an Android app for diagnosis of cerebral palsy. This chapter presents a review of the requirement specifications from Chapter 1 to evaluate whether or not the prototype fulfils its requirements. Suggestions for further development of the app and further work for the project are also given. Finally, an overall conclusion regarding the process of this master's thesis is presented.

6.1 Prototype Requirements

Register a User Profile

The user is able to register new profiles and everything is working as planned. The username is unique and the password must be re-entered to avoid misspelling. On login the password and username are checked against the SQL Database where the information is stored. If a valid username and password is entered, the user is logged in to the prototype.

Register a Child with a QR Code

The user is able to register a child by scanning a QR code and to enter a username and the child's term date. The QR-code contains a child ID, hospital code, country code and an API key. The API key will only be used in the final app. The child's details are saved in a SQL Database on the device, and the information can be edited or deleted.

Record a Video in the Right Timeframe

The prototype calculates the correct dates for the user to record a video of the child from the term date. Only if the user is within the timeframe for a recording, the option to record a video is visible.

The prototype uses the existing camera app on the device, because there was not enough time to implement a camera function within the app itself. This is sufficient for the prototype, but it is not optimal. The default font of the count for the length of the video is too small on the screen for the user to see from a distance, and the camera screen is lacking a frame to target the child.

Notifications on the Device

The prototype gives the user a notification when the user is within the timeframe for a recording and every day at 12:00 until a video of the child is sent. There was not enough time to implement a notification to notify the user a week in advance.

Send a Video

The prototype sends the video using an existing email app on the device, with the video as an attachment. The size of the video is limited to 25 MB, which corresponds to about 10-15 seconds of video on the test phone. This is not optimal, but this solution will be different when the video is sent to the hospital server in the final app.

6.2 Further Development

This section consists of the improvements needed for the prototype, the suggestions for further development of the final app and a recommendation for further work in the project.

6.2.1 Prototype

The prototype contains a few bugs that needs to be fixed. They are as follows:

- Scanning a QR code with a wrong JSON-format makes the prototype shut down. This bug is odd because the action to interpret the QR code is in a Java try block. The Java try block should throw an exception if unable to finish successfully, not cause the prototype to crash
- When pressing the back button in combination with the prototype being forwarded to the existing camera app, the prototype may shut down or place two frame layouts on top of each other. This problem will disappear when the camera is implemented in the app itself, but to improve the prototype this must be fixed
- The NTNU logo (see figure 4.2) is not visible on all devices. This is odd, because the bug does not seem to have anything to do with the screen format

- An `onActivityResult()` method should be implemented in the “Send video” activity. This will ensure that the video is only marked as submitted if the email action was successfully executed

6.2.2 The Final App

As discussed in Chapter 4.2.1 suggestions to improve the app were given. Some of them should be discussed further with the professional app developers, others should be implemented to improve the user experience or the functionality of the app.

These are the suggestions that should be implemented:

- An explicit back button should be included in the app, in addition to the back button on the device
- The Feedback function in the main menu should be renamed. It may be mistaken for a function where the user should give feedback about the app
- The app should not log out the user automatically when closed. It should remember the logged in user for a certain amount of time
- The screen orientation in the prototype is only suitable for vertical screen orientation. Both orientations should be adaptive
- The app should have its own camera function with the specified requirements implemented. For instance there should be a frame on the screen and the font of the timer or countdown should be bigger than default

These suggestions should be further discussed:

- There should perhaps be a profile manager to give users a password hint or a new password, and a way to delete old profiles
- The file names in the list of recorded videos should perhaps include number of weeks after term date the video was recorded
- The solution for choosing a child and show menu in the app should be re-evaluated in further development
- The pause function in the camera should possibly be removed in further development of the app

6.2.3 Further Work

Testing of the Prototype

When the user information is written by experts, and the instruction video is designed by professionals, this should be implemented in the prototype. Then, the prototype should be tested on health personnel and parents. This testing will give valuable insight into what should be included in the final app.

Professional App Developers

The research group has initiated contact with professional app developers. The requirement specifications, along with a mock-up of the app developed in the preliminary project have been sent to four different companies. The client is awaiting their response preferably before summer 2016. When they respond, the suggestions for improvements mentioned in further development in Chapter 6.2.2 should be discussed. They will probably have suggestions of their own. It is important that the different solutions are tested on relevant users throughout the process to optimise the app.

