



A Mobile Prototype for Assisted Analysis in Physiotherapeutic Rehabilitation

Bachelor, Computer Science

Department of Computing, Mathematics and Physics

Faculty of Engineering and Science

Submission date: 01.06.2019

Number of words: 10208

Anja Fonn Salen

Malik Aasen

Ådne Siqveland

I confirm that the submitted work is independently produced and that all references and sources are clearly stated according to *Forskrift om studier og eksamen ved Høgskulen på Vestlandet, § 9-1.*

BACHELOR PROJECT TITLE PAGE

<i>Report title:</i> A Mobile Prototype for Assisted Analysis in Physiotherapeutic Rehabilitation	<i>Date:</i> 01.06.2019
<i>Author(s):</i> Anja Fonn Salen, Malik Aasen, Ådne Siqveland	<i>Number of pages:</i> 48
	<i>Number of appendix pages:</i> 4
<i>Education programme:</i> Computer science	<i>Number of CDs/DVDs:</i> 0
<i>Contact person at HVL:</i> Ilona Heldal	<i>Classification:</i> None
<i>Remarks:</i>	

<i>External company/contact:</i> Eirik Grøneng	<i>External contact reference:</i>
<i>External contact person:</i> Eirik Grøneng	<i>Phone:</i> +47 915 13 900

<p><i>Summary:</i></p> <p>This project involves creating an application that will assist the treatment of patients in physiotherapy. This solution consists of an Android mobile application that will be able to help physiotherapists assess a patient's performance of physical exercises, as well as make a visual overview of a patient's progress.</p> <p>This is done by developing an application that shows reference exercises and exercises from the patient. By building up a reference library where the patient can compare their own exercises with a reference exercise, and by having the opportunity to see their own execution of exercises develop over time, the application can contribute to more effective training.</p> <p>This report will describe the process of developing such an application.</p>
--

Keywords:

Android	Mobile application	Java
---------	--------------------	------

Western Norway University Of Applied Sciences, Faculty of Engineering and Science

Postal address: Postboks 7030, 5020 BERGEN Address: Inndalsveien 28, Bergen

Phone. 55 58 75 00

Fax 55 58 77 90

 E-mail: post@hvl.no

 Home page: <http://www.hvl.no>

PREFACE

This report documents the development surrounding our bachelor thesis “A Mobile Prototype for Assisted Analysis in Physiotherapeutic Rehabilitation”. The project was done by Anja Fonn Salen, Malik Aasen and Ådne Siqveland.

We would like to thank the project owner, Eirik Grøneng, for an exciting bachelor project. He has been very helpful throughout the project by giving us valuable feedback and information on further improvements.

We would also like to thank the media lab at HVL for helping us with the recording of videos for this prototype.

Lastly, we would like to thank our contact person at HVL, Ilona Heldal, for giving us guidance on writing and structuring this bachelor thesis.

TABLE OF CONTENT

PREFACE.....	III
1 INTRODUCTION.....	1
1.1 GOAL.....	1
1.2 CONTEXT	2
1.3 LIMITATIONS.....	2
1.4 RESOURCES	3
1.5 ORGANIZATION OF THE REPORT	3
2 PROJECT DESCRIPTION	4
2.1 PRACTICAL BACKGROUND.....	4
2.2 PROJECT OWNER	4
2.3 PREVIOUS WORK.....	4
2.3.1 <i>Existing solutions</i>	5
2.3.2 <i>Assisting literature</i>	5
2.4 INITIAL REQUIREMENTS SPECIFICATION	5
2.5 INITIAL SOLUTION IDEA	6
2.5.1 <i>Purpose and functionality</i>	6
2.5.2 <i>Versioning</i>	7
2.5.3 <i>Design and architecture</i>	7
3 PROJECT DESIGN	9
3.1 POSSIBLE APPROACHES.....	9
3.1.1 <i>Cross-Platform Mobile Application</i>	9
3.1.2 <i>Native Android Application</i>	9
3.1.3 <i>Discussion of alternative approaches</i>	9
3.2 SPECIFICATION	10
3.3 SELECTION OF TOOLS AND PROGRAMMING LANGUAGES	10
3.3.1 <i>Tools</i>	10
3.3.2 <i>Languages and libraries</i>	11
3.4 PROJECT DEVELOPMENT METHOD	12
3.4.1 <i>Development method</i>	12
3.4.2 <i>Project plan</i>	12
3.4.3 <i>Risk management</i>	12



3.5	EVALUATION METHOD.....	12
4	DETAILED DESIGN.....	14
4.1	ARCHITECTURE.....	14
4.1.1	<i>Database architecture.....</i>	<i>14</i>
4.1.2	<i>Storage Architecture.....</i>	<i>15</i>
4.1.3	<i>Application architecture.....</i>	<i>16</i>
4.2	KEY FEATURE IMPLEMENTATION.....	18
4.2.1	<i>Video codec and compression.....</i>	<i>18</i>
4.2.2	<i>Video player.....</i>	<i>19</i>
4.2.3	<i>Reference videos.....</i>	<i>19</i>
4.3	USE CASES.....	20
4.3.1	<i>Use case 1: Record and compare execution.....</i>	<i>20</i>
4.3.2	<i>Use case 2: Show an existing execution.....</i>	<i>21</i>
4.3.3	<i>Use case 3: Create a new category.....</i>	<i>22</i>
4.4	USER INTERFACE DESIGN.....	23
4.4.1	<i>Android UI components.....</i>	<i>23</i>
4.4.2	<i>UI layout.....</i>	<i>24</i>
4.4.3	<i>UI Language.....</i>	<i>29</i>
5	EVALUATION.....	30
5.1	EVALUATION METHOD.....	30
5.2	EVALUATION FROM EIRIK GRØNENG.....	30
5.3	EVALUATION FROM STUDENTS.....	31
6	DISCUSSION.....	33
6.1	CLOUD STORAGE - SOLUTION.....	33
6.2	DEVELOPMENT APPROACH AND PROGRAMMING LANGUAGE.....	33
6.3	DEVELOPMENT METHOD AND ALTERNATIVE MODELS.....	33
6.4	UI.....	34
6.5	CHOOSING CORE FUNCTIONALITIES.....	34
6.6	FURTHER WORK AND IMPROVEMENT.....	35
7	CONCLUSIONS.....	36
7.1	REACHED GOALS.....	36
7.3	DIFFERENT USES.....	36



8	LITERATURE/REFERENCES	37
9	APPENDIX	39
9.1	APPENDIX A: INITIAL APP DESIGN SKETCH, MADE BY PROJECT OWNER.....	39
9.2	APPENDIX B: PROJECT PLAN	40
9.3	APPENDIX C: RISK ANALYSIS.....	42

1 INTRODUCTION

To perform physical exercises correctly, either for rehabilitation or to do some sport activities more efficiently, can be vital. For physiotherapeutic rehabilitation, the exercises often need to be practiced regularly over a longer period, and during this period the responsible physiotherapist may not always be present to ensure one is doing the exercise correctly. Since there are many promises from technologies to better assist training, this work focuses on designing and developing a mobile application that helps therapists assist their patients better, and patients to perform the exercises that helps their rehabilitation or improved conditions better. There are currently no viable solutions in place that allows a physiotherapist to give precise feedback to a patient performing an exercise. Such feedback may include incorrect angle of body positioning or the patient putting emphasis on the wrong muscle. There are also no easy ways to track his patients' progress from their last appointment.

In physiotherapy, the rehabilitation process is defined as a process designed to minimize the loss associated with acute injury or chronic disease, to promote recovery, and to maximize functional capacity, fitness and performance (Physiopedia, 2018). This process is achieved by attending appointments at a physiotherapist.

1.1 Goal

The goal of this project was to develop a mobile application that will allow a physiotherapist to give tailored guidance on different physical exercises to his patients. The guidance will be in the form of a description and a reference video, as well as giving the opportunity to track progress in each exercise.

The application will allow the physiotherapist to track a patient's progress by recording a video of him/her performing an exercise, and then compare it to a previous recording of him/her performing the same exercise.

The application needs to be easy to use and should not require any previous experience in physiotherapy. It is also important the application is supported by the project owner's device.

The reference videos are described in Chapter 4.2.3, and information about the tracking progress process in Chapter 2.5.1.1. We discuss further details of the chosen application platform in Chapter 3.1.

1.2 Context

The project owner, Eirik Grøneng, is a physiotherapist that has for a long time wished for a custom application to better assist his patients. When a patient has finished an appointment and has been assigned some exercises to perform, it is important that they perform these with correct form. Correct form implies that the patient is following the guidelines and technique described by the physiotherapist. Performing an exercise with incorrect form could potentially worsen the very problem that the patient seeks to solve with physiotherapy. Eirik Grøneng wants to reduce the risk of this happening by utilizing a tool that allows him to give guidance tailored for each patient.

If the patient is seeking help with a problem that requires a longer rehabilitation process, it can, in some cases, be hard to track progress from one appointment to the next appointment. This requested tool will allow the physiotherapist to record the patient performing an exercise and store this footage. The physiotherapist can then alongside the patient review and compare the footage to a reference video or previous recordings to identify and track the progress during a given time period.

Eirik Grøneng came up with the idea for the application, and we later discussed the functionalities and planned a design for it together. We have been given some creative freedom when it comes to the design of the application as long as the core functionalities are present.

1.3 Limitations

The biggest limitation we foresee is our previous knowledge in application development in general. The project was originally thought to be an Android and iOS application, however there will not be enough time to develop a cross-platform application since the project group would have to start from scratch. The project group's limited knowledge in Android app development, is also a small limitation. At the start of this project, two of the three members of the group had no previous experience with mobile app development for Android.

The scope of the application is also quite large, and while time itself is not a limitation, the project might be too large to complete within a three-month timeframe.

There could be other features of the application that we will not be able to develop, as well as extensive user testing. All mobile applications used for health requires validated, careful tests that proves the claimed health-benefits. In our case the app improves the rehabilitation process of a patient. Even if these user-tests would be beneficial for improving the application and getting qualitative results, due to the size of the project we may not be able to collect a test group, design test cases and distribute a prototype for them to test. The manageable tests for

this project will mainly focus on the applications functionality and design and getting qualitative feedback from the project owner and physiotherapists.

