

CS310 Computer Science Project
Final Report
The Creation of an Android Application to Aid a
Student's Revision of Edexcel GCSE
Mathematics

Ryan O'Hara
u1702761

May 4, 2020

Contents

1	Abstract	5
2	Introduction	5
2.1	Problem Outline	5
2.2	Project Specification	6
2.3	Report Structure	9
3	Background	10
3.1	MyMaths	10
3.2	Khan Academy	11
3.3	Maths Watch	12
3.4	Independent Revision Tools	12
3.5	Application Development	14
4	Methodology	14
5	Design	16
5.1	Visual Design	16
5.2	Control Flow	19
5.3	Data Flow	21
6	Implementation	22
6.1	Fragments	22
6.2	Questions	23
6.3	Control and Data Flow	27
7	Testing	28

7.1	White Box Testing	29
7.2	Black Box Testing	30
8	Risks	30
8.1	Scope Creep	31
8.2	Personal Issues	32
8.3	Poor Project Management	32
9	Project Management	34
9.1	Schedule	34
9.1.1	Project Specification Schedule	34
9.1.2	Progress Report Schedule	35
9.1.3	Schedule Evaluation	36
9.2	Tools	37
9.2.1	Android Studio	38
9.2.2	Git	38
9.3	Miscellaneous Project Management	39
10	Results and Discussion	39
10.1	Requirements	39
10.2	Discussion	41
11	Future Work	42
11.1	Project Improvements	42
11.1.1	Question Storage	42
11.1.2	Accessibility Improvements	43
11.1.3	Content Grouping	43

11.1.4 Trello	44
11.2 Project Extensions	45
11.2.1 Currently Incomplete Requirements	45
11.2.2 User Validation	46
11.2.3 User Added Questions	46
11.2.4 MyMaths Homework Feature	47
12 Legal, Social, Ethical, and Environmental Issues	48
12.1 Legal Issues	48
12.1.1 Android Studio	48
12.1.2 Git	49
12.2 Ethical Issues	49
13 Conclusion	49
14 References	50

1 Abstract

This report will finalise the development of an Android application designed to be used for revision purposes by students sitting Edexcel GCSE Mathematics in November 2021 or later. This report will explain the decision to produce this application, as well as provide both a high-level and low-level view of the application. I will discuss the problem which this project aims to solve, as well as other similar products, in order to justify the decision to produce this application.

This report will also analyse the success of the project in regard to whether all requirements have been met, as well as whether the application solves the problem as intended. The risks associated with this project will be evaluated and discussed, as will the project management techniques which were followed throughout development. Any potential issues concerning this project will be evaluated.

2 Introduction

The introduction of this report aims to complete three tasks. The first task is to provide an understanding of the problem which this project wishes to solve. The second task is to explain the features required by this project to be deemed successful. These features were laid out in the project specification document[34] which was produced at the beginning of the project. Finally, this introduction aims to provide the reader with an understanding of the report structure, giving brief descriptions of each section in the report to explain their purpose.

2.1 Problem Outline

The difficulty of GCSE Mathematics recently increased due to a change in specification by Edexcel. This increase in difficulty was due to the UK Government's reform of the English and Mathematics GCSEs in November 2013[1]. The relevant section from the Department of Education's written statement to parliament is as such: "The new mathematics GCSE will demand deeper and broader mathematical understanding ... The new mathematics GCSE will be more demanding". This demand for a deeper and broader mathematical understanding enforced a syllabus change for Edexcel's GCSE Mathematics qualification. The teaching of this new syllabus commenced in 2015, and it was first examined in 2017[2].

This increase in difficulty for GCSE mathematics is achieved through the introduction of more content into the specification. Because of this increase

in content, a higher proportion of the academic year is dedicated to teaching, and students are left with less time in the classroom to revise for their exams. This leaves students in the position of having to revise topics for their final examinations at home, typically without the support of a teacher or tutor.

A study from 2018 demonstrated that 95% of American teenagers have access to a smartphone[3]. Also, a study conducted in 2017 stated “Recent statistics show that around 69% of the total digital media time spent by Americans was taken up by mobile devices and 87% of the total mobile time was spent in apps in December 2016”[4]. More students than ever before have access to smartphones, and they are spending the majority of their time on these smartphones in applications. More students have access to a smartphone than have access to a desktop or laptop computer as of 2017[3].

Because of this increase in the availability of smartphones, students are now turning towards technology to aid their learning and revision for their GCSE exams. This project aims to build a mobile application which can be used on the Android mobile operating system, which is the majority operating system (OS) for smartphone devices with a market share of 73% between February 2019 and February 2020[5]. This allows the application to be available to the majority of students studying for GCSE mathematics.

This project aims to create an application which focuses on testing a student’s current knowledge instead of teaching them new information. It is designed to be used alongside classroom teaching, or other less conventional teaching methods which aim to replace a classroom-like environment, such as Khan Academy. This application aims to focus on testing skills already gained by the user from a classroom-like environment, and will emulate final year GCSE Mathematics questions to suitably prepare students for their end-of-year exams.

This application will have to provide users with the ability to practice skills gained from classroom learning. It must contain questions of GCSE exam standard so that students can effectively prepare for their exams using the application. It must also have the ability for users to input answers, and it must mark the answers and provide visual feedback to the user. A set of requirements created to complete these tasks is available below in the project specification subsection of this introduction.

2.2 Project Specification

This project aims to create an android application similar to the MyMaths homework functionality discussed in the background section of this report. This will enable students to attempt exam questions in a specific topic in Edexcel GCSE Mathematics by inputting an answer. This answer will then be compared to the pre-calculated answer, and the student will get a visual response as to

whether they have answered the question correctly or incorrectly. Where the student answered the question incorrectly, the correct answer will be displayed to the student so they can compare their answer with the correct one.

To achieve this task a set of requirements were detailed in this project's specification[34]. These requirements follow the MoSCoW method for specifying requirements. The MoSCoW method groups requirements into four categories: must, should, could, and won't. All of the 'must' requirements are vital to the success of the project and so need to be completed by the end of the project. At the completion of all of the 'must' requirements, a Minimum Viable Product (MVP) is created and the product could be shipped. 'Should' requirements are features which are not essential to the creation of an MVP but are features which are expected in the final product. 'Could' requirements are features which are not expected in the final product, but if all other requirements are completed before the deadline, the 'could' requirements will be implemented. 'Won't' requirements are features which will not, under any circumstances, be present in the final product.

The requirements present in the project specification are as follows.