Extended Parental Survey

An extended parental survey should be executed as soon as possible. Perhaps a wider selection of parents should be involved, and the participants should get more information about the project before answering the questionnaire. In that way, they can see how the app correlates with the general follow-up program from the hospital. Because of the limited relevant test group, the survey should be conducted for a longer period of time. It might also be a good idea to include parents at collaborating hospitals to increase the participant group.

6.3 Final Conclusion

In this master's thesis the requirement specifications for the app has been presented. They include changes made after the preliminary report was completed, and decisions made in this project.

In the beginning of the project, the development platform was decided. The client did not have any preferences other than the requirements concerning the camera function. A web app was inadequate and native app was chosen over hybrid due to my experience with Android and Java programming, and because of the improved user experience it provides.

The prototype was developed in Android Studio. This tool was appropriate because of the simple user interface and the well equipped and documented Support Library. Java was the programming language and the layouts were designed in XML.

When the prototype was nearly finished, it was tested by the client and a handful of others. I made two tests for them to conduct; a usability test and a technical test. The feedback from the participants was positive, and the client was content with the result. The participants came up with some suggestions for further development of the app. Some of the improvements were implemented in the prototype, others are suggested as a part of further work. The notification feature on the device was not completed by the time the prototype was tested, but I have implemented and tested it later on.

The features in the app meets most of the requirement specifications. Some changes have been made to the notifications, as described in Chapter 4.1.5 and the camera is not implemented as planned. Instead, the prototype uses the existing camera app on the device. Also, the graphical countdown to the video recording is lacking the active marker indicating today's date.

The QR code feature is an important security aspect of the prototype. It makes sure that only persons with a valid QR code can get access to the prototype. It also guarantees correctly registration of the children and ensures that the assessment results are linked to the right child. This quality is essential for the safety of personal information.

A questionnaire was also made in this thesis. It involves evaluating the parent's thoughts about recording the video at home. Unfortunately, very few people responded to the survey, and no statistical conclusion could be made. It did however give the research group valuable insight in the parent's vulnerability and information on how to approach the parents when initialising the research project.

This thesis has confirmed that the idea behind the project is feasible and that this app can be a helpful tool for diagnostics of children with CP. It can make the recordings less stressful for the children, and will be a particularly useful tool for those who live far from hospitals. Health personnel will also benefit by using the app, since this method is much easier than the equipment package they use today.

The prototype and the report will be presented to the research group. The results of this project will be used in the further work of assessing children's risk of developing CP.

Bibliography

Adde, L., Helbostad, J., Jensenius, A. R., Taraldsen, G., Grunewaldt, K. H., Støen, R., 2010. 'Early prediction of cerebral palsy by computer-based video analysis of general movements: a feasibility study'. *Developmental Medicine & Child Neurology*, 52 (8), pp. 773–778.

Adde, L., Helbostad, J. L., Jensenius, A. R., Taraldsen, G., Støen, R., 2009. 'Using computer-based video analysis in the study of fidgety movements'. *Early Human Development* , 85 (9), pp. 541–547.

Andersen, G., 2016a. *Diagnostisering og utredning*. Available at: `http://www.cp.no/om-cerebral-parese/diagnostisering-og-utredning/`, (Accessed: 15 April 2016).

Andersen, G., 2016b. *Hva er CP?* Available at: `http://www.cp.no/om-cerebral-parese/hva-er-cp/`, (Accessed: 15 April 2016).

Android Developers, 2011a. *AlarmManager*. Available at: `https://developer.android.com/reference/android/app/AlarmManager.html`, (Accessed: 23 May 2016).

Android Developers, 2011b. *MediaController*. Available at: `https://developer.android.com/reference/android/widget/MediaController.html`, (Accessed: 06 June 2016).

Android Developers, 2011c. *Services*. Available at: `https://developer.android.com/guide/components/services.html`, (Accessed: 23 May 2016).

Android Developers, 2012a. *Pure Android*. Available at: `https://developer.android.com/design/patterns/pure-android.html`, (Accessed: 06 June 2016).

Android Developers, 2012b. *System Permissions*. Available at: `https://developer.android.com/guide/topics/security/permissions.html`, (Accessed: 21 May 2016).

Android Developers, 2013a. *Common Intents*. Available at: https://developer.android.com/guide/components/intents-common.html, (Accessed: 25 May 2016).

Android Developers, 2013b. *Scheduling Repeating Alarms*. Available at: https://developer.android.com/training/scheduling/alarms.html, (Accessed: 23 May 2016).

