“GradeRunner”

App for Managing Coursework Deadlines

Lyle Christine BA (Hons), MRes, MPhil

School of Computing Science

Sir Alwyn Williams Building

University of Glasgow

G12 8RZ

A dissertation presented in part fulfilment of the requirements of the Degree of Master of Science in Software Development at the University of Glasgow

September 2018

Abstract

GradeRunner is a mobile phone application which helps students track their coursework requirements, grades, exam details and student representatives. This Android app is designed to contribute to a reduction of stress in student's busy schedules, and the app does so with an uncomplicated menu system. Today's students, whether in school, college or university, are bombarded with coursework deadlines, assignment feedback, and demanding exam schedules: **GradeRunner** is an app which helps the modern student manage their time efficiently and prioritise the obligations of their qualification.

The following dissertation discusses the motivation behind creating **GradeRunner**, the technical process of building the application, and the level of success achieved by the final product.

Contents

Chapter 1 Introduction

Academic coursework requirements are demanding – in both volume and complexity. Staying organised is one of the key ingredients in order for students to successfully prepare for, and complete, their coursework and exam requirements. When facing such demands, there is the potential risk for students to lose track of deadlines, miss classes or become disheartened should they receive a less-than-excellent grade.

Subsequently, an application which can assist with a student's routine of tracking coursework requirements, grades and attendance would be a welcomed asset during their academic period. The software which has been created for this Master's project is called **GradeRunner** and is designed to be a lightweight mobile phone application. The aim of **GradeRunner** is to allow students, whether in school, college or university, to input assignment details, grades and feedback, student representative details and exam information. This data is persistent and can be reviewed at any time in order to allow an app user to organise their time in an appropriate manner, while also presenting the options of viewing their grade achievements or contacting a representative should they have any problems or concerns.

This dissertation provides details on the motivation for building **GradeRunner**, the aims of the app, and how the software was built. The thesis is organised in the following manner:

Chapter 1 provides a brief introduction of the app and the structure of the report; **Chapter 2** is more detailed in providing a background to why the software would benefit students and how the requirements of the software were gathered. This chapter also provides an analysis of similar apps currently on the market in order to gain an understanding of the common features in these types of applications, before presenting the functional and non-functional requirements of the **GradeRunner** project; **Chapter 3** is focused on how the app was designed and implemented, and provides an example of **GradeRunner's** features using a series of screenshots; **Chapter 4** provides testing and evaluation information which is divided between a user evaluation involving research participants and a questionnaire, alongside a practical test of the functional and non-functional aspects of the app; **Chapter 5** concludes the dissertation by reflecting on the achievements, weaknesses and possible future work of the project.

A video demonstration of **GradeRunner** can be viewed here: https://youtu.be/NkyimsbVaj8

The entire **GradeRunner** Android Studio project can be downloaded from GitHub here: https://github.com/Lylio/GradeRunner

Chapter 2 Analysis & Requirements

2.1 Background

There is an increasing amount of evidence which suggests today's students are experiencing a variety of psychological tolls which are impacting on their degree performance, reducing the enjoyment of their coursework and hampering the formation of social relationships (Chaffin, 2018). Some of these pressures are highly visible and understandable, such as financial struggles (Quinn, 2017). Other issues however are less tangible and involve mental health and emotional problems – for example, paradoxical feelings of isolation resulting from digital social media, or cultural anxieties born from a perception of an increasingly angry, fractured society (Dodgson, 2018). The consequences of such pressures can be catastrophic: in 2018, University drop-out rates in England rose for the third year in a row (Weale, 2018), while a recent study has shown that student suicide has overtaken the general population for the first time (Rudgard, 2018). Furthermore, a cursory search on popular student community website **The Student Room** (www.thestudentroom.co.uk) reveals many posts with similar themes such as "*struggling with the workload at university*", "*can't cope with the workload*" and "*2 months in at uni and struggling with the workload*".

Demands at University come hard and fast, and coursework deadlines along with class attendance no less so. Subsequently, a practical mobile app which can help document essay requirements, record grades and catalogue class attendance has the potential to support students throughout their study. The idea is that this application will serve three main purposes:

1. Enable a student to note their coursework requirements, deadlines and exam information viewed as list formats: in essence it helps the student stay organised and allows the student to quickly access an overview of their impending coursework schedule which helps the student to prioritise their workload.
2. The app will allow the student to add grades and feedback once their coursework has been assessed. I feel it would be beneficial for a student's esteem to have access to an overview of how well they have performed during each semester. Being able to glance over a list of (hopefully) impressive percentage marks and positive comments would really lift a student's spirits, especially during times of doubt and stress.
3. Students will be able to keep a record of which lectures and tutorials they have either attended or missed. Such information could be valuable if a student has become tardy in their on-campus presence. Furthermore, such information could also be useful for international students who must legally submit attendance information due to visa requirements.

2.2 Requirements Gathering

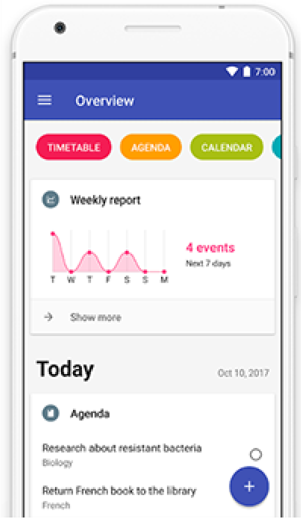
There were two main strategies to gather the requirements for this application. The first approach was a reflective session of role-play in which the needs of a potential student user of the application were imagined and sketched out. This process was informed by my own experiences as a student, as well as from anecdotal evidence which I have built up through many years of being exposed to various communities of university students. Casual conversation in the presence of other students often revolves around coursework deadlines and I have developed an awareness that many students are often frustrated in trying to keep track of the various requirements of their curricula. Such discussions have also revealed that many students feel dissatisfied with the main IT systems provided by higher education institutions. While such information systems administered by universities and colleges usually deliver an *adequate* level of service with respect to course details, timetable schedules and staff profiles, there is nonetheless dismay among both students and staff that these systems can be convoluted, sluggish and bloated with respect to access and navigation. It is precisely such frustrations that steered me towards adopting three important design principles: web and mobile technology should have a clear purpose; applications should also not try to be too many things at once; and user experience should be guided by a minimal and clear navigation aesthetic.

The second method of inquiry involved researching similar organisational products to my own idea and evaluating their strengths and weaknesses. The Google Play store (play.google.com) was the obvious source for this research as the selection of applications on Google Play are designed for the Android platform. By using keyword searches such as “study planner”, “timetable”, “to do” and “deadlines app”, it was possible to compile a list of products which were thematically comparable to my proposed **GradeRunner** application. By comparing these products, I was able to observe the common patterns of functionality in the apps while also taking note of user reviews and ratings on the Google Play store. Evaluating consumer complaints or praise is a useful aspect of product development, particularly when trying to determ the value of an application’s features and whether the product does (or does not) deliver a positive user experience.

Analysis

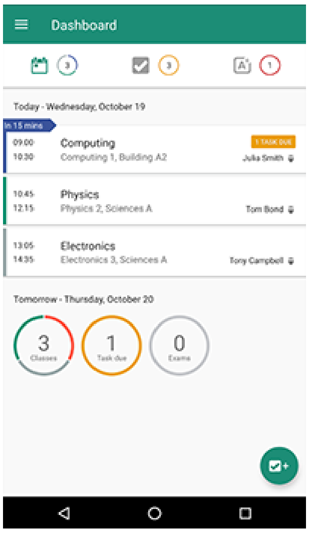
Four of the more popular student apps on Google Play are **School Planner**[1], **My Study Life**[2], **Timetable**[3] and **myHomework**[4].

School Planner (55,000 downloads)



School Planner is described as "a handy app for students of all ages that was designed to help you organize your career as a student and have everything under control". The main features of this app are comprehensive and include agenda, calendar, timetable, grades, subjects, attendance, teachers and recordings. Adding content is an easy process of selecting a section and clicking a button to insert an important reminder. For example, the agenda section has a large floating button which allows the creation of an alarm reminder, exam or homework deadline. **School Planner** also has a main overview screen which displays impending events based on content that the user has inputted. The presentation is clean and navigation straightforward – it is not likely a user of this application would be confused with respect to creating and locating content. Furthermore, the application can be used offline and users are not required to sign-up or login in order to use the app. Overall, **School Planner** performs strongly.

