Heriot Watt University

School of Mathematics and Computer Science

MEng Software Engineering

Higher Education Study-Planner

Deliverable 1: Final Year Dissertation

Matthew Frankland

SUPERVISOR: Frank Broz

Declaration

I, Matthew Frankland, confirm that this work submitted for assessment is my own and is expressed in my own words. Any uses made within it of the work of other authors in any form (e.g. ideas, equations, figures, text, tables, programs) are properly acknowledged at any point of their use. A list of the references employed is included.

Signed: Matthew Frankland

Date: 08/11/2019

Abstract

Students' learning in higher education is currently affected by a lack of mobile applications for planning and tracking their academic life. Resources that are available are hindered by a number of critical flaws including no mobile optimisation, badly designed UXs and out of date architectures. The aim of this project is to develop a mobile (iOS and Android) application and web-based portal that fixes these persisting issues and introduces some new unique features.

Each component of the mobile application will have an intuitive design and UX. The components of the mobile application will include file sharing boards, notification reminders, shared calendars, locally stored timetables, space to take lecture notes and instant messaging support.

The online portal will allow lecturers to connect with students. Lecturers will be able to view publicly available details on students, send out notifications, respond to messages, and add new events to shared calendars.

Abbreviations

UX – User Experience

UI – User Interface

PNS - Push Notification Service

SNS – Simple Notification Service

PHP – Hypertext Preprocessor

HTTP – Hypertext Transfer Protocol

IPC – Inter Process Communication

SQL – Structured Querying Language

HTML – Hypertext Manipulation Language

CSS – Cascading Style Sheet

FSR – Functional System Requirement

NFSR – Non-Functional System Requirements;

OS – Operating System

MVC – Model View Controller

Table of Contents

1. Introduction, Aims and Objectives

1.1 Introduction

Students' learning in higher education is currently affected by a lack of mobile applications for planning and tracking their academic life. Resources that are available are hindered by a number of critical flaws including no mobile optimisation, badly designed UXs and out of date architectures. The aim of this project is to develop a mobile (iOS and Android) application and web-based portal that fixes these persisting issues and introduces some new unique features.

Nearly all students and lecturers work asynchronously across multiple different areas of study. Therefore, it is imperative that any system for recording academic work, checking upcoming events, and communicating with colleagues be fast, simple, and easy to access. A high-level objective of this project is to meet these three criteria. This will be achieved by optimising the application's code for the local web browser or mobile device.

Another high-level objective of this project will be to design a contemporary UI. Evidence has shown that a well-designed UI can result in productivity, accessibility and user retention all increasing (Harte *et al.*, 2017). Market data and regular prototype feedback will be gathered to inform design decisions and achieve this objective.

1.2 Aims

This project's aim is to implement and deploy an innovative mobile study planner (iOS and Android) which works in tandem with an online portal for lecturers. Each component of the mobile application will have an intuitive design and UX. The components of the mobile application will include file sharing boards, notification reminders, shared calendars, locally stored timetables, space to take lecture notes and instant messaging support. The online portal will allow lecturers to connect

with students. Lecturers will be able to view publicly available details on students, send out notifications, respond to messages, and add new events to shared calendars.

1.3 Objectives

1.3.1 Mobile Implementation

1.3.1.1 Front-End View

Implement a front-end view for the mobile study planner. This will be done using the REACT-Native framework for compatibility with both iOS and Android. The font-end will provide an initial framework to work with on future development as well as an intuitive UI for users to interact with.

1.3.1.2 Intermediary Controller

Implement an intermediary controller in PHP to facilitate HTTP POST requests between the front-end view and the database. The controller will protect against malicious attacks, such as SQL Injection and Cross Site Scripting, validate form data and handle cookies.

1.3.1.3 Database

Implement a MySQL database for storing user data including user's personal information, hashed password and local front-end settings. The database will be implemented in the relational database language MySQL because of its scalability and high performance.

1.3.2 Website Implementation

1.3.2.1 Font-End View

Implement a front-end view for lecturer portal using HTML5, CSS3 and Core-JS. The front-end view will facilitate user login, instant messaging services to students, update class calendars and share files on class sharing boards.

1.3.2.2 Integration to Back-End System

Implement additional scripts on the PHP controller to facilitate the website's integration into the existing back-end database and student application. Lecturer accounts will be marked with special permissions as their accounts will be able to access functions that students will not.

1.3.3 UX Evaluation using a Closed Beta

Implement a usability evaluation in a small, closed format with a single class of students. The evaluation will consist of the students and lecturer of a class using the application for a short period time. The results of this evaluation will influence changes to the system before it moves to an open usability testing.

1.3.4 UX Evaluation using an Open Beta

Implement a wider open access beta UX evaluation via TestFlight and Android Beta Testing. Access to this beta version will be via a requested link and testers will be asked to complete the usability questions in section 1.3.4. Responses will be sent via email. Select responders will also be asked to participate in a focus group where a deeper insight will be sought into what they would change about the system.

1.3.5 Deployment

There are two methods available for deploying the mobile application to the public: through an app marketplace or through a distribution link. Distribution links could be useful during usability testing however on public deployment the mobile application will be published on the Apple App Store and Google Play. Regular logs will be maintained to mark any occurring errors that may need fixed.

2. Background

2.1 Mobile Applications in Higher Education

Mobile devices are quickly becoming vital tools for achieving academic success. The reason for this is that they provide simple methods for communicating with peers and accessing information. This has presented new opportunities for developing classroom learning through utilising these mobile devices as knowledge bases, management tools, and means of storing data (Peters, 2005).

With the importance of mobile devices in education getting ever higher, it has become necessary to understand what devices students own and how student's use these devices. A survey by Educause Center for Applied Research (2013) found that the majority of students in higher education own some form of mobile device with many using these devices for academic work. However, the frequency of mobile technologies being used within the classroom was found to be dependent upon lecturer's personal attitudes towards these devices. When the lecturer's attitude towards mobile devices in the classroom was positive, 49% of those surveyed said they were more likely to proactively engage with their course.

A further survey on the same topic as the survey above was done to help further mobile device adoption within the classroom (Chen *et al.*, 2013). This research had three areas of investigation which were identifying students most common devices, digital skill levels, and normal behaviours when using mobile devices.

2.1.1 Device Ownership Survey (Chen *et al.*, 2013)

The following data on device ownership among students in higher education was gathered by Educause through a survey conducted at the University of Central Florida in 2012 (Figure 1 (Chen *et al.*, 2013)). 91% of respondents said they particularly favoured owning a smartphone during higher

education. Tablet devices had a much lower favourability rating among student (37%) however this figure was found to have been impacted by age and education level. If the survey results were narrowed to just graduates or older students, then favourability levels for tablet devices increased.

