Timetable Mobile Application

Oskar Ciebien

B.Sc.(Hons) in Computing in Software Development

April 10, 2022

Final Year Project

Advised by: Mr. Martin Hynes
Department of Computer Science and Applied Physics
Atlantic Technological University Galway-Mayo (ATU)
Formerly known as Galway-Mayo Institute of Technology (GMIT)

Contents

List of Figures

About this project

Abstract This is a dissertation of a Timetable Application which has been targeted at Android mobile devices. It has been developed with React Native as the framework and Firebase has been used for the Back-End of this project, with the realtime database to show all of the data instantly.

This application offers the ability to create timetables to the users. The timetables can be modified as they user wants it, the classes or modules can be created, deleted or updated.

The front-end of the application is made to look simple and easy to use, for new users. There are many screens to choose from, with each of them offering different functionalities. It is simple to switch between screens, as there is a tab menu at the bottom of the screen, with main screens.

In order to use the application, users need to create an account with their email and password. The application checks if the user inputs an existing account's email, short password, an actual email and many more.
It also offers account management options, ranging from changing the password to deleting the account. There is a little tutorial on the home page of the application for the users that might find the application difficult to use at the beginning.

Authors Oskar Ciebien is a final year student at Atlantic Technological University Galway-Mayo (ATU) formely known as Galway-Mayo Institute of Technology (GMIT). Currently pursuing a Bachelor of Science (Honours) in Computing in Software Development. He has interest in Software Development and would like to pursue a career in software development after graduation.

Chapter 1

Introduction

This is the introduction chapter of this dissertation. It outlines the layout, the objectives, the information about the source code, the project timeline and the project difficulties as well as the learning outcomes of the Applied Project and Minor Dissertation module.

1.1 Layout of the Dissertation

Each chapter of this dissertation will mention different aspects of the preparations, research, technologies and the project itself.

The following are the chapters in this dissertation with a brief description of what each of them will cover in greater detail:

- **Chapter 1 - Introduction -** Chapter 1 which is the current chapter, will describe the project objectives, information on the project's source code and documentation, the projects timeline and difficulties and finally the learning outcomes of the Applied Project and Minor Dissertation module.
- **Chapter 2 - Methodology -** This chapter will discuss the approach on the project, the tools used while developing this project, the source control used, information on the project meetings conducted with the supervisor and the research which was done throughout the process of creating this project.
- **Chapter 3 - Technology Review -** Chapter 3 will provide a review on all of the technologies and the languages that have been used. As well as provide reasoning to why and where they have been used in the project.

- **Chapter 4 - System Design -** Will describe the project as a whole, the design and applications developed. Code snippets will be used and screenshots of the application to make it easier to understand and see how the application was developed.
- **Chapter 5 - System Evaluation -** Will discuss the types of testing that have been carried out while developing the application.
- **Chapter 6 - Conclusion -** The final chapter will conclude the project and discuss the evaluation, the objectives, the improvements that could be done in the future, the negatives of the project and the future plans for the project, after submission.

1.2 Project Source Code and Documentation

The project source code is available at a GitHub repository, under the following link `https://github.com/Oskar-Ciebien/Timetable_Application`.

A brief description of the project, the project requirements and the instructions are available at the repository in the repository README file.

All the documentation and screencasts of the project are also available at the repository in a folder called Documents.

1.3 Goal of the Project

The goal of this project was to create a fully working Timetable application developed in React Native, Expo and Firebase as the main technologies.

The student's goal was to learn about and learn to use the technologies mentioned above. The student has never used any of them, therefore this project will provide the student with more knowledge on the new framework, platforms and services which were all used in conjunction to create this project.

The project allows the user to register and log in using email and password. Thanks to this authentication individual information from that account will be saved to a realtime database and could be accessed on any device after logging into the same account.

Additional idea for the project was to publish the working solution to the Google Play Store since the application was mainly targeted at Android mobile devices.

The Timetable application, as the name suggests needs to have a timetable, which could be viewed and added to at any time by the user as well as remove or modify the classes added to the timetable.

1.4 Project Plans

There have been two project plans created. Each of the plans were planned out for each of the semesters. Both of them were designed in weekly intervals, with at least one objective to be done each week.

The student has consulted with the supervisor on the mentioned plans and has given their best to follow the weekly objectives as to the objectives set out in the plans, but there have been minor differences in objectives done compared to the plans. The github repository commits fully reflects the work done for the project.

Week	Week Starting	Plans
1	20th of September 2021	Think of Ideas
2	27th of September 2021	Think of Ideas
3	4th of October 2021	Think of Ideas
4	11th of October 2021	Think of Ideas
5	18th of October 2021	Decide on an Idea
6	25th of October 2021	Research
7	1st of November 2021	GitHub, Presentation, Dissertation
8	8th of November 2021	Add Supervisor to Repository, Create Project, Write-Up Project Plan
9	15th of November 2021	Work on the Demo and Dissertation
10	22nd of November 2021	Write more on Dissertation and Research
11	29th of November 2021	Work on the Presentation
12	6th of December 2021	Finish Presentation and Simple Demo
13	13th of December 2021	Present Presentation and Demo

Figure 1.1: Semester One Plan.

The first plan was more focused on the idea of the project, the research and planning of the project and the technologies that would be used for the de-

velopment of the project.

Week	Week Starting	Plans
1	17th of January 2022	Restart Project
2	24th of January 2022	Login & Register
3	31st of January 2022	Login & Register
4	7th of February 2022	Other Security Type (Google or Two-Factor)
5	14th of February 2022	Other Security Type (Google or Two-Factor)
6	21st of February 2022	Timetable Functionality
7	28th of February 2022	Timetable Functionality
8	7th of March 2022	Front-End of the App
9	14th of March 2022	Final Touches & Dissertation
10	21st of March 2022	Dissertation
11	28th of March 2022	Dissertation and/or Presentation
12	4th of April 2022	Presentation

Figure 1.2: Semester Two Plan.

The second plan is mainly targeted at development and the dissertation of the project. Where the student would have objectives set out only to focus on the development of the application and the write up of the dissertation.

1.5 Project Difficulties

As the application was being developed, there have been some difficulties on the way. The difficulties that have been faced are as follows:

- Get familiar with the technologies that have been chosen for this application - React Native, Firebase, Expo were the main technologies used. These technologies were completely new to the student. The student sometimes found it difficult to understand the logic of those, but thanks to this course he was able to adapt to new technologies quicker.
- The first project idea. The student decided to restart the project completely after the Christmas review. It was all done to make the project seem and work better. The reason why the project was restarted, was because of a new idea for the login and register functionality.
- Time and priorities were definitely large factors that had to be taken into consideration while developing this project. The student sometimes was forced to decide what to focus on first, or how much time he

should spend developing a feature and if he was able to finish it in the given time. This type of approach was the solution that worked very well and let the student to be more efficient.

- Some of the features like Google Authentication, Delete and Update were very difficult parts to develop in this application. Especially as the student was not familiar with the technologies.

1.6 Learning Outcomes

Applied Project and Minor Dissertation module offers many learning outcomes. Which are very important when it comes to future job search or further studies in the field of computer science. The learning outcomes of this module are as follows:

- Demonstration of the application with the use of appropriate research methodologies and techniques which are all related to software development.
- Demonstration of the awareness at the current state of the art of computing.
- Application of critical thinking during challenging computer and software based problems.
- Design and implementation of a computing solution project which requires research.
- Integration of many different technologies in order to develop and deliver a working solution project.
- Being critical in regards to the work and research.
- Reflection on the strengths, weaknesses and any future potential of the project.

Thanks to these learning outcomes and the project being worth 100% of the overall mark for this fifteen credit module. The student gets a taste of project management, development, research and planning. Which are all very important for career growth in the future.

1.7 Chapter Conclusion

This chapter has described what each chapter is about, it provided the project's source code and documentation link. The goal, plans and difficulties have also been discussed. Finally the learning outcomes of the module have been pointed out.

Chapter 2

Methodology

2.1 Overview

This methodology chapter will describe the various methodologies that have been used while developing this project. It also discussed the approach, meetings, the source control and the tools that have been used in the development. Additionally it will go over some research methods that were taken into account before making the final decisions.

2.2 Research Methodology

There are different approaches to research. These approaches enhance the process and prove the way the research has been conducted [20].

Figure 2.1: Image of Qualitative Research. Adapted from [1]

The Research Methodology used for this project was the **Qualitative Research Methodology**.

Qualitative Research relies on information that is text based. This research is done from first-hand. Information can be obtained through observations, documents and artifacts [21].

This Research Methodology was a good pick for this project, as it allowed the author to research information through documentations, articles and stack overflow in order to be able to fix issues or to know more on the field of technology, such as a programming language or a framework that was used in this project.

2.3 Software Methodology

Software Methodologies allow to divide up the development process into smaller steps which in the end can help with design, development and management of the project [22].

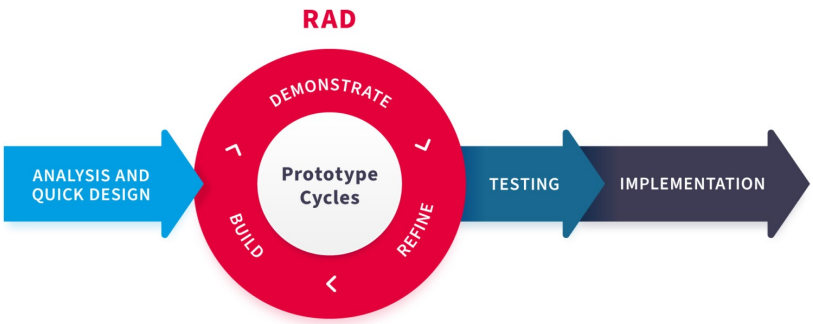

Figure 2.2: Image of Rapid Application Development. Adapted from [2]

This project has been developed using the **Rapid Application Development (RAD)** or sometimes also referred to as **Rapid Application Building (RAB)**.

This Software Methodology suits very well with this type of project as it has been developed by only one person, it allows for developing many features and then deciding which of the few options to keep [23]. This way the development of the application or project is simpler to manage as decisions and changes can be made at nearly any time of the development.

