**ANDROID MOBILE APP COMPANION**

**FOR UNIVERSITY STUDENTS**

**CHING MING SIEW**

**A project report submitted in partial fulfilment of the**

**requirements for the award of Bachelor of Science**

**(Hons.) Software Engineering**

**Lee Kong Chian Faculty of Engineering and Science**

**Universiti Tunku Abdul Rahman**

**April 2018**

TABLE OF CONTENTS

[TABLE OF CONTENTS ii](#_Toc471743933)

[LIST OF TABLES iv](#_Toc471743934)

[LIST OF FIGURES v](#_Toc471743935)

[LIST OF SYMBOLS / ABBREVIATIONS vi](#_Toc471743936)

[LIST OF APPENDICES vii](#_Toc471743937)

**CHAPTER**

[1 INTRODUCTION 7](#_Toc487451512)

[1.1 Background of the problem 7](#_Toc487451513)

[1.2 Problem Statement **7**](#_Toc487451514)

[1.3 Project Objectives 8](#_Toc487451515)

[1.4 Proposed Solution 8](#_Toc487451516)

[1.5 Proposed Approach 9](#_Toc487451517)  
1.6 Scope of the Project 9

[2 LITERATURE REVIEW 1](#_Toc487451518)1

[2.1 Introduction 1](#_Toc487451519)1

[2.2 Literature Review](#_Toc487451520) 11

[2.2.1 Methodology Comparison 11](#_Toc487451522)

[2.2.2 Android Framework Comparison **17**](#_Toc487451522)

[2.2.3 Market Similar Application Comparison](#_Toc487451522) 21

[2.4 Summary 28](#_Toc487451523)

[3 METHODOLOGY AND WORK PLAN 2](#_Toc487451524)9

[3.1 Introduction 2](#_Toc487451525)9

[3.2 Work Plan 30](#_Toc487451526)

[3.2.1 Work Breakdown Structure](#_Toc487451527)

[3.2.2 Gantt Chart 31](#_Toc487451527)

[4 PROJECT INITIAL SPECIFICATION](#_Toc487451529) 32

[4.1 Introduction 32](#_Toc487451530)

[4.2 Prototype 3](#_Toc487451531)2

[REFERENCES](#_Toc487451535) 38

LIST OF TABLES

[Table 1: Framework Comparison 21](#_Toc472528047)

LIST OF FIGURES

Figure 1.1: Agile Methodology Life Cycle 13

Figure 1.2: Scrum Methodology Life Cycle 15

Figure 1.3: fall Methodology Life Cycle 17

[Figure 2.1: Interface of Evernote Application 22](file:///C:\Users\ching\Desktop\FYPI_-SE-_Template-R4.docx#_Toc471748521)

[Figure 2.2: Interface of Evernote Application](file:///C:\Users\ching\Desktop\FYPI_-SE-_Template-R4.docx#_Toc471748521) 23

[Figure 2.3: Interface of Evernote Application 2](file:///C:\Users\ching\Desktop\FYPI_-SE-_Template-R4.docx#_Toc471748521)3

[Figure 3.1: Interface of School Planner Application 2](file:///C:\Users\ching\Desktop\FYPI_-SE-_Template-R4.docx#_Toc471748520)4

[Figure 3.2: Interface of School Planner Application 2](file:///C:\Users\ching\Desktop\FYPI_-SE-_Template-R4.docx#_Toc471748520)5

[Figure 3.3: Interface of School Planner Application 25](file:///C:\Users\ching\Desktop\FYPI_-SE-_Template-R4.docx#_Toc471748520)

[Figure 4.1: Interface of My Study Life Application 26](file:///C:\Users\ching\Desktop\FYPI_-SE-_Template-R4.docx#_Toc471748520)

[Figure 4.2: Interface of My Study Life Application 27](file:///C:\Users\ching\Desktop\FYPI_-SE-_Template-R4.docx#_Toc471748520)

[Figure 4.3: Interface of My Study Life Application 2](file:///C:\Users\ching\Desktop\FYPI_-SE-_Template-R4.docx#_Toc471748520)7

[Figure 5.1: WBS](file:///C:\Users\ching\Desktop\FYPI_-SE-_Template-R4.docx#_Toc471748520) 30

[Figure 5.2: Gantt Chart 31](file:///C:\Users\ching\Desktop\FYPI_-SE-_Template-R4.docx#_Toc471748520)

[Figure 6.1: Interface of Prototype](file:///C:\Users\ching\Desktop\FYPI_-SE-_Template-R4.docx#_Toc471748520) 33

[Figure 6.2: Interface of Prototype 33](file:///C:\Users\ching\Desktop\FYPI_-SE-_Template-R4.docx#_Toc471748520)

[Figure 6.3: Interface of Prototype 3](file:///C:\Users\ching\Desktop\FYPI_-SE-_Template-R4.docx#_Toc471748520)4

[Figure 6.4: Interface of Prototype 34](file:///C:\Users\ching\Desktop\FYPI_-SE-_Template-R4.docx#_Toc471748520)

[Figure 6.5: Interface of Prototype 3](file:///C:\Users\ching\Desktop\FYPI_-SE-_Template-R4.docx#_Toc471748520)5

[Figure 6.6: Interface of Prototype 35](file:///C:\Users\ching\Desktop\FYPI_-SE-_Template-R4.docx#_Toc471748520)

[Figure 6.7: Interface of Prototype 3](file:///C:\Users\ching\Desktop\FYPI_-SE-_Template-R4.docx#_Toc471748520)6

[Figure 6.8: Interface of Prototype 36](file:///C:\Users\ching\Desktop\FYPI_-SE-_Template-R4.docx#_Toc471748520)

[Figure 6.9: Interface of Prototype 3](file:///C:\Users\ching\Desktop\FYPI_-SE-_Template-R4.docx#_Toc471748520)7

[Figure 6.10: Interface of Prototype 37](file:///C:\Users\ching\Desktop\FYPI_-SE-_Template-R4.docx#_Toc471748520)

## INTRODUCTION

### Background of the Problem

University is a place where students come learning professional skill, social skill, and many more that get them ready before they get into social. Main focus of students or students’ goal are often getting good grades in subjects and graduate with flying colours. In Malaysia, studying in university often means that is the time for students to get independent, because most of the university students have to leave their parents and start an independent university life without guidance.

Without someone there to remind you want to do next, even some of the university students who graduated from their secondary school with excellent result end up performing bad in their university academic. One of the main reason causing this scenario to happen is because of their poor time management.

This project is to create an android mobile application that helps student to organise their time and keep user reminded of what should they be doing now or what is important at the moment instead of forgetting some minor task that should be completed as soon as possible.

### Problem Statement

These problem has been discovered through observation, survery and research.

#### Students are poor in managing their time

Balancing between acedemic, social life and sleep is never a simple task. University student often sleep irregularly and do most of their task at the last minute.(Ronald, 2016)

#### Students are pooi in managing their pocket money

Most of the university student especially those who sign for PTPTN failed to use their pocket money wisely. Most of the cases are, if the pocket money is expected to cover the expense for a month, student always overspending at the beginning of the month and cause short of pocket money at the end of the month.(Catherine, 2016)

#### Similar market application does not offer full package of function

There are good similar application on the market but their application does not offer full package of function that we needed. Instead of using 2 to 3 application to complete requested task, if a single application can do all the task that will be much convenient.

### Project Objectives

* To develop an application that helps students to organise their time and improve their time management.
* To develop an application that reduces student overspending problem.
* To develop an application that combines useful function from various application into one application to bring convenience to student.

### Proposed Solution

In order to solve the problems mentioned and achieve the objective of the project, I have proposed some potential solutions that is possible to be implemented.

One of the solution aiming at the student poor time management ability is an mobile application that could alert user when due date of an event is around the corner – exam, midterm and assignment event alert notification. This due date reminder requires user to set what is the event name, when will the event occur and when you want the application to alert you. Also it can show a list of next coming event as widget that could be pinned on user’s home screen as a widget. This widget feature also can be implemented to timetable storing function where users can refer to the widget and know which class they are going and where is the class located.

The proposed solution aiming at the second problem statement is expenditure planning feature built into the application which record the initial pocket money user getting every month and record every expenses that user spent. Also this expenditure planning feature generate graph at the end of the month which shows user’s spending, for example, 50% of the pocket money is spent on food and drinks in the past month. This can let users to have a clear idea in where their money went.

For market application does not offer full package feature problem, is simply combine more feature into one application to make it more suitable to be used by university students. This project will include more feature such as use camera to capture notes or use mic to record voice and upload into application based on subject, to-do list that records small event that need to be done, CGPA calculator which show student their current CGPA and also access to UTAR feedback system to give rating or comment to facilities, services, lecturers, tutors and etc.

### Proposed Approach

For this project I will be using Agile methodology which dividing software development task into distinct phases and build application in short cycles. This can improve precision of program flow function by function. Limitation of mobile application compare to using computer to do all these task are screen size, memory, and performance of mobile phone. Therefore interface and program flow design have to be extra compact in order to run smooth on majority of the current mobile phone. Flexibility, countinuons improvement and fast delivery time is what Agile focus, it embrace changes from client from time to time to make sure client satisfied with their product. For this Android based project, the main application that I will be using to complete this project is Android Studio and with the help of React Native fragment.

### Scope of the Project

#### Context

This project is focus on the development an android application for UTAR students to organise their time, expenses, learning source and bring convenience to students.

#### Users

The users of the application is aiming for UTAR students, because it includes feedback to UTAR feedback system which students need to sign in with their UTAR ID and password to get access to this feature.

#### Features

Features that will be included in the mobile application.

* Timetable storing feature.
* Timetable showing widget.
* Event (midterm, final exam and assignment) alert notification feature.
* Event (midterm, final exam and assignment) countdown widget.
* Capture notes and upload to application feature.
* Record voice and upload to application feature.
* Feedback to facilities, services, lecturer and tutor feature.
* To-do list feature.
* Expenditure planning.

## LITERATURE REVIEW

### Introduction

When starting a mobile development, few questions may be raised. What differentiates mobile application development and traditional application development? What options are available? What are the important design principles for mobile application? These questions are answered as each literature is reviewed in the following sections: software engineering issues in mobile development; comparison between native application and mobile web application; iOS and Android operating system; and human-computer interaction in mobile(Noh, 2014).

### Literature Review

#### Methodology Comparison

Methodology in software engineering refer to a process method which dividing software development task into distinct phases to guild developer what is the next task after their current, to improve software design, software stability, product management and project management. In other words methodology is also known as software development life cycle. There are many methodology in software development for example, Agile, Waterfall, Scrum and etc. By choosing methodology decide what process method developer want to develop their software, and choosing a suitable methodology could greatly improve the effectiveness of building a software development project.

Agile Methodology

Instead of heavy planning at the beginning of the project, Agile methodology is in view of an incremental, iterative approach. Agile methodologies are open to changing requirements over time and encourages constant feedback from the end users. Cross-functional teams work on iteration is to produce a working product.

Advantages of using Agile:

* **Change is embraced**: With shorter planning cycles, it’s easy to accommodate and accept changes at any time during the project. There is always an opportunity to refine and reprioritize the backlog, letting teams introduce changes to the project in a matter of weeks.
* **End-goal can be unknown**: Agile is very beneficial for projects where the end-goal is not clearly defined. As the project progresses, the goals will come to light and development can easily adapt to these evolving requirements.
* **Faster, high-quality delivery**: Breaking down the project into iterations (manageable units) allows the team to focus on high-quality development, testing, and collaboration. Conducting testing during each iteration means that bugs get identified and solved more quickly. And this high-quality software can be delivered faster with consistent, successive iterations.
* **Strong team interaction**: Agile highlights the importance of frequent communication and face-to-face interactions. Teams work together and people are able to take responsibility and own parts of the projects.
* **Customers are heard:** Customers have many opportunities to see the work being delivered, share their input, and have a real impact on the end product. They can gain a sense of ownership by working so closely with the project team.
* **Continuous improvement**: Agile projects encourage feedback from users and team members throughout the whole project, so lessons learned are used to improve future iterations.

Disadvantages of using Agile:

* **Planning can be less concrete**: It can sometimes be hard to pin down a solid delivery date. Because Agile is based on time-boxed delivery and project managers are often reprioritizing tasks, it’s possible that some items originally scheduled for delivery may not be complete in time. And, additional sprints may be added at any time in the project, adding to the overall timeline.
* **Team must be knowledgeable:** Agile teams are usually small, so team members must be highly skilled in a variety of areas. They also must understand and feel comfortable with the chosen Agile methodology.
* **Time commitment from developers:** Agile is most successful when the development team is completely dedicated to the project. Active involvement and collaboration is required throughout the Agile process, which is more time consuming than a traditional approach. It also means that the developers need to commit to the entire duration of the project.
* **Documentation can be neglected:**The Agile Manifesto prefers working software over comprehensive documentation, so some team members may feel like it’s less important to focus on documentation. While comprehensive documentation on its own does not lead to project success, Agile teams should find the right balance between documentation and discussion.
* **Final product can be very different:** The initial Agile project might not have a definitive plan, so the final product can look much different than what was initially intended. Because Agile is so flexible, new iterations may be added based on evolving customer feedback, which can lead to a very different final deliverable.

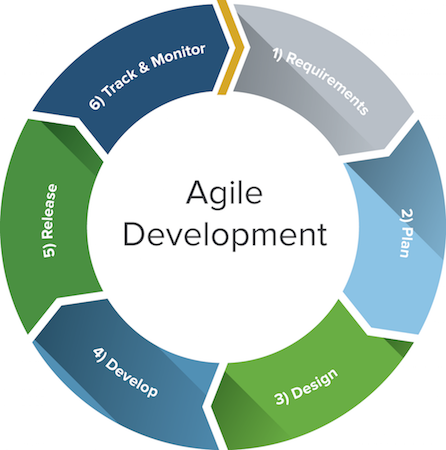


Figure 1.1: The Agile Development Cycle

Scrum Methodology

Scrum is a subset of Agile and one of the most popular process frameworks for implementing Agile. It is an iterative software development model used to manage complex software and product development. Fixed-length iterations, called sprints lasting one to two weeks long, allow the team to ship software on a regular cadence. At the end of each sprint, stakeholders and team members meet to plan next steps.

Advantages of using Scrum:

* **More transparency and project visibility:** With daily stand-up meetings, the whole team knows who is doing what, eliminating many misunderstandings and confusion. Issues are identified in advance, allowing the team to resolve them before they get out of hand.
* **Increased team accountability:** There is no project manager telling the Scrum Team what to do and when. Instead, the team collectively decides what work they can complete in each sprint. They all work together and help each other, improving collaboration and empowering each team member to be independent.
* **Easy to accommodate changes:** With short sprints and constant feedback, it’s easier to cope with and accommodate changes. For example, if the team discovers a new user story during one sprint, they can easily add that feature to the next sprint during the backlog refinement meeting.
* **Increased cost savings:** Constant communication ensures the team is aware of all issues and changes as soon as they arise, helping to lower expenses and increase quality. By coding and testing features in smaller chunks, there is continuous feedback and mistakes can be corrected early on, before they get too expensive to fix.

