

---

# Timetable Mobile Application

---

**Oskar Ciebien**

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

APRIL 2, 2022

**Final Year Project**

Advised by: Mr. Martin Hynes

Department of Computer Science and Applied Physics  
Galway-Mayo Institute of Technology (GMIT)



# Contents

<b>1</b>	<b>Introduction</b>	<b>5</b>
1.1	Layout of the Dissertation . . . . .	5
1.2	Project Source Code and Documentation . . . . .	6
1.3	Goal of the Project . . . . .	6
1.4	Project Timeline . . . . .	7
1.5	Project Difficulties . . . . .	7
1.6	Learning Outcomes . . . . .	7
1.7	Chapter Conclusion . . . . .	8
<b>2</b>	<b>Context</b>	<b>9</b>
2.1	Overview . . . . .	9
2.2	Chapter Conclusion . . . . .	9
<b>3</b>	<b>Methodology</b>	<b>10</b>
3.1	Overview . . . . .	10
3.2	Research Methodology . . . . .	10
3.3	Software Methodology . . . . .	11
3.4	Meetings . . . . .	11
3.4.1	Meetings at the Beginning . . . . .	12
3.4.2	Regular Meetings . . . . .	12
3.5	Source Control . . . . .	12
3.5.1	Why was it picked? . . . . .	12
3.6	Tools . . . . .	12
3.6.1	Visual Studio Code . . . . .	13
3.6.2	Android Studio - Emulator . . . . .	13
3.7	Chapter Conclusion . . . . .	13
<b>4</b>	<b>Technology Review</b>	<b>14</b>
4.1	Overview . . . . .	14
4.2	Technologies Used . . . . .	14
4.3	Git . . . . .	14

4.3.1	What are the Advantages? . . . . .	14
4.3.2	What are the Disadvantages? . . . . .	15
4.4	GitHub . . . . .	15
4.4.1	What are the Advantages? . . . . .	15
4.4.2	What are the Disadvantages? . . . . .	15
4.4.3	Node . . . . .	16
4.4.4	React Native . . . . .	16
4.4.5	Firebase . . . . .	16
4.4.6	Android Studio . . . . .	16
4.5	Chapter Conclusion . . . . .	16
<b>5</b>	<b>System Design</b>	<b>17</b>
5.1	Overview . . . . .	17
5.2	Web Application . . . . .	17
5.3	Android Application . . . . .	17
5.4	Chapter Conclusion . . . . .	17
<b>6</b>	<b>System Evaluation</b>	<b>18</b>
6.1	Overview . . . . .	18
6.2	Graphical User Interface Testing . . . . .	18
6.3	End to End Testing . . . . .	19
6.4	Functional Testing . . . . .	20
6.5	Chapter Conclusion . . . . .	20
<b>7</b>	<b>Conclusion</b>	<b>21</b>
7.1	Overview . . . . .	21
7.2	Context and Objectives . . . . .	21
7.3	Evaluation . . . . .	21
7.3.1	Improvements . . . . .	21
7.3.2	Downfalls . . . . .	21
7.3.3	Overall . . . . .	21
7.4	Chapter Conclusion . . . . .	21

# About this project

**Abstract** This project is a Timetable Application developed for Mobile Devices. It differs from some of the other timetable applications currently available on the App Store and Google Play Store, because it has a common and a reliable security feature. This feature is the Google Two Factor Authentication. The user can disable or enable this feature at their own discretion. This Timetable Application also has a customisation feature which is the dark mode. This dark mode allows the user to switch between dark and light modes in the app. It changes the application theme colours from dark to light and vice versa. This feature will provide ease of use at any time of the day. Most importantly this mobile application will have easy to use Front-End interface. With a menu at the bottom of the screen with options to be picked from. In the application the user will have many screens to choose from. The Timetable screen, which displays the weekly schedule. The settings screen which offers settings in relation to the securities and the choice of theme within the application as well as account options. There is also the main menu screen which displays the information about the application and the author of the application. I have used React Native to develop this application and Firebase for the Back-End of this Timetable Application, which will store all of the user information, settings, and timetable data. The database merges whenever an internet connection is established with the device. The database that I have used is also real-time, which allows for instant display of data when it has been modified.

