Jukka Koivu

DEVELOPING A MOBILE APPLICATION AS A DIGITAL INTERFACE FOR THE KAWA MODEL THERAPY

DEVELOPING A MOBILE APPLICATION AS A DIGITAL INTERFACE FOR THE KAWA MODEL THERAPY

Jukka Koivu
Bachelor's thesis
Spring 2018
Information Technology
Oulu University of Applied Science

ABSTRACT

Oulu University of Applied Sciences
Degree Programme in Information Technology

Author: Jukka Koivu
Title of thesis: Developing a Mobile Application as a Digital Interface for the Kawa Model Therapy
Supervisor: Veikko Tapaninen
Term and year of completion: Spring 2019 Pages: 55 + 1

The aim of the thesis was to create a working demo product of a mobile application assisting in the delivery of Kawa Model occupational therapy in a clinical setting. The focus was ease of use for the patient and the therapist, especially when they are facing challenges with the traditional pen and paper method. The idea for the topic came during project-based studies when it was suggested as a collaboration with two occupational therapy students by a lab master.

Having interest in learning more about the mobile development and upon consideration, it was chosen that the framework for the project would be React Native, since it can work natively on both iOS and Android devices and alongside Firebase from Google for storing data. NodeJS was used for the local development server in order to test out the application in the real device.

The prototype made progress during the thesis work, but despite that, it did not reach the goal of being fully functional. This was only because the scope of the work was large for a one-person development team to handle. However, the development will continue after this thesis to make a finished demo product of the mobile application since several occupational therapists are of interest to try the prototype.

Keywords: React Native, JavaScript, NodeJS, Android, iOS, Firebase

TIIVISTELMÄ

Oulun ammattikorkeakoulu
Degree Programme in Information Technology

Author: Jukka Koivu
Opinnäytetyön nimi: Developing a Mobile Application as a Digital Interface for the Kawa Model Therapy
Työn ohjaaja: Veikko Tapaninen
Työn valmistumislukukausi ja -vuosi: Kevät 2019 Sivumäärä: 55 + 1

Opinnäytetyön tavoitteena oli luoda toimiva prototyyppi versio tuotteesta, joka on suunnattu mobiili laitteille ja käytettäväksi toimintaterapiassa hyödyntäen Kawa mallia. Keskittyä sovelluksen helppokäyttöisyyteen asiakkaan ja terapeutin väillä etenkin, jos käyttäjällä on vaikeuksia perinteisen kynän ja paperin kanssa. Aiheen idea selkeni projekti lähtöisen opiskelun aikana, jossa sitä ehdotettiin yhteistyössä kahden toimintaterapeutti opiskelijan kanssa kurssin ohjaajan ehdotuksen myötä.

Kiinnostus oppimaan enemmän mobiili sovellusten kehittämisestä ja harkinnan tuloksena alustaksi projektille valittiin React Native, koska sillä on mahdollista tehdä sovellus molemmille iOS ja Android alustoille yhtenäisesti sekä Firebase Googlelta tiedon säilyttämiseen. NodeJS:ä käytettiin paikallisena kehitys palvelimena sovelluksen testaamiseen virtuaaliympäristössä sekä oikeassa laitteessa.

Haasteista huolimatta sovellus eteni hyvin vaikkakin opinnäytetyön aikana täysin valmiiseen prototyyppiin ei päästy sekä työnmäärä oli suuri yhden henkilön kehittäjä tiimille. Siitä huolimatta, että opinnäytetyö tulee päätökseen niin sovelluksen kehittäminen tulee jatkumaan, että prototyyppi tulee valmiiksi.

Asiasanat: React Native, JavaScript, NodeJS, Android, iOS, Firebase

PREFACE

I would like to thank Ulla-Maija Seppänen for introducing me to this interesting and challenging project and Emilia Kuronen and Laura Kämäräinen for helping with the application process and handling the testing with the students and therapists. Also special thanks to Atpunk from the React community, he helped with challenges that seemed overwhelming but were eventually conquered due to his great patience and knowledge, so that the application could progress easily from that point on. I would like to also give my thanks to my thesis supervisor Veikko Tapaninen for his constant guidance and helpful advice regarding the direction and focus of my work. Special thanks to my fiancée Luiza for the moral support and amazing graphical artwork that she did for the Kawa application.

Oulu, 30.01.2019,

Jukka Koivu

CONTENTS

VOCABULARY

AI – Artificial Intelligence is the part of science that aims to create intelligent machines that can act and react like humans.

Android – a mobile operating system developed by Google and available for mobile devices like phones, tablets and smart watches.

APK – a packaging file format that is being used by the Android operating system to distribute and install mobile applications.

Design Thinking – a method used by designers and developers to solve complex problems and to find more desirable solutions for their clients.

DevLAB – it is a project-based training program where students from various fields can take part of developing concepts and product demos in healthcare, environment and technology.

HIPAA – a U.S. law that regulates various standards designed to protect patients' health information.

iOS – a mobile operating system that was developed by Apple and is being used in mobile devices such as phones, tablets and smart watches.

mHealth – a sub branch of eHealth known as mobile health, which refers to the practice of medicine and healthcare with the assistance of mobile devices for example, mobile phones, tablets etc.

NPM –a package manager that can be used to download dependencies, share packages with people or borrow them.

UI design – the process of creating and validating what a user interface looks like and what kind of physical actions it has.

1 INTRODUCTION

In recent years, information technology has been an essential part of the development and improvements in healthcare systems around the world. A lot of work is being put in the research of technologies that will assist and help perfect everything from day to day patient care to studying revolutionary new treatments. These innovations are also very quickly being adapted to different sectors of healthcare such as nursing, various forms of therapy and patient information.

Occupational therapy is one of the fields that can benefit a lot from the large-scale introduction of modern assistive technologies, be it specialized devices or simple mobile applications which can assist both professionals and clients having to make adjustments in their day-to-day lives.

The Kawa Model is a particular therapeutic method used in occupational therapy, in which the client, under the supervision of a professional, draws with pen and paper a river where they subsequently add various elements which best describe their current problems and thoughts. This helps both to more easily identify life challenges that need addressing.

The aim of this thesis is to offer a new way to apply this type of therapy for those situations where it would be recommended to use an easier tool than traditional drawing, be it because of time constraints, a patient's functional impairments or personal preference. The way to implement it was in the form of a mobile application that can be used from phone or tablet devices. The interest for the topic came from wanting to try new tools that are currently relevant in the programming world and learn in the process, while being part of a project that could have an impact in the lives of many.

The thesis consists of two parts, which are split into theory and the technical part where the implementation of the product is presented in detail.

2 PROJECT BACKGROUND AND OBJECTIVES

2.1 Background of the project

The main idea and the background work for the Kawa mobile application were developed and laid out by Emilia Kuronen and Laura Kämäräinen, who are occupational therapists and were also doing their bachelor thesis about it. They had come up with an idea of the mobile application for the Kawa Model therapy. Currently the therapists are applying the method in the traditional way, which is pen and paper. While simple, this can prove to be a challenge for many clients with functional impairments.

Through the references of one of the DevLAB teachers, the Kawa application became a thesis topic for both parties. Also, the DevLAB course provided a setting for testing and validating the paper prototype with users and therapists. This provided helpful insight regarding design features, usability and also possibilities to expand the Kawa method beyond its current limitations.

There has been a previous application for an iOS platform created by the international Team Kawa, but the use of the application has ended since it not working after updates for iOS. From the moment the iOS application stopped being functional, the creators of the Kawa model and other professionals who were using it daily were hoping that a new mobile application would be made for the Android platform, since the latter has a considerably bigger market share, with a lot more users worldwide. With that, occupational therapists could use it again in their work and the popularity of it could expand internationally.

2.2 Objective of the project

The primary objective of the project was to develop a mobile application that would allow the delivery of the Kawa model therapy in a clinical environment by means of a tablet device. An important aspect, which needed to be considered during all stages of the project, was the usability and user experience since very

often patients undergoing occupational therapy have some form of functional impairment, be it physical or cognitive in nature.

Since the concept of the application was laid down in collaboration with two other students from the field of the occupational therapy who were pursuing their bachelor's thesis on the topic, it was essential to complete a working prototype which could be tested in the clinical practice.

The work of the project started in the beginning of January 2018 during the DevLAB where we created a paper prototype version of the Kawa application. It was tested with the students in the school and the occupational therapists gave the prototype to be tested with their colleagues. It has gained interest among the therapists whom the product has been introduced.

2.3 Scope of the project

The application offers a simple way for the users to work with the therapist within the guidelines of Kawa and it allows the therapist to save the sessions for observing the patient's progress over time.

With the use of an Android or iOS handled device, such as a tablet or smartphone, a person can access two different interfaces, which correspond to the two steps of the Kawa process. The first one is a panoramic view of a river and the second one is a cross-section of that river. In each of these, the user will be able to add, drag and drop different visual elements relevant to the Kawa therapy, which the clinician will help to interpret. At any point the therapist can save the client's progress for a later review. The data thus obtained would be synchronized to the therapist's account.

3 MOBILE TECHNOLOGIES IN HEALTHCARE

Over the past few years the use of mobile applications in day to day life is becoming the norm in most parts of the world. These technologies have steadily made their way into the healthcare sector, in hospitals and clinics. In the year 2018 it is estimated that the healthcare application market is valued to be around 28.32 billion dollars and in five years it is expected to almost quadruple, which is a very significant growth. (Mobilebusinessinsights, Date of retrieval 22nd September 2018).

