Increase Time Efficiency of Students using a Student Manager Application

BSc in Computer Science

School of Computing, Science and Engineering

Final Year Project Report

Author: [redacted]

Supervisors: Ian Drumm and Norman Murray

Academic year: 2020-2021

Word Count Part 1: 4478

Word Count Part 2: 9421

ABSTRACT

Time management is one of the key skills needed for a student to be successful in their university journey (Britton & Tesser). This has become increasingly difficult especially now that students are learning remotely and not getting the full university experience. This dissertation will attempt to improve this by providing a tool that students can use that will help them manage their time more efficiently. An in-depth investigation has been done to see what affects student grades the most, and it is shown that the two biggest factors are mental health and time management (Randstad, 2019). It seems that one also affects the other, so improving time management skills will indirectly help with one's mental health. Therefore, providing this application is important and can change the lives of many students.

React Native was used to build the application and agile was used for the development methodology with three sprints. Each sprint focused on implementing a specific set of features that was gathered from a focus group. Several forms of tests such as unit testing, acceptance testing, and regression testing took place to see if those requirements have been met and if the report has achieved the overall goal. Based on the feedback of the testers, it can be said that the application achieved its goal as the testers concluded that using the application would have made a huge difference in their first year of studies.

TABLE OF CONTENTS

TABLE OF FIGURES

1 INTRODUCTION

1.1 MOTIVATION

There is a rising concern when it comes to the mental health of students in higher education as it has recently been on the decline. This is a major issue that universities and the government as a whole need to address and find a long-term solution for. At the time of writing this paper, students are also dealing with a pandemic which further emphasises the need for a solution. It has been shown that students that suffer from mental health issues tend to perform worse than those that do not. This can be shown by their grades and the interactions student have with the course. A major component that can help students with this is for them to take back control of their time and improve their time management skills.

1.2 PROJECT AIM

This project will attempt to make the life of first-year Computer Science students at the University of Salford easier by creating an application that helps them manage their time more efficiently. This will give them more control and awareness of their daily activities and in turn, not be overwhelmed by the workload they have. The aim is to make the application easily accessible, easy to use, and fit for purpose.

1.3 CORE OBJECTIVES

1. Explore a variety of development approaches suitable for building a mobile application such as Android studio, Angular, React, Xamarin, etc before choosing an approach suitable for developing the final product. **(Sprint 1)**
2. Survey students as to what they would like to see in the application and design the product based on their feedback. Create use cases for the user interaction scenarios and UML diagrams to represent the architecture of the system. **(Sprint 1)**
3. Investigate different hosting services such as Poseidon (university server), AWS, docker swarm, etc to host the database and scrips that will be used by the application. **(Sprint 1)**
4. Research into building Restful APIs using the appropriate technologies. **(Sprint 1)**
5. Allow students to have the ability to customise their lessons; add additional activities; mentioned missed lessons to alter the timetable to go to a different slot; mention subjects they are struggling with to add more lessons of that subject in the timetable. **(Sprint 1 & 2)**
6. Students will have the ability to interact with each other by creating a chat option in the lessons where support can be provided, and questions can be answered by peers **(Sprint 3)**.
7. Investigate biometrics and geolocation with the chosen technologies to let users register to classes. Lecturers could be altered if there are students that have low attendance **(Sprint 3)**.
8. Explore the best way to have a user-friendly to-do list and implement it so students can manage their tasks easily **(Sprint 3)**.
9. The application needs to be accessible to all. Investigate vision impairment, colour blindness, voice option, change font, and size of the text.
10. Evaluate the product at each stage by running different tests such as Unit Testing, Acceptance Testing, UI Testing, Load Testing, and Regression Testing to make sure that the product is meeting the requirements.

1.4 OPTIONAL OBJECTIVES

1. Add a feedback page where students can submit feedback to improve the user experience and usability of the application **(Sprint 4)**.
2. Add a superuser that can directly make changes to certain elements of the applications that normal users cannot and give them the power to monitor user interaction with the application to see where students spend most of their time **(Sprint 4)**.

1.5 PROJECT APPROACH

The project will follow the Agile Methodology where it will be broken up into three sprints. A demo, along with an evaluation will take place at the end of each sprint. The objectives for each sprint will be based on a task priority matrix that was collected during a focus group session.

1.6 EXISTING SOLUTIONS AND PROJECT CONTRIBUTION

Time management applications have been around for a long time and provide amazing support to people. However, these applications are generic and target the masses – even those that advertise them to students. Therefore, to the best of my knowledge, there is not an application in the market that is designed and customised for students that do not involve a heavy setup. This project will fill the market gap by creating an application that is adapted and customised to work with university students without much upfront configuration. A more detailed look into the different applications will be covered in a later section.

1.7 LIMITATIONS

The project is aimed towards First Year Computer Science Students at The University of Salford. Therefore, it will not be suitable for students from other courses or universities. However, the infrastructure will be in place to easily extend the application to accommodate several courses and multiple universities. The scope of the project was intentionally small to focus on the aims of the project which was proving the hypothesis that using a student manager application will help students with their time management skills and will in turn improve their grades.

2 LITERATURE REVIEW

2.1 PROBLEM AREA

2.1.1 MENTAL HEALTH AND STUDENT PERFORMANCE

The mental state of students plays a major role in their university experience. It has been shown that mental health is one of the most detrimental factors to a student's performance and grades. Over a quarter of the students in the UK suffer from mental health and this figure grows each year (Randstad, 2019). This is caused by the university lifestyle and the new environment that the students find themselves in. This is also backed by research done using a national dataset which shows that the mental health conditions of students "significantly increased over the study period (Randstad, 2019). This is a huge issue that the government has failed to address and find a solution to. The situation has gotten worse due to the pandemic and mental health has reached an all-time low.

There is a clear correlation between the mental health of a student and how engaged they are with their studies. A psychological assessment was done that used a dual-factor mental health model on 561 higher education students. The results showed that "students without clinical symptoms, those with high subjective well-being were more engaged in the college experience than their peers with low subjective well-being. (Antaramian, 2015)". Another assessment was done where 195 students were interviewed. The researchers found that "71% indicated increased stress and anxiety due to the COVID-19 outbreak (Farzan Sasangohar, 2020)". Two major factors contributed to this. One is the difficulty in concentrating at home, and the other is the increased concern with academic performance.

89% of the interviewed students said they had difficulty concentrating on university work (Farzan Sasangohar, 2020). Lockdown played a role in this. They reported that studying from home was difficult as it was a "distractive environment and a more suitable place to relax rather than to study (Farzan Sasangohar, 2020)". Another major factor that affected their concentration was the lack of accountability and lack of interaction during online lectures. All this contributes to students not being able to focus on their general studies which causes them to fall behind and that leads to them getting lower marks.

82% of the interviewed students said they had concerns about academic performance and how transitioning to remote education will affect them (Farzan Sasangohar, 2020). The major area of concern was the quality of teaching and the constant technical issues. The "participants mentioned the uncertainty about their grades under the online learning environment to be a major stressor (Farzan Sasangohar, 2020)". This resulted in many students losing their motivation to learn and developing tendencies to procrastinate.

2.1.2 TIME MANAGEMENT AND GRADES

Time is a limited and very important resource. Therefore, time management is one of the most important skills a person needs in order to become successful. It is shown that having good time management skills will increase student performance and is “related to academic achievement (Britton & Tesser)”. However, it is shown that when a student is dealing with mental health issues, productivity and motivation significantly drops (Farzan Sasangohar, 2020), which in turn affects a student’s ability to manage their time. Therefore, many university counselling services offer time management as a coping mechanism to tackle stress and anxiety (Macan, et al.). Another reason why counsellors push students to work on their time management skills is that “Students who perceived control of their time reported significantly greater evaluations of their performance (Macan, et al.)”.

2.1.3 SUMMARY

There is a clear correlation between the mental health state of students and how well they do in university. It has been shown that students that manage their time efficiently will do better in university than those that do not. This will also prepare students for the future as time management skills are essential for most challenges they will face. Being able to perceive control of time helps with mental health issues as it gives students the ability to gain back control over certain aspects of their life which will, in turn, reduce their stress and anxiety.

Overall, there is a need for students to be able to manage their time efficiently in order to have the best university experience and to also graduate with good grades while maintaining good mental health. Therefore, it is important and very beneficial for every student to have a service that makes it easier for them to manage their time and take control of their day.

2.2 EXISTING PRODUCT

Many time management applications in the market help people stay productive and achieve their daily goals. Some of those applications are general purpose and can be used by everyone, while others are geared more towards students to help them through their university journey. This section will investigate some of the top-rated applications on the market which will help identify the market gap that this project will attempt to fill. Having said that, this will not determine the features that the project application will have. There will be a different process that involves focus groups and user requirement gathering which will be discussed in a later section.

The criteria that will be looked at for each application are:

- User interface
- Features (pros and cons)
- Overall opinions

2.2.1 BOOSTED (HTTPS://PLAY.GOOGLE.COM/STORE/APPS/DETAILS?ID=COM.BOOSTEDPRODUCTIVITY.APP):

Boosted is a productivity and time tracking app with a 4.7 rating. The user interface of the application was very professional. However, the application was not intuitive, and it took some time to understand how to use it. Some of the features the application offered were only accessible on the paid version. Setting up the application takes a long time since students need to add their timetable from scratch. This is difficult to do since the application does not have a defined class or module option to pick from. Another downside is that it is not possible to display tasks on the calendar that the app offers. Overall, the only good thing about this application is the design and the concepts behind the features – since some of it does not work.

2.2.2 MY STUDY LIFE (HTTPS://PLAY.GOOGLE.COM/STORE/APPS/DETAILS?ID=COM.VIRBLUE.MYSTUDYLIFE):

My Study Life is a school planner app with a 4.4 rating. The application has a simple user interface and offers a lot of features that were easy to use. The main page is a dashboard to see the classes, exams, and tasks the student has which is useful. The process for adding a timetable was done manually but the application tried to simplify it by having distinct options for classes, exams, and tasks. Everything was free on the app which was nice since payments usually deter students from using the application. The only downside to the app was that students did not have the option to import their timetable or have it set with minimum configuration.

2.2.3 MY HOMEWORK (HTTPS://PLAY.GOOGLE.COM/STORE/APPS/DETAILS?ID=COM.MYHOMEOWORK):

My Homework is a student planner app with a 4.2 rating. The user interface looks old but was still easy to use. When the app first starts, it allows students to add classes, assignments, tests, and projects. This process takes a long time since it needs to be done manually. However, once that is done, the changes could be displayed in the calendar to keep track of. A very interesting feature that this application offered is that it connected to a different app called teacher.io which allowed teachers to interact with the students. Overall, the application was good and offered a good range of features that were easy to use. Besides the old-looking design, and the tedious work of setting up the timetable, it is a good app for students to use as a time management tool.

2.2.4 SCHOOL PLANNER (HTTPS://PLAY.GOOGLE.COM/STORE/APPS/DETAILS?ID=DALDEV.ANDROID.GRADEHELPER):

School planner is an app that helps students stay organised and has a rating of 4.6 stars. The user interface is simplistic yet easy to navigate and offers a lot of powerful features. The entire app uses google packages and the layout is very similar to those you would find in applications made by google. Setting up subjects and classes is easy but time-consuming. The app does not offer the user to ability to import their timetable. An interesting feature of the app is the ability to store the student's grades, attendance, and allow teachers to interact with the students. They also have a calendar view and a timetable view which give the students two different perspectives of what is to come during the day and week. This is something useful as it gives the student a better picture of what they need to do for the upcoming days. Overall, this is a great application and is the type of quality the project is aiming to achieve.

2.2.5 SUMMARY

Most of the applications that were looked at offered similar features such as adding classes and displaying them on the calendar. However, the apps differed when it came to design and user interface and was what made them stand out. Therefore, design needs to be a big part of the development process when it comes to building the student manager application.

Comparing these apps also gave me an insight into what they were missing. None of them had an import option or a built-in timetable. This meant that the students had to set up their timetable which is time-consuming. Some of these apps only had options to display school-related events such as lessons and exams. This restricted the user in my opinion, and it would have been better to allow students to add other activities as well. This would have meant that students have all their tasks in one place which would have allowed for easier tracking.

To conclude, reviewing these apps was very useful as it gave me lots of features and design ideas that could be used when building the application. However, as stated before, a focus group will be held to determine the main features that will be built for the application.

3 METHODOLOGY

3.1 DEVELOPMENT METHODOLOGY

The development methodology will have a huge impact on the successful completion of the project. Therefore, it is very important to pick one that is best suited to work well with this project. After researching and comparing 4 different methodologies (waterfall, rapid application, agile, and DevOps), the Agile methodology will be used due to its approach and advantages (Team, 2017). Thanks to its iterative approach and incremental deliverables, it focuses on working on individual features and delivering small end-to-end functionality to get feedback on them. It is also amazing for managing risk, such as spotting bugs early; and its versatility when it comes to changing or adding new features (Terry, n.d.). This is exactly what is needed for the project since a big part is getting feedback from the supervisors and aligning expectations in order to progress in the right direction.

The Scrum Framework will be used for the project. Scrum is an agile process that focuses on delivering products within a short period of time through incremental deliveries (Anon., n.d.). These incremental deliverables, also known as sprints, will be based on the objectives of the project. The aim is to break the project into 3 sprint iterations. The first sprint will end a week before Christmas, the second will end in February, and the last one will end in April. For each iteration, there will be 3 phases, the planning phase, the demo phase, and the retrospective phase.

The planning phase will answer the following questions:

- what are the objectives of this sprint?
- how will those objectives be achieved?
- what will be demoed to the supervisors?

The demo phase will display the work done to the supervisors and get feedback on the progress made to see if the sprint goals were met. The retrospective phase will be a reflection of the previous sprint, the challenges that the project came across, and how to improve the next sprint.

3.2 DEVELOPMENT TOOLS

There is a wide range of different technologies and platforms to choose from when building a mobile application. Those technologies were divided into two categories; native applications (Android Studio for Android & XCode for iOS), and cross-platform applications (React Native & Xamarin). A study was undertaken in order to determine the best technologies to use for the project. This was done in two ways:

- Building a small application in each platform.
- Taking a deeper look into the performance of each platform.

3.2.1 NATIVE APPLICATIONS (ANDROID STUDIO & XCODE)

Android Studio is a Java/Kotlin/C++ and XML-based platform where android applications can be developed for different android devices. It is the official development environment and was built in 2013 by Google and JetBrains (Anon., n.d.). XCode is the official development environment for iOS applications. It is a Swift-based platform and was created by Apple in 2003 (Anon., n.d.).

Both Android Studio and XCode are native applications on their respective platforms so they have "full and direct access to the native platform functionalities" (Ebone, et al., 2018). This allows them to deliver applications with the best performance. An experiment done at Adekunle Ajasin University shows the difference in response time and the data usage on CPU in native applications and cross-platform applications. The results show that "while the native application is nearly immediately loaded, the hybrid counterpart is significantly slower by a factor of around 20 seconds (Ajayi, et al., 2018)". This shows that in terms of speed and performance, the best choice would be to go with a native approach.

On the other hand, native applications are limited to only being able to compile and deploy to their respective operating systems. In order to reach a wide audience, an application on both Android Studio and XCode would need to be built which would take a lot of time.

During the process of building the MVP in Android Studio, it was very easy to create the application. This is because java is the primary language that was taught in university and Android Studio is being taught in the final year. A simple page was created displaying a calendar to test out the difficulty and how the platform will perform. The experience was very smooth, and it was very easy to create the UI components. On the other hand, the MVP in XCode did not work out since there was no Mac device to test it on. An attempt was made to create a Virtual Machine (VM) and host MacOS on that machine then test the application that way. However, there were lots of complexities involved and the attempt was given up after a few days.

3.2.2 REACT NATIVE

React Native is a JavaScript-based framework created by Facebook in 2015. It is open-source and is used for mobile development. The framework is cross-platform, so it allows the application to compile to different devices using the same code base which saves a lot of time and resources (Anon., 2015).

When it comes to performance, a study has been done that found "applications developed with React Native did not perform as well as the native and Xamarin versions (Furuskog & Wemyss, 2016)". On the other hand, when comparing user experience in native platforms to React Native, there is little to no difference. This was shown with the User Experience Questionnaire (UEQ) and System Usability Scale (SUS) which "pointed towards equal user experiences since no significant differences could be found in the results of the UEQ and SUS (Hansson & Vidhall, 2016)".

Setting up the React Native environment was more complicated than expected but once the setup was complete, it was easy to use. Expo CLI was picked among the two options which were React Native Cli and Expo Cli since it has 3rd party wrappers that offer a lot of useful features, and you can easily convert to React native CLI if the need arises (Piyadigama, 2020).

A simple page was created for the MVP in React Native which displayed a React Native Calendar, and it took a day to achieve the task. It was easy to learn the structure and get started on React Native which meant that the development time would not take as long as other platforms. It also uses JavaScript which was taught in university and there is a big online community that provides support.

3.2.3 XAMARIN

Xamarin is a free and open-source tool that allows users to build Android and IOS applications using .NET and C# along with the Visual Studio environment. Xamarin was created by Microsoft in 2011 (Anon., 2011).

A study done at the University of Gothenburg compared 20 different cross-platform applications and weighted them against 10 architectural principles. These principles were taken from the Volo Group IT Governance. Xamarin ranked top 5 in several of them: "Autonomous and loose coupling, between components; Robust solutions & performance focus from the start; Maintainable solutions (Pazirandeh & Vorobyeva, 2015)". This shows that Xamarin has a strong base to work in since it was built with these principles in mind. Another experiment was done to test out the CPU and memory usage for android devices that ran using Xamarin. The results were impressive as it showed that for an application the size of 29.7MB, it only took 3% CPU space and required 125.34MB of memory usage (Goetz & Li, 2018).

An astonishing result was that Xamarin can achieve "75% shared code base for all platforms and nearly 100% in creating user interfaces using Xamarin.Forms (Furuskog & Wemyss, 2016)". This is amazing as it minimises the chances of there being UI issues across the different devices and gives you the confidence to know that your application will look exactly how you want it to look across different devices.

While working on the MVP in Xamarin, it became apparent that an iOS emulator would be needed to run the iOS application. Since there were no iOS emulators, it was not possible to test out the iOS application. However, it was still possible to write the iOS application on windows which was impressive (Pazirandeh & Vorobyeva, 2015). When it comes to the structure and code of Xamarin, it was not very straightforward, and it took some time to get used to the structure. Another barrier was the C# language as it was not familiar so this would slow down the development time.

3.2.4 SUMMARY

Four different platforms were compared to determine which technology would be best suited for the project. The native applications were the best in terms of performance. However, they are limited in terms of reach. On the other hand, the cross-platforms can compile to Android and iOS so it will be able to reach more people. The downside to using Xamarin is that there are slight configurations that need to be done when building for Android and iOS while React Native requires needs no additional configuration.

React Native will be the technology that will be used to build the application due to the ease of learning and the huge online support. Another very important reason is the popularity of React Native in the industry, and learning React Native will help me in the future.

3.3 HOSTING SERVICES

Having the appropriate server is an important decision since several applications need to be hosted. The goal for the servers is that:

- it can host a MySQL database.
- it can host PHP scripts that allow me to access the database.
- it allows me to host a Java web scraping application – needs to run periodically.

3.3.1 POSEIDON SERVER

Poseidon is the server owned and maintained by The University of Salford and making an account was as simple as sending an email. Setting up the database was straightforward, and the connection was tested using MySQL Workbench. However, there were firewall issues when it came to the PHP scripts connecting to the database and calling them from the application. To confirm that it was a Poseidon issue, the same process was repeated on an experimental server called 000webhost and the PHP scripts were accessible and data from the database was being displayed. Therefore, the only requirement Poseidon was able to handle was the first and failed on the second.

3.3.2 INTERNAL SERVER & DOCKER

A different approach was taken by attempting to host the server locally by using a spare laptop that is running ubuntu and using docker images to get the applications. Doker Portainer was used to get the SQL Database and was hosted on the server. MySQL Workbench was used to test the connection which worked. However, it was difficult to find docker images that allowed me to host an application that can be run periodically. The server was also not set up optimally and it was crashing randomly so a lot of time was spent on fixing the server and making sure it was running smoothly.

3.3.3 AMAZON WEB SERVICES (AWS)

AWS is a powerful set of tools that offer lots of different hosting services. Setting up the database was easy and straightforward. It was tested using MySQL Workbench. AWS also offers services to host the PHP scrips by using EC2 and S3 buckets. The PHP scripts were run on the URL and data was displayed. This shows that the PHP scripts are working as they should. Finally, AWS works nicely with Kubernetes and offers services that allow the application to run automated at certain times. This is perfect for the java web scraping application.

3.3.4 SUMMARY

Three different services have been researched to determine which one would be best suited to fill the requirements of the project. The Poseidon server had an easy database set up but had firewall issues that stopped me from accessing my database through the application. The internal server kept on crashing and fixing them created an extra layer of complexity. It also did not provide a docker image that allowed me to periodically run the java application. AWS offered numerous services that covered all my requirements. It had the database, E2 for the PHP scripts, and Kubernetes integration for the java application. Therefore, AWS will be used for the project to host the application.

4 REQUIREMENTS SPECIFICATION AND DESIGN

This chapter will discuss the requirements that are needed for the different sprints, also known as increments. It will look at the approach that was adopted to design the application and the justification for taking that approach. This chapter will be broken up into three sections, each representing an increment, and will consist of:

- The requirements for the increment.
- The use cases for the requirements.
- The User Interface (UI) designs that will be needed to achieve the use cases.

Having a clear and detailed requirement specification is vital to the success of any project, as stated by Daniel M. Berry in his book Ambiguity in Requirements Specification: “Unintended ambiguity in specification admits multiple interpretations of the underlying document (Berry & Kamsties, 2004)”. This could lead to an unclear implementation in the later stages as there might be an enormous time gap between the requirement specification and implementation phase. Therefore, this is a very important step and needs careful consideration to create a detailed road map that will aid in the development of the application.

Requirements are typically divided into two categories: functional & non-functional requirements. Functional requirements are the features and behaviours that a system needs to have, i.e., what the system should do (Glinz, 2005). Non-functional requirements are focused more on how well the application performs such as usability, scalability, and data integrity (Glinz, 2005). This chapter will be covering both requirements.

Use Case Diagrams are used to describe the requirements and desired operations of an application. They are tailored to capturing those requirements by creating scenarios where the user (also known as the actor) and the application interact with each other to perform the desired functions. It is a very famous design concept that many big software design tools provide, such as “Rational Software Architect from IBM and Visual Studio from Microsoft (Grechanik, et al., 2007)”.

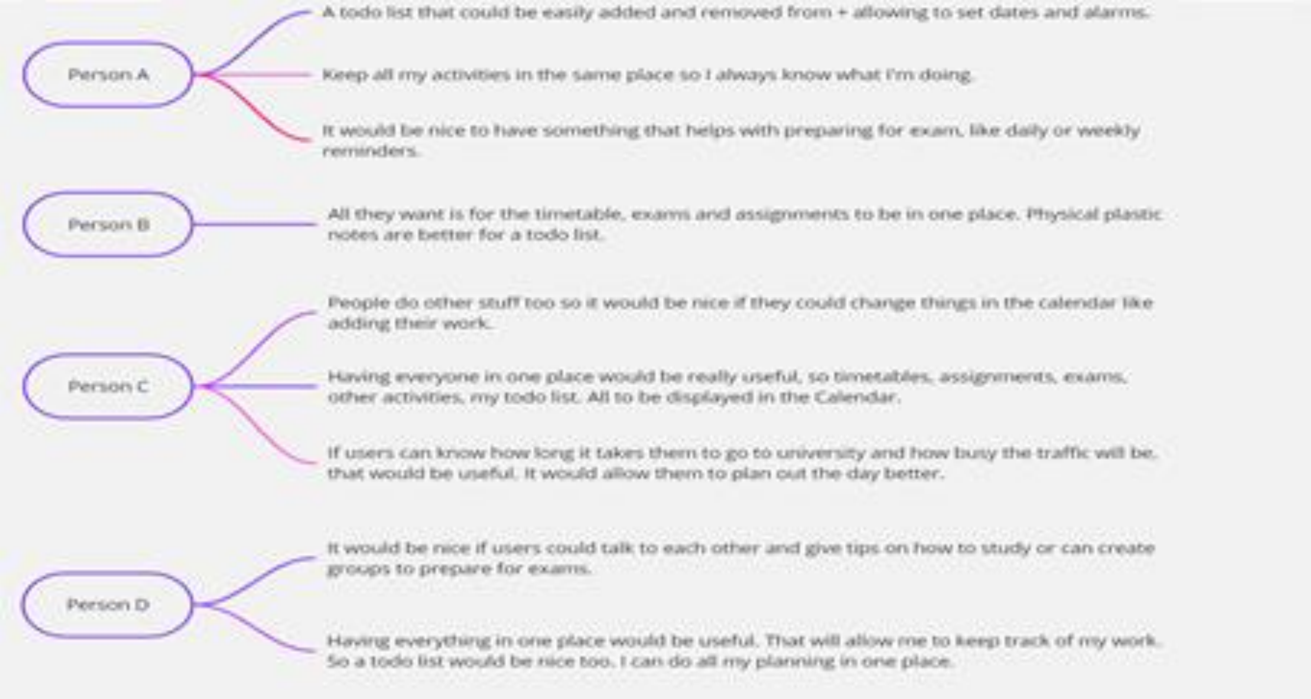
Creating screen prototypes is useful before starting the development of the application as “empirical literature suggests that requirements and design decisions exert a tremendous impact on software quality (Curtis, et al., 1988)”. Creating designs provides a concrete idea as to what the application will look like. Through several iterations of the design phase. A well-thought-out design will be ready to use in the development phase.

4.1 FOCUS GROUP

A focus group is typically a group of people discussing a specific issue or topic. These focus groups are very useful in getting different opinions and perspectives regarding topics especially when decision-making is involved (Kitzinger, 1994).

Therefore, a focus group was held at the beginning of the project to listen to students and what they would like to get from the application. Based on their feedback, a requirement specification was written up. Four students took part in the focus group and each one was provided with a consent form to sign and an information sheet for them to read. These can be found in the Appendix.

A software called Miro was used to write down the discussed points and Discord was used to share the screen. Participants shared their experience on how they managed their time during the first year and what could have helped with their time management skills. This opened the room for discussing different possible features that the application could have. Figure 4.1 shows the main features that each participant wanted from the application.



Person A

A todo list that could be easily added and removed from + allowing to set dates and alarms.

Keep all my activities in the same place so I always know what I'm doing.

It would be nice to have something that helps with preparing for exam, like daily or weekly reminders.

Person B

All they want is for the timetable, exams and assignments to be in one place. Physical plastic notes are better for a todo list.

Person C

People do other stuff too so it would be nice if they could change things in the calendar like adding their work.

Having everyone in one place would be really useful, so timetables, assignments, exams, other activities, my todo list. All to be displayed in the Calendar.

If users can know how long it takes them to go to university and how busy the traffic will be, that would be useful. It would allow them to plan out the day better.

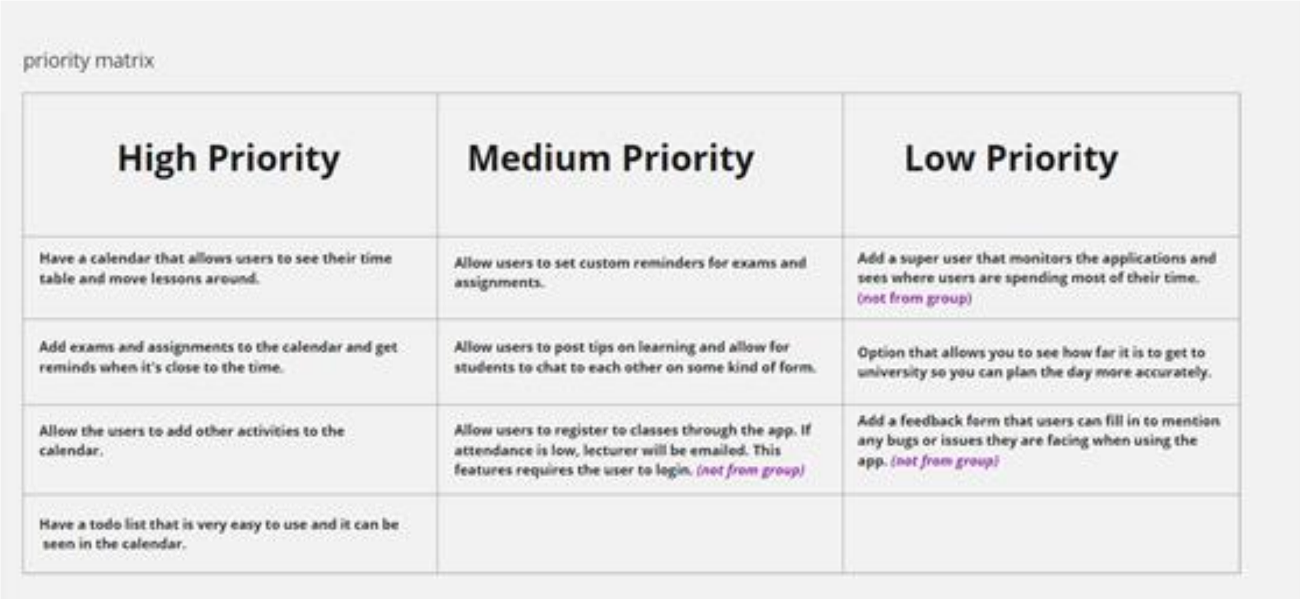
Person D

It would be nice if users could talk to each other and give tips on how to study or can create groups to prepare for exams.