Disadvantages of using Scrum:

* **Risk of scope creep**: Some Scrum projects can experience scope creep due to a lack of specific end date. With no completion date, stakeholders may be tempted to keep requesting additional functionality.
* **Team requires experience and commitment:** With defined roles and responsibilities, the team needs to be familiar with Scrum principles to succeed. Because there are no defined roles in the Scrum Team (everyone does everything), it requires team members with technical experience. The team also needs to commit to the daily Scrum meetings and to stay on the team for the duration of the project.
* **The wrong Scrum Master can ruin everything:** The Scrum Master is very different from a project manager. The Scrum Master does not have authority over the team; he or she needs to trust the team they are managing and never tell them what to do. If the Scrum Master tries to control the team, the project will fail.
* **Poorly defined tasks can lead to inaccuracies:** Project costs and timelines won’t be accurate if tasks are not well defined. If the initial goals are unclear, planning becomes difficult and sprints can take more time than originally estimated.

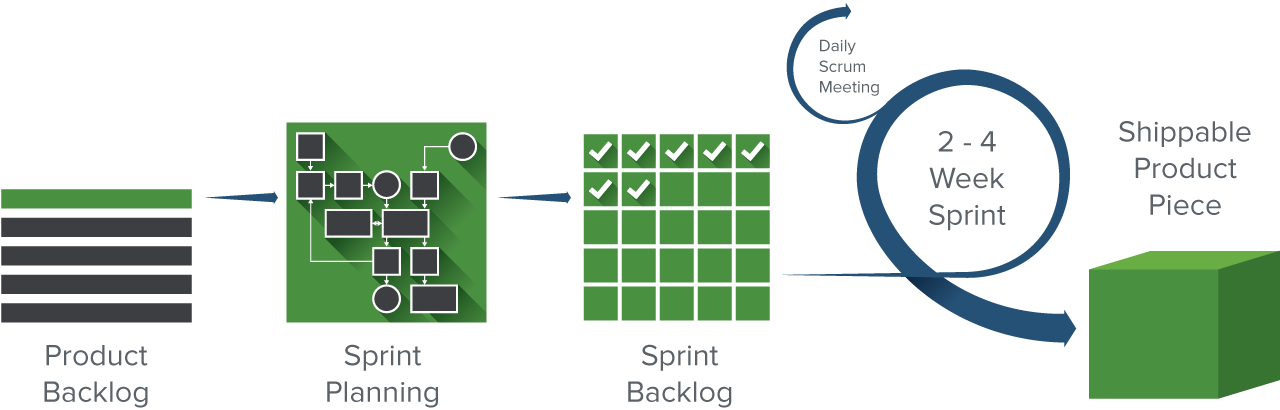


Figure 1.2: The Scrum Development Cycle

Waterfall Methodology

Waterfall methodology follows a sequential, linear process and is the most popular version of the systems development life cycle (SDLC) for software engineering and IT projects. It is sometimes planned using a Gantt chart, a type of bar chart that shows the start and end dates for each task. Once one of the eight stages are complete, the development team moves onto the next step. The team can’t go back to a previous stage without starting the whole process from the beginning. And, before the team can move to the next stage, requirements may need to be reviewed and approved by the customer.

Advantages of using Waterfall:

* **Easy to use and manage:** Because the Waterfall model follows the same sequential pattern for each project, it is easy to use and understand. The team doesn’t need any prior knowledge or training before working on a Waterfall project. Waterfall is also a rigid model; each phase has specific deliverables and review, so it’s easy to manage and control.
* **Discipline is enforced**: Every phase in Waterfall has a start and end point, and it’s easy to share progress with stakeholders and customers. By focusing on requirements and design before writing code, the team can reduce the risk of a missed deadline.
* **Requires a well documented approach:** Waterfall requires documentation for every phase, resulting in better understanding of the logic behind the code and tests. It also leaves a paper trail for any future projects or if stakeholders need to see more detail about a certain phase.

Disadvantages of using Waterfall:

* **Changes can’t be easily accommodated:** Once the team completes a phase, they can’t go back. If they reach the testing phase and realize that a feature was missing from the requirements phase, it is very difficult and expensive to go back and fix it.
* **Software isn’t delivered until late**: The project has to complete two to four phases before the coding actually begins. As a result, stakeholders won’t see working software until late in the life cycle.
* **Gathering accurate requirements can be challenging:** One of the first phases in a Waterfall project is to talk to customers and stakeholders and identify their requirements. However, it can be difficult to pinpoint exactly what they want this early in the project. Often times, customers don’t know what they want early on and instead, learn and identify requirements as the project progresses.

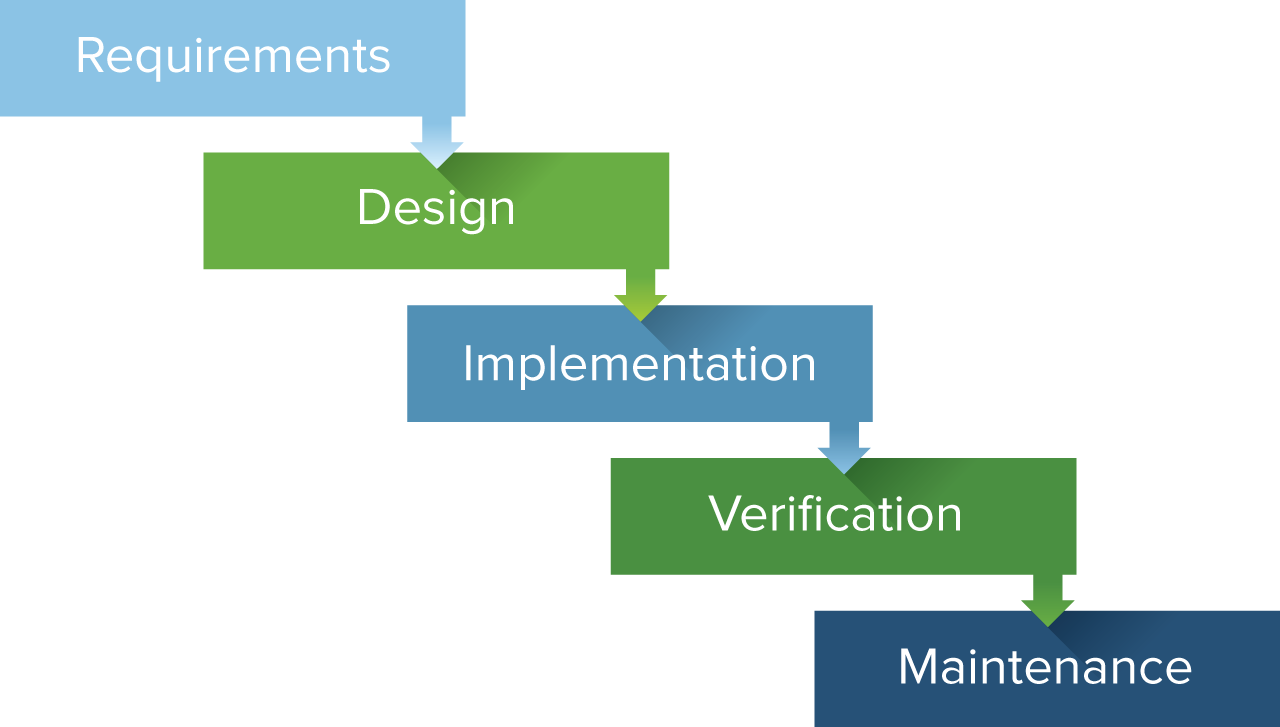


Figure 1.3: The Waterfall Development Cycle

#### Android Framework Comparison

The framework serves as a support or guild in program building, a framework may be a set of functions that did specific task.  Instead of using a lot of times writing code with own logics such as exception-handling logic, security functions, and own unit test function, there are better way to do it which is by implementing the framework. The programmer can simply import an exception handling framework, a security function framework or unit test framework which is well tested and structured into the program, this can greatly reduce the workload of programmer and save a lot of times. By implementing framework makes programmer work with complex technologies easier, and makes overall program more consistent in both interface and coding, fewer bugs, and more flexible (Clifton, 2003).

            There are a bunch of frameworks can be chosen from. But, it also makes the framework user hard to choose whether which framework is best for their program. Before choosing the right framework, the programmer has to know which type of application and which platform they want to build. Android, IOS and windows are a list of examples of platform and native app development, web app development and hybrid app are list of main types of application types.

Native Application

Native app development is an application program that has been developed use on a particular platform or device (Rouse, 2013). Advantages of native apps are high performance and high response between the interaction of user and application, this is because native apps are developed specifically for a mobile operating system (Swift for IOS and Java for Android) and always results in the best user experience (Madden, 2016). Others than better performance and functionality, native apps also always have a better user interface due to automatically inherit the operating system’s user interface makes the user feel those native apps like an integrated a part of the device. Downsides of native apps would be cost and time if this developer wants to cross their application to another platform, because developers need to create an application form to ground up in other programming language in order to make it run at other platform (Viswanathan, 2017).

Web Application / HTML5 Application

Web app development, is the other type of Android application which resides on the server and need to accessed via Internet. Web apps usually have to load the application from server everytime user want to use. HTML and CSS are commonly used on this type of application with the interactive parts in Java. Web apps are developed to work interchangeably between different operating system or platform as long as those devices can surf the Web or access internet (Aghion, 2014). Advantages of developing a web application instead of native application are they are easy to install, save device’s storage and always up-to-date due to it’s partially install on the device, partitions on server, and have to refresh form time to time to get the latest update. Disadvantages of Web apps are it would not work without internet, data is less secure when it’s in the cloud, and of course, it would not provide instant response than a native application because it needs to read and write data from and to the cloud (Fox, 2015). Many Web apps are designed to be as light as possible to increase the performance and respond of the application, therefore it is often have a simpler user interface or less functionality than native apps. Also there are features that cannot be done with web apps, which is, for example, trigger notification, running web apps in background, access accelerometer information from device, complex gestures and etc (Budiu, 2013).

Hybrid Application

Hybrid apps development hybrid apps are part native apps and part web apps. Similar to native apps, it has to be installed on your device, and similar to web apps, it contains the same HTML code components which can be reused on different platform (Budiu, 2013). Hybrid apps are often primarily built as a web app, then wrapped in a thin native container that provides access to native platform features (Korf and Oksman, 2016).

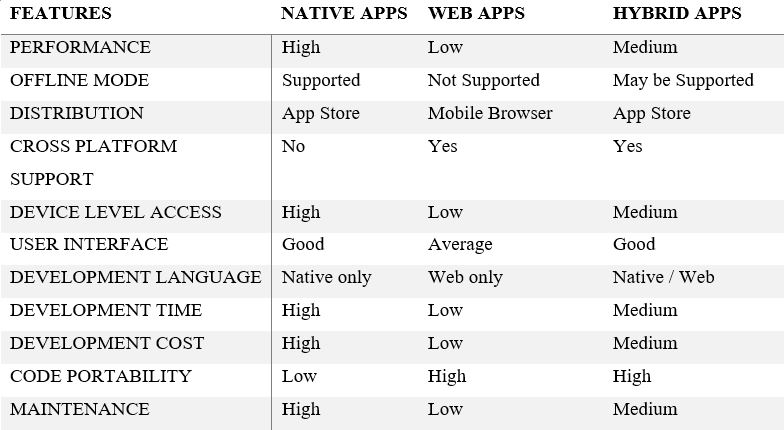


Table 1: Framework Comparison

After selected type of application, there are 4 main points to consider when choosing your mobile development platform which are from framework speed considerations, feature of framework support, framework risk and also development cost. Xamarin, Ionic and React Native are 3 popular tools which provide framework support.

Xamarin

Xamarin is a tool that has emerged as one of the key mobile app development technologies. Xamarin uses C# to target multiple devices which allow programmers to reuse about 90% of code across major platforms. It is  cross-platform technology for native mobile application development. Xamarin can create business mobile applications that looks and feels like native apps. Although Xamarin is a comparatively new technology, but it still a community of more than 1.4million developers from across the world. The pros of using Xamarin are it has Microsoft backing which is that biggest tech giants in the world, over 70% of the code is reusable between iOS and Android, which greatly reduce development cost and Xamarin almost have the same level of performance as from a natively developed app. Cons of using Xamarin are its cost, with the Business license starting at 999USD which are very expensive. Xamarin only supports specific technological components that are available within its library and application crated through Xamarin are overcomplicated to allow that smooth integration (Pathan, 2017).

Ionic

Ionic framework is an open-source SDK for hybrid mobile app development, it is technically built on top of Apache Cordova. This framework integrates well with single page applications that are known as Angular. It is a framework that essentially allows developers to create hybrid mobile apps using web technologies, like HTML, CSS and HavaScript (Alexseyanki, 2017). The pros of using Ionic are programmer only need to develop an application once and it can be deployed across iOS, Android and Windows devices, the use of Angular creates a powerful SDK for building rich and robust applications and since Ionic is focused on HTML, CSS and JavaScript, it enables quick development, low cost and minimal maintenance. Cons of using Ionic are the in-app performance may not be as swift as if the application were developed natively for each device, the use of Angular demands that developers have a specific skill set for building complex apps, and building in-app navigation can be notably complex, as the router is tricky to maneuver at present.

React Native

React Native is different from Ionic and Xamarin, it doesn’t build a web app, a HTML5 app or hybrid app, it builds a real mobile app that’s indistinguishable from an app built using Objective-C or Java. Instead of developing hybrid apps which run in a web-view, the main purpose of this framework is to build native apps. Fundamental user interface building blocks as regular iOS and Android apps are integrated into React Native apps and high-performance mobile app that is indistinguishable from one that is built using Swift, Objective-C or Java. React Native allows a proportion of the code to be shared between platforms and empowers developers to create apps which feel less clunky and perform better than hybrid apps. Pros of React Native are it provides numerous ready-to-apply components that can accelerate the development process and it enables copying the code base between iOS and Android makes cross-platform easier. Cons of React Native are it still lacks navigation components to provide users with a seamless user experience, React Native still lacks some components despite its maturity, and since it is operating on Facebook’s license, Facebook has a right to revoke the license (BSD) to use React and React Native if you get into a patent-related dispute with them (Chrzanowska, 2017).

#### Market Similar Application Comparison

There are many applications on the Google Play Store which can offer great help for university student to keep their study life, including time management, expenditure management and study materials organized. By comparing the strength and weakness of similar application could lead this project in a more precise direction. Evernote, School Planner and My Study Life are three applications which are highly rated and over 1 million downloads from Play store. After installing and experienced the functionality, user interface and program flow, below are reviews and opinion from me and other users.