Rapid Application Development is made up of many phases. These include the following [23]:

- **Requirements Planning Phase** - Which includes the scope of the project, requirements and design.
- **User Design Phase** - This phase goes through how users would like to see the project. What kind of inputs, outputs, design they would like to have.
- **Construction Phase** - Focuses on the building of the project. But since this is RAD, any changes or decisions can be made at any time. This phase also includes testing of the project or the application to ensure it is safe to move to the next phase.
- **Cutover Phase** - This is the final phase of RAD. It includes final testing and user training in some situations if it is required.

The RAD Software Methodology was one of the best approaches to pick for this project. As, once again it was an individual project and there was no other communication and cooperation involved. It was much easier to make quick changes or improvements

2.4 Meetings

The project meetings were scheduled weekly. They were attended by both the supervisor and the student. They were all carried out on-line as it was more comfortable and convenient for both the supervisor and the student. The meetings were about fifteen minutes long. The supervisor and the student met over Microsoft Teams. This was the preferred communication software as it was used over the last years in college throughout online learning due to the pandemic and it is included in the university's office package.

2.4.1 Meetings at the Beginning

The meetings at the early stages of the project were based on the planning and possible ideas for the project. The supervisor was giving the student guidance and suggestions on the project ideas. The project plan and the project design brief have been done during that time. These were also consulted with and given to the supervisor. These meetings took a couple of weeks in the first semester.

2.4.2 Regular Meetings

Regular meetings consisted of:

- Current progress on the project.
- The problems encountered during the week.
- Feedback from the supervisor.
- Plans for the following week.
- Changes in the project brief or weekly project plan if any.
- Any questions from the student on the module.

2.5 Source Control

The source control used during the development of this project was **Git** and the hosting provider of the online repository was **GitHub**.

2.5.1 Why were they picked?

Figure 2.3: Git logo. Adapted from [3]

Git was the preferred source control as it has been used throughout the course and it is considered as the most popular and preferred source control in the industry.

Figure 2.4: GitHub logo. Adapted from [4]

Most importantly GitHub was used in this project as it was a requirement for this project set out by the project coordinator. There were also other reasons why GitHub was used, and these are:

- It was used throughout the course.
- The student is familiar with using it.
- It is considered as the most popular hosting provider for Git repository and as an industry standard.
- The student uses GitHub as his portfolio to host all of his projects to share to potential recruiters.

2.6 Tools

In order for this project to be developed in the most convenient way possible. The most popular and considered as best, were the tools have been used. The tools are the following:

2.6.1 Visual Studio Code

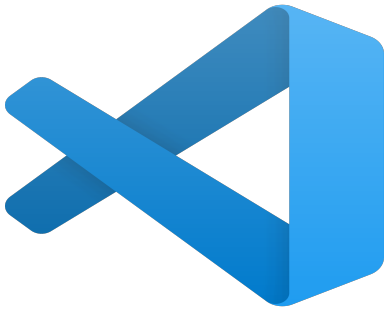

Figure 2.5: Visual Studio Code logo. Adapted from [5]

Visual Studio Code has been used as the Integrated Development Environment (IDE) during the development of this project.

It is very popular, easy to use and a widely modifiable piece of software, thanks to many developers working on extensions in order to improve it.

The student is very familiar with this IDE, because:

- This piece of software has been used throughout the course.
- Easy to use.
- Considered as the most popular IDE.
- It is a perfect IDE to use with the JavaScript language.

2.6.2 Android Studio - Emulator

Figure 2.6: Android Studio logo. Adapted from [6]

Android Studio is the most popular and a great IDE to use when it comes to Android Development. While the only way that it was used in this project was the built-in emulators.

Since this project is aimed at Android devices. An android emulator was a must. Thanks to Android Studio the testing process has been reduced tremendously, as it is much faster to test an application on the same PC or laptop as it is developed with nearly no lag, than to fiddle with cables connected with a physical mobile android device and wait until the application compiles and installs. An emulator is much faster, as it is using the computer's specifications, while a testing android device could have outdated hardware.

Android Studio made testing more convenient as well, because controlling a device with the same mouse and a keyboard is more comfortable and efficient.

2.7 Chapter Conclusion

The Methodology chapter has outlined the research and software methodology that have been used in the process of making this project. It also discusses the meetings that have been conducted with the supervisor throughout the year. Git and GitHub have been explained as the source control and hosting provider of this project, and finally the two main tools, visual studio code and android studio were mentioned. Which made the development quicker and easier.

Chapter 3

Technology Review

3.1 Overview

This chapter will focus on all of the different technologies that were used to develop this timetable application. Their advantages and disadvantages as well as how they were used in the project will be discussed.

3.2 Git

Figure 3.1: Git logo. Adapted from [3]

Git is an open source distributed version control system. It is designed in such a way to be able to manage small and large projects with ease, while being fast and efficient [24].

Git is released under the GNU General Public License version 2.0, which means it is an open source project. Thanks to this license, all developers are

free to share, change and use it for free [24].

This distributed version control system, is very easy to learn, fast and efficient, hence why it is so popular and used by many of the largest companies in the world such as Google, Facebook, LinkedIn and Microsoft [24].

What are the Advantages?

Git has many advantages and one of them is how distributed it is. Instead of using one repository, each of the developers have their own local repository and a history of commits [25]. This makes it fast, scalable and easier to work with.

Git is one of the most popular version controls in the world. Thanks to that, nearly all developers have had experience with Git and will not have to learn it from the beginning [25].

What are the Disadvantages?

Git is an excellent tool, but as any tool it also has some disadvantages.

One disadvantage of this piece of software is that it is slower on Windows machines [26]. Which could be a big downside for power users of Windows machines when using Git for development. But in most cases this will not have a major difference.

Git also does not support binary files [26], which could be important for some developers.

One of the annoying disadvantages of Git is the fact that it does not see empty folders inside of the repository and they cannot be committed, unless a file will be added or moved to that empty folder. Then it will recognise the folder [26].

How was it used in this project?

In this project Git has been used as the version control system. It was used to connect to GitHub and push commits from the local repository to the hosted repository up in the cloud in the GitHub Servers.

3.3 GitHub

Figure 3.2: GitHub logo. Adapted from [4]

GitHub is a hosting service for version control, with the use of the above mentioned Git [27].

It offers source code, project management and many more useful collaboration features. [27].

What are the Advantages?

The GitHub Pages that it offers allows developers to showcase their projects and portfolios on a static page completely for free. They can be customised as the developers would like it with the paid themes [28]. But the source code of the static web page has to be public and the name of the repository has to be specific, such as NameOfAccount.github.io where NameOfAccount is the username of the user's account.

GitHub can be accessed anywhere. Whether it is from a phone, windows PC or a Mac. Windows and Mac code sharing is simple and they can both do the same actions. While users can only access it through the website on the phone only to view the repositories and projects [28].

What are the Disadvantages?

With Git having some disadvantages, and GitHub working very well with Git, this also means that it can have some bad sides as well, which it unfortunately does.

As any other service provider website, GitHub also provides services, but not all of the services are available for free. For users that do require a lot of repositories, teams etc. this could mean that they will be forced to buy a plan [29].

Since GitHub is a website. Any websites sometimes might encounter an error or update where the website will have to be put on a downtime for some time. This could create some problems and inconvenience when trying to push code up onto a repository online. Especially when working in a team. This has happened once while working on this project.

How was it used in this project?

GitHub acted as the hosting provider of the project's repository. It worked with Git to host the repository with all the needed files, including Documentations, the Project and the Read Me file.

3.4 Node.js

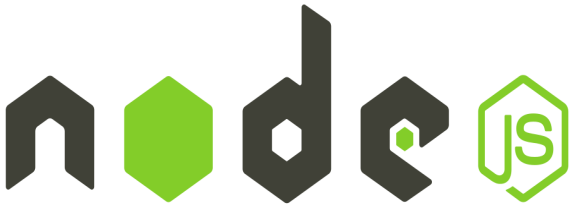

Figure 3.3: Node.js logo. Adapted from [7]

Node.js, is an open-source piece of software that works on many platforms. It has a backend JavaScript runtime environment, which runs on a so called V8 engine [30].

Node allows the creation of web servers and networking tools with the use of JavaScript. It has a collection of modules that are responsible for various core functionalities [30].

The only language that Node supports natively is JavaScript as the name suggests .js. But there are many languages that can compile to JavaScript, some of these include Dart and TypeScript [30].

What are the Advantages?

Node.js is open source. This means that it is completely free. All modules, libraries and code samples are free to use [31].

Since it is open source. Developers try to help each other out. It is community driven. There are many developers who contribute to Node.js, and they are always willing to help out or to share their knowledge with somebody else [31].

What are the Disadvantages?

Too many modules. There is a lot of unnecessary modules, which could over complicate things [31] [32]. There can be situations where there are two modules, which are called similarly, this could cause confusion on which module to choose.

Most JavaScript developers do not have a lot of experience with Node.js. Even experienced Software Developers tend to find themselves struggling with Node.js sometimes [32].

How was it used in this project?

Node.js was downloaded specifically for this project in order to get the Node Package Manager. Since it is essential for this project to be built and to run as expected.

3.5 Node Package Manager (NPM)

Figure 3.4: NPM logo. Adapted from [8]

Node Package Manager or NPM is a package manager for Node.js. It is an open source project. It's goal is to help JavaScript developers share packaged modules of code [33].

The registry is a collection of packages which are open source and readily available to everyone to use for front-end, mobile applications and more. Developers can easily install any of the packages as well as publish packages [33].

What are the Advantages?

As mentioned above, the Node Package Manager is an open source project. Which is always a great thing when it comes to software development. It will most likely get updates for a very long time as individual developers will always try to improve on something and most importantly it will be secure. Just as any other open-source project, for example Linux, it is open source but it will stay secure as any bug will end up being fixed by an experienced software developer.

What are the Disadvantages?

NPM might be the preferred and most popular package manager that is used currently, but it definitely has its disadvantages as well. NPM can sometimes take its time to install or reinstall a package, when compared to other package managers. This shows that there is definitely a problem with the performance of this package manager [34]. But it does not mean that it is bad. It does the job, but it takes it's time.

How was it used in this project?