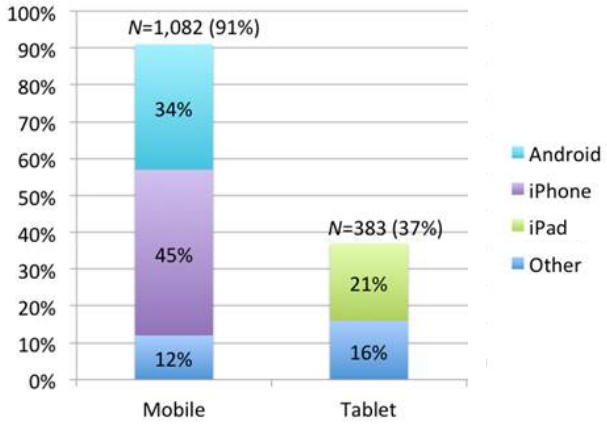

Figure 1. Device Ownership (N=1,082), Educause, 2013

The survey went on to ask whether students used their devices for academic purposes. As Figure 2 (Chen *et al.*, 2013) shows, 58% of respondents who said they owned a smartphone said that they did use their device for academic purposes versus 82% for respondents who said they owned a tablet. Students age, ethnicity, academic year, grade point average and gender all affected the results. The key findings being:

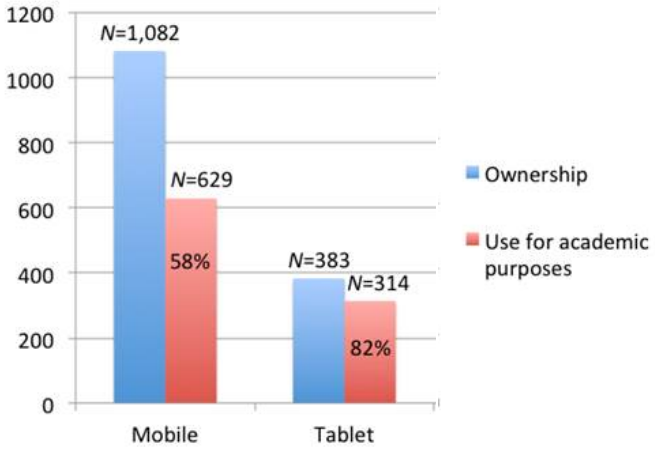

Figure 2. Comparing ownership and use for academic purposes (N=1,082), Educause, 2013

- The age of a student had a small impact upon whether they use their mobile device for academic purposes.

- Male students use their devices for academic purposes more than female students.
- No significant factor could be drawn for tablet devices as their use for academic purposes was high.
- There was negative correlation between mobile device ownership and grade point average. The study hypothesised that those with a lower academic rating are more drawn to their mobile phones as a means of distraction from hard study. It did not however draw any fixed conclusions as to why this correlation occurred.

The survey also aimed to identify the most popular apps among students, see Figure 3 (Chen *et al.*, 2013). Student's most popular use for their smartphones was accessing to social media, listening to music and playing games. Students were also asked to identify the apps they used the more for academic purposes. Results included 'Flash Cards'; 'iTunes U'; 'CourseSmart'; and 'MobileLearn'.

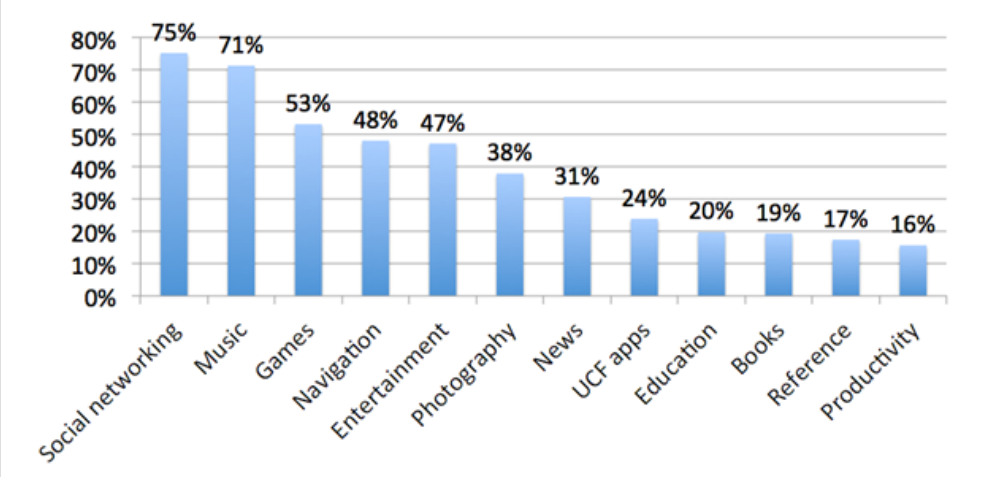

Figure 3. Most popular apps by category rated by students (N=933), Educause, 2013

2.2 Mobile Technology Review

2.2.1 Software

2.2.1.1 Third-Party Libraries

Third-party libraries are reusable software modules which are created and maintained by a single developer or development group. Third-party libraries have a number of uses such as connecting

your app to social media sites, displaying advertisements that form a revenue stream, and adding the ability to extract analytical information. Third-party libraries can also extend core functionality and make development faster and cleaner (Ma *et al.*, 2016).

Using third-party libraries has many potential advantages including decreasing development time because of code reuse and increasing efficiency on executed tasks. Disadvantages of third-party libraries are that they often have little technical support and can cause security flaws in your application if its code contains vulnerabilities (Raemaekers *et al.*, 2012) (Haefliger *et al.*, 2008).

An example of a well design third-party library that will be good to use in this project is react-native-firebase (invertase, 2019). This is a third-party library which connects an application to all Firebase services. This widely used third-party library would be useful for gathering analytical information during the usability studies. It is also is highly reviewed, well-maintained, and well-documented.

2.2.1.2 PNS

Data dissemination requests can fall into one of two categories, either a pull or push request. A pull request is a user made request for a particular piece of data from an external server. The server handles this request and sends the data back to the requester. A push request occurs without an explicit user request. A push request is instead initiated by an external server, usually through a service, which pushes the data to a user device. A push notification service is a push request service that sends data to a user's device in the form of notifications (Guo *et al.*, 2013).

The main flaw in push requests is that data has the potential to be pushed to a device without the user's permission. A fix for this is to implement a push request system using tokens. The token model allows users to preapprove notification servers before any data is sent to their device. This model works by a device requesting a unique token from the notification server upon first

connection. The device stores this token in the application server and thereafter a copy of this token is sent with every notification message. The notification server can send the notification to the user's device if the notification server has the token in its approved list. See Figure 4 (Haefliger, 2008).

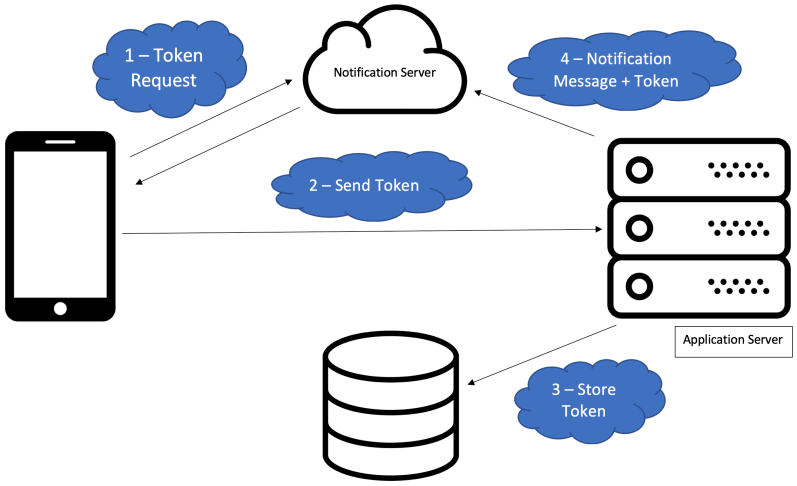

Figure 4: Token System for Push Requests

Push requests have three main types: blanket push, filtered push, and subscribe/publish push. Blanket push is a method for pushing a regular message to a device without the need for device filtering. There are many free services that offer blanket push requests however they come with a high bandwidth load on the application server and customization options are limited (Guo *et al.*, 2013).