Evernote

Evernote is basically an application that help user to take note literary everything thanks to it provide a bunch of input method.From using on-screen keyboard to type in, use camera to capture, use mic to record audio, attach an attachment, and handwriting to sketch Evernote have it all. Also, it provides a reminder function that allows user to enter event name, event description and when users want it to notify user about these events. Evernote support multi platform, including Android, iOS and Windows. Once the user uploads their notes into the cloud through Evernote, those notes can be synced easily across computer, phone or tablet. With crossing multi platform, store material in the cloud and a variety of methods to store users document Evernote can easily be the best notes taking application.

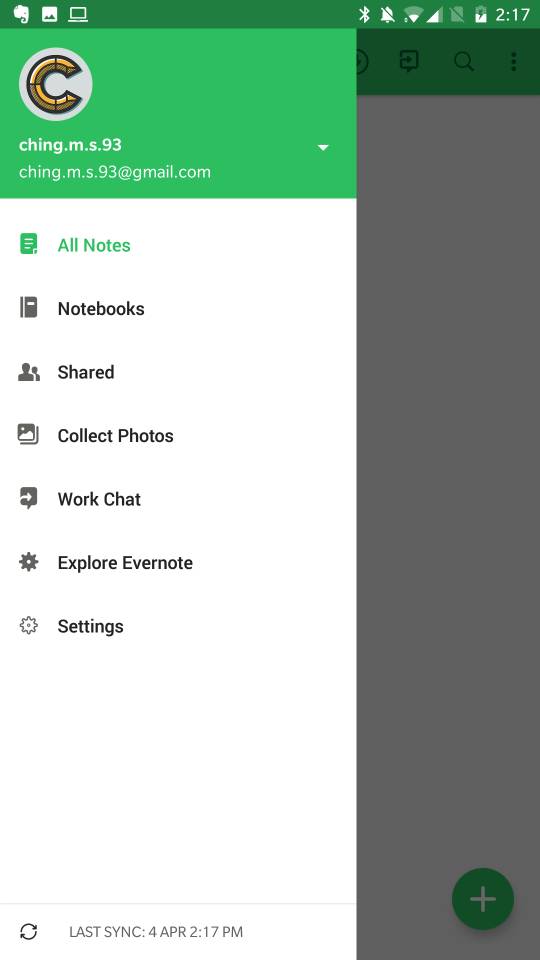


Figure 1.1: Interface of Evernote Application

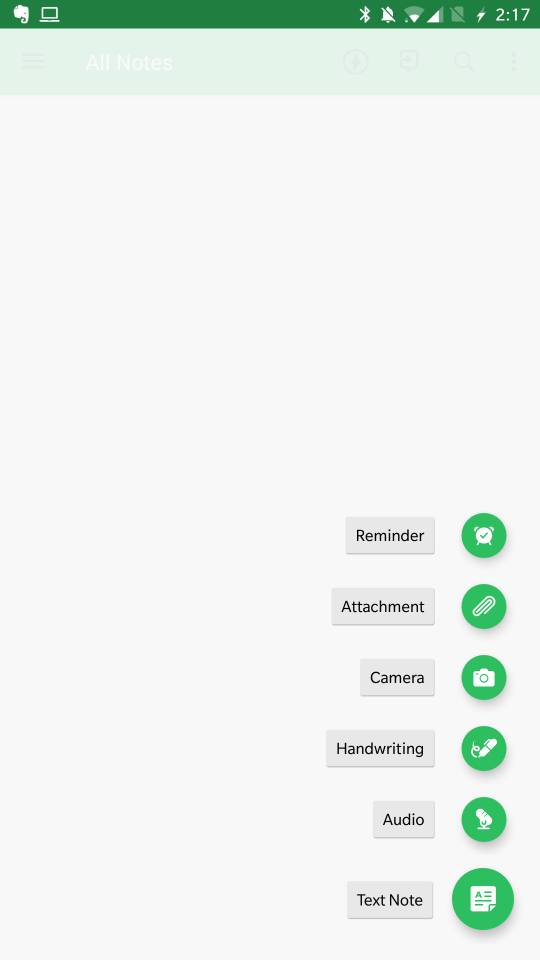


Figure 1.2: Interface of Evernote Application

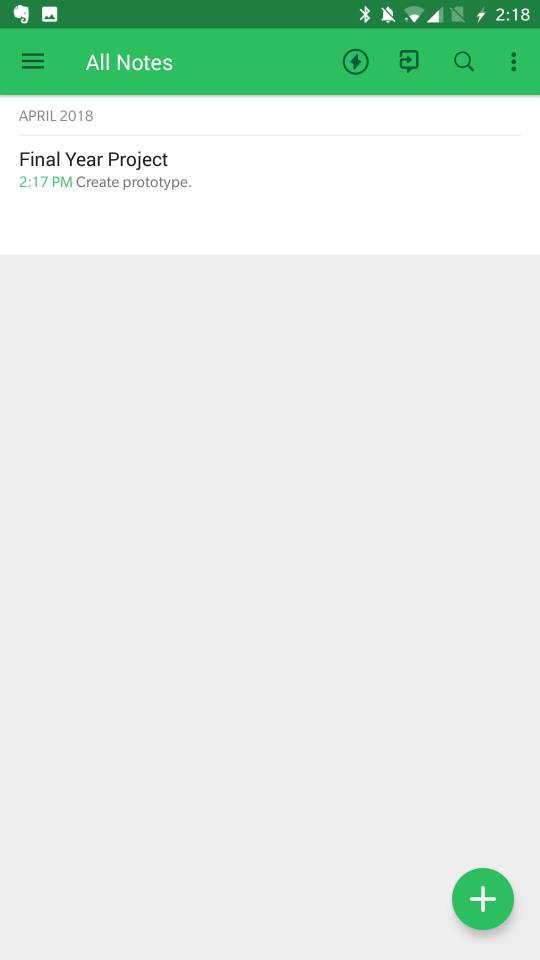


Figure 1.3: Interface of Evernote Application

School Planner

School Planner’s idea is very close to this project, it has feature like timetable, CGPA calculator, reminder and future planning in calendar or agenda view and also voice recorder feature. If compare to Evernote as a notes taking application, Evernote is still the best due to its variety method to input your notes, but School Planner has more feature like CGPA calculator and attendance checking feature which makes it a more suitable application for university students. When comes to portability this application has Google Drive Backup support but the downside is School Planner only support Android devices.

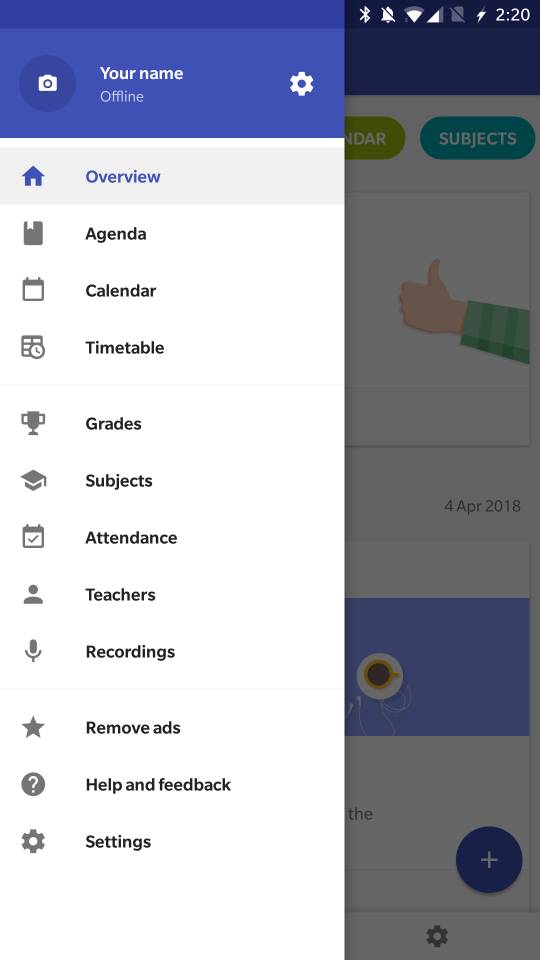


Figure 2.1: Interface of School Planner Application

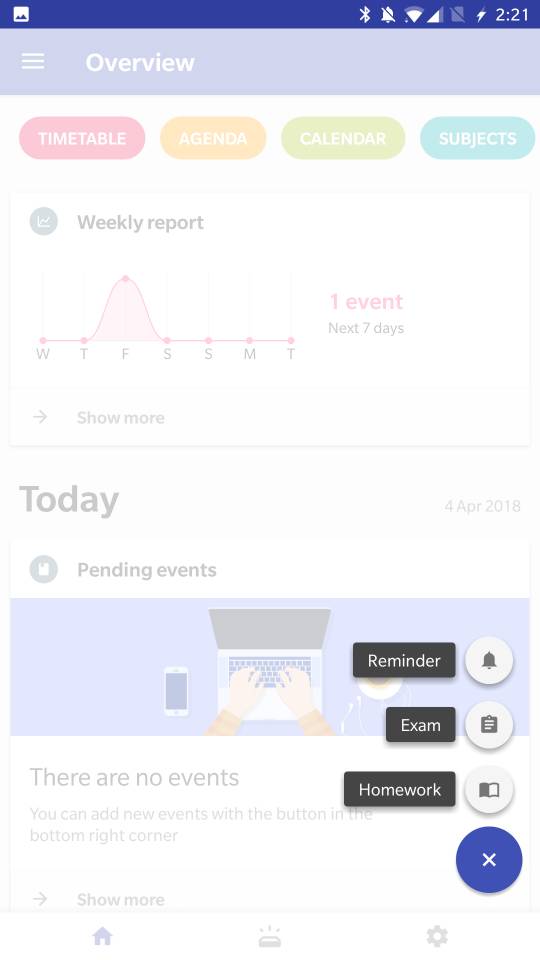


Figure 2.2: Interface of School Planner Application

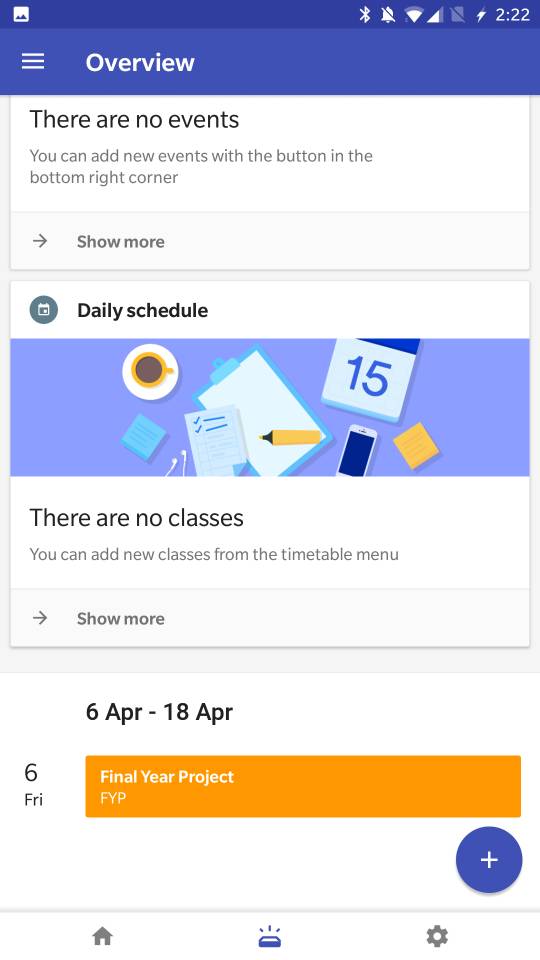


Figure 2.3: Interface of School Planner Application

My Study Life

My Study Life provide features like task and exam reminder, these reminders can be viewed in calendar view. It also has timetable feature, but comparing to School Planner, My Study Life is lack of feature. But different user may have different needs, if user wants an all round study planner, then School Planner is the one, but if the user wants a simple and light application for reminding user what to do next, My Study Life can fulfil these requests perfectly either. Also, other advantages of using My Study Life are, this application able to works on different major platforms like Android, iOS and Windows. User can easily view their schedules from different devices which using different platform.