Having everything in one place would be useful. That will allow me to keep track of my work. So a todo list would be nice too. I can do all my planning in one place.

Figure 4 1 - The main features discussed during the focus group

At the end of the focus group, a matrix was shown to the participants that allowed them to categorize their features, from lowest to highest priority. The result of that exercise can be seen in figure 4.2. These features are the features that will be used for this project and will be split into different increments.



priority matrix

High Priority	Medium Priority	Low Priority
Have a calendar that allows users to see their time table and move lessons around.	Allow users to set custom reminders for exams and assignments.	Add a super user that monitors the applications and sees where users are spending most of their time. *(not from group)*
Add exams and assignments to the calendar and get reminds when it's close to the time.	Allow users to post tips on learning and allow for students to chat to each other on some kind of form.	Option that allows you to see how far it is to get to university so you can plan the day more accurately.
Allow the users to add other activities to the calendar.	Allow users to register to classes through the app. If attendance is low, lecturer will be emailed. This features requires the user to login. *(not from group)*	Add a feedback form that users can fill in to mention any bugs or issues they are facing when using the app. *(not from group)*
Have a todo list that is very easy to use and it can be seen in the calendar.		

Figure 4 2 - The priority Matrix from the discussed features

4.2 INCREMENT 1

4.2.1 REQUIREMENTS AND GOALS

4.2.1.1 LOGIN AND REGISTER – USERS SHOULD BE ABLE TO LOGIN IF THEY HAVE AN ACCOUNT OR REGISTER IF THEY DO NOT

The first thing the user should see when they use the application is an option to sign in and an option to register. This is vital as this is a data-driven project and the application will be different for each user. Therefore, the application needs to know which user is logged in to show the correct information. The specifications that are needed for this requirement are:

- A user should be able to type in their email and password to log in.
- A user should have the option to register if they do not have an account.
- The email that is used to login or register needs to be a university email.
- The information that is needed from the user when registering is:
    - The university they go to.
    - The course they take.
    - The year they are in.
    - The group they are put in.

4.2.1.2 TIMETABLE – USERS SHOULD BE ABLE TO VIEW THEIR TIMETABLE AND CUSTOMISE IT

One of the main objectives for this increment is for the user to view their timetable and have the power to personalise it. This is a huge benefit to students as they can easily move between classes, granted there is space, which minimises the possibility of missed classes. For the timetable to work and represent the correct information, certain information is needed from the user, these are:

- The university the user goes to.
- The course they take.
- The year they are in.
- The group they have been allocated to.

When it comes to editing the timetable, restrictions will be in place to ensure the changes are possible and that there is a fair distribution of students per lesson. These restrictions are:

- Students cannot place two lessons on the same day and time.
- Lessons cannot overlap.
- Students can only move lessons to a day and time where that specific lesson is taking place.

4.2.1.3 CALENDAR – USERS SHOULD BE ABLE TO VIEW THEIR EVENTS THROUGH A CALENDAR

A very powerful feature for the user to have is a calendar with a monthly, weekly, and daily view which allows them to see events that they have, such as assignments, exams, and other extracurricular activities. This gives the user a lot more control over their time and an awareness of how busy they are on any given day. The major tasks needed to achieve this requirement are:

- The calendar needs to differentiate between the different tasks by representing them differently in the marked dots.
- The user should be able to move freely between the different views (monthly, weekly, daily) without any difficulty.
- The user should be able to view, edit, add and delete tasks. Those changes need to be reflected on the calendar.

4.2.2 USE CASE DIAGRAM

As shown in figure 4.3, there is only 1 primary actor, which is the user, and that is represented on the left side of the Use Case Diagram. In this scenario, there are no secondary actors, and this can be shown by the absence of an actor on the right side. The rectangle shape in the centre represents the application, and the oval shapes inside it show the actions that the application can perform. The lines represent the relationship between the actor and the application. The use case diagram represents the features of the first increment and links to the requirements specification of the first increment. The Use Case Diagram was done by a tool called Creately (Creately, n.d.).

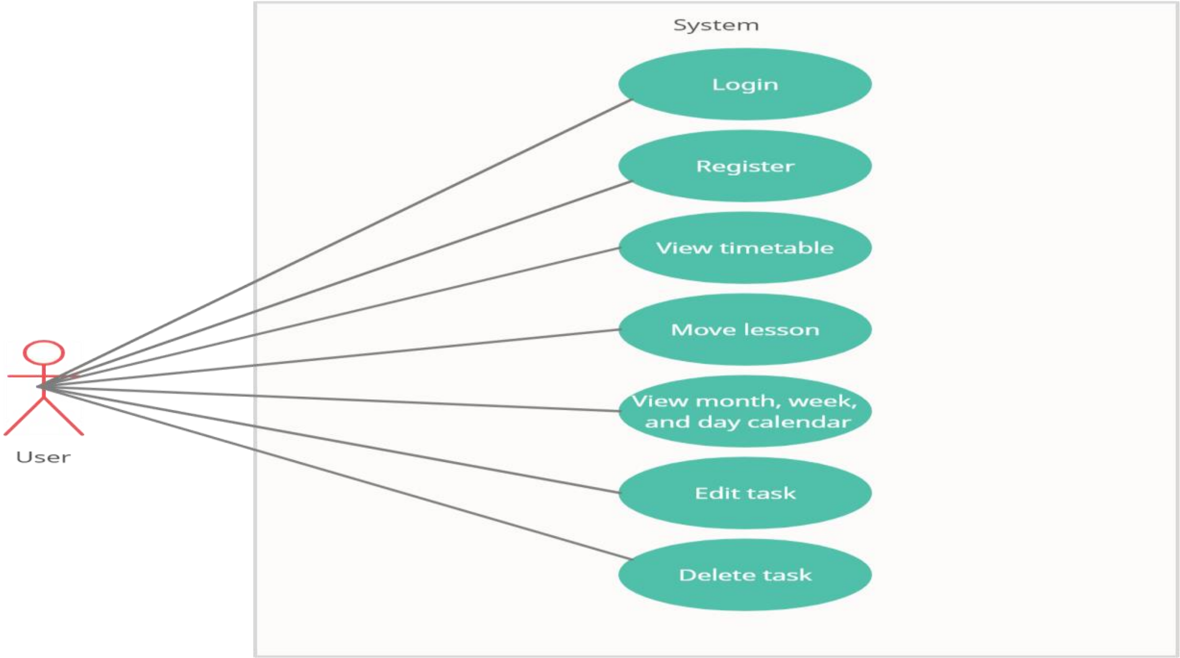


Figure 4 3 - Sprint 1 Use Case Diagram

4.2.3 PROTOTYPE USER INTERFACE DESIGNS

Figures 4.4 and 4.5 show the initial sketch designs for the login and register screens. Figure 4.6 shows the sketch design for the timetable screen. Figures 4.7 and 4.8 show the sketch designs for calendar screens. These designs were done to represent the requirement specifications and use case diagram of the first increment.

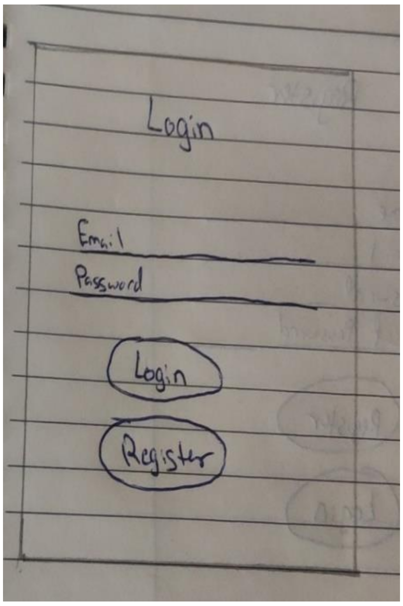


Figure 4 4 - Login Sketch Design

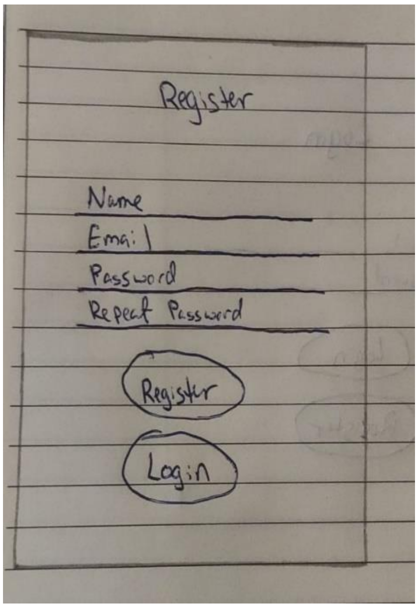


Figure 4 5 - Register Sketch Design

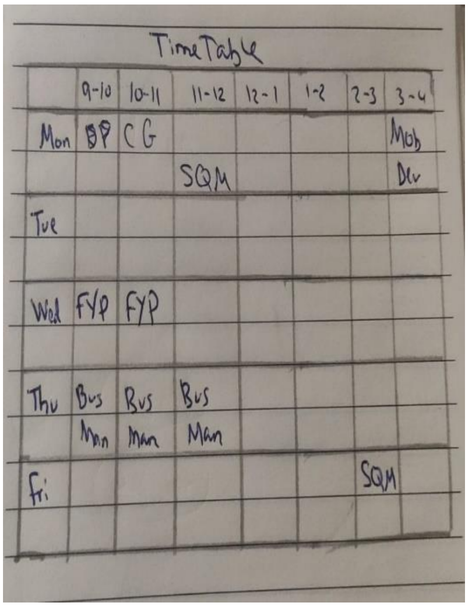
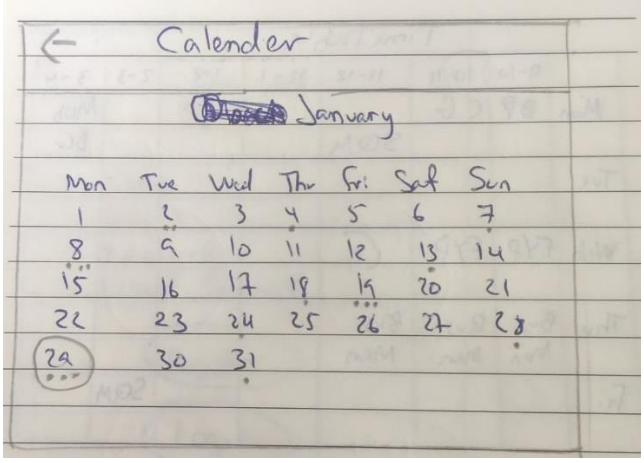
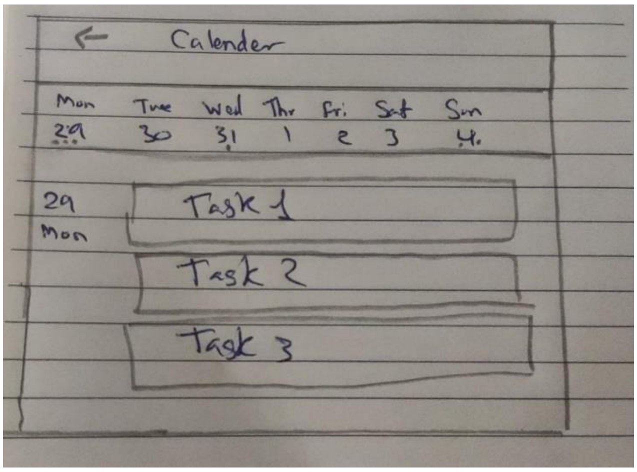


Figure 4 6 - Timetable Sketch Design



Mon	Tue	Wed	Thr	Fri	Sat	Sun
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

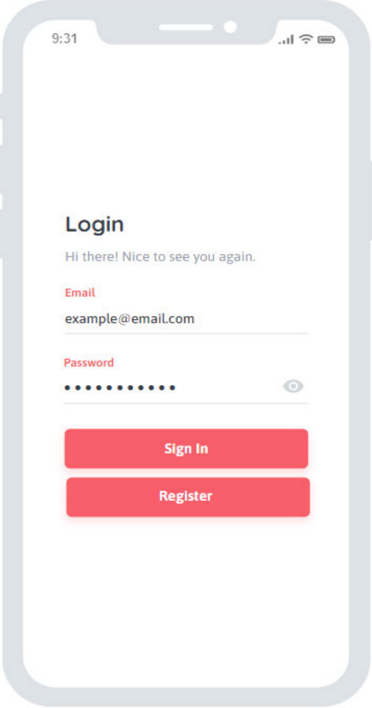
Figure 4 7 - Calendar Month Sketch Design



Mon	Tue	Wed	Thr	Fri	Sat	Sun
29	30	31	1	2	3	4.

Figure 4 8 - Calendar Day Sketch Design

These sketch designs were later enhanced using a software tool called Moqups (Moqups, 2021) to represent the design of the application more accurately. Figures 4.9 and 4.10 show the designs of the login and register screens. Figure 4.11 shows the design of the timetable screen. Figures 4.12 and 4.13 show the design of the calendar screen.



9:31

Login

Hi there! Nice to see you again.

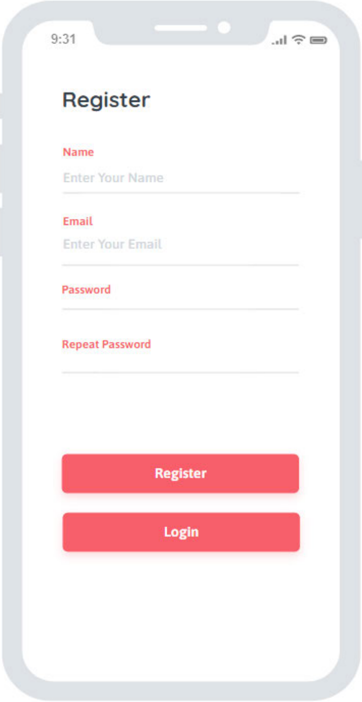
Email

example@email.com

Password

•••••••••••

Figure 4 9 - Login Design



9:31

Register

Name

Enter Your Name

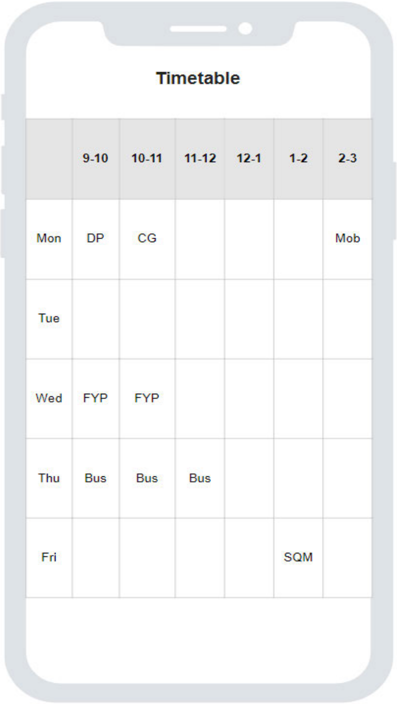
Email

Enter Your Email

Password

Repeat Password

Figure 4 10 - Register Design



Timetable

	9-10	10-11	11-12	12-1	1-2	2-3
Mon	DP	CG				Mob
Tue						
Wed	FYP	FYP				
Thu	Bus	Bus	Bus			
Fri					SQM	

Figure 4 11 - Timetable Design

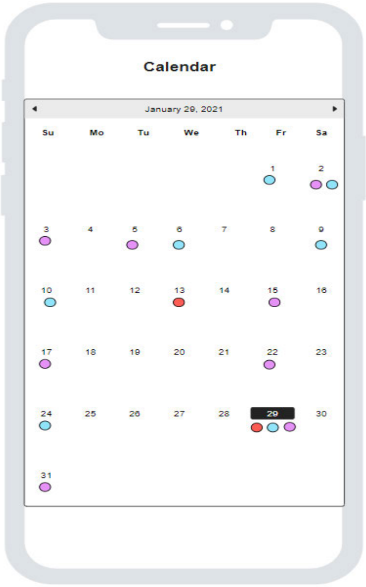


Figure 4 12 - Calendar Month Design

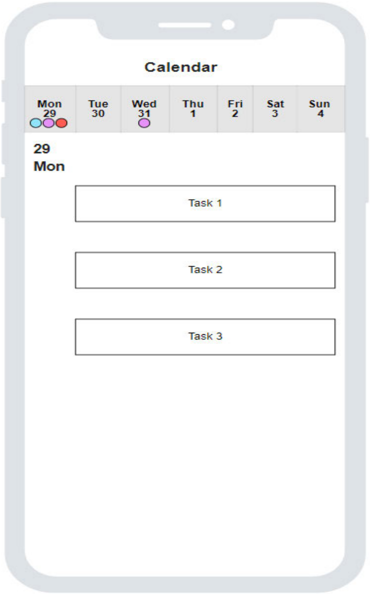


Figure 4 13 - Calendar Day Design

4.3 INCREMENT 2

4.3.1 REQUIREMENTS AND GOALS

4.3.1.1 PROFILE – USERS SHOULD BE ABLE TO VIEW THEIR PROFILE AND EDIT THEIR PERSONAL INFORMATION

There needs to be a way for the user to view and edit their personal information. This is very important as the data needs to be updated when the users' situation changes, such as moving to a different year or changing groups. Therefore, the specifications for this requirement will be:

- ➢ Detailed information about the user should be displayed on the profile screen.
- ➢ The user should be able to upload and delete their profile picture.
- ➢ The user should be able to update their information and the changes should be reflected on the timetable.

4.3.1.2 DASHBOARD – USERS SHOULD BE ABLE TO SEE A SUMMARY OF ALL THE IMPORTANT POINTS

Having a dashboard is very useful as it allows the user to quickly summarise what needs to happen and give them an overview of the week. Therefore, it should not overload the user with information but instead give them just enough to make them aware of how much time they have during the week. The specifications for this requirement are:

- ➢ A summary should be given of the assignments, exams, and activities that the user needs to do for the week.
- ➢ The overall attendance of the student should be shown.
- ➢ The user should be able to see if they have a class at the current time and have the option to register for that class.

4.3.1.3 TO-DO LIST – USERS SHOULD BE ABLE TO ADD TASKS AND CHECK THEM OFF AS COMPLETE

A to-do list can benefit the user as it helps them plan out their time and make sure that the tasks that need to be done are taken care of. It is also shown to reduce stress and make the person more relaxed. This will help them with their work since students usually have assignments and other activities to take care of. The specifications for this requirement are:

- The user should be able to add, edit, and delete tasks.
- The tasks need to be differentiated between the different types (exam, assignment, and other activities).
- Tasks can be filtered based on type and date.
- Tasks should be able to be marked as complete.

4.3.2 USE CASE DIAGRAM

Figure 4.14 shows the use case diagram that represents the second increment and shows the requirements mentioned above. The previous use case diagram will not be added here due to space constraints and to maintain simplicity. The primary actor is represented on the left side and the application is the rectangle in the centre. The oval shapes are the actions that are done by the application and the line represents the relationship between the user and the application.

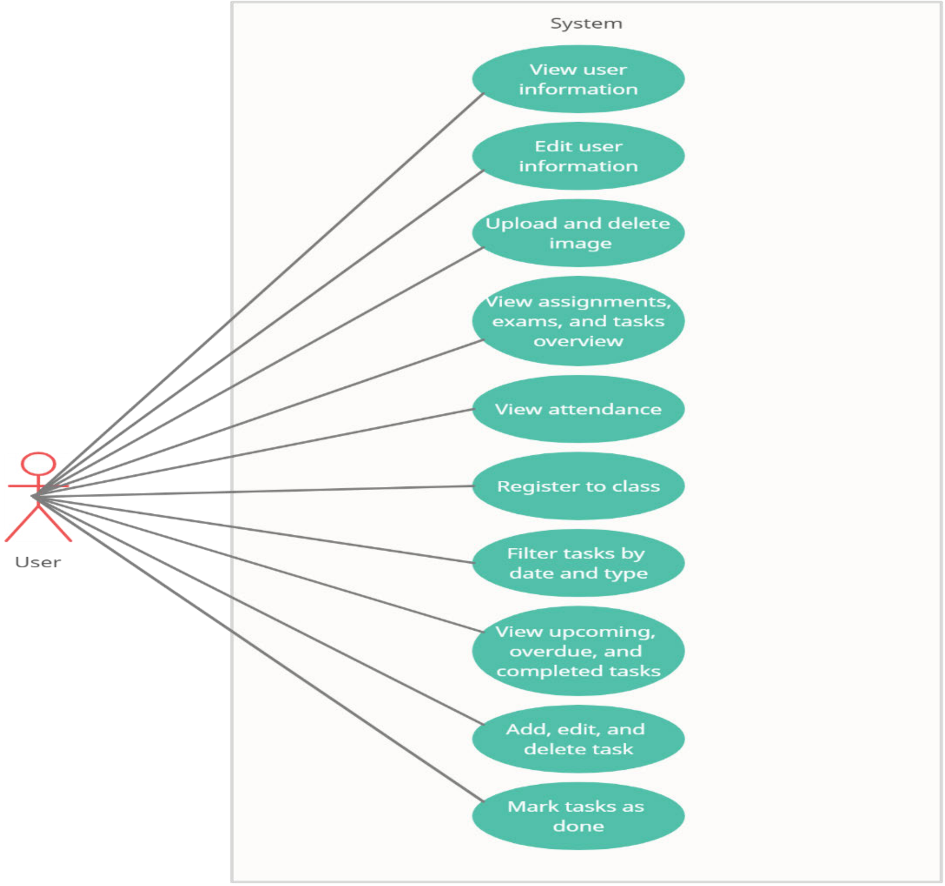
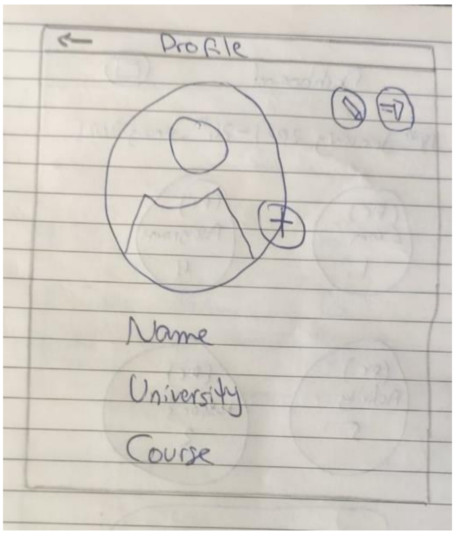
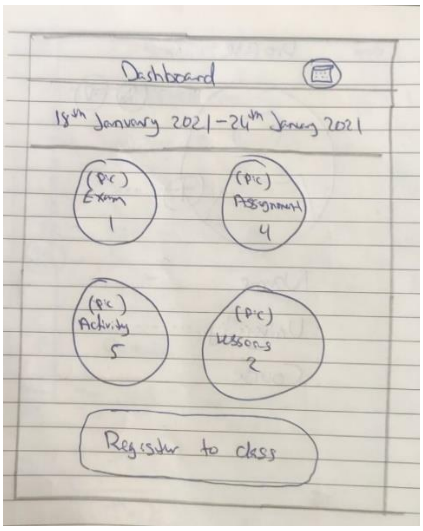

System
View user information
Edit user information
Upload and delete image
View assignments, exams, and tasks overview
View attendance
Register to class
Filter tasks by date and type
View upcoming, overdue, and completed tasks
Add, edit, and delete task
Mark tasks as done
User

Figure 4 14 - Sprint 2 Use Case Diagram

4.3.3 PROTOTYPE USER INTERFACE DESIGNS

Figure 4.15 shows the initial sketch designs for the profile screen. Figure 4.16 shows the sketch design for the dashboard screen. Figure 4.17 shows the sketch designs for to-do list screens. These designs were done to represent the requirement specifications and use case diagram of the second increment.





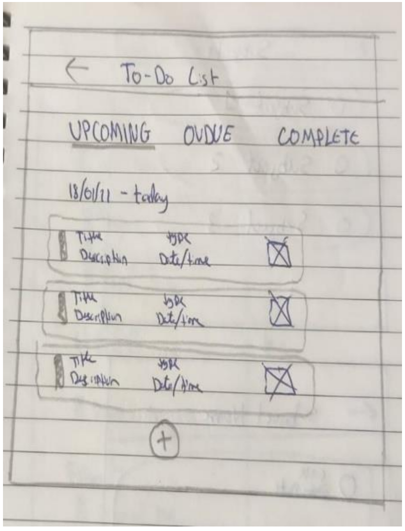
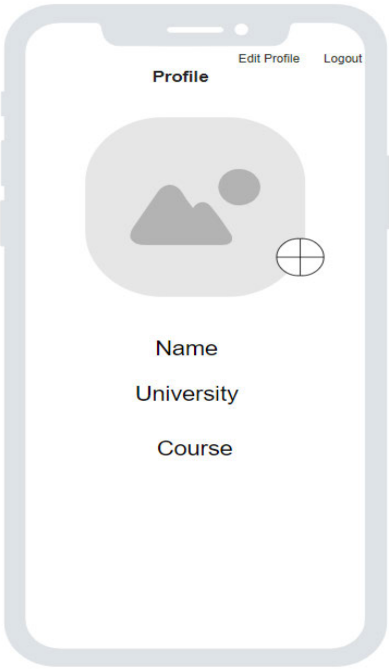


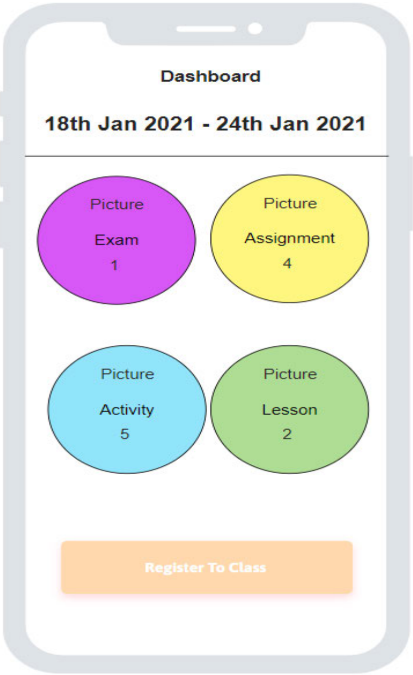
Figure 4 15 - Profile Sketch Design

Figure 4 16 - Dashboard Sketch Design

Figure 4 17 - To-Do List Sketch Design

These sketch designs were later enhanced using a software tool called Moqups (Moqups, 2021) to represent the design of the application more accurately. Figure 4.18 shows the design of the profile screen. Figure 4.19 shows the design of the dashboard screen. Figure 4.20 shows the design of the to-do list screen.





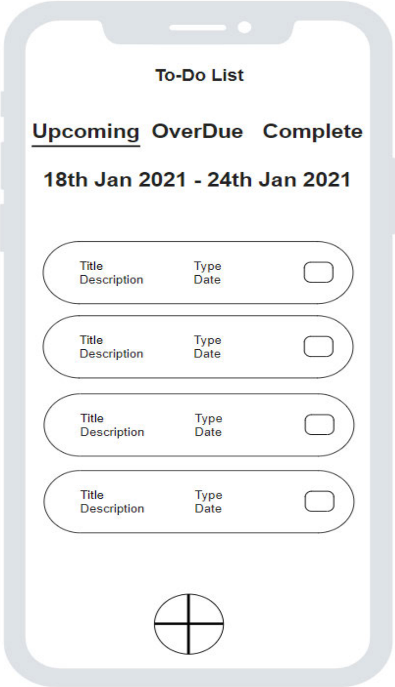


Figure 4 18 - Profile Design

Figure 4 19 - Dashboard Design

Figure 4 20 - To-Do List Design

4.4 INCREMENT 3

4.4.1 REQUIREMENTS AND GOALS

4.4.1.1 SUBJECT FEED – USERS SHOULD BE ABLE TO POST AND COMMENT

A very useful feature for the users is to have a way for them to communicate and help each other with their subjects. Therefore, a social feed where users can post on their subjects and comment on other people's posts would be very useful to have. The specification requirement for this will be:

- The user can pick what subject they want to view the posts of.
- The user should be able to make a post to any of the subjects they take.
- The user should be able to comment on any post on any subject that they take.
- The image and name of the user and the time of the post should be included in the posts and comments.
- The number of comments should be reflected in the post.