3.1 How mobile apps influence healthcare professionals

Since the technology behind mobile applications has been improving over past few years, the impact has been widespread, changing the way healthcare professionals work but also having an effect on the healthcare market. This also means that the patients can be discharged from the hospital earlier and the healthcare worker can use a secure connection, called HIPAA-compliant secure messaging, to communicate with the other colleagues. They can also retrieve the patient's electrical health records securely using the mobile devices to ease their work. Thus, they use the time that they have with the patients more effectively.

(Mobilebusinessinsights, Date of retrieval 22nd September 2018).

Mobile technology opens the possibility for unified communications and workflow in the clinical environment to provide applications with a good user interface that will improve the efficiency of the healthcare staff. This also entails the use of mobile scheduling tools that allow the healthcare workers to enter their shifts so that the application can memorize these to make future scheduling entries faster. In addition, the staff managers can communicate and notify the workers of possible changes in their schedules. (Mobilebusinessinsights, Date of retrieval 22nd September 2018).

3.2 mHealth in occupational therapy

mHealth can be broadly defined as the usage of mobile devices in the healthcare sector and is part of eHealth as a general field of research and development. mHealth is being promoted worldwide by both the World Health Organization (WHO) and the International Telecommunications Union (ITU) because it has the potential to positively affect the delivery of healthcare, especially in developing countries, thanks to its low cost and high availability. (Dicianno et al, 2015) mHealth is not limited to only mobile applications, but also includes non-screen wearable devices as well as virtual and augmented reality systems.

Since the new innovations and the increase of mobile device use in both everyday life but also in clinical settings, it was only natural that occupational therapists would also take the opportunity to optimize their work, guide their patients towards applications that would assist with their everyday challenges and improve their functional independence. Mobile applications also help in keeping the occupational therapy relevant for the current times. While in the past, activities such as handicraft, were the basis for rehabilitation, these days there is a lot more emphasis put on restoring, to the best extent, a person's ability to function as normally as possible in their everyday life. Mobile devices can serve as communication tools, memory aids, personal organizers and provide extensive accessibility features, often tailored to individual needs. (Verdonck and Ryan, 2008)

The new technological tools are also offering new ways to work with people with disabilities affecting their speech, hand coordination or memory. They can also reduce the need for office visits and allow therapists to assist their clients remotely, while the latter can be in their safe environment at home. (Erickson, 2015)

Maybe in the future the clients could possibly use wearable smart clothes during the therapy, while the therapists can track their client's mood and other changes through special sensors, thus allowing making their work more targeted and more efficient. (OTPotential, Date of retrieval 27th January 2019).

There can also be some limitations and things to be taken into account, for example, gamification could have a negative impact on users who are prone to addictions of this nature. This would mean that therapists would need to consider how much technology the user would be able to benefit from in moderation. (Verdonck and Ryan, 2008). Some therapists might develop a preference for mobile solutions and then discover that certain patients, especially elders or people who are unfamiliar with the technology, could have a lot of resistance to this type of methods.

Overall, the biggest challenge to the large scale spread of mHealth technologies in occupational therapy is the lack of research in the area. As the scientific literature is still very sparse, it is down to professionals to try to make balanced decisions about what the benefits could be for their clients. It is thus very important to start evaluating interventions of this type based on considerations from evidence-based medicine. (Ravenek and Alvarez, 2016).

3.3 AI in healthcare applications

In the future there is a strong possibility that the AI will be part of the mobile applications. However, those working on such projects have a lot of challenges ahead, to be solved and figured out before this is going to happen on a wide scale. There are many things to consider, such as the infrastructure of the hospital and privacy of the patients. There are also concerns that it will reduce some jobs in the healthcare industry. (Mobilebusinessinsights, Date of retrieval 22nd September 2018).

AI could review the patient's electronical health care records and search for the symptoms.

3.4 Benefits of technology in healthcare

New technology innovations bring several benefits in the healthcare sector. They allow hospitals, clinics and therapists to stay on top of various healthcare related trends. These new forms of applications and software can also help to

automatically identify risk factors and recommend right treatments on prevention by comparing the patient's data to many other patients.

This also improves the communications between nurses, doctors and therapists via mobile applications, apps in the patient's rooms that offer faster and more practical connectivity and having meetings in real time. Also, this facilitates medical assessment and treatment for patients who are living in the remote and rural areas and require regular access to medical services. They can remotely be connected to healthcare professionals.

Mobile applications allow people to book appointments to therapist, nurses, doctors or any other health care specialist more easily and have access to their medical records and check their test results. (Elcomcms, Date of retrieval 27th January 2019).

Even there are various benefits of the new innovative technologies, there are still some risks and dangers that come along with them. Are they secure enough since many mobile devices are vulnerable for hacking, malware and viruses when they are not used in a secure environment? Also, they cannot completely remove the human error since it can still happen while using mobile devices. Personal data can also be lost or be stolen easily. (Aimseducation, Date of retrieval 27th January 2019). Most importantly, it is essential to raise the question of acceptability from both professionals and patients, that is, how much could the human factor be removed from the medical act while still preserving trust that the best treatment is being offered.

4 KAWA MODEL THERAPY

The Kawa Model method was developed by occupational therapists in Japan in 1999. The name Kawa means river and is used to describe the life journey of a client from their birth to the end of their life. This model of therapy is useful in helping the client to put their major life events into perspective, as well as projecting desired outcomes for the future. Kawa relies on the use of visual elements as metaphors for emotions, situations and people. There are certain elements that are drawn by the user describing their current life situation, for example a rock represents problems or difficult times and driftwood represents the client's personal attributes and resources they have.

The Kawa Model also reflects the Eastern culture where it emphasizes harmony between the person and environmental factors and believes that it will enhance their well-being. The information will help to widen the space in the client's life flow (river) without needing to remove all the elements from the river and shaping it to a smoother and stronger life. (Kulttuurit kohtaavat toimintaterapiassa, Kuronen and Kämäräinen, Date of retrieval 26th January 2019).

4.1 Traditional method

Traditionally, during the first phase of a Kawa therapy session, the client will draw a river on paper, to which they will add elements like rocks, driftwood, trees, which are all symbols for their current life events.

In the second phase they are asked to draw a cross section of the river, the focus being taken on identifying any deeper issues. The therapist will encourage the client to discuss the significance of their drawings, in order to establish a better understanding of their connection to current challenges and to work out ways of helping the client better move towards their life goals.

FIGURE 1. Example of traditional method of Kawa Model

The client has the river and the elements they want to be drawn in the scene and including the line where they think their life is currently situated as shown in the Figure 1.

4.2 Technology as a tool for Kawa model therapy

In the modern age of technology, health applications, smart assistive devices or internet delivered therapies are becoming more widespread every year, both for personal and clinical use. This will enable therapists to have a better track of the client and possibility to even work remotely with the clients. Also, it will make the therapy sessions easier for the people when they not have to draw since they can create a river and choose a different theme for it to make it more enjoyable and more open and relaxed. Since the user can drag and drop the elements and move the elements with their hand and add as many, they need to describe their life flow in the river without the need of paper and pen.

4.2.1 Benefits

There are many use cases where the Kawa mobile application can benefit the clients while they are in the therapy session. Those who cannot or do not want to draw can use the application to create their own river where they can add traditional elements to it as well as new elements which would broaden the range of possible interpretations, making an in depth analysis and allowing for cultural differences to be expressed. Because the elements can be easily added, moved or removed, corrections can be made in just seconds. The whole intervention applied in this matter would take significantly less time than when the traditional method is used.

Patients with motor deficits who have their hand dexterity affected, might find it a lot simpler to use a mobile device, or even guide the therapist to build the river with them.

The saving feature is useful when tracking the progress between sessions. The therapist can get synchronized data from multiple patients, look at the rivers whenever needed and pick up the session from wherever it left off last time.

4.2.2 Previous Kawa application

On the market there was, at some point in time, an existing application for occupational therapists about the Kawa model. This application was acknowledged by the creators of the Kawa. There is not much information available about the application itself currently, other than it allowed the user to create their river, view their river and choose various themes for their river. There were also similar elements that the user could drag and drop to the scene, adjust the size of the river bottom and record a voice memo.

It was developed for the iOS platform but not for Android devices. For unknown reasons, the application broke after an iOS device update and the creators have not fixed it since.

5 PROTOTYPING

A prototype is an early version of a product, which helps the creator learn more about how the users interact with it, to test a concept and serve as a basis for other processes. The main benefits of prototyping will be presented in the following chapters, as well as information on how a paper prototype is created. The prototype can be made from anything, depending on the needs and possibilities, from natural materials to actual physical websites where the user can click and move around to browse the content. (Usability.gov, Date of retrieval 26th January 2019).

5.1 Prototyping

Creating a prototype is an essential part for the product designers who are aiming to reach their wanted end results. First it is important to brainstorm ideas for the design of a product prototype for example if it is a mobile phone or tablet device, and which type of features are included in prototyping.

In the first phase of creation it is usually formed a basic frame of the application with any form of the paper and creating elements, such as clickable buttons or using tools like Marvel, InVision, Sketch Principle.

The second step of the process includes the interaction part where the prototype of the product is taken to be tested with real users to find if solutions are solving the problems of the end user and to find out how the users are interacting with the application

This allows the validation of the design and flow of the application before the actual development of the product will start.

Developers should use prototyping as part of different stages of Design Thinking. It can be used as an ideation method since it allows the developer as well as the user of the prototype to explore and find alternative solutions. The Figure 2 shows the phases of design thinking.