The Node Package Manager was a very important technology in this project. With the help of npm, most of the features of the application that needed to be installed would not have been possible without npm. Thanks to npm the project ran as expected which was and is needed for development and testing.

3.6 React Native

Figure 3.5: React Native logo. Adapted from [9]

React Native is a JavaScript based framework that Meta, formerly known as Facebook has released in 2015 and it is still maintaining it. React Native combines native development with React, hence the name React Native. React Native code is written in JavaScript but it is then rendered with native code for a given device [35].

What are the Advantages?

Most important feature of this framework is the ability to develop cross-platform application with one code [35]. Which then transforms it into native code for other operating systems like Android and iOS. This feature is a great advantage, which decreases the cost and time it takes to develop applications. It is well suited for start-ups, as it does not require different developers to develop working versions for different systems like iOS or Android.

React Native also offers a great feature called ”Fast Refresh”, this feature allows for as the name suggests a fast refresh while developing and testing an application. The application refreshes when a file has been saved, without

the need of restarting the application [35].

What are the Disadvantages?

One major disadvantage that have been encountered in this project is the lack of information in the error messages. There have been times where the error messages seemed the same while debugging, therefore it was very difficult to debug the application. There was no specific ”hint” as to what causes the problem in the code.

Another disadvantage that could cause a bigger issue for some users is the loading time of the application. While developing this application. React Native takes a longer amount of time to start than if it was developed natively through Android Studio for example. But this is a logical issue as the code is not written in native code.

How was it used in this project?

In this project React Native was used to build the Front-End of the application, as well as to connect with other components in order to have the project working as planned.

3.7 Firebase

Figure 3.6: Firebase logo. Adapted from [10]

Firebase was founded in 2011. From 2014 it has been acquired by Google. It is rapidly gaining respect and fame amongst developers. Large multinational companies are trusting Firebase services.

This back-end service provider offers many services such as, databases, authentication, storage and hosting. With that it also offers analytics to see how the application is evolving. Firebase also offers many paid plans as well as a limited free plan.[36].

What are the Advantages?

One of the advantages that Firebase offers is that it is quite simple to integrate on iOS, Android and Web [36]. This does not mean that it is very easy to integrate. It all depends on the technologies that are being used with it also.

Firebase offers many services, all in one package. They offer databases, analytics, storage and hosting. This is a great service to consider, as it allows for quick scaling and with many services in one, it offers great integration. Definitely will change a lot when it comes to start-ups or even if developers would like to host and or develop their own small projects.

The documentation is well written and very helpful. They offer an example code snippet to each feature of their service in the documentation [36]. This does not mean that it will always help, as the code may differ a lot, depending on how the source code of a project is written, but it will always be a good advice.

What are the Disadvantages?

iOS devices do not seem to be favoured by Firebase. Which makes sense as Android is sponsored and focused by Google more than other systems. This is a disadvantage as Firebase claims it is a cross-platform service, but it is not completely true [37].

Firebase has a major limitation, which could be a large factor to consider when deciding if this service is right for a project. The mentioned limitation is availability. Firebase will not work in countries where Google is banned or not allowed in [37].

There are two available plans to pick from when choosing this service. The ”Spark Plan” is completely free, but has it’s limitations, it is great for very small applications, it also depends which services the applications will use, and it is great for development and testing. There is also the ”Blaze Plan” which is a Pay As You Go plan. This is not necessarily a good idea, as we can never be sure how viral the application will be in the next month or so.

This could increase the costs drastically without making enough profits from the application itself [37].

How was it used in this project?

Firebase managed all the back-end of the project. The database and the authentication features of this service were used, to store the user's information as well as the timetable information.

3.8 Android Studio

Figure 3.7: Android Studio logo. Adapted from [6]

Android Studio is a great piece of technology which provides tools for building applications for any Android Device [38]. It also offers many emulators which are based on real devices such as Google Pixel phones and different versions of Android such as KitKat, Oreo etc. With that there is also an IDE integrated within this software. Making it completely easy to program applications natively for Android devices in Java or Kotlin programming languages.

What are the Advantages?

Android Studio has a built in editor. It is called the Visual Layout Editor, it allows for quick and easy manipulation of GUI elements for the application, without touching the code [38].

One of the most important advantages of it is the emulator feature. There are multiple options of devices and versions of android to pick from. This

emulators are fast, quick and very simple to install. It has complete functionality of a standard phone. There is access to the camera, volumes, tilting of the screen and many others [38].

The Integrated Development Environment (IDE) inside of Android Studio is based on the IntelliJ IDE. Which is one of the most popular IDE's in the world. Additionally Android Studio carries some other functionality which enhances the productivity [39].

What are the Disadvantages?

Emulators are big in size. They take a lot of memory and storage. This could be an issue on devices used for development with smaller storage capacity like some laptops. Certain laptops or PCs can get warm when running an emulator in the background, as they use a lot of memory to run.

How was it used in this project?

Android Studio emulator allowed for the project to be tested more comfortably and more efficiently. Which helped to develop, test, visualise and connect everything together.

3.9 Expo

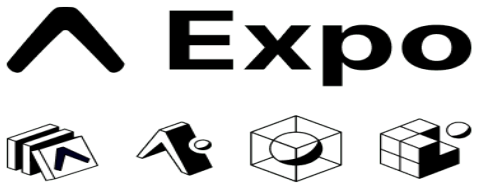

Figure 3.8: Expo logo. Adapted from [11]

"Expo is a framework and a platform for universal React applications". It is a great collection of tools and services which can be used with React Native in order to develop, build and deploy cross-platform applications using only

one language which is JavaScript or TypeScript [40].

What are the Advantages?

The Metro Bundler or the Expo Developer Tools, makes the whole development process a whole lot easier and faster. With just one click of a button or option picked in the development console or in the web browser, the developer is able to run the application on their iOS or Android emulator as well as a website.

Thanks to Expo-Cli the creation of a project is much quicker and it provides templates to use from, to give developers a little bit of a boost at the start if they decide to use one of the templates [41].

What are the Disadvantages?

Expo App sometimes disconnects with the Expo Developer Tools app or the Metro Bundler. It can be frustrating and make the developer loose concentration while developing.

How was it used in this project?

Expo was used to build the APK of the application in the end, and it also allowed for compiling the application on an android emulator and an android phone over a wireless connection. Which was very convenient and fast to use. The Expo-CLI was also used to initialise the project in the beginning.

3.10 LaTeX

Figure 3.9: LaTeX logo. Adapted from [12]

LaTex is a typesetting system. It is most commonly used for writing scientific and technical documentations. It is completely fre to use [42]. The Overleaf website provides LaTeX environment to write LaTeX documents with ease.

What are the Advantages?

One of the huge advantages of LaTeX is efficiency. With the use of table of contents, or bibliography or references and table of figures etc. the authors do not have to insert anything or refresh it manually. LaTeX adds everything to it automatically after it gets compiled.

What are the Disadvantages?

The appearance of LaTeX may not suit everyone. It still allows the writers to have some customisation in their documents through the use of packages, but in the end the document will look plain as it was built for scientific and technical documentation which do not need a lot of customisation and pretty looks.

Writing in LaTeX is completely different than in any other text editors or text processing programs, such as the very popular **Word**. In LaTeX we need to use commands, packages etc. It is not as simple as a drag and drop to insert an image to a document. See the below figure to see how a simple table is created in LaTeX.

```
\documentclass{article}
\usepackage{multirow}
\begin{document}
\begin{center}
\begin{tabular}{ |c|c|c|c| }
\hline
col1 & col2 & col3 \\
\hline
\multirow{3}{4em}{Multiple row} & cell2 & cell3 \\
& cell5 & cell6 \\
& cell8 & cell9 \\
\hline
\end{tabular}
\end{center}
\end{document}
```

col1	col2	col3
Multiple row	cell2	cell3
	cell5	cell6
	cell8	cell9

Figure 3.10: LaTeX table with code. Adapted from [13].

How was it used in this project?

Without LaTeX this dissertation would not be as comfortable to write as it was. This technology was first used by the students of this course at the beginning of fourth year. It is always great to learn new technologies which could make the students switch from the traditional text processing softwares to LaTeX.

3.11 Chapter Conclusion

The Technology Review chapter provided descriptions of the technologies used in this project, with the most important advantages and disadvantages of them and with an explanation to where that technology was used in the project. This project would not have been possible without the use of these technologies.

Chapter 4

System Design

4.1 Overview

This chapter goes over the system design for this project. There are many different design techniques and components used in this project, which are important to how the application functions. This chapter will also go over the two types of applications that have been developed.

4.2 Android Mobile Application

The Android application was the main focus of this project. As the targeted devices of this project are Android devices.

Why Android Devices?

The reasons why Android devices have been picked as the main target of this application are as follows:

- Android devices are the most popular devices used currently worldwide. They take over 70% of global market share, while over 20% of the global market share are iOS devices [43].
- It is easier to develop Android applications than iOS applications.
- With the picked technologies for this project, there are more limitations for iOS devices than the Android devices.

4.2.1 Login and Register

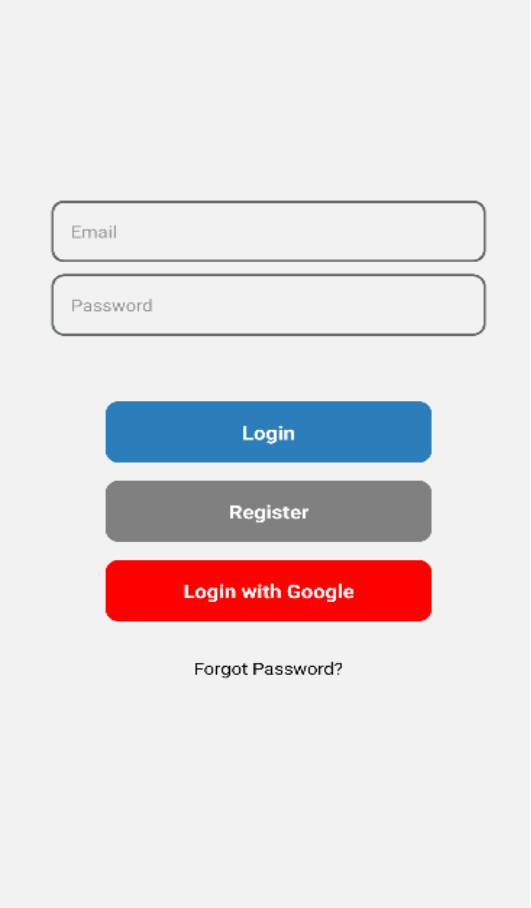

Figure 4.1: Login and Register Screen of the Application.