**Authors** Oskar Ciebien is a final year student at Atlantic Technological University (ATU) formerly known as Galway-Mayo Institute of Technology (GMIT). Currently pursuing a Bachelor of Science (Honours) in Computing in Software Development.

# Chapter 1

## Introduction

This is the introduction chapter of this dissertation. It outlines the layout, the objectives, the redirection to 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

The following chapters will cover different aspects of the dissertation:

- **Chapter 1 - Introduction** - Chapter 1 which is the current chapter, will describe the project objectives, the redirection to 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 - Context** - Chapter 2 will explain the overall planning of the project.
- **Chapter 3 - Methodology** - This chapter will discuss the approach on the project, the tools used while developing the project, source control, the meetings conducted with the supervisor and the research which was done throughout the process of creating this project.
- **Chapter 4 - Technology Review** - Chapter 4 will provide a review on all the technologies and the languages that have been used.
- **Chapter 5 - System Design** - Will describe the project as a whole, the design and applications developed.

- **Chapter 6 - System Evaluation** - Will discuss the types of testing that have been carried out while developing the application.
- **Chapter 7 - Conclusion** - The final chapter will conclude the project and discuss the evaluation, objectives, downfalls and the improvements that could have been done on the project.

## 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](https://github.com/Oskar-Ciebien/Timetable_Application).

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

All the documentation and videos are available at the repository in the folder called Documents.

## 1.3 Goal of the Project

The goal of this project was to create a Timetable application, which would differ from most of the timetable applications currently available on Google Play Store or Apple App Store.

The project will also allow 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 database and could be accessed on any device after logging into the account.

The goal of the timetable was to be editable with or without an internet connection, but to sync whenever an internet connection would be established.

At the start the initial idea was to develop a mobile timetable application and to publish it on Google Play Store, as the application is mainly targeted at Android Devices. This idea will hopefully come true, as the application will most likely stay updated and worked on after submission.

## 1.4 Project Timeline

## 1.5 Project Difficulties

As I was developing the timetable application I have faced many difficulties. The difficulties that I have faced are as follows:

- Get familiar with the technologies that I have chosen - React Native, Firebase and Android Studio. I never used any of them before. I sometimes found it difficult to understand the logic of those, but thanks to this course I was able to adapt to new technologies quickly.
- My first project idea. I decided to restart the project completely after the Christmas review. It was all done to make the project seem and work better.
- Time and priorities were definitely large enemies while developing this project. I sometimes was forced to decide what to focus on first, or how much time I should spend developing a feature and if I was not able to finish it in the given time, then I should focus on something else. This solution worked very well for me and enabled me to be more efficient.

## 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.

## 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, timeline and difficulties have also been set out. Finally the learning outcomes of the module have been written out.



# Chapter 2

## Context

### 2.1 Overview

### 2.2 Chapter Conclusion

# Chapter 3

## Methodology

### 3.1 Overview

This methodology chapter will describe the various methodologies that have been used in developing this project. It goes over the Approach, Meetings the Source Control and the Tools that have also been used. Additionally it will go over some Research methods that were taken into account before making decisions.

### 3.2 Research Methodology

There are different approaches to research. These approaches enhance the process and proves the way the research has been conducted [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 [2].

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

### 3.3 Software Methodology

Software Methodologies allow us to divide up the development process into smaller steps which in the end can help with design, development and management of the project [3].

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 is developed by only one person, it allows for developing many features and then deciding which of the few options to keep [4]. This way the development of the application or project is simpler as decisions can be made at nearly any time.

RAD is made up of many phases. These include the following [4]:

- **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 see.
- **Construction Phase** - Building of the project. But as this is RAD, any changes or improvements can still happen. This phase also includes testing of the project or application.
- **Cutover Phase** - Final phase. It includes final testing and user training in some situations.

### 3.4 Meetings

The project meetings were scheduled weekly. They were attended by both the supervisor and myself. They were all carried on-line as it was more comfortable. The meetings were under fifteen minutes long.