1.4 Resources

The project owner, Eirik Grøneng, has been an important resource for developing knowledge required for the basic requirements and design for such an app. He has been very helpful in providing us with information, details and sketches around the application's shape and features. A continuous dialogue and consultations with him have been very helpful throughout the project.

We use our own personal computers and mobile devices, combined with Android studio, to develop the application. To ensure we have an efficient workflow together, we will utilize Git with the free hosting service GitHub. We go into further details of the tools and languages used in this project in Chapter 3.3.

1.5 Organization of the report

Chapter 1 of the rapport describes the goals of the project, the current limitations, and resources used to build the application. It will also introduce the project owner, and his motivation for the project

Chapter 2 will go more in depth about where the project stands in terms of intention, existing work and requirements for the solution.

Chapter 3 contains the different possible development approaches we met when we started the project. They are discussed, and a final approach is selected with an explanation why that one was chosen. Requirements and further resources needed are also defined, along with the chosen development methodology, the project plan and the risks the group might encounter during the project, as well as how to deal with them.

Chapter 4 discusses the detailed approach we took when creating the design and architecture of our application.

Chapter 5 describes the evaluation methods used, and the evaluation results we obtained from Eirik Grøneng and a test group of fellow students.

Chapter 6 includes a discussion of the consequences of our chosen approach of development, how the choices we made influenced the results, what we would do differently if we were to redo the project, and gives a suggestion for further work.

Chapter 7 gives a summary of the projects goals and illustrates how the goals were reached.

2 PROJECT DESCRIPTION

This project will describe the development of an Android application based on the idea and the requirements from the physiotherapist Eirik Grøneng. The application aims to serve as a tool for him and other physiotherapists to increase the effectivity of their patient treatments.

2.1 Practical background

The project owner and the project group have not previously collaborated before. This has not hindered any communication or progress with the project. We received initial plans and requirements on how the application should look when finished.

2.2 Project owner

Eirik Grøneng graduated as a physiotherapist in 2015, with further education in Cognitive Therapy (2016). He is currently working on a specialization in clinical orthopaedic medicine, with in-depth expertise in the examination and treatment of muscular and skeletal disorders.

Grøneng has since 2016 worked in private practice - most of the time as a physiotherapist and general manager at a private clinic in Hardanger. He has extensive experience in the examination and treatment of sports injuries, as well as injury prevention and performance-enhancing training. In 2018 he received the FIFA Diploma from the FIFA Medical Centre of Excellence - a competence certificate in sports medicine, especially aimed at football.

Eirik Grøneng's patients are people who seek help regarding injuries, injury prevention, performance-enhancing training and treatment of muscular and skeletal disorders. There does not exist a way of visualizing the progress of a patient at this time, so every feedback and correction that he makes of a patient's form in an exercise is done orally.

Communication between the project owner and the project group takes place during meetings, over email and text messaging.

2.3 Previous work

When developing a new application, it is important to research earlier, existing work, that could assist the development process and improve the result. We were not given any existing code or reference applications to base this project on. Our project aims to improve the way feedback can be given to a patient executing exercises for rehabilitation or for physical training and allow the physiotherapist to more easily track the patient's progress. We looked for applications that

aimed to solve either one of those problems, and literature that could assist us in the development of an even better application.

2.3.1 Existing solutions

When searching through app stores and app libraries across different platforms we were able to identify a few applications that had similar purposes. One application that had very similar functionality to our planned application is *Coach's Eye*. This application allows you to record a video of yourself performing an action, trim this video down, change the playback speed and draw lines and angle measurements on top of it (Coach's Eye, 2019). However, this application is more targeted towards sports coaches to improve their athletes' performance.

Video Merge – Side By Side is an application that can inspire our application. It can merge two videos side by side and using this merged video to compare a patient's progress. This application does however not allow you to trim and alter the playback speed. The application will also the two separate videos into one video together, whereas we aim to have two videos playing side by side simultaneously (Video Merge - Side By Side, 2019).

2.3.2 Assisting literature

When developing an application for Android, we first need to understand what developing an Android application entail. We looked for web articles and books that could give us more insight than we currently had and potentially nudge us in a direction we might not have looked at first.

One of the project group members have previously attended a course that taught Android development. During this course he read parts of the book *Android Programming: The Big Nerd Ranch Guide*. This book lays out the fundamental parts of the Android programming, as well as teaching important development practices when developing for Android (Phillips, Stewart & Marsicano, 2017). We used this book to gain a fundamental understanding of the Android activity lifecycle, which is a difficult and very important concept to understand. The activity lifecycle is what we use to create an application with multiple activities. In our application we need to retrieve data from the systems camera app, and still manage the session and data from the already active activity. This requires knowledge and understanding of the activity lifecycle to implement.

2.4 Initial requirements specification

We were initially given some requirements for the application. This application does not only rely on our work, but also the physiotherapists as well as their patients. The physiotherapists must

provide their patients with proper information about the exercises, and the patients need to read it as well as putting in the effort to execute the exercises properly.

- Possibility of recording an execution of a selected exercise/movement.
- Options for cutting of the video length (start/stop).
- Auto-fit of tempo to match predetermined length (determined by archive video or standard length).
- A few pre-recorded reference videos provided by the project owner.
- The possibility for patients to record their own reference videos.
- Side-by-side comparison between recorded execution, and selected reference video
- The possibility for patients to record their own reference videos, and for physiotherapists to provide instructions and video execution.
- Compare own performance to a reference video, made by a physiotherapist or yourself, in split screen.
- Possibility to start/stop, control the speed of the playback, and rewind.
- Arrows/dashes and angles (gradients) can be placed on top of the selected video for more precise analysis.
- Username/password for individualization and premium authorization (However, we decided later that having a login option with username and password would be redundant for this current project at this time).

2.5 Initial solution idea

The initial solution idea we are envisioning is a tool that a physiotherapist may use to assist his physiotherapy. The tool should be mobile and lightweight. This led us to conclude that an easy-to-use mobile application with a simple design fit the purposes of this work best.

2.5.1 Purpose and functionality

The mobile application will act as a tool that can be used for two purposes. These two purposes will cover the problems that acts as a motivation for this project, described in Chapter 1.1.

2.5.1.1 Exercise execution feedback

The first purpose is the ability to give more precise feedback on the exercise a patient is performing. The physiotherapist will first record the performance of the patient, such as bench press or squat. He will then be able to add overlays on-top of the recording, consisting of objects such as a ruler, angle measurements and lines. The recording will then be played side-by-side with a video of a correct execution of the same exercise. By adding these overlays onto the video and comparing the current recording with the video containing the reference execution, the

physiotherapist will be able to identify if there are subtle changes that needs to be made for better performance.

Reviewing the video alongside the patient may also assist the patient in understanding where he might need to make changes for improvement.

2.5.1.2 Tracking patient progress

Another aim of the application is to give the physiotherapist the ability to track a patient's progress in a more visual way. After the physiotherapist has recorded the patient performing an exercise, potentially giving feedback on it (as described in Chapter 2.5.1.1), he may save the recording and store it. This recording can then later be used as a comparison video alongside a new recording. By doing this, the physiotherapist will be able to see if the new execution contains improvements or if the patient needs further guidance to perform his/her exercises better. By comparing previous and new recordings the physiotherapist can track the progress of the patients.

2.5.2 Versioning

The application will come in two different versions, a free version and a premium version. The premium version will have to be purchased with a one-time payment whose price has yet to be decided. In the free version of this application, the user will not be allowed to save any recordings. This makes the free version only able to cover the first purpose described above in Chapter 2.5.1.

The premium version will contain the features contained in the free version; furthermore, allow a user to save him or her recording exercises. This unlocks the ability to use this application to cover the second purpose of this project.

2.5.3 Design and architecture

Appendix A contains a sketch made by the project owner and depicts the initial requested layout for the application. The sketch also contains a rough idea for how the design of the application should be. The only requirement we have is to develop the application similar to the sketch. However, both the project group and project owner are aware that changes to this initial layout may be needed to allow a better solution during the design and after the tests. A better solution classifies as a solution that would result in a better experience for the user, such as improved design or added functionality.

2.5.3.1 Video

The application will store previously filmed videos, as well as let the user film new videos. The user will get to choose a reference video for the exercise and let them cut and edit the newly filmed video in order to compare them. In this frame the user will also get to place angle measurements and arrows etc. on the video to highlight differences. The reference video and newly filmed video will be compared next to each other, and then let the user save the newly filmed video if they have the premium version of the application.

Videos can also be accessed through the archive if the user would like to look at a previously filmed video.

2.5.3.2 Category

Categories will act as a folder that allows the user to sort the videos. This is important to allow the application to be structured and easily navigated when there are many videos saved. The user himself decides how he wants to structure the categories. He may wish to name categories after patients, or categories after the exercise. The user can have as many categories as he wishes. When all videos inside a category has been deleted, the category itself will be deleted. The application will be shipped with four existing categories, each representing an exercise. They will consist of one reference video each (see Chapter 4.2.3).

3 PROJECT DESIGN

In this chapter we will look at the different development approaches possible and conclude which one promises the best result to match the requirements. We will also discuss the different risks involved in this project.

3.1 Possible approaches

The project owner requested a mobile application that he could instruct his patients to use if necessary. He did not have any further specific requirements on the application software or architecture. This opened different possible approaches which we will discuss in this chapter.

3.1.1 Cross-Platform Mobile Application

An approach we considered was developing cross-platform. When referring to cross-platform, this report refers to iOS and Android. This is due to these two operating systems taking up 97.43% of the worldwide mobile operating market share (StatCounter Global Stats, 2019).

To develop cross-platform we either need to develop two separate applications, one for android and one for iOS, or adopt a framework that can compile and build to both operating systems. Using these types of framework removes the problem of having to make two applications, however they have their own architecture and rules that no members of the project group are knowledgeable in.

3.1.2 Native Android Application

Another approach would be to develop a native application for Android. Android is an open source, Linux-based software stack (Android Developers, 2019). Native Android applications can be written in either Java or C++. When using Java, the application will be running on a Java Virtual Machine that runs on top of the Linux kernel. C++ allows us to write applications that gets directly ran on the Kernel, which can in bigger application yield a performance boost.