At first, once the user starts up the timetable application. The Login and Register screen is displayed, as shown on the figure above, where the user is presented with two input boxes for the email and password and three buttons for Login, Register and Login with Google. The Google function was not implemented unfortunately, but the attempted code is left in the application.

4.2.2 Login Functionality

When a user decides to login. They are required to input their email and password. They can then press on the Login button.

After pressing the login button the application checks with the authentication system from firebase, if the email and password that the user has entered match.

If they do, the user is then authenticated and will be transferred to the Home Page.

If the details entered by the user do not match or if the password is too short, then an alert box will be displayed on the screen, to inform the user about the error. The figure below displays the user not found alert.

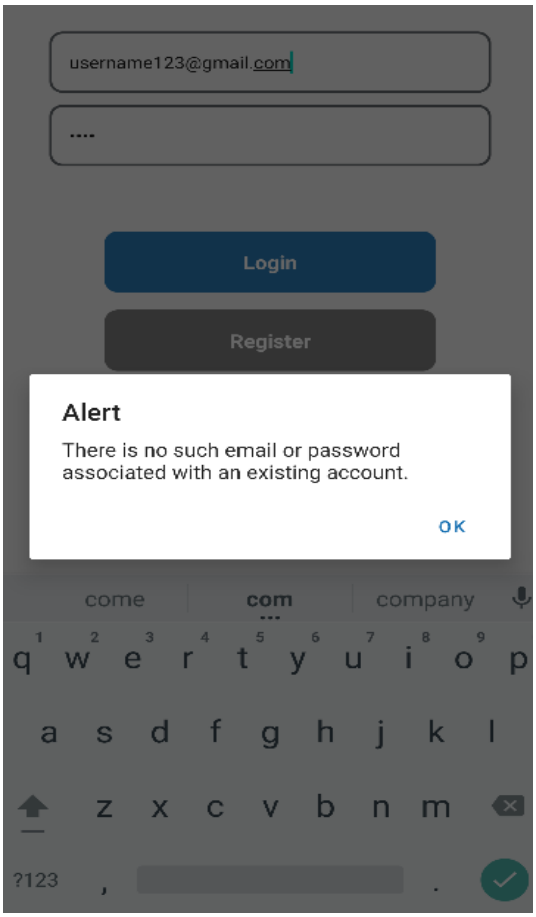

Figure 4.2: No such account alert box in the Application.

4.2.3 Register Functionality

In order to register in the app. The user has to give their email and password in the same input boxes provided.

Next up the user can click on the Register button, which will prompt firebase authentication system to register a new user with the given email and password.

On a successful register, the user details are saved in the firebase authentication system and then the user is moved to the home page of the application.

The system also checks if the email is a real email and if the password is too short. This will prevent the user from registering and prompt an alert box to let the user know of the situation. Please look at the figures below.

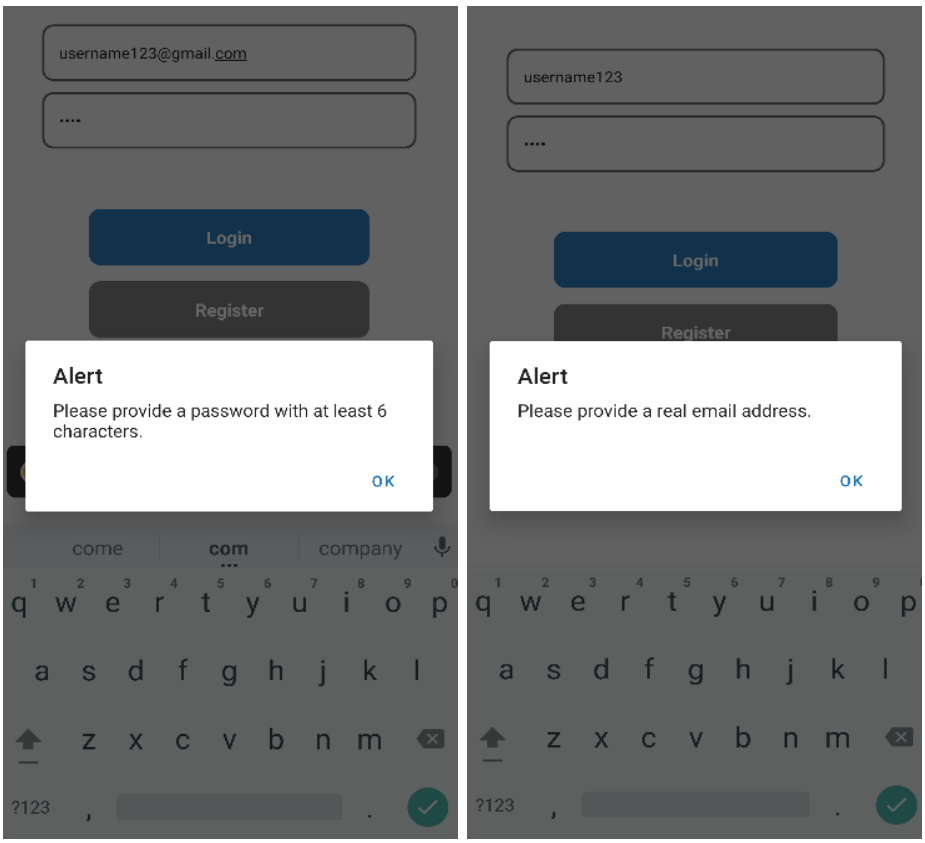

Figure 4.3: Checks for password length and real email in the Application.

4.2.4 Forgotten Password Functionality

When the user finds themselves in a situation where they cannot remember their password for their account. They can use the Forgot Password feature in the application. Which is available from the Settings and the Login and Register screen in the application.

As a good coding practice there is only one Forgot Password screen and the application remembers from which screen it was accessed through, so then the back buttons in the app and the back button of the device work properly and do not move a user from login screen to settings without authenticating.

This feature is usually not placed in the application when the user is logged in. The reasons for having a forgot password screen inside of the application when the user is authenticated are as follows:

- The application uses asynchronous storage and therefore keeps the user logged in after a user leaves the application while logged in. They might forget their password after some time of not logging in.
- Or the user might want to change their password every so often, because of security purposes and they might end up with the password being forgotten as they did not have to log in often, thanks to the application keeping the user logged in.

This asynchronous storage feature has it's advantages and disadvantages. But thanks to that it should increase user satisfaction, as they do not have to waste a few seconds to log in every time they want to access their timetable, which most likely will be looked at many times a day.

4.2.5 Home Page

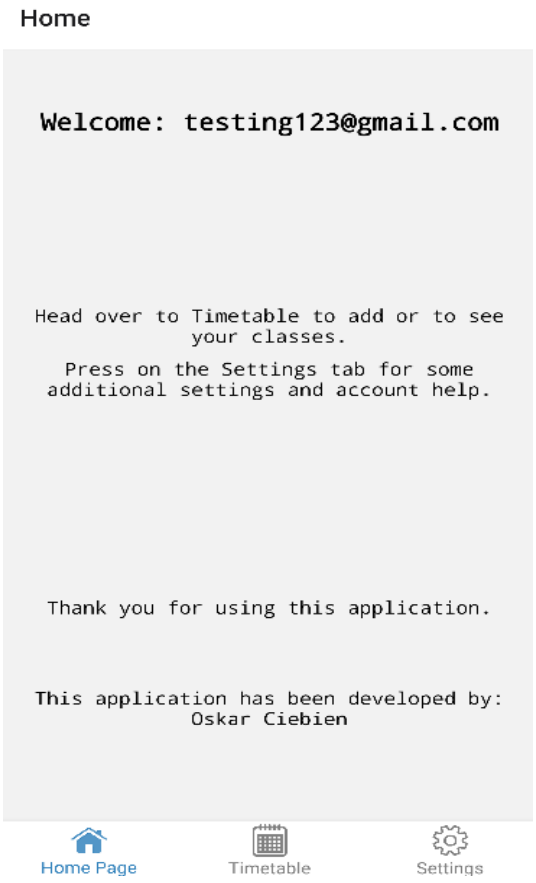

Figure 4.4: Home Page in the Application.

Once the login or register process has been passed through successfully. The user is then transported to the home page screen. On this screen the user is greeted with information about the application and a brief tutorial on how to begin using the application.

4.2.6 Timetable Page

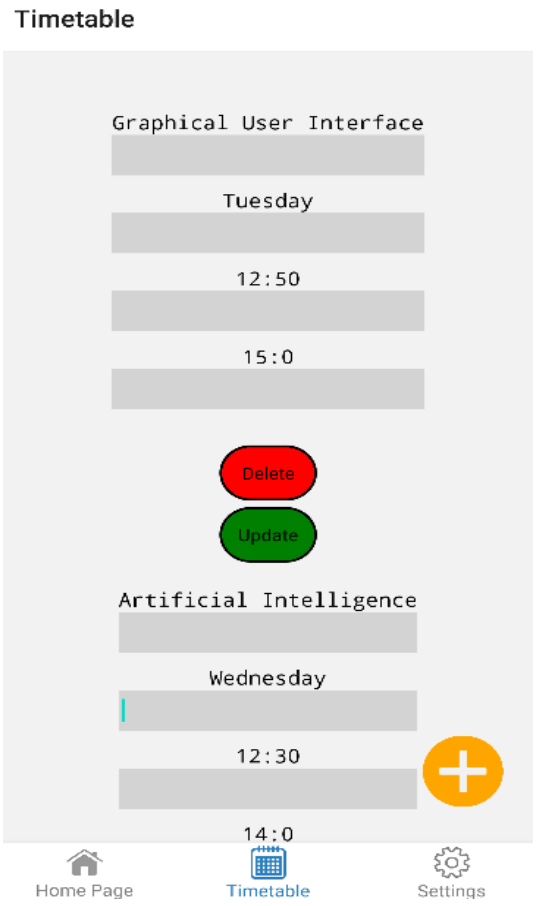

Figure 4.5: Timetable Page in the Application.