Android Developers, 2013c. *WakefulBroadcastReceiver*. Available at: https://developer.android.com/reference/android/support/v4/content/WakefulBroadcastReceiver.html, (Accessed: 02 June 2016).

Android Developers, 2015a. *Android Studio Overview*. Available at: http://developer.android.com/tools/studio/index.html, (Accessed: 11 March 2016).

Android Developers, 2015b. *Controlling the Camera*. Available at: http://developer.android.com/training/camera/cameradirect.html, (Accessed: 19 April 2016).

Android Developers, 2015c. *Recording Videos Simply*. Available at: http://developer.android.com/training/camera/videobasics.html, (Accessed: 19 April 2016).

Android Developers, 2016a. *Android Studio Release Notes*. Available at: http://developer.android.com/tools/revisions/studio.html#Revisions, (Accessed: 18 April 2016).

Android Developers, 2016b. *Platform Versions*. Available at: https://developer.android.com/about/dashboards/index.html, (Accessed: 21 May 2016).

Android Developers, 2016c. *Storage Options*. Available at: https://developer.android.com/guide/topics/data/data-storage.html, (Accessed: 21 May 2016).

Åse, J., 2015. *"Svanemerking" av helseapper*. Available at: https://tekst.joaa.se/svanemerking-av-helseapper-1087a9fe5968#.5qyr1mm3m, (Accessed: 04 June 2016).

Barry, B., 2011. *Java For Dummies*, 5th Edition. Hoboken: Wiley.

Bosak, J., 2007. *The Birth of XML*. Available at: http://lim.univ-reunion.fr/staff/fred/Enseignement/XML/data/The-Birth-of-XML.pdf, (Accessed: 19 April 2016).

Buen, C. Ø., 2015. *Utvikling av krav til en mobilapplikasjon for diagnostisering av cerebral parese*. Specialisation Thesis. Department of Engineering Cybernetics, Norwegian University of Science and Technology.

CEBRA, 2015a. *About CEBRA*. Available at: `https://www.ntnu.edu/lbk/about-cebra`, (Accessed: 05 April 2016).

CEBRA, 2015b. *Cerebral Palsy (CP) Group*. Available at: `https://www.ntnu.edu/lbk/cp`, (Accessed: 05 April 2016).

Cerebral Palsy Alliance, 2014. *2014 Grant Recipient Researchers*. Available at: `https://research.cerebralpalsy.org.au/our-work/our-researcher-program/2014-grant-recipient-researchers/`, (Accessed: 20 April 2016).

Cline, A., 2015. *7 UI design tips when developing mobile apps*. Available at: `http://blog.fluidui.com/7-ui-design-tips-when-developing-mobile-apps/`, (Accessed: 06 June 2016).

Cowan, J., Fang, A., Grosso, P., Lanz, K., Marcy, G., Thompson, H., Tobin, R., Veillard, D., Walshand, N., Yergeau, F., 2008. *Extensible Markup Language (XML) 1.0 (Fifth Edition)*. Available at: `https://www.w3.org/TR/REC-xml/`, (Accessed: 19 April 2016).

Eason, J., 2014. *Android Studio 1.0*. Available at: `http://android-developers.blogspot.no/2014/12/android-studio-10.html`, (Accessed: 18 April 2016).

Einspieler, C., Prechtl, H. F. R., 2005. 'Prechtl's assessment of general movements: a diagnostic tool for the functional assessment of the young nervous system'. *Mental Retardation & Developmental Disabilities Research Reviews*, 11 (1), pp. 61–67.

Einspieler, C., Prechtl, H. F. R., Ferrari, F., Cioni, G., Bos, A. F., 1997. The qualitative assessment of general movements in preterm, term and young infants - review of the methodology. *Early Human Develpment* , 50 (1), pp. 47–60.

Forbrukerrådet, 2016. *Runkeeper sporer brukerne nr appen ikke er i bruk*. Available at: `http://www.forbrukerradet.no/vi-mener/2015/fpa-digital-2015/runkeeper-sporer-brukerne-nar-appen-ikke-er-i-bruk/`, (Accessed: 04 June 2016).

GitHub Help, 2013. *GitHub Glossary*. Available at: `https://help.github.com/articles/github-glossary/`, (Accessed: 25 May 2016).

Gjerstad, L., 2016. *Cerebral Parese. I Store medisinske leksikon*. Available at: `https://sml.snl.no/cerebral_parese`, (Accessed: 15 April 2016).

Google Transparency Report, 2016. *Email encryption in transit*. Available at: `https://www.google.com/transparencyreport/saferemail/?hl=en-GB`, (Accessed: 25 May 2016).