The Android application which will be created for this project must:

1. Have a start page with several clickable buttons.
2. Be able to run on Android devices with OS versions between Android 6 (Android Marshmallow) and Android 10.
3. Have a list of each module found in Edexcel's GCSE Mathematics Specification[6], with each module being a selectable button.
4. Generate revision questions where numerical values are generated by Java's inbuilt random number generator and are appropriate for the difficulty of the question.
5. Give the user the ability to input numerical answers to the revision questions generated (requires 4)
6. Provide the user with feedback relating to whether their answer matches the correct answer for the question, and where the user's answer is incorrect, display the correct answer.

The Android application which will be created for this project should:

7. Have a variety of testing difficulties, ranging from easy to hard for each module.
8. Have a way of tracking user progress over a number of days or tests per module (requires 5).

9. Display model methods for working out the correct answer for a question where the user inputted an incorrect answer (requires 6).

The Android application which will be created for this project could:

10. Include resources similar to those which would be used in a classroom environment to teach the user about a topic.

Project success was also defined in the project specification. To be deemed successful, the final product produced by this project will have completed:

- all of the ‘must’ requirements defined above
- at least one of the ‘should’ requirements defined above
- the ‘could’ requirement does not have to be completed

The target audience for the final product of produced by this project is students sitting Edexcel GCSE Mathematics examinations. The average age of students sitting these exams are between 15 and 16 years old. Although, students as early as year 7 (age 11 - 12) begin learning content which is examinable in the GCSE Mathematics exam. The application will be primarily designed for 15-16 year olds, but needs to be accessible to students as young as 11. This means there are a group of non-functional requirements which are not outlined in the project specification, but were created during development to ensure the target audience was taken into account. These non-functional requirements are shown below:

- the language used in the application should be at a year 7 reading level
- the application should be simple and minimalistic so as to not confuse new users
- an 11 year old who has not previously used the application but is familiar with smartphones should be able to use the application effectively with minimal prompting

These functional requirements enable anyone learning the content for the Edexcel GCSE Mathematics exams to be able to benefit from the application.

2.3 Report Structure

This report will start by running through the background to this project. Section 3 will outline the current landscape of the problem by exploring what has currently been achieved by other people in the field. It will be followed by section 4, which will be an explanation of the software development methodology which was used throughout the development of this application. The methodology section will explain how the work was completed, for example, it will outline a typical development session. It will also explain any changes made to this methodology throughout this project.

Section 5 will outline the design of the final product. It will provide a high level overview of this application. This high level view will explain the control and data flows of the program, and any specific design choices which were made throughout the development of this project. Section 6 will discuss the specific implementation details of the project. For example, how the questions were created in the application. Testing will be discussed in section 7. This will consist of outlining how testing was conducted, and the results of user testing.

The risk analysis conducted throughout this project will be explored in the risks section, which is section 8. An outline of risks will be produced, followed by how each risk was handled, which will then be followed by the effect each risk had on the final application. Project management will be discussed in section 9. This section will explain several details of the project that cannot be understood by only viewing the source code, for example, the development schedule, or weekly meetings with key stakeholders. These details do not appear in the source code but have a great impact on the success of the project. The results section will follow the project management section, and will be section 10. This is where the requirements will be evaluated and the success of the project will be determined.

The future work section is section 11, which is extra work that could be conducted to improve this project, and turn it from a proof of concept into an application which could be released onto the Google Play store. Section 12 will discuss the legal, social, ethical, and environmental issues associated with this project. The conclusion is section 13, which will summarise key findings in the report.

The final section in this report will be the references section, with a section number of 14. Harvard referencing style is adhered to throughout this report.

3 Background

There are three main GCSE Mathematics revision tools in use today that are recommended by schools. All of the main revision tools are explained in depth in their respective sections. Only one of the three main revision applications recommended by schools has a fully functional mobile application. There are also a number of other independent revision tools that are not actively recommended by schools, which can be downloaded and used on mobile devices. Some of these tools are explored in the independent revision tools subsection. Finally, this section will discuss the research conducted at the beginning of this project into Android application development.

3.1 MyMaths

The first tool is MyMaths[7], which is a website that aims to enrich a student's classroom learning experience by supplementing material covered in the classroom. The main feature of MyMaths is the online homework feature. This feature enables teachers to set homework to their students. This homework covers a specific topic, for example Venn diagrams, and the students have to answer questions on this topic. Example homework questions on the topic of Venn diagrams are displayed in Figure 1.

The screenshot shows a homework interface for 'Conditional probability 1'. It features a Venn diagram with two overlapping circles, T (tea) and C (coffee), within a universal set ξ . The numbers in the diagram are: 47 in the T-only region, 24 in the intersection, 37 in the C-only region, and 8 outside both circles. The interface includes two questions, Q1 and Q2, with a 'Total' score section on the left and a 'Mark it' button at the bottom right.

Question	Score	Answer	Mark
Q1: The Venn diagram shows the number of people who like tea (T) and who like coffee (C).	0/8		
Q2: You can click on the Venn diagram to help you. It is not marked.	0/13		
A person is selected at random. What is the probability that...			
they like coffee?		<input type="text"/>	[2]
they only like tea?		<input type="text"/>	[2]
A person who likes tea is selected. What is the probability that...			
they like coffee too?		<input type="text"/>	[2]
they only like tea?		<input type="text"/>	[2]
Total	0/21		

Figure 1: Venn diagram homework questions from the MyMaths website.

The student's scores are then calculated as a percentage and sent back to

the teacher.

The values generated for every homework are unique each time a student attempts a homework. This prevents a student from memorising a specific answer to pass the homework, also known as ‘answer memorisation’ and is discussed in detail in the implementation section of the report. A teacher can also see the number of attempts that a student required to complete the homework, and their highest score. Providing teachers with the ability to see the number of attempts that a student needed to complete the homework for a particular topic highlights where a student is struggling. It can enable one on one teaching to boost the student’s knowledge in that particular topic. A student can also attempt a homework which has not been set by their teacher. This allows any student to revise any topic in GCSE Mathematics on their own.

MyMaths also gives students the ability to take a lesson in any topic in GCSE Mathematics. These lessons walk through an example question in the topic, and then provide practice questions which are of a similar style to the homework questions. These questions are then marked, and where the student answers the question incorrectly, the correct answer is displayed next to the answer box.

MyMaths has an iOS and Android application for tablets, but not a specific application for smartphones[8]. This prevents students from utilising MyMaths as a revision tool, due to no regular access to a tablet.