Filtered push is a method for pushing a message to a filtered device list. This type of push request requires more administration by the user but does allow for higher degree of customization and uses less bandwidth than blanket push requests (Guo *et al.*, 2013).

A subscribe/publish push allows users to set preferences on their device on what notifications they want to receive. This is combined with external data on the application server to form classification groups. This type of push request uses less bandwidth on the application server than filtered push, but the scalability of these requests is usually untested before application rollout (Guo *et al.*, 2013).

2.2.1.3 Secure Storage

As society has increasingly begun to use mobile devices for security sensitive tasks, like storing passwords and banking information, it has become imperative that mobile device data is stored securely. REACT-Native can use a number of security mechanisms that are available to do this: (Cooijmans *et al.*, 2014)

- Android KeyStore

Android KeyStore is a method of storage that is unique to the Android operating system. It uses IPC technology to communicate with a service called KeyStore which stores cryptographic keys in a separate software container. When data is in the KeyStore it can be used for cryptographic operations, but it cannot be exported back out of the KeyStore. This means that a user entered password cannot be used to encrypt any keys.

- Shared Preferences/NSUserDefaults

Shared Preferences (Android) or NSUserDefaults (iOS) can be used to store and retrieve data. Data is stored unencrypted in the app directory of the local filesystem based upon a key value. This is an unsecure way of storing data as anyone with root level access on the device will be able to see the unencrypted data. It is however the simplest method for storing data so can be useful for storing general system files.

- Keychain

Keychain is a secure method of storage unique to the iOS operating system. The keychain API allows an application to store small pieces of data such cryptographic keys and certificates for secure communication.

Keychain or KeyStore can be accessed in REACT-Native through use of a third-party library, for example react-native-keychain (oblador, 2019).

2.2.1.4 Limitations

Smartphones have a lower screen resolution compared to tablet devices. This limits the UX when a large amount of data is displayed at one time. Problems may occur if, for example, development is only carried out on tablet devices. The result would be users performing a lot of scrolling in order to view the data that was available on the screen. A better method of presenting data, that overcomes this limitation, is to structure data in a controller based hierarchical format (Gong, 2004).

Mobile devices are usually owned by only one person and therefore are personal to each individual. This fact is normally reflected in the program filesystem and could cause problems if, for example, folder names have been customised. Device users may also have differing skill and therefore some services that a developer would expect to be running may not be because they have not been set up by the user (Gong, 2004). It would therefore be best in this project's applications to stick to core libraries as much as possible.

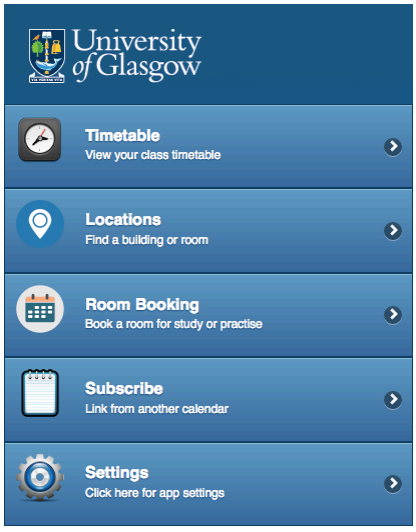

Figure 6: Timetable & Room Finder (University of Glasgow, 2019)

Figure 5: myHWU (Collabro, 2017)

2.2.2 Existing Applications

Many higher educational institutions have created mobile applications or mobile friendly versions of their website to try and take advantage of research into students' mobile practices. Two such examples are 'myHWU' (Collabro, 2017) by Heriot-Watt University and 'Timetable & Room Finder' (University of Glasgow, 2019) by Glasgow University.

These apps offer useful tools to students when they are not in classes. This often includes information on shops, study locations, event timetables and public transport. All of these applications do not offer any, or very limited, benefits to students for during their academic learning. This project will seek to implement features that are specifically for academic learning such as file sharing boards, shared calendars and locally stored timetables.

As figures 5 and 6 shows, both 'myHWU' (Collabro, 2017) and 'Timetable & Room Finder' (University of Glasgow, 2019) have bland UIs. 'myHWU' has a lot of functionality which makes the UI seem busy and cluttered. This can turn users away when they are looking for something specific, such as their student record. A good aspect of 'myHWU' however was its use of colour, font and font weight to control the hierarchy of elements that are on screen compared with 'Timetable & Room Finder' which relies heavily on the font size to control the hierarchy of elements. 'Timetable & Room Finder' has a smaller range of functionality which makes it an easier app to use however the UX is still hampered by the standard and somewhat old-fashioned table format.

Mobile devices also offer some of the functionality that will be built into this project such as shared calendars through a local Android or iOS calendar app. Shared calendars through the local Android or iOS calendar app is not however tailored towards an educational environment. Many features that would be beneficial in this environment, such as individual permissions, are therefore missing. This project would aim to fix these flaws by tailoring all features, including shared calendars, for the academic environment that the app is eventually intended for.

3. Requirements Analysis

3.1 System Requirements Analysis

The following system requirements, which are derived from the aims and objectives of this project, are essential if the mobile application and web-based portal are to function correctly. A matrix of system requirements is laid out below.

	Requirement	**Priority**
FSR-1	The System Must Allow Users to Login and Register	M
FSR-1.1	The Mobile Application Must Allow Student's to Login and Register	M
FSR-1.2	The Online Portal Must Allow Lecturer's to Login and Register	M
FSR-1.3	The System Must Allow Users to Reset Their Password if They Forget It	M
FSR-1.4	The System Should Be Able to Assign Student's to Their Academic Class Groups After Students Login to Retrieve Their Timetable	S
FSR-2	The System Must Have a Data Protection Policy	M
FSR-2.1	The System Must Request Consent to Save Any User Entered Data	M
FSR-3	The System Must Generate User Profile's on Registration	M
FSR-3.1	The System Should Allow User's to Add Extra Details to their User Account	S
FSR-3.2	The System Must Allow User's to Set Privacy Controls on Their Profile Details	M
FSR-3.3	The System Could Allow User's to Set a Profile Picture	C
FSR-3.4	The System Should Allow Lecturer's Access to Public Student Details	S

FSR-4	The System Must Be Able to Access Student's Teaching Timetables When Requested	M
FSR-4.1	The System Could Be Able to Save Student's Timetables on Local Mobile Storage	C
FSR-5	The System Must Implement Shared Calendars	M
FSR-5.1	The System Must Allow Lecturer's to Update Course Shared Calendars	M
FSR-5.2	The System Could Incorporate Shared Event Calendars for Student Union and University Main Office	C
FSR-6	The System Must Have File Sharing Boards	M
FSR-6.1	The System Should Have Sub-Group Student Boards	S
FSR-6.2	The System Might Offer Previews of Files	W
FSR-6.3	The System Could Have Subgroup Boards for the Student Union and University Main Office	C
FSR-7	The System Must Give System State Notifications	M
FSR-7.1	Lecturer's Must Be Able to Push Notifications to Select Users	M
FSR-7.2	University Office Staff Should Be Able to Push Notifications to Select Users	S
FSR-7.3	Student Union Staff Could Be Able to Push Notifications to Select Users	C
FSR-8	The System Must Be Able to Store User's Notes	M
FSR-8.1	The System Could Export Notes to Mobile or Laptop File Systems	C
FSR-8.2	The System Might Have Organisational Tools to Sort Notes	W