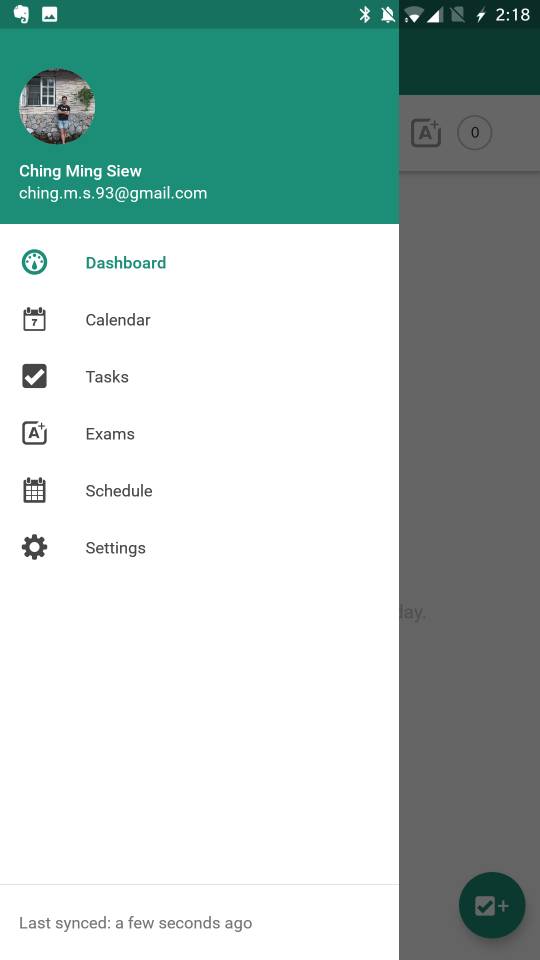


Figure 3.1: Interface of My Study Life Application

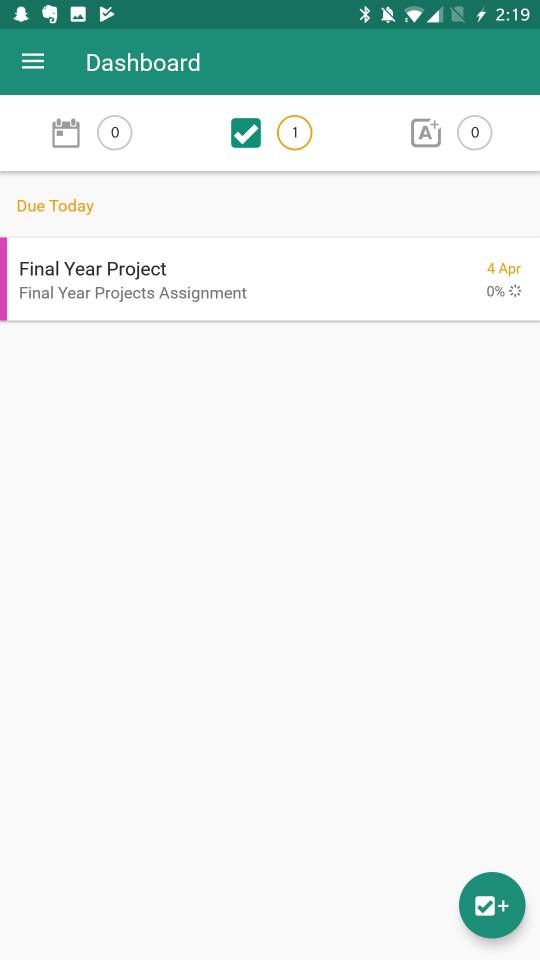


Figure 3.2: Interface of My Study Life Application

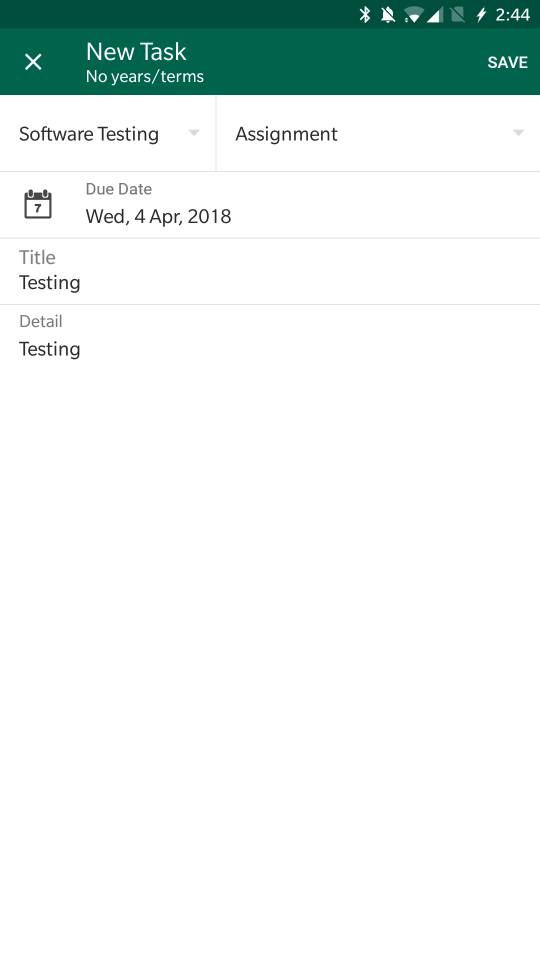


Figure 3.3: Interface of My Study Life Application

### Summary

In short, Agile methodology is best when developer doesn’t have a clear direction and they embranced to changes at any statre of development. That’s thereason why Android Movile App Companion fir University Students is more suitable to use Agile Methodology, because there are still many uncertainty inside this Final Year Project especially the retrive and update data from and to UTAR feedback system. Other than that, since this project is focus more on single platform, which is Android, a solid native application will be a better choice other than web application and hybrid application, React Native will be implement into this Final Year Project since it is recommended by most of the developers which provide aids in software developing.

## METHODOLOGY AND WORK PLAN

### Introduction

Agile development, in its simplest form, offers a lightweight framework for helping teams, given a constantly evolving functional and technical landscape, maintain a focus on the rapid delivery of business value. As a result of this focus, the benefits of agile software development are that organizations are capable of significantly reducing the overall risk associated with software development. In particular, agile development accelerates the delivery of initial business value, and through a process of continuous planning and feedback, is able to ensure that value is continuing to be maximized throughout the development process. As a result of this iterative planning and feedback loop, teams are able to continuously align the delivered software with desired business needs, easily adapting to changing requirements throughout the process. By measuring and evaluating status based on the undeniable truth of working, testing software, much more accurate visibility into the actual progress of projects is available. Finally, as a result of following an agile process, at the conclusion of a project is a software system that much better addresses the business and customer needs.

### Work Plan

#### Work Breakdown Structure

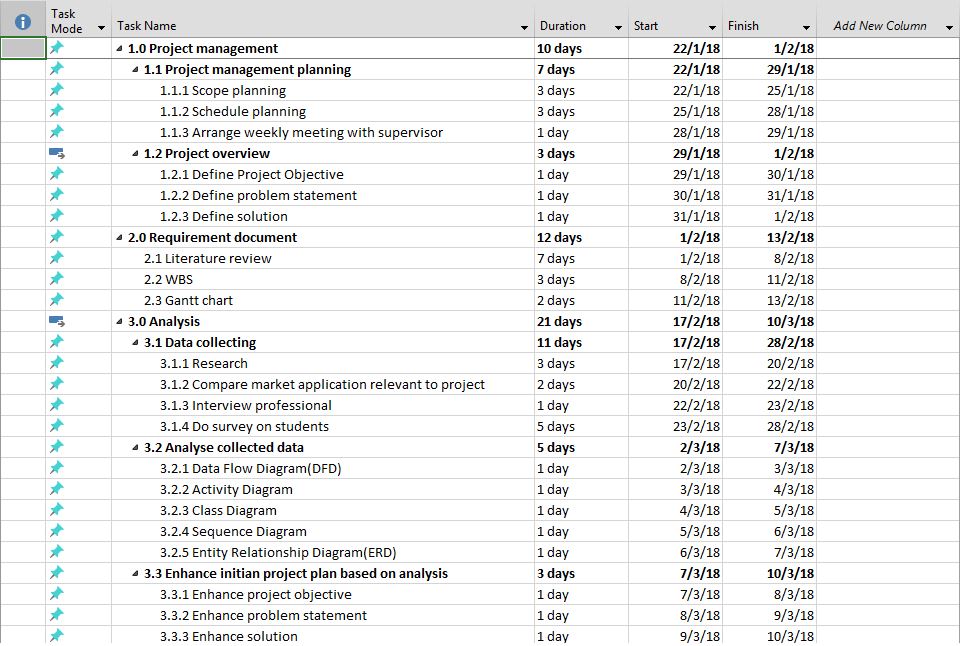


Figure 4.1: WBS

#### Gantt Chart

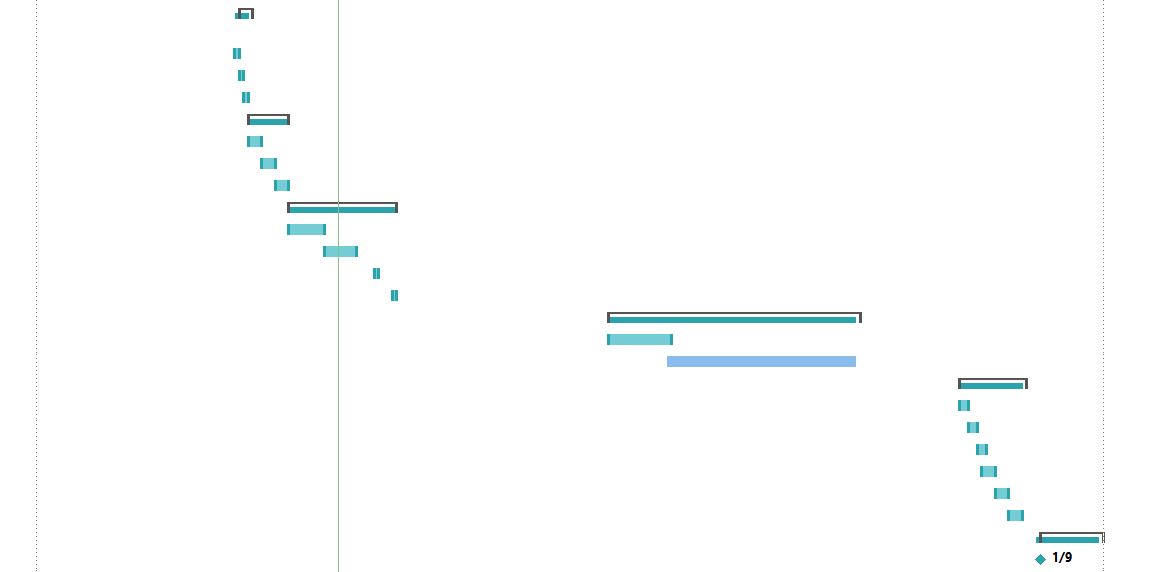
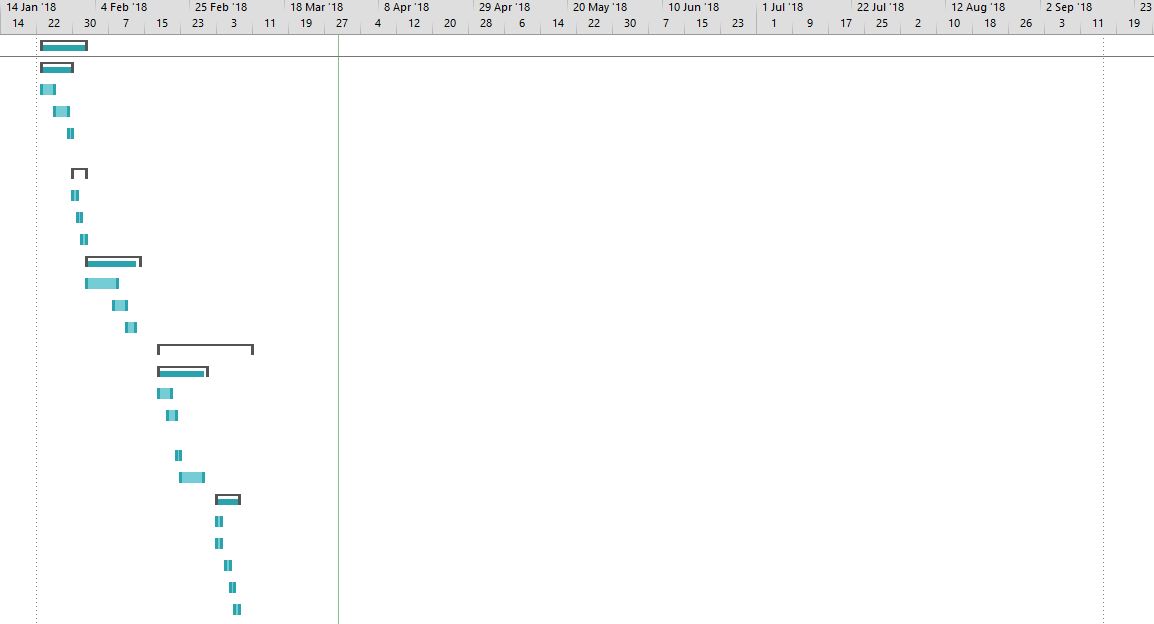


Figure 4.2 : Gantt Chart

## PROJECT SPECIFICATION

### Introduction

This Final Year Project, Android Mobile App Companion for University Student is proposed by Mr Chan Kok Leong. What Mr Chan wants is an application that can store the student's timetable, set reminders for exams, upload notes into application taken by a mobile camera, give feedback about university facilities, and also store student expenses record. Android Studio was the choice of software to build this Final Year Project. My Final Year Project had separated these feature into 5 tabs, and using a Bottom Navigation Bar to navigate into each interface. There are Dashboard, Timetable, Subject, Feedback and also Expenditure Planning.

### Dashboard

The Dashboard is designed to store event details such as the event's title, event's type,  event's date and also event's time. These events are categorized into 3 types, which are Assignment type, Exam type and To-Do List type. There 4 tabs in the Dashboard interface, including "All" tab which show all of the events and sort event by event's date and time, "Assignments" tab which show only Assignment type events, "Exams" tab which show only Exam type events and "To-do Lists" tab which show only To-do List type events. The Dashboard allows users to create, read, update and delete events.