### 3.4.1 Meetings at the Beginning

The early meetings were based on the planning and possible ideas for the project. The project timeline and the project design brief has been done during those meetings. These early stage meetings took a couple of weeks.

### 3.4.2 Regular Meetings

Regular meetings consisted of:

- Current progress of the project.
- The problems encountered.
- Feedback from the supervisor.
- Plans for the following week.
- Changes in the project brief or weekly project timeline.

## 3.5 Source Control

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

### 3.5.1 Why was it picked?

This was the preferred source control as it is the most preferred and used source control in the industry. As well as all the student's project are based on GitHub which serve as the student's portfolio for the future.

## 3.6 Tools

In order for this project to be developed in the most convenient way. The most popular and best tools had to be used.

### 3.6.1 Visual Studio Code

Visual Studio Code has been used in development of this project as it a very popular, easy to use and widely modifiable piece of software, thanks to many developers working on extensions in order to improve it.

This Integrated Development Environment (IDE) has been used throughout the years of the Software Development course. Thanks to that, it made the ease of use better, as the student was already used to this tool.

### 3.6.2 Android Studio - Emulator

Since this project is aimed at Android devices. An emulator has been used to test the application in the quickest way possible.

The use of an emulator while developing an application is also convenient as there is no delay while using. The developer does not have to wait for the application to install on an android device etc. it is readily available straight away with no cables in the way. With the use of the same device that is used to code.

## 3.7 Chapter Conclusion

This Methodology chapter has went through the research methodology and software methodology that has been done in the process of making this project. It also discusses the meetings that have been conducted with the supervisor. Git and GitHub have been explained as the source control of this project, and finally the two main tools that were used by the student accordingly.

# Chapter 4

## Technology Review

### 4.1 Overview

In this chapter, the different types of technologies that have been used while developing this project will be discussed with reasoning why they have been picked over the alternatives.

### 4.2 Technologies Used

### 4.3 Git

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 [5].

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 [5].

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 biggest companies in the world such as Google, Facebook, LinkedIn and Microsoft [5].

#### 4.3.1 What are the Advantages?

One of Git's advantages is how distributed it is. Instead of using one repository, each of the developers has their own local repository and a history of

commits [6]. This makes it fast, scalable and easier to work with.

Git is one of the most popular version controls. Thanks to that, nearly all developers have had experience with Git and will not have to learn it from the beginning [6].

### 4.3.2 What are the Disadvantages?

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

One disadvantage of this piece of software is that it is slower on Windows machines [7]. Which could be a big downside for power users of Windows machines when using Git for development.

Git also does not support binary files [7], which could be important for some developers.

## 4.4 GitHub

### 4.4.1 What are the Advantages?

The Pages that it offers allows developers to showcase their projects and portfolios on a static page for free. They can be customised as the developers would like it with the paid themes [8].

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 on the phone only to view the repositories and projects [8].

### 4.4.2 What are the Disadvantages?

With Git having some disadvantages, and GitHub uses Git, this also shows that it does have some bad sides as well.

As any other service provider websites, GitHub also provides services, but not all are free. For users that do require a lot of repositories, teams etc.

this could mean that they will be forced to buy a plan [9].

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.

### **4.4.3 Node**

### **4.4.4 React Native**

React Native is a JavaScript based framework. Facebook has released React Native 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 [10].

React Native also offers a great feature called "Fast Refresh", this feature allows for as the name suggests a fast refresh while developing an application. The application refreshes on saving a file, without the need of restarting the application [10].

### **4.4.5 Firebase**

### **4.4.6 Android Studio**

Android Studio is a great piece of technology which provides tools for building applications for any Android Device [11]. 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.

I have used Android Studio for my android emulator. I have used the Google Pixel 4 XL with Android version 11 and API 30. Android Studio is a great program, not only it allows for building apps, but also it has emulators.

## **4.5 Chapter Conclusion**



# Chapter 5

## System Design

### 5.1 Overview

This chapter goes over the system design for the project. Since React Native allows for developing many applications with one code, this chapter will also go over the two types of applications this project was developed for.