FSR-9	The System Could Integrate Instant Messaging Support Between Lecturers and Students	C

NFSR-1	The system should only allow registered users to access the system	M
NFSR-2	Both mobile and desktop applications should be scalable to the end user's device	S
NFSR-3	All users should be able to access the systems servers at the same time	M
NFSR-4	System response times should not exceed 30 seconds	C
NFSR-5	Easy to read text or alternative text should be provided to make both mobile and desktop accessible to all	M

Key

FSR – 'Functional System Requirement'; *NFSR* – 'Non-Functional System Requirement'

M – Must Have; *S* – Should Have; *C* – Could Have; *W* – Would Like to Have

3.1.1 Use Cases

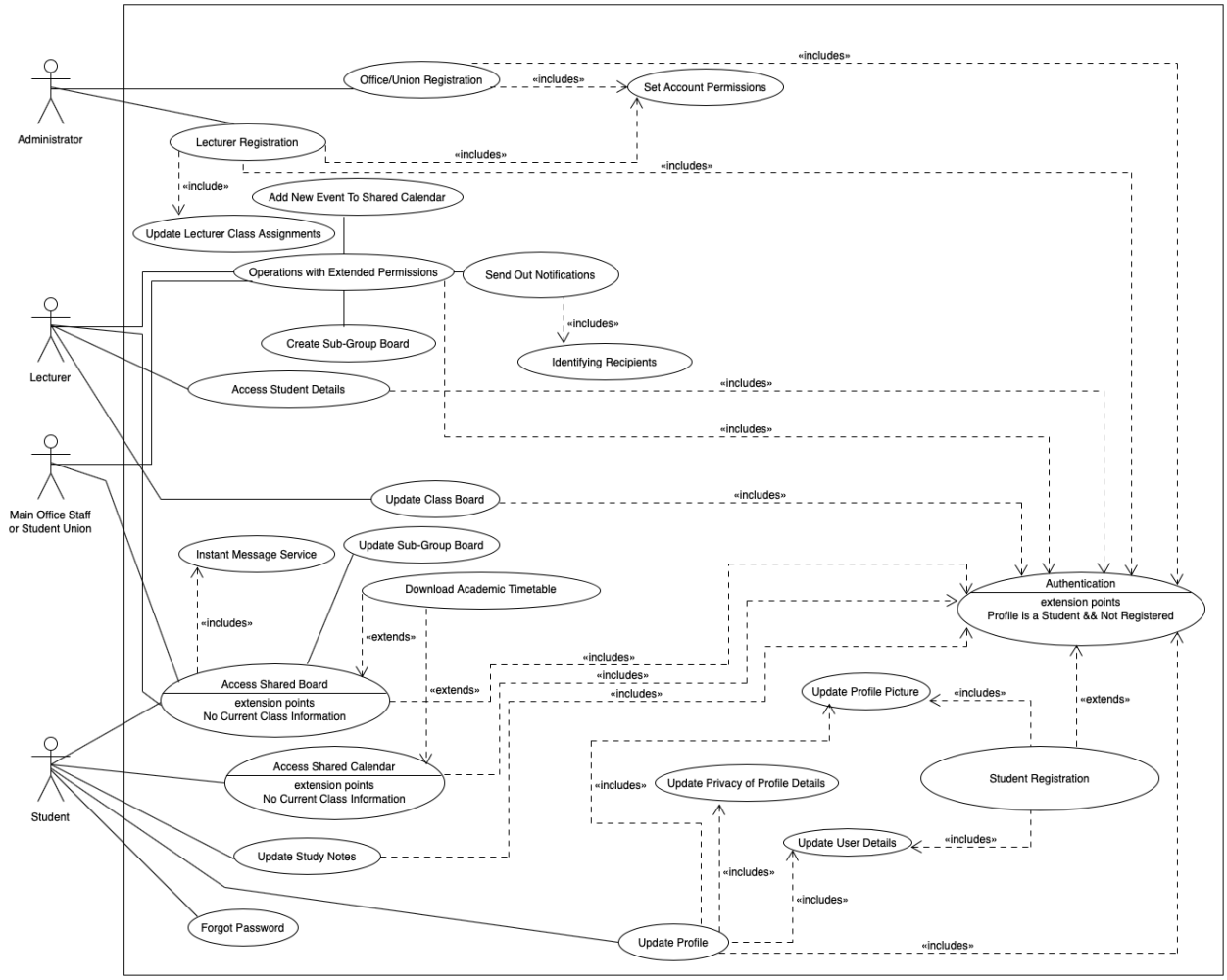

Figure 7: Use Case for System Requirements

Figure 7 is a graphical use case diagram which describes the functional system requirements listed in this section. This use case diagram shows how users interact with the higher education study planner and how individual components of the system interact with each other. Textual descriptions for each use case can be found in the appendix.

3.2 User Requirements Analysis

User requirements, which describe what the user expects the software to be able to do, will be gathered once development has started. These additional requirements will be gathered through a small focus group where users will be presented with the problem that this app is proposing to

solve. Focus group participants will then be asked a series of questions which will include the following:

- Are there any additional features you would like to see included in this project?
- What do you see the limitations of this project being?
- Are there any additional problems you would like to see this project attempt to solve?
- What core features would you identify as being most critical to this project's success? Additionally, what core features are least valuable to this project's success?
- How would you judge whether this project is a success?

3.3 Challenges

3.3.1 Access to University Database

Universities currently store student data, including sensitive data, in a central database. Access to this database will be required in order to request a user's academic timetable. The user's timetable will then be stored on the local device. This database request needs to be done in a secure manner that avoids any and all sensitive data.

3.3.2 Storing Student Information

Data such as user passwords, application media, push notification certificates, and user timetables will all need to be stored on local device memory. There may not, however, be any available space on the local device in which to save this data. This project's database will integrate with the available memory in order to reduce the weight on the local device where possible.

3.3.3 Compatibility with all Smartphone Types

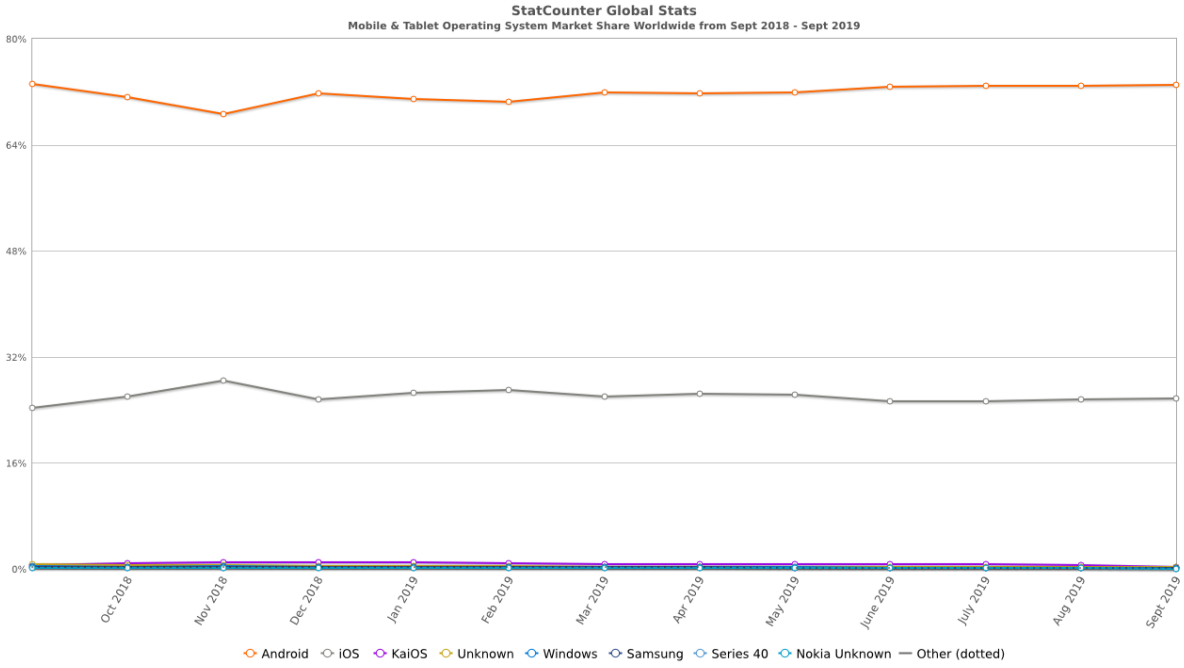

Figure 8: Mobile & Tablet Operating System Market Share Worldwide from Sept 2019 – Sept 2019, GlobalStats, 2019