When a user clicks or taps on the Timetable icon from the tabs menu. They will be moved to the timetable screen where they will be able to add new timetable using the icon on the bottom right hand corner in a shape of a addition sign. They will also be able to see their classes in a scroll able column.

At the bottom of each class, there are two buttons.

- **Delete** - This red button is responsible for deleting the classes. On each tap the class is then removed from the database instantly.
- **Update** - The update button updates the class, but only the items that have been filled into the text boxes just below each item.

4.2.7 Add Class Page

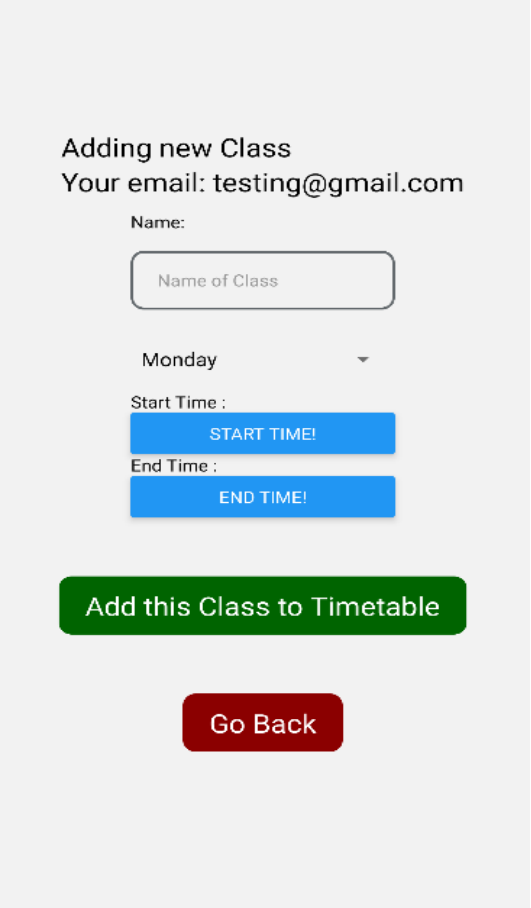

Figure 4.6: Add New Class Page in the Application.

After the tap on the addition icon the user is then moved to a new screen, where they can add their new class. There are four things that make up a class in the timetable, which are:

- **Name** - The name of the class or module, for example Artificial Intelligence.
- **Day** - The day when the class or module occurs, for example Tuesday.
- **Start Time** - The time at which the class or module starts.
- **End Time** - The time at which the class or module ends.

There are two buttons on this screen, one is responsible for adding the class to the timetable and the other is used to go back, which does the same functionality as the back button on android devices.

4.2.8 Settings Page

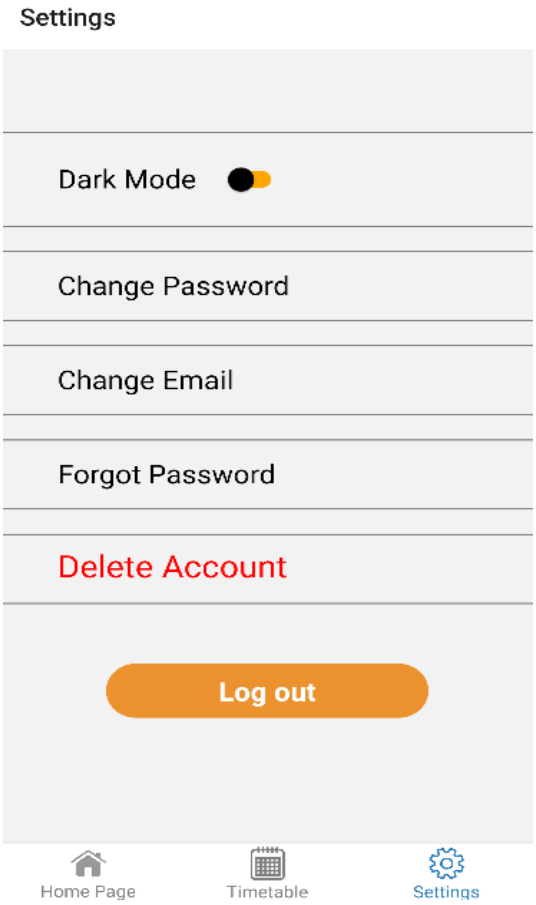

Figure 4.7: Settings Page in the Application.

Settings page can be accessed by pressing on the gear icon with the text Settings on the tab menu at the bottom of the screen. On this page there are many functions which are very important for the account management.

4.2.9 Dark Mode

Dark mode has been set as an additional feature of this project. Unfortunately it has not been implemented and only the switch on the Settings screen works.

4.2.10 Change Password

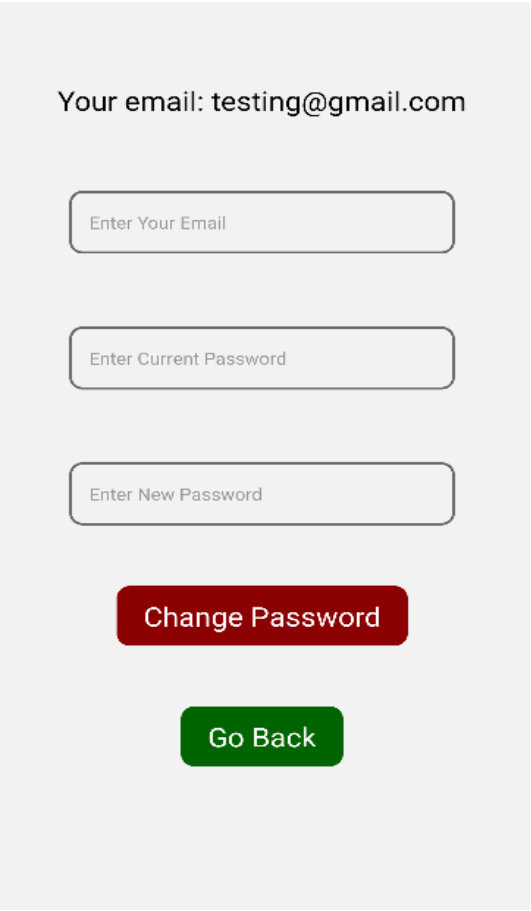

Figure 4.8: Change Password Page in the Application.

The Change Password screen is responsible for changing the password whenever the user sees it as a necessity, for example when they forget their password or decide to change it every so often for security reasons.

On this page the user is asked to enter their email, current password and the new password. With this information, the application is able to verify if the email and current password are valid and match the account details. Therefore the password can be changed in the Firebase Authentication service.

```
// Change Password
const changePassword = () => {
  console.log(oldPass);

  reauthenticate(oldPass);

  updatePassword(auth.currentUser, newPass)
    .then(() => {
      // Update successful.
      console.log("Password has been changed to: ", newPass);
    })
    .catch((error) => {
      // An error ocurred
      const errorCode = error.code;
      const errorMessage = error.message;

      console.log("Error with changing password: ", errorCode, errorMessage);

      // Display different error messages to the user
      if (errorCode == "auth/user-not-found") {
        alert(
          "There is no such email or password associated with an existing account."
        );
      } else if (errorCode == "auth/invalid-email") {
        alert("Please provide a real email address.");
      } else if (errorCode == "auth/invalid-password") {
        alert("Please provide a password with at least 6 characters.");
      } else if (errorCode == "auth/weak-password") {
        alert("Please provide a password with at least 6 characters.");
      } else {
        alert(errorMessage);
      }
    });
};
```

Figure 4.9: The Change Password function from Project Source Code.

On the above figure, the code that is being executed after the ”Change Password” button is pressed is shown. This code also checks if the password is of right length, if the email is a real email and if the email and password match in the reauthenticate method, which is shown in the figure below.

```
// Re-authentication
reauthenticate = (oldPass) => {
  console.log(oldPass);

  const credential = EmailAuthProvider.credential(user.email, oldPass);

  console.log(credential);

  reauthenticateWithCredential(user, credential)
    .then(() => {
      // User re-authenticated.
      console.log("Authentication passed, user: ", credential);
    })
    .catch((error) => {
      // An error ocurred
      const errorCode = error.code;
      const errorMessage = error.message;

      // Alert about authentication error
      alert(
        "There was a problem with Authentication. Please try again",
        error.message
      );

      console.log("Error with re-authentication: ", errorCode, errorMessage);
    });
};
```

Figure 4.10: Reauthenticate function from Project Source Code.

The reauthenticate function checks if the account matches and reauthenticates the user as sometimes the user might not be authenticated in the app after using it for a while. The change password firebase method also requires for the user to be authenticated. That is also why the current password is needed in this case, as the application passes on the password to reauthenticate and then changes the password.

```
<View style={styles.inputContainer}>
  <TextInput
    placeholder="Enter Your Email"
    value={email}
    onChangeText={(text) => setEmail(text)}
    style={styles.input}
  />
</View>
```

```
const [email, setEmail] = useState("");
const [oldPass, setOldPass] = useState("");
const [newPass, setNewPass] = useState("");
```

Figure 4.11: Forgot Password Text Input and Constants from Project Source Code.

The above figures describe the code structure for each of the inputs in this given screen and how the constants change on text change.