3.2 Khan Academy

The second main GCSE Mathematics revision tool recommended by schools is Khan Academy[6], which is a non-profit organization which aims to “provide a free, world-class education for anyone, anywhere”[9]. Khan Academy’s main function is to teach new information to people. It does this through the use of short, lecture-like videos covering particular concepts. For example, there is a six minute video covering how to find a side of a triangle using the sine rule given two angles and another side[10]. These videos aim to replace classroom teaching for those who are unable to attend school. There is a video for every topic in GCSE Mathematics and so a student could exclusively use Khan Academy to learn the content for their exams. Khan Academy also has a testing feature, where for a certain topic four questions come up which the user has to answer one by one. This is, however, not available for all topics, and the user is limited to answering four questions at a time.

Khan Academy have a mobile application in both major app stores, the Apple app store[11] and the Google Play store[12]. Their application runs on all major devices and has the same functionality as the website version of Khan Academy. Users can create an account for free to keep track of their progress,

although anyone can access videos for all topics available on their website and application without one. This gives everyone with access to an internet connection the ability to watch any of Khan Academy's videos.

3.3 Maths Watch

Maths Watch[13] is an older tool which was used for teaching purposes in schools. It provides educational videos for students to learn from, as well as providing walk-throughs of past exam paper questions. It is designed to replace classroom teaching, and be used as a homework tool. Maths Watch is a subscription service, which only schools can subscribe to. If a student's school is not subscribed to Maths Watch then a student cannot access it.

Maths Watch has two main methods for distributing their video content. The first method is through DVDs posted to schools who buy them. Students who attend these schools can then buy these discs containing all of the videos provided by Maths Watch. The second method is through their online portal, this provides the ability for paid users to stream video lessons. The Maths Watch service also states on the subscription page of their website[14] that they provide "A bank of 1000s of interactive exam-style questions for students' independent work with instant feedback". This is similar to MyMaths where a user can submit their answers and get feedback on whether they were correct, and if they were incorrect have the correct answer displayed.

The Maths Watch service is used by students across the UK and is an incredibly valuable tool to aid their learning. Unfortunately, Maths Watch does not provide a mobile application on either of the major app stores (Apple app store, or the Google Play store). Only schools can purchase Maths Watch, so if a students' school is not a Maths Watch partner, then the student cannot access Maths Watch.

3.4 Independent Revision Tools

There are several other mathematics revision applications on both major app stores, examples of which can be seen below as Figures 2 and 3. These applications mostly focus on testing a user's knowledge using some form of revision questions. These questions are either limited to multiple choice answers, or they are questions which have the same values each time they appear. This leads to two problems: random guessing for the multiple choice questions, or answer memorisation for the questions which do not change their values. This leads to these applications not being endorsed by schools.

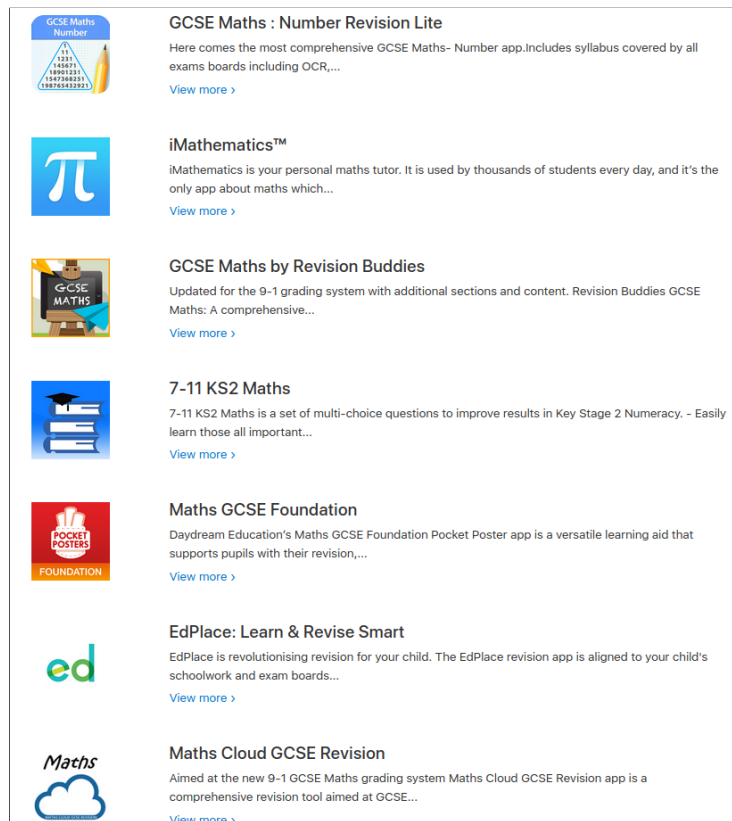


Figure 2: Revision applications on Apple's app store.

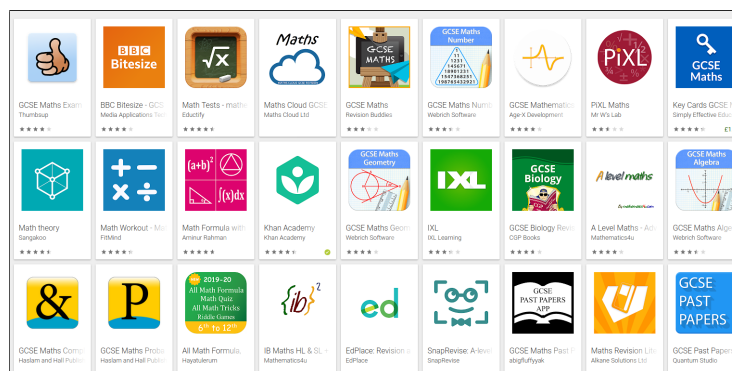


Figure 3: Revision applications on the Google Play store.

3.5 Application Development

Research into Android application development consisted of:

- Working through Head First Android Development[15]
- Reading through the Android Documentation[16]
- Creating small Android applications available on GitHub[17]

These tools were utilised in learning the basics of Android application development before any code was written for this project. The Android documentation helped with any programming issues within the code during the development of this application.

4 Methodology

This section will explain how the work necessary for this project was completed. It will describe the process by which the final application was developed at a high level, and section 6 will explain the lower level ideas with examples from the final application's source code.

The process by which work was completed within this project, often referred to as the methodology, changed throughout the duration of the project. The project specification stated that "This project will make use of the scrum software engineering methodology". The scrum methodology was not followed throughout the early stages of this project. Instead, a non-specific agile methodology was utilised, as each development cycle planned only the next requirement to be added as a feature. Each development cycle did not create user stories, or have a pre-defined length of between one and four weeks for development, which are both key features of the scrum methodology.

The non-specific agile methodology utilised for the early stages of this project consisted of three main steps. The first step was selecting the next requirement to be implemented into the program. Then, any decisions that needed to be made concerning implementation details would be made, and a small plan would also be created detailing these decisions. Finally, the code would be written for the requirement. The plan ensured that the requirement was implemented using the correct method. Once the requirement was completed, there would be no review, and so development would move onto the next requirement.