As Figure 8 shows, the combined market shares of iOS and Android make up the majority of the mobile OS industry. There are, however, a number of other mobile OS platforms including KaiOS, Windows and Samsung. As this project's mobile application is aimed at as wide a student user base as possible, development will also take into account these smaller mobile OS.

Some advanced components of newer iOS and Android versions may not be available to users running older versions or alternative mobile OS. Newer components will only be used in this project if the feature it is intended for does not fall within the core functionality. This will make sure all core features are available to all users.

4. Design

The development of this project will be completed solely by myself and I will pick a design methodology to reflect this. The design methodology that I will use to complete this project will resemble the agile development process. I will set realistic sprint deadlines and assign a moderate number of tasks to each sprint in order to balance development, requirements gathering and testing.

My initial planning for each sprint will change once user requirements have been gathered. During my initial planning I will leave enough time to take these changes into account. I will use Git, a free open source version control system, to track my work; plan sprints; open tasks and bug fixes; and log what additional development is needed.

4.1 MVC

The system architecture of this project will be designed using an MVC approach. This will allow the logic and testing to be equally divided between sprints and provide an organizational method for keeping the system code organized using version control.

The benefit of using an MVC architecture is that any modification to the mode of one component will not affect another. The scale of the project is also large and thus this architecture is ideal for keeping control over the essential behaviour and identifying what behaviour is not as important.

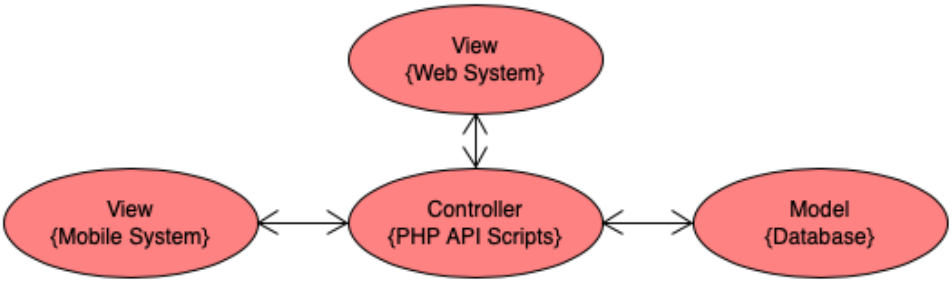

Figure 9: MVC

4.2 Initial Sketch of System Architecture

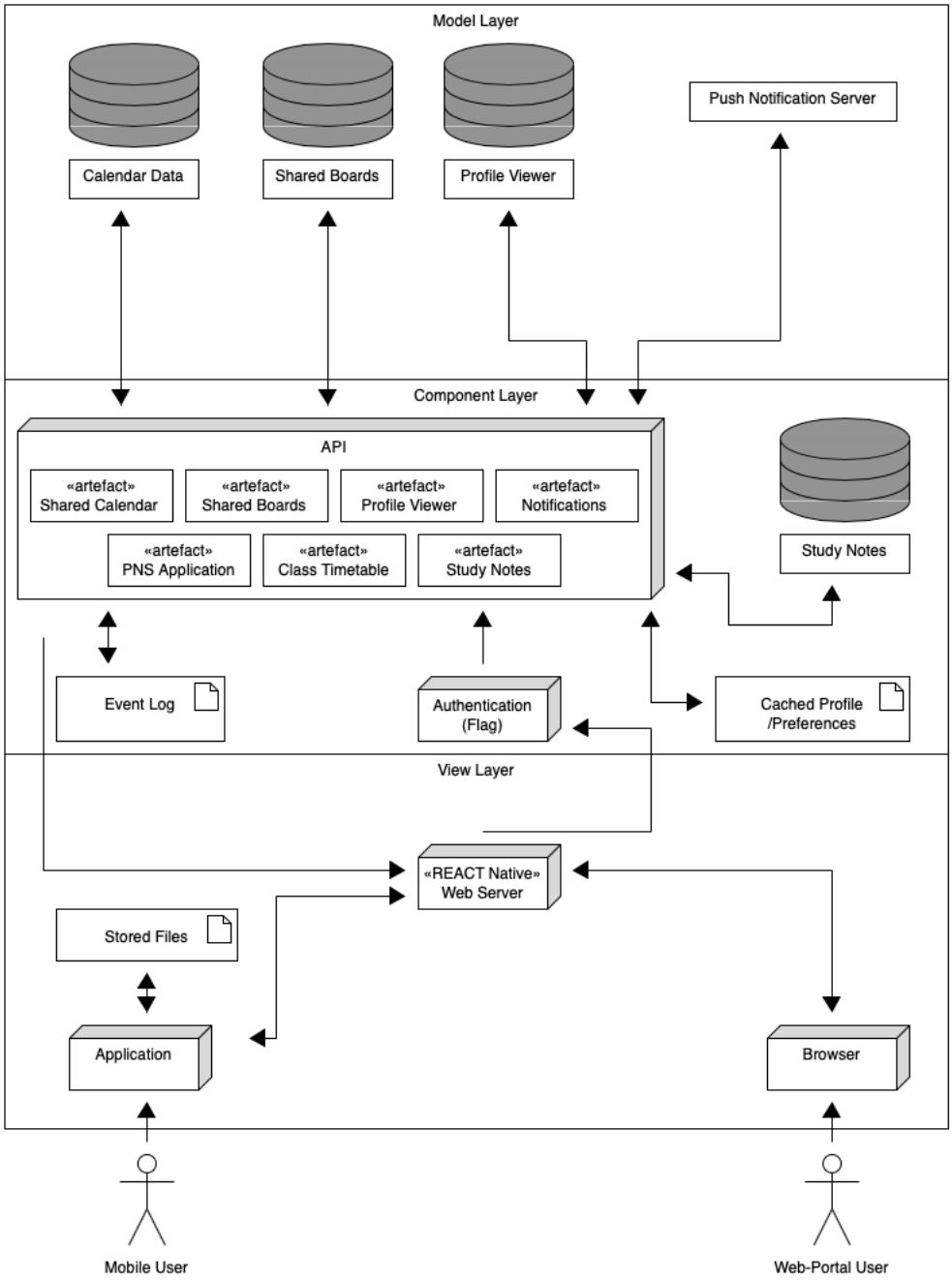

Figure 10: System Architecture

Figure 10 shows an initial sketch of the architecture for this project based upon a client-server architecture model. The client side of the architecture is represented by both of the applications, mobile and desktop, and the server side of the architecture is represented by a separate server which handles the processing.