4.4.1.2 SETTINGS – USERS SHOULD BE ABLE TO CONFIGURE THE APPLICATION TO SUIT THEIR NEEDS

Accessibility is very important when building an application, and the settings will address this. The settings page is a way for users to customise the application so it can be easier for them to use. This should address many important points such as making it easier for older people and people with vision impairment. It also gives users the ability to report any issues they come across while using the application. The specification requirement for this will be:

- The user should be able to pick what day the calendar should start on.
- The user should be able to increase or decrease the font size of the text to help people with vision impairment.
- The theme should be changeable to help people with colour blindness.
- The user should be able to report issues.

4.4.2 USE CASE DIAGRAM

Figure 4.21 shows the use case diagram that represents the third increment and shows the requirements that were mentioned above. The previous two use case diagram will be omitted for the sake of simplicity and due to space constraints. This shows the interaction between the user, which is the actor on the left side, and the application which is the rectangle shape in the centre of the image.

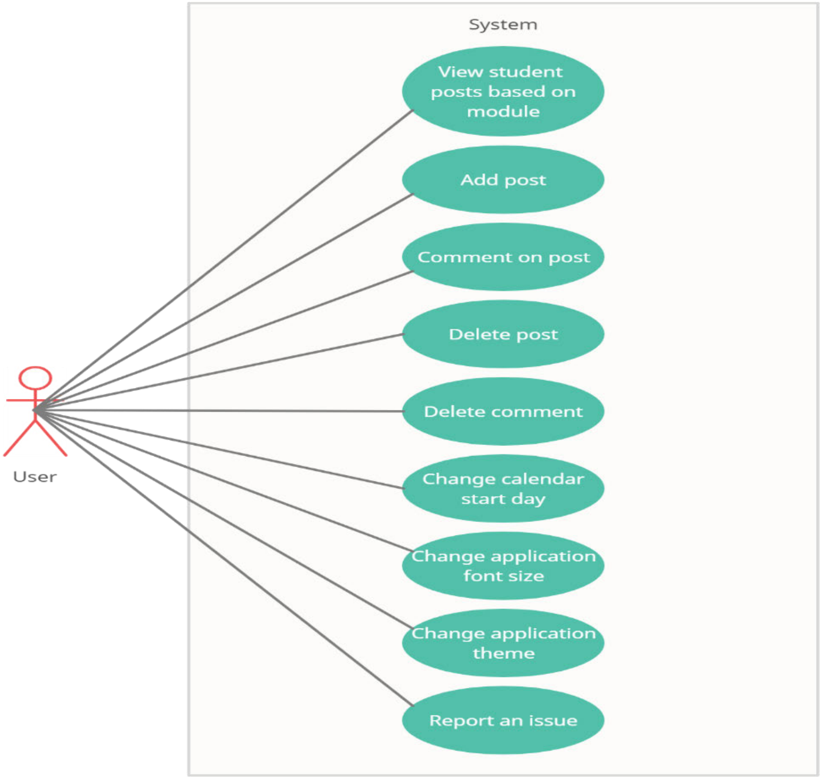


Figure 4 21 - Sprint 3 Use Case Diagram

4.4.3 PROTOTYPE USER INTERFACE DESIGNS

Figures 4.22 and 4.23 show the initial sketch designs for the subject feed screens. Figure 4.24 shows the sketch design for the settings screen. These designs were done to represent the requirement specifications and use case diagram of the third increment.

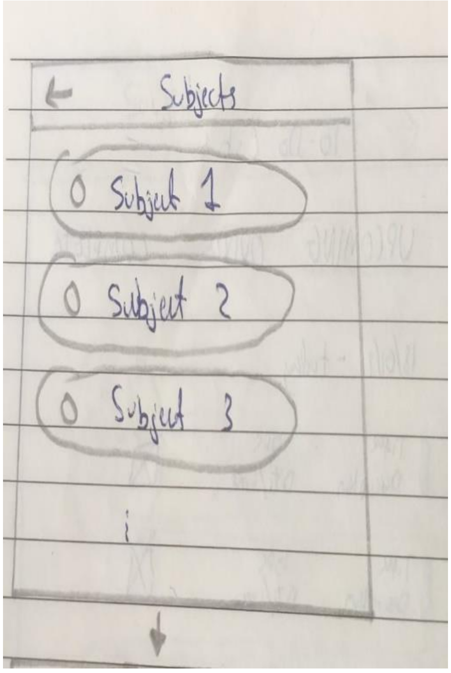


Figure 4 22 - Subject List Sketch Design

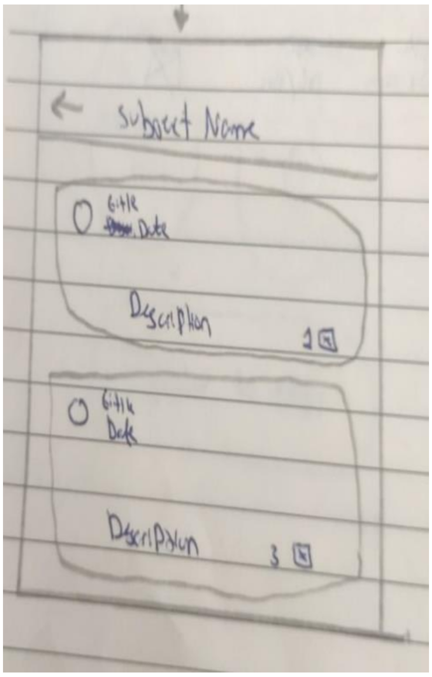


Figure 4 23 - Subject Feed Sketch Design

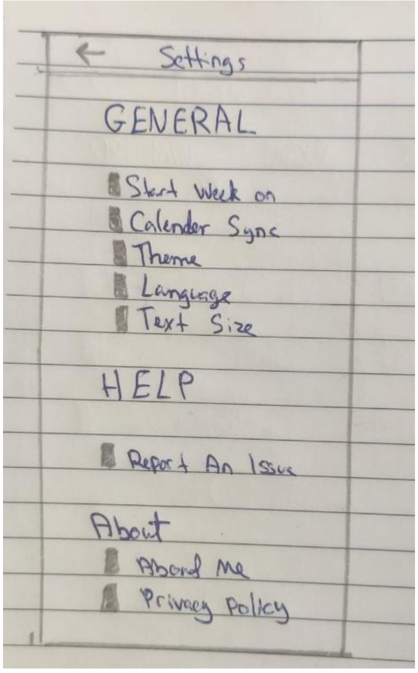


Figure 4 24 - Settings Sketch Design

These sketch designs were later enhanced using a software tool called Moqups (Moqups, 2021) to represent the design of the application more accurately. Figures 4.25 and 4.26 show the designs of the subject feed screen. Figure 4.27 shows the design of the settings screen.

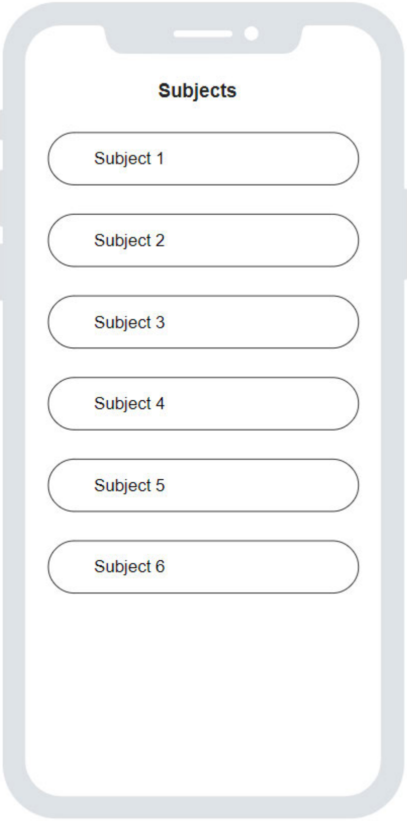


Figure 4 25 - Subject List Design

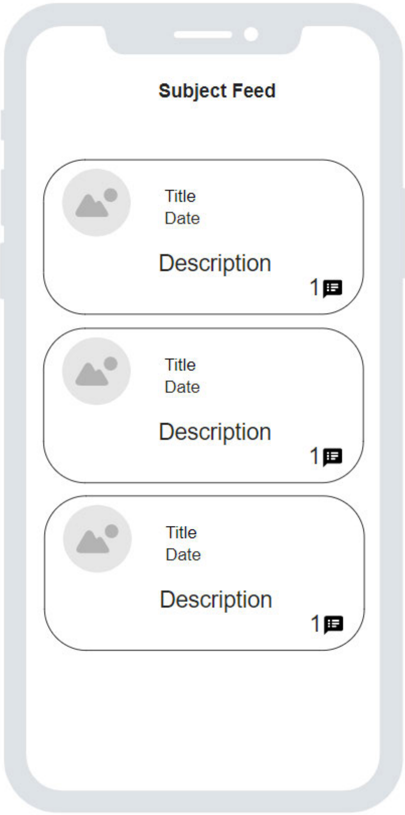


Figure 4 26 - Subject Feed Design

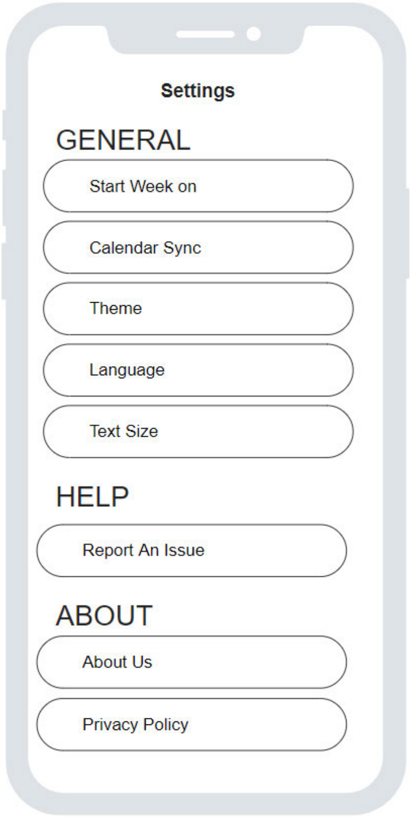


Figure 4 27 - Settings Design

4.5 SUMMARY

This chapter covered the specification requirement that was needed to complete this project. It covered all the main features that were mentioned in the Gantt Chart and those were broken down into use cases and designs that aids in the development process. Since these requirements were broken up into increments, it was possible to get feedback which aided in improving the requirements and making changes to better improve the user experience.

5 DEVELOPMENT AND IMPLEMENTATION

This chapter will cover the development and implementation of the requirement specification that was mentioned in the previous chapter. It will be split into three increments. Each increment will look at the challenges faced and how it was overcome. It will also look at the libraries that were used to implement the functionalities and their justification. A comparison will be made at the end of each increment comparing the sketch designs that were done in the previous chapter to the Minimal Viable Product (MVP) that was produced.

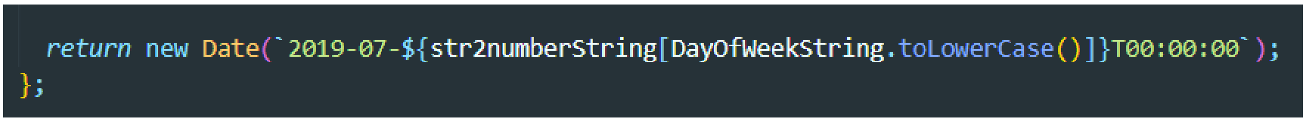
The development and implementation phase consists of writing the program and building the requirements while keeping it as close as possible to the specification. This phase is important when it comes to the success of the project as this determines what the end-user will see and experience. If this phase is poorly managed, then it is very likely that the project as a whole will fail. Therefore, making sure that this phase adheres to the requirements and implements them accordingly is essential.

5.1 INCREMENT 1

5.1.1 DEVELOPMENT CHALLENGES AND SOLUTIONS

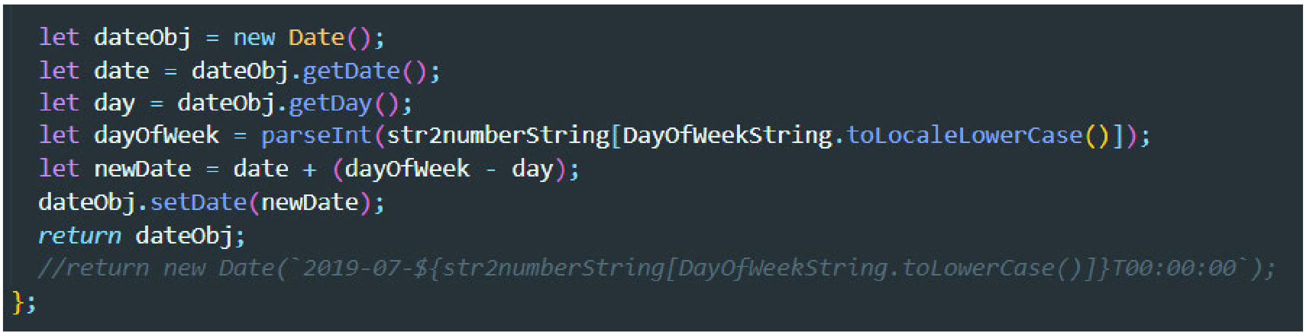
5.1.1.1 TIMETABLE LIBRARY HAVING HARD-CODED VALUES

The developers of the timetable library stopped maintaining it in 2019. This meant that the library was not up to date and bugs that are present were not being worked on. The biggest issue was that one of the main functions that dealt with the dates was hardcoded to July 2019. Since the code was not dynamic, it caused problems whenever I try to display an event in the library with a specific date as it would always position itself on a wrong day. Figure 5.1 shows the code from the library. To fix the issue, I edited the code and made it dynamic so that it will work for any date that is passed in and will display it on the correct day. Figure 5.2 shows my version of the library function and what I did to solve the issue.



```
  return new Date(`2019-07-${str2numberString[DayOfWeekString.toLowerCase()]}T00:00:00`);
};
```

Figure 5 1 - Timetable Library Issue

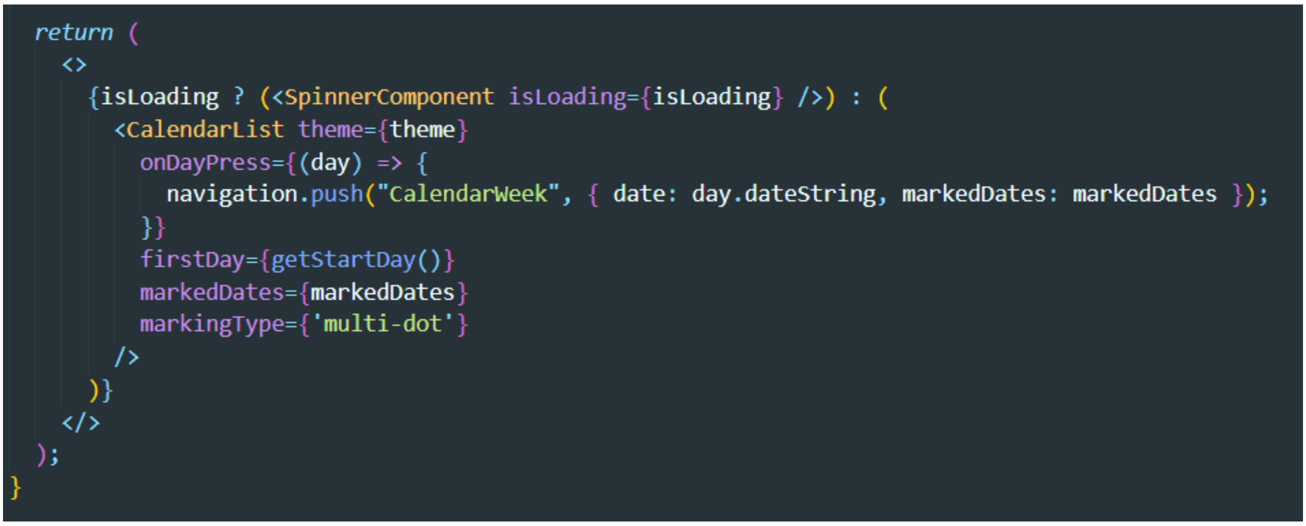


```
  let dateObj = new Date();
  let date = dateObj.getDate();
  let day = dateObj.getDay();
  let dayOfWeek = parseInt(str2numberString[DayOfWeekString.toLocaleLowerCase()]);
  let newDate = date + (dayOfWeek - day);
  dateObj.setDate(newDate);
  return dateObj;
  //return new Date(`2019-07-${str2numberString[DayOfWeekString.toLowerCase()]}T00:00:00`);
};
```

Figure 5 2 - Timetable Library Issue Solution

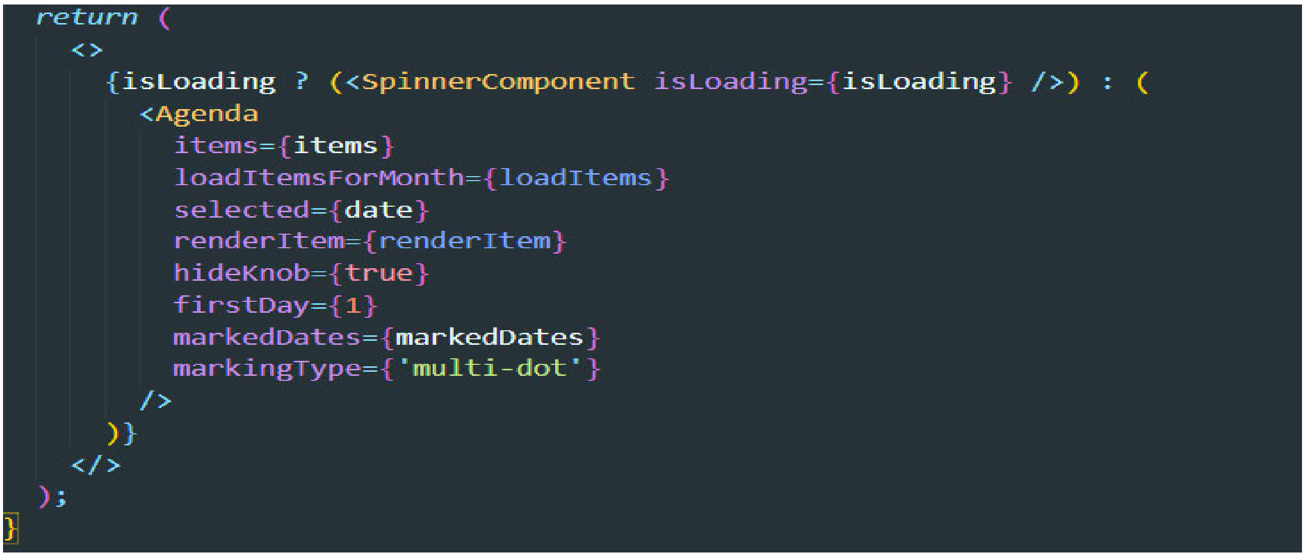
5.1.1.2 NOT TAKING FULL ADVANTAGE OF THE AGENDA COMPONENT IN THE CALENDAR LIBRARY

The calendar library is very powerful and offers a wide range of features. The two most important ones being the Calendar Component and the Agenda Component. The Calendar Component displays a calendar view that can be customised and edited in various ways. The Agenda Component has all the features of the Calendar Component alongside a detailed view of the days. Therefore, ideally, only the Agenda Component would be needed to achieve one of the requirements of the first increment. However, the Agenda Component requires a date object to be passed in and that day will be displayed as the default starting view. This is not the planned design which was starting with the month view and allowing the user to pick the day they want to view. The solution was to use both components rather than just the Agenda. Figure 5.3 shows the code that represents the Calendar Component, and figure 5.4 shows the code that represents the Agenda component. This implementation will slightly affect the performance due to using two heavy components rather than one, but it meets the requirements that were specified in the previous chapter.



```
  return (
    <>
      {isLoading ? (<SpinnerComponent isLoading={isLoading} />) : (
        <CalendarList theme={theme}
          onDayPress={(day) => {
            navigation.push("CalendarWeek", { date: day.dateString, markedDates: markedDates });
          }}
          firstDay={getStartDay()}
          markedDates={markedDates}
          markingType={'multi-dot'}
        />
      )}
    </>
  );
}
```

Figure 5 3 - Calendar Component



```
  return (
    <>
      {isLoading ? (<SpinnerComponent isLoading={isLoading} />) : (
        <Agenda
          items={items}
          loadItemsForMonth={loadItems}
          selected={date}
          renderItem={renderItem}
          hideKnob={true}
          firstDay={1}
          markedDates={markedDates}
          markingType={'multi-dot'}
        />
      )}
    </>
  );
}
```

Figure 5 4 - Agenda Component

5.1.2 TOOLS AND LIBRARIES USED

Two libraries were used for the development of this increment. These libraries are:

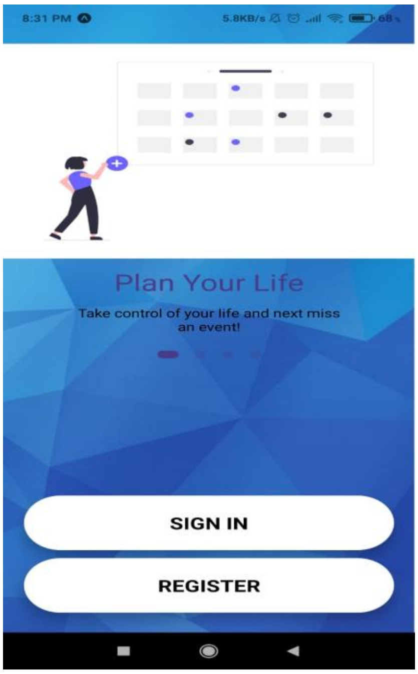
- react-native-timetable (https://github.com/gomjellie/react-native-timetable)
- react-native-calendars (https://github.com/wix/react-native-calendars)

When it came to choosing a library that will help with the implementation of the timetable, the react-native-timetable library was the only one that was available to use. This library had many downsides such as it not being maintained since 2019 and have broken code. However, since this was the only option available, there was nothing else that could be done but to use it and fix the issues that it had. Doing everything from scratch without any library would have taken more time and effort (gomjellie, 2019).

On the other hand, there were a plethora of libraries that offered solid implementations of a calendar. However, the WIX calendar library was the one that was chosen for this project. The reason for that was due to its versatility in the features that it offered and the ability to customise it. The library provides everything that is needed to achieve the requirements for this increment. It was also the most popular library that was used when it came to a calendar implementation and it is still being maintained, at the time of writing, by a group of developers (Wix.com, 2021).

5.1.3 MINIMAL VIABLE PRODUCT

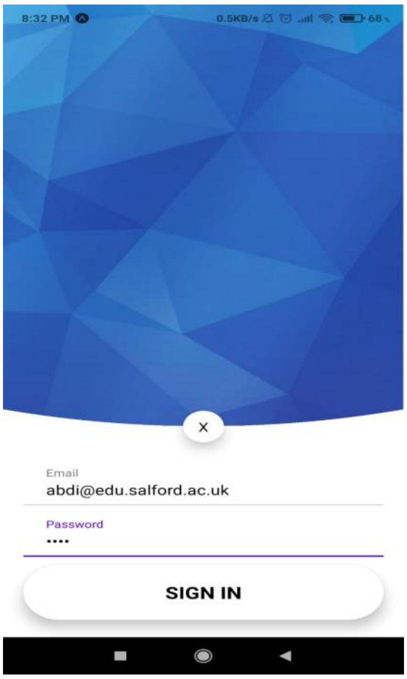
Figures 5.5, 5.6, and 5.7 display the outcome of the implementation of the login and register. The initial designs were simple and did not have any complex features. However, the implementation was a lot more complex and advanced as it used smooth animation and transitions to display the login screen. This is great as it displays the richness of the platform in creating complex UI.



Plan Your Life

Take control of your life and next miss an event!

Figure 5 5 - Login & Register Home Screen



Email
abdi@edu.salford.ac.uk

Password
••••

Figure 5 6 - Login Screen

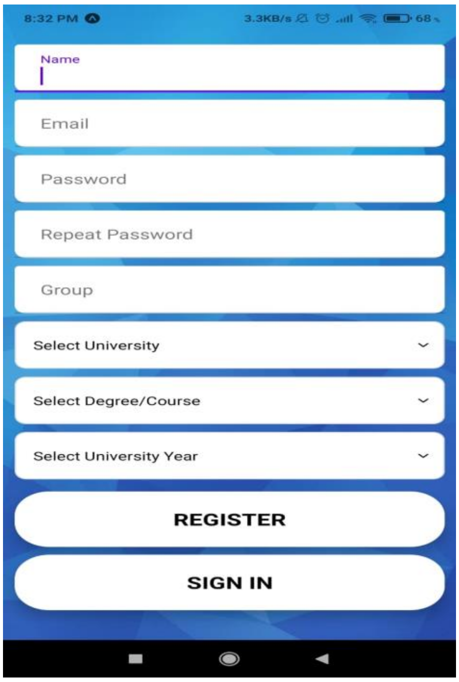
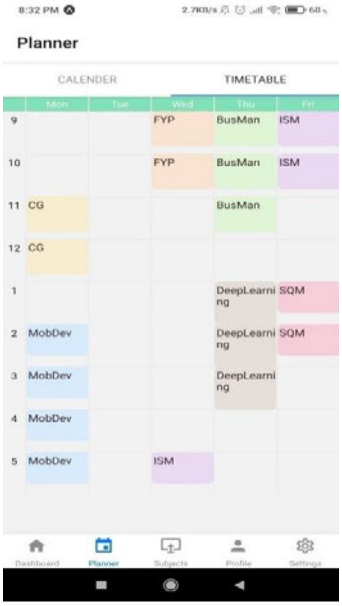


Figure 5 7 - Register Screen

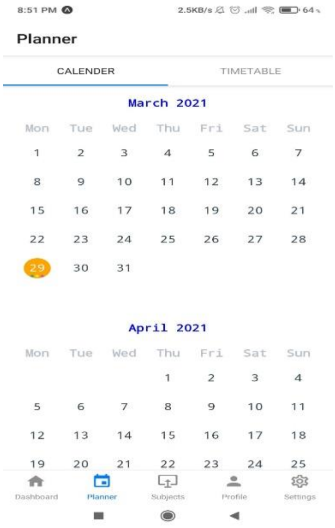
Figure 5.8 displays the outcome of the implementation of the timetable. The initial design that was made for this requirement did not look like the product that was made. This is due to the constraints of the timetable library as it did not give the ability to customise the UI. However, I do believe a good job was done in trying to present the UI in an easy and intuitive way.



	Mon	Tue	Wed	Thu	Fri
9			FYP	BusMan	ISM
10			FYP	BusMan	ISM
11	CG			BusMan	
12	CG				
1				DeepLearning	SQM
2	MobDev			DeepLearning	SQM
3	MobDev			DeepLearning	
4	MobDev				
5	MobDev		ISM		

Figure 5 8 - Timetable Screen

Figures 5.9 and 5.10 display the outcome of the implementation of the calendar. The initial design for this requirement was very similar to the implementation that was made. This is a great sign that the development phase was able to adhere to the design and follow it very closely.

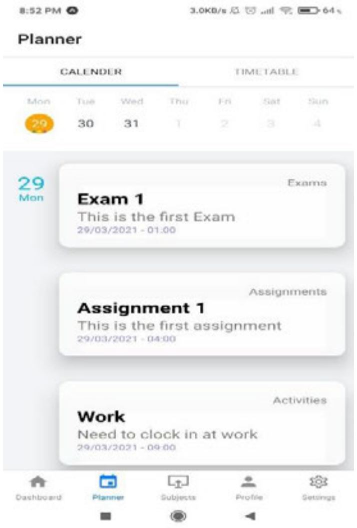


March 2021

Mon	Tue	Wed	Thu	Fri	Sat	Sun
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

April 2021

Mon	Tue	Wed	Thu	Fri	Sat	Sun
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25



Mon	Tue	Wed	Thu	Fri	Sat	Sun
29	30	31	1	2	3	4

29
Mon

Exams
Exam 1
This is the first Exam
29/03/2021 - 01:00

Assignments
Assignment 1
This is the first assignment
29/03/2021 - 04:00

Activities
Work
Need to clock in at work
29/03/2021 - 09:00

Figure 5 9 - Calendar Month Screen

Figure 5 10 - Calendar Day Screen

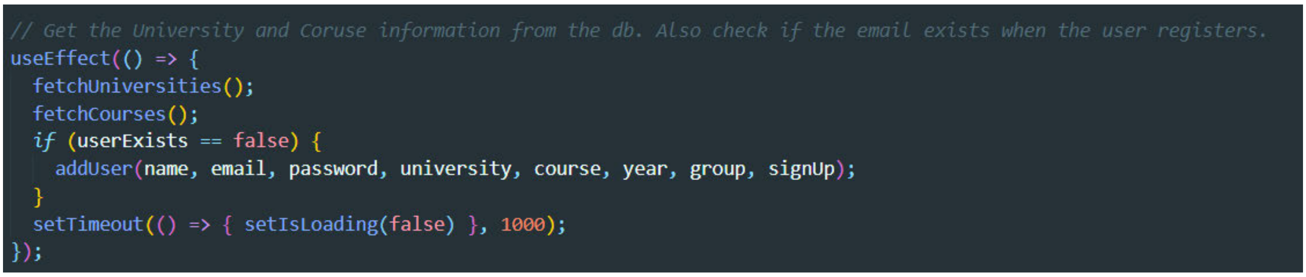
A demonstration was done to a group of volunteers showing the MVP to get some feedback on their thoughts on the features and any suggestions as to how to improve them. The volunteers liked the design and idea of the application. The only improvement that was given was to give the user a way to know how full the class is before changing their timetable.