The application would be developed through Android Studio using Java, which is a programming language all members of the project group have experience with.

3.1.3 Discussion of alternative approaches

The most desirable solution would be to have the application not exclude any mobile operating systems. This would ensure that the client is able to reach a broad spectre of customers. However, due to limited experience and time constraints, the group will go for the Android

approach. A big deciding factor for choosing this approach is that it allows for development in Java. This allows the group to start development right away, as well as being more time efficient.

3.2 Specification

Our main goal is to produce a functioning product, and based on our previous knowledge, we believe that choosing to develop an Android application is the best option. Even though some of the physiotherapists might be using iOS products, we believe that choosing to develop an Android application based on the current knowledge in the group, will overall give a better functioning product.

We chose to develop through Android Studio and use the Java language, because the developers all have experience with at least one of these technologies. We do not have any experience with developing for iOS, and because of our limited time this would possibly take up too much of our time to learn.

3.3 Selection of tools and programming languages

3.3.1 Tools

Android Studio

Android Studio is the official integrated development environment (IDE) for Google's Android operating system and is designed specifically for Android application development. It was developed by JetBrains in collaboration with Google and has a similar layout to JetBrains' well known IDE called IntelliJ.

It is also possible to use other IDE's for developing Android applications, although requiring external software and tools in order to compile and test the application. Using Android studio allows us to both develop, compile and test the application in the same software, allowing for an easier environment to develop the application in.

GitHub

GitHub is web-based hosting service for version control using Git. This is where we save the project, as well as check versions and keep control of issues. It is the easiest way to make sure all group members are up to date on the project.

We chose GitHub over alternatives such as Bitbucket and GitLab, because these alternatives are more suited for bigger teams and enterprises. GitHub is well suited for smaller groups like ours, and the project group has experience with it from previous school projects.

3.3.2 Languages and libraries

Java

Java is one of two languages one can choose to work with when developing an Android application through Android Studio. It is the language the project group is most familiar with, which is why we chose to work with it over Kotlin.

If we were to work with Kotlin, we would have to spend more time learning the language, which would result in slower development of the application. Kotlin also has limited online learning resources, in comparison to Java which has a wide open-source ecosystem (Atha, 2018). It would also be difficult to find an experienced mentor to help us with Kotlin if we get stuck.

FFmpeg

FFmpeg is the leading multimedia framework which can handle video and audio files and streams ranging from very old formats to the new cutting-edge formats (FFmpeg, 2019). It allows you to encode and decode from many different audio- and video formats, with a big range of options in the encoding and decoding. Such options include compression rate, max bitrate, video scaling and duration trimming.

When developing our application, we needed to allow for videos to be trimmed down, and therefore had to seek libraries that allowed this. We could have used smaller Android specific libraries for trimming videos, however the extra functionalities that FFmpeg offered made it a better choice. We will discuss the chosen video compression and codec more detailed in 4.2.1.

Room

Room is a persistence library that provides an abstraction layer over SQLite that allows for more robust database access while still harnessing the full power of SQLite (Android Developers, 2019). It is a part of Android Jetpack which is a collection of libraries and tools for Android developers. Room allows you to create an easy manageable database structure for your application, without having to worry about persistency and integrity with manual written SQLite queries.

We initially considered building the database with SQLite directly, however using Room gave us so many assurances and benefits that the extra time it took us to learn it would still be outweighed by the benefits it provided.

ExoPlayer and ExoMedia

ExoPlayer is a media player for Android that is developed by Google. It provides a range of extra features that the Android's MediaPlayer API does not support (ExoPlayer, 2019). ExoMedia is an

easier to implement media player that bases itself on ExoPlayer. The goal of ExoMedia is to be as similar as Android's media player and still include the extra features that ExoPlayer offers.

When we first made the application, we used Android's media player. However, using Android's media player left us with some problems that would have to be solved by either using an altered version of it, or by using another library such as ExoMedia. The problems are discussed in in Chapter 4.2.2.

3.4 Project development method

3.4.1 Development method

We have chosen to develop using the agile framework for software development called Scrum. Scrum is designed for teams of three to nine, however since we are at the bottom of that range, we may not be able to take full benefit of this framework (Scrum.org, 2019). We will use Scrum to break the work needed to be done into sprints. Each sprint spans across two weeks, and after the sprint is done the project group will look over what has been done and re-plan for the next iteration.

Scrum is based on continuous learning and adjusting to be able to handle any problem that might occur. This will make it easier for the team to make changes during development if we get new requirements or ideas along the way. The goal is to make the team work as a unit, and to include the entire team when working towards a goal.

3.4.2 Project plan

To map out the course of the project, a GANTT chart was made. See further information in Appendix B.

3.4.3 Risk management

To map out the different risks that might occur, a risk analysis has been performed. You can read more about this analysis in Appendix C.

3.5 Evaluation method

To develop an application that promises to be useful for physiotherapists we would need to test the design, functionality and whether the application can assist in the rehabilitation of their patients. Testing the application's design and functionality is straightforward, and we aim to have Eirik Grøneng test the application, as well as some students to get a broader test group. As we

discussed in Chapter 1.3, we will not be able to test the impact this application may have on the rehab of a patient due to time constraints.

4 DETAILED DESIGN

When developing a mobile application, it is important that the reasoning for using one architecture or component over another one is clear. In this chapter we will discuss the architectures of the application, the process of implementing some of the key features, the different use-cases and finally the chosen UI design.

4.1 Architecture

When developing the application, it is very important that we select the right architecture, tools and libraries. By researching the different technologies beforehand, we made sure to omit as much extra work as possible.

4.1.1 Database architecture

To have control over all the videos and categories a user may create we will store the data in a relational database, which allows for efficient managing of the data. We will utilize the Android Room persistence library, which is an Android architecture component that provides an abstraction layer over SQLite for local database storage. Storing the data in an SQLite database in combination with Room makes it easy for us to retrieve data in a convenient way.

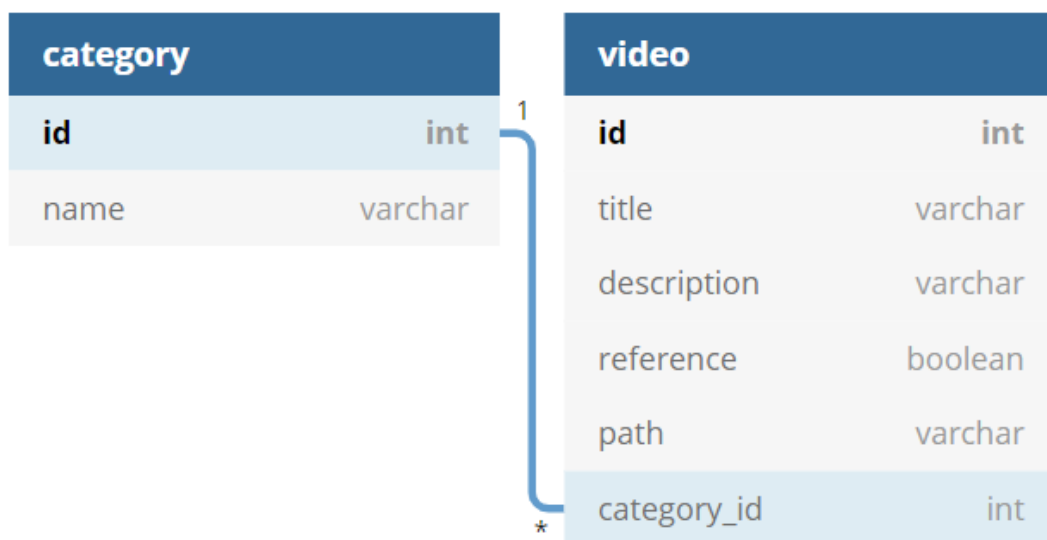


Figure 4.1. Database relationship diagram

Figure 4.1 illustrates the design of our database. For the categories, we only need to store the name of the category. We use a one-to-many relationship to connect multiple videos to one

category. In the video table we need store all the details that a video requires. Considering we cannot store the video itself in the database, we need to reference the path to where it is stored on the device. The reference is stored in the video database entity.

4.1.2 Storage Architecture

Considering the application should be able to work offline, we decided to only use local storage on the user's device. This choice saves us the hassle with online storage, including finding an appropriate online storage plan, as well as the resources required to maintain it.

dir = "External Storage/DCIM/TrimmedVideos"

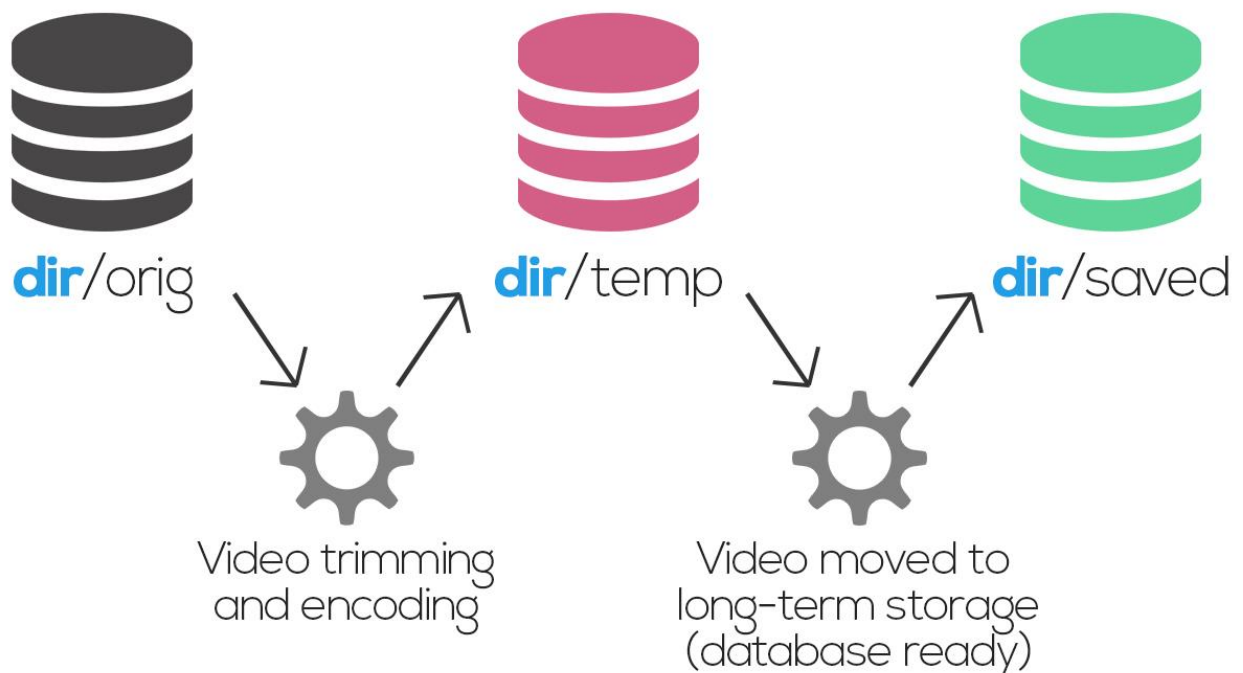


Figure 4.2. The different stages of a recording in storage

As shown in Figure 4.2, when a video is recorded, it will go through multiple stages before it is finally saved for long-term storage. We have set the system up in such way, so we have better control over the videos. For this project we need to know which videos are not encoded and trimmed, which videos that are encoded and trimmed but not saved, and the videos that are saved.