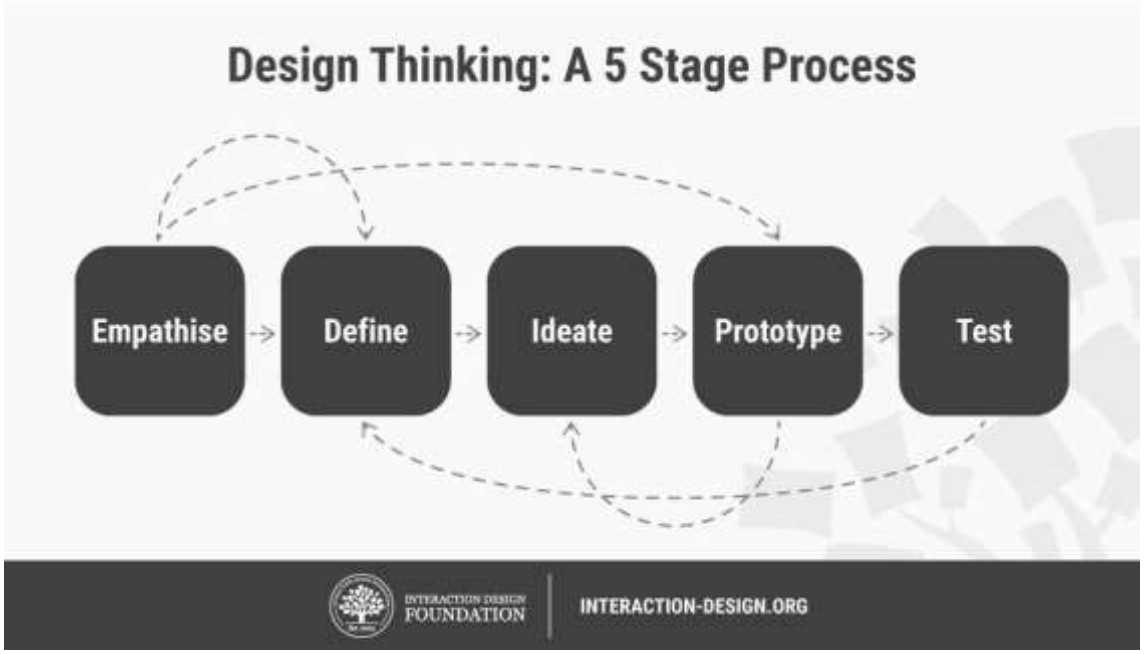

FIGURE 2. Stages of design thinking

Prototypes fulfill certain purposes, such as exploring problems, ideas and what kinds of opportunities would be possible for the product. Learning and understanding more about the solution to the problem by engaging with the end users and stakeholders offer a great experience and facilitates design decisions. This can be even used to sell the idea to the stakeholders or to inspire markets to take new sights on thinking and doing. (Interaction Design Foundation, Date of retrieval 26th January 2019).

If development is planned without consideration for the testing it can become costly in terms of both time and money. Without knowing what the users think of the product, whether the application is solving their problems or whether there are changes to be made and then get retested for usability and functionality, the outlook is uncertain. In conclusion, when doing tests with the prototype, the feedback from the users will increase the value of the product and offer a better understanding and sometimes a new perspective for the development and design team. That will prompt them to address any requirements and make necessary changes before the product is launched, or otherwise discontinue the development process.

5.2 Paper prototyping

Paper prototyping has existed for a long time in the world of development, for anything from software, mobile applications or anything else that can be visualized in the form of paper. It is also an inexpensive way to get feedback from the users before starting to program the software or mobile application.

In short, paper prototyping is an early sample design of the product. When creating a paper prototype for a mobile application, there would be a simple frame of a phone which will contain all the paper screens.

The users that are testing the prototype will be given a paper which contains the user interface of the application and elements like buttons, menu and content on it, which would illustrate user interactions. The goal is to accurately show how the real application would function. For example, if they press a certain button, the paper would have to be switched to the appropriate one, which would correspond to the behaviour of the app when pressing the button.

This will offer an interactive way to gain feedback from the users of the product before starting to program it.

One can use the following things to create the paper prototype.

- Paper
- Cardboard
- Scissors
- Glue
- Pencils
- Erasers

One can craft the prototype from a very simple one to very complex. Depending on the time available, the complexity of the application being prototyped and the level of detail necessary to accurately illustrate the features. It can take some time to design draw and cut the shape of the frame and elements. (Justinmind, Date of retrieval 22nd of September 2018).

5.2.1 What you cannot be got from paper prototyping

Even though paper prototyping is a great way to test an application, unfortunately it cannot offer the same feeling as when interacting with it on a real device. Still, this method offers a good way to get feedback from users before going further in the development.

The following chapter will go through the steps that were taken for the creation of the application and what things had to be considered in the process of planning the application. (Justinmind, Date of retrieval 22nd of September 2018).

5.3 Benefits of prototyping

It is cost effective to have changes to be made in the early development process rather than integrating them later, when the product is already developed. This is a good reason to consider creating prototypes early in the development. Prototyping also allows the developer to gather data and feedback from the users in the early stages of the development process while still planning and designing the mobile application.

One of the advantages of doing prototyping is getting user feedback in early stage of the development if you are uncertain whether your product is solving the problem of the end users and if there is space for it in the market. (Usability.gov, Date of retrieval 26th January 2019).

6 TECHNOLOGIES

This chapter will focus on presenting the technologies that have been used in the creation of the Kawa application and explain in more detail how different languages and services were integrated in this project. These technologies are:

- React Native framework
- JavaScript (ES6)
- Firebase
- NodeJS
- Jest

6.1 React Native

React Native is a framework that was announced in 2015 by Facebook. It is built on top of ReactJS, which uses JavaScript. Both ReactJS and React Native have been developed by a team from Facebook. With React Native the developer can build applications for both iOS and Android platforms without needing to know how to learn Java or Swift. Since it is UI focused, it can help mobile applications to load faster and feel smoother. In the past few years, the framework has been gaining popularity among developers because of the possibility to work on both platforms at the same time. It is also an open source which means that there can be plenty of offer and other platform support e.g. Windows.

There are benefits for using React Native in the development of the mobile application, for example,

- Cross-platform
- Performance

Basically, React Native is offering a simultaneous development for both platforms using a singular code base. In performance React Native does well since it is compiled into native code that works on both platforms. React Native has the

ability to reuse components and it is simple to maintain. (Medium, Date of retrieval 25th January 2019).

6.2 Firebase

Firebase is a large scale platform made by Google, which suits various people's needs for creating mobile, small or larger scale applications or anything else with it.

It also works as a real time database and cloud-based data storage. Data is not the only thing that can be stored. There can be also different types of files e.g. images, audio and videos that can be stored to the Firebase storage. It has a built-in authentication system that allows the developer to create easily login and registration. For example, if it is required in the development, authentication can be anonymous. It can require an email and password or even use social media accounts e.g. Facebook or Google.

After creating the project for the product, one can get access to various useful tools. One such example is Analytics, where a person can get detailed information about how the users have engaged to the application. Another is Quality, where the developer can view crash reports and follow the performance of the application. Yet another convenient tool is Test Lab, which allows to test applications across devices when downloading the APK of the product for Android or iOS. It also allows the developer to create cloud messaging or in-app messaging, so they can send users notifications if necessary.

6.3 NodeJS

NodeJS is an open source asynchronous JavaScript runtime that is designed for building scalable network applications. In the project Node was used as a development server to test an application in a localhost server. It also comes with NPM which is being used to install various dependencies to the mobile application project for example, in this case the Kawa application. (W3Schools NodeJS, Date of retrieval 25th January 2019).

6.4 Jest

Jest is a JavaScript Testing framework that allows the developer to create a test environment for the project with ease and is designed to ensure correctness of the code. For example, the developer can run tests and then use “—coverage” to gather it up as a report from the whole project.

FIGURE 3. Example of jest in action

It will run tests that are set up for the files and it will notify with red all the tests that have failed. It shows what lines are the issues and from what file they are from after the tests have finished. The Figure 3 shows an example of jest.

7 IMPLEMENTATION

The Kawa mobile application project was developed using the technology described in the previous section of the thesis. In the end of the thesis the product should be a finished demo version of the application for the occupational therapist to use.

The first step was to set up a choice of editor and required plugins to aid with the work and programming. Setting up and initializing the React Native project was also required, as well as setting the project to GitHub, so that it can be under the version control. In case any mistakes were made, this made it easier to roll back to the previous version. All of this will be reviewed more closely in later chapters.

7.1 Planning

The creation of the application started with a plan that was made along with two occupational therapists. Designing a paper prototype was quite essential in any circumstance, since it helped to test out and ask feedback from users about the features or functionalities to improve and to get ideas on the possible changes that would make the application more accessible, engaging or relevant for the therapeutic process.

The application was then planned on the whiteboard, in order to make a list what the application needs and a checklist of the things to be developed. Also, at this stage it was first considered what kinds of technologies could be used in the development of this product. The Figure 4 shows the whiteboard that was used for the project planning.