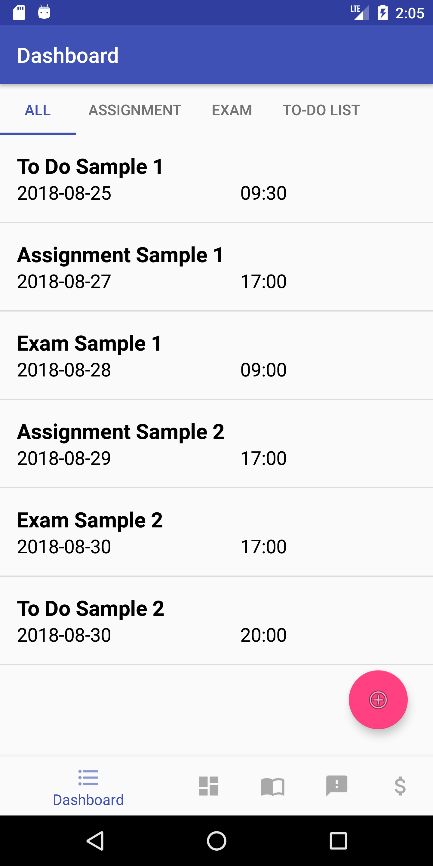


Figure 6.1: Interface of Dashboard – All Tab

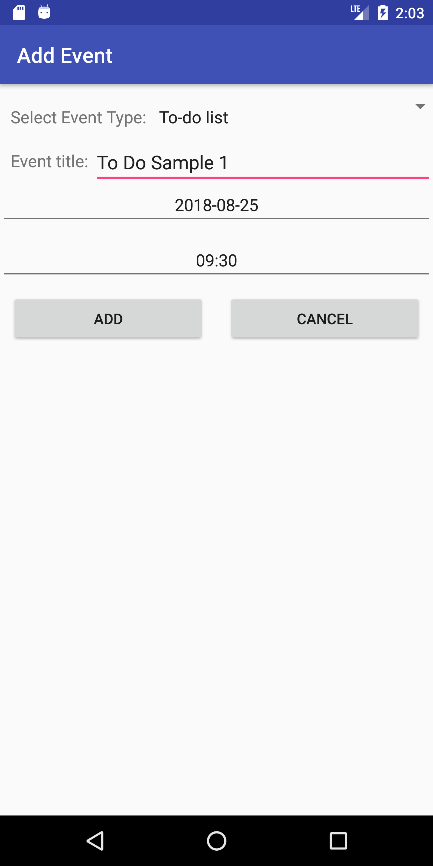


Figure 6.2: Interface of Dashboard – Add Event Menu

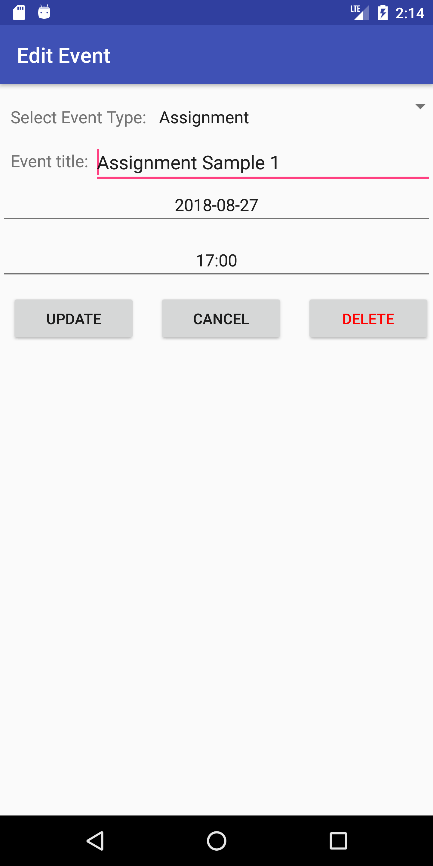


Figure 6.3: Interface of Dashboard – Edit Event Menu

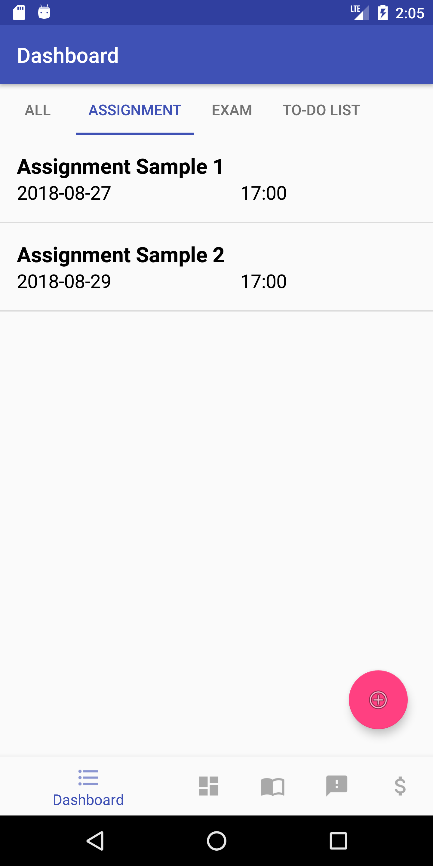


Figure 6.4: Interface of Dashboard – Assignment Tab

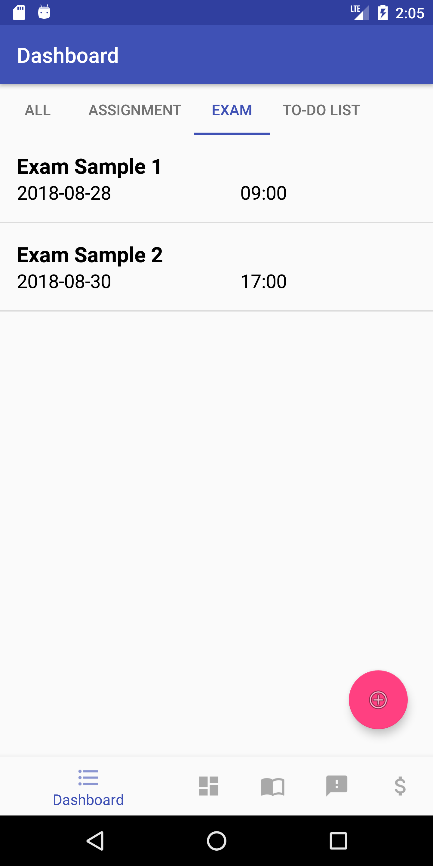


Figure 6.5: Interface of Dashboard – Exam Tab

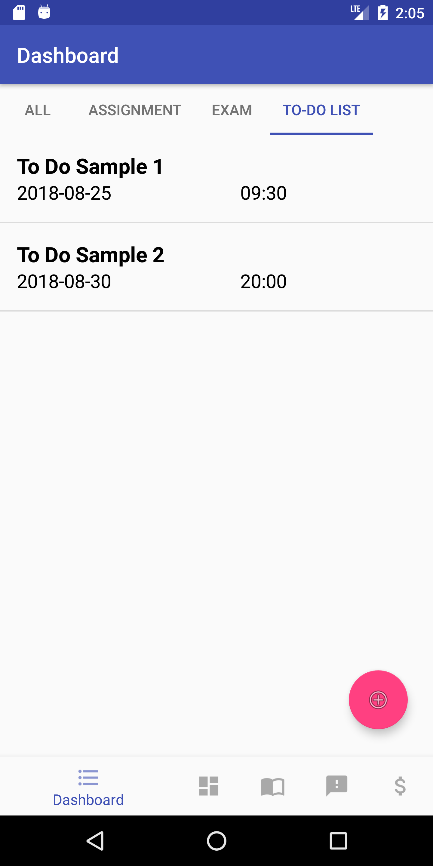


Figure 6.6: Interface of Dashboard – To-Do List Tab

### Timetable

The Timetable is designed to store user timetable details such as subject's title and time of subject's lessons. The Timetable is shown in the matrix table pattern which contains 5 columns as days from Monday to Friday, and 13 rows as time from 8am to 8pm. The Timetable allows users to create, read, update and delete classes.

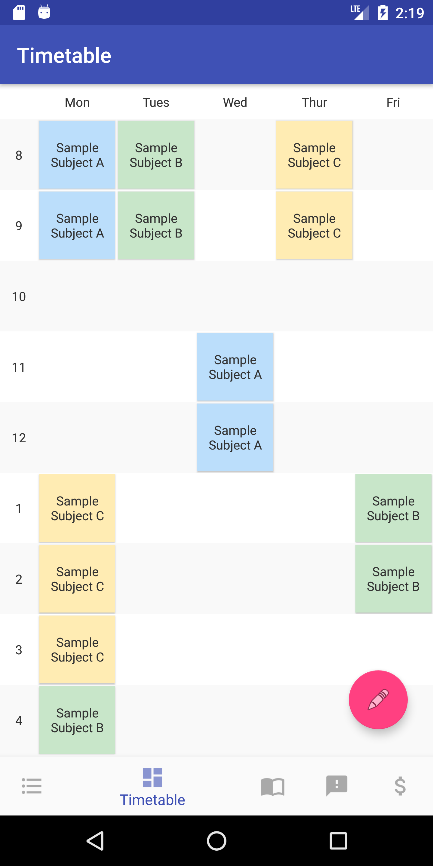


Figure 6.7: Interface of Timetable

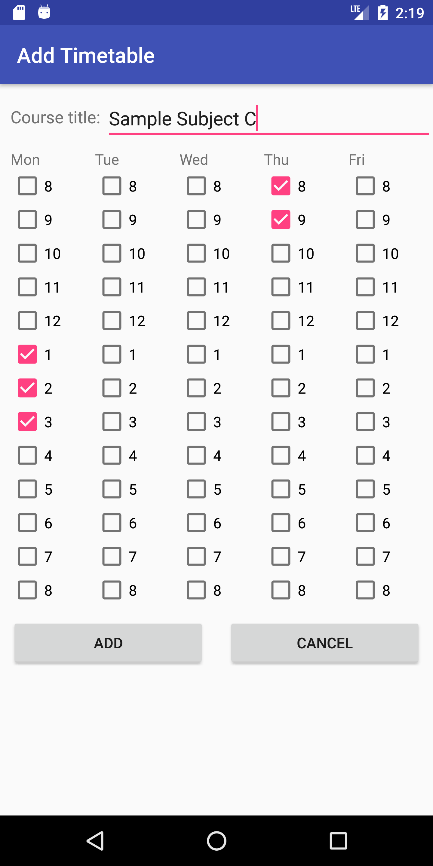


Figure 6.8: Interface of Timetable – Add Subject Menu

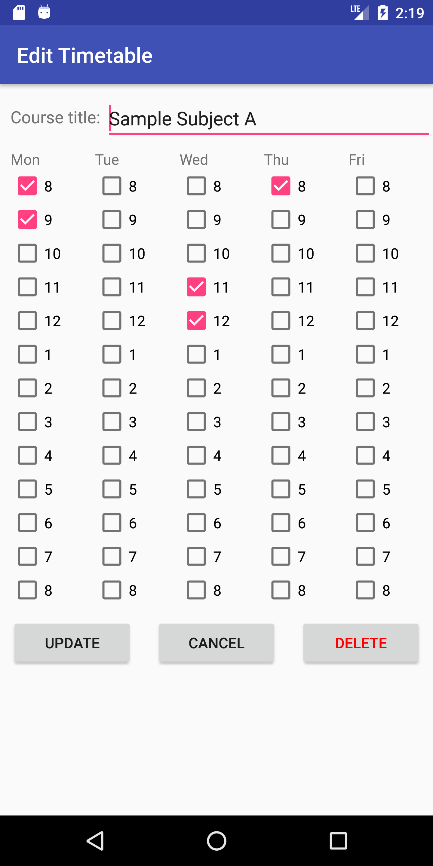


Figure 6.9: Interface of Timetable – Edit Subject Menu

### Subject

The Subject is designed to store subject's title as folder, and inside each subject folder allows users to store images to a specific subject folder either from the device's camera or select image from device's storage. The Subject allows users to create, read, update and delete subjects and images.

### Feedback

The Feedback allows users to give feedback to the university's facilities, courses, lecturers and tutors. Because of we cannot retrieve and write data from or into UTAR system, and different student has their own set of feedbacks. Therefore The Feedback basically redirects user to UTAR Portal to log in and complete the Student Survey. Users also allowed to check the period of Student Survey in The Feedback.



Figure 6.10: Interface of Feedback

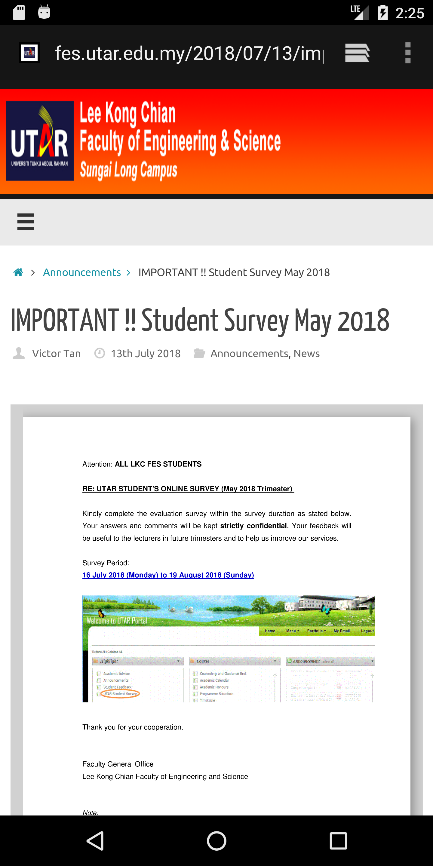


Figure 6.11: Interface of Feedback – Check Survey Period

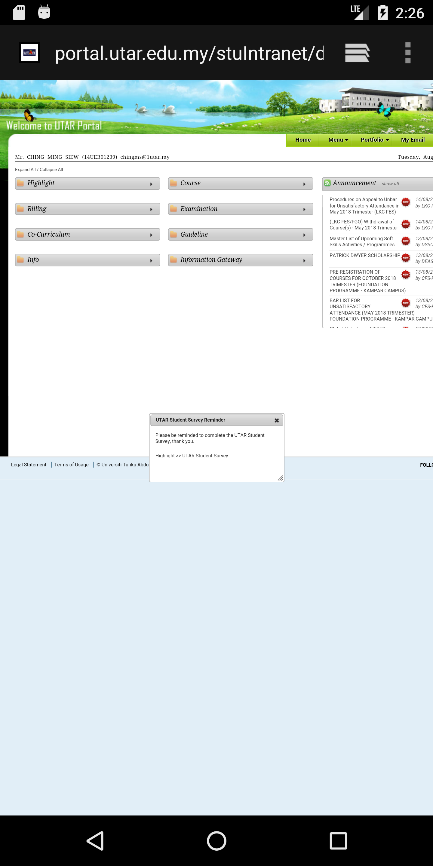


Figure 6.12: Interface of Feedback – UTAR Portal

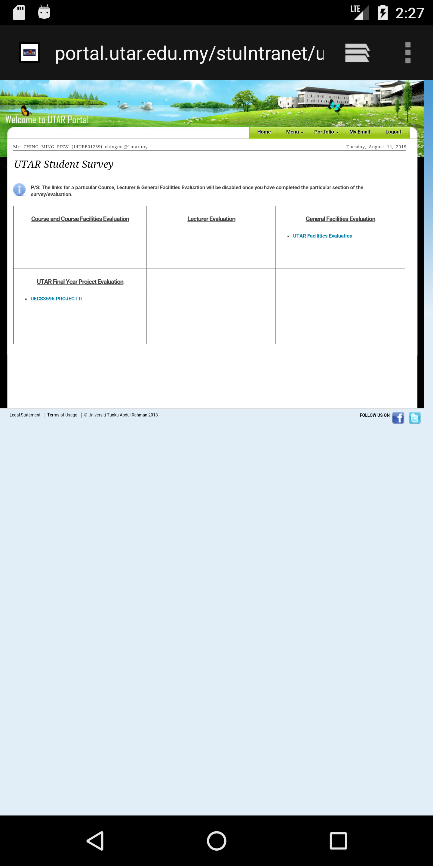


Figure 6.13: Interface of Feedback – UTAR Student Survey

### Expenditure Planning

The Expenditure Planning is designed to record every expense of users. It allows user to set initial amount of money they have, and by adding expense will automatically deduct these expenses from the initial amount of money and show remaining money user left. Expenses are separated into 9 categories, which are Education, Entertainment, Shopping, Auto and Transport, Personal Care, Health and Fitness, Food and Dining, Fees and Charges and Others. It also provides a pie chart for user to view their expenses summary.