All aspects of development were continuously monitored and controlled throughout this project. Midway through the project it was noticed that the methodology was not being followed as set out in the project specification. This problem

is discussed in more detail later in section 8. This deviation from the initial project plan was disruptive to the progress of the project, and so the methodology followed throughout development was changed midway through the project. It was still a non-specific agile methodology, as no user stories were written, but each development (sprint) cycle consisted of one week, with specific time being dedicated to development each week. The other major change to the methodology was to include sprint reviews. These two changes meant that the methodology could be viewed as a variation on scrum, which used the project's requirements instead of user stories.

The final methodology change for this project was implemented during the last month of development. It aimed to help with writing this report and continuing development in parallel. Initially, all sections of the report were aimed to be written in parallel to decrease the amount of idle time in the writing process due to bottlenecks. This meant that multiple sections were worked on simultaneously when writing the report. This caused problems by increasing time spent context switching between writing different sections of the report, which can be viewed as wasted time. Once the context switching harmed development progress more than any bottlenecks, the decision was taken to change the methodology from scrum to the Kanban development methodology.

This change in methodology manifested itself in two key ways. The first was using Trello[18] as a way of visualising workflow, which is a key principal in the Kanban software development methodology. A Kanban board was set up for writing the report, and can be seen as figure 4.

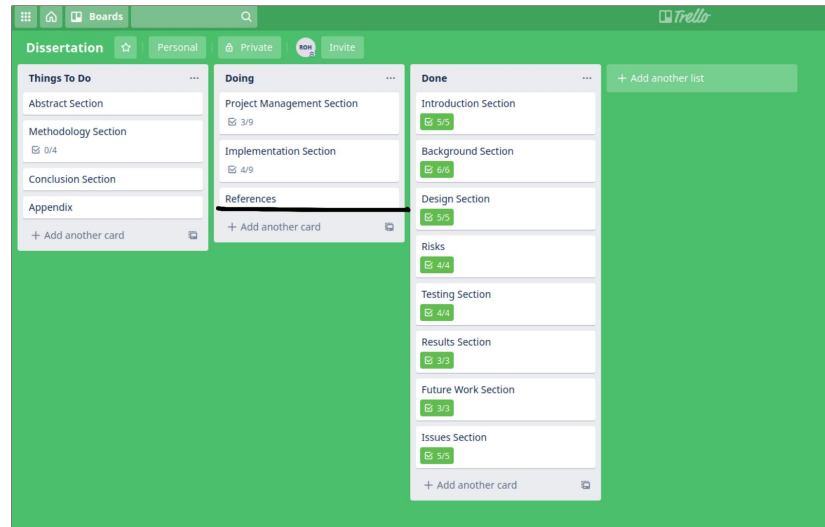


Figure 4: Trello board used as a to-do list for this report. The black line present is the work in progress limit for the ‘Doing’ section. This limits the amount of sections which can be worked on at any point in time to three.

This Trello board was created to effectively visualise the work that had to be completed for this report to be deemed finished. Work is pulled from the ‘Things To Do’ section into the ‘Doing’ section. Once the work is completed, it is moved into the ‘Done’ section. The second change to implement Kanban was the introduction of work in progress limits. Work in progress limits reduce the amount of time spent context switching by limiting the amount of tasks which can be worked on at any one point in time. There is one work in progress limit set in figure 4 and is denoted by a black line under the References’ card in the ‘Doing’ section. This work in progress limit decreased the amount of time spent context switching, and also narrowed the focus of each development session. Instead of having the choice between working on several sections of the report, the choice was between three sections of the report at any one time. This narrowed focus increased the productivity of the report writing, and also increased the quality of the text written.

5 Design

This section of the report will explain the visual design philosophy which was adhered to throughout development. It also aims to discuss the control and data flow of the final application.

This application had a target audience of school children between the ages of 11 and 16. This meant that any design would have to be simple and intuitive to prevent users from getting ‘lost’ within the application. The graphs for both the control and data flow for this application are acyclic and is shown by Figures 8 and 9 which appear in their relevant sub-sections. Users can only follow certain paths, and will only end up in the same place by pressing the inbuilt backwards button in the Android operating system, so they cannot get confused about the layout of the application. Also, the number of pages was kept to a minimum for the application to function for the same reason. This minimalist design aims to prevent confusion by keeping the application simple to understand and use.

5.1 Visual Design

The visual design of this application is minimalist. Each activity only has the necessary information displayed on it, and where possible, only one item is displayed per activity. Below are a few examples of this minimalist design. Figure 5 is a screenshot of the application’s start page taken from the physical Google Pixel 2 device.

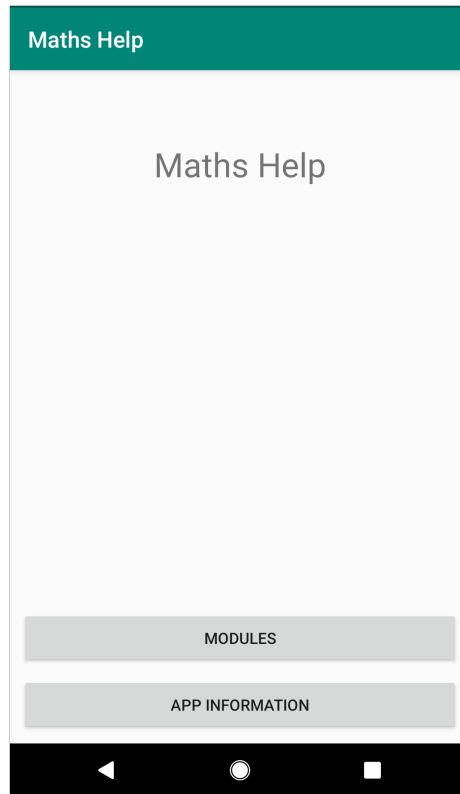


Figure 5: Start page of the application screenshot from the physical Google Pixel 2 testing device.

This screen is made up of the least amount of distinct elements that is possible. As it is the start screen of the application, and is the first thing a new user will see when using the application, it has to have the application name displayed, which is the “Maths Help” text at the top of the screen. The bottom of the screen is dominated by two buttons, one which starts the modules activity (the modules button), and one which starts the information activity (the app information button). This is the least amount of elements with which to follow the control and data flow graphs displayed below.

An Android activity is the application equivalent to a webpage on a website. Activities are viewed as one screen in an application[19] and are intended to complete one specific task. If another task needs completing then the application should call another activity. Activities can contain fragments, that should be thought of as “a modular section of an activity ... (sort of like a ‘sub activity’ that you can reuse in different activities)”[20].