HelseRespons, 2015. *HelseRespons*. Available at: `http://helserespons.no/`, (Accessed: 04 June 2016).

Hollung, S., Andersen, G., Wiik, R., Bakken, I., Vik, T., 2015. 'What is the prevalence of cerebral palsy in Norway?'. *Developmental Medicine & Child Neurology*, 57, pp. 91–92.

JSON.org, 2003. *Introducing JSON*. Available at: `http://www.json.org/`, (Accessed: 18 April 2016).

Long, B., 2014. *Transparency Report: Protecting emails as they travel across the web*. Available at: `https://googleblog.blogspot.no/2014/06/transparency-report-protecting-emails.html`, (Accessed: 25 May 2016).

Marcroft, C., Khan, A., Embleton, N. D., Trenell, M., Plötz, T., 2014. 'Movement Recognition Technology as a Method of Assessing Spontaneous General Movements in High Risk Infants'. *Frontiers in neurology* , 5, pp. 284.

Marszałek, K., 2016. *Native vs hybrid - Demystifying the technology dilemma*. Available at: `http://howwedostartups.com/articles/Native-vs-Hybrid`, (Accessed: 11 March 2016).

Owen, S., 2016. *Official ZXing ("Zebra Crossing") project home*. Available at: `https://github.com/zxing/zxing`, (Accessed: 23 May 2016).

Poushter, J., 2016. *Smartphone Ownership and Internet Usage Continues to Climb in Emerging Economies*. Available at: `http://www.pewglobal.org/2016/02/22/smartphone-ownership-and-internet-usage-continues-to-climb-in-emerging-economies/` , (Accessed: 05 June 2016).

Rahmati, H., Dragon, R., Aamo, O. M., Adde, L., Stavdahl, Ø., Gool, L. V., 2015. 'Weakly supervised motion segmentation with particle matching'. *Computer Vision and Image Understanding* , 140, pp. 3042.

Rossen, E., 2014. *Java: IT. I Store norske leksikon*. Available at: `https://snl.no/Java%2FIT`, (Accessed: 12 March 2016).

Rouse, M., 2008. *Object-oriented programming (OOP)*. Available at: http://searchsoa.techtarget.com/definition/object-oriented-programming, (Accessed: 19 April 2016).

Solheim, S. O., 2016. *Spastisitetsbehandling*. Available at: http://www.cp.no/om-cerebral-parese/behandling/, (Accessed: 15 April 2016).

Solli, B., Eriksen, M., 2015. *Staten skal godkjenne helseapper*. Available at: https://www.nrk.no/trondelag/staten-skal-godkjenne-helseapper-1.12682141, (Accessed: 04 June 2016).

Spittle, A., 2015. *The feasibility of smart-phones in the early detection of cerebral palsy*. Available at: http://healthsciences.unimelb.edu.au/research2/physiotherapy-research/children/smart-phones-cerebral-palsy, (Accessed: 20 April 2016).

Statistics Norway, 2016. *Cerebral parese*. Available at: http://www.ssb.no/fodte/, (Accessed: 04 April 2016).

Thornsby, J., 2015. *The Ins and Outs of Gradle*. Available at: http://code.tutsplus.com/tutorials/the-ins-and-outs-of-gradle--cms-22978, (Accessed: 19 April 2016).

UNICEF, 2013. *Children and Young People with Disabilities: Fact Sheet*. Available at: http://www.unicef.org/disabilities/files/Factsheet_A5__Web_NEW.pdf, (Accessed: 05 June 2016).

Valle, S. C., Støen, R., Sæther, R., Jensenius, A. R., Adde, L., 2015. 'Test-retest reliability of computer-based video analysis of general movements in healthy term-born infants'. *Early Human Development*, 91, pp. 555–558.

Williams, J. M., 2015. *9 essential UI design tips*. Available at: https://webflow.com/blog/9-essential-user-interface-ui-design-tips, (Accessed: 06 June 2016).

World Cerebral Palsy Day, 2015. *Christa Einspieler*. Available at: https://worldcpday.org/our-campaign/medical-therapeutic/christa-einspieler/, (Accessed: 20 April 2016).

Appendix

Appendix A-D is included in the Appendix.ZIP file

Appendix A Prototype Source Code (Android Studio Project)

Appendix B Technical Test Form (in Norwegian)

Appendix C Usability Test Form (in Norwegian)

Appendix D Parental Survey (in Norwegian)