5. Evaluation Strategy

5.1 First Usability Evaluation

This project's first usability evaluation will be conducted in a small closed environment. Participants will carry out this evaluation in pairs with one person using the mobile application and the other using the web-based portal. Participants will primarily vary in technological skill, age, and role (roles include lecturers and students). All participants will be asked to sign a consent form prior to taking part in this evaluation.

This usability evaluation will have the following aims:

- To determine whether the product produced meets the system requirements laid out in section 3.1.
- To determine whether the product produced meets the user requirements that will be gathered as described in section 3.2.
- To collect data, both qualitative and quantitative, that can inform further design decisions prior to the project's deployment.
- To identify any issues with the UX or usability.
- To identify any major bugs that impeded upon the systems use in its current form.

Participants will be told the purpose of both applications prior to this evaluation beginning. No explanation will be given on finding functionality as this forms part of this evaluation. Participants using the mobile application will be asked to complete the following initial tasks:

- To register an account using provided details
- To update the new profile with further provided details
- To upload a generic profile picture
- To log out, reset their password, and then log back in again

- To make some generic notes and test the organisational tools that are provided

At this stage users would normally need to register with classes by logging in to their higher educational institutes timetable system. For the purposes of this evaluation the participants registered account will be provided with generic classes, which will be added my myself, along with a generic academic timetable. These classes will be associated with the lecturer account that the other participant in the pair is being asked to use. Both participants in the pair will then be asked to complete the following further tasks:

- The web-based user should add an event to a classes auto generated shared calendar. The mobile user should verify that this event has appeared in their app.
- The web-based user should upload a provided file to their classes auto generated board. The mobile user should verify that this file has appeared in their app.
- The mobile user should send a message to the web-based user. The web-based user should send a reply.
- The mobile user should take their device offline and verify that their account's academic timetable can be viewed through the device's local storage.
- The web-based user should view the profile of the mobile user and verify that the details they can view are only those that are set to public. The mobile user should set some of their public details to private and the process should be retested.
- The web-based user should send out a series of notifications to the mobile user individually, their class group and to everyone in their filtered list. The mobile user should then check they have received all three notifications.

Some of the questions asked after these tasks have been completed will include:

- What tasks took the most time to complete?

- What functions were the most difficult to find on the UI?
- How did you find the system's usability?
- Was anything about the system misleading?
- Did you encounter any bugs?

Additional questions will be added based upon the user requirements from Section 3.2 once these are gathered. Space will also be left for participants to leave comments.

5.2 Second Usability Evaluation

The second usability evaluation will be a more open public evaluation and involve testing of the mobile application only. Students will be invited to download the application through a public beta testing program. Participants will be asked to complete an online Google Form which will include questions on the user interface; the effectiveness of the application; and the systems performance and security. Analytics will also be gathered from the application while it is running on individual devices.

Participants consent will be gathered through the beta test software prior to the participant downloading the application. Consent will include the gathering of analytical data. A custom version of the application will be made available to participants based upon their university. This will allow participants to log in to their own universities academic timetable server.

5.2.1 Automated Unit Testing

This usability evaluation will also use third-party libraries to generate automated unit tests. A log will be generated detailing what exceptions are encountered, what tests fail and the state of the system when tests fail. This data will be beneficial in diagnosing any bugs that users report. Consent for these tests will gathered prior to user participation in this evaluation.

6. Project Management

6.1. Timetable

Week Beginning	Task	Description
25/11/2019	Begin Implementation of Mobile Application	Begin to code the student portal
2/12/2019	User Requirements Gathering	Gather together a small group of students and lecturers from Heriot-Watt to discuss this project's user requirements
2/12/2019	Revise User Requirements	Adjust the project requirements (Section 3) based on the newly gathered user requirements
9/12/2019	Begin Implementation of Web-Portal	Begin to code the lecturer's web portal
16/12/2019	Begin Implementation of Controller	Begin to code the PHP scripts that will form the API that interacts with the Model
16/12/2019	Begin Implementation of Model	Begin to implement the mySQL based database
6/1/2019	Integrate Components	Combine the developed components into one single implementation
6/1/2019	Debug	Test the implementation and debug any problems that are identified

13/1/2019	Test in 1st Usability Evaluation	Conduct the usability evaluation (Section 5.1)
13/1/2019	Prepare Results from 1st Usability Evaluation	Compile the results from the usability evaluation questionnaire
13/1/2019	Evaluate Results of 1st Usability Evaluation	Identify the impact of the usability evaluation on development
20/1/2019	Implement Appropriate Changes From 1st Usability Evaluation Results	Implement the changes identified as being necessary as a result of the usability evaluation
20/1/2019	Write Automated Tests	Add automated tests into the code base that will run automatically on user devices during public beta deployment
20/1/2019	Implement Analytics into App	Add analytical code into the code base that will gather data on how participants use the application during public beta deployment
27/1/2019	Write Google Form for 2nd Usability Evaluation	Write the questionnaire that participants of the 2nd usability evaluation (Section 5.2) will be asked to complete

27/1/2019	Launch App in Public Beta for 2nd Usability Evaluation	Deploy the code base to public beta test suites so that advertisements for this usability evaluation can link to the app
3/2/2019	Promote 2nd Usability Evaluation	Put up posters, send emails, and promote participation in this usability evaluation
17/2/1029	Evaluate Results of 2nd Usability Evaluation	Identify the impact of the usability evaluation, questionnaire, analytics, and automated tests on development
17/2/2019	Implement Appropriate Changes From 2nd Usability Evaluation Results	Implement the changes identified as being necessary as a result of the usability evaluation
24/2/2019	Verify Compliance with Apple App Store and Google Play Store	Verify the project's compliance with the codes of conduct on the Apple App Store and Google Play Store
2/3/2019	Deploy Application and Web-Based Portal	Deploy and Advertise this project's applications

6.2. Analysis of Risks

ID	Risk	Description	Severity	Mitigation Plan
1	Loss of data	Data may become corrupt or accidently deleted.	Major	Data won't be stored in one location. I will ensure data is stored in several locations such as GitHub, hard disks, and USBs.
2	Acquiring new skills	Time may be required to learn new skills e.g. new programming languages.	Minor	I will ensure I have ample time to learn new skills. This may require extra hours of work however this will ensure the end product is completed on time.
3	Data is not stored securely	User data is not properly stored and leads to hackers obtaining it.	Major	All secure data will be encrypted and stored separately from encryption keys.
4	Requirements gathered are ambiguous.	Requirements aren't concise and are open to developer interpretation.	Minor	I will ensure that requirements are to the point during requirements gathering. I will make sure I am clear what issues are being tackled and ensure clarity on all requirements

				throughout the development process.
5	End User Expectations are not met	The user expects to receive something different compared to the project applications.	Severe	I will check with the intended end users that the application is meeting their expectations during development. I will ensure that they know what the system requirements are and that they agree with them during the user requirements gathering session.
6	Unforeseen requirements	Requirements are missed before the end of the project.	Moderate	I will update the project requirements and ensure all documentation is correct in relation to the updated requirements. I will ensure that any component that relates to a changed requirement is redone and that any new requirement features are implemented.
7	Underestimating the project size.	The project is oversimplified during the design stage.	Major	I will ensure requirements are prepared properly and that every requirement that is a

				'must have' is completed. The complexity of this project will be thoroughly understood before development begins.
8	Risk of unsustainable user growth.	The project is successfully deployed however the traffic on the system grows faster than expected.	Major	I will utilise bulk actions, create automated services within the app to reduce manual oversight, and create a system support page via a FAQ website.
9	Risk of app being denied.	The app is denied deployment into Google Play or Apple App Store.	Moderate	I will research the rules of each app store during a sprint. I will make sure all rules are adhered to before deployment.
10	Risk of only developing for a single platform.	A supported platform for deployment disappears that greatly reduces support base.	Minor	Android and iOS make up over 90% of mobile operating system market share therefore these platforms must take priority over all others as they make up the largest user base.
11	Risk of cyber attacks	The code has a vulnerability that causes bugs or	Moderate	I will manage exposure of data to client views through controller. Data will be

		viruses to get into client's systems or data is returned that should have been secure.		checked before being allowed to be download to user accounts. No data will be downloaded to user file systems other than txt or pdf documents.
12	Third party library licences are not adhered to	No deceleration of third-party licences used in this project is given.	Minor	I will make sure licence files are included in the final code base and that appropriate references are made within the source code.