4.2.11 Change Email

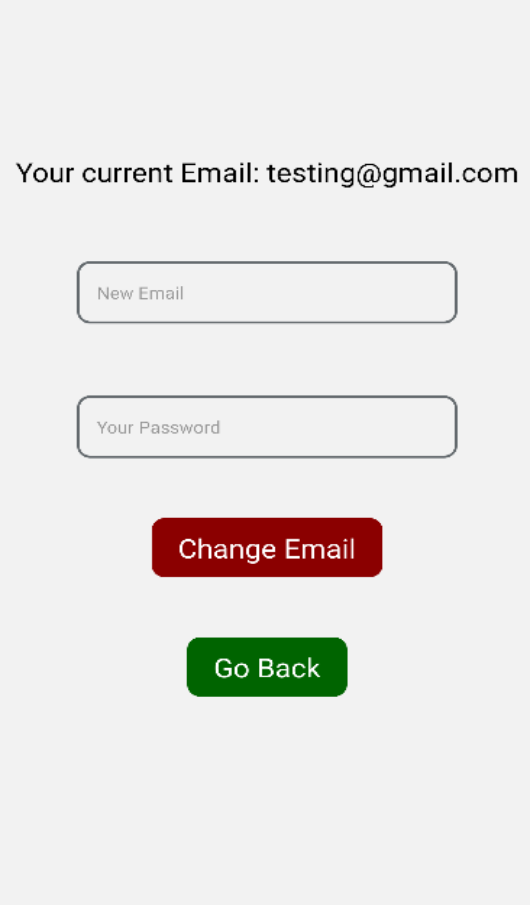

Figure 4.12: Change Email Page in the Application.

The screen to change the user's email, is responsible for allowing the user to change their email. It takes the very same approach as the previous change password screen. It firstly needs to authenticate the user once again, before proceeding to changing the email.

The design of this screen looks very much the same, except there is one less input box. The input boxes on this screen are for inputting the new email and for the current password. The current password is there to confirm the user's choice and to reauthenticate the user with the firebase authentication

service. The email does not have to be inputted as the email can always be accessed, not like a password.

```
// Change Email
const changeEmail = () => {
  console.log(pass);

  reauthenticate(pass);

  updateEmail(auth.currentUser, email)
    .then(() => {
      // Email updated!
      console.log("Email has been changed to: ", email);
    })
    .catch((error) => {
      // An error occurred
      const errorCode = error.code;
      const errorMessage = error.message;

      console.log("Error with changing email: ", errorCode, errorMessage);

      // Display different error messages to the user
      if (errorCode == "auth/user-not-found") {
        alert(
          "There is no such email or password associated with an existing account."
        );
      } else if (errorCode == "auth/invalid-email") {
        alert("Please provide a real email address.");
      } else if (errorCode == "auth/invalid-password") {
        alert("Please provide a password with at least 6 characters.");
      } else if (errorCode == "auth/weak-password") {
        alert("Please provide a password with at least 6 characters.");
      } else {
        alert(errorMessage);
      }
    });
};
```

Figure 4.13: Change Email function from Project Source Code.

The above figure displays the code for changing the email address.

4.2.12 Forgot Password

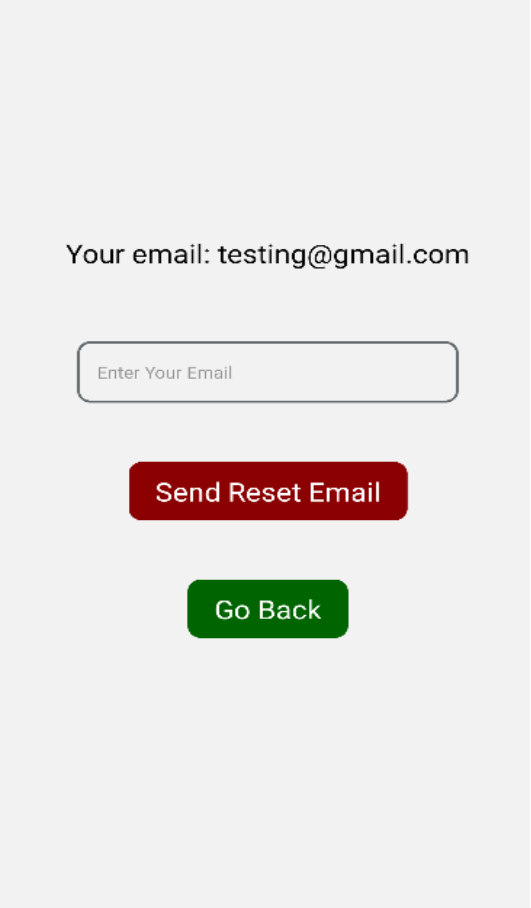

Figure 4.14: Forgot Password Page in the Application.

This screen is available from two screens. It is available from the login and register screen and the settings screen. It has exactly the same functionality. No matter from which screen it is opened from.

It takes in only one text input, which is the email. The user can then confirm their decision by pressing on the button to send an email with a new password. After receiving the email from the firebase authentication service. The user can log in with the new password that was sent in that email.

```
// Forgot Password
const forgotPassword = () => {
  sendPasswordResetEmail(auth, email)
    .then(() => {
      // Password reset email sent!
      console.log("Password reset email has been sent to: ", email);
    })
    .catch((error) => {
      const errorCode = error.code;
      const errorMessage = error.message;
      console.log("Error: ", errorCode, errorMessage);

      // Display different error messages to the user
      if (errorCode == "auth/user-not-found") {
        alert("There is no such email associated with an existing account.");
      } else if (errorCode == "auth/invalid-email") {
        alert("Please provide a real email address.");
      } else {
        alert(errorMessage);
      }
    });
};
```

Figure 4.15: Forgot Password function from Project Source Code.

The above figure displays the code for the forgot password function.

4.2.13 Delete Account

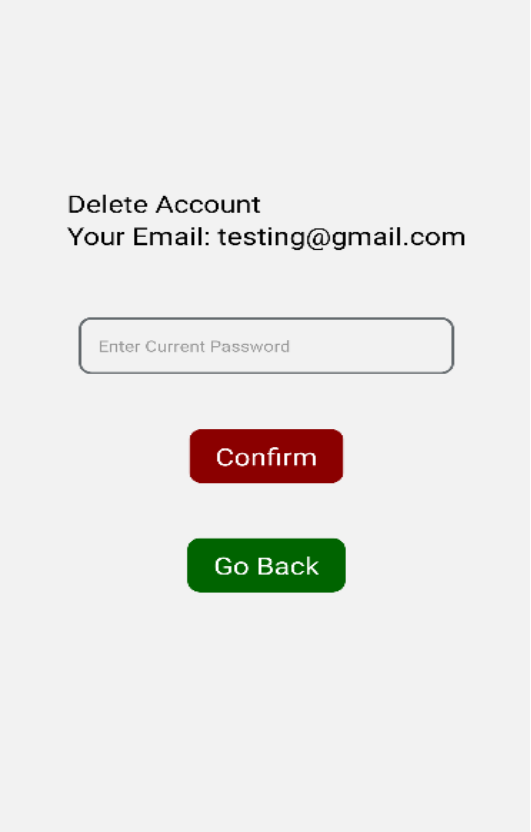

Figure 4.16: Delete Account Page in the Application.

The delete account screen is responsible for deleting the account. It can only be opened from the settings screen.

On this screen the user has to confirm the delete action by inputting their current password and then pressing on the Confirm button. Once that button is pressed, the application has to reauthenticate the user, in the same way as it is done on the other screens, mentioned previously. If the reauthentication has passed successfully, then the application alerts the user if they are definite about deleting their account. They are given a Yes or No alert box. If no is pressed, the alert box is closed. But once yes is pressed all of users information from firebase authentication is deleted along with any information related to this account in the database is also deleted.

```
// Remove user account and details from database
const deleteAccount = () => {
  console.log(pass);

  console.log("Deleting account: " + user);

  console.log(userAuthenticated);

  if (userAuthenticated == true) {
    deleteUser(user)
      .then(async () => {
        // Class Reference
        const userRef = ref(database, "users/" + user.uid);

        // Remove user from database
        remove(userRef);

        console.log("Deleted Account - Successful");

        // Class Reference
        const classRef = ref(database, "classes/" + user.uid + "/");

        // Remove user's classes from database
        remove(classRef);

        console.log("Deleted User's Timetable - Successful");

        // Navigate back to Login Screen
        navigation.replace("Login");
      })
      .catch((error) => {
        // An error ocurred
        const errorCode = error.code;
        const errorMessage = error.message;
        console.log("Error: ", errorCode, errorMessage);

        // Display different error messages to the user
        if (errorCode == "auth/invalid-password") {
          alert("Please provide a password with at least 6 characters.");
        } else if (errorCode == "auth/weak-password") {
          alert("Please provide a password with at least 6 characters.");
        } else {
          alert(errorMessage);
        }
      });
  } else {
    console.log("Account has not been deleted.");
  }
};
```

Figure 4.17: Delete Account functions from Project Source Code.

The above figure displays the code for deleting the account with database contents of the timetable associated with that account.

4.2.14 Log Out

The log out button is situated on the settings screen. Once the log out button is pressed the application logs the user out. The user of the application is then navigated back to the login page. Where they can log in again, with a different account or register with a new one.

4.2.15 Account Management

All of the user accounts are managed through Firebase. Each user gets a unique identification number (UID) through which they can be identified. This UID can also be used for other services, like the databases which will match individual information for each user.

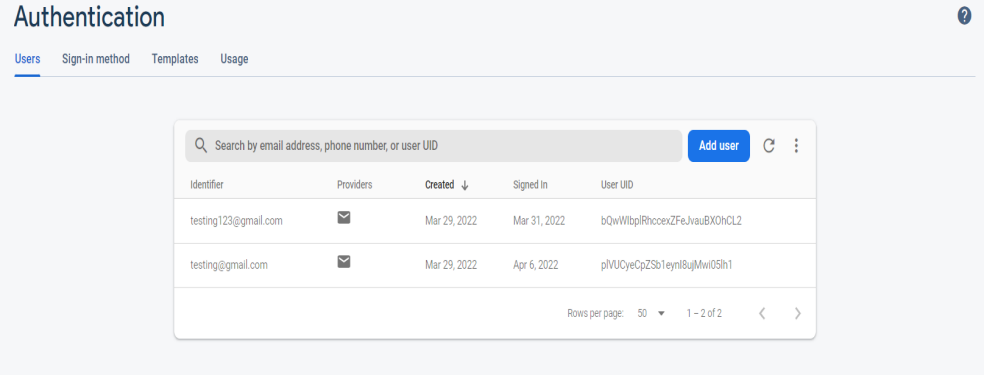

Figure 4.18: Firebase Authentication Project Page.

As shown on the figure above, the administrator of the firebase project can search for a user, delete a user and also add a new user. They are also able to see when the user has created their account and when was their last login. They can also see what kind of account the user has created. In this case the accounts have been created with an email and password.