### 5.2 Web Application

### 5.3 Android Application

### 5.4 Chapter Conclusion

# Chapter 6

## System Evaluation

### 6.1 Overview

This chapter will discuss the types of testings that were involved while developing the application. This way it will analyse various aspects of the project.

### 6.2 Graphical User Interface Testing

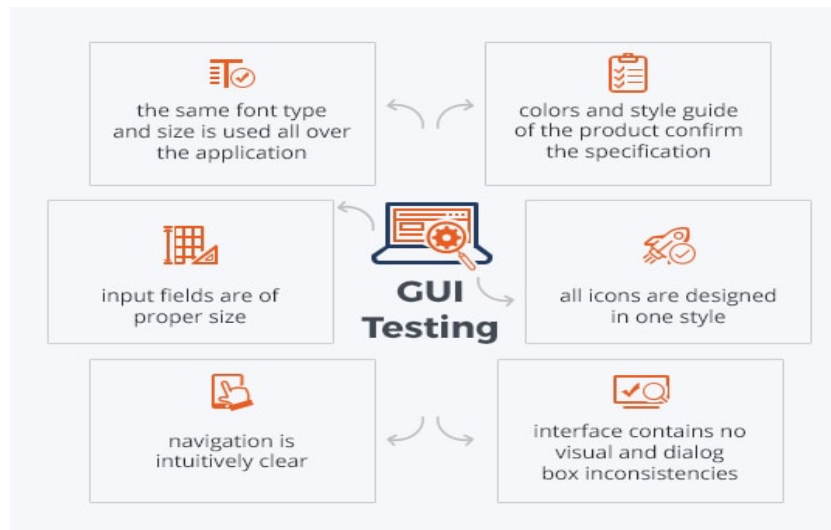


Figure 6.1: Example of GUI Testing. Adapted from [12].

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 [13]. These tests can be performed using certain software tools or by developers and designers opinion. It depends what is being developed.

In this project I have used Graphical User Interface Testing to test all of the components at the Front-End of my project. I have tested to see if the buttons are of the right size and at the correct position. I had to make sure that all Front-End components match the style of my application and do not overlap each other. I have completed this type of testing by going using my opinion on the elements of the Front-End.

## 6.3 End to End Testing



Figure 6.2: Example of End to End Testing. Adapted from [14].

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 [15].

In my project I have put myself into the mindset of a user and tested the application as I would like it to work as a user of the application. During

testing, I have checked if I was given the right feedback to my actions. I have tested if I could retrieve the right information as well as send the right information to the other systems of the projects which for example were the Authentication and the Real-time Database.

## 6.4 Functional Testing



Figure 6.3: Example of Functional Testing. Adapted from [16].

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 [17].

I have used Functional Testing in this project by testing each function one by one. This was a lengthy process, but thanks to that I was able to see what I have done wrong or not yet implemented so that a specific function could work as expected.

## 6.5 Chapter Conclusion

# **Chapter 7**

## **Conclusion**

### **7.1 Overview**

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

### **7.2 Context and Objectives**

### **7.3 Evaluation**

#### **7.3.1 Improvements**

#### **7.3.2 Downfalls**

#### **7.3.3 Overall**

### **7.4 Chapter Conclusion**

# Bibliography

- [1] Wikipedia, “Methodology.”
- [2] Wikipedia, “Qualitative research.”
- [3] Wikipedia, “Software development process.”
- [4] Wikipedia, “Rapid application development.”
- [5] Git, “Git’s website.”
- [6] Atlassian, “Why git for your organization.”
- [7] P. Pedamkar, “Introduction to git.”
- [8] CodeInstitute, “What is github?.”
- [9] J. Clancy, “The advantages and disadvantages of using github.”
- [10] R. Native, “React native.”
- [11] A. Studio, “Android studio.”
- [12] U-Tor, “Gui testing: What, why, how?.”
- [13] 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.
- [14] A. R. Chowdhury, “All you need to know about end to end testing.”
- [15] SmartBear, “Combine api and ui testing for confidence at every layer of your application.”
- [16] C. Technologies, “Functional testing.”
- [17] Wikipedia, “Functional testing.”