6.3. Considerations of Professional, Legal, Ethical, and Social Issues

Software, including third party libraries, will only be used if it is open source, if a free student licence has been obtained, or if a bought licence has been obtained. All code, particularly in regard to code that is open source, will be appropriately accredited.

This project's usability evaluations and requirements gathering studies will be carried out in an ethical manner. Participants in studies or evaluation will be made aware of any risks, how their data will be used, and that they can stop at any time. Participants will be recruited via posters placed around Heriot-Watt, through targeted email requests within Heriot-Watt or other university institutions, and via face to face requests to students and lecturers at Heriot-Watt. No participants will be under the age of 16.

A. References

Harte, R., Glynn, L., Rodríguez-Molinero, A., Baker, P.M., Scharf, T., Quinlan, L.R. and ÓLaighin, G., 2017. A human-centered design methodology to enhance the usability, human factors, and user experience of connected health systems: a three-phase methodology. JMIR human factors, 4(1), p.e8.

Peters, K., 2005. Learning on the move: Mobile technologies in business and education. Flexible Learning Advisory Group.

Dahlstrom, E., Walker, J.D. and Dziuban, C., 2013. ECAR study of undergraduate students and information technology (p. 2013). 2013.

Seilhamer, R., Chen, B., Sugar, A., Berge, Z.L. and Muilenburg, L.Y., 2013. A framework for implementing mobile technology. Handbook of mobile learning, pp.382-394.

Collabro., 2017. myHWU. online, <https://apps.apple.com/gb/app/myhwu/id1260559655>, accessed 10 October 2019.

University of Glasgow., 2019. Timetable & Room Finder. online. <https://apps.apple.com/us/app/timetable-room-finder/id1479855374>, accessed 11 October 2019.

Chen, B. and Denoyelles, A., 2013. Exploring students' mobile learning practices in higher education. Educause Review, 7.

Chen, B. and Denoyelles, A., 2013. Device ownership (N=1,082). digital image. Educause. viewed 11 October 2019. <https://er.educause.edu/-

/media/images/articles/2013/10/chen_figure1.jpg?la=en&hash=3F590B0D38DA4D6EDFA84DE5F81ACBCAB4BED32F>.

Chen, B. and Denoyelles, A., 2013. Comparing ownership and use for academic purposes (N-1,082). digital image. Educause. viewed 11 October 2019, <https://er.educause.edu/-/media/images/articles/2013/10/chen_figure2.jpg?la=en&hash=8CAE27A5043E9D1044555DCEA0A42CDEB8E721C9>.

Chen, B. and Denoyelles, A., 2013. Most popular app categories rated by students (N=933). digital image. Educause. viewed 11 October 2019. <https://er.educause.edu/-/media/images/articles/2013/10/chen_figure3.png?la=en&hash=A6C92CC9DE2C3D42B0D26DD845ABD4514B16F130>.

Ma, Z., Wang, H., Guo, Y. and Chen, X., 2016, May. LibRadar: fast and accurate detection of third-party libraries in Android apps. In Proceedings of the 38th international conference on software engineering companion (pp. 653-656). ACM.

Raemaekers, S., van Deursen, A. and Visser, J., 2012, March. An analysis of dependence on third-party libraries in open source and proprietary systems. In Sixth International Workshop on Software Quality and Maintainability, SQM (Vol. 12, pp. 64-67).

Haefliger, S., Von Krogh, G. and Spaeth, S., 2008. Code reuse in open source software. Management science, 54(1), pp.180-193.

invertase., 2019. react-native-firebase. online. available at: <https://github.com/invertase/react-native-firebase>. accessed 10 October 2019.

Li, N., Du, Y. and Chen, G., 2013, December. Survey of cloud messaging push notification service. In 2013 International Conference on Information Science and Cloud Computing Companion (pp. 273-279). IEEE.

Guo, W. and Liu, H., 2013, August. The analysis of push technology based on iphone operating system. In Proceedings of 2013 2nd International Conference on Measurement, Information and Control (Vol. 1, pp. 570-574). IEEE.

Cooijmans, T., de Ruiter, J. and Poll, E., 2014, November. Analysis of secure key storage solutions on Android. In *Proceedings of the 4th ACM Workshop on Security and Privacy in Smartphones & Mobile Devices* (pp. 11-20). ACM.

Gong, J. and Tarasewich, P., 2004, November. Guidelines for handheld mobile device interface design. In *Proceedings of DSI 2004 Annual Meeting* (pp. 3751-3756).

oblador., 2019. react-native-keychain. online. available at: <https://github.com/oblador/react-native-keychain>. accessed 11 October 2019.

GlobalStats. 2019. Mobile & Tablet Operating System Market Share Worldwide from Sept 2019 – Sept 2019. digital image. Stat Counter. viewed 29 October 2019, <https://gs.statcounter.com/os-market-share/mobile-tablet/worldwide/#monthly-201809-201909>.

B. Figures

C. Appendix

1. Textual Use Case Descriptions

Use Case ID	1
Use Case	Forgot Password
Description	User requests a password reset
Primary Actors	Heriot-Watt Student
Preconditions	1. Student has an Account
Main Flow	1. The user opens the mobile app 2. The user clicks on the password forgotten button 3. The user enters their registered e-mail 4. The user opens the link in the email sent to their inbox 5. The user enters a new password and retypes to confirm
Post Conditions	1. The user has a new password to login to authenticate with

Use Case ID	2
Use Case	Update Study Notes
Description	User updates their stored study notes
Primary Actors	Heriot-Watt Student
Preconditions	1. Include (Authentication) 2. The user has opened the notes tab

Main Flow	1. The user scrolls through their notes 2. The user edits the note they want to change 3. The user reorganises their notes
Post Conditions	1. The user's notes have been updated and saved

Use Case ID	3
Use Case	Access Shared Boards
Description	User accesses a shared study/sub-group board
Primary Actors	Heriot-Watt Student, Lecturer, Main Office Staff or Student Union
Preconditions	1. Include (Authentication) 2. User is on the shared board tab
Main Flow	1. The user selects the board they wish to access 2. Include (Instant Message Service)
Alternative Flow	No Current Class Information

Use Case ID	4
Use Case	Access Shared Calendar
Description	User view events in a shared calendar
Primary Actors	Heriot-Watt Student
Preconditions	1. Include (Authentication) 2. User is on the calendar tab