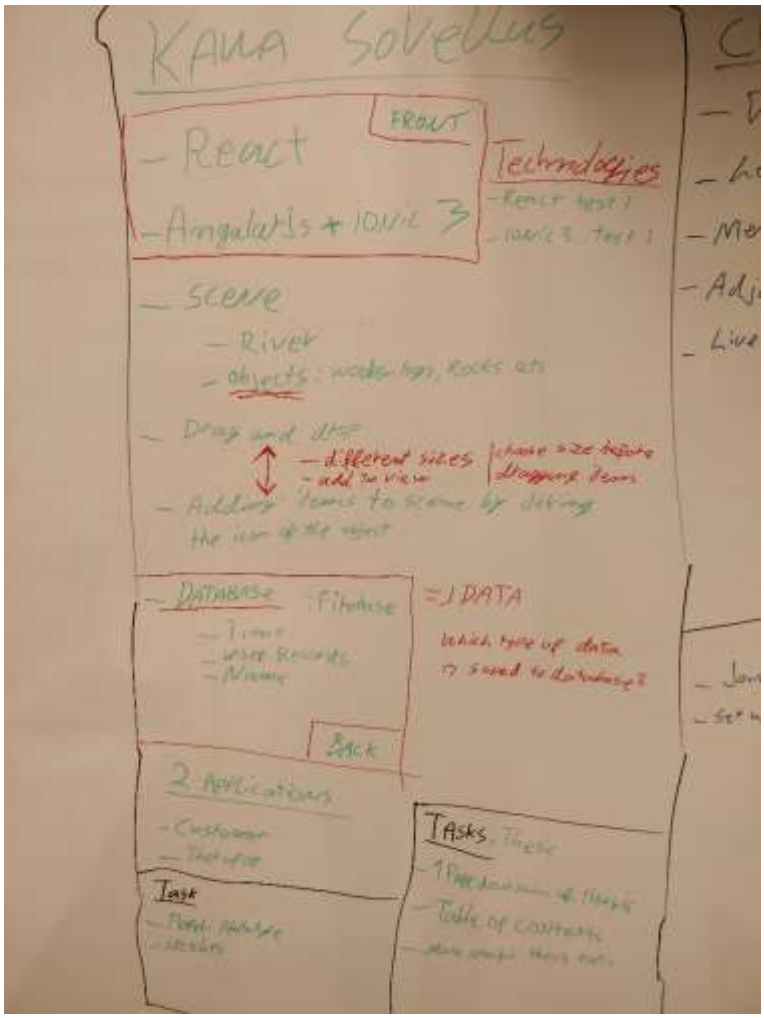

FIGURE 4. Project plan board

Having this as a starting point, it was easier to track changes being made, have an overview of the progress and prioritize tasks. This was a simple way to plan the development of an application of this small scale. Also, it allowed for a clear view of the requirements, thus helping in research the technologies that were best suited for the development.

7.2 Paper prototype

During the first phase of the development of this project, the prototype version of this application was created using paper and cardboard.

FIGURE 5. Paper prototype of Kawa application

The paper prototype method was used to get feedback from the users that were testing the application and this test included students and therapists. The Figure 5 above illustrates a part of the paper prototype. After this process was completed and results collected, the development of the real application started. This will be presented throughout the following chapters.

7.2.1 Feedback and results

After all the testing was completed, the results of testing were weighed based on the feedback received and the most valuable findings were extracted.

What was found is that most of our testers like the idea of this application and the therapists were very interested in testing it when the real prototype is ready to be used.

As all the results were mainly positive, it gave a good idea what elements were most wanted and requested to be included in the application and what kinds of considerations were of importance while programming the product.

7.3 Project management

The project management platform used during the development is called Taiga. Its advantage was that it is free, open source and simple to use.

In the later stages of the development, another programmer joined the team, and this provided the opportunity to make use of this type of tool. User stories for the project were set up and the first sprint for the product development was formed, listing tasks that needed to be completed, having a deadline set for them and allocating tasks to the team members. See the Figure 6 for the Sprint 1 view.

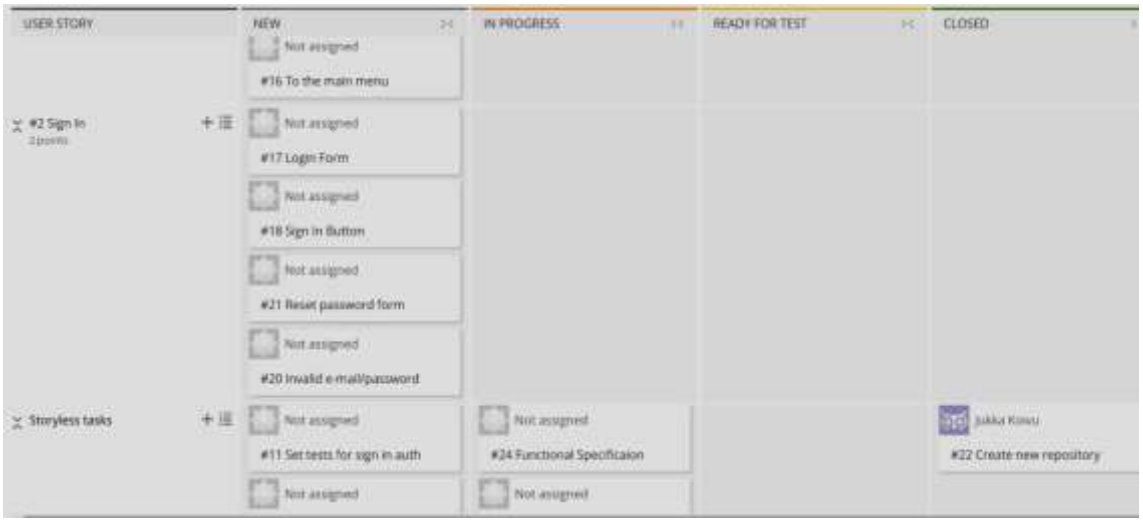

FIGURE 6. Taiga project management platform. Sprint 1

Also, there is going to be a written function specification documentation and a test plan for the project.

7.4 Software

7.4.1 Visual Studio Code

Visual Studio Code is the editing software that was used to create the product. It also comes with an integrated GitHub option, but it was decided to use Git with Git bash instead, this will be explained later.

Next in line was adding a number of plugins to help with the development after the installation was completed. The following plugins were added and used during the process:

- Color highlight
- ESLint
- Auto import
- React native tools

Color highlight was used when creating the styles for the product, for example when choosing the right color and having a preview of the color after adding the color tag. Within the Figure 7 is seen the blue and white highlight for the corresponding color tag.

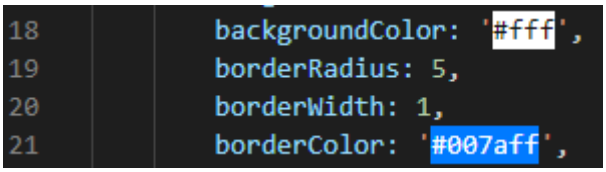

FIGURE 7. Example of color highlight plugin used in the styling

ESLint was used to find problems in the code without needing to execute it.

Auto import on the other hand was used to make importing files and other imports easier and faster. If adding a new file or writing a piece of code that have not been imported, it will automatically import it to the file.

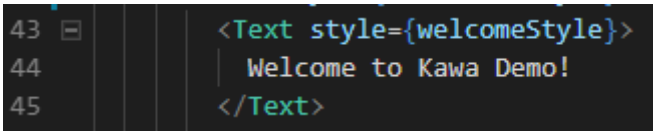

FIGURE 8. If text component was missing auto import will import it

React native tools is a handy plugin to have when developing an React Native mobile application. It will give code hinting while writing a code, for example when writing <Text>, it will present an option to auto complete that code.

7.4.2 Xcode

During the development process of the application, Xcode software is not used much, even though the application is made for both iOS and Android platforms since it requires a Mac laptop or desktop. However, if the opportunity arises, the product will also be built for the Apple devices such as tablets and mobile phones.

If this option became available at some point in the future, it would be necessary to install the software called Xcode from the Mac App Store. Then it would need to install right version of the command line tools for the software. Also, the purchase of the Apple Developer License is required for sharing the application.

7.5 Application set up

There were also other fundamental tools and software that needed to be installed before the actual programming of the mobile application could take in place. This will be explained next.

7.5.1 React Native project

For the start of the React Native, the following guide was followed from the documentation of React Native and what dependencies were required to install before starting. Firstly, installing NodeJS and its command line tools, e.g. npm, were used to install dependencies to the application. Also, the installation of Python2 and a recent version of the Java SE Development Kit (JDK) using Chocolatey (Package manager for Windows) were necessary.

After this step came the installation of Android Studio. During the installation process, the following things were selected to be installed.

- Android SDK
- Android SDK Platform
- Performance (Intel HAXM)
- Android Virtual Device

It is also necessary to go to systems settings and open the Android SDK to choose support from Marshmallow all way up to the latest Android version.

React Native tools required setting up environment variables in order to build mobile applications that are using native code. The same also applies for the JDK. See the Figure 9.

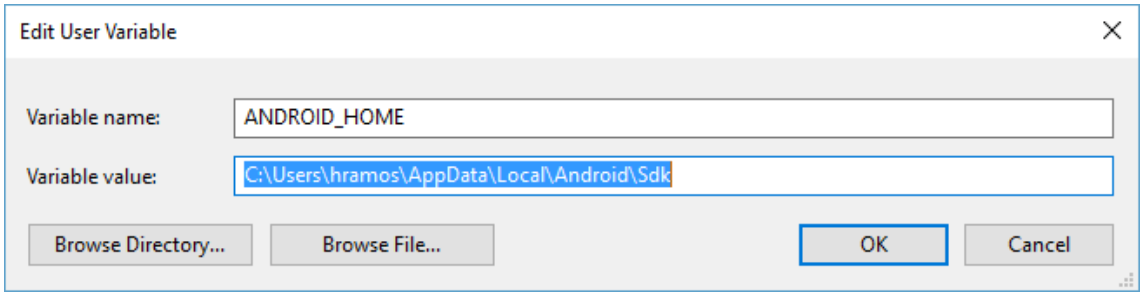

FIGURE 9. Setting environment variable for Android SDK

Next in line was creating the React Native project itself. This was done by using the command line interface and using the following command: “react-native init Kawa”, which will create a folder called Kawa that holds all the files required for the start template of the application after the installation process is completed. In a later chapter, it will be explained how building the product for the mobile device took place.