4.2.16 Database Management

Firebase provides two types of Databases. The firestore database and the realtime database. The database that has been picked for this project is the

realtime database.

This database has been picked, because receiving information instantly on any change is an important factor for this type of application. The data has to be returned instantly.

A timetable application is a type of application, where a user most likely will check on many times a day. A refresh button on the screen would not be as good for user experience as a realtime database which would update whenever there is a change detected.

Figure 4.19: Firebase Realtime Database Structure of this Project.

The above figures show how the firebase realtime database is structured in this project.

4.2.17 In-App Navigation

React Native provides many types of navigation. All with their pros and cons, different animations and functions. React Native offers the ability to

nest navigations on top of one another. This is especially used when the application is made in such a way, that there are drawers which have buttons which then move the user to a different screen etc.

In this project the nested Stack and Tab navigation have been used. Both of them play an important role in the application and are responsible for different screens and have connections in between them.

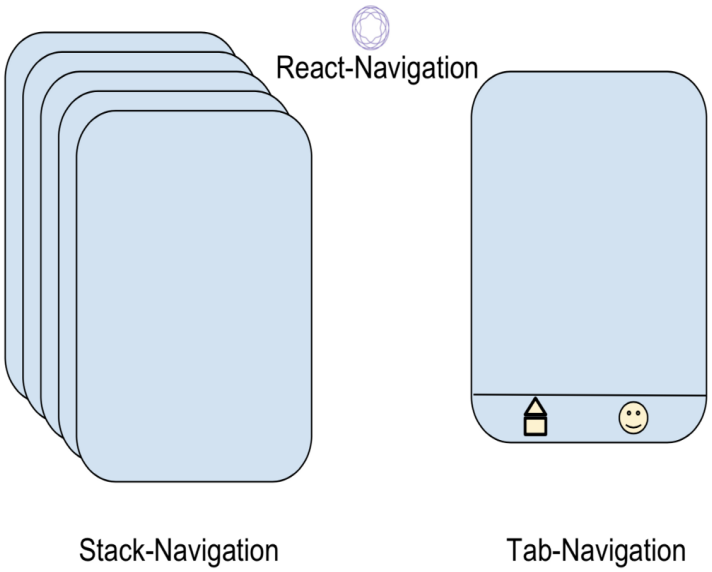

Figure 4.20: Tab and Stack navigation. Adapted from [14].

The Tab navigation is used for three main screens of the applications which are:

- Home Page
- Timetable
- Settings

These three screens can be picked from the tab navigator at the bottom of the screen. This tab navigator is available only on those three screens.

```
<Tab.Screen
  name="Timetable"
  component={TimetableScreen}
  options={{
    tabBarIcon: ({ focused }) => (
      <View
        style={{
          alignItems: "center",
          justifyContent: "center",
          marginTop: 0,
        }}
      >
        <Image
          source={require("./assets/icons/timetable.png")}
          resizeMode="contain"
          style={{
            width: 28,
            height: 28,
            tintColor: focused ? "#2b7eba" : "grey",
          }}
        />
        <Text
          style={{ color: focused ? "#2b7eba" : "grey", fontSize: 14 }}
        >
          Timetable
        </Text>
      </View>
    ),
  }}
/>
```

Figure 4.21: Tabs navigation in the project.

As shown on the figure above, that is how each tab is implemented in the code. Each tab has it's own icon, which is an image, but at a smaller size. All tabs have the same colours. They change colour when they are the active screen.

The Stack navigation is used for all other screens in the application. This includes the Login and Register and all the screens from the settings screen.

```
<Stack.Navigator>
  <Stack.Screen
    name="Login"
    options={{ headerShown: false }}
    component={LoginScreen}
  />
  <Stack.Screen
    name="HomeTabs"
    options={{ headerShown: false }}
    component={Tabs}
  />
  <Stack.Screen
    name="ChangePassword"
    options={{ headerShown: false }}
    component={ChangePasswordScreen}
  />
  <Stack.Screen
    name="ForgotPassword"
    options={{ headerShown: false }}
    component={ForgotPasswordScreen}
  />
  <Stack.Screen
    name="ChangeEmail"
    options={{ headerShown: false }}
    component={ChangeEmailScreen}
  />
  <Stack.Screen
    name="DeleteAccount"
    options={{ headerShown: false }}
    component={DeleteAccountScreen}
  />
  <Stack.Screen
    name="AddTimetable"
    options={{ headerShown: false }}
    component={AddTimetableScreen}
  />
</Stack.Navigator>
```

Figure 4.22: Stack navigation in the project.

As it is shown in the figure just above. The stack navigation is a very simple type of navigation compared to the tab navigation. Every screen of the application except the main three is a stack screen. Each stack screen in this project has their header disabled. This header shows a title of the screen on each screen if it is enabled. It has been disabled in this project as it does not look appealing.

These two types of navigation are connected together, by firstly loading the Login screen. Once a user is logged in, the stack navigation is calling the Tabs component, which actually calls a function called tabs. This function opens the home screen with the tabs menu at the bottom of the screen. From there, those screens from the tab menu have different screens that can be directed to. For example on the Settings screen, the user can switch to change email screen which is part of the stack navigation. This is the same logic as on the figure below.

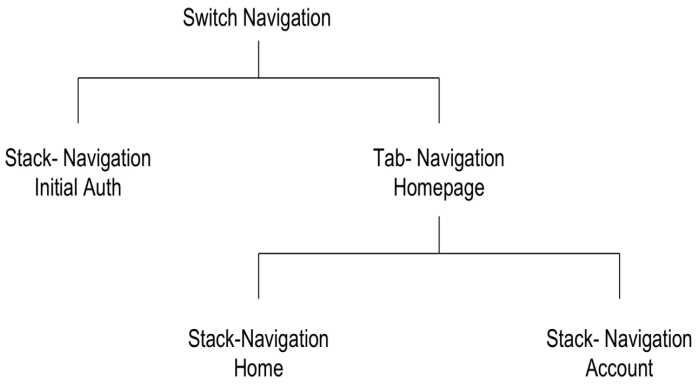

Figure 4.23: Connected Tab and Stack Navigation. Adapted from [14].

4.2.18 Types of Views

There were different types of Views used in this timetable application. The views used in this project were:

- **KeyboardAvoidingView** - This View allows to safely use the keyboard and input information into text boxes without the problem of having the information going out of the screen.
- **View** - Acts as a separator in the code. This view does the same work as tags in HTML for example.
- **ScrollView** - The ScrollView allows for the content on the page to be scroll able. It was used in this project, when creating a map to display all the timetable components from the realtime database.
- **SafeAreaView** - Thanks to SafeAreaView the content on the screen was placed in the right place as it was not obstructing or behind the navigation bar of the mobile device.

4.3 Web Application

Thanks to the fact that this application is built as a cross-platform, and with the use of Expo and React Native, the Web Application was also possible to build at the same time.

The Web Application of this project looked completely the same as the Android version of this project. It had some minor differences in the feedback animations when buttons were pressed, as well as the style of the application looked more stretched as the project was mainly aimed at Android devices, the web application was not taken into consideration. Nonetheless it was still looked at from time to time in order to test the functionality.

Some of the components that were used for the Android version of the project, were not working on the web application as some of the packages that were used only work for the Android devices.

4.4 Chapter Conclusion

This chapter has went over the two types of applications of this project. With the main focused application being the Android application. It described the many different features of the application, such as the navigation, the screens and the database.

Chapter 5

System Evaluation

5.1 Overview

This chapter will discuss the types of testings that were involved while developing the application. Thanks to this, the student will know if their software is stable and has been developed as it was planned.

5.2 Graphical User Interface Testing

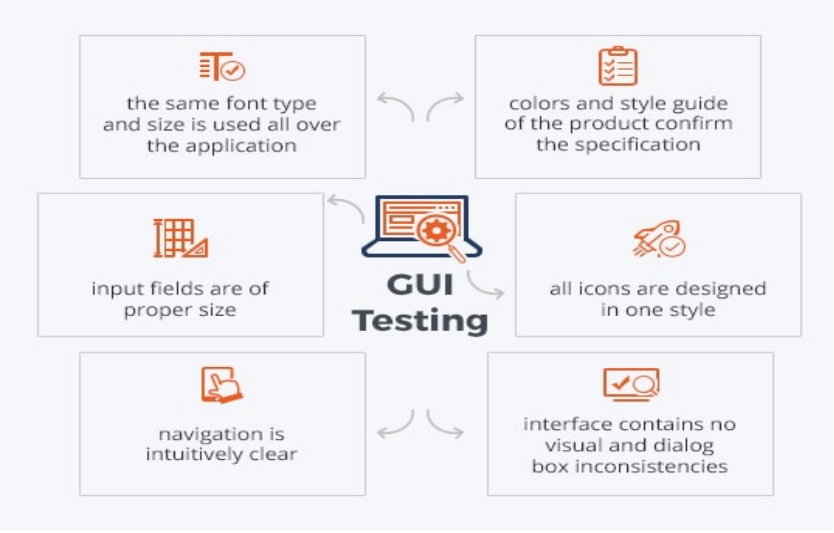

Figure 5.1: Example of GUI Testing. Adapted from [15].

Graphical User Interface Testing tests a piece of software with a Graphical User Interface (GUI). The piece of software that is being tested can have buttons, areas of text input, menus, text etc. [44]. These tests can be performed using certain software tools or by developers and designers opinion. It depends on what is being developed and how the development team sees it best.

In this project the student used Graphical User Interface Testing to test all of the components which are at the Front-End of the project. He tested to see if the buttons are of the right size and at the correct position. He had to make sure that all of the Front-End components match the style of the application and do not overlap each other. The student has completed this testing at their own discretion without the use of any software technologies.

5.3 Black Box Testing

Figure 5.2: Picture of Black Box Testing. Adapted from [16].

Black Box testing is a method of testing, that examines the functionality of an application without looking at the internal structures like code or technologies used [16].

This type of testing played an important role in testing of this application. It ensured that all software has been built correctly in the eyes of the user. That there are no major visual errors and that everything works as expected.