Figure 6.14: Interface of Expenditure Planning

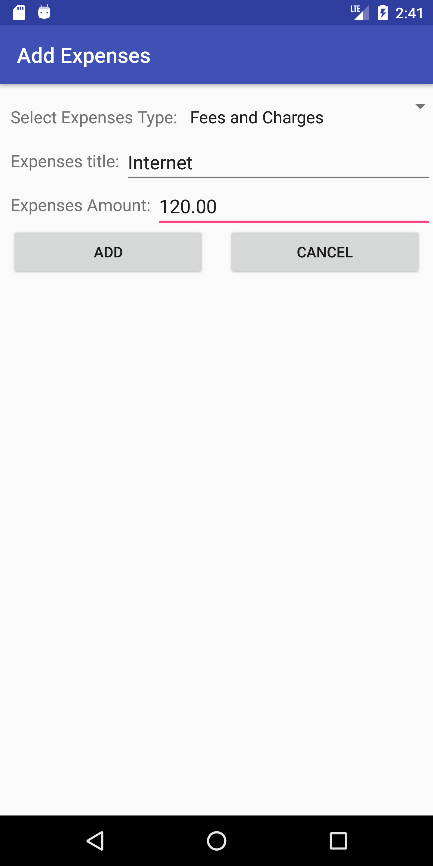


Figure 6.15: Interface of Expenditure Planning – Add Expense

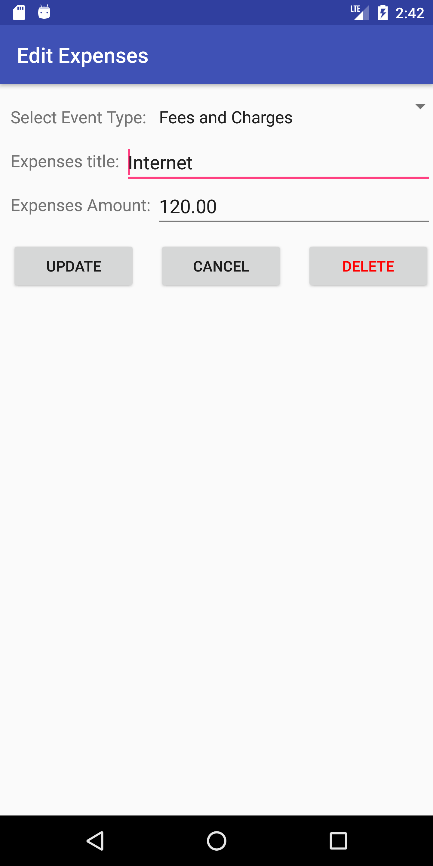


Figure 6.16: Interface of Expenditure Planning – Edit Expense



Figure 6.17: Interface of Expenditure Planning – Add Initial Amount

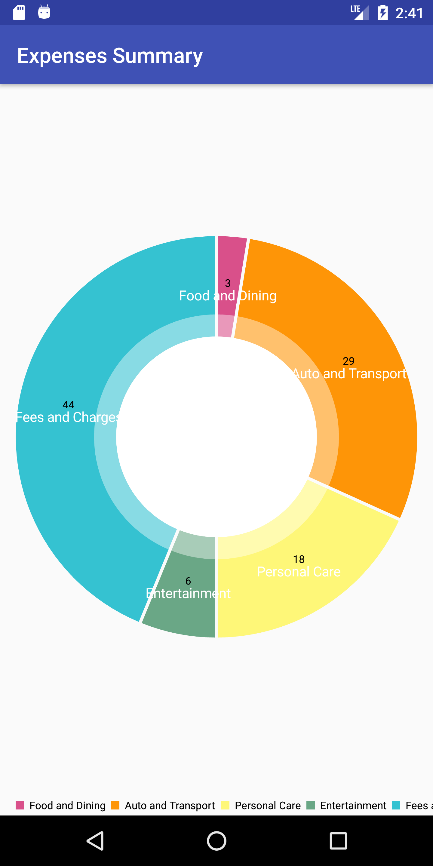


Figure 6.18: Interface of Expenditure Planning – Pie Chart of Expenses Summary

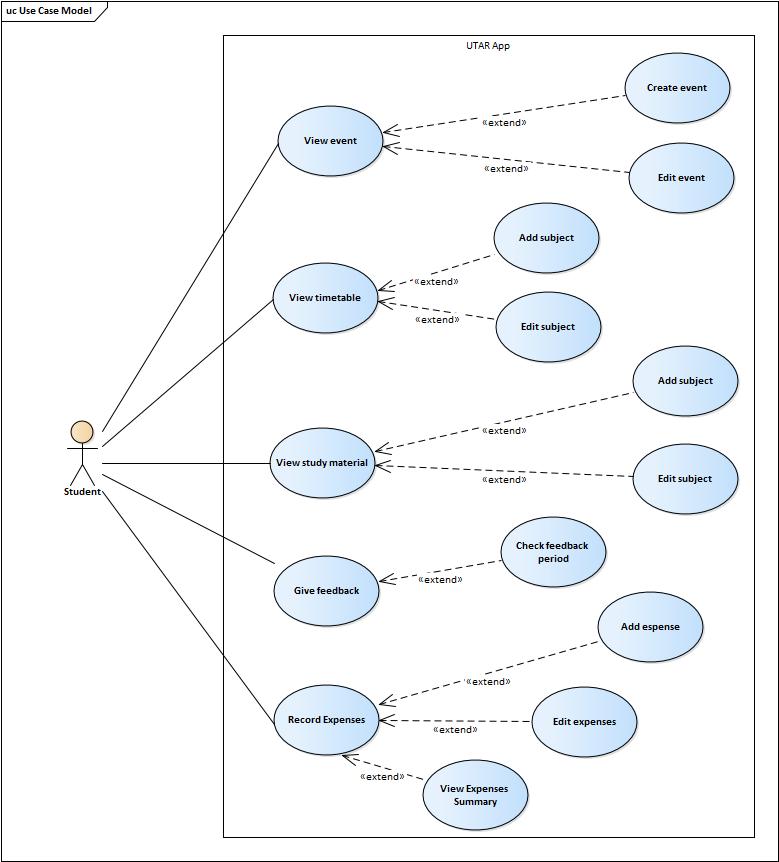
### Summary

This Final Year Project is designed based on requirement from Mr.Chan Kok Leong, several similar market applications, and also questionnaire. The Questionnaire is created using Google Form and please refer to APPENDIX A for complete result of questionnaire. In short, in terms of storing timetable, most students prefer screenshot using their device and store in their device storage as an image, only 7.7% of students use application to store their timetable. In contrast to the timetable storing method which not using any application, when it comes to storing event as schedule reminder, more than half of the students use an application to do the task. This is because almost all the Android smartphones nowadays have built in calendar application which allow user to schedule their event. Most of the students who use application for event  scheduling and reminder purpose had resulted in never to seldom forget or miss their events. Questionnaire targeted expenditure planning shown that, all students who did expenditure planning never had overspending problem. Only 11.5% of the respondent never overspent their pocket money, and 73.1% of the respondent never did expenditure planning. There are 57.7% of respondent claims that is not easy to find back specific study material, because they store those files or images all in one folder and it takes time to find a specific file or image.

# 

## SYSTEM DESIGN

### Use Case Diagram



### Use Case Description

#### View Event

|  |  |  |
| --- | --- | --- |
| Use Case Name: View Event | ID:001 | Importance Level: High |
| Primary Actor: Student | Use Case Type: Concrete | |
| Stakeholders and Interests:  Student – Student will be using the application to view the upcoming event. | | |
| Brief Description:  Student can view the upcoming event they created in the application. | | |
| Trigger:  When user click Dashboard button from the bottom navigation bar. | | |
| Relationships:          Association: Student          Include: n/a          Extend: Create Event, Edit Event          Generalization: n/a | | |
| Normal Flow of Events:   1. Student click Dashboard button from the bottom navigation bar. 2. System retrieve data from database and show on the list. | | |
| SubFlows: - | | |
| Alternate/Exceptional Flows: | | |

#### Create Event

|  |  |  |
| --- | --- | --- |
| Use Case Name: Create Event | ID:002 | Importance Level: High |
| Primary Actor: Student | Use Case Type: Concrete | |
| Stakeholders and Interests:  Student – Student will be using the application to create and store events. | | |
| Brief Description: Student can create and store event into the application. | | |
| Trigger:  When user click Add button from the Dashboard interface. | | |
| Relationships:          Association: Student          Include: n/a          Extend: n/a          Generalization: n/a | | |
| Normal Flow of Events:   1. User select event’s type from the drop down menu. 2. User enter event’s title. 3. User select event’s date. 4. User select event’s time. 5. User click Add button. 6. System store event’s details into database. | | |
| SubFlows: - | | |
| Alternate/Exceptional Flows: | | |

#### Edit Event

|  |  |  |
| --- | --- | --- |
| Use Case Name: Edit Event | ID:003 | Importance Level: High |
| Primary Actor: Student | Use Case Type: Concrete | |
| Stakeholders and Interests:  Student – Student will be using the application to update or delete events. | | |
| Brief Description: Student can edit and update an event or delete an event. | | |
| Trigger:  When user click an event from the list. | | |
| Relationships:          Association: Student          Include: n/a          Extend: n/a          Generalization: n/a | | |
| Normal Flow of Events:   1. User select event’s type from the drop down menu. 2. User enter event’s title. 3. User select event’s date. 4. User select event’s time. 5. User click Update button. 6. System rewrite event’s details into database. | | |
| SubFlows: - | | |
| Alternate/Exceptional Flows: Delete Event   1. User click Delete button. 2. System remove event’s data from database. | | |

#### View Timetable

|  |  |  |
| --- | --- | --- |
| Use Case Name: View Timetable | ID:004 | Importance Level: High |
| Primary Actor: Student | Use Case Type: Concrete | |
| Stakeholders and Interests:  Student – Student will be using the application to view timetable. | | |
| Brief Description:  Student can view time of specific subject class which they added previously to the timetable. | | |
| Trigger:  When user click Timetable button from the bottom navigation bar. | | |
| Relationships:          Association: Student          Include: n/a          Extend: Add Timetable, Edit Timetable          Generalization: n/a | | |
| Normal Flow of Events:   1. Student click Timetable button from the bottom navigation bar. 2. System retrieve data from database and show on the timetable. | | |
| SubFlows: - | | |
| Alternate/Exceptional Flows: - | | |

#### Add Event

|  |  |  |
| --- | --- | --- |
| Use Case Name: Add Event | ID:005 | Importance Level: High |
| Primary Actor: Student | Use Case Type: Concrete | |
| Stakeholders and Interests:  Student – Student will be using the application to add and store subject’s class’s times. | | |
| Brief Description: Student can add and store subject’s class’s times into application database and show on timetable. | | |
| Trigger:  When user click Add button from the Timetable interface. | | |
| Relationships:          Association: Student          Include: n/a          Extend: n/a          Generalization: n/a | | |
| Normal Flow of Events:   1. User enter subject’s title. 2. User tick subject’s class’s time with check box. 3. User click Add button. 4. System store subject’s details into database. | | |
| SubFlows: - | | |
| Alternate/Exceptional Flows: - | | |

#### Edit Timetable

|  |  |  |
| --- | --- | --- |
| Use Case Name: Edit Timetable | ID:006 | Importance Level: High |
| Primary Actor: Student | Use Case Type: Concrete | |
| Stakeholders and Interests:  Student – Student will be using the application to update or delete subject’s class’s time. | | |
| Brief Description: Student can edit and update subject’s class’s time or delete a subject. | | |
| Trigger:  When user click an subject’s block from the timtable. | | |
| Relationships:          Association: Student          Include: n/a          Extend: n/a          Generalization: n/a | | |
| Normal Flow of Events:   1. User enter subject’s title. 2. User tick subject’s class’s time with check box. 3. User click Update button. 4. System rewrite subject’s details into database. | | |
| SubFlows: - | | |
| Alternate/Exceptional Flows: Delete Subject’s Class’s Time   1. User click Delete button. 2. System remove subject’s data from database. | | |

#### View Study Material

|  |  |  |
| --- | --- | --- |
| Use Case Name: View Study Material | ID:007 | Importance Level: High |
| Primary Actor: Student | Use Case Type: Concrete | |
| Stakeholders and Interests:  Student – Student will be using the application to view study material. | | |
| Brief Description: Student can view study material such as image which store inside specific subject folder. | | |
| Trigger:  When user click Subject button from the bottom navigation bar. | | |
| Relationships: Subject          Association: Student          Include: n/a          Extend: Add Subject, Edit Subject          Generalization: n/a | | |
| Normal Flow of Events:   1. Student click Subject button from the bottom navigation bar. 2. System retrieve data from database and show on the list. | | |
| SubFlows: - | | |
| Alternate/Exceptional Flows: - | | |

#### Add Subject

|  |  |  |
| --- | --- | --- |
| Use Case Name: Add Subject | ID:008 | Importance Level: High |
| Primary Actor: Student | Use Case Type: Concrete | |
| Stakeholders and Interests:  Student – Student will be using the application to add and store as a folder. | | |
| Brief Description: Student can add and store subject into application database, show on list and work as a subject folder. | | |
| Trigger:  When user click Add button from the Subject interface. | | |
| Relationships:          Association: Student          Include: n/a          Extend: n/a          Generalization: n/a | | |
| Normal Flow of Events:   1. User enter subject’s title. 2. User click Add button. 3. System store subject’s title into database. | | |
| SubFlows: - | | |
| Alternate/Exceptional Flows: - | | |

#### Edit Subject

|  |  |  |
| --- | --- | --- |
| Use Case Name: Edit Subject | ID:009 | Importance Level: High |
| Primary Actor: Student | Use Case Type: Concrete | |
| Stakeholders and Interests:  Student – Student will be using the application to update or delete. | | |
| Brief Description: Student can edit and update a subject or delete a subject. | | |
| Trigger:  When user click the subject from the list. | | |
| Relationships:          Association: Student          Include: n/a          Extend: n/a          Generalization: n/a | | |
| Normal Flow of Events:   1. User enter subject’s title. 2. User click Update button. 3. System rewrite subject’s title into database. | | |
| SubFlows: - | | |
| Alternate/Exceptional Flows: Delete Subject   1. User click Delete button. 2. System remove subject from database. | | |

#### Give Feedback

|  |  |  |
| --- | --- | --- |
| Use Case Name: Give Feedback | ID:010 | Importance Level: High |
| Primary Actor: Student | Use Case Type: Concrete | |
| Stakeholders and Interests:  Student – Student will be using the application to give feedback to UTAR. | | |
| Brief Description:  Student can give feedback to UTAR’s facilities, services, subjects, lecturers and tutors. | | |
| Trigger:  When user click the “CLICK THIS BUTTON TO DIRECT TO STUDENT SURVEY” button from the Feedback interface. | | |
| Relationships:          Association: Student          Include: n/a          Extend: Check Feedback Period          Generalization: n/a | | |
| Normal Flow of Events:   1. User click “CLICK THIS BUTTON TO DIRECT TO STUDENT SURVEY” button. 2. System direct to UTAR Portal. 3. User login to UTAR Portal. 4. User click Highlight. 5. User click Utar Student Survey. | | |
| SubFlows: - | | |
| Alternate/Exceptional Flows: - | | |

#### Check Feedback Period

|  |  |  |
| --- | --- | --- |
| Use Case Name: Check Feedback Period | ID:011 | Importance Level: High |
| Primary Actor: Student | Use Case Type: Concrete | |
| Stakeholders and Interests:  Student – Student will be using the application to check feedback period. | | |
| Brief Description: Student can check valid period of UTAR Student Survey. | | |
| Trigger:  When user click the “CLICK THIS BUTTON TO CHECK VALID STUDENT SURVEY PERIOD” button from the Feedback interface. | | |
| Relationships:          Association: Student          Include: n/a          Extend: n/a          Generalization: n/a | | |
| Normal Flow of Events:   1. User click “CLICK THIS BUTTON TO CHECK VALID STUDENT SURVEY PERIOD” button. 2. System direct to check student survey period page. | | |
| SubFlows: - | | |
| Alternate/Exceptional Flows: - | | |