An example of an activity displaying a list fragment is the modules activity screen. This is displayed below as Figure 6, also taken from the physical Google Pixel 2 testing device.

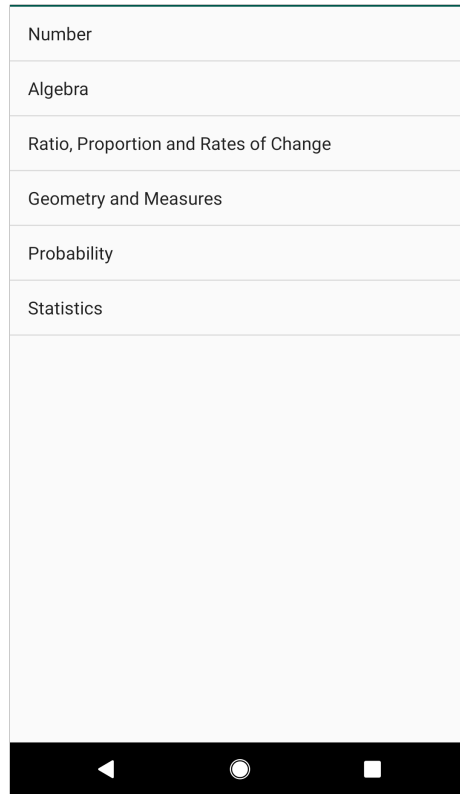


Figure 6: The modules activity page.

This screen is empty except for the selections that a user can choose. This basic design ensures that only what is essential for the application to function is included on each screen. All of the screens which contain a list could have included an explanation at the top of the screen informing the user to select one of the options. This was not included, as all nine users involved in black box testing intuitively understood how the lists functioned. It was deemed that an explanation would clutter the screen, users could see it as condescending, and it could negatively impact the user experience.

The question display activity also obeys the minimalist design of this application, only containing what is essential for the question to be answered. It can be seen as figure 7, which is a photo of the screen taken from the physical Google Pixel 2 testing device.

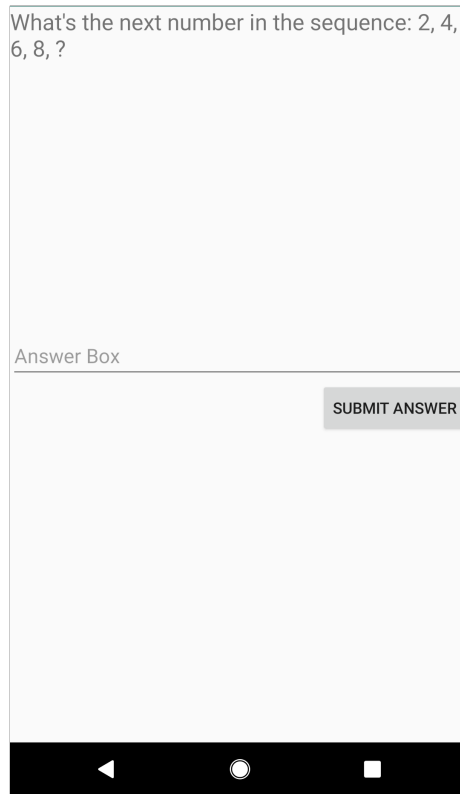


Figure 7: The question display page.

This activity consists of the question text, an answer box for the user to input text, and a submit button. For questions in which an image is required, the image appears between the question text and the answer box. A minimalist design is followed throughout the application, to include only what is necessary on each activity so as to not over-complicate the application.

5.2 Control Flow

Figure 8 is a graph for the control flow of the application. Each edge is bidirectional; the user can move forward by interacting with the application, and backwards by hitting the Android operating system back button. The user interacts with the application in one of two ways, either by pressing a button on their screen, or entering text into a text field.

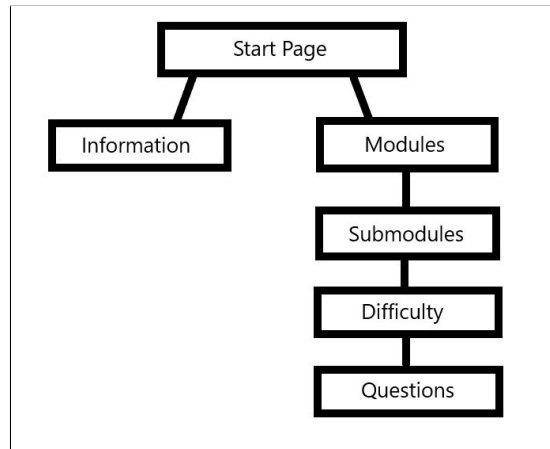


Figure 8: The control flow of the application.

The only choice in direction in this application is at the start page, where a user could select between the information page for the application and the modules list. Each node in this graph represents an Android activity. The modules, submodules, and difficulty nodes in the control flow graph are activities which only contain list fragments. The list fragment class is a subclass of the fragment class but is adapted to display a list of items and implement a listener, which will act when the user clicks on an element in the list. In the example of the modules node in the control flow graph, the list fragment displays all six modules which are examined in Edexcel GCSE Mathematics. When the user clicks a specific module, the modules activity collects the information of the module selected from the list fragment, and starts the submodules activity with the information of the selected module passed to it as an input. The submodules activity then displays a list fragment containing the submodules for the module selected by the user.

The control flow for this application is minimalist. It only includes pages which are absolutely necessary. The graph for the control flow of the application is intentionally acyclic. Without any loops, there are only two ways in which a user can end up at a certain page; they can go forward from the root node or they can go backwards from a child node. This is simple, and should not confuse users, as they can only follow linear paths within the application. If a user ends up at a page that they have encountered before, then they must have pressed the Android back button.

5.3 Data Flow

Figure 9 is a graph of the flow of data throughout the application. When an activity starts another activity, it can pass information to it. This enables a single directional information flow throughout the application. That is why the diagram displayed in Figure 9 is a directed graph. The direction of the data is denoted by an arrow, the side of the edge without the arrow is the activity sending the data, and the side of the edge with an arrow is the activity receiving the data.