5.3.1 End to End Testing

Figure 5.3: Example of End to End Testing. Adapted from [17].

End to End Testing is a very important part of testing. It is used in order to test the functionality and performance of a piece of software. The idea of this test is to pretend what a real user experience would look like while using the application [45].

The student had to put himself into the mindset of a user and test the application as he would like it to work as a user of the application. During testing, he has checked if the application gave the right feedback to the actions. He tested if the correct information was retrieved and sent to the other systems of the project which for example were the Authentication and the Real-time Database.

5.4 Functional Testing

Figure 5.4: Picture of Functional Testing. Adapted from [18].

Functional Testing is a type of black-box testing. It tests the functions of a software project, by giving the software inputs and testing the outputs. The tests are strictly focused on the functionality and not the software used to develop the piece of software, hence it is a type of black-box testing [46].

The way that Functional Testing was used in this project was by testing each function one by one. This was a lengthy process, but without Functional Testing the student would not be able to see what feature does not work properly, or that it does not work at all and most importantly that the features work as they have been expected to.

5.5 Firebase Test Lab

Figure 5.5: Firebase Test Lab. Adapted from [19]

Firebase Test Lab allows for testing applications in their labs on real devices. It allows developers to set up their own tests, set a time for how long the tests should be run and the amount of devices they would like the tests to be run. There is a limit to the tests that can be done on the free plan of Firebase [47].

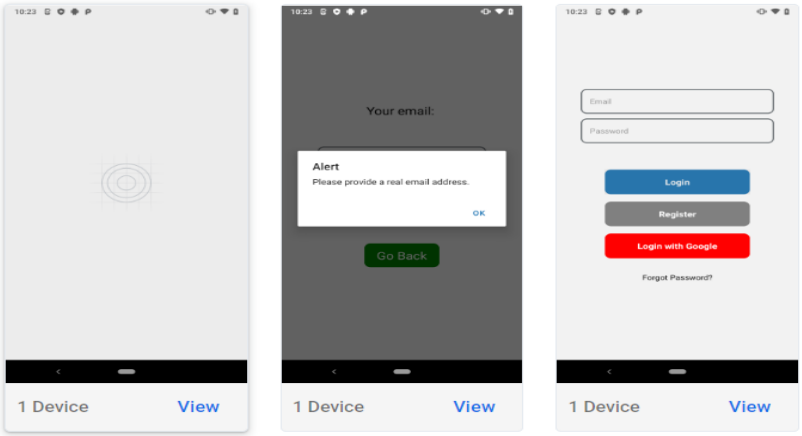

Figure 5.6: Timetable Application test result from Firebase Test Lab.

The developed timetable application was run through the tests, in order to make sure that the application behaves as expected. The figure above displays one of the tests that the application has been ran through. This

process was tedious but it was worth to see if the previous Functional and End to End tests also had a positive outcome.

5.6 Chapter Conclusion

The System Evaluation chapter, discussed the types of different testing methods that have been used in the project as well as provided some information about them.

Chapter 6

Conclusion

6.1 Overview

This is the final chapter of this dissertation it will conclude all the other chapters. Discuss the objectives and analysis of the project and also mention any negatives or problems encountered as well as the improvements that could be made in the future.

6.2 Objectives

This project had a few objectives. All of them had different importance. The objectives of this project are:

- To create a mobile application through which a user would have to use an email and password to log in or register.
- To create a timetable application in React Native which would display timetable information and allow for adding, deleting and modifying classes or modules to it.
- Publish the application to the Google Play Store or Apple App Store.
- Allow the user to create their own accounts in order to store their own personal data and be able to access it on any device with internet access.
- For the student to learn more about the technologies used to develop this project.

Not all of the objectives were met, but the student tried their best to accomplish all of them, while learning the new technologies and accompanying time for other course modules. The only objective that was not met was to publish the application to the Google Play Store or Apple App Store, as the application is not ready for that move yet. But continued work on this application has been planned and most likely will be continued after graduation.

6.3 Evaluation

This part of the conclusion will discuss the Improvement, Negatives and Overall experience of the project throughout the two semesters of the course.

6.3.1 Improvements

This application met most of the set objectives which were set at the beginning of the planning phase of the project. But with that there could be some improvements made to make the application seem and run better.

The reasons why the realtime database from the firebase database service was used, was because it was their first version of a database which means it is stable and that it is fully realtime. This database might in fact get deprecated in the future since they have created another database which has some differences in it. But it can not be made sure of yet.

One of the improvements that would have to be done to this application, would be the change from realtime database to the firestore database or even better work out a back-end database by the student themselves. But that would only be necessary if the realtime database was to get deprecated in the future. Which was not stated as of yet.

The much needed improvement of this application would be the front-end. As of right now the front-end of this application is good for testing and development, which is fine for the time being as the application is not released to google play store.

6.3.2 Negatives

Unfortunately there are always some negatives that could not have been avoided. There have been many problems encountered while developing this

project. Some of which include:

Unfinished Functions

The dark mode slider has been implemented in the project, but after many tries of implementing it. With all the knowledge of a new framework and technologies throughout the process of making this project. The functionality of dark mode has not been implemented.

Another unfinished function of this project is the Google Log In. Which would provide more security to the application, as Google provides great security features themselves on their accounts. This would also provide more options for the user, which some might find better to use. The work on this function has been done, but in the end the student was not able to make it in time, after many tries.

6.3.3 Future Plans

This project is a good starting point of making this application become something bigger and better in the future. Some of the future plans for this application are as follows:

- The first few things that would have to be done with this application are to fix all the negatives which were mentioned up above.
- The next would be to implement all the improvements into the application, or maybe even to find better solutions to the improvements.
- Publish the application up on the Google Play Store, once this timetable application gets to a good standard.
- With the application published on the Google Play Store and with hopefully some success, there could be some subscription fees or ads added to the application in order to monetise the project. This monetisation could fund all the monthly expenses that are being covered by the student while keeping this application public. Some of these expenses would include the payment for the Firebase plans if the student decides to stick with Firebase services.
- If the application would continue to be a success on the Google Play Store. The next step that would be taken, would be to ensure the iOS version of the application works and looks well. Once that would be

done, the timetable application would be published to the Apple App Store.

The good thing about making this a longer developed project is that the student will learn how to deal with newer version and updates of the technologies used, as well as learn on mistakes that would be made.

6.3.4 Overall

Throughout the two semesters working on this project. It has been a great experience to see how much planning, thinking and designing goes into creating a software project. It was a very enjoyable process to go through, and the student has learnt many lessons over the course of this year.

Thanks to this module, the student was able to see how a software developer feels with tight deadlines, which were created by himself. How much of a challenge it is to learn new technologies, and to force himself into learning more and more in order to become much better at what they would like to become and do in the future with more valuable knowledge and much needed experience.

It was very enjoyable but also challenging to undertake a project over a longer period of time than any other project given in this course. The weekly meetings with the supervisor forced self discipline and forced to be more organised out of the student. Which is a really positive outcome. Especially when the student is nearing an end on the course and heading into the industry very soon.

Over the duration of this course, the student was learning new technologies and learning more and more about software development where they gained more experience with software development. But this time, thanks to this project the student had to learn by themselves more than ever before. They used official documentations and used different methods of approaching the tasks while developing the project.

6.4 Chapter Conclusion

This chapter has concluded the whole dissertation. It went over the objectives that were met and not met. The improvements and negatives were also mentioned and the future plans for the application after graduation were discussed.

Bibliography

[1] L. K. University, "Can qualitative research be rigorous? part 1: What is qualitative research?."

[2] Breakout., "Rapid application development - 2021 complete guide."

[3] Wikipedia, "File:git-logo.svg."

[4] L. World, "Github logo."

[5] Wikipedia, "File:visual studio code 1.35 icon.svg."

[6] E. Hacks, "Everything you need to know before creating an android application."

[7] V. L. Zone, "Node.js."

[8] W. M. Commons, "File:npm-logo.svg."

[9] PNGKIT, "React native development - react native logo png."

[10] W. M. Commons, "File:firebase logo.png."

[11] Expo, "Logos."

[12] W. M. Commons, "File:latex logo.svg."

[13] Overleaf, "Tables."

[14] P. Doshi, "React navigation: Stack-navigation with tab-navigation."

[15] U-Tor, "Gui testing: What, why, how?."

[16] Wikipedia, "Black-box testing."

[17] A. R. Chowdhury, "All you need to know about end to end testing."

[18] C. Technologies, "Functional testing."

[19] Firebase, “Introducing firebase test lab.”

[20] Wikipedia, “Methodology.”

[21] Wikipedia, “Qualitative research.”

[22] Wikipedia, “Software development process.”

[23] Wikipedia, “Rapid application development.”

[24] Git, “Git’s official website.”

[25] Atlassian, “Why git for your organization.”

[26] P. Pedamkar, “Introduction to git.”

[27] Wikipedia, “Github.”

[28] CodeInstitute, “What is github?.”

[29] J. Clancy, “The advantages and disadvantages of using github.”

[30] Wikipedia, “Node.js.”

[31] O. Romanyuk, “Node.js is a great runtime environment - and here’s why you should use it.”

[32] AltexSoft, “What is node.js and its pros and cons.”

[33] NPM, “Npm official website.”

[34] A. Opidi, “Npm vs. yarn: Which package manager should you choose?.”

[35] R. Native, “React native’s official website.”

[36] Firebase, “Firebase’s official website.”

[37] Moqod, “Downsides of firebase: limitations to be aware of.”

[38] A. Studio, “Android studio.”

[39] A. Studio, “Meet android studio.”

[40] Expo, “Introduction to expo.”

[41] Expo, “Expo cli.”

[42] LaTeX, “Latex’s official website.”

[43] J. Wallen, "Why is android more popular globally, while ios rules the us?."

[44] I. Banerjee, B. Nguyen, V. Garousi, and A. Memon, "Graphical user interface (gui) testing: Systematic mapping and repository," *Information and Software Technology*, vol. 55, no. 10, pp. 1679–1694, 2013.

[45] SmartBear, "Combine api and ui testing for confidence at every layer of your application."

[46] Wikipedia, "Functional testing."

[47] Firebase, "Test in the lab, not on your users."