Initial recording

When a video is initially recorded with the application, it will be recorded using the built-in

camera app on the android phone. When the user is done recording the video, it will be stored in a custom made “orig” folder.

Trimmed recording

The encoded and trimmed version of the recording will be stored in “temp” folder. The initial recording in “orig” will now be deleted as it is no longer needed and will only take up space on the phone. If the premium version is not unlocked, this will be the last step in the recording storage process. The video will be deleted afterwards.

Saved recording

If the premium version is unlocked and the video is requested to be saved, it will be moved to the “saved” folder. It is also in this moving process that the video will be saved in the database.

Storage fail-safe measurements

Some recordings may have a big file size. It is therefore important that we do not store more recordings than necessary. If the application were to crash or perform unwanted behaviours, we will perform actions to ensure that the integrity of the storage remains. Such unwanted behaviour could be that the initial recording or a trimmed video fails to delete. The fail-safe measurement here would be to check if files that are supposed to be deleted still exists, and then proceed to delete them. Since we have divided the storage into different stages, each with its own folder, it will be easy to identify these files. The “orig” and “temp” folder should not contain any files when the application is not in use, and these folders will therefore be cleared on application exit and application start.

4.1.3 Application architecture

The user should be able to see all the videos and categories he may have created. This is dynamic data that we need to present in a reliable way. To achieve this, we need to use an appropriate model allows for dynamic data presented to the user.

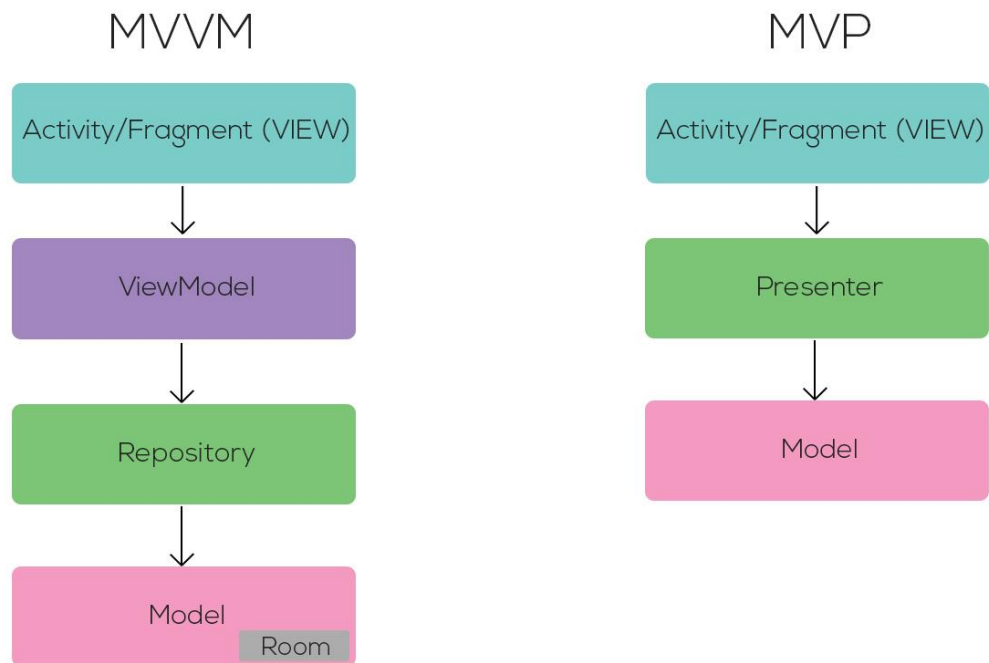


Figure 4.3. MVVM vs MVP in Android development

The main models we looked at was Model-View-Presenter (MVP) and Model-ViewModel-View (MVVM). MVP is a well-established model that is widely used in many different programming languages and projects. Being such a fundamental model, a wide range of documented and tested approaches in Android has been performed using it. However, support for MVVM in Android is much newer and has been implemented more tailored to Android components architecture.

4.1.3.1 The presenting layer

There are two main differences between MVVM and MVP in android; the part which prepares the data for the view and the model layer. In both models, the data preparation layer has responsibility to read data from the model layer and format it in such a way that the view needs to perform a minimal amount of operations on it before showing it. Android has made a custom class called ViewModel which allows for a better way to present this data fit for the Android lifecycle. Creating a generic presenter class will require extra work and resources to mesh with the same lifecycles.

4.1.3.2 The model layer

The model layer is also differentiated in MVVM. An Android component called Room, which we discussed more detailed Chapter 3.3.2, allows us to link the model to the ViewModel. This link

gives us extra features and compatibility assurance. It also possible to use the Room in MVP, although having the disability to use many of rooms useful features.

4.1.3.3 Model choice discussion

None of the project group members had any previous experience with MVVM, however since we are using the Room component for our model layer, we concluded that using ViewModel as the presenting layer fit our needs best. The added compatibility and functionality we get when combining these components allows us to more easily implement MVVM into our project, which would pay off the time needed to learn the model itself.

4.2 Key feature implementation

When developing the application, there are certain aspects that require more effort to implement. Such aspects may be features that are more difficult to understand and comprehend or features that might be more time consuming to implement. In this subchapter we will discuss some of the important features that required extensive effort to implement.

4.2.1 Video codec and compression

The application may eventually build up a comprehensive storage of recordings. It is therefore very important that these file sizes are as small as possible. To reduce video file sizes, we need to compress them. FFmpeg is the framework we chose to handle both the compression- and trimming of videos.

The video codec is the software that compresses the video. However not every device supports every codec, so we first had to find codecs that were supported by Android devices. We also wanted a codec that allowed for a compression customizability. We ended up using the H.264 codec, as it can compress videos that are supported by all android devices, and it allows us to set compression rate and compression speed.

When compressing videos there are multiple aspects we need to take into consideration. First, we want the compression to be as big as possible. Although, the greater the compression, the longer the compression process takes. A long compression like this would halt the flow of the application, and it is therefore important that the compression does not take up too much time.

Constant Rate Factor (CRF) and preset are the two main H.264 arguments we used to find the appropriate compression to speed rate. CRF will set the compression level. The level can be set between 0 and 51, where lower values result in less compression (Robitza, 2017). The preset argument is the one that sets the compression speed. The higher the preset speed, the faster the compression will be and the bigger the file size will be. We tested a wide combination of values

for these arguments and ended up with a combination that yields a reduced file size without the loss of too much quality, and the compression taking an acceptable time to complete

4.2.2 Video player

One of the core functionalities of this application is to present videos to the user. If the video shown to the user is not shown in an appropriate and intuitive way, the application can in worst case be rendered useless. We have therefore put a lot of thought and effort into what video player we are using.

One functionality of the application is that it allows the user to trim a video. When trimming a video, the user drags a ball that either indicates the starting position or end position of the. If the user is dragging the former, we want the video to display the frame at that position. This process is called scrubbing.

4.2.3 Reference videos

One of the most important resources in the app are the reference videos. Reference videos are the videos that will contain a near-perfect execution of an exercise. A user of the application may create custom reference videos if he chooses to, however we want the application to be usable without requiring a user to be able to record a well-performed execution. Therefore, we will ship this prototype with four pre-recorded reference videos already installed for four different categories.



Figure 4.4. Recording of reference videos in the media lab

These reference videos were recorded in HVL's media lab (see Figure 4.4), which our advisor booked a month in advance. On the 22nd of May, the group met up with the project owner, as well as a study friend of his who was going to act as the model performing the exercises. In order to get a more authentic reference video, we brought a barbell and two plates to get a realistic execution of a weighted exercise. The media lab contains high-end cameras, and the people working there are proficient in recording and editing, so they took care of the recording and editing. Afterwards we were provided four edited videos, each being a reference video representing its own category.

4.3 Use Cases

For the application to have a clear purpose, and a design that puts emphasis on this purpose, we need to make the different use cases this application should contain clear. We have concluded with three main use cases, two for the free version and one for the premium version, each described and presented with a sequence diagram in the upcoming paragraphs.

4.3.1 Use case 1: Record and compare execution

The most important use case is the ability to record a new video and compare it to an existing video. This use case depicts the core functionality desired from the project owner and will therefore be the feature we have prioritized the most.