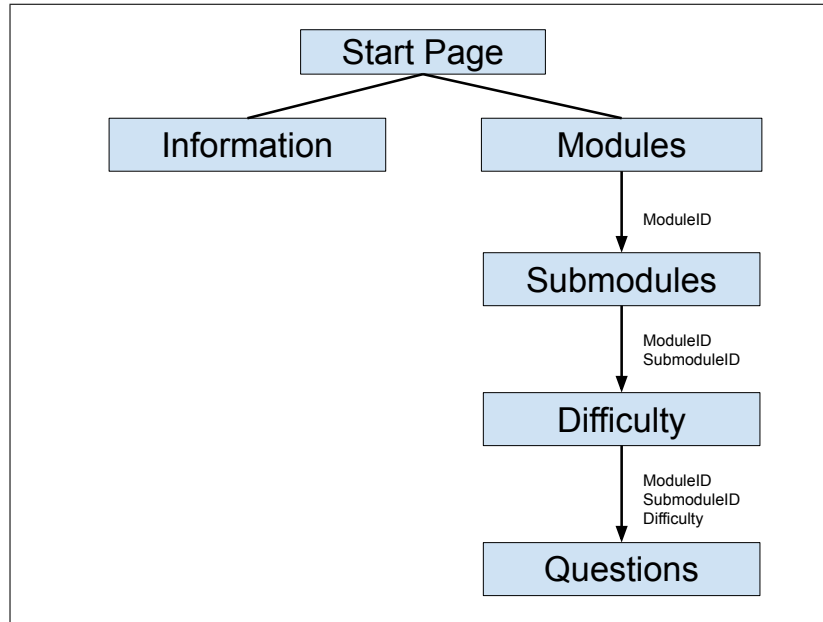


Figure 9: Diagram representing how data flows through the application. The information next to the arrows is the data passed from one activity to the next.

The data flow in this graph follows the minimalist design of this application. Data moves along a path in only one direction. A user can go back to a previous activity by using the inbuilt back button in Android. By doing this, the information selected further up the graph (closer to the root node) is preserved, but the information gathered along the edge which the application is now traversing is lost. This allows a user to utilise the back button to change the difficulty of the questions that they are answering whilst keeping the module and submodule information preserved. This enables simple use, where a user does not need to reselect all information to change one specific piece of data.

Information is passed along the modules path until it gets to the questions

activity. The questions activity then uses all of the information to select the questions that it will display.

6 Implementation

This section will discuss how the project was created, providing examples from the source code where appropriate. It will explain in depth some key features of Android development and how they have been applied to this project, and will also justify key decisions made throughout the development process.

6.1 Fragments

Android activities have been explained earlier in section 5. A fragment is very similar to an activity, and has a very similar life cycle. Fragments can be viewed as a modular section of an activity. One activity can contain multiple fragments, and the same fragment can be used in many activities. A fragment is designed to be used when a developer wishes to run the same code in multiple activities, and is a way of preventing developers from rewriting code. As fragments can be added to activities in the same way a text box is, by adding a tag in the eXtensible Markup Language (XML) data of the activity. This makes reusing code through fragments quick and simple.

Visual elements of the application are stored as XML data and then rendered on the device. Each activity has its own XML data to tell the device how it should be rendered. This allows each activity to have its own unique layout created to achieve the task assigned to that activity. Fragments also have their own XML data and code, which enables them to have a unique layout and be interactive, to suit their specific goals. Fragments, however, have to be attached to a parent activity. This allows fragments to act similarly to activities.

This project uses the `listFragment[36]` class, as opposed to the `listActivity[37]` class, to control the modules, submodules, and difficulty lists. There are examples, however, of where activities are used instead of fragments. This is due to the re-usability of the code. For example, the `displayQuestion` activity could be modified to utilise several fragments instead of just being an activity. This, however, was decided against. For consistency, the questions are displayed identically on all devices to ensure a good user experience. Since the code is only going to be used once, it has been written as an activity.

6.2 Questions

Three key decisions were made with regard to how questions were implemented in the application, with varying consequences.

The first key decision that was made regarding the implementation of questions into the application involved selecting the questions to be included. It was essential that these questions were of GCSE standard so that students could effectively revise for their exams. The decision was made to use questions from past GCSE exam papers in the application. Examination of the new Edexcel specification began in the summer of 2017 and was conducted twice an academic year, in November and June. This project was conducted between the fall of 2019 and the summer of 2020. This allowed for six different sets of past papers for the current Edexcel GCSE Mathematics specification at the beginning of this project. Each of these sets included three foundation tier papers and three higher tier papers. Each question paper included between 20 and 30 questions, which provided a lower bound of 720 past paper questions provided by Edexcel. This allowed for a large pool of questions to be added into the application, spread across all of the modules and submodules tested within Edexcel GCSE Mathematics.

The process of adding new questions into the application began with labelling each question from a past paper with the module and submodule that it belonged to. This information, combined with the difficulty of the paper that it was included in, meant that the user journey through the application could be mapped. Mapping the user journey enabled the amount of questions similar to the one in the exam paper, which were already included within the application to be viewed. This gave the ability to balance the questions, ensuring that each submodule had the same amount, which meant that each topic could be revised thoroughly.

The second key decision that was made regarding the implementation of questions into the application was their storage medium. There were three key methods of storing questions. They were: hard-coding the questions into the application, reading the questions from a JSON file on the user's device, or selecting specific questions from a SQLite database as SQLite is included by default in Android. For this project, the questions were hard-coded into the application. The two other implementation methods are explained in detail in section 11. The decision to hard-code questions was made due to the methodology used to develop this software. As discussed in section 4, when developing the application an MVP was created, with that MVP then being built on, to add more features. Because of this, when the question feature was initially being implemented into the application, only one question was included. Once that question was successfully tested, others were added. The fastest way to implement a single question into the application was to hard-code it into a class. Once the initial question was implemented, further questions were added in batches

of four.

The third key decision regarding question implementation was how to hard-code them into the application. A ‘question’ class was created, which included all of the information that a question would need within the application. A ‘question’ object would be created every time a question was required for the application, and would include all of the information about the question so that it could be displayed and answered. A screenshot of the question class can be seen below as Figure 10.

```
4  * Question class transmits information between activities about questions which have been
5  * generated for students.
6  * It stores the module and submodule information about a question, also difficulty settings
7  * once it has been included, and the path to any images required for the question to work
8  * properly.
9  * It also stores values about the questions and elements of questions to be displayed.
10 * */
11
12 import java.util.ArrayList;
13
14 public class question {
15
16     private int moduleID;
17     private int submoduleID;
18     private ArrayList<String> questionText;
19     private String pictureName;
20     private int difficulty;
21     private int answer;
22
23     public question() {
24         this.moduleID = 0;
25         this.submoduleID = 0;
26         this.questionText = new ArrayList<String>();
27         this.pictureName = "";
28         this.difficulty = 0;
29         this.answer = 0;
30     }
31
32     @ public question(int moduleID, int submoduleID, ArrayList<String> questionText, String pictureName, int answer) {
33         this.moduleID = moduleID;
34         this.submoduleID = submoduleID;
35         this.questionText = questionText;
36         this.pictureName = pictureName;
37         this.difficulty = 0;
38         this.answer = answer;
39     }
40
41     /* Getters for each variable in this class */
42     public int getModuleID() { return this.moduleID; }
43     public int getSubmoduleID() { return this.submoduleID; }
44     public String getQuestionText() {
45         String stringQuestionText = "";
46         for (int i = 0; i < this.questionText.size(); i++) {
47             stringQuestionText = stringQuestionText + this.questionText.get(i);
48         }
49         return stringQuestionText;
50     }
51     public String getPictureName() { return this.pictureName; }
52     public int getDifficulty() { return this.difficulty; }
53     public int getAnswer() { return this.answer; }
54
55     /* No setters as you have no need to change these values once created */
56
57 }
```

Figure 10: Source code for the ‘question’ class.