My Study Life (48,000 downloads)

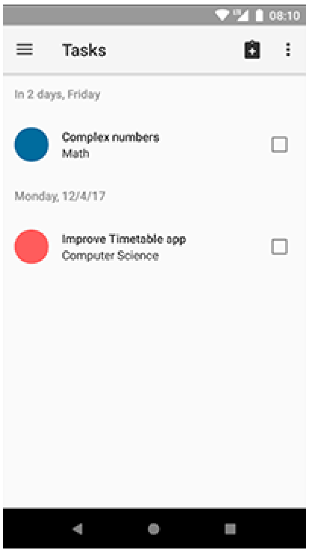


My Study Life allegedly "allows you to store your classes, homework and exams in the cloud making it available on any device, wherever you are". Subsequently, a sign-up and login process is mandatory for this product, and the application can only be launched when connected to the internet. Initial impressions were not strong as even when my devices were connected via wi-fi or mobile network, sometimes the app would complain that there was no internet connection and not allow the login process to proceed. When the app did work, the general user experience was lacklustre: while task and exam reminders could be created, the details did not appear in the calendar or schedule sections. The main dashboard presented these tasks as a list which was fairly clear and useful, but it

[1] https://play.google.com/store/apps/details?id=daldev.android.gradehelper
[2] https://play.google.com/store/apps/details?id=com.virblue.mystudylife
[3] https://play.google.com/store/apps/details?id=com.gabrielittner.timetable&hl=en_GB
[4] https://play.google.com/store/apps/details?id=com.myhomeowork

appears the calendar and schedule sections are redundant or broken. Overall, **My Study Life** is a weak application.

Timetable (40,000 downloads)



In 2 days, Friday

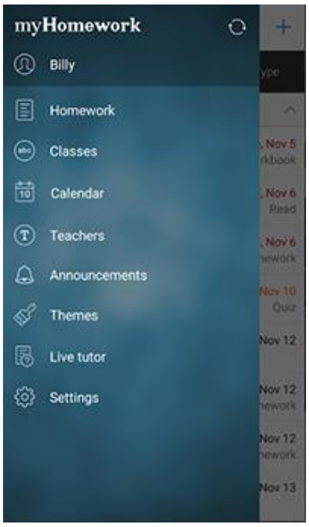
Complex numbers
Math ☐

Monday, 12/4/17

Improve Timetable app
Computer Science ☐

While **Timetable** is in principle a straightforward idea which focuses on creating timetables with task and exam dates, the execution is aesthetically weak. The overall presentation involves oversized textboxes in which to enter and view text data, and these boxes can be scrolled either vertically or horizontally without any indication of which scroll direction each field uses. Very little custom design has gone into the presentation in terms of fonts and graphics which results in **Timetable** feeling amateurish. Overall, this product was not impressive in design or functionality.

myHomework (28,000 downloads)



The **myHomework** app is described as “a modern design and simple interface making it easy for anyone to use” which is an accurate description of how streamlined this product is. The main sections include homework, classes, calendars, teachers and the ability to connect to the app’s companion website www.teachers.io for the purpose of live announcements. The live announcements is a novel and useful feature, but what the really striking aspect of this app is the ease of adding homework/tasks – by clicking the large ‘add’ button, a list menu appears with clear instructions of which data to enter: description, class, type, due date, priority, reminder, additional info. These headings appear on sperate lines making the process of creating tasks very clear; the procedure is obvious, and such a clear presentation is a strength of this product. Overall, **myHomework** gets many things right.

In summary, **School Planner** and **myHomework** are the strongest applications out of the four evaluations. Common features were observed across each product: the ability to create class/course information: the ability to create task/homework information; the ability to insert this information into a calendar; contact information for staff; location information for relevant venues. Such functionality is congruent with the motivation which has inspired the **GradeRunner** application.

2.3 Aims & Objectives

The aim of this application is to assist students with keeping track of their coursework requirements. From personal experience (and also witnessing the study patterns of other students), important information with regard to assessments, grades and exams can easily become 'scattered' across various locations and formats. For example, during my own studies, I have recorded critical information and deadlines using a plethora of sources such as a 'post-it notes' application on my Windows PC, dozens of text files entitled 'to do', 'exam info' and grades', several spreadsheets with class/lab attendance details, and all of this data had to be synchronised via various cloud platforms for backups and so I could access the files from my phone, tablet and campus. I ended up with the view that a more elegant solution would be a single-source lightweight application that could record such information. While it is to be expected that university coursework *content* is time-consuming and stressful, I feel there is a forgotten fact that the process of being organised is also *in itself* time-consuming and stressful. It is here I envisage my application being a positive appendage to the student experience.

2.3.1 Functional requirements – MoSCoW

Must Have:

1. The ability to create/edit/delete a list of courses
2. The ability to create/edit/delete an item from the list of courses
3. The ability to create/edit/delete coursework details to each course list item
4. The ability to create/edit/delete grade results for each course list item
5. The ability to see an overview of grades to date

Should Have:

1. The ability to create/edit/delete details for class attendance
2. The ability to create/edit/delete a list of exam details
3. A class representative section

Could Have:

1. User login authentication
2. A representation of average class attendance, e.g. an overall percentage or graph
3. The ability to add feedback comments in addition to coursework grades
4. Alarm/notifications of impending deadlines

Will Not Have:

1. A cloud-based database such as Google's Firebase or Amazon's AWS. Android's embedded SQLite database will be sufficient for this project.

2.3.2 Non-functional requirements

1. User data will be persistent (i.e. stored in a database)
2. The app will function offline (no internet connection required)
3. The database can be updated

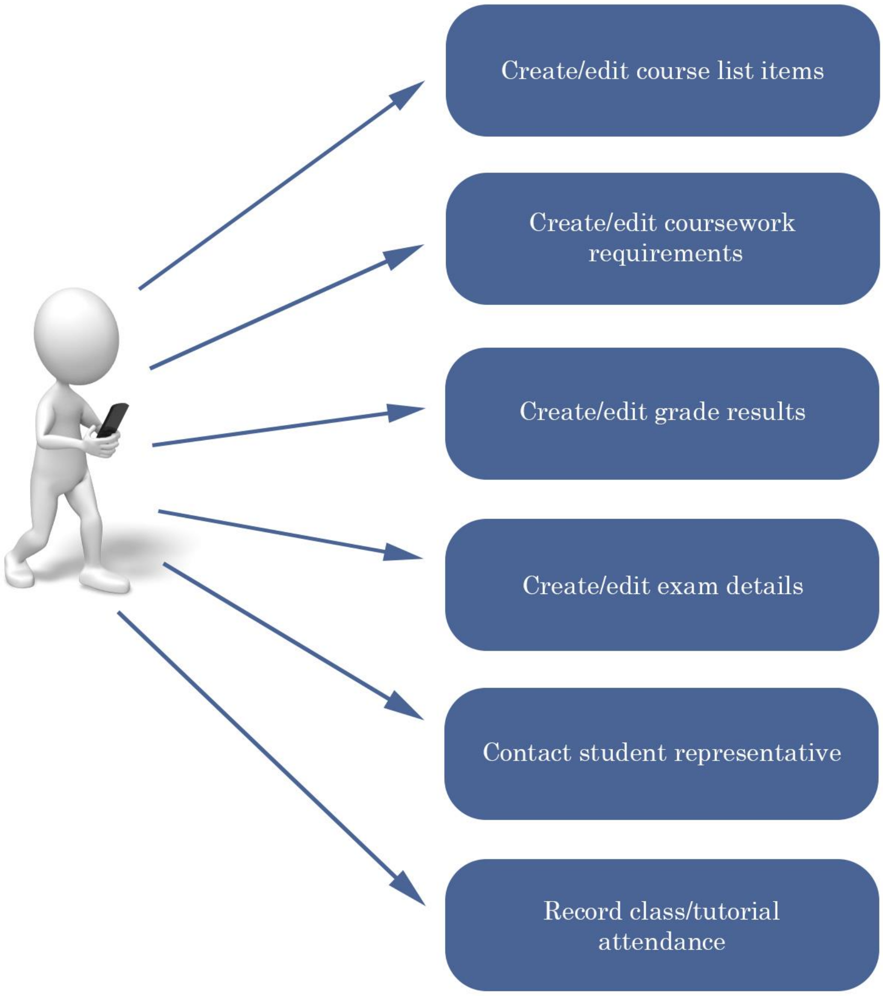

Create/edit course list items
Create/edit coursework requirements
Create/edit grade results
Create/edit exam details
Contact student representative
Record class/tutorial attendance

Figure 1: Use Case Diagram

Chapter 3 Design & Implementation

Having conducted the analysis and requirements phase of the project, which involved a reflective user role-play session and an evaluation of current market products, the core aspects of **GradeRunner's** backbone functional requirements were determined. With the functionality of the application largely confirmed, the next important step (before the focused phase of actually coding and building the product), was to consider design and implementation. Fundamentally, this phase involved: considering the system architecture (with a rationale of how major design decisions were made); details of how the product was actually implemented; and finally, a selection of screenshots as a visual presentation of how the app performed.