5.2 INCREMENT 2

5.2.1 DEVELOPMENT CHALLENGES AND SOLUTIONS

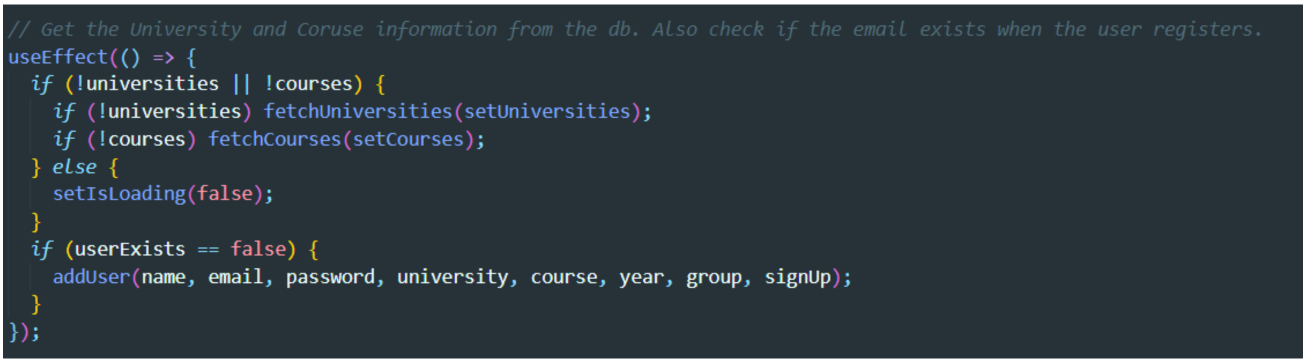
5.2.1.1 STATE HANDLING

One of the issues that were noted during the testing phase of the first increment was that at times, the application would suddenly crash, and it would need to be reloaded to carry on using it. After much research, the reasoning for that was the usage of setTimeOut, which delayed the executing of the UI for n number of seconds – waiting for the data to be loaded from the database. That code can be seen in figure 5.11. However, after removing the setTimeOut method and checking if the data is loaded, the issue was fixed, and the application no longer crashed. The fixed code can be seen in figure 5.12. This was caused due to lack of knowledge as this is a new framework that I am still learning.



```
// Get the University and Coruse information from the db. Also check if the email exists when the user registers.
useEffect(() => {
  fetchUniversities();
  fetchCourses();
  if (userExists == false) {
    addUser(name, email, password, university, course, year, group, signUp);
  }
  setTimeout(() => { setIsLoading(false) }, 1000);
});
```

Figure 5 11 - State Issue

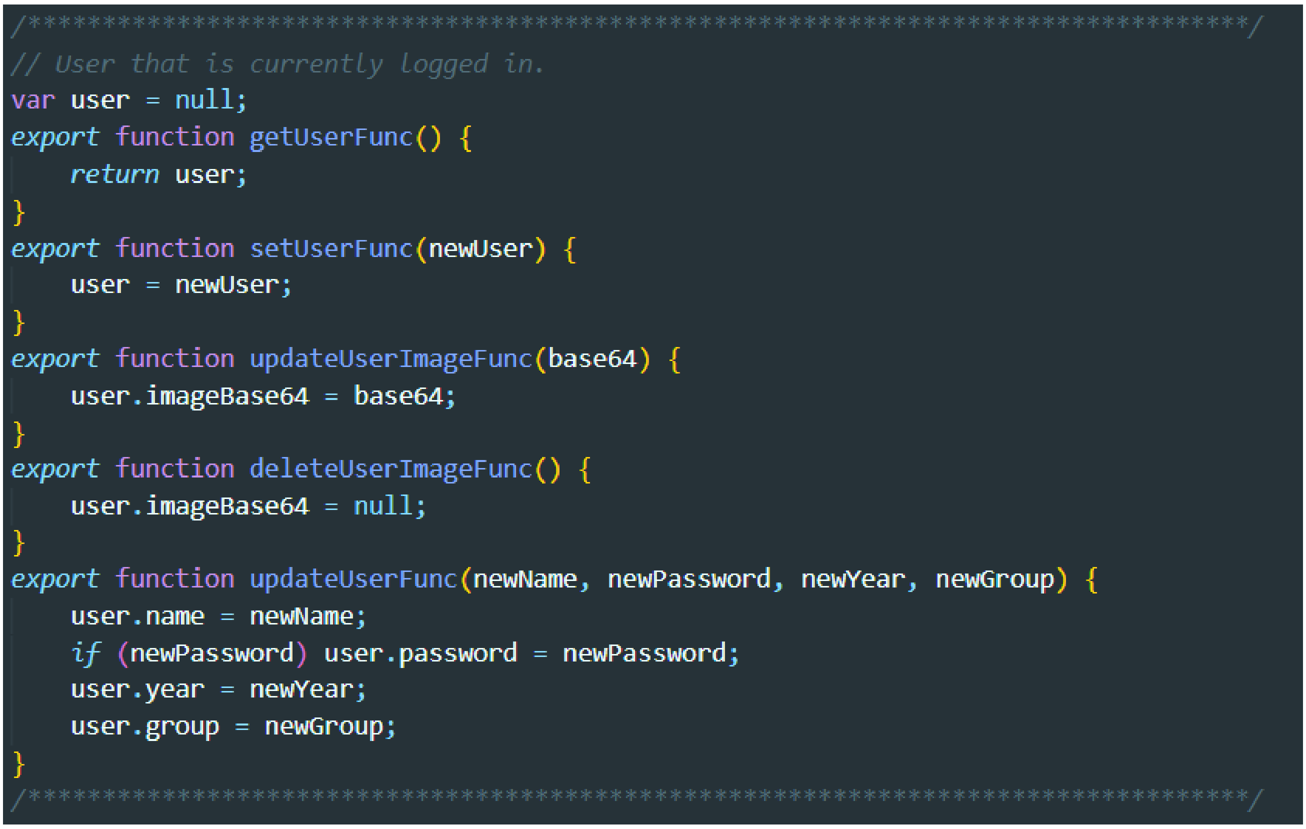


```
// Get the University and Coruse information from the db. Also check if the email exists when the user registers.
useEffect(() => {
  if (!universities || !courses) {
    if (!universities) fetchUniversities(setUniversities);
    if (!courses) fetchCourses(setCourses);
  } else {
    setIsLoading(false);
  }
  if (userExists == false) {
    addUser(name, email, password, university, course, year, group, signUp);
  }
});
```

Figure 5 12 - State Issue Solution

5.2.1.2 GLOBAL STORAGE

The idea behind react native is that each screen is independent of one another and the only way to pass data from one file to another is through navigation params. However, there was certain information, such as the logged-in user's detail, that was needed across multiple screens. Therefore, a file was created to hold global variables that the entire application would have access to, regardless of what screen it is in. This solves the issue of having shared information. However, it breaks one of the most important rules of building a react native application which is independence between screens. Figure 5.13 shows the global file, and it can be accessed simply by importing that file and using its functions.



```
/*******************************************************************************/
// User that is currently logged in.
var user = null;
export function getUserFunc() {
    return user;
}
export function setUserFunc(newUser) {
    user = newUser;
}
export function updateUserImageFunc(base64) {
    user.imageBase64 = base64;
}
export function deleteUserImageFunc() {
    user.imageBase64 = null;
}
export function updateUserFunc(newName, newPassword, newYear, newGroup) {
    user.name = newName;
    if (newPassword) user.password = newPassword;
    user.year = newYear;
    user.group = newGroup;
}
/*******************************************************************************/
```

Figure 5 13 - Global Variable File

5.2.1.3 TOP BAR NAVIGATION RENDERING

The to-do list was divided into three screens that were held in the top bar navigation component. Each screen represented different information (upcoming, overdue, and completed tasks). Whenever a task is marked as complete, it should move to the completed task screen and be displayed there. However, the top bar navigation renders all three screens at the same time and there is no way to re-render a screen if the user is not currently on it. Therefore, when a task is moved to complete, it would not update the UI until the page is exited and re-entered. Due to the lack of knowledge in the framework, I could not find an optimal solution for this issue. Therefore, the solution was that whenever a task is marked as complete, the application would navigate to a different screen then back to the to-do list to force all top nav screens to re-render and therefore display the correct information.

5.2.2 TOOLS AND LIBRARIES USED

Several powerful libraries were used in the development of this increment. The libraries are:

- react-native-dashboard (https://www.npmjs.com/package/react-native-dashboard)
- expo-location (https://docs.expo.io/versions/latest/sdk/location/)
- geolib (https://www.npmjs.com/package/geolib)
- expo-local-authentication (https://docs.expo.io/versions/latest/sdk/local-authentication/)
- expo-image-picker (https://docs.expo.io/versions/latest/sdk/imagepicker/)

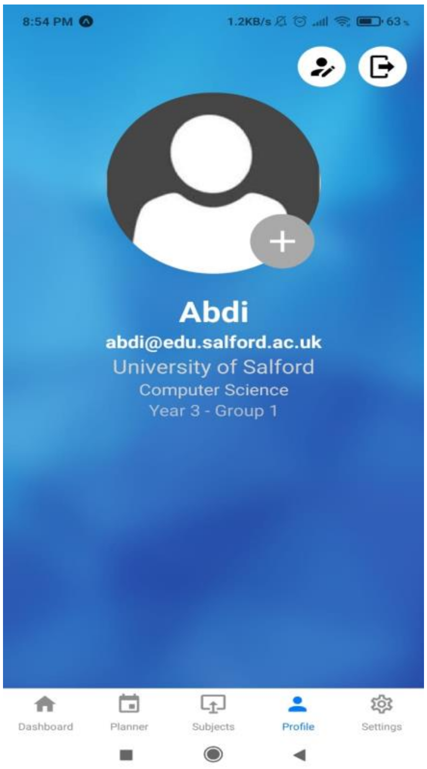
The dashboard was used to represent the main information that the user needed to know about. This information consisted of the tasks they needed to do for the week, which was broken up into exams, assignments, and other activities. There were not any other good libraries that I could find which offered a dashboard implementation and therefore, react-native-dashboard was used (Zappi, 2020).

Expo location was used to get permission from the user to turn on the device's GPS and to access the location of the device (Expo, 2021). Alongside it, geolib was used to calculate the distance between the user's location and the university's location (manuelbieh, 2020). Expo local authentication was used to access the device's biometrics such as face ID and finger ID to authenticate the user and see if we have the right person (Expo, 2021). These three libraries were used to check if we have the right user and if the user is within the vicinity of the university when they want to register for a class.

The image picker library was an excellent choice for accessing the gallery and getting an image from there. There were many options and libraries to choose from, but this was the easiest approach and allowed for a base64 conversion which was used to store the image in the database (Expo, 2021).

5.2.3 MINIMAL VIABLE PRODUCT

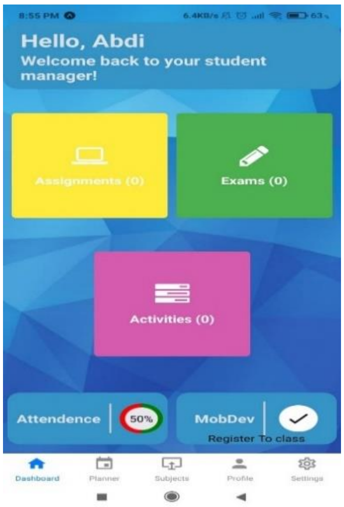
Figure 5.14 displays the results of the implementation for the profile. It looks similar to the initial design but there is a lot of space at the bottom of the screen. Therefore, some changes will need to be made and new UI components will need to be added to this section to enhance it. This will be a task for the next increment.



Abdi
abdi@edu.salford.ac.uk
University of Salford
Computer Science
Year 3 - Group 1

Figure 5 14 - Profile Screen

Figure 5.15 displays the dashboard. The UI ended up being different from the initial design due to the library that was used. The initial design had circle shapes representing the tasks while the library had squares. However, the UI looks better with this design, and the work it took to make it was not time-consuming.

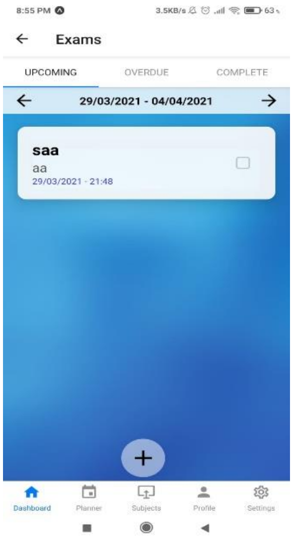


Hello, Abdi

Welcome back to your student manager!

Figure 5 15 - Dashboard Screen

Figure 5.16 shows the implementation of the to-do list. This looks similar to the calendar day view but with extra features and UI components. The initial design and the implementation are similar. However, extra UI components have been added to the final product to enhance the user experience and add extra useful functionality to the application.



saa
aa
29/03/2021 - 21:48

Figure 5 16 - To-Do List Screen

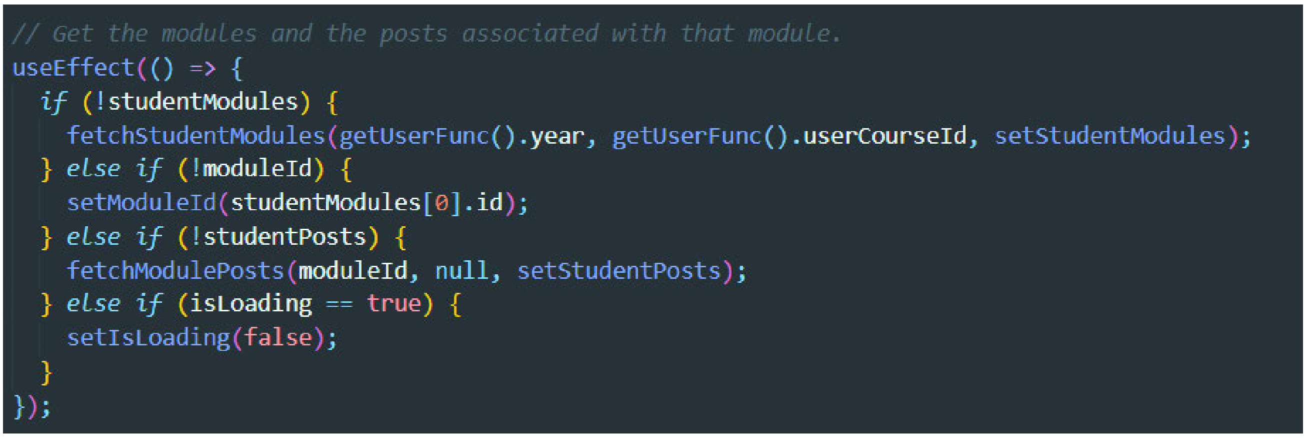
A demonstration was done to a group of volunteers at the end of this increment's implementation phase showing them the application to get feedback on their thoughts on the features and any suggestions as to how to improve them. They liked the application and the UI but no suggestions for improvements were given during the demonstration. I will be taking that as a positive sign.

5.3 INCREMENT 3

5.3.1 DEVELOPMENT CHALLENGES AND SOLUTIONS

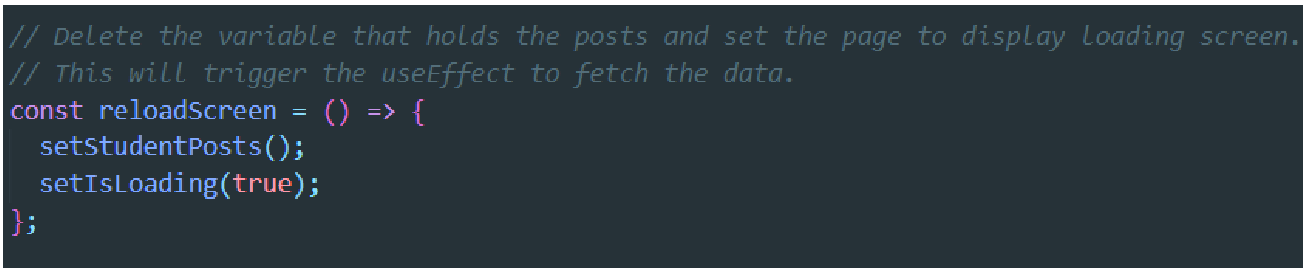
5.3.1.1 UPDATING SUBJECT FEED

Whenever a user makes a post, it would not appear on the subject feed page until the page is exited and re-entered. At first, I thought the issue was to do with the states not being set up correctly. However, as shown in figure 5.17, the states are not using a timer and are waiting for the data to be read properly. After much research, the source of the issue was found. The reason was due to the base64 image of the user. When a post is added, the name and image of the user are stored alongside the post information in the database. This was done to avoid going to the user table to get the user information every time a post needed to be displayed as it would take a long time and required more power. However, since the image size is massive, it sometimes takes a long time to add the image to the database which causes the newly added post not to be rendered since it is still not in the database. As a side note, this issue would not come up when the user had no image. The solution for this was to reload the page every time a post is added to give it more time which allows the data to be stored and therefore rendered on the screen. The solution can be seen in figure 5.18.



```
// Get the modules and the posts associated with that module.
useEffect(() => {
  if (!studentModules) {
    fetchStudentModules(getUserFunc().year, getUserFunc().userCourseId, setStudentModules);
  } else if (!moduleId) {
    setModuleId(studentModules[0].id);
  } else if (!studentPosts) {
    fetchModulePosts(moduleId, null, setStudentPosts);
  } else if (isLoading == true) {
    setIsLoading(false);
  }
});
```

Figure 5 17 - Subject Feed Loading Issue

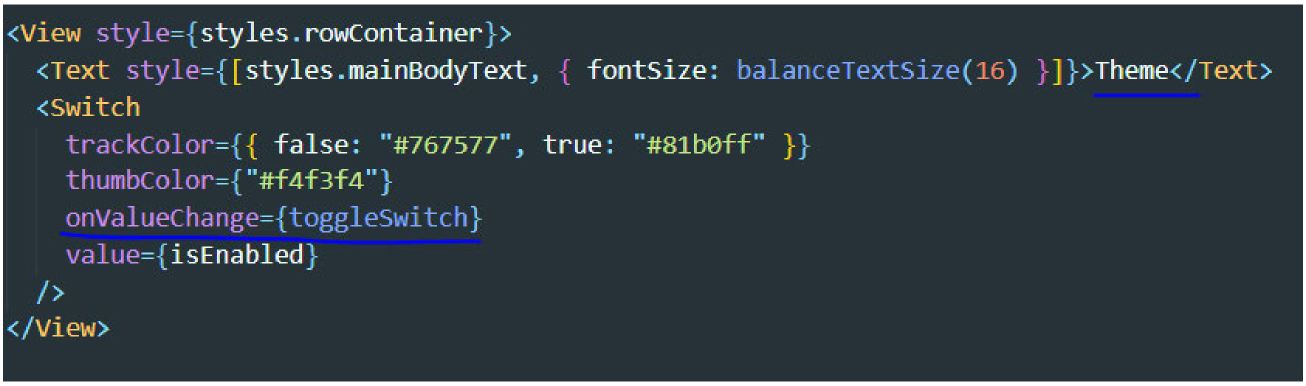


```
// Delete the variable that holds the posts and set the page to display loading screen.
// This will trigger the useEffect to fetch the data.
const reloadScreen = () => {
  setStudentPosts();
  setIsLoading(true);
};
```

Figure 5 18 - Subject Feed Loading Issue Solution

5.3.1.2 THEME ISSUE

The idea behind the theme change is to allow users to alternate between light and dark modes. This will help with their vision and reduce strain as it would make sense to use dark mode during the night to reduce the stress on the eyes. However, it does not seem possible to change the mode of the device using React Native. The only solution that was possible was to create a custom file and give it a default light background style and another default dark background style and use that file throughout the entire application. Therefore, once the theme is switched, it would change what mode to use and render it accordingly. Having said that, most of the pages were using background images so it would not make sense to do this. Also, this would require a huge refactoring of the entire application which unfortunately there was no time for. Therefore, as shown in figure 5.19, the UI component is there but the functionality has not been implemented. This feature will go into the chapter of the future work.



```
<View style={styles.rowContainer}>
  <Text style={[styles.mainBodyText, { fontSize: balanceTextSize(16) }]}>Theme</Text>
  <Switch
    trackColor={{ false: "#767577", true: "#81b0ff" }}
    thumbColor={"#f4f3f4"}
    onValueChange={toggleSwitch}
    value={isEnabled}
  />
</View>
```

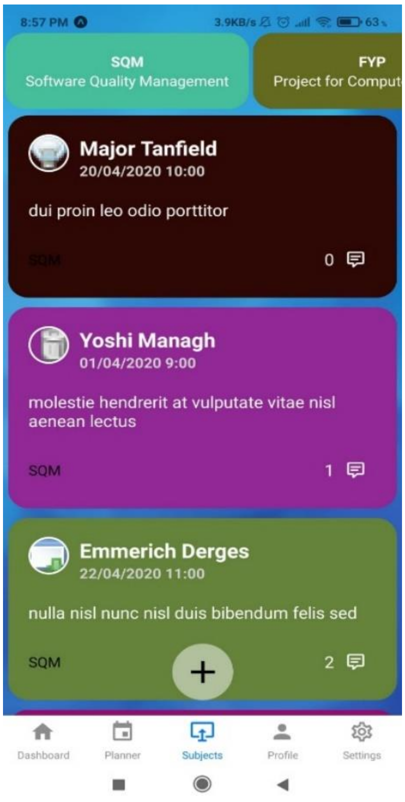
Figure 5 19 - Theme UI

5.3.2 TOOLS AND LIBRARIES USED

Most of the features that were developed during this increment were done from scratch without the aid of any big libraries. However, the only library worth mentioning is the qs library. During the development of the setting. There was a button that allowed users to send an email reporting any issues. The format was causing a lot of issues until I found qs which is a “query string parsing and stringifying library (ljharb & nlf, 2021)”. This made the formatting of the string easy which allowed me to send emails.

5.3.3 MINIMAL VIABLE PRODUCT

Figure 5.20 displays the result of the implementation for the subject feed screen. The design and implementation are very similar. The only difference is that the size and shape of the components have slightly changed to fit the phone and make it more user-friendly. Figure 5.21 shows the result of the implementation of the comment page. Figure 5.22 displays the settings screen. The UI was very similar to the initial design and that was a good sign. It was not too complex to design as it did not use any external UI libraries which meant that it was not a hard task getting it to look the same as the initial design.



SQM
Software Quality Management

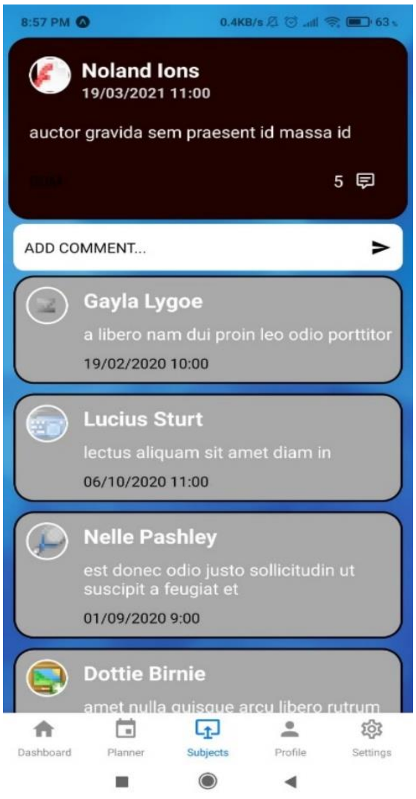
FYP
Project for Comput

Major Tanfield
20/04/2020 10:00
dui proin leo odio porttitor
SQM 0

Yoshi Managh
01/04/2020 9:00
molestie hendrerit at vulputate vitae nisl aenean lectus
SQM 1

Emmerich Derges
22/04/2020 11:00
nulla nisl nunc nisl duis bibendum felis sed
SQM 2

Figure 5 20 - Subject Feed Screen



Noland Ions
19/03/2021 11:00
auctor gravida sem praesent id massa id
SQM 5

ADD COMMENT...

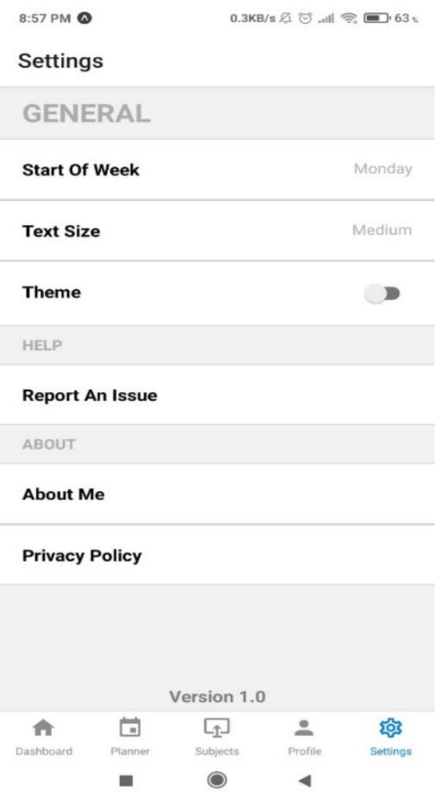
Gayla Lygoe
a libero nam dui proin leo odio porttitor
19/02/2020 10:00

Lucius Sturt
lectus aliquam sit amet diam in
06/10/2020 11:00

Nelle Pashley
est donec odio justo sollicitudin ut suscipit a feugiat et
01/09/2020 9:00

Dottie Birnie
amet nulla quisque arcu libero rutrum

Figure 5 21 - Comment Post Screen



Settings

GENERAL

Start Of Week Monday

Text Size Medium

Theme

HELP

Report An Issue

ABOUT

About Me

Privacy Policy

Version 1.0

Figure 5 22 - Settings Screen

The same group of volunteers that were used in the previous increments were brought together to present them with a demonstration of the features in this increment to get feedback from them. One of the feedbacks was the fact that the theme option in the settings was not working which has been noted above. Another was the fact that in the subject feed page, whenever a subject is picked, it should be the first one to be displayed in the subject list at the top of the screen. This would give the user an indicator as to what subject posts they are looking at.

5.4 SUMMARY

This chapter covered the development and implementation that was needed to achieve the specification requirements of the project. It covered the tools and libraries that were used to achieve the goals and discussed the challenges that were faced and how those challenges were overcome. Additional features were added to this project during the development phase and some of these features had bugs in them or were not fully completed. However, all the requirements that were mentioned in the previous chapter were met, which makes this a successful phase.

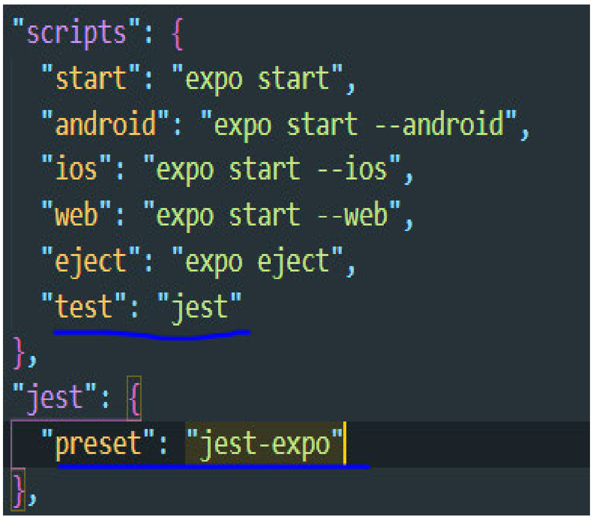
6 TESTING AND RESULTS

This chapter will discuss the testing that was done in the project and an analysis of the results. Similar to the previous chapters, this will be broken up into three increments. Each increment will cover the testing strategy that was done, along with a retrospective and an analysis of the results. To assess the success of the project, an extensive and detailed test strategy is required. This is to avoid "discrepancies in the software development process (Khan & Khan, 2014)". It will also ensure that the quality of the application is high, and no major issues are being missed out.