This class stores both the moduleID and submoduleID for a question. It also includes an array list, which stores data in a Java string class. This array list contains the question text data, which is then showed to the user by the displayQuestion activity. The getQuestionText() method iterates through the array list, accessing each individual element and concatenating it into a string to return to the displayQuestion activity. This data is stored in an array list instead of as a string for convenience. It is convenient because it enables the question text to be split on the numerical values which are used within the questions. The final array list will be in the form “text, (number, text)*” where

this form is viewed as a regular expression and a ‘,’ indicates that a new element is to be created within the array list. This allows numerical values to be put into the question text, whilst also providing access to these values if needed.

The string, `pictureName`, included in the ‘question’ class in Figure 10 is the file name of the image to be displayed along with the question text. Currently, only one image can be displayed. This could be extended to multiple images by storing the file paths of each image in an array list, which would then be iterated through. If this variable is set to the string “None” then the image view is hidden from the screen, and no image is displayed. The question difficulty is also stored in this class as a numerical value, being either zero or one. A zero stored in this field indicates that the question is of foundation tier difficulty, whereas a one value being stored indicates that the question is from a higher tier paper.

The question’s answer is also stored in the ‘question’ class. All question answers have been rounded to integer precision. Each question’s text includes information stating that the user’s answer should be rounded to the closest whole number value. Users are prevented from entering non-integer values into the text box through settings applied to the text box’s XML. Currently, no questions which require text based answers are included in the application, although they could be implemented. This type of question would be included in multiple choice format, with radio buttons displaying prospective answers, and users would select the answer that they thought was correct, and then click a submit button. The application would then send the answer as a value between zero and n , where n is the number of prospective answers being displayed, to be marked. The background of the correct answer could then be displayed as green to indicate that it is the correct answer, and the background of the incorrect answers can be changed to red. The user’s answer can be outlined to prevent the user forgetting the answer that they selected.

Where a numerical value is used in a question, it has been produced in one of two ways. It has either been produced using the Java random method, or it is calculated from other numerical values which have been randomly produced for the specific question. This ensures that numerical values for questions are numbers which have not been seen before by a user, or are not repeated often. This aims to prevent answer memorisation. Answer memorisation is when a user enters an incorrect answer for specific question values, and then remembers the correct answer which gets displayed. Next time a student is presented the same question with the same values, they input the correct answer which they remembered from the last time they attempted the question. This makes it look like the student has learned the method for answering the question, when in reality they have just memorised the answer for those specific values. This is why when a value is used in a question it is generated using the Java random method applied over a specific range of values. Any other values necessary for a question are calculated using the appropriate equations. For example, a SUVAT

question using the equation $v^2 = u^2 + 2as$ can have the values for u , a , and s generated using the random number generator. The value for v , however, cannot be randomly generated as it depends on the other variables in the equation. In this case, the value for v is calculated using the equation. If the value for v is not an integer, then the values for u , a , and s are set to other random numbers. This continues until an appropriate value is found for v . This allows the student to see new values often whilst attempting a specific question, but also ensures the answers are appropriate for the difficulty level of the question.

For a specific submodule, several questions are generated. When a user selects a specific submodule, one of the questions associated with that submodule is produced to be answered. These questions currently have an equal probability of being selected, but the frequency of each question appearing could be adjusted.

Figure 11 is an extract from the source code of the application. It is from the testQuestionGeneration.java file. This section of the source code generates the test questions and answers, and returns a 'question' object with all of the relevant information in it.

```

20 public static question calculationFoundationQuestion() {
21     float randomNumber = (float) Math.random();
22     if (randomNumber < 0.3333) { return calculationFoundationQuestionOne(); }
23     else if (randomNumber < 0.6666) { return calculationFoundationQuestionTwo(); }
24     else { return calculationFoundationQuestionThree(); }
25 }
26
27 // Structure and Calculation (foundation)
28 @
29 private static question calculationFoundationQuestionOne() {
30     ArrayList<String> questionText = new ArrayList<String>();
31     questionText.add("Which one of these numbers is the smallest: ");
32     int currentNumber;
33     int smallestNumber = 501;
34     for (int i = 0; i < 5; i++) {
35         currentNumber = (int) Math.round(Math.random() * 500);
36         if (currentNumber < smallestNumber) { smallestNumber = currentNumber; }
37         questionText.add(Integer.toString(currentNumber) + " ");
38     }
39     questionText.add(".");
40     return new question( moduleID: 0, submoduleID: 0, questionText, pictureName: "None", smallestNumber);
41 }
42
43 // Structure and Calculation (foundation)
44 @
45 private static question calculationFoundationQuestionTwo() {
46     ArrayList<String> questionText = new ArrayList<String>();
47     int randomNumberOne = (int) (Math.random() * 100);
48     int randomNumberTwo = (int) (Math.random() * 100);
49     questionText.add("What is ");
50     questionText.add(Integer.toString(randomNumberOne) + " + " + Integer.toString(randomNumberTwo) + "?");
51     int answer = randomNumberOne + randomNumberTwo;
52     return new question( moduleID: 0, submoduleID: 0, questionText, pictureName: "None", answer);
53 }
54
55 // Structure and Calculation (foundation)
56 @
57 private static question calculationFoundationQuestionThree() {
58     ArrayList<String> questionText = new ArrayList<String>();
59     int pow = (int) Math.round(Math.random() * 5);
60     questionText.add("What is 2^");
61     switch (pow) {
62         case 0: questionText.add("\u2070");
63         break;
64         case 1: questionText.add("\u00B9");
65         break;
66         case 2: questionText.add("\u00B2");
67         break;
68         case 3: questionText.add("\u00B3");
69         break;
70         case 4: questionText.add("\u2074");
71         break;
72         case 5: questionText.add("\u2075");
73         break;
74         default: questionText.add("\u2070");
75     }
76 }

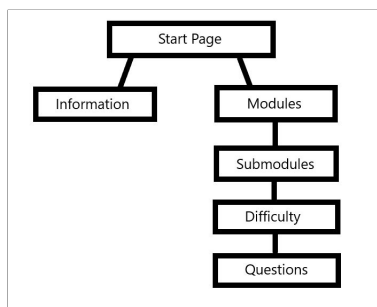
```

Figure 11: Source code for the random question generation.