3.1 Architecture

High level architecture

The **GradeRunner** app is based on an architecture that exists without the complications of a product which is bloated with unnecessary appendages which overwhelm or slow the desired slimline design of the final application. These design choices have not been made in order to make the life of the developer easy; they are sensible choices when considering if an app will be lightweight (i.e. a small install size; low RAM memory usage) and frequency of operation (i.e. how often the app would be used; how often the app will be quickly accessed via a shortcut). The high-level architecture diagram is demonstrated in **Appendix A**. This diagram demonstrates the interaction between: user; phone app; and data. The **GradeRunner** app shall be installed and run as any typical Android application which can be downloaded from the Google Play consumer store. No additional software or hardware shall be necessary to run the application.

Features

The **GradeRunner** interface is designed to be self-explanatory and therefore require little in the way of instruction while the user navigates the software. Nonetheless, it is of benefit to provide an overview of the main navigational features of the product for the benefit of system evaluators of the product. **GradeRunner's** UI has text fields (for the user to enter data) and 4 main buttons (for the user to manipulate data). This design choice was chosen in order to create an app which should be immediately self-explanatory, and therefore does not require hefty documentation with which a user must wade through prior to using the software.

The diagram below provides a typical example of the menu interface of each section of the app.

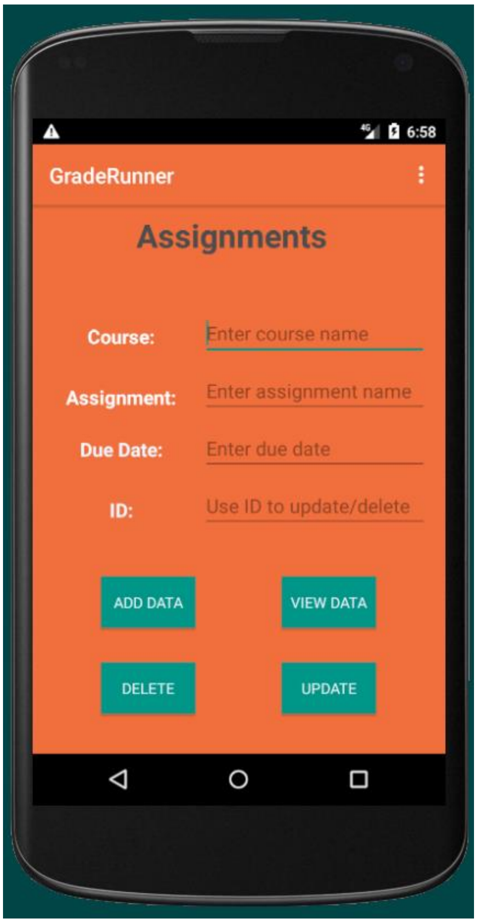


Figure 2: Typical menu interface

A student user of the **GradeRunner** app should find the interface to be self-explanatory and therefore easy to use. The benefit of this design choice is during potential times of stress, the purpose of the application is to assist rather than contribute to increased workload or stress levels. On first launching the app the user is presented with a launch screen with a self-explanatory menu which has buttons allowing the user to create, update, view and delete assignment items. This format is consistent with the other sections of the app which include: grades, attendance, representative and exams. The student will use this to input the activities for each course they have undertaken to study across their semesters. When the information for the exam diet becomes available, students will be able to input: dates; times; and location details of their duties during the exam period.

After entering the details of each course, the student will be able to click on 'view data' and be presented with a list of their courses, the assignment name/number and the due date for that particular assignment. Each course item can have its details updated or the entire coursework item can be completely deleted. This format is applicable to each section of the application. As the academic year progresses the student can record their grades for coursework assignments thus allowing students to see an overview of their achievements. This could be of benefit to students who find the emotional strain of academic life to be difficult as they will be able to see easily what they have achieved, hopefully offering them some reassurance.

The app will provide users with an efficient and effective way to communicate quickly with the class representative for each course. This will help the class representative to better understand the student experience and to be able to represent the students accurately. Within the app, students will select the 'settings' option which would take them to a page when they will be able to input the name and contact details for the representative for each of their courses. The student can then use this option after it has been set up to connect their representative directly by email.

Due to page space constraints, **Appendix B** presents an excerpt UML diagram as an example showing the Attendance Activity and Database Helper – the other sections of the app are consistent with this UML excerpt.

3.2 Implementation

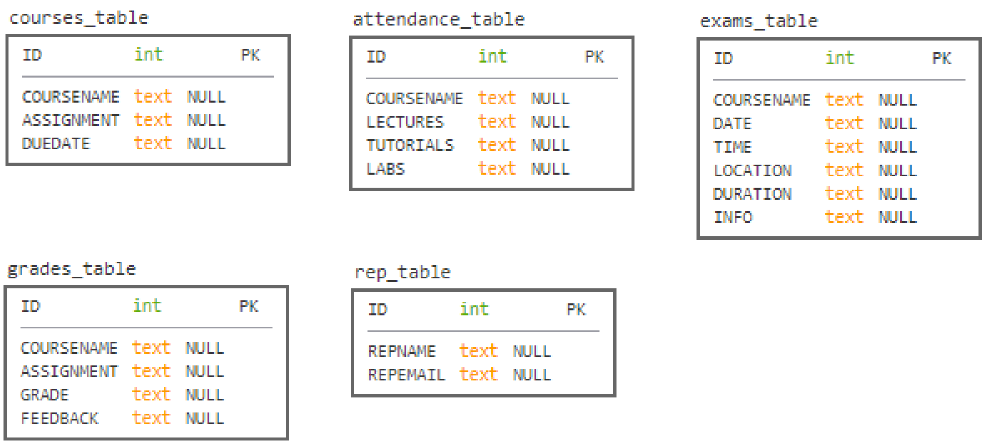
The **GradeRunner** application was built with **Android Studio**, the official integrated development environment for Google's Android operating system and originally based on JetBrain's Intellij IDEA. When initialising the project, the minimum SDK version chosen was API 15: Android 4.0.3 (IceCreamSandwich). This ensured the **GradeRunner** app would work on approximately 100% of devices. The main technologies used by Android Studio are Java programming for the purpose of creating 'activities' and XML for the purpose of creating layouts. In layman's terms, a layout is the visual presentation of what a user views on an app (such as a button), whereas an activity is some form of functionality (such as what the button does when pressed). In addition to Java and XML, the database component for the application was SQLite. Rather than a typical client-server database, SQLite is embedded into the Android operating system but still provides enough functionality for an application such as **GradeRunner**. While using a cloud-based database server such as Google's Firebase or Amazon's AWS were potential options, it was felt these services would create an unnecessary overhead for an application which is focused on swift performance.

During development of the application, previewing updates occurred in two main formats. First, Android Studio provides virtual devices, connected via the 'Android Debug Bridge' (ADB), which is a simulation environment designed to show the developer how their product will look on a real, physical device. There are a variety of options available when previewing work through the ADB, such as which model of Android phone or tablet should be simulated, which SDK should be used to compile the application, as well as all the functionality that would be expected with a real, physical android device: orientation; sound/mute options; power button; camera; and the generic home/back/menu buttons. In addition to virtual ADB simulated devices, it is also possible to connect a physical Android device to Android Studio via a USB cable (so long as the device is set to 'development mode') and run the application exactly as it would perform if installed on that device. The benefit of product testing with the ADB is that developers have the choice of nearly every Android phone or tablet on the current market, which can be very important in ensuring their app performs well regardless of the choice of Android product. However, ADB is notoriously slow – it takes a long time to initialise, and this can be frustrating, particularly if a developer has only edited one or two lines of code and wishes to quickly check their changes. Here, the benefit of a physically connected device is apparent as it is much quicker to initialise an application via a USB connected phone or tablet. For this project, a mixture of each method was used in order to test the **GradeRunner** app across multiple devices for the sake of thoroughness, while also not being slowed down when only minor updates needed to be checked.