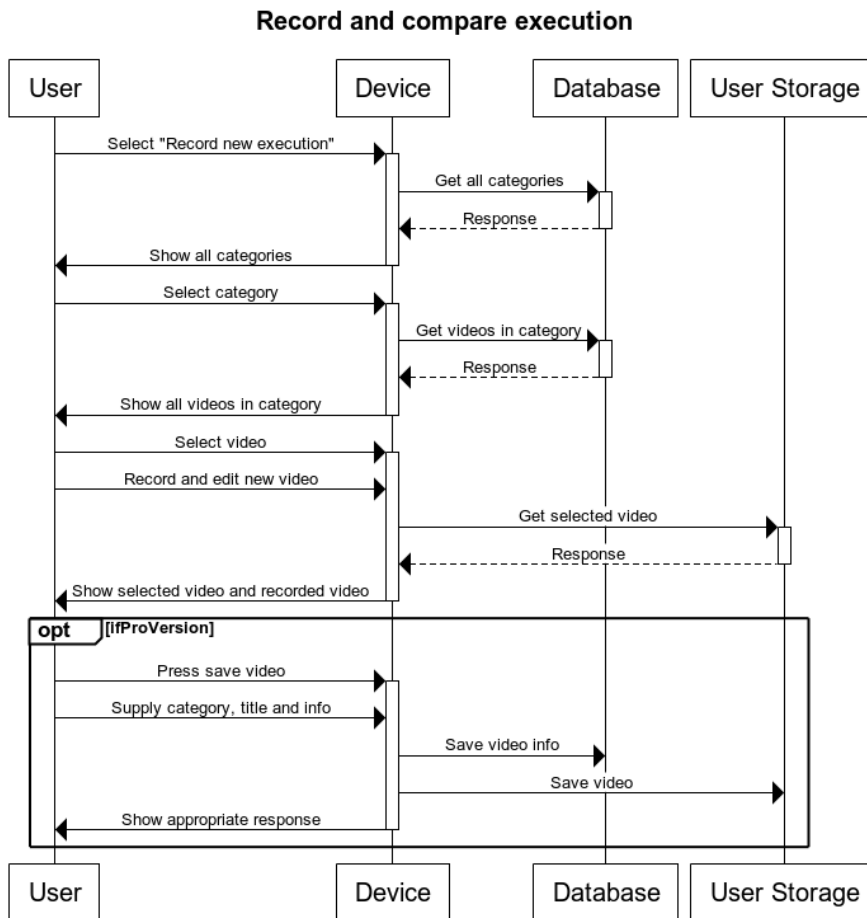


Figure 4.5. Sequence diagram representing use case 1

The user will press the button that indicates that he wants to record a new execution, and the application will list up all the categories that the user has available. The user is then going to select a video that he wants to use as comparison. The last step of this process is to now record an execution of the exercise that fits the selected category. If needed, the video may be trimmed and its speed altered, to better fit the comparison video. As depicted in Figure 4.5 if the user has the premium version, and wishes to save the newly recorded video, he may do so. An example of this use-case could be a physiotherapist having a patient perform a squat. He sees that the patient performs it incorrectly and decides to use this tool to more clearly illustrate what the patient does wrong. He selects the squat category, records the patient performing the squat, and lastly trims the recording to the desired length. Afterwards he is presented with two videos played in a parallel view.

4.3.2 Use case 2: Show an existing execution

If the user has already existing recordings of previous executions, and does not wish to record a new one, he can select to show details of a previous one. In the main menu, the user selects that he wants to enter the archive.

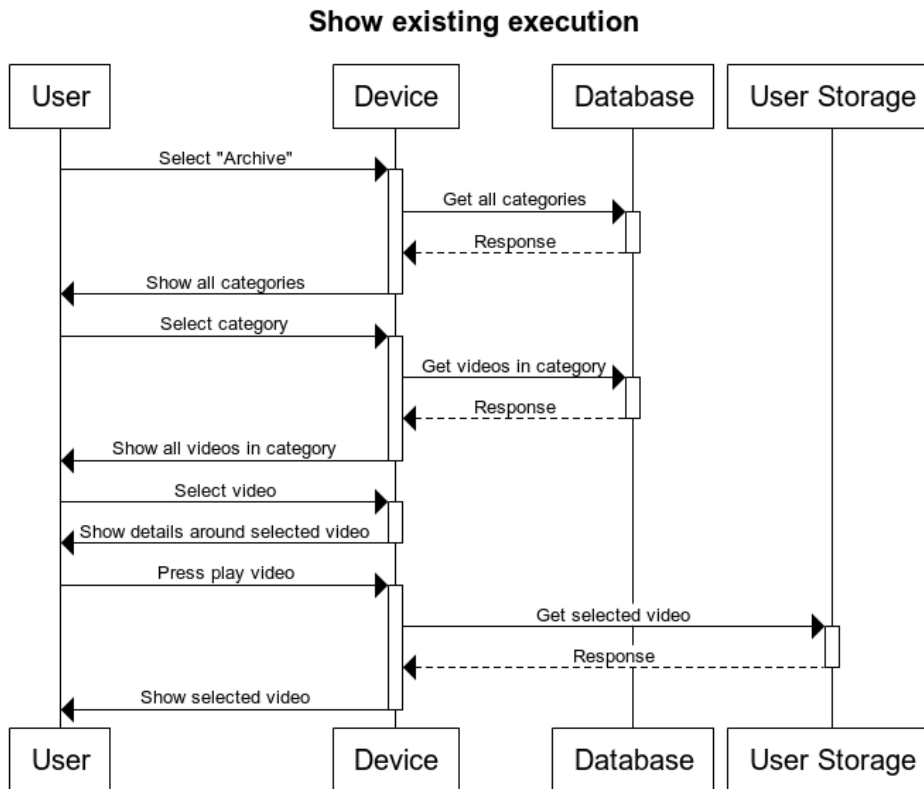


Figure 4.6. Sequence diagram representing use case 2

The following steps are identical to the first use case described. After the user selects a video, he will be displayed a screen with information about that execution. This is the info the user put in when creating the video. Lastly, if the user wishes to play the selected video, he may do so as well.

4.3.3 Use case 3: Create a new category

The last use case, which is only supported for the premium version, is the ability to create new categories. Categories are intended to act as a way for the user to sort the recordings. Each recording will fit a category that the user self chooses.

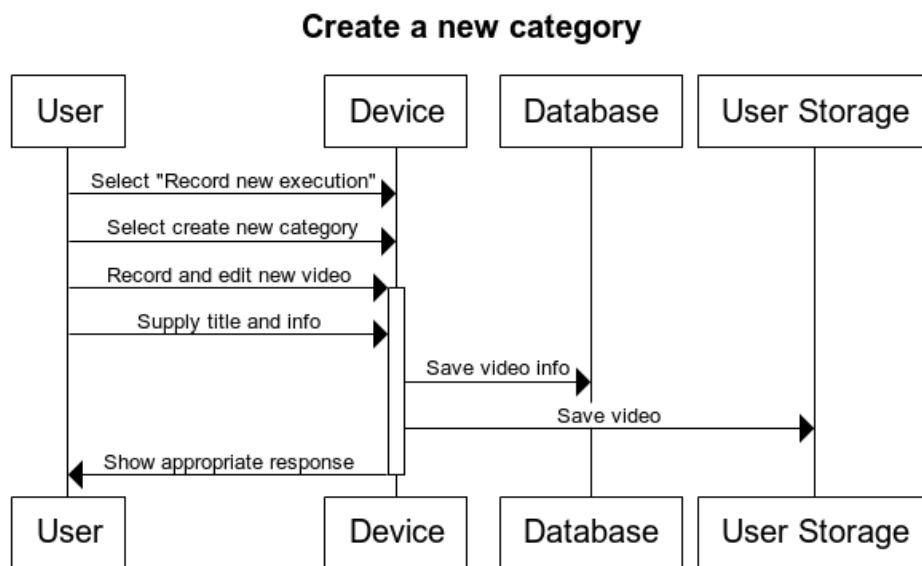


Figure 4.7 Sequence diagram representing use case 3

Considering the use case 1 described above, where the user will select a video to use as comparison, a category cannot be empty. If there are no videos in the category, the user has no videos to select, which in turn renders a category useless. A category should therefore never be empty.

As shown in Figure 4.7, after the user has selected that he wants to create a new category, he will also be required to supply the category with a recording. This recording will act as the initial comparison video for future recordings in this category.

4.4 User Interface Design

4.4.1 Android UI components

Android allows for easy application development by providing a variety of pre-built UI components. These are components such as structured layout objects and UI controls that will allow us to build the graphical user interface for the app.

The Android user interface is generally made with View and ViewGroup objects. The View class is a base class for creating interactive UI components usually called “widgets”, such as TextView, Checkbox, Buttons etc. The ViewGroup is a base class for layouts, and these objects acts as containers for the View objects (Android Developers, 2019). These layouts and objects are created in Android Studio using XML and the visual design editor, which enables us to preview the layout as we go. This makes it a lot easier for us to develop the design of the application.

4.4.2 UI layout

We have made sure that the layout and design of the application is as simple and easy to understand as possible. We want the functionalities of the application to be intuitive, so that the user will not have to spend a lot of time learning how to use the app.

4.4.2.1 Main menu

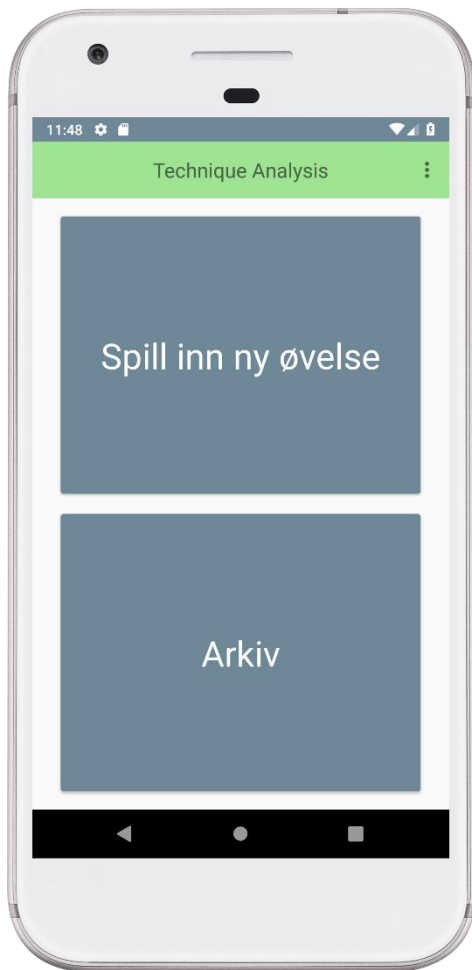


Figure 4.8 Main menu

When the user first opens the application, the landing page will be this main menu. The menu will give the user the option to either record a new execution, or to view an existing video by entering the archive. Both buttons will open the categories page.