The testing strategy will be broken up into two stages:

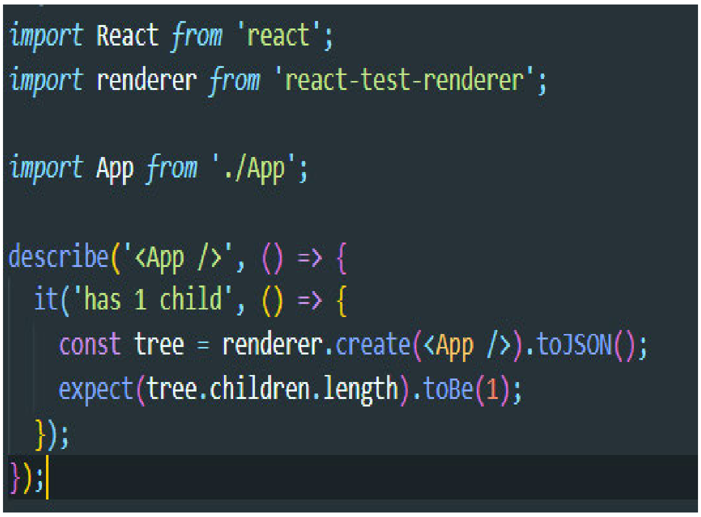
- Unit testing using Jest Testing Framework
- User Testing
- Regression Testing (if applicable)

Jest Testing Framework is an open-sourced JavaScript framework for unit testing and is designed to work with React and React Native applications (Expo, n.d.). Normal unit testing is not ideal since it is time-consuming and can easily get very complicated. Jest framework reduces that complexity and makes testing very quick and easy (Vaidya, 2020). The framework has many libraries for assertion along with test runners and mock libraries that simulate the entire application or parts of it (Vaidya, 2020). All that is needed is to install the dependencies and edit the package.json file to reflect that Jest is being used. Figure 6.1 shows the required changes in the json package file to use the Jest Framework. Figure 6.2 shows a very basic example of a Jest unit test which checks if the number of children the App component has is 1.



```
"scripts": {
  "start": "expo start",
  "android": "expo start --android",
  "ios": "expo start --ios",
  "web": "expo start --web",
  "eject": "expo eject",
  "test": "jest"
},
"jest": {
  "preset": "jest-expo"
},
```

Figure 6 1 - Json Package File



```
import React from 'react';
import renderer from 'react-test-renderer';

import App from './App';

describe('<App />', () => {
  it('has 1 child', () => {
    const tree = renderer.create(<App />).toJSON();
    expect(tree.children.length).toBe(1);
  });
});
```

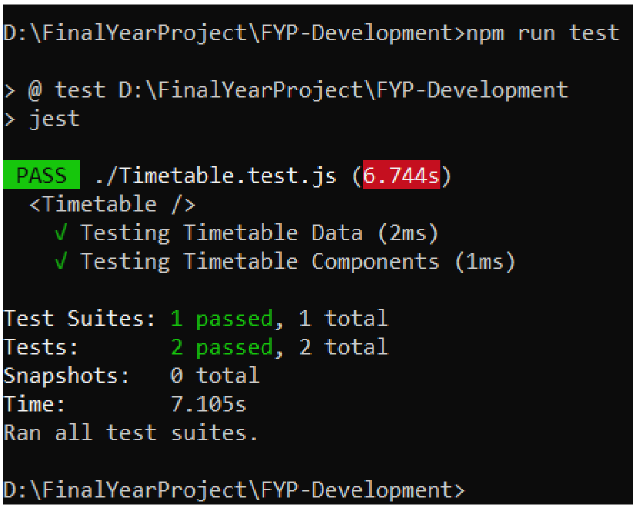
Figure 6 2 - Simple Jest Unit Test

6.1 INCREMENT 1

6.1.1 TESTING STRATEGY

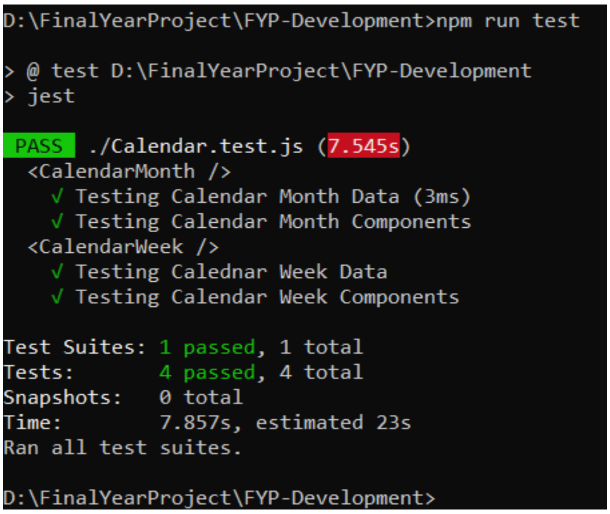
6.1.1.1 UNIT TESTING WITH JEST TESTING FRAMEWORK

Two different unit tests were executed for this increment to test out the timetable and the calendar. Figure 6.3 shows the results of the first test which was for the timetable screen. It checks if the data is correct and if the components were represented in the right way. Figure 6.4 shows the results of the second test which was for the calendar screen. It checks for the number of components and if they are correct. It also checks for the data that is used and if that can be passed to the component without causing issues.



```
D:\FinalYearProject\FYP-Development>npm run test

> @ test D:\FinalYearProject\FYP-Development
> jest

 PASS  ./Timetable.test.js (6.744s)
  <Timetable />
    √ Testing Timetable Data (2ms)
    √ Testing Timetable Components (1ms)

Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:   0 total
Time:        7.105s
Ran all test suites.

D:\FinalYearProject\FYP-Development>
```

Figure 6 3 - Timetable Jest Unit Test



```
D:\FinalYearProject\FYP-Development>npm run test

> @ test D:\FinalYearProject\FYP-Development
> jest

 PASS  ./Calendar.test.js (7.545s)
  <CalendarMonth />
    √ Testing Calendar Month Data (3ms)
    √ Testing Calendar Month Components
  <CalendarWeek />
    √ Testing Calednar Week Data
    √ Testing Calendar Week Components

Test Suites: 1 passed, 1 total
Tests:       4 passed, 4 total
Snapshots:   0 total
Time:        7.857s, estimated 23s
Ran all test suites.

D:\FinalYearProject\FYP-Development>
```

Figure 6 4 - Calendar Jest Unit Test

6.1.1.2 USER ACCEPTANCE TESTING

For user testing, a volunteer stepped forward to test the application to see if they could find any bugs in the application. They also gave feedback on how easy and intuitive it was for them to use the features that the application provided. A summary of the feedback is as follows:

- Was able to login and register without a problem.
- There was an issue where at times, the application would crash for no specific reason.
- The UI is good, but it could be a little bit smoother when transitioning from the timetable screen to the calendar screen.
- Tasks were displayed correctly. Was able to delete and edit tasks which worked fine. There was no option to add tasks and that should be there.
- Was able to move between lessons but no indication as to whether the class is full or not.

These were some great feedbacks and more discussion on these points will be made in the retrospective section.

6.1.2 ANALYSIS AND RETROSPECTIVE

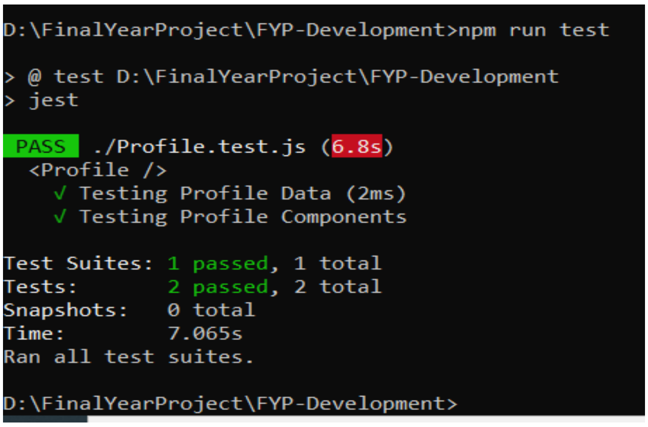
Based on the feedback from the volunteer and the demonstration that took place, it is safe to say that this was a successful increment. The requirement specifications were all met, and the designs of the sketches looked similar, to a certain extent, to the implemented screens. However, some extra checks were missed out such as having a lesson capacity where students cannot go to a class if it is full. This was not in the original specification, but it will be added as an optional task in the next increment. Another issue that the users pointed out was the fact that at times, the app would crash, and it would need to be reloaded to carry on using it. This is probably due to the lack of my knowledge in the framework, but this issue will be looked at in more depth in the next increment.

6.2 INCREMENT 2

6.2.1 TESTING STRATEGY

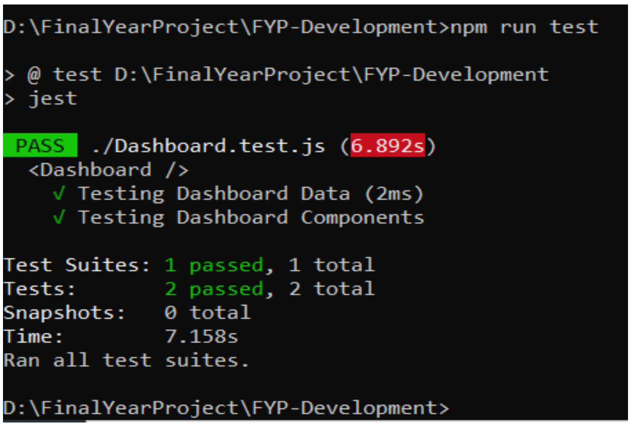
6.2.1.1 UNIT TESTING WITH JEST TESTING FRAMEWORK

Several unit tests were executed during this increment to make sure that all the components were set up correctly and that the data is present and in the right format. Figure 6.5 shows the results of the test that ran on the profile screen, figure 6.6 shows the results that ran on the dashboard screen, and figure 6.7 shows the results of the test that ran on the to-do list screen.



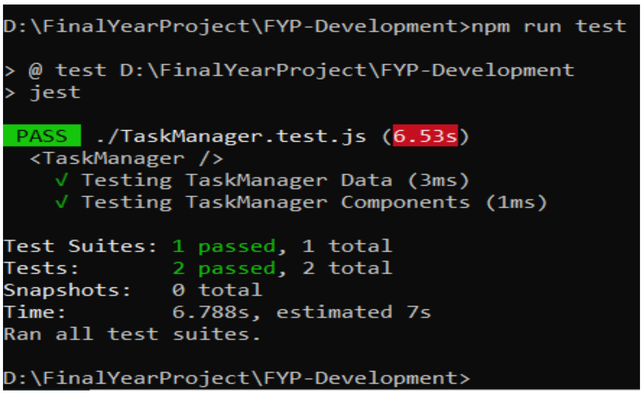
```
D:\FinalYearProject\FYP-Development>npm run test

> @ test D:\FinalYearProject\FYP-Development
> jest

 PASS  ./Profile.test.js (6.8s)
  <Profile />
    √ Testing Profile Data (2ms)
    √ Testing Profile Components

Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:   0 total
Time:        7.065s
Ran all test suites.

D:\FinalYearProject\FYP-Development>
```

Figure 6 5 - Profile Jest Unit Test



```
D:\FinalYearProject\FYP-Development>npm run test

> @ test D:\FinalYearProject\FYP-Development
> jest

 PASS  ./Dashboard.test.js (6.892s)
  <Dashboard />
    √ Testing Dashboard Data (2ms)
    √ Testing Dashboard Components

Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:   0 total
Time:        7.158s
Ran all test suites.

D:\FinalYearProject\FYP-Development>
```

Figure 6 6 - Dashboard Jest Unit Test



```
D:\FinalYearProject\FYP-Development>npm run test

> @ test D:\FinalYearProject\FYP-Development
> jest

 PASS  ./TaskManager.test.js (6.53s)
  <TaskManager />
    √ Testing TaskManager Data (3ms)
    √ Testing TaskManager Components (1ms)

Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:   0 total
Time:        6.788s, estimated 7s
Ran all test suites.

D:\FinalYearProject\FYP-Development>
```

Figure 6 7 - To-Do List Jest Unit Test

6.2.1.2 USER ACCEPTANCE TESTING

The same volunteer was used for this increment to test the application to see if they could find any bugs in the developed features. Some feedback was given during the test on how the application performed. The feedback is as follows:

- The profile presented the correct information, and the details could be changed. These changes are reflected in the profile.
- When the group is changed, the timetable keeps the custom lessons that were there previously. This is wrong and needs to be fixed.
- It would be good if there was an option to take a picture when choosing the profile image.
- The dashboard was intuitive and gave a range of good information.
- The current lesson to register for was not correct, based on the timetable.
- The attendance does not give that much information. Maybe having one for each subject would be more useful.

These were some great feedbacks and more discussion on these points will be made in the retrospective section.

6.2.1.3 REGRESSION TESTING

Regression testing was covered at the end of the second increment to ensure that none of the features that were developed in the first increment were broken. After running the same tests that were run by the volunteer in the previous increment, along with some additional tests, I can conclude that none of the features broke and everything worked as it did before.

On a side note, the additional feature that was mentioned in increment one where the class size would be displayed has not been implemented and will be left out. Another task that was omitted was adding the feature in the calendar where users can add tasks. This has been moved to the task screen that was developed during this increment.

6.2.2 ANALYSIS AND RETROSPECTIVE

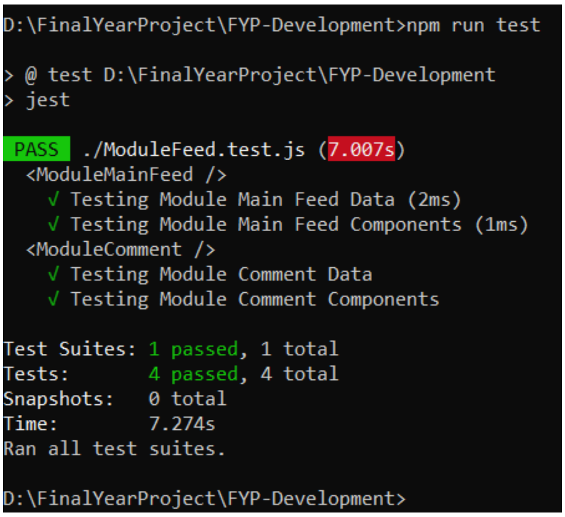
Based on the demonstration that took place and the volunteer's feedback, it was clear that this increment was very successful. However, a mistake was made in trying to achieve too many additional features that caused issues that affected the user experience. However, the requirement specifications were all met, and the designs of the sketches looked similar to the implemented screens, if not enhanced. One of the feedback that was given was to add an option where the user took a picture when adding an image to their profile. This will be looked at in the next increment and implemented. Another feedback was the attendance not being clear and a suggestion to breaking it down into subjects. However, since this was an extra feature, it will be kept the same unless there is some time at the end of the third increment where extra work can be done.

6.3 INCREMENT 3

6.3.1 TESTING STRATEGY

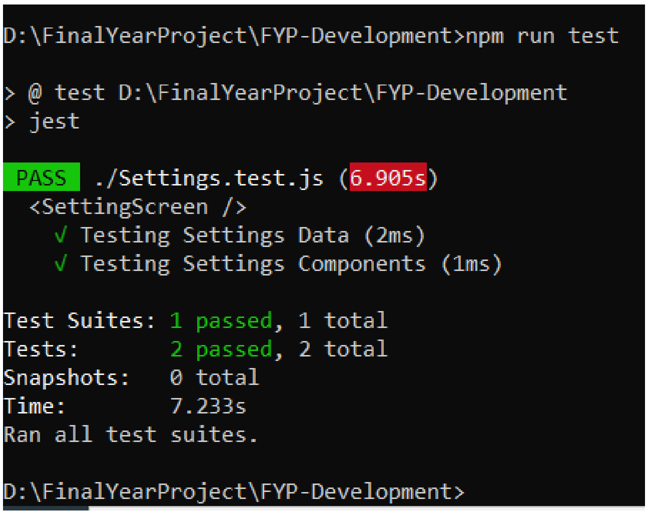
6.3.1.1 UNIT TESTING WITH JEST TESTING FRAMEWORK

Several Jest unit tests were executed during this increment to make sure that the data was being read correctly and that the components were structured as expected. Figure 6.8 shows the results of the test that ran on the subject feed screen. Figure 6.9 shows the results of the test that ran on the settings screen.



```
D:\FinalYearProject\FYP-Development>npm run test

> @ test D:\FinalYearProject\FYP-Development
> jest

 PASS  ./ModuleFeed.test.js (7.007s)
  <ModuleMainFeed />
    √ Testing Module Main Feed Data (2ms)
    √ Testing Module Main Feed Components (1ms)
  <ModuleComment />
    √ Testing Module Comment Data
    √ Testing Module Comment Components

Test Suites: 1 passed, 1 total
Tests:       4 passed, 4 total
Snapshots:   0 total
Time:        7.274s
Ran all test suites.

D:\FinalYearProject\FYP-Development>
```

Figure 6 8 - Subject Feed Jest Unit Test



```
D:\FinalYearProject\FYP-Development>npm run test

> @ test D:\FinalYearProject\FYP-Development
> jest

 PASS  ./Settings.test.js (6.905s)
  <SettingScreen />
    √ Testing Settings Data (2ms)
    √ Testing Settings Components (1ms)

Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:   0 total
Time:        7.233s
Ran all test suites.

D:\FinalYearProject\FYP-Development>
```

Figure 6 9 - Settings Jest Unit Test

6.3.1.2 USER ACCEPTANCE TESTING

The same volunteer that was used in the previous two increments was also used for this one. Their responsibilities were to test the application and provide feedback as to any bugs they found or any recommendations to improve the application. Some of the feedback that was given during the testing were as follows:

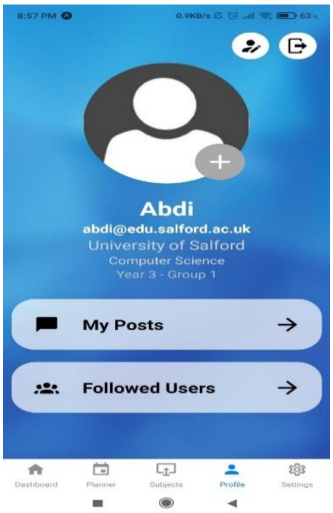
- The main subject feed page looked very nice and easy to use.
- It is hard to know what subject is being looked at since there is no indication. Maybe add the name of the subject in each post.
- Sometimes, the post does not show up when I add a new one.
- Maybe make the comments in different colours too, like the posts.
- The theme setting does not work.

These were some good feedbacks and more discussion on these points will be made in the retrospective section.

6.3.1.3 REGRESSION TESTING

Regression testing was covered at the end of the third increment to ensure that none of the features that were developed in the two increments were broken. After running the same tests that were run by the volunteer in the previous increments, along with some additional tests of my own, I can conclude that none of the features broke and everything worked as it did before.

Regarding the second increment, some changes were made where additional information was added to the profile screen and this can be seen in figure 6.10.



Abdi
abdi@edu.salford.ac.uk
University of Salford
Computer Science
Year 3 - Group 1

Figure 6 10 - Enhanced Profile Screen

6.3.2 ANALYSIS AND RETROSPECTIVE

Based on the demonstration that took place and the volunteer's feedback, this increment was a success and has achieved all the core requirements. The users were happy with the UI and said it was comfortable to navigate which meant that it was user-friendly. The settings offered a lot of options for a wide range of people to accommodate everyone and ensure accessibility. However, there were a few features that were not working as expected. The primary example is the theme options which allowed users to change the application from light mode to dark mode and vice versa. There was also the issue of the feed not being clear and the user not knowing what subject feed they are looking at. Some small minor changes will take place to fix this issue as it will be easy and quick. However, the theme will not be implemented during this project and will be talked about in the future works section.

6.4 SUMMARY

This chapter covered a variety of testing strategies that were used to ensure that the development of the application was done appropriately and that there were issues to be found. This chapter also focused on reflecting on what happened and making sure that each increment was optimised based on the feedback of the previous one. The majority of the tests passed and those that did not were fixed in the later increments and tested again. Overall, this was a successful testing phase as all the core tests passed and the feedback was positive from the testers.

7 CRITICAL EVALUATION

This chapter will evaluate the overall project and will discuss the different aspects and changes that were required to happen for the project to be successful. It will be broken up into several parts. The first part will go over the objectives and evaluate whether they have been achieved and if any of them needed to be modified. The second part will go over the project plan and how it held up to the execution of that plan. The third part will look at the final product and evaluate it to see if it met all the requirements and any comment that needs to be made. The penultimate part will be the lessons learned throughout this journey, and then this chapter will end with a summary.

7.1 EVALUATION OF THE OBJECTIVES

7.1.1 OBJECTIVE 1: EXPLORING DIFFERENT PLATFORMS TO BUILD THE APPLICATION

As shown in chapter 3.2, a variety of different approaches were reviewed to find which platform would be optimal for developing the application. React native was chosen for reasons mention in that chapter. Looking back, I am very happy with that decision as React Native was a very easy platform to learn and offered a wide range of powerful tools and libraries that helped in the development journey. It is also very popular and there was a lot of support that I found in the community that helped with most issues and problems that I had.

7.1.2 OBJECTIVE 2: SURVEY STUDENTS TO GET THE REQUIREMENTS OF THE APPLICATION

The survey took the form of a focus group, which allowed me to talk to a few students and get their input on what they would like to see. These were made into requirements and use case diagrams were drawn from those requirements. I believe holding a focus group was the most efficient approach to take as it can be all done in one go and you can create discussion and extract a lot of useful information, as opposed to other forms. This was also useful since we are in a pandemic and everything had to be done remotely.

7.1.3 OBJECTIVE 3: INVESTIGATING DIFFERENT HOSTING SERVERS

As shown in chapter 3.3, several different hosting servers were looked at to find which one would be the most optimal for this project. Amazon Web Services (AWS) was picked as it provides lots of powerful features and has services that fulfil my requirements to complete this project. AWS is also accepted internationally as one of the best hosting services and is used by many companies and will therefore benefit me in the future. An EC2 Bucket was created in AWS and a Relational Database was hosted on the Ubuntu Instance that is sat in the EC2 Bucket.

7.1.4 OBJECTIVE 4: BUILDING RESTFUL APIS

The React Native application did not directly access the database and instead used APIs to request the information it needed. These APIs were written in PHP and were hosted in AWS alongside the database. POST and GET fetch statements were used to connect to those APIs. These were done successfully, and data was able to be retrieved, stored, and updated using the APIs.

7.1.5 OBJECTIVE 5, 6, 7, & 8: TIMETABLE, BIOMETRIC, GEOLOCATION, TO-DO LIST, AND SUBJECT FEED

The timetable slightly deviated from the original objective since I did not have a clear idea of what needed to happen. After speaking to students and going through potential designs, the objectives for the timetable were changed. Some functionality had to be removed such as the one where students can put down lessons they have missed. This has been changed so users can register for a class and give them the option to change the timetable if they missed that lesson.

The registration was done using the device's biometrics to determine if this is the right person and the location of the user was used to determine if they are within the vicinity of the university. This was done successfully, and both these features worked as expected. The only downside is that extensive testing was not possible with the GPS since we were working remotely and in lockdown which meant going to University would be very difficult. Therefore, the location was changed to where I was residing during the testing phase to see if it would work or not.

The to-do list and the subject feed were achieved and did exactly what they intended to do. The user was able to add tasks and edit them as they wanted. The subject feed gave students the possibility to interact with each other and provide support in the subjects they take.

I was very happy with the objectives and believe the changes that took place were necessary to improve the user experience.

7.1.6 OBJECTIVE 9 & 10: ACCESSIBILITY AND TESTING

Accessibility was taken into consideration throughout the development of the application. This was done through the use of appropriate colours in the application and making sure the size of the fonts was readable for people with vision impairment. On the settings page, the user can configure certain aspects of the application to make their user experience easier and smoother. I was happy with the level of consideration that was put into place to ensure that the application can be accessible to all.

Testing was done at each stage of the application to ensure that the new features were developed correctly and adhere to the specification while at the same time testing the existing features and ensuring that they are still working as they should. Several different strategies were covered during this project to make me confident that enough emphasis was put into testing.

7.2 EVALUATION OF THE PROJECT PLAN & PRODUCT

Agile was the methodology that was used for this project. This allowed the project to be broken up into three increments which helped with managing the project and getting quick feedback to better improve the user experience. Thanks to this approach, all the objectives that were mentioned above were achieved.

Each increment was broken up into requirement specification, development, and testing phase. All the core requirements that were mentioned were accomplished. However, thanks to the feedback during the development and testing phase, users asked for other features that were more appropriate for the situation. Some of these were not fully completed. Therefore, it would have been nice to have planned for some extra time at the end of the last increment where additional work could be done. Having said that, thanks to time spent before the development phase in learning about React Native and JavaScript helped immensely with the project and made it possible to succeed.

One of the challenges that were faced was picking the correct library for the task at hand. Libraries are very powerful and help with the development, but they also bring a lot of constraints with them. One of the constraints I faced was the fact that you were bound by the UI of the component and could not change how the component looks. This made some of the designs look different from the final product. However, this is not all bad since they sometimes came out better than the design, the case being the dashboard screen.

During the first increment, I realised that the order of the increments was not correct since the tasks in increment 1 linked to the tasks in increment 3. Therefore, a decision was made to swap increments 2 and 3 to make the flow of the development a lot smoother. This was possible to notice thanks to the use of increment and the detailed planning that was done before the start of the project. I was happy with the swap and it made the entire journey a lot easier.

The testing phase covered different types of testing strategies such as regression testing, user testing, and unit testing. However, I feel that more types of testing could have been added to ensure that the application was working optimally. One of the tests that I would have liked to add is stress testing. This would allow me to know the amount of load the application can take by having a large number of users interact with it at the same time. This will help me find the sub-optimal areas in my features which will help me improve them. The database was ready for this testing as a thousand users were stored and all of them could log in and use the application at the same time. This test could be done by combining the features that the LoadView platform provides (Lee, 2021) with React's Pref to simulate many users using the application and monitoring the performance of the application.

7.3 LESSONS LEARNT

This project has allowed me to gain many valuable experiences and helped me grow as a person and as a professional. One of the biggest lessons learned was how to manage my time and break down major tasks into sub-tasks. This was shown by my detailed planning and how I broke up the major objectives into requirements which made them easy to implement. The use of tools such as Gantt Chart and Trello has helped me immensely in managing my time and being able to complete everything within the given timeframe. This will help in nearly all aspects of life as time management is a universal skill that is needed for everything.

A very important skill that this project helped me improve is how to do research. The first half of this project was focused primarily on research while gathering and reading journals and conference papers. This is where the most up-to-date and reliable information is found and being able to read and understand them is vital for my growth and my future work. Through reading many papers, I have learned how to quickly skim a paper to extract the useful information that I need and determine whether it is worth reading in detail or looking for a different paper.

Another lesson learned was how the React Native framework operates as I had to learn it from scratch. This had helped me improve my JavaScript knowledge and my code proficiency as the more I code, the more aware I am of mistakes that I am making and optimising my code so that I perform efficiently. This skill will help me in the future with getting a job and becoming a proficient Software Engineer.

8 CONCLUSION

An in-depth investigation has been done on the topic of time management and how that affects a student's performance. It was shown that there is a direct correlation between the student's time management skills and how well they perform in their studies. It was also shown that there is a link to mental health and how well they can cope and handle stress.

The goal of this project was to improve the time management skills of the first-year computer science students at the University of Salford. This was done by building the application and providing a wide variety of features that students agreed will help them keep track of their activities and help them know what needs to be done every day.

Using agile methodology and sprints, I was able to optimise the user experience by providing features that students wanted and thought would help them the most. Many mistakes were made, and several iterations had to be done in certain tasks to ensure that it worked as the specification said. This was a learning experience and improved a lot of my skills.

Based on the feedback that was given by the students that tested the application and the students that were part of the demonstration, I believe this project was a success and it achieved all the core goals that this project set out.

8.1 FUTURE WORK

To turn this application into a professional platform that will benefit all students, further work needs to be done. These include:

- Using a platform such as Kubernetes where the web scraping application sits. This will be configured to run once every semester so that the timetable is always up to date without user intervention.
- Extending the application so that it will accept different universities. The infrastructure for this is in place and all that is needed is to update the database and the courses that are needed. In relation to that, it would be useful to have a standard format where universities and courses can provide their timetable so that they can all be read by the same application. This will ensure consistency across several university timetables.
- Having a different user type for teachers where they can answer student questions, make announcements, upload the assignments, or update the timetable if changes are needed.
- Adding an alarm for reminders of tasks, assignments, and exams. The infrastructure is in place and all that is needed is to update the database and edit the code so that the alarm is set.

There are of course many other works that can be done to enhance the application to better help the users. However, these are the ones I consider to be the important next steps.

REFERENCES

Ajayi, O. O. et al., 2018. Performance Evaluation of Native and Hybrid Android. *Communications on Applied Electronics (CAE),* p. 5.

Anon., 2011. *Xamarin.* [Online]
Available at: https://dotnet.microsoft.com/apps/xamarin

Anon., 2015. *React Native.* [Online]
Available at: https://reactnative.dev/

Anon., n.d. *Android Studio.* [Online]
Available at: https://developer.android.com/studio

Anon., n.d. *Apple Developer.* [Online]
Available at: https://developer.apple.com/discover/

Anon., n.d. *What Is Scrum?.* [Online]
Available at: https://www.scrum.org/resources/what-is-scrum

Antaramian, S., 2015. Assessing Psychological Symptoms and Well-Being: Application of a Dual-Factor Mental Health Model to Understand College Student Performance. *Journal of Psychoeducational Assessment,* p. 1.

Berry, D. M. & Kamsties, E., 2004. Ambiguity in Requirements Specification. In: *Perspectives on Software Requirements.* Boston: Springer, Boston, MA, pp. 7-44.