Appendix C shows the development timetable; a development progression Gantt chart is included in **Appendix D**.

3.2.1 Database Schema

The database schema for each database helper class is demonstrated in the following diagram:



courses_table

ID	int	PK
COURSENAME	text	NULL
ASSIGNMENT	text	NULL
DUEDATE	text	NULL

attendance_table

ID	int	PK
COURSENAME	text	NULL
LECTURES	text	NULL
TUTORIALS	text	NULL
LABS	text	NULL

exams_table

ID	int	PK
COURSENAME	text	NULL
DATE	text	NULL
TIME	text	NULL
LOCATION	text	NULL
DURATION	text	NULL
INFO	text	NULL

grades_table

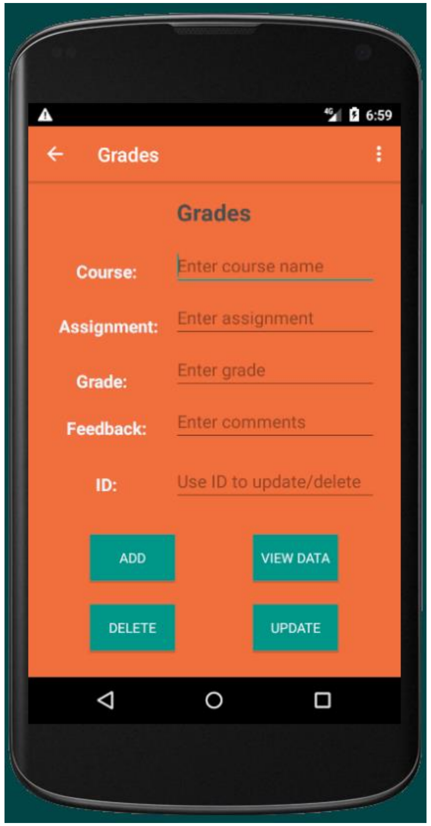
ID	int	PK
COURSENAME	text	NULL
ASSIGNMENT	text	NULL
GRADE	text	NULL
FEEDBACK	text	NULL

rep_table

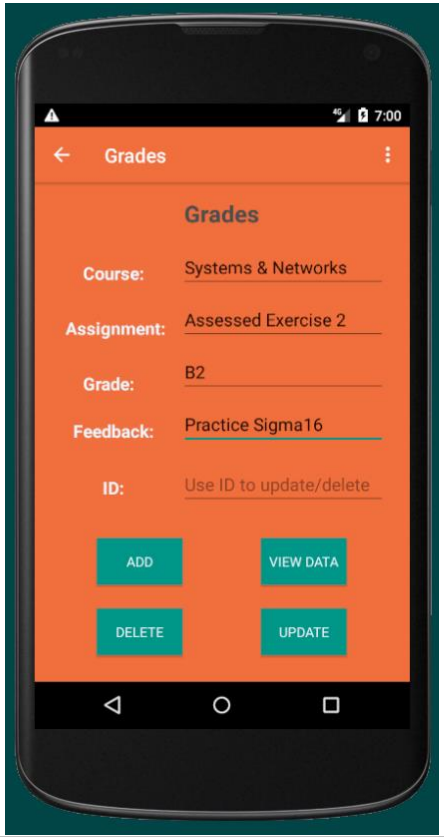
ID	int	PK
REPNAME	text	NULL
REPEMAIL	text	NULL

Figure 3: GradeRunner’s database schema

3.3 Screenshots



Grades section: empty text fields



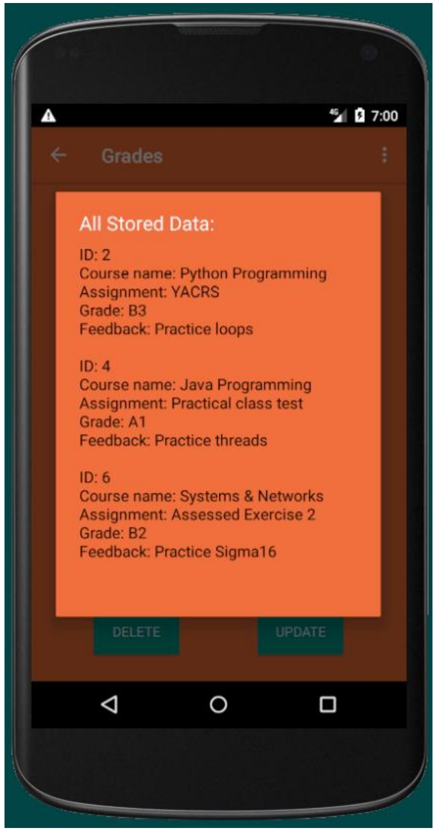
Course: Systems & Networks

Assignment: Assessed Exercise 2

Grade: B2

Feedback: Practice Sigma16

Text fields with data entered



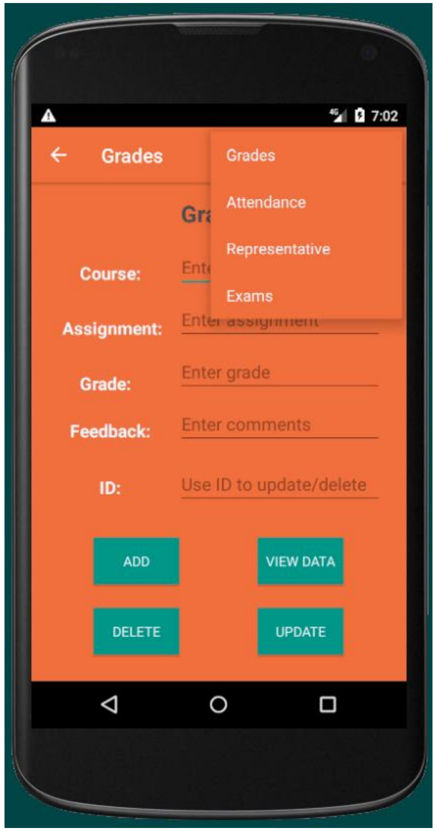
All Stored Data:

ID: 2
Course name: Python Programming
Assignment: YACRS
Grade: B3
Feedback: Practice loops

ID: 4
Course name: Java Programming
Assignment: Practical class test
Grade: A1
Feedback: Practice threads

ID: 6
Course name: Systems & Networks
Assignment: Assessed Exercise 2
Grade: B2
Feedback: Practice Sigma16

ADD button clicked then VIEW DATA button clicked



3-dot menu button clicked

Chapter 4 Testing & Evaluation

4.1 Explanation of evaluation strategy

The evaluation strategy of the **GradeRunner** app was conducted in two main parts. First, 12 participants (9 male and 3 female) were selected to try the app and then record their impressions and conclusions using a Computer System Usability Questionnaire. All 12 participants had prior experience of using mobile phone applications and were in the age range of ~25-35. When the participants were introduced to the **GradeRunner** application, no respondent stated that they regularly used a similar phone app to track their coursework requirements. Secondly, an evaluation of the functional and non-functional requirements of the application were conducted by myself in order to ensure the app performed as intended, and also experiment with different situations the application might be used in such as portrait/landscape mode, and how the system functions on different phone and tablet models.