Main Flow	1. The user selects the calendar they want to view 2. The user selects the day of the calendar they want to view
Alternative Flow	No Current Class Information

Alternative Flow ID	3.1/4.1
Use Case	No Current Class Timetable Information
Description	Details on Student's Classes Are Not Available
Primary Actors	Heriot-Watt Student
Preconditions	1. Include (Authentication) 2. Timetable was not found on local storage
Main Flow	1. Log in to timetable server 2. Retrieve data 3. Store locally 4. Return to Step 2 of preconditions in sender

Use Case ID	5
Use Case	Authentication
Description	Log in email and password are verified
Primary Actors	Heriot-Watt Student, Main Office Staff, Lecturer, Administrator
Main Flow	1. User has opened portal or application

	2. User has entered their registered email address 3. User has entered their registered password
Post Conditions	1. Authentication successful 2. User details retrieved
Alternative Flow	Profile is a Student && Not Registered

Alternative Flow ID	5.1
Use Case	Profile is a Student && Not Registered
Description	Student Has Failed to Log in Multiple Times in A Row
Primary Actors	Heriot-Watt Student
Preconditions	1. Log in email user attempted to use to access an account is not registered
Main Flow	1. User clicks on link to register 2. Include (Register)

Use Case ID	6
Use Case	Register
Description	New Registration with System
Primary Actors	Heriot-Watt Student
Main Flow	1. Student opens mobile application 2. Student selects registration button 3. Include (Update User Details)

	4. Include (Update Profile Picture)
Post Conditions	1. New profile created

Use Case ID	7
Use Case	Update Profile
Description	Update a Registered Profile with New Details
Primary Actors	Heriot-Watt Student
Preconditions	1. Include (Authentication)
Main Flow	1. Include (Update Profile Picture) 2. Include (Update User Details) 3. Include (Update Privacy of Profile Details)
Post Conditions	1. New Account Created

Use Case ID	8
Use Case	Instant Message Service
Description	Student send message to organiser of board
Primary Actors	Heriot-Watt Student, Lecturer, Main Office Staff or Student Union
Preconditions	1. Include (Authentication) 2. User Is in messaging screen in a shared board
Main Flow	1. User enters intended recipients name 2. Name verified as available within shared board

	3. User writes message 4. User Presses Send
Post Conditions	1. Message is received on recipients account

Use Case ID	9
Use Case	Update sub-group board
Description	Upload of a post onto a sub-group board
Primary Actors	Student, Lecturer, Main Office Staff or Student Union
Preconditions	1. Include (Authentication) 2. User on shared boards tab 3. User has selected a board that is not a class board
Main Flow	1. User selects add a new post 2. User uploads any relevant files into location in board's directory 3. User adds some text describing post 4. User presses post
Post Conditions	1. Successful upload of post 2. Any uploaded data stored successfully

Use Case ID	10
Use Case	Update Profile Picture
Description	Update the Profile Picture stored on file

Primary Actors	Student
Preconditions	1. Include (Authentication) 2. User on the update profile screen
Main Flow	1. User presses profile picture upload button 2. User selects a file which is a valid image type 3. File is uploaded and changed on screen
Post Conditions	1. New profile picture successfully updated, and data stored

Use Case ID	11
Use Case	Update Privacy of Profile Details
Description	Update privacy of user's details that are stored on file
Primary Actors	Student
Preconditions	1. Include (Authentication) 2. User on the update profile screen
Main Flow	1. User selects privacy dropdown next to relevant data 2. User updates privacy to their preferred choice
Post Conditions	1. New privacy settings successfully updated

Use Case ID	12
Use Case	Update User Details
Description	Update a user's details that are stored on file
Primary Actors	Student
Preconditions	1. Include (Authentication) 2. User on the update profile screen
Main Flow	1. User edits the text box containing info they want to change 2. User presses save
Post Conditions	1. New details successfully updated, and data stored

Use Case ID	13
Use Case	Update Class Board
Description	Upload a post onto a class board
Primary Actors	Lecturer
Preconditions	1. Include (Authentication) 2. User is on shared boards tab
Main Flow	1. User selects add a new post 2. User uploads any relevant files into location in board's directory 3. User adds some text describing post 4. User presses post
Post Conditions	1. Successful upload of post 2. Any uploaded data stored successfully

Use Case ID	14
Use Case	Access Student Details
Description	Access public details on a student
Primary Actors	Lecturer
Preconditions	1. Include (Authentication) 2. User is on profile viewer
Main Flow	1. User types in relevant user's name 2. User selects profile
Post Conditions	1. Public profile details of selected user appear on screen

Use Case ID	15
Use Case	Create Sub-Group Board
Description	Create a sub-group board for a select group
Primary Actors	Lecturer, Main Office Staff or Student Union
Preconditions	1. Include (Authentication) 2. User is on shared board tab
Main Flow	1. User selects create new board button 2. User creates enters their board name 3. User enters their intended recipients/group
Post Conditions	1. Board successfully created

Use Case ID	16
Use Case	Send Out Notifications

Description	Send out a notification to a filtered group of recipients
Primary Actors	Lecturer, Main Office Staff or Student Union
Preconditions	1. Include (Authentication) 2. User is on notification page
Main Flow	1. User enters message 2. User adds optional media files 3. Include (Identifying Recipients) 4. User presses send
Post Conditions	1. Recipients receive notification

Use Case ID	17
Use Case	Identifying Recipients
Description	Identify the recipients who will receive a push notification
Primary Actors	Lecturer, Main Office Staff or Student Union
Preconditions	1. Include (Authentication) 2. Notification is being prepared
Main Flow	1. Users are searched for by name 2. Filters are applied through bar next to search bar

Use Case ID	18
Use Case	Add New Event to Shared Calendar

Description	Add a new event to a calendar which is shared with multiple recipients
Primary Actors	Lecturer, Main Office Staff or Student Union
Preconditions	1. Include (Authentication) 2. User is on shared calendar tab
Main Flow	1. User opens relevant calendar 2. User selects add new event button 3. User adds title of event 4. User adds deadline 5. User adds priority 6. User presses add
Post Conditions	1. New event added to shared calendar 2. All members of shared calendar can view event

Use Case ID	19
Use Case	Lecturer Registration
Description	Create a lecturer account that has extended permissions
Primary Actors	Administrator
Preconditions	Include (Authentication)
Main Flow	1. Administrator opens lecturer registration page 2. Include (Set Account Permissions)

	3. Include (Update Lecturer Class Assignments)
Post Conditions	1. New lecturer profile created

Use Case ID	20
Use Case	Update Lecturer Class Assignments
Description	Add lecturer class assignments to an account
Primary Actors	Administrator
Preconditions	1. Include (Authentication)
Main Flow	1. Search for existing class 2. Class selected or option for adding new class 3. Class details are verified 4. Administrator presses add
Post Conditions	1. New class added to lecturer profile 2. Board and shared calendar created for class

Use Case ID	21
Use Case	Set Account Permissions
Description	Set relevant extended permissions to an account
Primary Actors	Administrator
Preconditions	Include (Authentication)

Main Flow	1. Permission levels selected using a dropdown menu next to an account function 2. Repeat step 1 for each individual account function
Post Conditions	1. New class added to lecturer profile

Use Case ID	22
Use Case	Office/Student Union Registration
Description	Create an office/union registration account that has extended permissions
Primary Actors	Administrator
Preconditions	1. Include (Authentication)
Main Flow	1. Administrator opens office/union registration page 2. Include (Set Account Permissions)
Post Conditions	1. New office/union profile created