Britton, B. K. & Tesser, A., 1991. Effects of Time-Management Practices on College Grades. *Journal of Educational Psychology,* p. 4.

Creately, n.d. *Draw Use Case Diagrams Online with Easy-to-Use Tools and Templates.* [Online]
Available at: https://creately.com/diagram-type/use-case

Curtis, B., Krasner, H. & Iscoe, N., 1988. A field study of the software design process for large systems. *AMC,* p. 1268–1287.

Ebone, A., Tan, Y. & Jia, X., 2018. *A Performance Evaluation of Cross-Platform Mobile Application Development Approaches.* Gothenburg, Sweden, IEEE, p. 1.

Expo, 2021. *ImagePicker.* [Online]
Available at: https://docs.expo.io/versions/latest/sdk/imagepicker/

Expo, 2021. *LocalAuthentication.* [Online]
Available at: https://docs.expo.io/versions/latest/sdk/local-authentication/

Expo, 2021. *Location.* [Online]
Available at: https://docs.expo.io/versions/latest/sdk/location/

Expo, n.d. *Expo.* [Online]
Available at: https://docs.expo.io/guides/testing-with-jest/#structure-your-tests

Farzan Sasangohar, 2020. Effects of COVID-19 on College Students' Mental Health in the United States: Interview Survey Study. *JOURNAL OF MEDICAL INTERNET RESEARCH,* p. 4.

Furuskog, M. & Wemyss, S., 2016. Cross-platform development of smartphone applications: An evaluation of React Native. *Digitala Vetenskapliga Arkivet,* p. 16.

Glinz, M., 2005. *Rethinking the Notion of Non-Functional Requirements,* Zurich, Switzerland: Department of Informatics, University of Zurich.

Goetz, J. & Li, Y., 2018. Evaluation of Cross-Platform Frameworks for Mobile Applications. *International Journal of Engineering and Innovative Technology (IJEIT),* p. 5.

gomjellie, 2019. *react-native-timetable.* [Online]
Available at: https://github.com/gomjellie/react-native-timetable

Grechanik, M., McKinley, K. S. & Perry, D. E., 2007. *Recovering and using use-case-diagram-to-source-code traceability links.* Dubrovnik, Association for Computing Machinery, pp. 95-104.

Hansson, N. & Vidhall, T., 2016. Effects on performance and usability for cross-platform application development using React Native. *Digitala Vetenskapliga Arkivet,* p. 73.

Khan, M. E. & Khan, F., 2014. Importance of Software Testing in Software Development Life Cycle. *International Journal of Computer Science Issues,* pp. 120-123.

Kitzinger, J., 1994. The methodology of Focus Groups: the importance of interaction between research participants. *Sociology of Health and Illness,* pp. 103-121.

Lee, G., 2021. *Load Testing Applications Written in React.js.* [Online]
Available at: https://www.loadview-testing.com/blog/load-testing-applications-written-in-react-js/

ljharb & nlf, 2021. *qs.* [Online]
Available at: https://www.npmjs.com/package/qs

Macan, T. H., Shahani, C., Dipboye, R. L. & Phillips, A. P., 1990. College students' time management: Correlations with academic performance and stress. *Journal of Educational Psychology,* p. 1.

manuelbieh, 2020. *Geolib.* [Online]
Available at: https://www.npmjs.com/package/geolib

Moqups, 2021. *Mobile App Wireframe.* [Online]
Available at: https://moqups.com/

Pazirandeh, A. & Vorobyeva, E., 2015. Evaluation of Cross-Platform Tools for Mobile Development. *Department of Computer Science and Engineering - Gothenburg University Publications,* p. 11.

Piyadigama, D., 2020. *Expo vs React-Native-CLI.* [Online]
Available at: https://dinukapiyadigama.medium.com/expo-vs-react-native-cli-7a3019b2760d

Randstad, 2019. *a degree of uncertainty: looking at student wellbeing..* [Online]
Available at: https://www.randstad.co.uk/employers/areas-of-expertise/student-support/student-mental-health-report-2020/#:~:text=Over%20a%20quarter%20of%20UK,1800%20highlights%20some%20alarming%20statistics.

Team, S. E., 2017. *Software Integrity Blog.* [Online]
Available at: https://www.synopsys.com/blogs/software-security/top-4-software-development-methodologies/

Terry, J., n.d. *Benefits of Agile Development.* [Online]
Available at: https://www.planview.com/resources/articles/benefits-of-agile-development/#:~:text=There%20are%20many%20advantages%20of,the%20constraints%20of%20the%20budget

Vaidya, N., 2020. *Jest Framework Tutorial: How to use it.* [Online]
Available at: https://www.browserstack.com/guide/jest-framework-tutorial#:~:text=Jest%20is%20an%20open%2Dsource,React%20Native%20based%20web%20applications.&text=It%20stands%20out%20by%20virtue,Node%20JS%2C%20Babel%20and%20TypeScript.

Wix.com, 2021. *React Native Calendars.* [Online]
Available at: https://github.com/wix/react-native-calendars

Zappi, G., 2020. *react-native-dashboard.* [Online]
Available at: https://www.npmjs.com/package/react-native-dashboard

Appendices

APPENDIX A – PROJECT LOGBOOK

SEMESTER 1

WEEK 01: 28/09/2020 – 04/10/2020 (02 HOURS)

FYP Lecture:

Had my introduction to the final year project module where I met up with different lecturers and they talked about their group and what kind of projects they have supervised in the past. This helped me get an idea of the kind of project I want to do. From hearing the different talks, I concluded that I want to join the Development (mobile and 3D focused) team that is run by Norman Murray and Ian Drumm since I want my project to be an android mobile app. The reason for that is I want to strengthen my java skills and the mobile development module seems interesting to me.

TODO:

- Think about what my project will be about.
- Write down a paragraph explaining what my project idea is and send it to lecturers to get some feedback on them.
- Email Norman Murray regarding the groups I want to join.

WEEK 02: 05/10/2020 – 11/10/2020 (04 HOURS)

FYP Lecture and Seminar:

Had a lecture on the project proposal by Norman Murray. This gave me an idea of how to format my project proposal and what needs to be in there. The slides were very important, I'll make sure I go back to them on weeks 3 and 4 to help me write my proposal. In the seminar, I talked about my project idea and my supervisors were happy with it. They gave me some suggestions on things to change but the overall idea was approved! I was told that I need to test out the web scaping aspect straight away to make sure that it can be done.

Activities:

I managed to email Norman Murray about the groups I want to be in and was added to the mobile and 3D development team which I was very happy with. During the first official meeting with the group, I had the chance to present my idea which I have formulated during the previous week and it was approved. The focus of my project will be a mobile app that will help students (mainly first-year computer Science students) better manage their time. So, it will be essentially a google calendar that is tailored to students. The entire week was spent on coming up with the idea and brainstorming to think of how it will work and how doable it is.

TODO:

- Need to write actual objectives for the project to make the idea more concrete.
- Need to do some background research as to what already exists out there.
- Research web scraping since I will somehow need to get the timetables from https://www.firstyearmatters.info/cs/

WEEK 03: 12/10/2020 – 18/10/2020 (07 HOURS)

FYP Lecture and Seminar:

Had a lecture on methodologies by Julian Bass. This helped me explore the different methodologies that I might use for the project. This lecture was also important for writing up the Methodology section in the project proposal. It also gave me some useful tools to design my project such as using UML diagrams and user stories to plan the architecture. We were advised to try and create a Minimal Viable Product (MVP) by Christmas to show our supervisors and get feedback on them. In the seminar, I spoke to my supervisors about succeeding in creating a small web-scraping application, and that I had no problems so far.

Activities:

I have written an application over the week that allows me to web scrap FYM timetables which worked nicely. I managed to do it in java, but I needed to use an external library. Gradle was used to import that library into my project. My next task was to try and format the data and store it in a database – still need to figure out how to connect a database to my mobile application. I might also convert the web scraping application into a python API stand-alone that periodically web scrapes FYM and updates the database – not sure how tough that is.

- I used this template to learn how to web scrape: https://www.scrapingbee.com/blog/introduction-to-web-scraping-with-java/
- This is where I got the Gradle plugin that I needed for the XPath and web scraping: https://htmlunit.sourceforge.io/dependency-info.html
- Used this to help me create the XPaths I wanted: https://devhints.io/xpath
- Google Chrome plugin to help me test out the XPaths: https://chrome.google.com/webstore/detail/xpath-helper/hgimnogjllphhhkhlmebbmlgjoejdpjl?hl=en

I also started writing my project proposal which helped me formulate my plan even more. I managed to write the introduction, objectives, and the methodology that I will be using – which I have concluded to be agile for reasons mentioned in the project proposal. The remaining of the proposal will be written this week hopefully so that I can show it to my supervisors and get some feedback from it.

TODO:

- Need to do some background research as to what already exists out there.
- Email the university to set up a university server.
- Need to research how to connect a database with a mobile application.
- Need to start making a detailed plan using Gantt Chart.
- Research the difference between timetable, schedule, and calendar.

WEEK 04: 19/10/2020 – 25/10/2020 (14 HOURS)

FYP Lecture and Seminar:

Had a lecture on Research methods by Chris Hughes. It was a really good lecture and he recommended that we start our literature reviews early to make sure that we did the correct research before getting into building the application. In my seminar, I reported to them what I was doing this week, which was focusing on my project proposal, and asked them some questions which I got answers to. I was told that I will not need to add the optional objectives in the Gantt Chart and that I will need to finish the ethical review and risk assessment by Monday.

Activities:

This whole week was focused on finalizing my project idea and making sure the project proposal is written and complete. I was fortunate enough to send my work to my supervisors and they gave me some amazing feedback. The biggest issues were that I was constraining myself by saying that I will build my application in Android Studio and was suggested to investigate different technologies. I was also told that my objectives looked like functional requirements, so I changed them so that it looks more like actual objectives. For the formatting of my proposal, I used the marking scheme excel sheet that was on Blackboard. Each question that needed answering was turned into a header and discussed in enough detail. I create several versions of the proposal with different formats till I came to my final version. The Gantt Chart was created using an online tool: https://www.teamgantt.com/.

I have also set up the university server and will be testing out the database and file manager in the upcoming weeks. I will also create other servers such as creating an account in AWS and using a Raspberry Pi alongside Docker to create a server. That will allow me to compare the different servers to see which one will be best suited for me and my project.

TODO:

- Finish off the Ethical Review and Risk Assessment this week and submit it.
- Look into the different technologies that I could use for my application.
- Start looking into literature papers such as ACM and IEEE (University Library is a great place to go to).
- Test out the university server and create an account with AWS and set up the Raspberry Pi.
- Need to do some background research as to what already exists out there.
- Start looking at the Gantt Chart and follow that plan. Make a Trello or Monday organize tasks.

WEEK 05: 26/10/2020 – 01/11/2020 (07 HOURS)

FYP Lecture and Seminar:

Had a lecture on Research Methodology by Chris Hughes. This was a useful lecture as it gave me an idea of how to structure my dissertation. There are several examples of how to connect my questions with my objectives and my aim. I was not able to make it to this week's seminar due to personal reasons, but I have emailed my supervisors on my progress and what I did that week. I also got feedback from them on the size and number of focus groups to hold which I will be following. I was advised to do 1 focus group and keep it in a reasonable size (I will go with 3-6 people).

Activities:

As per my Gantt Chart, this week was spent on learning how to hold a focus group. A really good website that gave me some useful information about how to conduct an online focus group was: https://uxalliance.medium.com/conducting-remote-online-focus-groups-in-times-of-covid-19-ee1c66644fdb. Their recommendation was to keep the time and participants small and to use the right tools to get the most out of them. I will be using Teams as it has a video option along with the feature for people to share their screens and interacting with other people's screens. I am also in the middle of writing a document that participants will need to sign before joining. A good template that I found which I am using as a reference and creating my form is found here: https://www.ed.ac.uk/files/imports/fileManager/Focus_Group_consent.pdf. Other useful links that I used for planning the focus groups are:

- https://www.citizensadvice.org.uk/Global/CitizensAdvice/Equalities/How%20to%20run%20focus%20groups%20guide.pdf
- https://www.nyla.org/max/userfiles/Documents/D._Focus_Group_Toolkit.pdf

TODO:

- Ask 4 people to take part in my focus group that will take place next week.
- Finish off the documentation and send it to the people that agree to take part in it.
- Hold the focus group and gather requirements then categorise them in terms of priorities.
- Carry on looking into different technologies that I could use for my application.

WEEK 06: 02/11/2020 – 08/11/2020 (23 HOURS)

FYP Lecture and Seminar:

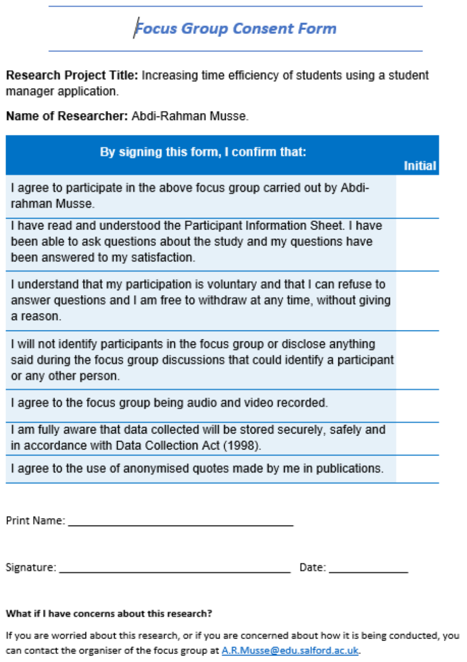
Had a lecture on Writing the Introduction and Literature Review by Julian Bass. This was beneficial to me as it gave me a good idea on how to structure my second deliverable which is the introduction, Literature Review, and Methodology. Since my second deliverable is coming soon, I will start writing my introduction on the weekend and use the tips mentioned on this lecture notes to format it well. In my seminars, I talked about doing the focus group and the steps I have taken to prepare for it. Ian Drum shared a very useful video on how to manage my Trello account to make it work with the agile methodology and I will be using that to make my Trello journey more efficient.

Activities:

I was able to finalise my focus group consent form and found 4 participants for my focus group. I have emailed them the form and managed to hold the focus group. I wrote down some questions and topics to discuss during the meeting. I also continued my research on different technology to use and I am comparing the 3 different environments which are: Android Studio, React Native, and Xamarin. From my research so far, it is very likely that the two platforms that I will pick from will be React Native and Xamarin. I might create an MVP using those two platforms over the weekend to give me a better idea of which technology to pick.

Focus group:

I gathered four students to take part in my focus group. Before the start of the focus group, I send them the consent form to sign and give back, along with a participant information sheet. Once they read that and they were happy with that, we held the focus group. Below is a picture of both the consent form and the participant information sheet.



Focus Group Consent Form

Research Project Title: Increasing time efficiency of students using a student manager application.

Name of Researcher: Abdi-Rahman Musse.

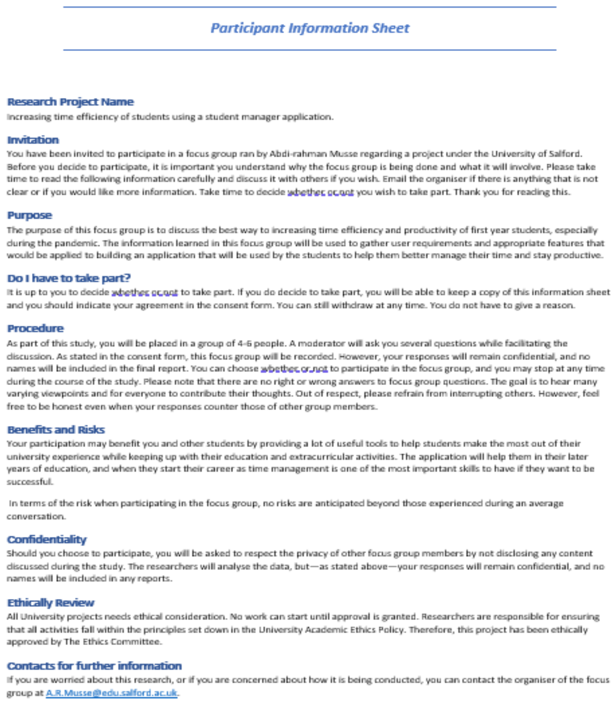
By signing this form, I confirm that:	Initial
I agree to participate in the above focus group carried out by Abdi-rahman Musse.	
I have read and understood the Participant Information Sheet. I have been able to ask questions about the study and my questions have been answered to my satisfaction.	
I understand that my participation is voluntary and that I can refuse to answer questions and I am free to withdraw at any time, without giving a reason.	
I will not identify participants in the focus group or disclose anything said during the focus group discussions that could identify a participant or any other person.	
I agree to the focus group being audio and video recorded.	
I am fully aware that data collected will be stored securely, safely and in accordance with Data Collection Act (1998).	
I agree to the use of anonymised quotes made by me in publications.	

Print Name: ______________________________

Signature: ______________________________ Date: ____________

What if I have concerns about this research?

If you are worried about this research, or if you are concerned about how it is being conducted, you can contact the organiser of the focus group at A.R.Musse@edu.salford.ac.uk.



Participant Information Sheet

Research Project Name

Increasing time efficiency of students using a student manager application.

Invitation

You have been invited to participate in a focus group ran by Abdi-rahman Musse regarding a project under the University of Salford. Before you decide to participate, it is important you understand why the focus group is being done and what it will involve. Please take time to read the following information carefully and discuss it with others if you wish. Email the organiser if there is anything that is not clear or if you would like more information. Take time to decide whether or not you wish to take part. Thank you for reading this.

Purpose

The purpose of this focus group is to discuss the best way to increasing time efficiency and productivity of first year students, especially during the pandemic. The information learned in this focus group will be used to gather user requirements and appropriate features that would be applied to building an application that will be used by the students to help them better manage their time and stay productive.

Do I have to take part?

It is up to you to decide whether or not to take part. If you do decide to take part, you will be able to keep a copy of this information sheet and you should indicate your agreement in the consent form. You can still withdraw at any time. You do not have to give a reason.

Procedure

As part of this study, you will be placed in a group of 4-6 people. A moderator will ask you several questions while facilitating the discussion. As stated in the consent form, this focus group will be recorded. However, your responses will remain confidential, and no names will be included in the final report. You can choose whether or not to participate in the focus group, and you may stop at any time during the course of the study. Please note that there are no right or wrong answers to focus group questions. The goal is to hear many varying viewpoints and for everyone to contribute their thoughts. Out of respect, please refrain from interrupting others. However, feel free to be honest even when your responses counter those of other group members.

Benefits and Risks

Your participation may benefit you and other students by providing a lot of useful tools to help students make the most out of their university experience while keeping up with their education and extracurricular activities. The application will help them in their later years of education, and when they start their career as time management is one of the most important skills to have if they want to be successful.

In terms of the risk when participating in the focus group, no risks are anticipated beyond those experienced during an average conversation.

Confidentiality

Should you choose to participate, you will be asked to respect the privacy of other focus group members by not disclosing any content discussed during the study. The researchers will analyse the data, but—as stated above—your responses will remain confidential, and no names will be included in any reports.

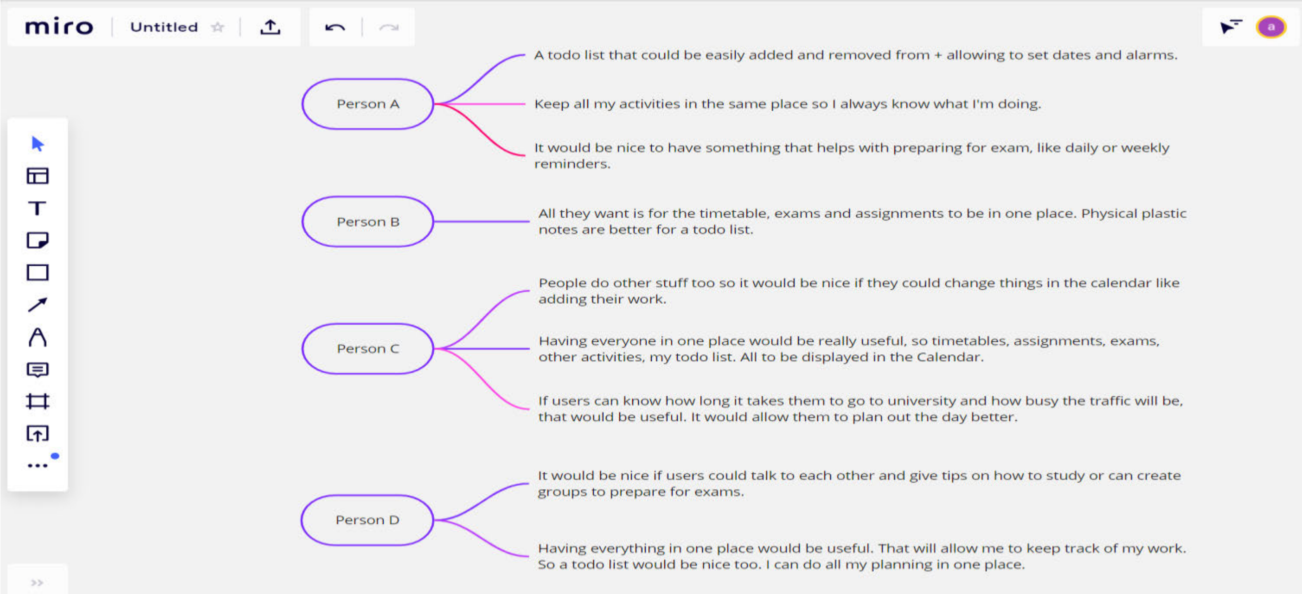
Ethically Review

All University projects needs ethical consideration. No work can start until approval is granted. Researchers are responsible for ensuring that all activities fall within the principles set down in the University Academic Ethics Policy. Therefore, this project has been ethically approved by The Ethics Committee.

Contacts for further information

If you are worried about this research, or if you are concerned about how it is being conducted, you can contact the organiser of the focus group at A.R.Musse@edu.salford.ac.uk.

The structure of the focus group was very relaxed. It started with an introduction and setting the ground rules. Following that, I asked a question and gave the participants a few minutes to think about the answer. The question was: *"think about your experience during your first year and how you managed your time. What difficulties did you face and how did you deal with them?"*. Once the participants shared their experience, it allowed me to ask the follow-up question which was *"what features would have made your life easier during the first year when it comes to time management and productivity?"*. From there I was able to have a discussion with the participants about the different features they would like to see, and which ones they wanted the most. This allowed me to priorities the features. https://miro.com/app/dashboard/ was used to write down what everyone thought and the priorities of the features. The call took place in Discord as it allowed for multiple callers with good quality and it was the most accessible tool for all the participants. Below is an image showing the points that the participants made.



Person A

- A todo list that could be easily added and removed from + allowing to set dates and alarms.
- Keep all my activities in the same place so I always know what I'm doing.
- It would be nice to have something that helps with preparing for exam, like daily or weekly reminders.

Person B

- All they want is for the timetable, exams and assignments to be in one place. Physical plastic notes are better for a todo list.

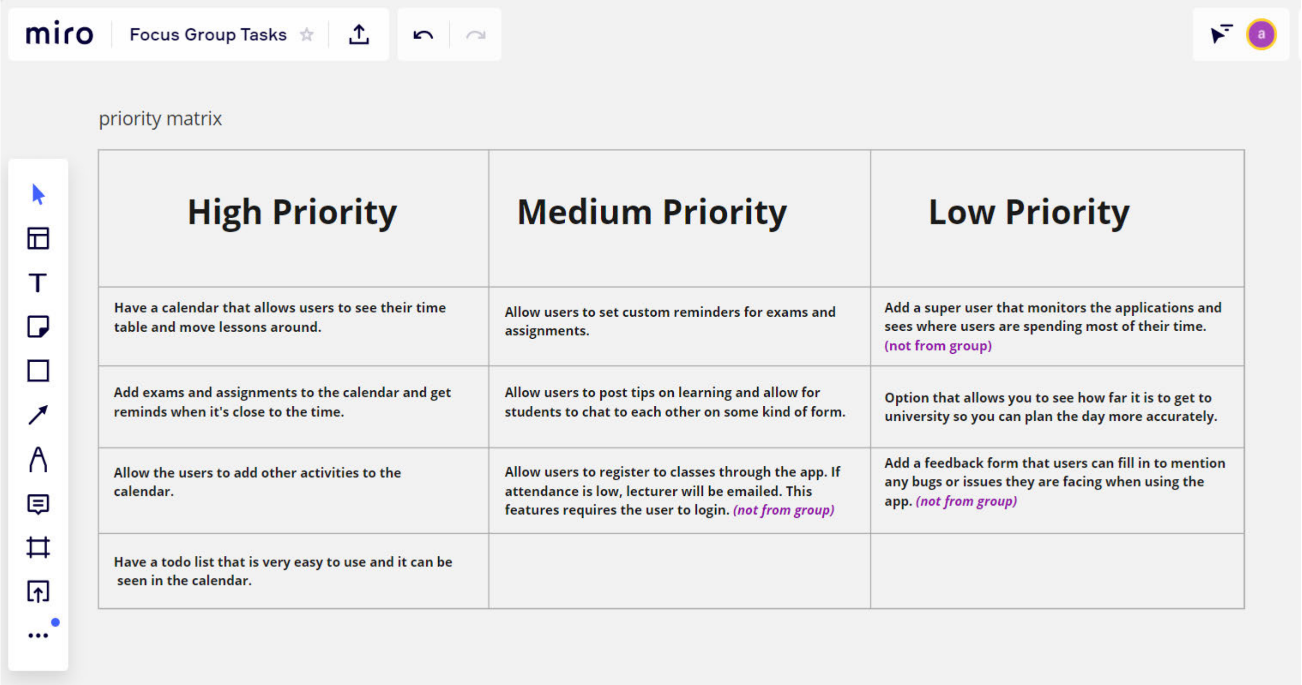
Person C

- People do other stuff too so it would be nice if they could change things in the calendar like adding their work.
- Having everyone in one place would be really useful, so timetables, assignments, exams, other activities, my todo list. All to be displayed in the Calendar.
- If users can know how long it takes them to go to university and how busy the traffic will be, that would be useful. It would allow them to plan out the day better.

Person D

- It would be nice if users could talk to each other and give tips on how to study or can create groups to prepare for exams.
- Having everything in one place would be useful. That will allow me to keep track of my work. So a todo list would be nice too. I can do all my planning in one place.

Once the discussion has ended. I wrapped the session up by asking them to look back at the notes and to create distinct features. Once they identified features, I asked them to rank them in terms of priority. The image below shows exactly that. I have also added a few features of my own and those have been differentiated from the others as seen in the image below.



priority matrix

High Priority	Medium Priority	Low Priority
Have a calendar that allows users to see their time table and move lessons around.	Allow users to set custom reminders for exams and assignments.	Add a super user that monitors the applications and sees where users are spending most of their time. (not from group)
Add exams and assignments to the calendar and get reminds when it's close to the time.	Allow users to post tips on learning and allow for students to chat to each other on some kind of form.	Option that allows you to see how far it is to get to university so you can plan the day more accurately.
Allow the users to add other activities to the calendar.	Allow users to register to classes through the app. If attendance is low, lecturer will be emailed. This features requires the user to login. *(not from group)*	Add a feedback form that users can fill in to mention any bugs or issues they are facing when using the app. *(not from group)*
Have a todo list that is very easy to use and it can be seen in the calendar.		

Overall, running the focus group was an amazing experience and I learned a very useful skill which was gathering user requirements. I made a few mistakes during the process such as mentioning a feature that I had in mind rather than letting them come up with what they thought was best. However, in the end, I believe that this was a great success. The features that need to be built have been collected so it is now time to move on to the design stage. I will try to use a tool to quickly design the layout of the pages and have another focus group with the same people to see what they think of the designs.

TODO:

- Need to finalise my research on what technology to use. Maybe create an MVP on the 2 remaining platforms and that will help me make my decision.
- Do a basic layout design from the conclusion that was made in the focus group.
- Carry on looking at literature papers such as ACM and IEEE for the literature review.
- Test out the university server and create an account with AWS and set up the Raspberry Pi.

WEEK 07: 09/11/2020 – 15/11/2020 (21 HOURS)

FYP Lecture and Seminar:

The lecture for week 7 has been cancelled and rescheduled to week 9 so nothing happened during the Wednesday 9 am session. I took that time to carry on working on my second deliverable which is due in December. In the seminar, I talked about getting closer to the decision on which technology to use and that I will be making it at the end of the week. I also talked about building a small application on each platform to test out how comfortable I would be working in them.

Activities:

This week was dedicated to deciding what technology to use. I attempted to build a small application on four different platforms (2 native and 2 cross-platformed) which were Android Studio, XCode, Xamarin, and React Native. I was successful at Android Studio, Xamarin, and React Native but could not do anything with XCode since it needed a macOS and I did not have one. I attempted to make a VM that runs macOS but that did not work and was a failure. Therefore, I was not able to test the iOS side of things yet. I will attempt to use an iPhone from someone I know or will find a cheap second-hand one to buy.