#### Record Expeses

|  |  |  |
| --- | --- | --- |
| Use Case Name: Record Expenses | ID:012 | Importance Level: High |
| Primary Actor: Student | Use Case Type: Concrete | |
| Stakeholders and Interests:  Student – Student will be using the application to view recorded expenses. | | |
| Brief Description: Student can view the expenses recorded previously. | | |
| Trigger:  When user click Expenditure Planning button from the bottom navigation bar. | | |
| Relationships:          Association: Student          Include: n/a          Extend: Add expenses, Edit expenses, View Expenses Summary          Generalization: n/a | | |
| Normal Flow of Events:   1. Student click Expenditure Planning button from the bottom navigation bar. 2. System retrieve data from database and show on the list. | | |
| SubFlows: - | | |
| Alternate/Exceptional Flows: - | | |

#### Add Expeses

|  |  |  |
| --- | --- | --- |
| Use Case Name: Add Expenses | ID:013 | Importance Level: High |
| Primary Actor: Student | Use Case Type: Concrete | |
| Stakeholders and Interests:  Student – Student will be using the application to add and record expenses. | | |
| Brief Description:  Students can record their day to day expenses and store into application. | | |
| Trigger:  When user click Add button from the Expenditure Planning interface. | | |
| Relationships:          Association: Student          Include: n/a          Extend: n/a          Generalization: n/a | | |
| Normal Flow of Events:   1. Student select type of expense from the drop down menu. 2. Student enter expense’s title. 3. Student enter amount of the expense. 4. Student click Add button. 5. System store expense’s detail into database. | | |
| SubFlows: - | | |
| Alternate/Exceptional Flows: - | | |

#### Edit Expeses

|  |  |  |
| --- | --- | --- |
| Use Case Name: Edit Expenses | ID:014 | Importance Level: High |
| Primary Actor: Student | Use Case Type: Concrete | |
| Stakeholders and Interests:  Student – Student will be using the application to update and edit or delete expenses. | | |
| Brief Description:  Students can edit and update expenses into database or delete expenses from database. | | |
| Trigger:  When user click Add button from the Expenditure Planning interface. | | |
| Relationships:          Association: Student          Include: n/a          Extend: n/a          Generalization: n/a | | |
| Normal Flow of Events:   1. Student select type of expense from the drop down menu. 2. Student enter expense’s title. 3. Student enter amount of the expense. 4. Student click Update button. 5. System rewrite expense’s detail into database. | | |
| SubFlows: - | | |
| Alternate/Exceptional Flows: Delete Expense   1. Student click Delete button. 2. System remove expense from the database. | | |

#### Edit Expeses

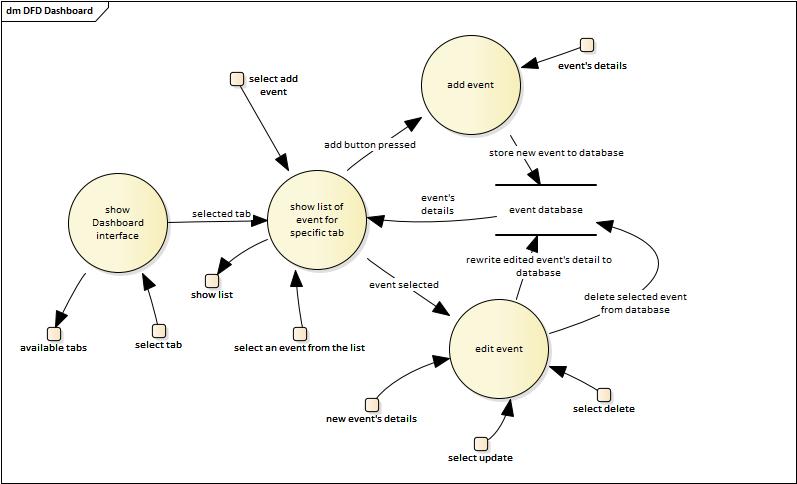
|  |  |  |
| --- | --- | --- |
| Use Case Name: Edit Expenses | ID:015 | Importance Level: High |
| Primary Actor: Student | Use Case Type: Concrete | |
| Stakeholders and Interests:  Student – Student will be using the application to update and edit or delete expenses. | | |
| Brief Description:  Students can edit and update expenses into database or delete expenses from database. | | |
| Trigger:  When user click Add button from the Expenditure Planning interface. | | |
| Relationships:          Association: Student          Include: n/a          Extend: n/a          Generalization: n/a | | |
| Normal Flow of Events:   1. Student select type of expense from the drop down menu. 2. Student enter expense’s title. 3. Student enter amount of the expense. 4. Student click Update button. 5. System rewrite expense’s detail into database. | | |
| SubFlows: - | | |
| Alternate/Exceptional Flows: Delete Expense   1. Student click Delete button. 2. System remove expense from the database. | | |

#### View Expeses Summary

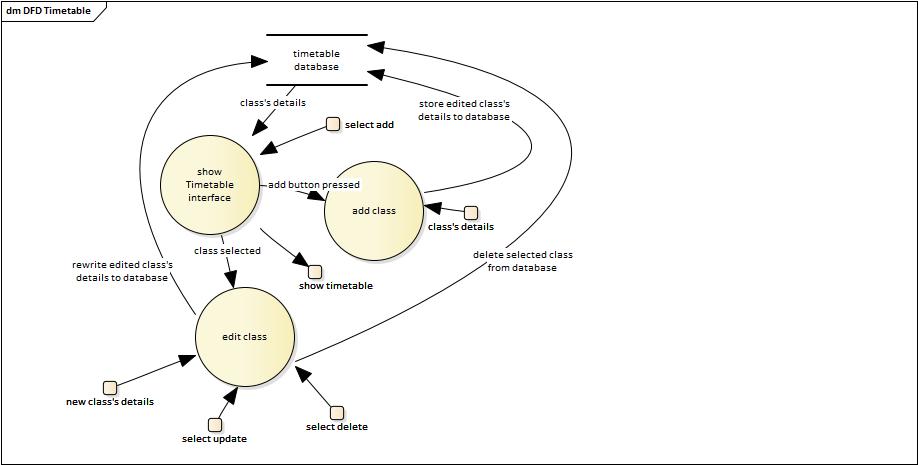
|  |  |  |
| --- | --- | --- |
| Use Case Name:  View Expenses Summary | ID:016 | Importance Level: High |
| Primary Actor: Student | Use Case Type: Concrete | |
| Stakeholders and Interests:  Student – Student will be using the application to view summary of expenses. | | |
| Brief Description:  Student can view the summary of expenses in a Pie Chart which calculate percantage of total expenses in each category. | | |
| Trigger:  When user click Remaining Amount number from the Expenditure Planning interface. | | |
| Relationships:          Association: Student          Include: n/a          Extend: n/a          Generalization: n/a | | |
| Normal Flow of Events:   1. Student click Remaining Amount number. 2. System load new interface and generate a pie chart to show the summary of expenses. | | |
| SubFlows: - | | |
| Alternate/Exceptional Flows: - | | |

### Data Flow Diagram

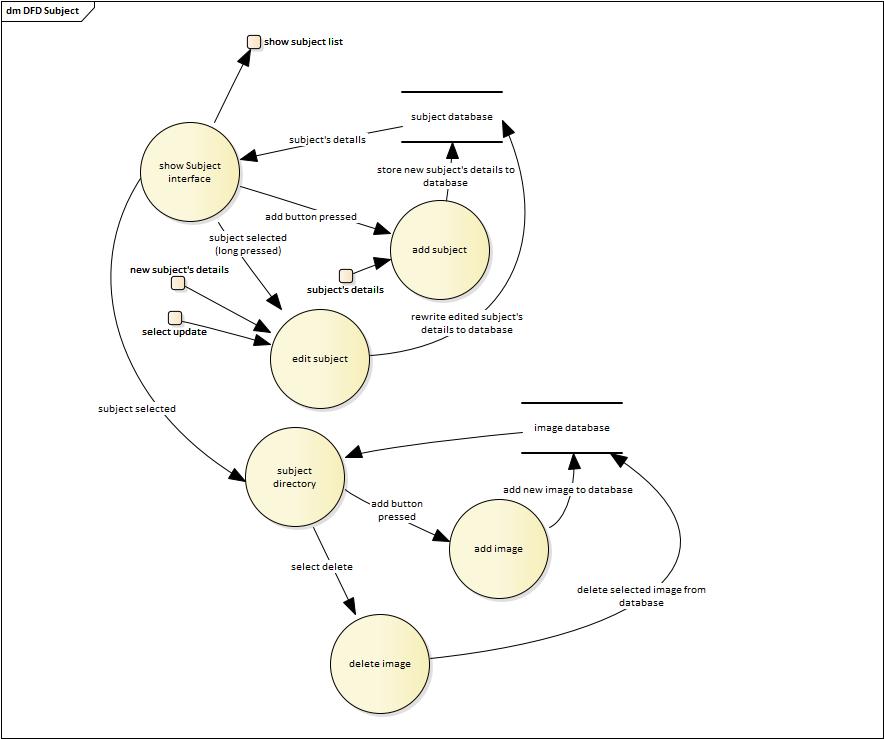
#### Dashboard Data Flow Diagram



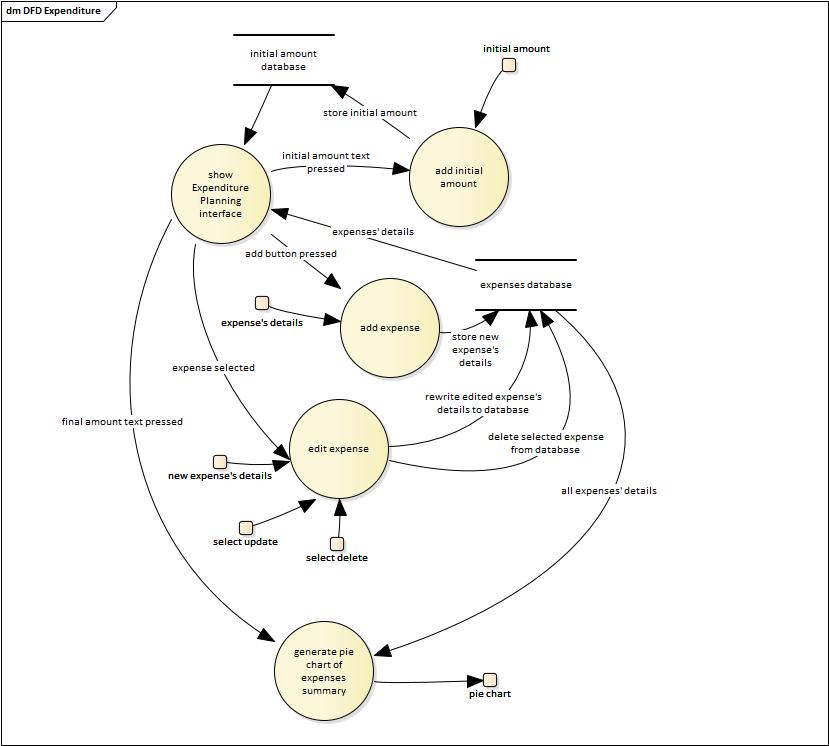
#### Timetable Data Flow Diagram



#### Subject Data Flow Diagram



#### Expenditure Planning Data Flow Diagram



## SYSTEM TESTING

### Introduction

The first paragraph in a subsection should align with left margin. General alignment for texts in paragraph should be “justified”.

Subsequence paragraphs should be indented 1.27 cm (0.5 inch) from the left margin. General alignment for texts in paragraph should be “justified”.

## CONCLUSIONS AND RECOMMENDATIONS

### Conclusions

The chapter should start by stating the major aim of the project/study, summarize the highlights of the previous chapters, and mention the achievements of the project in the conclusions section. The conclusion is to be written in present perfect tense. For example, “This study has presented a method to simulate fault cases in …”

Just as conclusions are based on the findings of the study, so are recommendations based on the conclusions. The recommendations flow logically from the discussion of the findings and conclusions drawn from them. This chapter is usually concluded in the Recommendations section by stating the constraints of the study and recommendations for other possible investigations as a follow-up to eliminate those constraints or to improve on the efficiency of the developed solution.

### Recommendations for future work

The first paragraph in a subsection should align with left margin. General alignment for texts in paragraph should be “justified”.

Subsequence paragraphs should be indented 1.27 cm (0.5 inch) from the left margin. General alignment for texts in paragraph should be “justified”.

A new paragraph should not begin on the last line of a page. A subsection title should not begin on the last line of a page. A new chapter must start on a new page. Spacing between the last line of text and equation is 1.5 lines.

## REFERENCES

Aghion, Z. (2014). *What are some examples of hybrid native and HTML5 iPhone or Android applications?* - Quora. [online] Quora.com. Available at: https://www.quora.com/What-are-some-examples-of-hybrid-native-and-HTML5-iPhone-or-Android-applications [Accessed 18 Mar. 2018].

Budiu, R. (2013). *Mobile: Native Apps, Web Apps, and Hybrid Apps*. [online] Nielsen Norman Group. Available at: https://www.nngroup.com/articles/mobile-native-apps/ [Accessed 14 Mar. 2018].

Chrzanowska, N. (2017). *React Native - Pros and Cons Of Facebook’s Framework.* [online] Netguru.co. Available at: https://www.netguru.co/blog/react-native-pros-and-cons [Accessed 6 Mar. 2018].APPENDICES

Fox, L. (2015). *The Advantages and Disadvantages of Web Apps.* [online] Available at: https://www.objectiveit.com/blog/the-advantages-and-disadvantages-of-web-apps [Accessed 9 Mar. 2018].

Nerur, S., Mahapatra, R. and Mangalaraj, G. (2005). *Challenges of migrating to agile methodologies.* [online] Available at: https://dl.acm.org/citation.cfm?id=1060712 [Accessed 16 Mar. 2018].

Noh, H.(2014). *Literature Review: Starting Mobile Application Developmenterur for E-Sport Portal.* [online] Available at: https://people.cs.uct.ac.za/~cpatrick/Honours/files/LiteratureReviewNHXHAY001.pdf [Accessed 16 Mar. 2018].

Pathan, Z. (2017). *What are the advantages and disadvantages of developing apps on Xamarin cross-platform?* - Quora. [online] Quora.com. Available at: https://www.quora.com/What-are-the-advantages-and-disadvantages-of-developing-apps-on-Xamarin-cross-platform [Accessed 17 Mar. 2018].