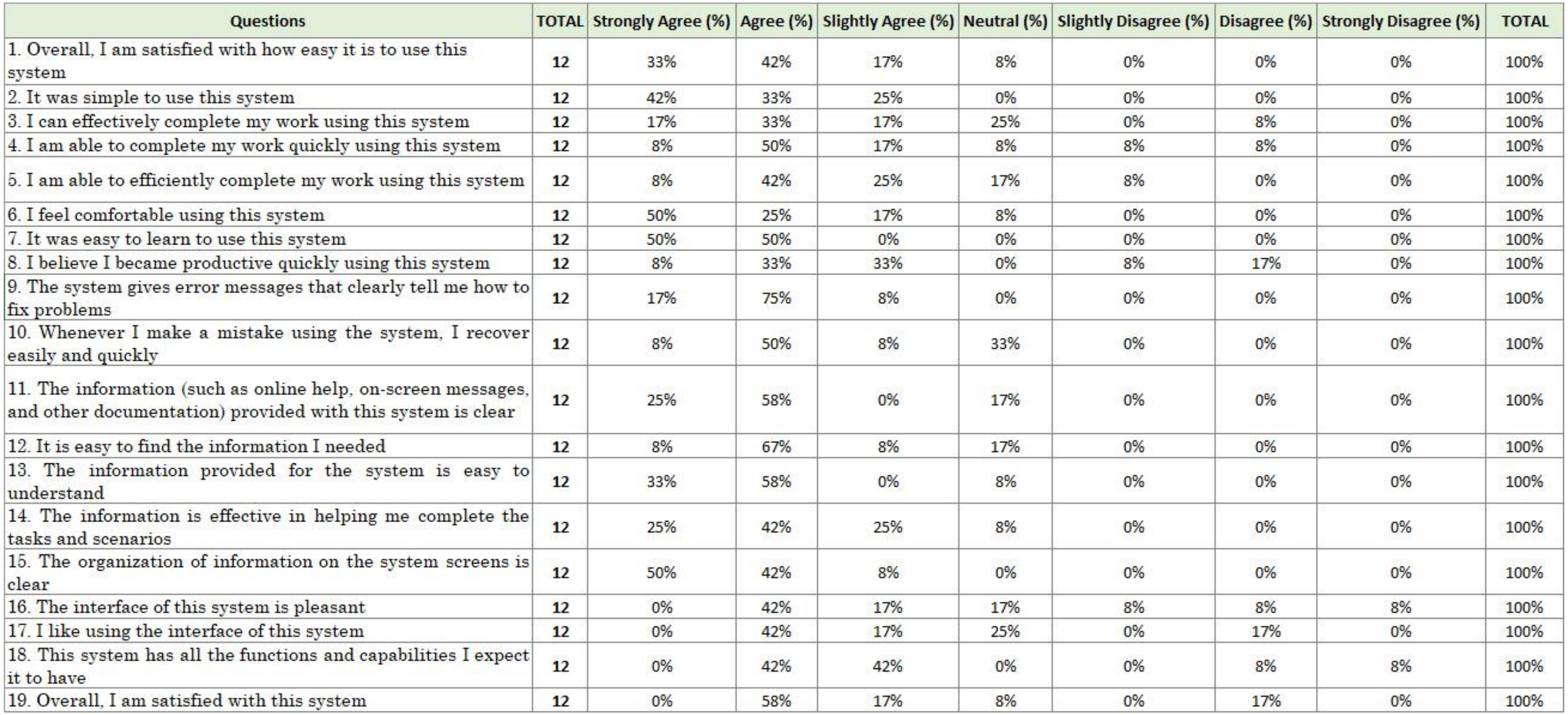
Participants were chosen from a mix of students, in addition to graduate friends and family. Testing was completely anonymous and the demonstrations were conducted using a laptop running Android Studio with the Nexus 4 mobile phone emulator. In order to determine whether the application was sufficiently self-explanatory, the research participants were given a cursory introduction:

"***GradeRunner*** *is an app for recording your academic coursework details – can you attempt to record and view some information on several sections of the app?*".

After the research participants had sufficiently engaged with the **GradeRunner** app, the individuals were asked to complete a Computer System Usability Questionnaire (Lewis, 1995) as demonstrated in **Appendix E**.

The specific areas of evaluation for the functional and non-functional requirements of the application were decided through drawing upon my own experience as a mobile phone application user and also as an experienced student. Where a virtual device was required to test these areas, a Nexus 4 mobile was the ADB choice; where a physical device was required to test these areas, a Nexus 5X mobile was connected via USB.

4.2 Software Testing: Computer System Usability Questionnaire



Questions	TOTAL	Strongly Agree (%)	Agree (%)	Slightly Agree (%)	Neutral (%)	Slightly Disagree (%)	Disagree (%)	Strongly Disagree (%)	TOTAL
1. Overall, I am satisfied with how easy it is to use this system	12	33%	42%	17%	8%	0%	0%	0%	100%
2. It was simple to use this system	12	42%	33%	25%	0%	0%	0%	0%	100%
3. I can effectively complete my work using this system	12	17%	33%	17%	25%	0%	8%	0%	100%
4. I am able to complete my work quickly using this system	12	8%	50%	17%	8%	8%	8%	0%	100%
5. I am able to efficiently complete my work using this system	12	8%	42%	25%	17%	8%	0%	0%	100%
6. I feel comfortable using this system	12	50%	25%	17%	8%	0%	0%	0%	100%
7. It was easy to learn to use this system	12	50%	50%	0%	0%	0%	0%	0%	100%
8. I believe I became productive quickly using this system	12	8%	33%	33%	0%	8%	17%	0%	100%
9. The system gives error messages that clearly tell me how to fix problems	12	17%	75%	8%	0%	0%	0%	0%	100%
10. Whenever I make a mistake using the system, I recover easily and quickly	12	8%	50%	8%	33%	0%	0%	0%	100%
11. The information (such as online help, on-screen messages, and other documentation) provided with this system is clear	12	25%	58%	0%	17%	0%	0%	0%	100%
12. It is easy to find the information I needed	12	8%	67%	8%	17%	0%	0%	0%	100%
13. The information provided for the system is easy to understand	12	33%	58%	0%	8%	0%	0%	0%	100%
14. The information is effective in helping me complete the tasks and scenarios	12	25%	42%	25%	8%	0%	0%	0%	100%
15. The organization of information on the system screens is clear	12	50%	42%	8%	0%	0%	0%	0%	100%
16. The interface of this system is pleasant	12	0%	42%	17%	17%	8%	8%	8%	100%
17. I like using the interface of this system	12	0%	42%	17%	25%	0%	17%	0%	100%
18. This system has all the functions and capabilities I expect it to have	12	0%	42%	42%	0%	0%	8%	8%	100%
19. Overall, I am satisfied with this system	12	0%	58%	17%	8%	0%	17%	0%	100%

Figure 4: Excel analysis of Computer System Usability Questionnaire

Questionnaire Analysis

Q1. Overall, I am satisfied with how easy it is to use this system

A large majority of respondents thought **GradeRunner** was easy to use with 75% of individuals strongly agreeing or agreeing. Not a single participant disagreed.

Q2. It was simple to use this system

A large majority of respondents felt the system was simple to use. Seventy-five percent of individuals either strongly agreed or agreed. A quarter of participants slightly agreed while no-one disagreed.

Q3. I can effectively complete my work using this system

A third of respondents agreed with this statement, while a quarter were neutral. Eight percent of participants disagreed while just over a third either slightly or strongly agreed (34%).

Q4. I am able to complete my work quickly using this system

Half of respondents agreed with this statement while 17% slightly agreed. Almost a quarter were neutral, slightly disagreed or disagreed (24%).

Q5. I am able to efficiently complete my work using this system

Just over two-thirds of participants either agreed or slightly agreed with this statement (67%). Seventeen percent were neutral while 8% slightly disagreed.

Q6. I feel comfortable using this system

A large majority of respondents either strongly agreed or agreed with this statement (75%). No respondent disagreed.

Q7. It was easy to learn to use this system

All participants either strongly agreed (50%) or agreed (50%) with this statement.

Q8. I believe I became productive quickly using this system

A third of respondents either agreed or slightly agreed with this statement. Eight percent slightly disagreed while 17% disagreed.

Q9. The system gives error messages that clearly tell me how to fix problems

Three-quarters of respondents agreed with this statement while 17% strongly agreed. No participants disagreed.

Q10. Whenever I make a mistake using the system, I recover easily and quickly

Half of respondents agreed with this statement while a third were neutral.

Q11. The information (such as online help, on-screen messages, and other documentation) provided with this system is clear

A large majority of respondents either strongly agreed or agreed with this statement (83%). No-one disagreed with the statement.

Q12. It is easy to find the information I needed

Over two-thirds of participants agreed with this statement (67%). No respondent disagreed.

Q13. The information provided for the system is easy to understand

Over half of respondents agreed with this statement (58%) while a third strongly agreed. No-one disagreed with the statement.