Android Studio was really good and easy to understand but it restricts me in terms of the people I reach since it can only be deployed to android phones. Xamarin was complicated to understand and it was not easy to run the application. It was also hard to understand the structure, plus the code was not straightforward. React Native was hard to set up but once the setup was complete, it was very easy to run the code and test it out. The codebase was easy to understand and I had some prior knowledge of JavaScript which made it easier. Due to those reasons, I decided to go with React Native.

I have also completed a section in the methodology for the 2nd deliverable which is talking about what technologies I looked at. I will need to carry on with that and complete the other sections.

TODO:

- Need to find a way to test out the application to see how it looks on iOS devices.
- Complete the methodology section for the 2nd deliverable.
- Start typing up the literature review.
- Do a basic layout design from the conclusion that was made in the focus group.
- Test out the university server, create an account with AWS, and set up a Raspberry Pi.

WEEK 08: 16/11/2020 – 22/11/2020 (16 HOURS)

FYP Lecture and Seminar:

Had a lecture on software architecture by Ian Drumm. This was useful as it talked about lots of different approaches that I can take to build my application. It also gave me a good insight into the different levels of design, from a very high-level system overview to looking into low-level architecture such as OOP and MVC. Ian also talked about a very important architecture style which was Representational State Transfer (REST). This fits perfectly into something that I want to make for the project, so it helped me get a better understanding of how to achieve it and what to search for. In my seminar, I talked about what I did this week, which was mainly learning how React Native works and cleaning up the methodology section for the 2nd deliverable. I also mentioned that I found an iPhone to test against. Ian reminded me that the university will provide some platform or service that will be used in the second year which will allow me to test my application on iOS devices.

Activities:

This week was heavily focused on learning the React and React Native Framework. I watched lots of different videos on Pluralsight to get me up to speed with how it works and teach me the basics. The videos I watched are:

- https://app.pluralsight.com/library/courses/react-big-picture/table-of-contents
- https://app.pluralsight.com/library/courses/react-native-big-picture/table-of-contents
- https://app.pluralsight.com/library/courses/react-js-getting-started/table-of-contents
- https://app.pluralsight.com/library/courses/react-native-getting-started/table-of-contents
- https://app.pluralsight.com/library/courses/react-fundamentals-update/table-of-contents
- https://app.pluralsight.com/library/courses/react-native-fundamentals/table-of-contents

It was easy to understand most of the concepts and it is nice that React is structured in such a way that most of the syntax is basic JavaScript and does not introduce too many new syntaxes. I also built a few applications while following watching videos and I was happy with them. Once I was comfortable with React and React Native, I moved on to trying out some stuff on my own. The next big challenge was how to design a multiple-page application since all the videos I have watched were single pages. I watched a few YouTube videos to get me up to speed with that and I was surprised that I had to download packages to allow me to navigate between screens. The videos I have watched are links I have checked out are:

- https://docs.expo.io/guides/routing-and-navigation/
- https://reactnative.dev/docs/navigation
- https://www.youtube.com/watch?v=nQVCkqvU1uE
- https://www.youtube.com/watch?v=a9jSyZXYGn8&list=PLQWFhX-gwJbmmqcP-9zMXBaxQbGKfIJY2
- https://www.youtube.com/watch?v=Hln37dE19bs

It took a lot longer than I expected to build a basic application that handles navigation, but it was complete and working. The final thing I looked at is some calendar libraries that I can use since that is the primary idea of my application. I need a calendar that I can play with and edit as I wish. There were some promising finds, and I will be doing more research and trying to implement them in the upcoming weeks. The links to the libraries are:

- https://github.com/wix/react-native-calendars
- https://www.npmjs.com/package/react-native-calendars

I managed to complete my 2nd draft of the methodology section and I am happy with it. I will be reviewing it again at a later date, but the next thing I need to do is work on is the literature review. I already have some papers that I have gathered in the previous weeks so I will put all that together and start typing something soon.

TODO:

- Do a basic layout design from the conclusion that was made in the focus group and from the background research on existing apps.
- Test out the university server, create an account with AWS, and set up a Raspberry Pi.
- Create the database(s) and set up the needed tables and structure.

WEEK 09: 23/11/2020 – 29/11/2020 (29 HOURS)

FYP Lecture and Seminar:

Had a lecture on requirement engineering by Lee Griffiths. This was an interesting topic as it showed me how to gather requirements for my project. This is something that I have already done but it was good to see that I was on the right path and that the approach that I took was correct. For example, lee mentioned the requirement matrix which breaks up the requirements into different categories. Even though mine was a little different from his, it still showed that it will make completing the tasks a lot easier since I know which ones to start with. Lee also talked about design and emphasizes the importance of that. I will need to create some design (at the very least do paper designs in the next week) so that I have a high-level picture of how the whole application will look like. Lee also gave some tips on what to talk about in the requirements section of the deliverables which I will be following. In my seminar, I talked about the different servers that I investigated and which one I picked.

Activities:

This week was spent on setting up the different servers and looking into the best one to use for my application. The goal for the server is:

- it can host a MySQL database (will be using the MySQL Workbench application to test out the connection and write my queries in).
- it can also host the PHP scripts that allow me to access the database (there was a good tutorial on how to create the PHP scripts on https://www.codexworld.com/connect-access-remote-mysql-database-cpanel-php/).
- it allows me to host my java web scrapping application since I will need to run this periodically without me interacting with it.

The first server that I set up was the university server https://poseidon.salford.ac.uk/. I was able to set up my database and connect it to MySQL Workbench. However, I struggled to connect my PHP scripts to MySQL and running them. After struggling for hours and emailing the Poseidon team. It turns out that some firewall issues were stopping me from doing that. After talking to the university team that looked after Poseidon, I was told that I couldn't change the firewall permissions. This meant that I could not carry on with Poseidon and hit a wall.

To make sure that I wasn't being silly and making a mistake, I created an account with a different free server provider called https://www.000webhost.com/. I did the same steps that I did in Poseidon (setting up the database, testing the connection in MySQL Workbench, writing the PHP scripts) and when I tested out the scripts, they were able to connect to the database and display some outputs. This finalized my decision that Poseidon could not be used. However, I also did not want to use 000webhost since they delete your account if I am not active for a month. They also did not offer features that allow me to host my java web scrapping application.

I took a different approach where I tried to host my server using a spare laptop. I used docker to get the applications that I need and host my database there (https://www.portainer.io/). The setup was complicated and there were many issues, but I managed to set up my database and connect to it using MySQL Workbench. However, I got stuck on finding reasonable images that I could use to host my PHP scripts and gave up at that point. There was also an issue where the server would randomly go to sleep which meant that my connection to the database would go off. This was a massive reason to not use this approach since it would create an extra layer of complexity and it could easily crash on me during development or in a demo and it could take a long time to fix it.

The final server that I looked at was AWS https://aws.amazon.com/. I was very impressed with it. I was able to easily set up my database and connect it to MySQL Workbench. There was a very useful video that helped me with this which is: https://www.youtube.com/watch?v=Ng_zi11N4_c. It also seems that I can host my PHP scripts by using EC2 or Amplify but I will need to do more research into that. It also seems that I can host my java web scrapping applications on it using AWS lambda. I will spend then next week or two looking into those different options and making sure that I can use it and that it works accordingly.

From the research that I have done, it seems that AWS will be the best option to pick. However, I need to make sure that I test out the AWS Lambda service and the EC2/Amplify service to make sure that I can host my APIs.

I was also able to collect some useful papers for the literature review and I started writing on them, I am planning on finishing the first draft by the end of next week. This will give me plenty of time to write my introduction and to make any changes that I need to make without stress or worrying about time constraints.

TODO:

- Look into existing applications and compare their features to what I want to do, and how they could benefit me.
- Do a basic layout design from the conclusion that was made in the focus group and from the background research on existing apps.
- Look into different AWS services to see if it fits my needs.
- Extend my web scrapping application so that it works completely, and it is standalone and robust enough to work on different timetables.

WEEK 10: 30/11/2020 – 06/12/2020 (31 HOURS)

FYP Lecture and Seminar:

Final Year Project lectures have ended for semester 1 last week. During the seminar, I talked about what I was busy with this week which was improving on the web scrapping application and testing it out. I mentioned that I did not look more into the AWS services and will be looking into it next week. I also said that I am planning to have an MVP ready for the week 12 meeting.

Activities:

This week was spent mostly on improving my web scrapping application and making sure that works as I need it to. Initially, I used the Gradle Framework but since I will not be building the application on Android Studio, I changed my Framework and moved it to Maven. It was not hard to find the same decency on Maven, so I used the “htmlunit” maven dependency. The link to the maven dependency is https://mvnrepository.com/artifact/net.sourceforge.htmlunit/htmlunit/2.45.0. The goal was to make this application as robust and flexible as possible so that it can adapt to any changes made to FYM.

There were many hurdles along the way and I can across many issues. One of the biggest ones being that some timetables have different structures so I needed to accommodate for those and this pushed me to make sure that my application can adapt to changes within the HTLM Dom without me changing the code.

The web scraping is complete. The application requires a URL, and it will return a map that contains the timetable. Even though the application is designed for the timetable for first-year computer science students, it accepts most of the timetables on the First Year Matters timetable. The next task is to host the application on AWS and find a way to run it periodically.

The literature review has also been complete where I talked about the relationship between mental health, student performance, time management, and grades. I also found different applications that are out there that provide time management tools for students. I have talked about the pros and cons of each and how they will benefit me. It also gave me a better idea of the designs that I want to have, and it will make it easier to draw my screens – which needs to be done next week!! The 2nd deliverable is nearly complete. All that is left is to finish off the introduction and then go over it one more time to make any final changes.

TODO:

- Do a basic layout design from the conclusion that was made in the focus group and from the background research on existing apps.
- Look into different AWS services to see if it fits my needs.
- Finish the 2nd deliverable.
- Start working on my MVP and create a calendar page that connects to the database and displays the timetable.

WEEK 11: 07/12/2020 – 13/12/2020 (17 HOURS)

FYP Lecture and Seminar:

Final Year Project lectures have ended for semester 1. During the seminar, I talked about looking into AWS and seeing how it works along with learning about its different services. The primary ones that I said I will be focusing on are Lambda and EC2. During the meeting, we were also told that we can do demos when we come back for semester 2, which will be in February. Therefore, I got more time to work on my MVP.

Activities:

Most of this week was spent researching AWS and understanding how it works. My focus was to find a service that will host my web scraping application and call it periodically. The video on https://www.youtube.com/watch?v=vyLvmPkQZkI helped me understand how to set up a Lambda service and use python scripts (which I might use for the API calls to get data from the database). I also watched another video that had a similar approach but for java, in which I would be able to host my web scraping application – https://www.youtube.com/watch?v=JeJ46YlpPqw. I made an account on there and tried to see how the lambda service worked but it didn't work as I expected. The problem was that I did not know how to call the lambda whenever I wanted, and I did not want to have to run the test every time I wanted to do the tasks. I will be looking at this, along with other approaches, next week and over the holidays.

If the lambda does not work, I might move over to Kubernetes with AWS and see if that can host my java application and run it periodically. If all else fails, I will use Volley (from maven) and use HTTP POST/GET requests to call some APIs and to store the information in the database. Some examples that could help me are https://www.itsalif.info/content/android-volley-tutorial-http-get-post-put and https://mvnrepository.com/artifact/com.android.volley/volley/1.1.1.

During this week, I also thought about my MVP and how to present it. However, I am still not sure what approach to take. I might show my web scraping application and the output it puts out, which is just some text. However, if it has been pushed back till after the holidays, then I can have some of the react-native application to show which I will attempt to build during the holidays.

TODO:

- Do a basic layout design from the conclusion that was made in the focus group and from the background research on existing apps.
- Try to get the Lambda service working on AWS or attempt Kubernetes or Volley approach.
- Start working on my MVP and create a calendar page that connects to the database and displays the timetable.

WEEK 12: 14/12/2020 – 20/12/2020 (14 HOURS)

FYP Lecture and Seminar:

Both the Final Year Project lectures and the seminars have ended for semester 1.

Activities:

Since the 2nd deliverable was due this week, that is all that I worked on to ensure I do well on it.

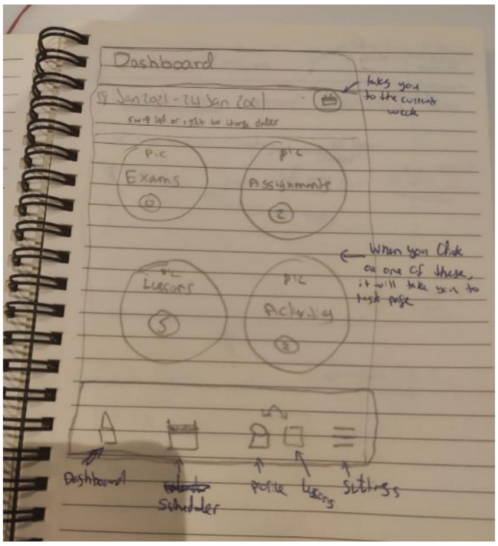
TODO:

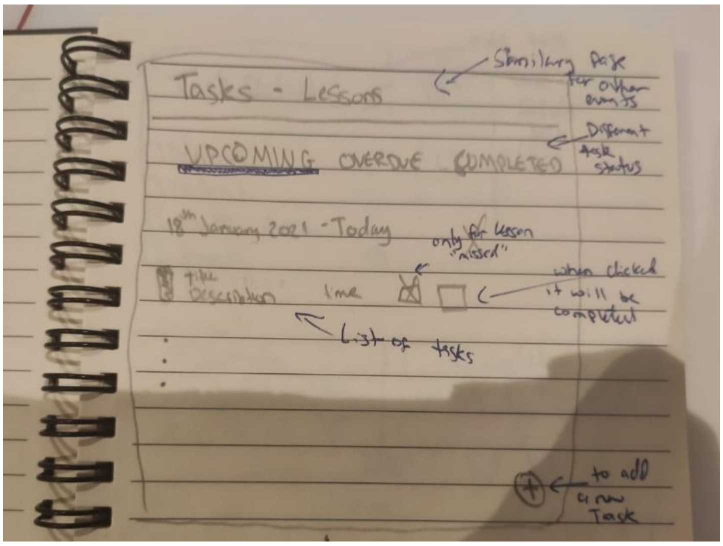
- Do a basic layout design from the conclusion that was made in the focus group and from the background research on existing apps.
- Try to get the Lambda service working on AWS or attempt Kubernetes or Volley approach.
- Start working on my MVP and create a calendar page that connects to the database and displays the timetable.

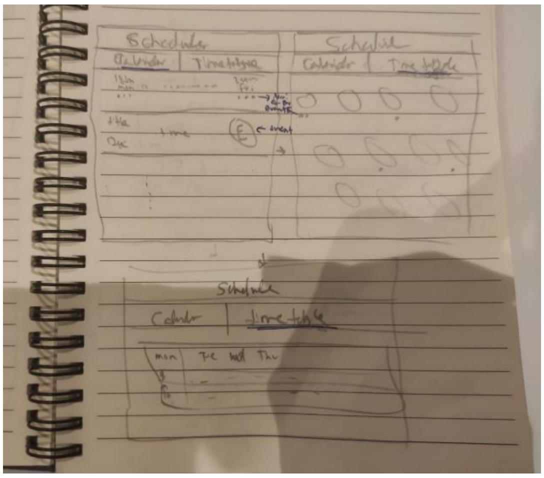
OVER THE CHRISTMAS HOLIDAYS

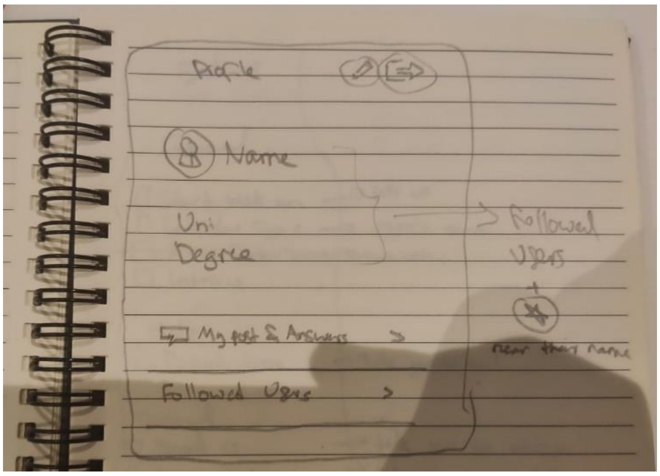
Activities:

Over the holidays, the first task I tackled was the design of my application. I had a look at the applications that I have mentioned in my 2nd deliverable and looked at what made them stand out. I noted down the good and bad from each one. This then gave me a rough idea of how I want my application to look. I started roughly drawing out, on paper, the different pages. After several drawing attempts and keeping in mind accessibility and usability, I believe I got a reasonably well design. The drawing is my first version of my UI design. This will most likely change slightly when I start building the application. The images below show the different pages of the application.


Dashboard
18 Jan 2021 - 24 Jan 2021
takes you to the current week
swipe left or right to change dates
Pic Exams
Pic Assignments
Pic Lessons
Pic Activity
When you click on one of these, it will take you to task page
Dashboard
Scheduler
Profile
Lessons
Settings







The second task that I did was learning more about React Native by watching tutorials on Pluralsight to better understand the framework and remember the JavaScript language. This was and will continue to be an ongoing thing for a long while, so it is difficult to evaluate whether I am successful in this task or not. I will see how much I learned when I start building my application. During the learning stage, I did build small stand-alone applications that did a variety of tasks. These applications were a follow-along of the videos I have watched.

The third major task that I completed this holiday was the setup of my back-end server. I decided to take a break from figuring out how to host my web scraping applications since it was taking too much time. I decided to instead focus on setting up my database and a place to host my API scripts that will be called by my application. I created an RDBS in AWS which worked fine. I was able to connect to my database from an application called MySQL Workbench and add tables and rows. The next challenge was to find a way to host my API scripts. After some research, I came to the conclusion that the best AWS service to use would be an EC2 bucket. I created an Ubuntu instance in AWS and installed apache, PHP, and MySQL server. This would allow me to host my database in the ubuntu instance, but I was finding it difficult to display my database to MySQL workbench. Therefore, I decided to stick to the AWS RDBS and host the API calls in the ubuntu instance.

For a quick test, I created a test table and populated it with some data, created a quick PHP script that gets all the data from that table, created a React Native application, and then called the fetch method in JavaScript to call the API and display the information. This worked beautifully and I was happy with that. The problem that I encountered was sending some information to the API through using POST. After many hours of research, I managed to send information to the database and insert a new row.

The penultimate task that I did over the holidays was to create an empty react native application and set up my navigation based on the design that did. I used the official document to get a general overall understanding of how it works (https://reactnavigation.org/docs/getting-started). Once I understood the basics, Medium gave me the next step https://medium.com/wesionary-team/combining-stack-navigator-with-tab-navigator-in-react-native-react-navigation-253656f45181. This gave me a more detailed example that combined multiple types of navigation. The YouTube videos https://www.youtube.com/watch?v=nQVCkqvU1uE helped me understand the whole process a lot more. I was able to combine Stack Navigation, Top Tab Navigation, and Bottom Tab Navigation.

The final major task that I did over the holiday was to look at some libraries that provide me with a calendar view to display the dates along with the events that the student has. I have found two very useful libraries which are: https://github.com/wix/react-native-calendars and https://github.com/stephy/CalendarPicker. After playing with both libraries, I decided to go with the Wix one since it had pretty much everything I wanted, and it was easier to use and looked better. It also had the agenda view which was useful as it showed me the days and all the tasks that need to be done for that day. I have some dummy data that I am adding to the calendar for events. The next step for this would be to populate the database and use real data.

Problems:

The first issue that I came across was that lambda was not working for me the way I expected it. I wanted to have something that can host my web scraping application and will run it for me periodically without me having to do anything. However, that is not what lambda does and therefore I had to look for other services such as Kubernetes. I will be looking into it in the upcoming weeks.

Another minor problem was that I could not connect to phpMyAdmin since I could not figure out the right information to connect to it. Therefore, I decided to use the RDBS that AWS provides.

The final problem that I had a lot was to do with React Native. I was having a difficult time converting class components to functional components and vice versa. I was also struggling to understand what props are and how they are used. One other thing that I still do not fully understand is knowing what parameters are available to use. For example, you can use navigation and route by passing it as a parameter in the component, but only if it is mentioned in the navigation component. It will take some time for me to fully understand it all, but I am slowly getting there.

TODO:

- Connect the database to the application to represent the right information in the calendar.
- Start looking for libraries that can be used for the timetable.
- Carrying on with the MVP build.

SEMESTER 2 (PART 1)

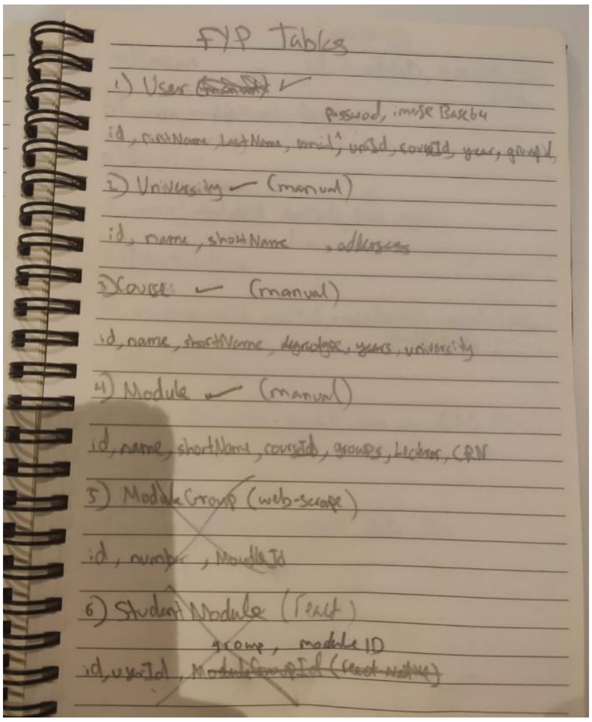
WEEK 01: 01/02/2021 – 07/02/2021 (8 HOURS)

FYP Lecture and Seminar:

There was no lecture in week 1. During the seminar, I talked about what I did over the holidays and the progress that I made. It was mentioned in the meeting that we would be getting our results for the 2nd deliverable next week. This will help get some useful feedback to prepare for the final deliverable. We were also told when the demo would be taking place and it was agreed upon that it will be in week 4.

Activities:

This week was focused on two things. Planning out what database tables are needed for the application and looking for good libraries to use for the timetable. I drew out all the tables I think I will need and planned out the columns it will have. Below is a picture that shows the tables that I think I will need. I also found one library for the timetable (https://github.com/gomjellie/react-native-timetable#readme), but it is not maintained so I will carry on looking for something else.



FYP Tables

1) User ~~(manual)~~ ✓

id, firstName, lastName, email, password, image Base64, uniId, courseId, year, groupId,

2) University ✓ (manual)

id, name, shortName, address

3) Course ✓ (manual)

id, name, shortName, degreetype, years, university

4) Module ✓ (manual)

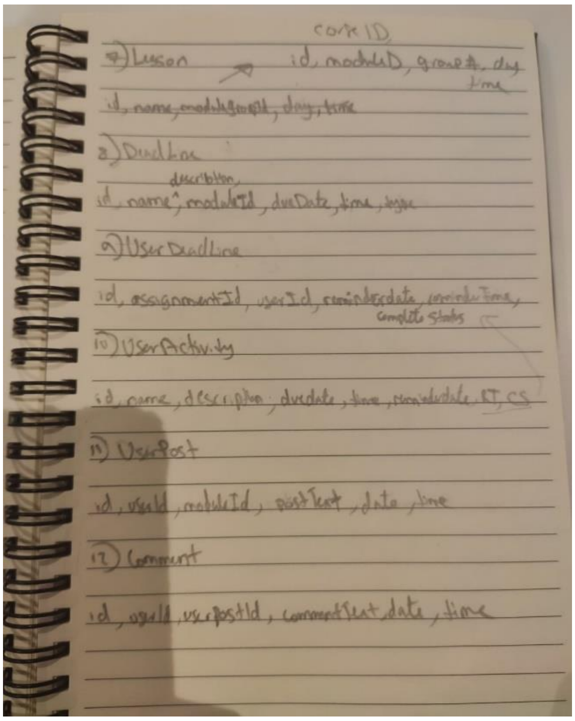
id, name, shortName, courseId, groups, Lecturer, CRN

~~5) ModuleGroup (web-scrape)~~

~~id, number, ModuleId~~

6) Student Module (React)

id, userId, group, module ID ~~ModuleGroupId (react-native)~~



7) Lesson id, moduleID, group #, day, time, courseID

id, name, moduleGroupId, day, time

8) DeadLine

id, name, description, moduleId, dueDate, time, type

9) UserDeadLine

id, assignmentId, userId, reminderdate, reminderTime, complete Status

10) UserActivity

id, name, description, duedate, time, reminderdate, RT, CS

11) UserPost

id, userId, moduleId, postText, date, time

12) Comment

id, userId, userpostId, commentText, date, time

Problems:

No new problems.

TODO:

- Connect the database to the application to represent the right information in the calendar.
- Carry on looking for libraries that can be used for the timetable.
- Carrying on with the MVP build.

WEEK 02: 08/02/2021 – 14/02/2021 (23 HOURS)

FYP Lecture and Seminar:

There was no lecture in week 2. During the seminar, I talked about finding a timetable library which I will be implementing. This is the same library that I found last week. Since I couldn't find a better one, I decided to try using this one. I left the task of connecting the database for later – after having both the timetable and calendar view fully implemented which will be in either week 3 or 4. We were reminded that the demos were going to be on week 4.

Activities:

I could not find any other library to use for the timetable apart from the one from last week and I did not want to do it from scratch. Therefore, I decided to use the library that I found last week. I managed to implement the timetable using the library and input some fake data. However, since the library was not being maintained, it was giving me incorrect outputs and displaying the wrong information. Therefore, I will need to edit the library files next week.

Problems:

The timetable library is out of date and inaccurate. Either fix the files or create my custom timetable.

TODO:

- Connect the database to the application to represent the right information in the calendar.
- Find a solution to the issue with the timetable library.
- Carrying on with the MVP build.

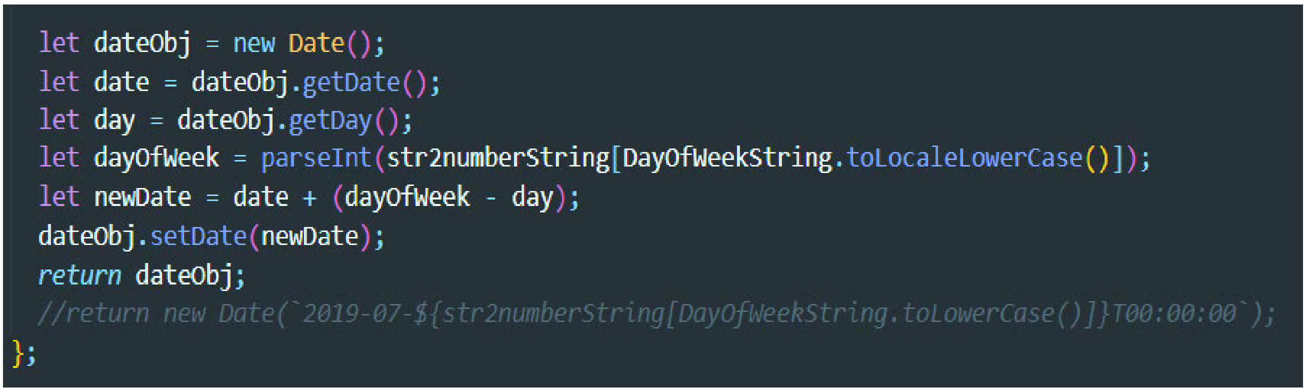
WEEK 03: 15/02/2021 – 21/02/2021 (29 HOURS)

FYP Lecture and Seminar:

Had a lecture with Lee covering the experimental project and how to approach it. Since my project is building a product, I was not the target. However, it covered some important topics such as testing and how to approach it. I will be using that in the evaluation phase of my project. During the seminar, I talked about what I did and the problems I was facing. I told my supervisors about the difficulties I was having with displaying information that I got from the database due to the asynchronous nature of the fetch method.

Activities:

This week was focused on fixing the timetable library. I went into the files that were stored in the modules area and found the functions that decided what is represented. I noticed that the dates were hardcoded to July 2019 and I changed it to be dynamic. Now it will always represent the correct information. The front end for it is finished so I will need to populate and get the data from the database to represent the right information. Below is an image of the file that was edited and the changes that I made.



```
  let dateObj = new Date();
  let date = dateObj.getDate();
  let day = dateObj.getDay();
  let dayOfWeek = parseInt(str2numberString[DayOfWeekString.toLocaleLowerCase()]);
  let newDate = date + (dayOfWeek - day);
  dateObj.setDate(newDate);
  return dateObj;
  //return new Date(`2019-07-${str2numberString[DayOfWeekString.toLowerCase()]}T00:00:00`);
};
```

I also went back to the database and populated some of it with the correct information. However, I was having issues displaying the information. After researching, I found out that the fetch method was asynchronous. Therefore, I would try to display the screen right after calling the fetch, but the data might or might not be there. To fix this issue I used a Timeout method to wait for a second before displaying the screen. That seems to have fixed the issue but it’s not ideal so I will be looking for a permanent solution.