4.4.2.2 Options menu

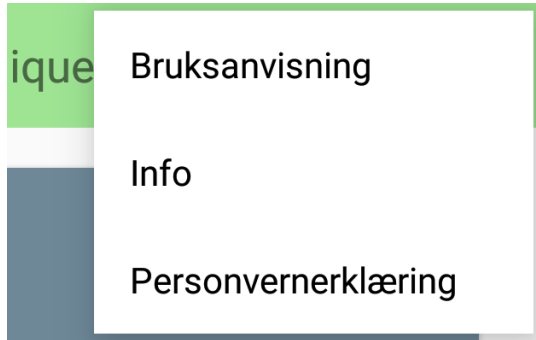
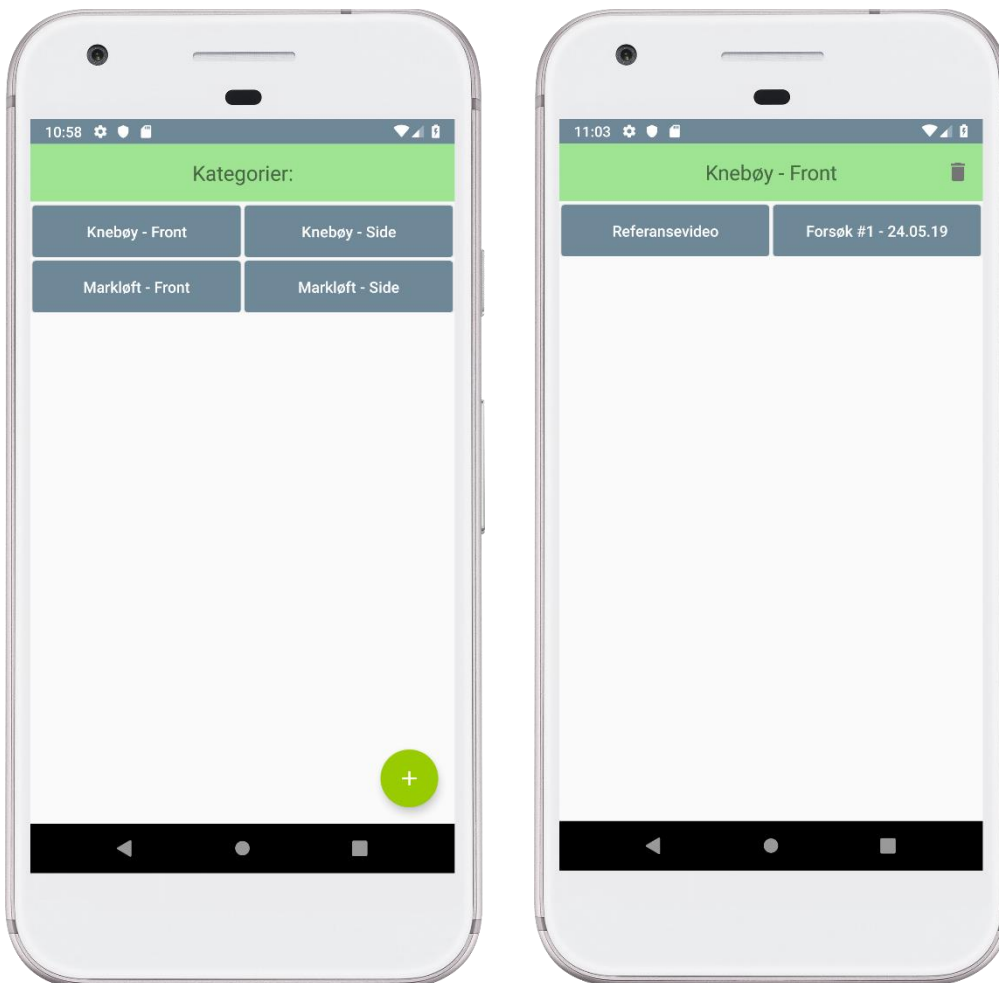


Figure 4.9. Options menu

The options menu (see Figure 4.9) is accessible by clicking on the three dots in the top-right corner. This menu will give the user quick access to the user guide, practical information about the application, and the privacy policy.

4.4.2.3 Categories and videos page



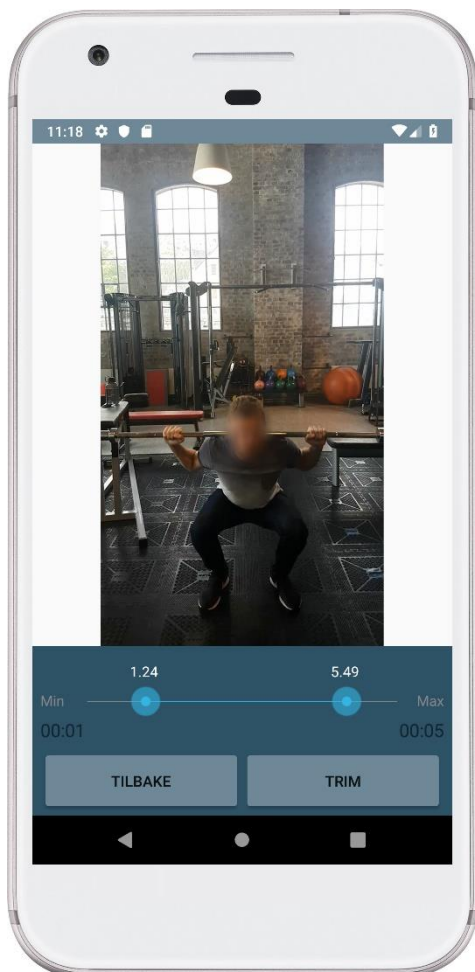
*Figure 4.10 Category page**Figure 4.11 Videos page*

Whether the user wants to record a new comparison video, or he wants to view a previous recording, he must select the appropriate category. The category page (see Figure 4.10) will contain all the categories that the user has available. If the premium version is unlocked, the floating button in the bottom-right corner will take the user to the category creation page.

When the user selects a category, it will take him to the videos page (see Figure 4.11). This page will contain all the videos in the selected category. Depending on the action the user selected in the main menu, when the user selects a video, he will either be transferred to the video info page or the recording new execution page.

The floating action button in the bottom right corner on the category page is the button the user presses to create a new category.

4.4.2.4 Trim video page

*Figure 4.12 Trim video page*

After the user has recorded their execution of selected exercise, the trim activity is displayed. This activity contains a self-made toolbar the group created in order to make the app more user friendly, as well as make it easier to go back to previous activities. As shown in figure 4.12 the toolbar contains two sliders in order to make the workflow less complicated, and the video will be played in a loop using the time interval from the user input.

After the video has been trimmed, a progress bar is showed. The groups reasoning behind this, is that it makes the apps workflow smoother, instead of having a blank screen, the user will know how far the processing has come and how much time is left till completion.

4.4.2.5 Compare videos page

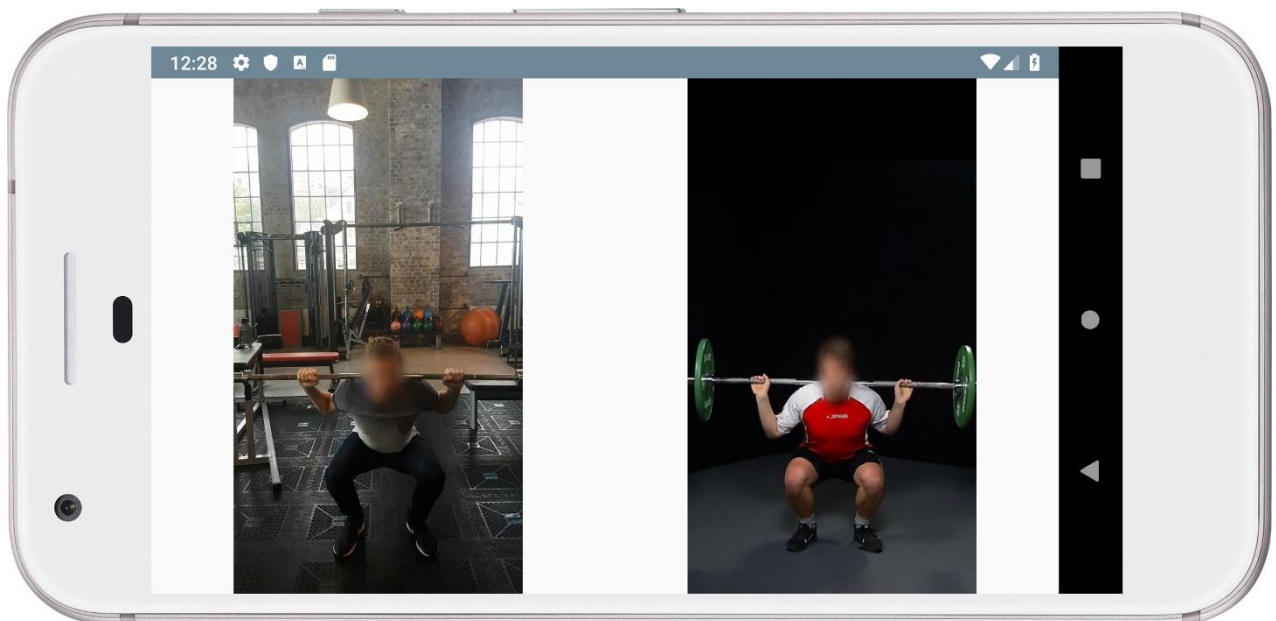


Figure 4.13 Comparison page

Perhaps the most important activity in the app is the compare activity, as this displays the core- and most important feature in the app, comparing two videos. The activity will start two videos side-by-side in landscape mode and restart them both when the last video is done running. Using ExoMedia allows us to keep the aspect ratio on the videos so they do not appear squished.

4.4.2.6 Video info page



Figure 4.14 Video info page

When the user has chosen a video from the archive, they get redirected to the video info page. This page displays the title of the video, as well as the description that the user has given it. The user can then choose to view the video by clicking the button on the page, which will display the video on a new page.