Q14. The information is effective in helping me complete the tasks and scenarios

Half of the participants either slightly agreed or strongly agreed with this statement, while 42% agreed. No-one disagreed.

Q15. The organization of information on the system screens is clear

The vast majority of respondents either strongly agreed or agreed with this statement (92%). Not a single participant disagreed.

Q16. The interface of this system is pleasant

Fifty-nine percent of respondents either agreed or slightly agreed with this statement. Almost a quarter of participants either slightly disagreed, disagreed or strongly disagreed with the statement (24%).

Q17. I like using the interface of this system

Fifty-nine percent of respondents either agreed or slightly agreed with this statement. A quarter were neutral while 17% disagreed.

Q18. This system has all the functions and capabilities I expect it to have

A large majority of participants either agreed or slightly agreed with this statement (84%). Sixteen percent either disagreed or strongly disagreed.

Q19. Overall, I am satisfied with this system

Three-quarters of respondents either agreed or slightly agreed with this statement. Eight percent were neutral while 17% disagreed.

4.3 Evaluation: Functional and Non-Functional Requirements

Functional Requirements			
Test	Description	Expected Result	Actual Result
Creating and viewing items in any of the Assignments, Grades, Attendance, Representative or Exams sections	Entering data in any section of the app, clicking the ADD DATA button then clicking the VIEW DATA button	The data would be written to the database and appear in the pop-up list	✓ The data is correctly saved in the database and presented in the pop-up view for all sections of the app
Updating and viewing items in any of the Assignments, Grades, Attendance, Representative or Exams sections	Altering data in any section of the app, clicking the UPDATE button, then clicking the VIEW DATA button	The update should overwrite the information in the database and appear in the pop-up list	✓ The updated information is correctly overwritten in the database and presented in the pop-up view for all sections of the app
Deleting items in any of the Assignments, Grades, Attendance, Representative or Exams sections	Entering an item ID number in any section of the app, clicking the DELETE button, then clicking the VIEW DATA button	The selected item should be removed from the pop-up view	✓ The selected item is correctly removed from the database and does not appear in the pop-up view for all sections of the app
Trying to delete or update items in any of the Assignments, Grades, Attendance, Representative or Exams sections without specifying an item ID number	Clicking the DELETE button, or altering data and clicking the UPDATE button, without an item number being entered into the ID text field	A 'toast' pop-up should say 'you must enter an ID'	✓ The app correctly displays the error message if the user tries to update or delete without entering an item ID number

Non-Functional Requirements			
Test	Description	Expected Result	Actual Result
Portrait/Landscape	Changing the orientation of the phone from portrait to landscape	As the app is supposed to be used in portrait mode (and subsequently has no landscape xml layouts), the view would be unusable	✓ The expected result accurately predicts that the buttons and text views are 'off screen' in landscape mode
Android back button	Clicking the back button of the Android device	The app should return to the previous section, not the home Assignments section	✓ The app correctly returns to the previously visited section
App back button	Clicking the back arrow of the app	The app should return to main Assignments section, not the previously visited section	✓ The app correctly returns to the main Assignments section
Running another app at same time as GradeRunner	A chess application was opened at the same time	The newly opened up should become the forefront application while GradeRunner is paused in the background	✓ GradeRunner correctly waits in the background and can be selected to become the forefront application
Swapping wifi to mobile connection	Switching off the phone's wifi connection to force a network connection	Changing between wifi or network connections should have no effect on the app	✓ There is no effect on the application when swapping between a wifi or network connection
Different devices: physical and emulator	The app should function normally regardless of being tested using Android Studio's ADB virtual devices or if an actual physical device is connected	The application will function exactly the same if tested on a virtual device or a physical device	✓ The application functions exactly the same on a virtual device as it does a physical device

Chapter 5 Conclusion

5.1 Product Status & Achievements

GradeRunner is a functional app which achieves the objectives as originally envisaged during the requirements gathering stage. There are no major technical issues which lead to the app freezing or behaving unexpectedly, which implies the underlying code is robust. The Java and XML code interacts with each other correctly to produce easy-to-use menu interfaces and buttons, which successfully write text data to the SQLite database when a user interacts with the software.

Ultimately, what has been achieved is the production of a self-explanatory, lightweight mobile phone application which does not try to be too many things at once. As a developer, it is important to learn the value of software which is straightforward for users and not to be tempted to 'gold plate' an application with bloated and unnecessary features.

From a technical perspective, the main achievements can be attributed to learning how apps are built using the Android Studio IDE. Specifically, there was a learning curve in understanding how activities are created using Java and how layouts are created using XML. Once these concepts were understood, a further learning curve was travelled when experimenting with object and text field positions, and implementing the code to create button and database functionality.

Feedback from the user testing reveals several important conclusions. First, most users feel **GradeRunner** is easy to use, not difficult to understand and is largely a satisfying application which has a fairly high usability value. However, feedback also reveals that the aesthetics and presentation of the application could be developed further – **GradeRunner's** minimalistic visual layout perhaps feels overly basic for some users. Nonetheless, the questionnaire evidence reveals that this basic layout leaves a system user in no doubt of what the app does, which section of the app they are engaging with, and what mistake has been made should an error message appear. Ultimately, aesthetic choices when developing digital products straddle a fine line between making the application look attractive and professional, but retaining clarity of where the user is, what the user can do, where the user can go and where the user has been.

During the development period, as expected, a lot of time was spent debugging problems – this highlights the need for developers to include a generous amount of time to solving problems when predicting how long a development period will last. In **GradeRunner's** case, debugging issues were largely attributed to, first of the learning process involved when using the unfamiliar Android Studio IDE, but secondly also the technical detail required when communicating with a database. Nonetheless, the solving of such problems and bugs should be noted as an achievement and the experience has contributed significantly to my own personal development as a software engineer.

5.2 Weaknesses and Limitations

From my own perspective as the developer of **GradeRunner**, in addition to the user feedback, I feel the app's biggest weakness is the lack of sophistication in its aesthetic presentation. Such disappointment in how the application is visually presented highlights just how large a task it is for a product to be developed by a single person, and offers a significant insight as to why professional development companies have specific departments separated into dedicated areas such as programming, graphic design and testing.

Furthermore, while previously stated that it is important an application does not try to be too many things at once, it is equally important that an app has enough features and functionality to become a popular product. **GradeRunner** could be criticized for lacking in features, a concern that was slightly elucidated from user feedback. However, an argument could be made that the urgency of deadlines encourages the production of functional software (even if that means lacking in features), rather than a tardy approach which produces non-functional software that overruns a deadline due to attempting to implement too many unnecessary features. Again, this is a fine line which any software developer must become accustomed too.

5.3 Future Work

Given more time and/or development team members, there are several areas that **GradeRunner** could be expanded on. First is the obvious fact that the software is an Android-only app in its current state, and so an Apple OS version would be an advantageous conversion to make. However, I feel the biggest improvement would be a connection between the application and the school/college/university department that the student user is matriculated with. This way, information such as coursework deadlines and exam timetables could be updated *ad hoc* from the academic departments, rather than the student having to enter in their own information.

As previously mentioned, presentation and aesthetics would also be an obvious area to make improvements on. I feel the orange/green theme works well together, but there are many more options for combinations of fonts, button styles and object locations throughout the application.

Alarms and notifications were included in the 'could have' section of the MoSCoW analysis. This would be an obvious benefit to a student where reminder notifications could encourage the app user to prioritise a particular piece of coursework. I would also change the field type of the student representative's email address so that when a user taps/clicks this field, the email client of their phone opens up, ready to send a message to the representative.

In conclusion, overall – I very much enjoyed the experience of creating **GradeRunner**. I have plans to submit the app to the Google Play store. While I doubt their quality control team would allow such a basic application to be made available to the public for sale, if it makes me a millionaire, be in no doubt I shall make a generous donation to the University of Glasgow as a thanks for 12 months of extreme hard work, but extreme great fun. It has been a pleasure.
Many thanks.