REACT FRAMEWORK (CREATING A WEB APPLICATION WITH REACT NATIVE). (2018). *International Journal of Recent Trends in Engineering and Research*, 4(3), pp.642-646.

Reynolds, M. (2014). *Xamarin mobile application development for Android.* Birmingham, UK: Packt Pub., pp.100-112.

Rouse, M. (2018). *What is native app?* - Definition from WhatIs.com. [online] SearchSoftwareQuality. Available at: http://searchsoftwarequality.techtarget.com/definition/native-application-native-app [Accessed 1 Apr. 2018].

Rouse, M. (2015). *What is framework?* - Definition from WhatIs.com. [online] WhatIs.com. Available at: http://whatis.techtarget.com/definition/framework [Accessed 15 Mar. 2018].

OmkarSoft Blog. (2017). *Top Advantages and Disadvantages of Native App.* [online] Available at: https://www.omkarsoft.com/blog/top-advantages-disadvantages-native-app/ [Accessed 15 Mar. 2018].

VanderLeest, S. and Buter, A. (2009). *Escape the waterfall: Agile for aerospace - IEEE Conference Publication.* [online] Ieeexplore.ieee.org. Available at: http://ieeexplore.ieee.org/abstract/document/5347438/authors [Accessed 25 Mar. 2018].

Viswanathan, P. (2017). *The Pros and Cons of Native Apps and Mobile Web Apps*. [online] Lifewire. Available at: https://www.lifewire.com/pros-and-cons-of-native-apps-and-mobile-web-apps-2373173 [Accessed 14 Mar. 2018].

Smartsheet. (2018). *What's the Difference? Agile vs Scrum vs Waterfall vs Kanban.* [online] Available at: https://www.smartsheet.com/agile-vs-scrum-vs-waterfall-vs-kanban [Accessed 4 Apr. 2018].Rouse, M. (2015). What is framework? - Definition from WhatIs.com. [online] WhatIs.com. Available at: http://whatis.techtarget.com/definition/framework [Accessed 15 Mar. 2018].

## APPENDICES

APPENDIX A: Questionnaire Result

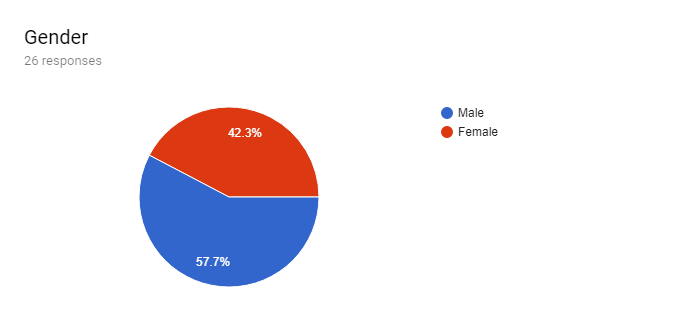


Figure 7.1: Questionnaire

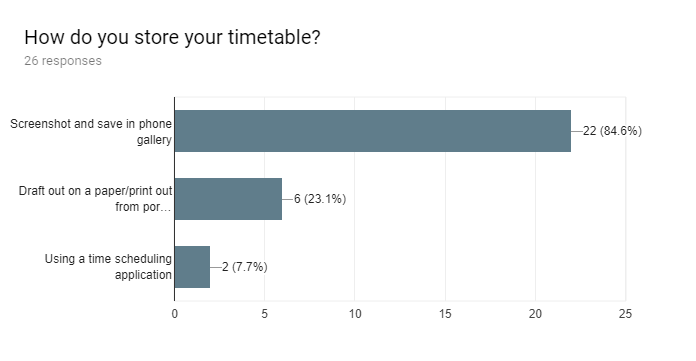


Figure 7.2: Questionnaire

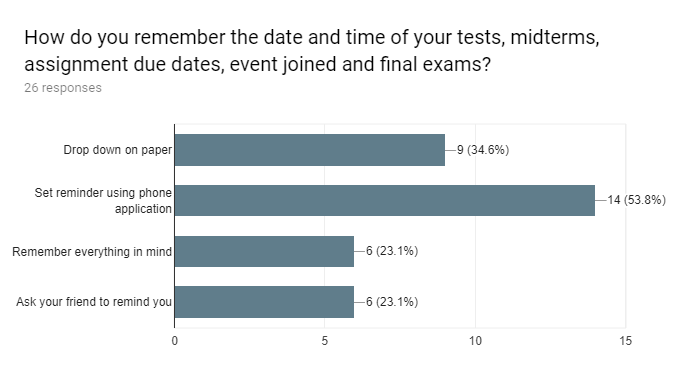


Figure 7.3: Questionnaire

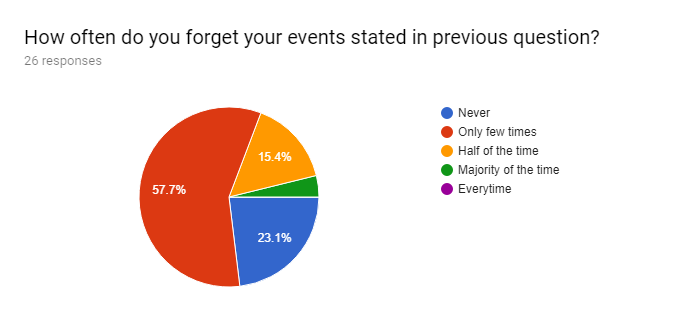


Figure 7.4: Questionnaire

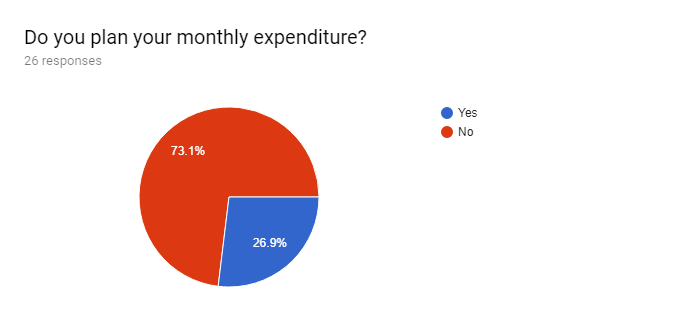


Figure 7.5: Questionnaire

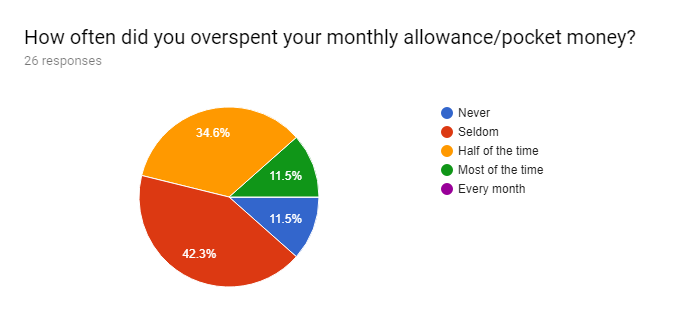


Figure 7.6: Questionnaire

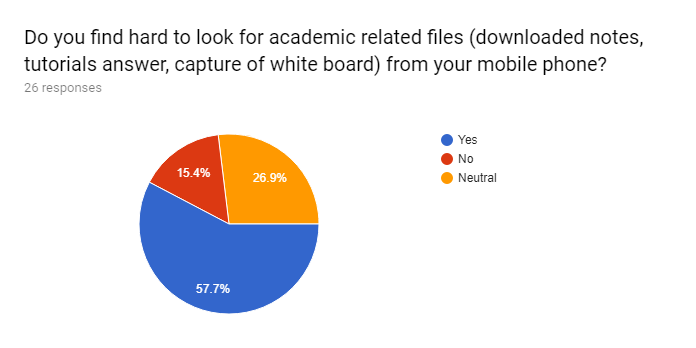


Figure 7.7: Questionnaire

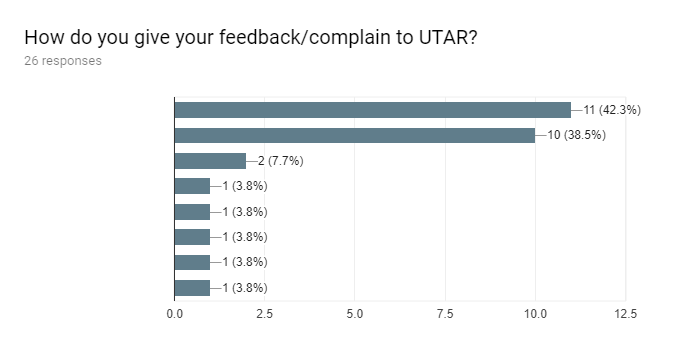


Figure 7.8: Questionnaire

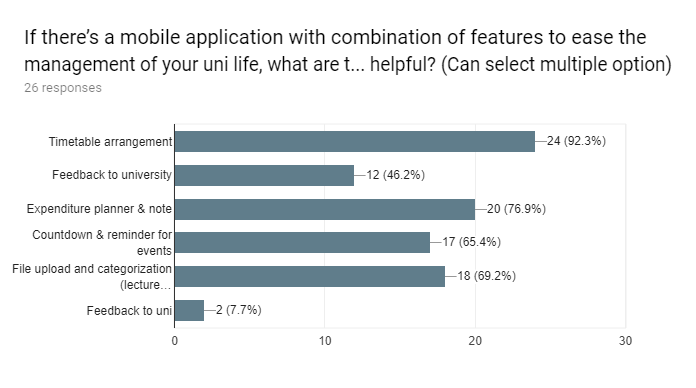


Figure 7.9: Questionnaire

APPENDIX B: User Acceptance Testing Result

User Acceptance Testing Form.

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Test Case | Steps | Expected Result |
| C1.0 | Add event. | 1. Press Add icon. 2. Fill in all details. 3. Press Add button. | 1. New added event wil be added into the list. |
| C1.1 | Edit an event. | 1. Select an event from the list. 2. Edit event’s details. 3. Press Update button. | 1. New event’s details will rewrite old event’s details. 2. Updated event will be added into the list. |
| C1.2 | Delete an event. | 1. Select an event from the list. 2. Press Delete button. | 1. Selected event will be remove from the list. |
| C1.3 | Switch between tabs. | 1. Press any of the tab to show specific type of events. | 1. Each tab will only show specific type of events. |
| C2.1 | Switch to Timetable interface. | 1. Press Timetable from bottom navigation bar. | 1. Interface will show timetable. |
| C2.2 | Add class | 1. Press Add icon. 2. Fill in all details. 3. Select class’s time. 4. Press Add button. | 1. New added class will be added into the table. |
| C2.3 | Edit class | 1. Select a class block from the timetable. 2. Edit class’s details. 3. Edit class’s time. 4. Press Update button. | 1. New class’s details will rewrite old class’s details. 2. Updated class will be added into the table. |
| C2.4 | Delete class. | 1. Select a class block from the timetable. 2. Press Delete button. | 1. Selected class will be remove from the table. |
| C3.0 | Switch to Subject interface. | 1. Press Subject from bottom navigation bar. | 1. Interface will show subject list. |
| C3.1 | Add subject | 1. Press Add icon. 2. Fill in all details. 3. Press Add button. | 1. New added subject will be added into the list. |
| C3.2 | Edit subject | 1. Select(long press) a subject from the list. 2. Edit subject’s details. 3. Press Update button. | 1. New subject’s details will rewrite old subject’s details. 2. Updated subject will be added into the list. |
| C3.3 | Delete subject | 1. Select(long press) a subject from the list. 2. Press Delete button. | 1. Selected event will be remove from the list. |
| C3.4 | Enter subject directory | 1. Select a subject from the list. | 1. Interface will show subject folder directory. |
| C3.5 | Add image from device’s camera | 1. Press Camera icon. 2. Select Camera icon. 3. Take a picture. | 1. Image captured by the camera will be added into specific subject folder directory. |
| C3.6 | Add image from device’s gallery | 1. Press Camera icon. 2. Select Gallery icon. 3. Choose a picture. | 1. Image selected will be added into specific subject folder directory. |
| C3.7 | Delete image | 1. Select image from the list. 2. Press delete button. | 1. Selected image will be remove from the list. |
| C4.1 | Switch to Feedback interface. | 1. Press Feedback from bottom navigation bar. | 1. Interface will show feedback interface.. |
| C4.2 | Check Student Survey valid period | 1. Press “CKICK THIS BUTTON TO CHECK VALID STUDENT SURVEY PERIOD” button | 1. Application will direct user to check Student Survey period page. |
| C4.3 | Do Student Survey | 1. Press “CKICK THIS BUTTON TO DIRECT TO STUDENT SURVEY” button 2. Login UTAR Portal 3. Click Highlight. 4. Click Student Survey. 5. Complete Student Survey. | 1. Application will direct user to UTAR portal login page. |
| C4.3 | Do Student Survey | 1. Press “CKICK THIS BUTTON TO DIRECT TO STUDENT SURVEY” button 2. Login UTAR Portal 3. Click Highlight. 4. Click Student Survey. 5. Complete Student Survey. | 1. Application will direct user to UTAR portal login page. |
| C5.1 | Switch to Expenditure Planning interface. | 1. Press Expenditure Planning from bottom navigation bar. | 1. Interface will show initial amount, list od expenses and remaining amount. |
| C5.2 | Add initial amount. | 1. Press initial amount number text. 2. Enter initial amount. 3. Press Add button. | 1. New initial amount number will be show at the initial amount text field. |
| C5.3 | Add expense. | 1. Press Add icon. 2. Enter expense’s details. 3. Press Add button. | 1. New added expense will be added into the list. |
| C5.4 | Edit expense | 1. Select a expense from the list. 2. Edit expense’s details. 3. Press Update button. | 1. New expense’s details will rewrite old expense’s details. 2. Updated expense will be added into the list. |
| C5.5 | Delete expense | 1. Select expense from the list. 2. Press delete button. | 1. Selected expense will be remove from the list. |
| C5.6 | Check expenses summary | 1. Press Remainning amount number text. | 1. Interfae will show a pie chart about summary of expenses. |

Evaluation Form

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Description |  |  |  |  |  |
| I found the system’s interface (buttons, dialog, icons) and easy way to performing system functions. | 1 | 2 | 3 | 4 | 5 |
| I felts the system responded in a consistent and predictable way. | 1 | 2 | 3 | 4 | 5 |
| Information was easy to find. | 1 | 2 | 3 | 4 | 5 |
| Feedback given by the system was proper and sufficient. | 1 | 2 | 3 | 4 | 5 |
| The queue information provided is enough for me to decide whether to leave the service center while waiting. | 1 | 2 | 3 | 4 | 5 |
| I would like to use this system in real life. | 1 | 2 | 3 | 4 | 5 |

Instruction: Circle or tick on the chosen answer.

1 = Strongly Disagree

2 = Disagree

3 = Neutral

4 = Agree

5 = Strongly Agree