Problems:

I need to find a good solution for the asynchronous issue in the fetch calls.

TODO:

- Finish off populating the database and representing the right information.
- Find a solution for the fetch and asynchronous issues.
- Carrying on with the MVP build.

WEEK 04: 22/02/2021 – 28/02/2021 (32 HOURS)

FYP Lecture and Seminar:

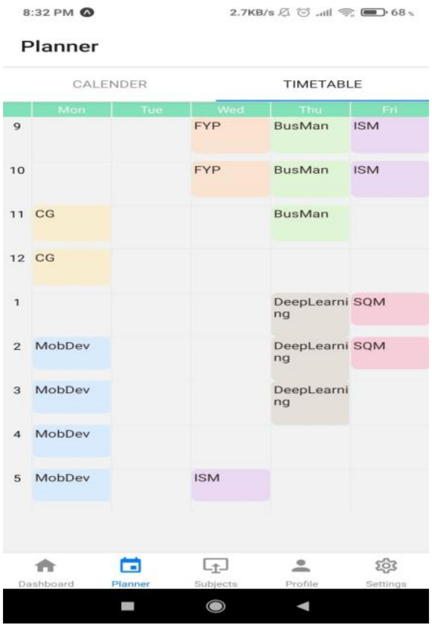
There was no lecture in week 4. This week's seminar was used as a demonstration period where each student presented what they've been doing in semester 1 and what they have produced. During my turn, I was able to show my react-native application and the features that I have developed. I showed the timetable screen and the calendar screen which the supervisors were happy with. I didn't get much feedback on how to improve the application but I already knew what I had to do so I could carry on with my development.

Activities:

The login and register screens were created before the demo. This was done to get the right information based on the user that is logged in. Since this is a data-driven project, things will look slightly different for different users.

PHP scripts were written that connected to the database and allowed me to retrieve the timetable information from the database. This was stored in the database by using the java web scraping application where it got the timetable from first-year-matters and stored it in the database. This was read into the application using Fetch statements which allows me to represent the correct information in the database. Therefore, the backend for the timetable was complete and I was able to display the timetable of the student that is logged in.

A cool feature that was developed this week was the ability to move lessons around to suit the student's needs. This was done to achieve the requirement which is having a timetable that was customised by the student. The user is now able to pick a lesson and move it to anytime they want. The image below shows how the timetable screen looks.



	Mon	Tue	Wed	Thu	Fri
9			FYP	BusMan	ISM
10			FYP	BusMan	ISM
11	CG			BusMan	
12	CG				
1				DeepLearning	SQM
2	MobDev			DeepLearning	SQM
3	MobDev			DeepLearning	
4	MobDev				
5	MobDev		ISM		

Problems:

The calendar needs to have the correct marked dates – not displayed correctly at the moment.

The tasks are not presented correctly, need to represent them in the correct format.

The moving lesson feature is great but it needs to be changed so that the lesson time is restricted to a time the lesson is on.

TODO:

- Find a solution for the fetch and asynchronous issues.
- Restrict the move of the lessons in the timetable.
- Fix the issue with the marked dates.
- Fix the issues with the task display.

WEEK 05: 01/03/2021 – 07/03/2021 (22 HOURS)

FYP Lecture and Seminar:

There was no lecture in week 5. During the seminar, I talked about the problems I fixed. I told my supervisors that I worked on connecting the backend code to the calendar screen. The UI was there and I had dummy data before, but now it's getting the data from the database. Also mentioned that I can display a detailed view of those tasks which gives more information. I also had some problems such as the asynchronous issue that I fixed.

Activities:

The week was focused on fixing lots of small bugs that were in the software and connecting the backend code to the calendar screen. The asynchronous issue was fixed by removing the wait timer, setting a loading page, checking if the data is present, then reloading the screen again and removing the hiding the load page. This ensured that the page would only load once the data is present and it will wait as long as it needs. The trigger is updating a state which causes the page to reload.

The UI for the timetable changed so that the users are now restricted to only put down dates that lessons are actually on. This caused the dateTimePicker to change to a drop-down where the students pick a day and time. There were some issues with the marked dates in where the same date could not be marked by multiple markers. This has been fixed after seeing the examples given by the official link: https://github.com/wix/react-native-calendars.

All the tables and data are populated in the backend so that it can handle the calendar screens. APIs were created in AWS to interact with the database and fetch POST and GET statements were created in the application to interact with the APIs. This worked nicely and the calendar now has real data that is stored in the database.

Problems:

- Some bug fixes in the login and register screen.

TODO:

- Need to create a page to add and edit the tasks through the application.
- Research some dashboard libraries and implement them.
- Look into biometrics in react native.

WEEK 06: 08/03/2021 – 14/03/2021 (27 HOURS)

FYP Lecture and Seminar:

This week's lecture was with Julian Bass and talked about how to write our dissertation and the best way to approach it. He went over the two flows which are:

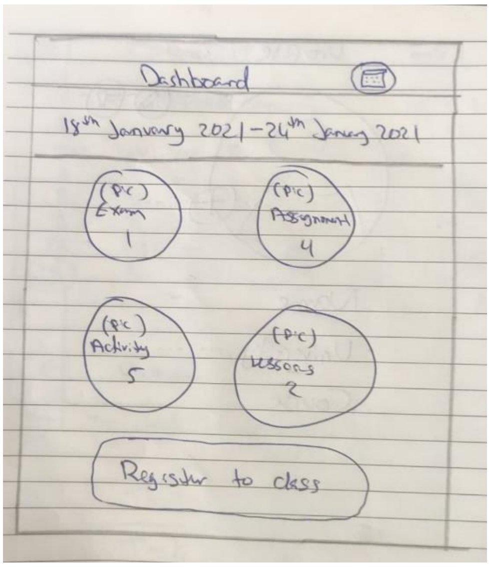
- structuring everything then filling the parts in
- Just typing everything that comes to mind and then creating headers later on

This lecture gave me an idea of how to structure my report.

During the seminar, I talked about what I did this week which was quite a lot. I worked on finding some libraries for Dashboard and implementing it, reading on biometrics and implementing it, and fixed some bugs.

Activities:

The first thing that I worked on was looking for different dashboards that I can use. The only one that I could find which was useable was the react-native-dashboard. The link to that is https://www.npmjs.com/package/react-native-dashboard. This dashboard is not the same as what I had imagined my page to look like, which can be seen in the image to the side. However, the design simplistic and looks good. Therefore, I have used the library and implanted it to have square dashboards instead of circles.



The second task was to integrate biometrics into the application to allow users to register for a class when it's time. This required a lot of research on different libraries since there are a plethora of options for expo-CLI. However, the libraries that I decided to use are as follows:

- expo-location (https://docs.expo.io/versions/latest/sdk/location/)
- geolib (https://www.npmjs.com/package/geolib)
- expo-local-authentication (https://docs.expo.io/versions/latest/sdk/local-authentication/)
- expo-image-picker (https://docs.expo.io/versions/latest/sdk/imagepicker/)

A few other activities that were accomplished this week were to create a page that allows users to add new tasks and edit existing ones. This was previously done directly through the database for the sake of testing but that's changed now. Another thing that was done is fixing a few bugs in the login screen as the animation was not working as smoothly as it should and would cause issues with different screen resolutions.

Problems:

No problems this week.

TODO:

- Test biometrics both with happy and unhappy paths.
- Need to mark the task as complete – we don't want to always delete.
- Maybe add some animations to make it look smooth and nice.

WEEK 07: 15/03/2021 – 21/03/2021 (18 HOURS)

FYP Lecture and Seminar:

This week's lecture was the last lecture of the module and it was with Norman. It reminded us what the dissertation should contain and giving us what the poster should have and how the demos will take place. This gives me an indication that I should start thinking about a poster and how to design it. I will have a look at this during the holidays and plan it out. During the seminar, I talked about what I did which was starting the dissertation, finishing off the biometrics and dashboard screen, and marking tasks as complete so that they move to a different screen.

Activities:

The first task that I worked on this week was finished off the dashboard. This was two parts. One was cleaning up the dashboard library and the second was testing the biometrics and making sure they work fine. There were some issues with the GPS location but other than that, everything else worked fine. The next task was to mark tasks as complete. This required some changes in the database so that we know the state of the task. Then some UI changes took place to click a task to move it to the complete screen. This was done and tested and it worked perfectly. The page had to be reloaded to see the change take place.

Problems:

- Had an issue with the GPS location where if the user's GPS is not turned on, it will ask to turn it on but crash. This only happens the first time the app is ran.

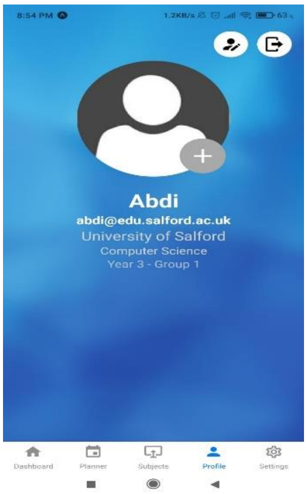
TODO:

- Look at libraries that can be used to create the user feed and comments and implement it.
- Create a profile.
- Create a settings page.
- Testing, testing, and testing.

OVER THE EASTER HOLIDAYS

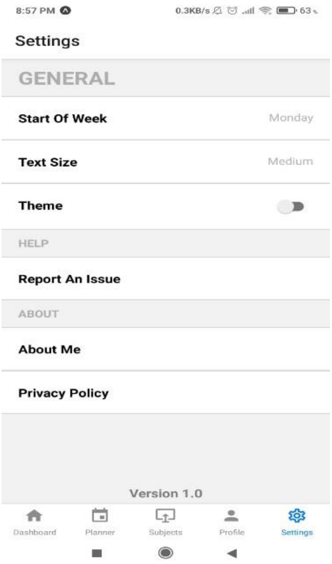
Activities:

The first thing that was done over the holidays was to creating the profile page so that users can see their information and edit it. This will be reflected in their timetable and the lessons they see. I looked for some libraries that have a profile design but couldn't find any so I decided to make my own profile page. This wasn't too hard and it was easy and quick. The profile can be seen below. This also required some new APIs in AWS and some fetch statements to interact with those APIs.



Abdi
abdi@edu.salford.ac.uk
University of Salford
Computer Science
Year 3 - Group 1

The second task was creating a settings screen so users can customise the application and help with useability and accessibility. This page was focused on allowing users to change font sizes and themes to help people with vision impairment and colour blindness. This page also allows some general changes to the application to make it the users. It also allowed the user to report any issues they might be facing. This settings page was not difficult and it was easy to accomplish. Some libraries were found that could help with the process such as react-native-settings, but I opted out on it and did it from scratch as it was easy and looked better. The theme was let out as it required a lot more work for it to work. The image below shows how the settings page looked after completing the implementation.



Settings

GENERAL

Start Of Week: Monday

Text Size: Medium

Theme

HELP

Report An Issue

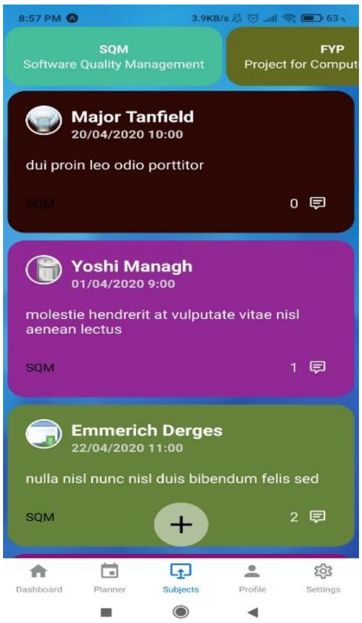
ABOUT

About Me

Privacy Policy

Version 1.0

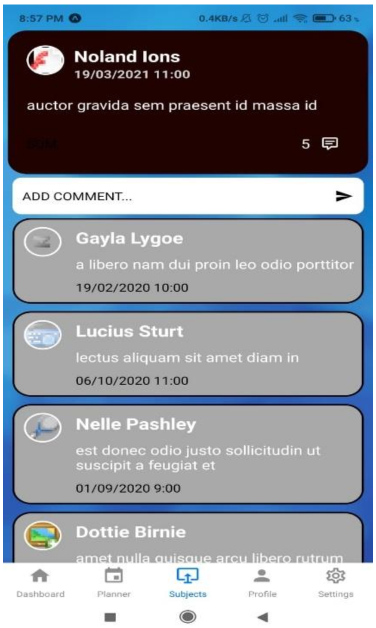
The penultimate task was to create a page where users can make a post and users can interact with them by commenting on those posts. The idea is that each module will have its own feed and people on that module can help each other with work and support. A lot of research was done to see if any libraries exist that would help me with the process, but unfortunately, nothing could be found. Therefore, I decided to do it all from scratch. This required the profile page since the user's information will be needed. The idea is that the user can scroll through the modules and click on the one they want and the feed that is associated with that module will be displayed. The user has the option to click on the post and add a comment or add a new post. This process was not too difficult and was straightforward to do. There were a few issues that I faced while implementing this feature but it was not difficult and all was solved within the day. https://www.mockaroo.com/ was used to generate the data for the uses and was stored in the database.



Major Tanfield
20/04/2020 10:00
dui proin leo odio porttitor
SQM 0

Yoshi Managh
01/04/2020 9:00
molestie hendrerit at vulputate vitae nisl aenean lectus
SQM 1

Emmerich Derges
22/04/2020 11:00
nulla nisl nunc nisl duis bibendum felis sed
SQM 2



Noland Ions
19/03/2021 11:00
auctor gravida sem praesent id massa id
SQM 5

Gayla Lygoe
a libero nam dui proin leo odio porttitor
19/02/2020 10:00

Lucius Sturt
lectus aliquam sit amet diam in
06/10/2020 11:00

Nelle Pashley
est donec odio justo sollicitudin ut suscipit a feugiat et
01/09/2020 9:00

Dottie Birnie
amet nulla quisque arcu libero rutrum

The final task that was done during the Easter holidays was testing out the 3rd sprint and also doing some regression testing to make sure the features that were done in the previous sprints still worked fine. I had a user test the application in depth to see if they could find any issues with it. Then I ran unit tests to make sure the components are set up correctly and that there are no mistakes in the data. The final test was repeating the same activity that I mentioned to the features of the previous sprints.

Problems:

- Not being able to take a picture so opted out of choosing a picture from the gallery.
- Finding a way to change the size of fonts and the theme. I was able to do the font on the setting page to prove that it works but the theme would require more work and a big revamp which I didn't do.
- Had problems setting up Jest Unit Testing Framework. After online research, I managed to fix it and it works now.
- There was an issue with adding a post when the user has an image since it takes time to read the base64. Therefore, testing was done with a user with no image.

TODO:

- All that’s left now is to carry on with the dissertation.
- The problems mentioned above will either be fixed in semester 2 or will be omitted and be mentioned in the dissertation.

SEMESTER 2 (PART 2)

The rest of the semester was spent on writing the dissertation. No more implementation was done.

APPENDIX B – PROJECT PROPOSAL

Increase Time Efficiency of Students using a Student Manager Application

AIM

The aim of this project is to create a mobile application to help first year Computer Science students at the University of Salford manage their time more efficiently.

CLIENTS AND CUSTOMERS

For the project, the clients for the project will be the School of Computing Science & Engineering in the University of Salford, along with the lecturers that teach the students. The customers will be the first year Computer Science students as they will be the ones using the application to help them better manage their time.

UNDERTAKING THE PROJECT

Moving into university is a big step in one’s life and it could be very stressful and nerve wracking, especially in this climax with the Covid-19 pandemic. Students are experiencing university in a new and unique way where everything is taught online. This creates lots of issues such as not getting the “University Experience” and losing out on the face-to-face interaction with peers and teachers. This could have a detrimental effect on one’s mental health. This makes it harder for students to study effectively and will also have a negative impact on one’s time management.

On the other hand, students who perceived control of their time reported significantly greater evaluation of their performance in their studies and overall satisfaction. Students that learn time management at an early stage in life will be better prepared for the work life and will make the transition from education to employment a lot easier and less stressful.

There is no tool that the University provides that helps students with their time management. Although there are many free online applications that are general purposed, this application is geared towards the first year Computer Science students who are new to university. It will give them the tools needed to manage their time and provide them with the skills needed to face the future.

BENEFITS FOR THE CLIENTS AND CUSTOMERS

The project will benefit the clients by creating an atmosphere where teachers will be able to support students with the important tasks due to students being productive and doing their work on time. This will enhance the quality of the teaching. The project also creates room for future expansion so the University can use this application for all its students and keep it as an in housed centralised platform.

The project will also benefit the customers by providing a lot of useful tools to help students make the most out of their university experience while keeping up with their education and extracurricular activities. The application will have a scheduler that is tailored towards students. It will have their timetables, which they can alter and customise, along with their exams and assignments on there. They will also have the option to add other activities such as work, gym, etc. The application will help them in their later years of education, and when they start their career as time management is one of the most important skills to have if they want to be successful.

BENEFITS FOR THE STUDENT UNDERTAKING THE PROJECT

The project will give me the opportunity to develop my technical skills and knowledge in an area that I am interested in, which is building a mobile application and creating software that helps people – in this case, helping students improve their time management skills. This will be a good challenge and will allow me to experiment with lots of different technologies that I would not have come across otherwise; in turn supporting my journey of becoming a competent professional software engineer. It will also improve my portfolio as well as opening the door for a potential career in the mobile development sector.

Aside from my technical skills, this project will immensely help with my soft skills. Reading literature papers and doing research will improve my ability to quickly extracting useful information and think critically as to whether the information fits into my project and how it can help me develop my project idea. Making sure that my project is on track while keeping my other modules up to date is going to push me to improve my time management skills along with my organisational skills. These will be very useful in the future when I start my career as they will make my transition from university to the employment a lot smoother. It will also make me more desirable as an employee.

CORE OBJECTIVES

1. Explore a variety of development approaches suitable for building a mobile application such as Java/Android studio, Angular, React, Xamarin, etc before choosing an approach suitable for developing the final product.
2. Survey students as to what they would like to see in the application and design the product based on their feedback. Create use cases for the user interaction scenarios and UML diagrams to represent the architecture of the system.
3. Look into different hosting services such as Poseidon (university server), AWS, docker swarm, etc to host the database and scrips that will be used by the application.
4. Research into building Restful APIs using the appropriate technologies for web scraping the timetables of first years and extracting information from the database to give to the application.
5. Allow students to have the ability to customise their lessons; add additional activities; mentioned missed lessons to alter the timetable to go to a different slot; mention subjects they are struggling with to add more lessons of that subject in the timetable.
6. Students will have the ability to interact with each other by creating a chat option in the lessons where support can be provided, and questions can be answered by peers.
7. Investigate biometrics and geolocation with the chosen technologies to let users register to classes. Lecturers could be altered if there are students that have low attendance.
8. Explore the best way to have a user-friendly to-do list and implement it so students can manage their tasks easily.
9. Application needs to be accessible to all. Need to investigate vision impairment, colour blindness, voice option, change font and size of text, dark mode.
10. Evaluate the product at each stage by running different tests such as Unit Testing, Acceptance Testing, UI Testing, Load Testing, and Regression Testing to make sure that the product is meeting the requirements.

OPTIONAL OBJECTIVES

1. Add a feedback page where students can submit feedbacks to improve the user experience and usability of the application.
2. Add a superuser that can directly make changes to certain elements of the applications that normal user cannot and give the superuser the power to monitor user interaction with the application to see where students spend most of their time.

DEVELOPMENT METHODOLOGY

Picking the appropriate methodology is very important as it will have a huge impact in the successful completion of this project. After some research, and comparing 4 different methodologies (waterfall, rapid application, agile, and DevOps), the logical conclusion to come to would be to use Agile methodology due its approach and advantages. Thanks to its iterative approach and incremental deliverables, it focuses on working on induvial features and delivering small end-to-end functionality to get feedback on them. It is also amazing for managing risk, such as spotting bugs early, and its versatility when it comes to changing or adding new features. This is exactly what is needed for this project since a big part of it is getting feedback from the supervisors and aligning expectations to progress in the right direction.

SOFTWARE AND TECHNOLOGY USED

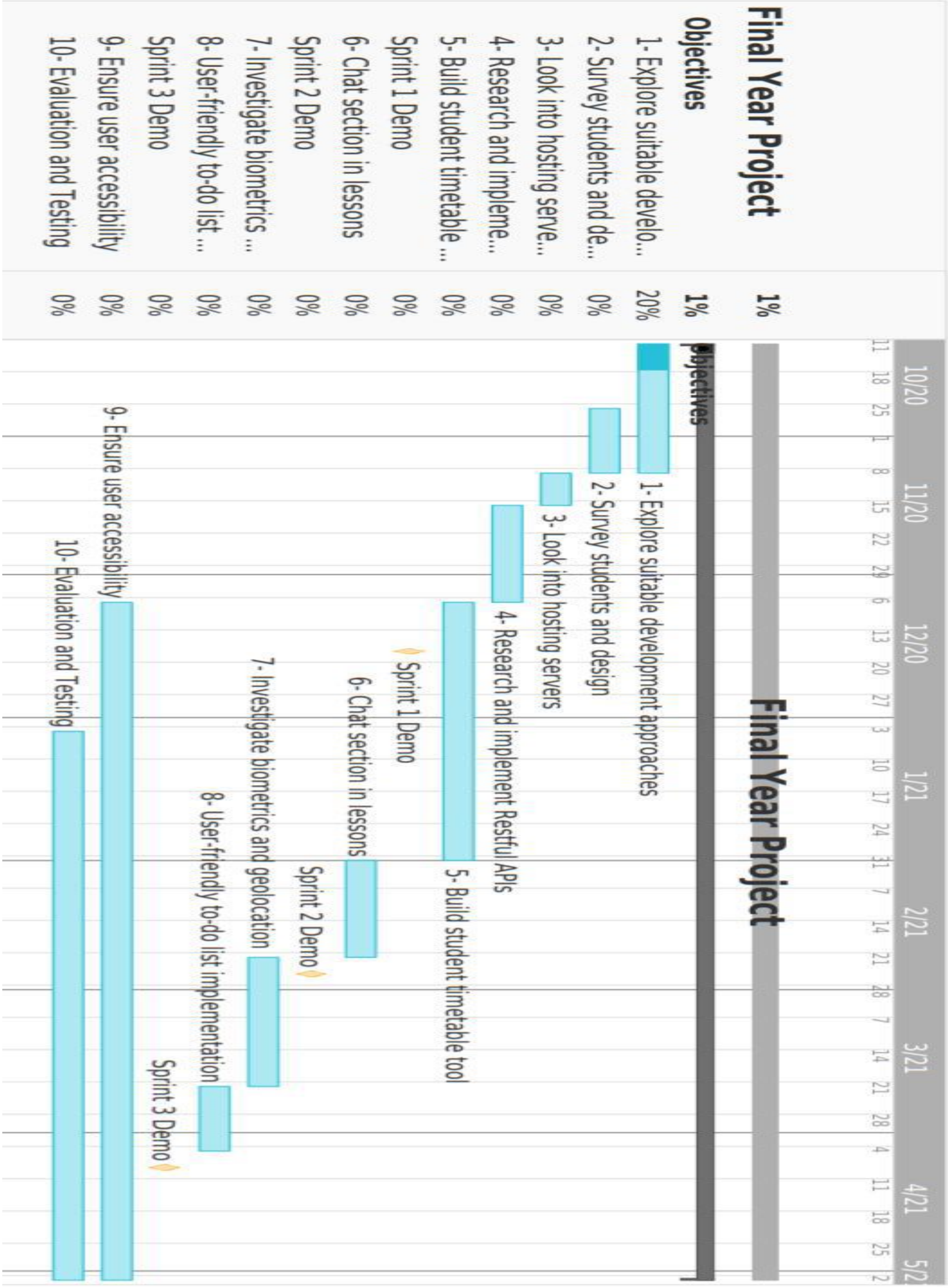
The project will be developed using my personal laptop which is a Windows 10 Home. The processor is intel(R) Core (TM) i5-8250U CPU @ 1.60GHz - 1.80 0Hz. The installed RAM is 8GB and has a 1TB SSD.

The technologies that will be used to build the application will need to be further researched to determine which one is best suited for the project. Having said that, the project can be re-created using a mid-end computer with a minimum requirement of Windows 7, 3GB RAM and 3GB disk space.

To avoid the complications that come with accidental file deletion or laptop corruption, Git and GitHub will be used. This will also be useful for source control which will help save time and will allow for rollbacks if a mistake is made during the coding journey. Git can also branch out which allows for parallel experimentation on the code base and merging them with little to no effort.

Microsoft word and OneDrive will be used to write the logbook (which will be shared to supervisors), project proposal, and final report. For organisational purposes, Gantt Chart and Trello will be used to plan the project out and to keep it on track. The internet will be used to do research on the project and finding solutions for problems the project faces.

APPENDIX C – PROJECT GANTT CHART



Final Year Project	1%
Objectives	**1%**
1- Explore suitable develo...	20%
2- Survey students and de...	0%
3- Look into hosting serve...	0%
4- Research and impleme...	0%
5- Build student timetable ...	0%
Sprint 1 Demo	0%
6- Chat section in lessons	0%
Sprint 2 Demo	0%
7- Investigate biometrics ...	0%
8- User-friendly to-do list ...	0%
Sprint 3 Demo	0%
9- Ensure user accessibility	0%
10- Evaluation and Testing	0%

APPENDIX D – FOCUS GROUP CONSENT FORM

Research Project Title: Increasing time efficiency of students using a student manager application.

Name of Researcher: [redacted].

By signing this form, I confirm that:	Initial
I agree to participate in the above focus group carried out by [redacted].	
I have read and understood the Participant Information Sheet. I have been able to ask questions about the study and my questions have been answered to my satisfaction.	
I understand that my participation is voluntary and that I can refuse to answer questions and I am free to withdraw at any time, without giving a reason.	
I will not identify participants in the focus group or disclose anything said during the focus group discussions that could identify a participant or any other person.	
I agree to the focus group being audio and video recorded.	
I am fully aware that data collected will be stored securely, safely and in accordance with Data Collection Act (1998).	
I agree to the use of anonymised quotes made by me in publications.	

Print Name: ______________________________________

Signature: _______________________________________ Date: _____________

What if I have concerns about this research?

If you are worried about this research, or if you are concerned about how it is being conducted, you can contact the organiser of the focus group at [redacted]

APPENDIX E – FOCUS GROUP PARTICIPANT INFORMATION SHEET

Research Project Name

Increasing time efficiency of students using a student manager application.

Invitation

You have been invited to participate in a focus group ran by [redacted] regarding a project under the University of Salford. Before you decide to participate, it is important you understand why the focus group is being done and what it will involve. Please take time to read the following information carefully and discuss it with others if you wish. Email the organiser if there is anything that is not clear or if you would like more information. Take time to decide whether or not you wish to take part. Thank you for reading this.

Purpose

The purpose of this focus group is to discuss the best way to increasing time efficiency and productivity of first year students, especially during the pandemic. The information learned in this focus group will be used to gather user requirements and appropriate features that would be applied to building an application that will be used by the students to help them better manage their time and stay productive.

Do I have to take part?

It is up to you to decide whether or not to take part. If you do decide to take part, you will be able to keep a copy of this information sheet and you should indicate your agreement in the consent form. You can still withdraw at any time. You do not have to give a reason.

Procedure

As part of this study, you will be placed in a group of 4-6 people. A moderator will ask you several questions while facilitating the discussion. As stated in the consent form, this focus group will be recorded. However, your responses will remain confidential, and no names will be included in the final report. You can choose whether or not to participate in the focus group, and you may stop at any time during the course of the study. Please note that there are no right or wrong answers to focus group questions. The goal is to hear many varying viewpoints and for everyone to contribute their thoughts. Out of respect, please refrain from interrupting others. However, feel free to be honest even when your responses counter those of other group members.

Benefits and Risks

Your participation may benefit you and other students by providing a lot of useful tools to help students make the most out of their university experience while keeping up with their education and extracurricular activities. The application will help them in their later years of education, and when they start their career as time management is one of the most important skills to have if they want to be successful.

In terms of the risk when participating in the focus group, no risks are anticipated beyond those experienced during an average conversation.

Confidentiality

Should you choose to participate, you will be asked to respect the privacy of other focus group members by not disclosing any content discussed during the study. The researchers will analyse the data, but—as stated above—your responses will remain confidential, and no names will be included in any reports.

Ethically Review

All University projects needs ethical consideration. No work can start until approval is granted. Researchers are responsible for ensuring that all activities fall within the principles set down in the University Academic Ethics Policy. Therefore, this project has been ethically approved by The Ethics Committee.

Contacts for further information

If you are worried about this research, or if you are concerned about how it is being conducted, you can contact the organiser of the focus group at [redacted]