References

Chaffin, J. (2018). *UK universities act to tackle student mental health crisis.* Financial Times. Retrieved June 6, 2018, from https://www.ft.com/content/8e53ee5a-1e13-11e8-aaca-4574d7dabfb6

Dodgson, L. (2018). *Lonely millennials are at a greater risk of developing anxiety and depression — but the reasons for their isolation are unclear.* Business Insider UK. Retrieved June 6, 2018, from http://uk.businessinsider.com/lonely-millennials-twice-as-likely-to-develop-anxiety-and-depression-2018-4

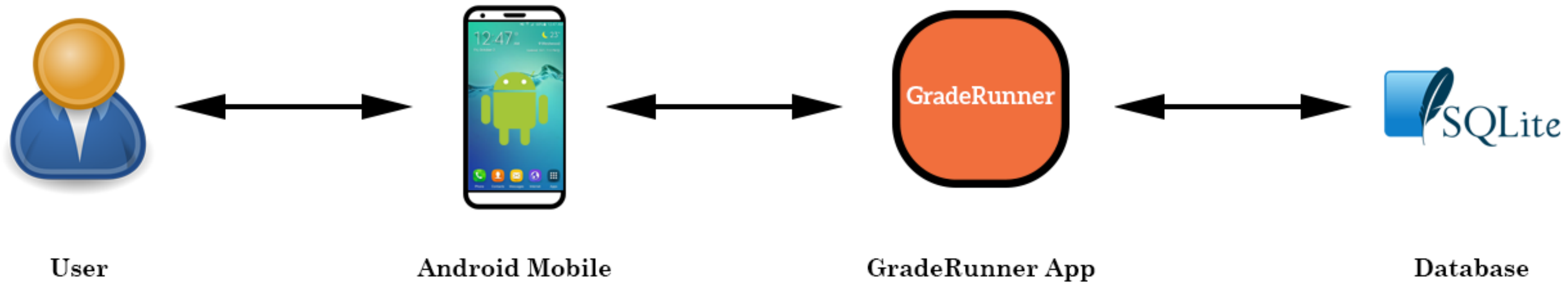
Lewis, J. R. (1995). IBM Computer Usability Satisfaction Questionnaires: Psychometric Evaluation and Instructions for Use. *International Journal of Human-Computer Interaction*, 57-78.

Quinn, B. (2017). *Students struggling with finances in 'desperate state', claims union.* The Guardian. Retrieved June 6, 2018, from https://www.theguardian.com/money/2017/aug/26/average-student-food-budget-less-than-25-a-week-nus

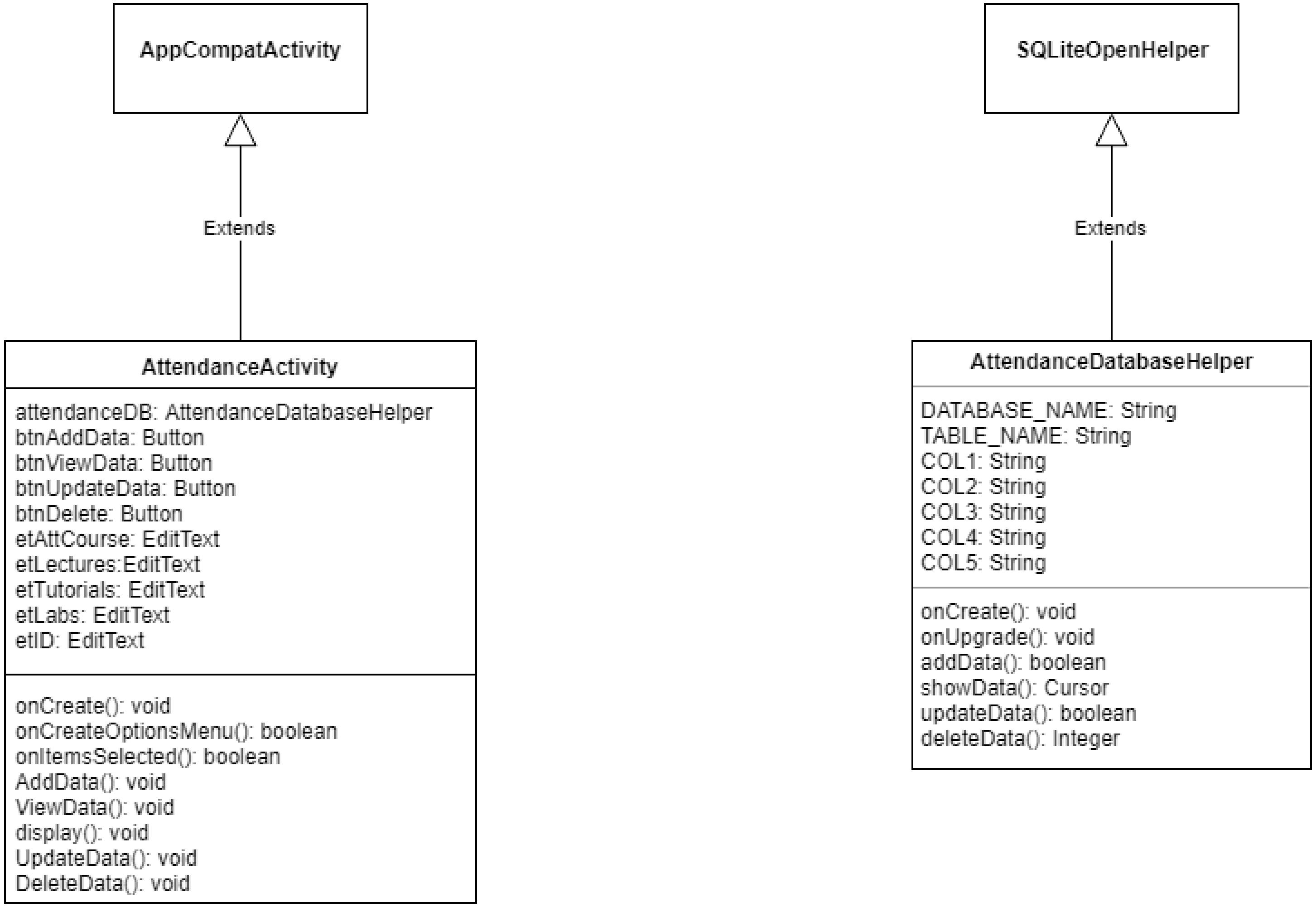
Rudgard, O. (2018). *Universities have a suicide problem as students taking own lives overtakes general population.* Telegraph. Retrieved June 6, 2018, from https://www.telegraph.co.uk/news/2018/04/12/universities-have-suicide-problem-students-taking-lives-overtakes/

Weale, S. (2018). *University drop-out rates in UK rise for third successive year.* The Guardian. Retrieved June 6, 2018, from https://www.theguardian.com/education/2018/mar/08/university-drop-out-rates-uk-rise-third-year