For the Mac and iOS devices, the set up follows a similar process as in Android, with minor differences e.g. using the Xcode and requiring the license.

7.5.2 Version control

After the application was created by using the React Native command tools it was also time to add a version control to the project. For this project it was set as a private repository, which will be holding the project code. This allows the version control and reversing the code back if something goes wrong.

When working on different tasks e.g. making sign in form or creating the river functions, those had their own branches separate from the master branch to avoid any complications when working in a team.

After each task was ready and tested, a pull request was created and reviewed before merging the request to the master branch to avoid and fix any appearing merge conflicts.

Git command line tools were used in the version control and the Git Bash as the main command panel for creating branches, adding files, creating commits and pushing the changes to a corresponding branch. Most used commands used for this project were:

- git checkout -b sign-in (depending of the task)
- git add “.” (for adding all the changes) or the specific file
- git commit -m “sign in form created”
- git push origin sign-in (or master)

After the task is ready and committed to the repository, the developer can create a new pull request, which is shown in the Figure 10.

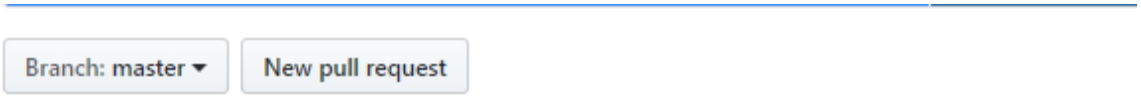

FIGURE 10. Create pull request in repository

7.5.3 Firebase real time database

There was one more thing to set up before the programming would take place and that was setting up a Firebase project. The authentication method was set to email and password and was initialized to the project by adding it to a Firebase configuration file, so that the application can connect to the real time database. In the Figure 11, the empty configuration file which is filled with the right data is shown.

These details are created by Firebase when a project is created in there. This configuration file was named as “Credentials.js” and was added to the app folder.

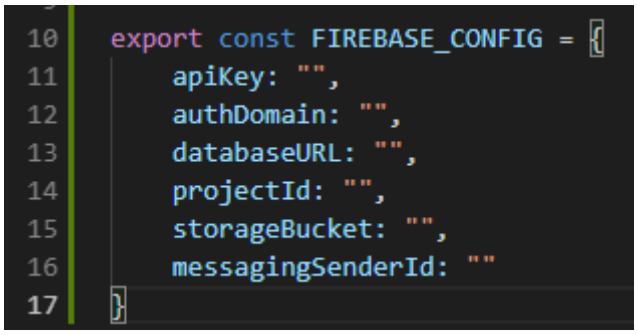

FIGURE 11. Firebase config file that connects application to database

7.5.4 Debugging

React Native Tools also include a feature called remote debugging which can be enabled in both the simulator or the actual physical device, depending on which one is used. The developer menu can be opened by shaking the device if using the physical device e.g. mobile phone (The Figure 12 shows the developer menu) and then start remote JS debugging which will open a local host in the browser http://localhost:8081/debugger-ui/. After that is done, the developer can open console view there, where all of the errors and console logs can be seen.

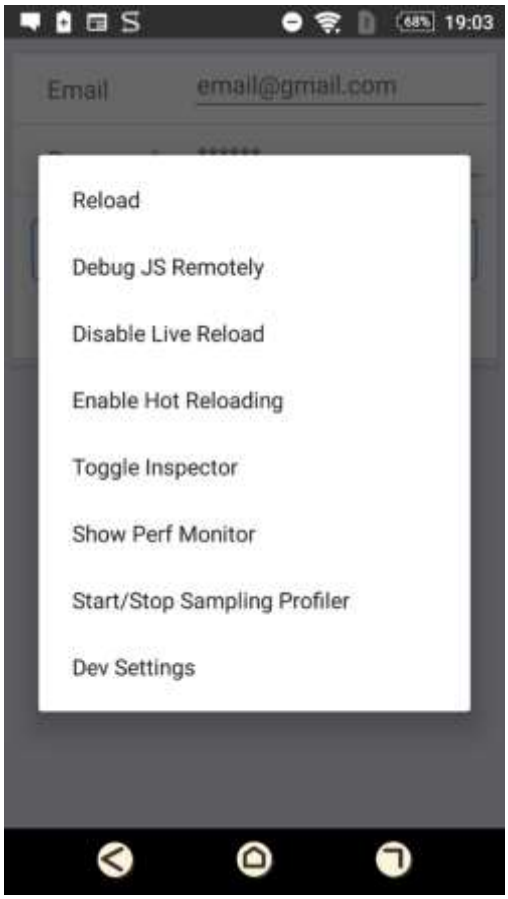

FIGURE 12. Debug menu opened

7.6 User interface

For the design of the user interface the developers have decided on something simple, which would be easy to use by both the clients and the therapists. The colors that are being used in the application are varied, but not too bright or in high contrast so that they do not irritate the users. They are reminiscent of the ones that would be used in the traditional, pen and paper technique.

As it has been mentioned, the idea here revolves around simplicity and accessibility, so that even those people who cannot use their hands very well to draw on the paper could use the prototype with the touch pen to move the elements around the river and interact with the application.

7.7 Kawa application

The Kawa application consists of authentication, menu, river and drag and drop functionality. These are explained more thoroughly in the chapters 7.7.1-7.7.9.

7.7.1 Authentication

The authentication of this product is handled by the Firebase. There is a login form that allows the therapist to sign in and sign up to the application to use it or reset their password if they have forgotten it.

Firstly, what was needed was to create a login form where the therapist signs into the application. The Login form can be seen in the Figure 13.

FIGURE 13. Sign in form

For this it was necessary to create email and password input fields since in the Firebase project, email was chosen as the authentication method. Then the login and sign up buttons were created so that the user can sign into the application after they have created an account in the sign-up view. If the password is forgotten, they can retrieve it by going to the reset form and submitting their email which shortly goes to the email.

The functions handling the authentication processes are simple and come straight from the Firebase documentation.

In order to login, the therapist needs to add their email, username and password when they are creating their account for the first time. After that, the user is created to the Firebase. Afterwards, in the login form the user gets authenticated by dispatching the email and password, which is handled in the authentication actions file. The Figure 14 presents a piece of code that handles this event.

```
export const loginUser = ({ email, password }) => {
    return (dispatch) => {
        dispatch({ type: LOGIN_USER })

        firebase.auth().signInWithEmailAndPassword(email, password)
            .then(user => loginUserSuccess(dispatch, user.uid))
            .catch(() => loginUserFail(dispatch))
    }
}

export const createUser = (email, password, username) => {
    return (dispatch) => {
        dispatch({ type: CREATE_USER })

        firebase.auth().createUserWithEmailAndPassword(email, password)
            .then(user => {
                if (user.uid) {
                    firebase.database().ref(`/users/${user.uid}/`)
                        .set({ username: username })
                }
            }
            )
            .then(Actions.pop())
    }
}
```

FIGURE 14. Login & create user functions

Resetting a password has its own simple function that is called when the reset password is request by the user. The Figure 15 illustrates a function that handles the resetting of the user's password.

```
31      resetPassword () {
32              firebase.auth().sendPasswordResetEmail(this.state.email)
33              .then(() => {
34                  ToastAndroid.show('Email was sent to reset your passw
35              })
36      }
```

FIGURE 15. Function handling password reset

The reducer will handle the form state which is set to empty string to have an empty input field when the user opens the application and after the sign-in and sign-up the forms are cleared. This is handled by a switch case which returns true or false. As in this case, it is not true, it returns this error case “login_user_failed”. This can be seen in the Figure 16.

```
17  export default (state = INITIAL_STATE, action) => {
18      switch (action.type) {
19          case EMAIL_CHANGED:
20              return { ...state, email: action.payload };
21          case PASSWORD_CHANGED:
22              return { ...state, password: action.payload };
23          case LOGIN_USER:
24              return { ...state, loading: true, error: '' };
25          case LOGIN_USER_SUCCESS:
26          console.log("LOGINTEST:", action.payload)
27              return { ...state, ...INITIAL_STATE, user: action.payload }
28          case LOGIN_USER_FAIL:
29              return { ...state, error: 'Authentication Failed!', passwor
30          default:
31              return { ...state }
32      }
33  }
```

FIGURE 16. Switch case handling the authentication actions

7.7.2 Main screen

The main screen of the application consists of three elements which are buttons and a sign-out option to logout which will redirect the user back to the main view. (The main view can be seen in the Figure 17)