4.4.2.7 Save video page

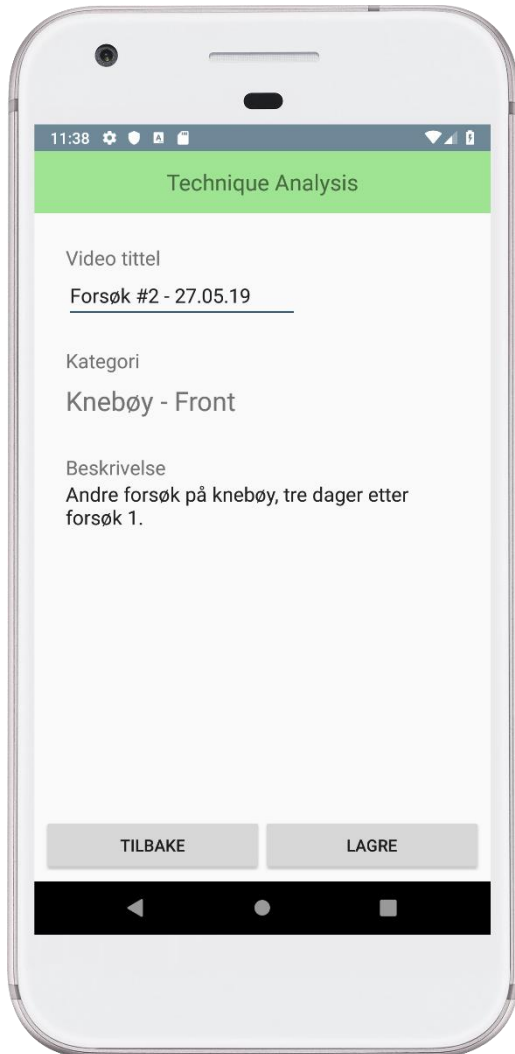


Figure 4.15 Save video page

If the user has recorded a new video, either for comparison or for creating a new category, he may save it. When the user wants to save a video, he must provide a title for the video and a description for the video. The title for the video will be shown in the videos page shown in 4.4.2.3. The description is only visible in the video info page described 4.4.2.6.

4.4.3 UI Language

The language for the UI is set to Norwegian. This was the most natural choice considering both the project owner and the project group are fluent in Norwegian. A good practice when developing Android applications is to not hardcode any messages presented to the user, but rather leave them in XML string files (one for each language). We have followed this practice, making it easy for the application to support more languages without having to change any code.

5 EVALUATION

Evaluation of the application is a crucial part of development. It is important to have opinions from the project owner, and through him opinions from other physiotherapists and patients, but also opinions about development. Since we are evaluating the usability of the prototype, we can use non-stakeholders to also give feedback.

5.1 Evaluation method

Instead of having recurring evaluation phases to get consistent feedback on the development of the prototype, we performed evaluation after the first version of prototype was finished. This evaluation will contribute to better understanding the usability of the application at its first prototype stage.

Since this application is meant for physiotherapists that may not have much IT knowledge, it is important that the different use cases are very intuitive and easy to understand. There will not be enough time to evaluate whether the application helps shorten a patient's rehab time, so the focus of the evaluations will mainly target the application's UI and functionality.

We had tests carried out by both the project owner, and by fellow students. This is to gather a larger test group so we can receive broader feedback. For Eirik Grøneng, since we were not present when he evaluated the prototype, we sent over some questions for him to answer. These questions will cover his thoughts of the application, if there are aspects that he would rather have done differently, and if the application is usable at the current stage.

We had, to gain further feedback on the usability and intuitiveness of the prototype, students to test as well. This way we gained insight of how the application was for people who have no previous insight into its design and use cases. We asked them questions both while they were using the application, as well as afterwards, to gather as much feedback as possible.

5.2 Evaluation from Eirik Grøneng

We did not have the ability to meet up with Eirik Grøneng to present the prototype to him, and instead opted for sending the prototype as an Android application package (APK) over email. This also meant we could not ask questions live, and instead had to rely on sending him some questions beforehand.

Q: How are the functionalities in the prototype? Is there anything you feel is missing in the application?

A: In the trim video page, it is rather difficult to see where the in the video the trimming happens. To make the trimming more straightforward, it would be preferable to have the video pause when the trimming scrub is being dragged.

I would have liked to have the ability to “draw” under the video comparison page. It would also be preferable if the comparison page supported vertical view, as well as support slow motion playback.

Q: Is the UI intuitive? Is there anything that was too vague in the apps design and layout?

A: When you save a new recording, it should rather return to the main menu. At its current state it is a bit unclear where in the app you end up after saving a recording. Other than that, the app is clean, and easy to understand and navigate through.

Q: Do you see benefits of utilizing the prototype in its current state, or does it lack some functionality for it to be useful for you?

A: The prototype lays a good foundation for future work. It is to this date missing some features before it can be used as initially wanted.

Q: Your general thoughts of the prototype?

A: The app is built on the requested UI and layout. The core functionality supporting the ability for parallel playback of recordings with matching length is present. The page for playback is easy to understand, and the presentation of the recordings (both initial reference videos, and newly recorded) is as requested. The archive is easily accessible and gives a clear overview over the content.

To sum up, the first prototype contains the two most essential functions that was requested and lays a good foundation for the project to be continued in the future. There are still some features missing before it can be used as I had planned.

5.3 Evaluation from students

Testing an application with only 5 people lets you identify almost as many usability problems as you would find using a larger test group (Nielsen, 2012). We therefore gathered a test group consisting of 5 students in the 21-25 age group, and we were present when they evaluated the application. They were not gathered in one group, but rather evaluated the prototype individually. These students were people with existing app development knowledge, and people with little or no IT knowledge. Despite their different IT knowledge, they all had very similar comments on both the application’s design and functionalities.

The students found the general design of the application very intuitive and easy to use, however, they all had some comments on the functionalities. The biggest issue they found, was that it was

difficult to trim the newly recorded video for it to match the reference video. It would have been better if it was possible to see the reference video next to the video while you are trimming it. This would make it a lot easier to match the starting and stopping points of the two videos.

We also got a comment on how the categories are sorted. At this point, videos of the same exercise that are shown from different angles will be listed as different categories. The test subject thought it would be better if it was all just put into one category, so that if you for example click on the category called “Squat”, you would get videos of the squat both from the front and the side etc.

The test group also pointed out that you have to click through a lot of screens before you get to the desired functionality. This could be fixed by giving more information on each screen, which would reduce the number of different layouts needed. It should also be easier to get back to the main page without having to go through each screen again.

Overall, the test group gave positive feedback on the general user experience of the application. We got no complaints on the manoeuvrability of the application, as they all thought it was easy to navigate through the screens, and that the design was simple and intuitive.

6 DISCUSSION

In this chapter we will discuss the consequences of our chosen approach of development and how the choices we made influenced the results. We will also discuss alternatives to our approaches, what we would do differently if we were to redo the project, as well as the potential for further development of the application.

6.1 Cloud storage - solution

Over the course of the last few months, there were certainly some things that we wish we could have done differently with the project, along with some decisions that we believe will benefit us as well as the project owner. One such decision was local storage vs cloud-based. Early in the planning phase, we thought about opting for a cloud-based solution instead of our current local storage. Pros of this would be that we get a centralized storing location, as well as a more future-proofed product. This would also benefit us, as having cloud experience is highly valuable in today's work-market. On the downside, cloud-solutions are expensive, and often require regular maintenance, not to mention this would increase the complexity of our app together with the estimated time. Therefore, we opted for local storage, as this solution is free and did not require someone to constantly make sure that the cloud-storage is up and running.

6.2 Development approach and programming language

Early in the planning phase we also had to plan our development approach and programming language. Our two options were either Java for Android or React native for cross-platform development. We chose to develop the application in native Android with Java, which is a programming language the developers knew well from previous projects. This allowed us to start developing right away without having to learn a new programming language. If we had chosen to develop cross-platform, or with a different programming language, we might have been able to get the app on iOS, but it would not have allowed us to be able to finish as much of the application as we currently have. If we were to do the project again, we would have chosen the same approach.

6.3 Development method and alternative models

After a couple of weeks, we realized that while we could use scrum to some extent, we would not be able to fully utilize it. The planned sprints fell out due to changes to the scope of the project, as well as different time schedules amongst the group members. This

led us to just work on the open tasks without a specific project management model, where the developers worked on open tasks whenever they had the time. Falling behind on development was a big consequence of not defining a model and following it. The trimming workflow took way longer than needed due to a single person taking responsibility for the whole feature themselves, and the other members doing their tasks and focusing on the thesis afterwards.

We believe the reason that we could not use scrum effectively was due to us not defining a specific model early on. We made a list of tasks, estimated the time needed to complete them, and wrapped scrum around it, and then started development freely with no care for the sprints. A change that should have been made early on would have been to specifically choose scrum and planned the project and workflow in accordance to the agile framework instead of the other way around. Another option would have been to use a different model, for example waterfall, where everybody works sequentially on the project, and there is a clear path that the project should follow. It also does not consider user feedback important, which would have suited us due to our limited user-testing possibilities.

6.4 UI

Over the course of the project, some tasks took more time than estimated, thus were prioritized more than others. This led us to prioritize UI less, and functionalities more. While the layout and overall look of the application is important, we concluded that a prototype should contain most of the heavy-work, and future versions could polish the design of the application, instead of spending large amounts of time fixing bugs, adding large features and so on. Thus, the Technique Analysis app has a simple user interface that consists of non-moving parts, basic buttons and colour-schemes. Our intention is that the project owner can continue making a final product, with our prototype creating a solid base layer that supports further development.

6.5 Choosing core functionalities

Towards the end of the project we realized that we would not be able to finish all the functionalities of the application due to limited time. This included not being able to finish the premium version of the application, as well as the opportunity for the user to add objects such as a ruler, angle measurements and lines to the videos they record. We did however manage to finish all the core functionalities of the application, which we deemed most important. We decided that the core functionalities were the functionalities that allowed for the user to record themselves and compare the video to a previously filmed

video, as well as look at older videos in the archive. Without these functionalities, the application would not meet the goals of the project.