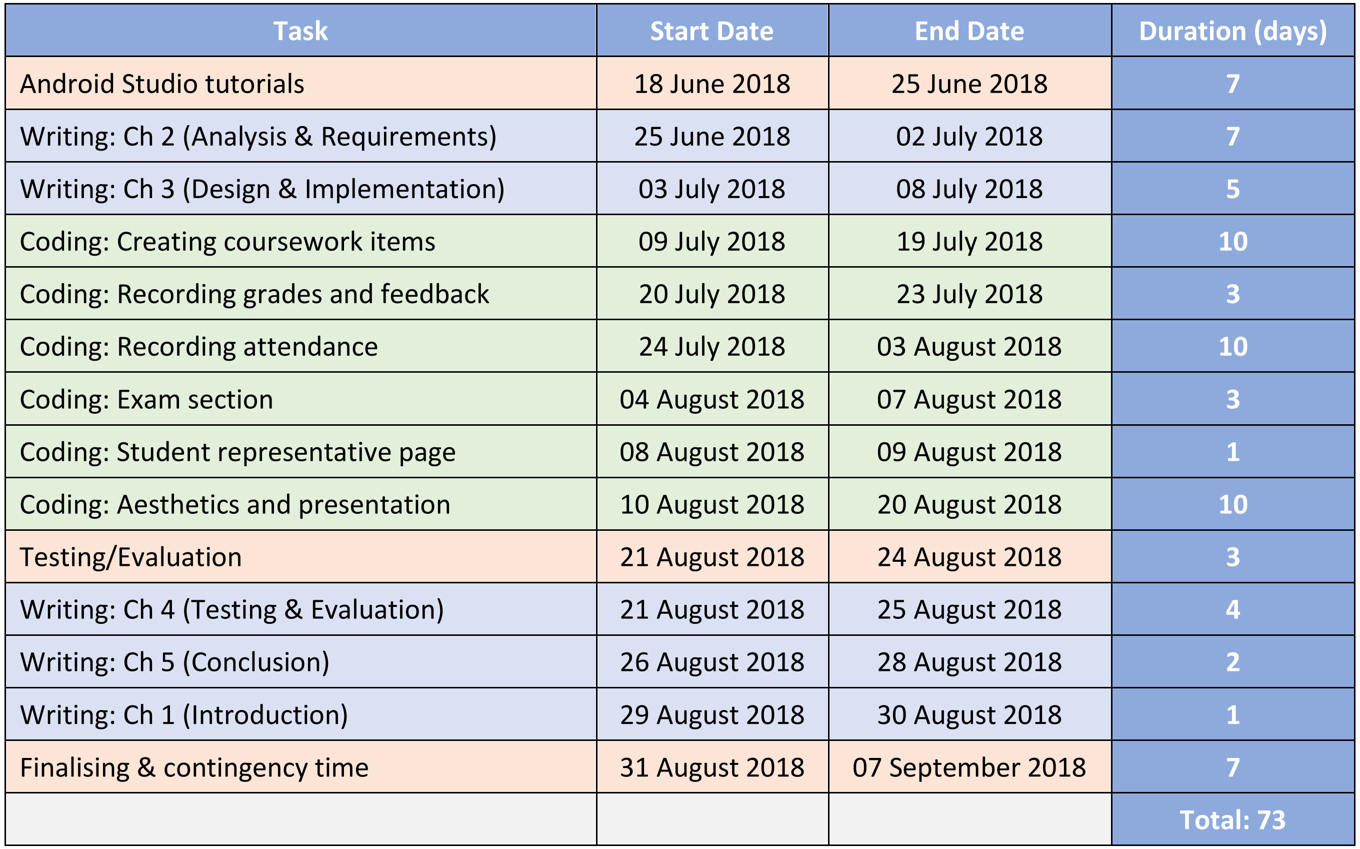
Appendix A – High Level Architecture



Appendix B – UML Example

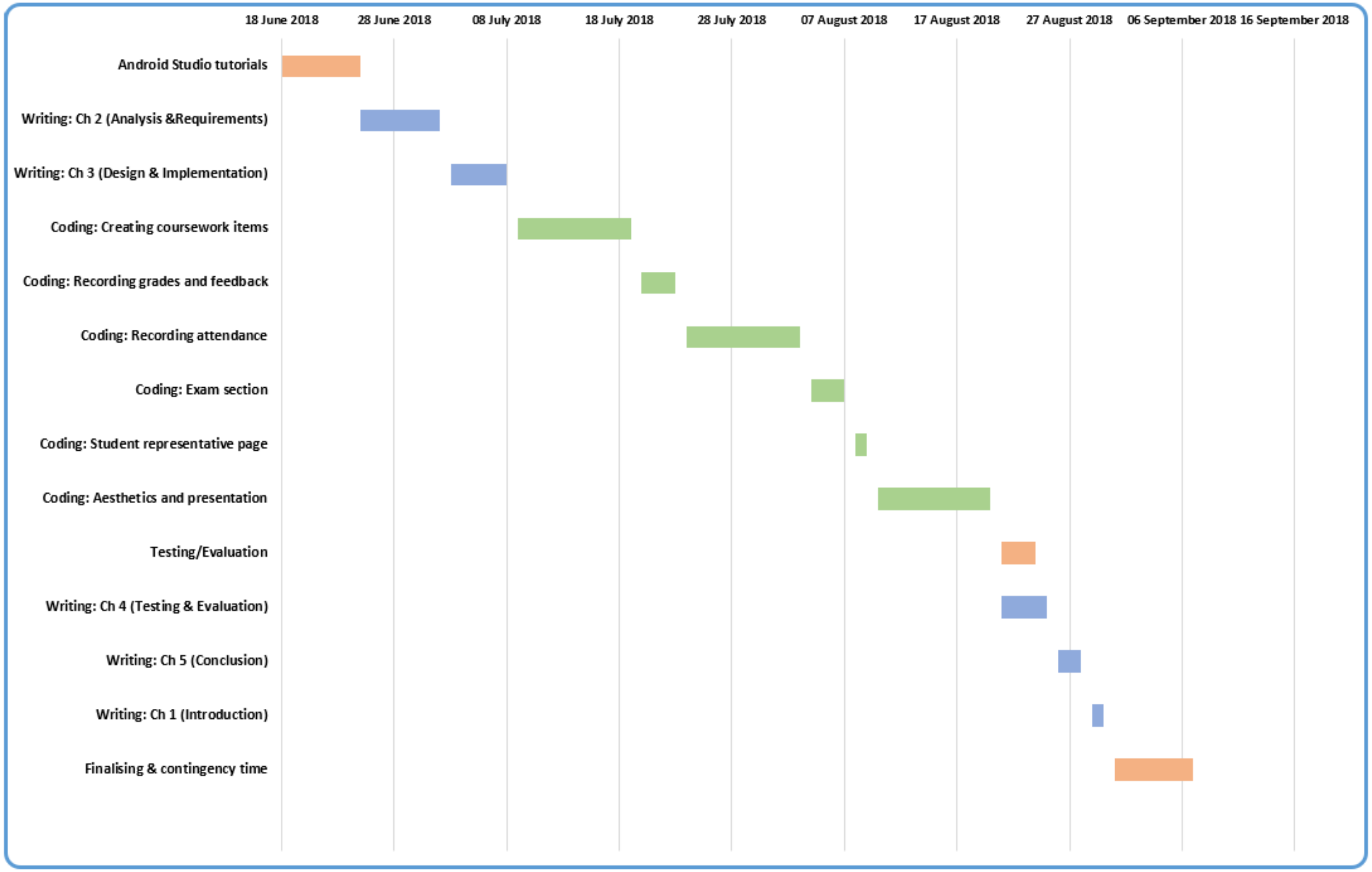


Appendix C – Timetable



Task	Start Date	End Date	Duration (days)
Android Studio tutorials	18 June 2018	25 June 2018	7
Writing: Ch 2 (Analysis & Requirements)	25 June 2018	02 July 2018	7
Writing: Ch 3 (Design & Implementation)	03 July 2018	08 July 2018	5
Coding: Creating coursework items	09 July 2018	19 July 2018	10
Coding: Recording grades and feedback	20 July 2018	23 July 2018	3
Coding: Recording attendance	24 July 2018	03 August 2018	10
Coding: Exam section	04 August 2018	07 August 2018	3
Coding: Student representative page	08 August 2018	09 August 2018	1
Coding: Aesthetics and presentation	10 August 2018	20 August 2018	10
Testing/Evaluation	21 August 2018	24 August 2018	3
Writing: Ch 4 (Testing & Evaluation)	21 August 2018	25 August 2018	4
Writing: Ch 5 (Conclusion)	26 August 2018	28 August 2018	2
Writing: Ch 1 (Introduction)	29 August 2018	30 August 2018	1
Finalising & contingency time	31 August 2018	07 September 2018	7
			Total: 73

Appendix D – Gantt Chart



Appendix E – Computer System Usability Questionnaire

Strongly Disagree (-3)	Disagree (-2)	Slightly Disagree (-1)	Neutral (0)	Slightly Agree (+1)	Agree (+2)	Strongly Agree (+3)

1. Overall, I am satisfied with how easy it is to use this system ________
2. It was simple to use this system ________
3. I can effectively complete my work using this system ________
4. I am able to complete my work quickly using this system ________
5. I am able to efficiently complete my work using this system ________
6. I feel comfortable using this system ________
7. It was easy to learn to use this system ________
8. I believe I became productive quickly using this system ________
9. The system gives error messages that clearly tell me how to fix problems ________
10. Whenever I make a mistake using the system, I recover easily and quickly ________
11. The information (such as online help, on-screen messages, and other documentation) provided with this system is clear ________
12. It is easy to find the information I needed ________
13. The information provided for the system is easy to understand ________
14. The information is effective in helping me complete the tasks and scenarios ________
15. The organization of information on the system screens is clear ________
16. The interface of this system is pleasant ________
17. I like using the interface of this system ________
18. This system has all the functions and capabilities I expect it to have ________
19. Overall, I am satisfied with this system ________