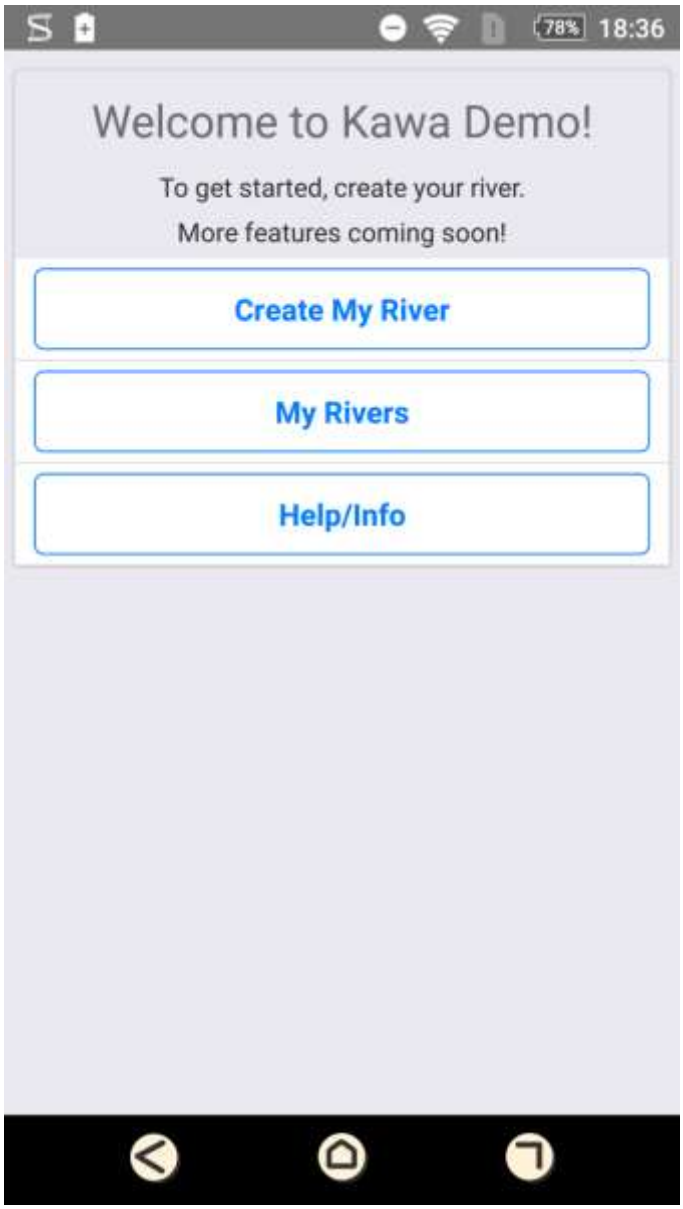

FIGURE 17. Main view of the application

Through these buttons, the user can navigate to create a river, continue their river on the next therapy session if needed and read information on how to use the application, which is mostly aimed for the therapists.

Buttons are created using the text component and adding own styling to give them a button like look. For the navigation of the mobile application it was used “react-native-router-flux” and set actions to take the user to a new view and the

logic is set in the “Router.js” file and importing the views. In the Figure 18 the routing structure can be seen.

```
const RouterComponent = () => {
    return (
        <Router>
            <Scene key="root" hideNavBar>
                <Scene key="auth">
                    <Scene
                        hideNavBar
                        key="login"
                        component={LoginForm}
                        initial
                    />
                    <Scene
                        key="signUp"
                        component={SignUpForm}
                    />
                    <Scene
                        key="resetPassword"
                        component={ResetPassword}
                    />
                </Scene>
                <Scene key="client" hideNavBar>
                    <Scene
                        key="menu"
                        component={Main}
                    />
                </Scene>
```

FIGURE 18. Navigation logic using react-native-router-flux

7.7.3 River

In the prototype of the application the user can create the river where they can add various elements that describe their life. The saved river will be stored in the real time base for the later continuation of the therapy. In the Kawa model there are two different phases of the river which will be explained in the following chapters.

7.7.4 Create river

During the prototype of the application, in the main menu of the application, the user presses the button “Create my river” to create the river as shown in the Figure 17 in the Chapter 7.7.2. A new scene of the river is created for the user who has signed in with the username that was assigned during the sign-up process.

When the user creates the river, a following function is called, it creates a new scene for the user and allows the user to return to that same scene when they come to the therapy again. After the button is pressed the user is taken to their created river scene. The Figure 19 shows the function that gets called.

```
async createScene(scene) {
  var user = await firebase.auth().currentUser
  let newSceneKey = await firebase.database().ref(`users/${user.uid}
  this.props.setActiveScene(newSceneKey)
  Actions.main()
}
```

FIGURE 19. Create a scene for the user

7.7.5 Phase one of the river

The user can interact in this scene of the river with a fab button that contains all available elements that the user can add to the scenery. In this Phase one, the user can add the following objects to the river:

- Tree
- Rock
- Flower
- Wave

The elements were created as their own components that the user can individually add. The Figure 20 presents a tree component.

```
import React from 'react'
import Draggable from './Draggable'

// Only modify these if the source image has been modifier
const WIDTH = 73
const HEIGHT = 100

// Modify this to change the base size of the image
const BASE_SCALE = 0.75;

no references found for Tree
export default Tree = ({ scale = 1 }) => (
  <Draggable
    itemLabel={'tree'}
    width={WIDTH * BASE_SCALE * scale}
    height={HEIGHT * BASE_SCALE * scale}
    source={require('../../assets/images/tree1-01.png')}
  />
)
```

FIGURE 20. Tree element

In the element component there was also “draggable.js” which handles the drag and drop functions for all the components.

After adding an element, the user can drag and drop the elements around the scene while coordinates are being tracked. Another element is also going to be a vertical line that can be moved horizontally. The Figure 21 illustrates part of the elements that the user can add to the river.

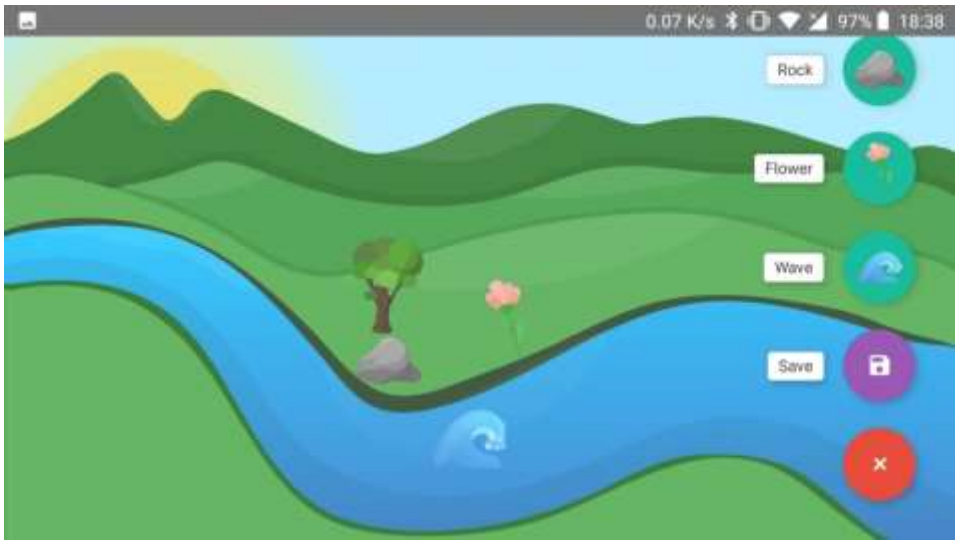

FIGURE 21. Phase one view of the river

After the user has added and placed all the elements required describing their current life situation, the user can save their scene and they are taken to the second phase of the therapy which will be discussed in more detail in the Chapter 7.7.6. All the elements coordinates are saved locally into the database, so the user can continue the therapy if needed.

```
23      onSavePress() {
24          this.props.saveDraggable(this.props.treeX, this.props.treeY, this.p
25          this.props.saveDraggable(this.props.rockX, this.props.rockY, this.p
26          this.props.saveDraggable(this.props.flowerX, this.props.flowerY, th
27          this.props.saveDraggable(this.props.waveX, this.props.waveY, this.p
28          Actions.phasetwo()
29          ToastAndroid.show('River was saved', ToastAndroid.SHORT)
30      }
        1 reference
31      onAddTree() {
32          this.props.addTree(this.props.treeX, this.props.treeY, this.props.u
33      }
```

FIGURE 22. Implementation of the saving coordinates and adding element to view

This function handles saving the coordinates of the element components in the scene when the user presses the Save button in the fab. Also, the function below will handle the adding tree elements to the river scene. Both functions can be seen in the Figure 22.

7.7.6 Phase two of the river

After the user has completed the first phase and saved their river scene, they are taken to the phase two where they have a close-up scene of the river. In this view the user can also add elements but this time there are new types of elements that they can add. The phase two river scene can be seen in the Figure 23.

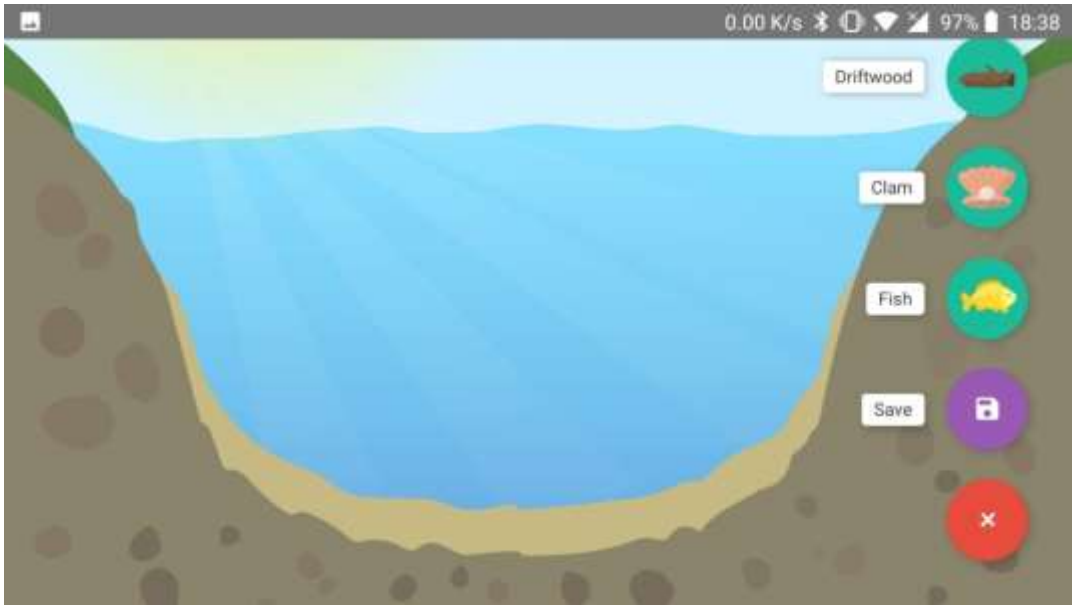

FIGURE 23. Phase two view of the river

It follows the same functionality as the first phase and here the element components that the user can add to this scenery are:

- Rock
- Driftwood
- Fish
- Plants
- Clam

After the client is ready and saves the scene, the data is dispatched to the database and locally for the next therapy meeting if necessary and it is taken to the menu view.