6.6 Further work and improvement

After having the application evaluated by the project owner and a test group of fellow students, we have gotten a clearer insight of the work that remains before the application can be utilized as initially requested.

Right now, a lot of the potential for improvement lies with the graphical interface, as this was the area in which we lacked the most expertise. Therefore, the UI has much room for improvement, both design-wise and making the application more intuitive and easier to use. From the feedback we learned that some of the functionalities were a bit awkward to use and should have been designed better. However, we still recommend prioritizing getting the core functionalities implemented first and opting for a basic UI until the key features have been completed.

The trimming functionality, and the functionality that allows the user to view the newly recorded video in parallel with the reference video, still misses some features for the project owner to use it at its current state. The ability to draw and add objects to the recorded videos should also be prioritized. This is a feature that further assists in this application's first purpose, namely giving feedback on exercise execution. In the evaluation feedback we received from the project owner, this is a feature that he felt is needed. If the prototype was to be further developed, it would be important to consider adding this function.

Future work after having implemented the remaining features and redesigning the UI, would involve moving the application to a cloud-based solution, where user-data would be hosted in order to further improve the app based on those data. This solution is quite popular in today's development process, as you for example get to identify features that are not used that much, which either means that they are bad and need improvement, or just does not fit in the application. A lot of the future improvement relies on having the application go through user tests, given that the application is meant to be used by people of different ages, genders, nationalities and so forth. After having shown the app to our fellow students, we got a small taste of some improvements that could have been made, and with a more diverse test group, we believe that the feedback from said group would largely contribute to making further improvements to the application.

7 CONCLUSIONS

The goal of this project was to develop a mobile application that will allow a physiotherapist to give tailored guidance on different physical exercises to his patients. As we discussed earlier in this thesis, getting an answer if we reached this goal is not within our timeframe, and thus the goal was to create the mobile application Mr. Grøneng requested. However, after a month of developing we realised that this would not be feasible in such a short amount of time and started implementing key features in order to produce a functional prototype.

7.1 Reached goals

After getting feedback we would say that we somewhat reached our goal. There are still some features that we would have liked to implement such as drawings and angles, as well as a few errors to fix after user feedback. Be that as it may, the app contains the key features requested, and thus most of the heavy work is done.

7.3 Different uses

Since the application is mainly about video comparison, we know after having researched similar solutions that the application is relevant in many other settings. Take sports for example, instead of comparing two videos of penalties in football and see if they are alike, you could play two videos with different angles to get a bigger picture of how the athlete shoots the ball. Similarly, you could film a lecture and display a video of the lecturer, and another of the whiteboard or computer screen.

Another situation this application could be used is regular weight training. A lot of people perform exercises wrong either when they start working out, or when trying a new exercise. This could lead to injuries in varying sizes, so if the app were to be filled with reference videos of various exercises, it could be used to prevent physical injuries. Most of the projects group members have experienced a workout related injury that needed care from a physiotherapist, so if enough resources were put into the app, the number of people with muscular and skeletal related injuries would hypothetically decrease.

8 LITERATURE/REFERENCES

- Android Developers. (2019). Platform Architecture. Available from <https://developer.android.com/guide/platform> [Accessed 26.03.2019].
- Android Developers. (2019). Room Persistence Library. Available from <https://developer.android.com/topic/libraries/architecture/room> [Accessed 18.04.2019].
- Android Developers. (2019). User Interface & Navigation. Available from <https://developer.android.com/guide/topics/ui> [Accessed 21.04.2019]
- Atha, H. (2018). Java Vs Kotlin – Which Should You Choose For Android Development. Available from <https://www.moveoapps.com/blog/java-vs-kotlin/> [Accessed 08.05.2019].
- Coach's Eye. (2019). Coach's Eye Video App. Available from <https://www.coachseye.com/> [Accessed 27.03.2019]
- ExoPlayer. (2019). ExoPlayer Home. Available from <https://exoplayer.dev/> [Accessed 21.05.2019]
- FFmpeg. (2019) About FFmpeg. Available from <https://www.ffmpeg.org/about.html> [Accessed 20.05.2019]
- Nielsen, J. (2012). How Many Test Users in a Usability Study? Available from <https://www.ngroup.com/articles/how-many-test-users/> [Accessed 01.06.2019].
- Phillips, B., Stewart, C., & Marsicano, K. (2017). *Android Programming: The Big Nerd Ranch Guide* (3rd ed.).
- Physiopedia. (2018, October 2). Rehabilitation in Sport. Available from [https://www.physio-pedia.com/index.php?title=Rehabilitation in Sport&oldid=198967](https://www.physio-pedia.com/index.php?title=Rehabilitation_in_Sport&oldid=198967) [Accessed 28.05.19]
- Robitza, W. (2017). CRF Guide (Constant Rate Factor in x264, x265 and libvpx). Available from <https://slhck.info/video/2017/02/24/crf-guide.html> [Accessed 13.05.19]
- Scrum.org. (2019). What is Scrum?. Available from <https://www.scrum.org/resources/what-is-scrum> [Accessed 05.04.19].
- StatCounter Global Stats. (2019) Mobile Operating System Market Share Worldwide. Available from <http://gs.statcounter.com/os-market-share/mobile/worldwide> [Accessed 27.03.2019].

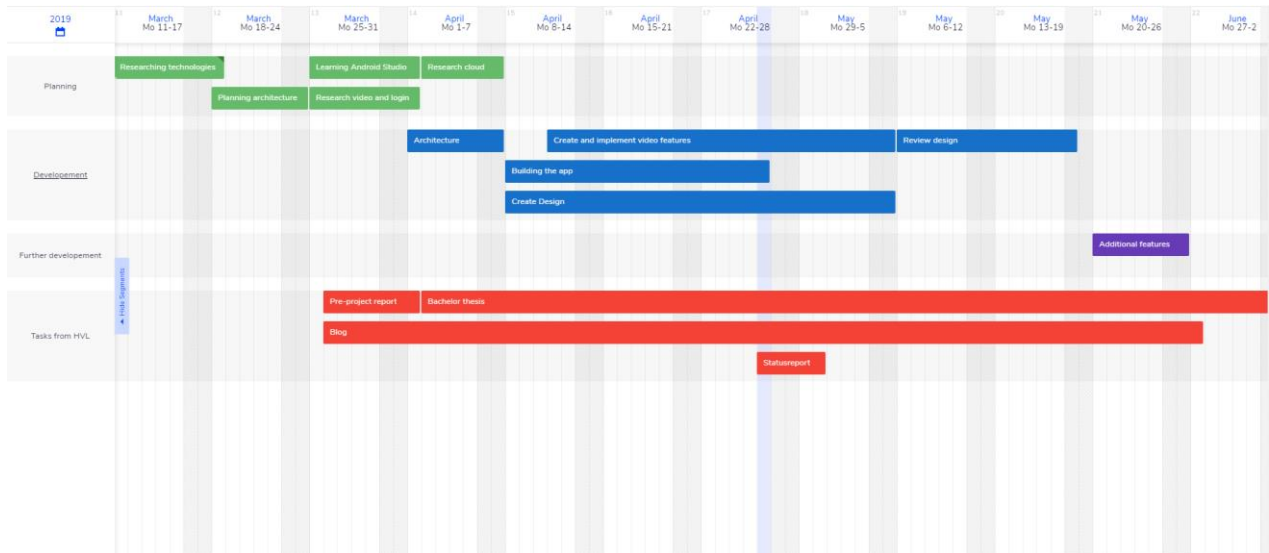
Video Merge - Side By Side. (2019). Iris Studios and Services (Version 1.4). Available from <https://play.google.com/store/apps/details?id=com.irisstudio.videomerge> [Accessed 27.03.2019]

9 APPENDIX

9.1 Appendix A: Initial app design sketch, made by project owner



9.2 Appendix B: Project plan



The tasks are split into four categories: Green are planning tasks, blue tasks are reserved for development, purple tasks are for further development, and the tasks from HVL are set to red.

Part 1 Green

The planning period revolves around the group learning the different technologies, researching multiple approaches, as well as planning the architecture for the application. During this time the M2 assignment is handed in, and the development will start soon.

Part 2 Blue

This period is all about development, as well as implementing several key features. The group will start by building the foundation for the application, and then iteratively implement the features requested by the client. During this time, the pre-project report is handed in, and the group will present their project as well as their progress so far.

Part 3 Purple

The further development period will mostly serve as a safety measure for the group. If the key features of the applications are not finished, this period will focus on finishing the app. However, if the app is done, this period is more about implementing additional features, retouching design, and developing for IOS.

Part 4 Red

The academic part of this project will run continually along the course of the project. Tasks in this period include blogging, pre-project report, presentations, EXPO posters and finally the EXPO itself.

9.3 Appendix C: Risk analysis

Risk	L	C	RF	Measures	Stakeholder	Phase
The application is unusable by the client	2	3	6	Continuous dialogue between group and client, as well as conduct tests on the software	Client	Continuous
The report and project impact each other negatively	2	3	6	Finish the writing as soon as possible, along with a defined working process	Group	Continuous
Larger workload than expected	3	3	9	Work according the project plan, and ensure efficient use of resources	Group	Continuous
Misunderstanding between project group and client, or project group and supervisor	3	2	6	Keep a record of meetings, and get clarification if something is unclear	Group	Continuous
Illness or absence of project group members	2	3	6	Make sure that everybody knows every aspect of the project, as well as being reachable online	Group	Continuous

L = likelihood that the risk occurs

C = severity of the consequences

RF = risk factor (L multiplied by C)

The analysis uses a scale from 1 – 5 to declare the likelihood that said risk might happen, as well as a similar scale to describe the severity of the consequences if the risk occurs (Appendix A). In order to grade the risks, the two factors are multiplied so that we get a risk factor. If the score is below 10, it will not have a big impact if it occurs. If the score is between 10 – 25, a plan to reduce the risk and its consequences should be in place. If it is above 25, it is crucial that these risks are handled, or at least that a plan is in place. Should any of the risks listed in the analysis occurs, the group will refer to this sheet when handling it.