7.7.7 Drag and drop elements

Creating the drag and drop functionality for the components that are tracked in real time when the user is dragging and dropping the elements to the scene and then being saved to database. It was also the most time-consuming part of the project since it required searching from different sources and finding the right person to ask advice from in the React Native Discord group.

```
export const saveDraggable = (x, y, user, item) => {
    return (dispatch) => {
        FirebaseHandler.setCoordinates(x, y, user, item)
        .then(
            Actions.client()
        )
    }
}

export const setCoordinates = (coordX, coordY, itemLabel) => {
    console.log("X:", coordX, "\nY:", coordY, "\nITEM LABEL:", itemLabel)
    return function (dispatch) {

        if (itemLabel == 'tree') {
            dispatch({ type: SAVE_TREE_XCOORDINATES, payload: coordX })
            dispatch({ type: SAVE_TREE_YCOORDINATES, payload: coordY })
        } else if (itemLabel == 'rock') {
            dispatch({ type: SAVE_ROCK_XCOORDINATES, payload: coordX })
            dispatch({ type: SAVE_ROCK_YCOORDINATES, payload: coordY })
        } else if (itemLabel == 'flower') {
            dispatch({ type: SAVE_FLOWER_XCOORDINATES, payload: coordX })
            dispatch({ type: SAVE_FLOWER_YCOORDINATES, payload: coordY })
        }
    }
}
```

FIGURE 24. Setting coordinates in reducer

Here the function receives both x and y coordinates from the components that the user is currently dragging and dropping and then being dispatched to the handler function that can be seen in the Figure 24.

```
export default class FirebaseHandler {
    1 reference
    static async setCoordinates(x, y , user, item) {
        await firebase.database().ref(`users/${user}/${item}/`)
            .update({
                coordX: x,
                coordY: y
            })
    }
}
```

FIGURE 25. Handler that saves coordinates to database

When the client presses the Save button from the menu this function will be called. It will save and update the x and y coordinates of the elements. This is shown in the Figure 25.

7.7.8 Load saved river

Since the river is being saved to the database and locally, it allows the client to continue making changes to their scene of river if they need to continue their session with the therapist. Saving locally also helps if the user accidentally closes the application and opens the application again. Since the data is locally saved into the store, the user can continue where they left off.

The user can go to their previous river from the main menu, where they will have an option to continue the river.

During the new therapy session, the client who goes through the process of the two phases again and saves the coordinates of the components again. This time the coordinates will be updated to the existing coordinates of those elements in the database.

7.7.9 Information view

The information view was done as a scrollable view where the therapist can scroll through various information about each part of the application. For example, what happens in the login, different scene views, the function of each button that they

find in the menu and when the client is pressing the button. There are also details of how to work with each of the phases during the therapy session. The Figure 26 shows the info view.

FIGURE 26. How to use application info view

For making the info view, ready components were used from the react-native dependency, as presented in the Figure 27.

```
26      <ScrollView>
27          <CardSection style={sectionStyle}>
28              <Text style={textStyle}>
29                  Alkunäytöllä terapeutti täyttää ensin asiakaan nimen,
30                  <Text style={textBold}>Name</Text> = Terapeutti luo as
31                  <Text style={textBold}>Huom.</Text> Ainoastaan terapeu
32                  eikä asiakas pääse näkemään muiden asiakkaiden tuotoks
33              </Text>
34          </CardSection>
```

FIGURE 27. Scrollable info view

7.7.10 Debugging & problem solving

Sometimes, during the development process of the application, there were errors that kept showing up every now and then. To solve these, console logs needed to be set up to functions where the problem could have been coming from, using the remote debugging tool that opens in the Chrome browser to allow the viewing of those console logs. A view of the browser console log can be seen in the Figure 28.

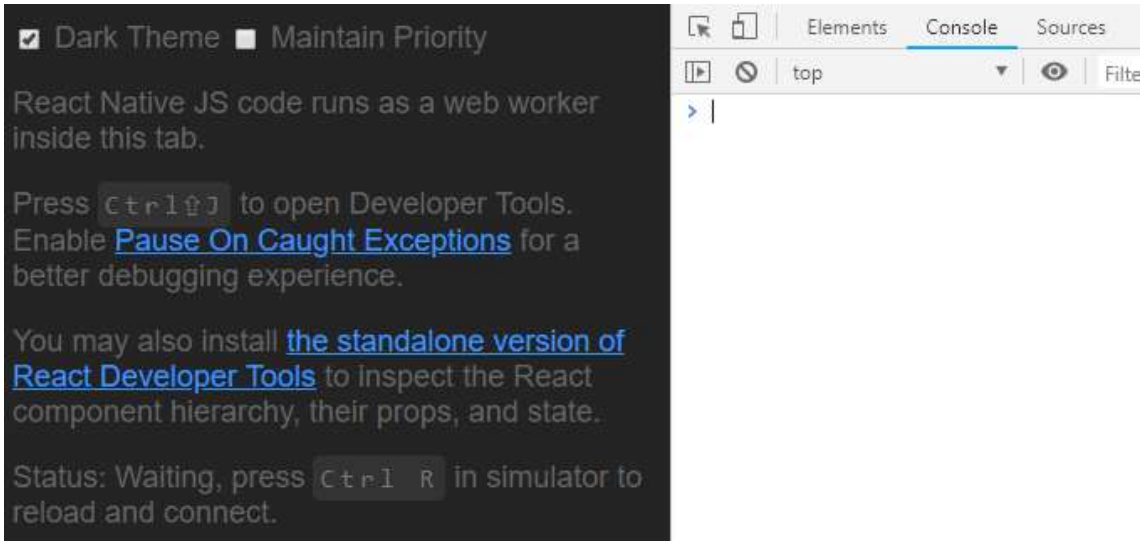

FIGURE 28. Chrome debugging in browser for the application

Sometimes, the best option was to use the debugging in the browser since some of the error messages appeared on the screen of the mobile device as seen in the Figure 29.

Also, during the development, a test environment was created using jest for the project to monitor and have tests for the files in order to have coverage and to see if all the functions are tested or if any tests are missing, which also will help when preventing any further errors.

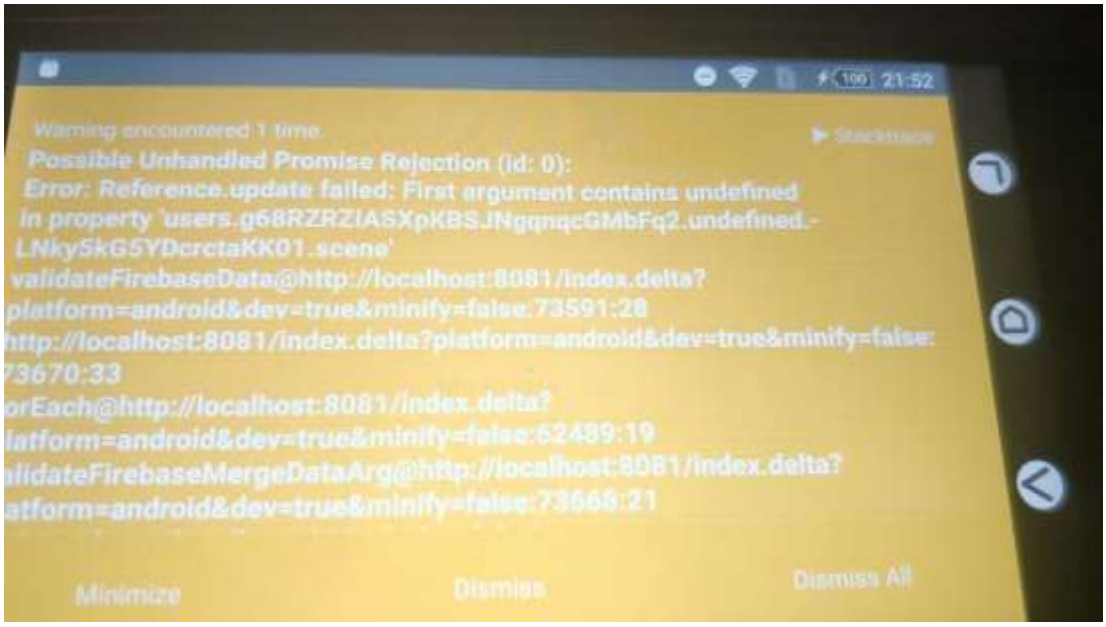

FIGURE 29. Image of an error that happened

Here the problem was that the pathing reference was wrong after some changes had been made to the adding elements function and creating scene function. It could not be resolved correctly because there were a couple of parameters missing. In order to solve this, the missing parameters had to be added for the functions and then the application needed to be launched again in order to see whether the error popped up again. Similar steps were followed for all the errors that occurred and will be also followed for future errors. This helps when adding console logs and using the remote browser debugger to see if the code is working correctly after fixes or if it still needs some adjustments.

7.7.11 Licensing the application

Signing and certifying an application to Android devices is free. This makes the development and testing much easier. When testing the product with the actual

users and their devices, it is easily seen if there are any problems that arise. The developer needs to pay around 25$ to get the application to the Google Play Store, which is quite affordable for any independent developer. The application can be shared without placing the product to the Google Play store by offering a link to download it, for example from Google Drive.

For iOS devices this is slightly different. The application can be built without any cost, but the developer is required to have an Apple developer license to certify and share the application with others. The developer license costs 99$ a year which is 4 times more than that for Android. The advantage is that it comes with plenty of tools for developers.

8 CONCLUSION

During the thesis work and the development process of the mobile application for the Kawa Model therapy, the aim was to create a mobile application for iOS and Android platforms using the React Native framework. During the end of the development process and thesis work there would be a demo version of the product. The project was done in cooperation with two occupational therapy students, Emilia Kuronen and Laura Kämäräinen, who had done all the groundwork and research about Kawa in their own thesis work.

Results from the paper prototype testing indicated that there was interest among the users and the therapists who used the prototype. This information was taken into account when choosing with the project members all the elements that are being in the demo application. A detailed view of the testing results can be seen in Appendix 1. Since the work was shared with people from another field, it was interesting to learn about occupational therapy and the Kawa Model, as well as the applications of technology in the healthcare sector as a whole. The process of creating the product at first proceeded as planned but then there were some challenges that delayed the development process, which was difficult for the both sides. After receiving help from a person outside of the project everything started going well again and progress was made with the application. As the end of the thesis drew nearer, it was clear that the scope of the project is quite large for a student developer who is just beginning to learn about mobile development.

This does not mean that the development of this project would end here, and the product would not see the daylight. Knowing that both team members, Laura and Emilia, are still interested in the project after graduation and willing to present the demo to other therapists, the development of this application will continue. There is still a lot to improve and perfect within the application, and with that comes a great learning potential. In the end, the aimed goal of the demo was not reached but all the work did not go to waste at all since the amount of information about

React Native, Redux, NodeJS, JavaScript that was gained here was extremely helpful in terms of continuation of the project.

Towards the end of the project, there was a new member that joined to work in the project as their company-oriented project which comes as a great opportunity for the continuation of the Kawa application. The next goal from this point forward is to develop it to a point when it is fully functional and ready to be tested extensively in a clinical setting. With professional feedback, it could become clearer whether it could be successfully developed for a wider market and made commercially available for therapists around the world to use.

The prototype got positive feedback from the occupational therapists that had seen or tried the early version of the prototype during the development process of the Kawa application. The type of feedback that was gained from the users was that the application was easy to use, with a simple UI, and it also helped in deciding which elements are going to be used by the clients in the prototype version of the product. There were also mentions that the application feels like the Kawa Model that is currently being used, which was helpful in confirming that the product is in line with the clinical necessities and heading into the right direction. There is also interest among the therapists to try the product when the prototype reaches the stage where it is ready to be used, with all functionalities in place.

9 REFERENCES

About NodeJS. Date of retrieval 26.01.2019.

https://nodejs.org/en/about/

Adiseshiah, E. G. 2018. Paper prototyping: Your step by step guide. Date of retrieval 08.09.2018.

https://www.justinmind.com/blog/paper-prototyping-your-step-by-step-guide/

Banova, B. 2018. The impact of Technology on Healthcare. Date of retrieval 27.01.2019.

https://www.aimseducation.edu/blog/the-impact-of-technology-on-healthcare/

Dam, R. and Siang, T. 2019. Design Thinking: Get Started with Prototyping. Date of retrieval 26.01.2019.

https://www.interaction-design.org/literature/article/design-thinking-get-started-with-prototyping

Dicianno, B. E., Parmanto, B., Fairman, A. D., Crytzer, T. M., Yu, D. X., Pramana, G., ... and Petrazzi, A. A. 2015. Perspectives on the evolution of mobile (mHealth) technologies and application to rehabilitation. Physical therapy, 95(3), 397-405.

Erickson, K. 2015. Evidence considerations for mobile devices in the occupational therapy process. The Open Journal of Occupational Therapy, 3(2), 7.

Getting started React Native. Date of retrieval 11.09.2018.

https://facebook.github.io/react-native/docs/getting-started

Installing Android Studio. Date of retrieval 26.01.2019.

https://developer.android.com/studio/

Installing Xcode. Date of retrieval 23.01.2019.

https://developer.apple.com/xcode/

Jest. Date of retrieval 26.01.2019.

https://jestjs.io/

Kelly W. 2018. Mobile apps for healthcare professionals: Current and future trends. Date of retrieval 09.12.2018.

https://web.archive.org/web/20181209131820/https://mobilebusinessinsights.com/2018/03/mobile-apps-for-healthcare-professionals-current-and-future-trends/

Kuronen, E. and Kämäräinen, L. 2018. Kulttuurit kohtaavat toimintaterapiassa. Date of retrieval 26.01.2019.

http://www.theseus.fi/bitstream/handle/10024/153932/kuronen_emilia_kamarainen_laura.pdf?sequence=1&isAllowed=y

Nielsen, J. 2003. Paper Prototyping: Getting User Data Before You Code. Date of retrieval 26.01.2019.

https://www.nngroup.com/articles/paper-prototyping/

Prototyping. Date of retrieval 26.01.2019.

https://www.usability.gov/how-to-and-tools/methods/prototyping.html

Rauv, S. 2017. The Impact of Technology in Healthcare. Date of retrieval 27.01.2019.

https://www.elcomcms.com/resources/blog/the-impact-of-technology-in-healthcare-trends-benefits-examples

Ravenek, M., and Alvarez, L. 2016. Use of mobile ‘apps’ in occupational therapy: Therapist, client and app considerations to guide decision-making. World Federation of Occupational Therapists Bulletin, 1-7.

T. Y. Siang and Interaction Design Foundation. Copyright terms and license: CC BY-NC-SA 3.0. Date of retrieval 26.01.2019.

https://www.interaction-design.org/literature/article/design-thinking-get-started-with-prototyping

Teoh and Iwama, 2010. About the Kawa Model. Date of retrieval 12.09.2018.

http://www.kawamodel.com/v1/index.php/about/

Verdonck, M. C., and Ryan, S. 2008. Mainstream technology as an occupational therapy tool: technophobe or technogeek?. British Journal of Occupational Therapy, *71*(6), 253-256.

Prototyping results (Phase 2) APPENDIX 1

Vaihe 2: Uoma, nykyhetki | Lähes kaikki valitsi | Yli puolet valitsi

N=16	Kivi	Kala	Rapu	Ajopuu/ tukki	Simpukka	Kasvit
1	Haaste	Läheinen ihminen	Uhka	Haaste mihin voi vaikuttaa	Henk.koht. Vahvuuksia	Vahvuus
2	Ongelma	Perhe	Epävarmuus	Arkiset vastoinkäymiset	Hyviä ominaisuuksia	Tukipilarit
3	Pysyvämpi haaste	Ystävät	Oudot ihmiset	Tuttuus, turvallisuus	Jotain positiivista	Elämän perusasiat
4	Haaste, minkä voittaminen palkitsee	Perhe	Hyviä ominaisuuksia/ ihmisiä	Ympäristö/ ennalta-arvaamaton	Sisäinen voima (helmi sisällä)	Vapaa-aika
5	Minä itse/ tuki ja turva perheelle	Tuttuja ihmisiä	Voi nipistää	Ongelma/ haaste	Läheiset	Haaste
6	Pysyvä asia	Perhe	Liikkuva ongelma	Muutos	Hyviä ominaisuuksia/ ihmisiä	Hyötykasvit
7	Pysyvämpi haaste/ ongelma	Läheiset	Ihminen	Vaikeudet	Myönteinen asia	Koti
8	Pysyvämpi asia elämässä	Hyviä ominaisuuksia/ ihmisiä	Välittäminen (tietyn ihmisen)	Estää näkemästä valoa	Arvokas asia	Turva
9	Haaste	Tärkeät ihmiset	Positiivinen asia	Haaste (helpompi kuin kivi)	Turvapaikka (auennut simpukka)	Este/ haaste
10	Oma (haastava) Ominaisuus	Kanssauijat	Oma hyvä ominaisuus	Ystävät	Hyvä/ tärkeä asia	Sekaannus/ hämmentävä asia
11	Pysyvämpi asia	Ihmiset	Negatiivisuus	Oma positiivinen piirre	Kaunis/ esteettinen	Oma voimavara, mukavuusalue
12	Työt	Ihmissuhteet		Toisten tuki, suojaava tekijä	Koti	Suoja
13	Haasteet (myös posit.)	Perhe		Haaste	Turvallisuus	Kehitys, kasvu
14	Vaikeampi ongelma	Läheiset, tyttöystävä		Ulkoinen haaste		Koti, perhe
15	Vahvuus	Läheiset, yhteenkuuluvuus		Ei niin pysyvä haaste		Ympäristö
16		Itse		Ei niin pysyvä ongelma		Uuden alku