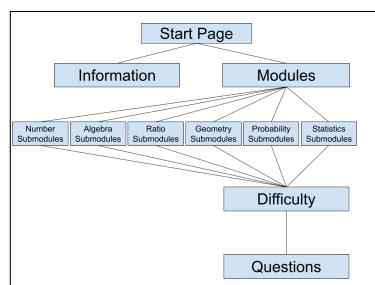
The `testQuestionGeneration` class, which is displayed above as Figure 11, is a factory class which returns random questions given the module and submodule details. The `calculationFoundationQuestion()` method is called when the user selects the number module, followed by the structures and calculation submodule. It returns one of three questions written for the structures and calculation submodule, selected using Java's random number generator. The random question is generated by the appropriate private method in the factory class. For example, `calculationFoundationQuestionOne()` is a private method which generates a 'question' object with the appropriate information. This method generates five random numbers between zero and five-hundred, and the user is asked to input the smallest number of the five. This 'question' object is then returned to the `calculationFoundationQuestion()` method, which then returns the 'question' object to the `displayQuestion` activity, allowing the question to be shown to the user. Each submodule has its own equivalent of the `calculationFoundationQuestion()` method in this factory class.

6.3 Control and Data Flow

As is stated earlier in this report, a list fragment is used to display both the modules and submodules lists, as well as the difficulty list. It was decided that the `moduleID` selected from the 'modules' list fragment was to be passed to one 'submodules' list fragment that would collect the list to display based on the `moduleID` that it received. Instead, six list fragments could have been created, one for each module, and when a `moduleID` was selected from the 'modules' activity, it could call one of these six list fragments, depending on the `moduleID` that the user had selected. This would have greatly increased the complexity of the application from a development perspective, although it would not have changed the user experience. Figure 12 below shows the control flow diagram for this application, as well as the complex control flow diagram which splits the submodules activity into six individual submodule activities, each representing a separate module.



(a) A diagram of the control flow of the final application.



(b) A diagram of the control flow for the hypothetical complex application.

Figure 12: A comparison between the two choices in control flow for the application.

The second image, Figure 12b, has almost twice as many activities as Figure 12a, and as such would have increased the time spent developing the application, without providing benefit to users. This is why the decision was made to have only one activity, which would fetch the appropriate list of submodules, depending on the module selected by the user. This would also make adding a module or submodule simple, as it would just require adding the name of the new module or submodule to the appropriate array. If the other method of fetching a list of submodules was implemented, as Figure 12b displays, then adding another module or submodule would involve changing the control flow of the application by adding extra activities. This would have increased the complexity of the control flow of the application, without adding benefit to the user, and goes directly against the minimalist design principle used for this application.

7 Testing

This section of the report will describe the general testing methodology conducted throughout this project, and will explain why the testing methodology was changed. It will then explain the two kinds of user testing conducted for this application.

The project specification[34] stated that this project would implement test-driven development. The progress report[35] described that test-driven development would not be used as the testing methodology for this project, and that user testing would be used throughout the project in its place.

The decision to change from test-driven development to user testing was

made because this project creates a tangible final product in the form of an Android application. An Android application is judged by users for unquantifiable reasons, such as the “feel”, or “ease of use” of the application. This can only be tested by a person using the application, and since this determines how satisfied a user is with the product, testing for the unquantifiable qualities of the application was prioritised over writing testing code. Writing unit tests for an Android application was deemed to be an inefficient use of time, as this time could be spent improving the user experience of the application.

All application testing was conducted on a physical Google Pixel 2 smartphone, as well as at least one emulated virtual device. The emulation software used is built into the Android Studio IDE. It is called Android Virtual Device and “provides almost all of the capabilities of a real Android device”[21]. The emulator provides an environment to test the application on a variety of smartphones and tablets with different capabilities, to ensure that the application works on a wide range of systems running the Android OS. The application was tested on three virtual devices:

- HTC Nexus One (Smartphone)
- HTC Nexus 10 (Tablet)
- Google Pixel 3 (Smartphone)

These testing devices give a good variety in the Android operating system, between both tablet and smartphone, old and new. These devices were chosen because if the application works well on all four devices (the physical Google Pixel 2 smartphone as well as the three virtual devices) then the application was likely to work on most hardware running the Android OS.

Two types of user testing were conducted throughout this project, white box testing and black box testing.

7.1 White Box Testing

White box testing is when an individual who has access to the source code of the application, in this case the application’s developer, is given access to it and asked to complete a specific task. It was conducted after each sprint cycle. White box testing consisted of the developer downloading the application onto the devices mentioned above, and then testing that the application compiled and ran without errors. Other small tests were also conducted when integrating new features, called feature tests. An example of a feature test would consist of the developer creating expected states of the application given specific inputs, and then following the input sequence on all four testing devices to ensure that

the application finished in the expected state. Visual tests were also conducted, where an activity's layout was drawn on paper, and then loaded on each of the testing devices. The layout on the testing devices was compared with the layout drawn on paper, and if the two layouts were identical then the activity passed. This testing method was used to find visual errors in the application, and was conducted when changes were made to the XML of an activity.

7.2 Black Box Testing

Black box testing is when an individual who has no access to the source code is given the application itself, and asked to complete a specific task. It was conducted using individuals outside of the development team. This testing was primarily conducted on the physical Google Pixel 2 testing device. Verbal consent was requested and obtained from each user, in which they agreed to participate in the testing procedure and have their use of the application observed. If it was the first time the user had seen the application, then they were asked to use it as if they had just downloaded it from their normal app store. Whereas, if a user was familiar with the application, then they were asked to complete a task which utilised a new feature. The users could also be provided with two images, one displaying the most up-to-date visual style of the application, and one image of an older style; they would then be asked to compare the two aesthetics, without being made aware of which design is the most recent. Where there was a change in a feature that the user had used before, they were not notified of the change, and how they dealt with the change was observed.

Black box testing was conducted after every major feature was added to the application, with the results being used to improve the feature. Users of different mathematical abilities were asked to test the application, from colleagues who use mathematics every day in their fields of study, to those who last encountered it when they took their GCSEs.

8 Risks

This section will discuss some of the risks associated with this project. The risks will be outlined, which will explain the causes of the risks and the effect that they would have had on the project. Once the risk has been outlined, the discussion will centre around risk handling. All risks were handled in one of four ways. The risks were either: avoided, transferred, mitigated, or accepted. Where a risk was avoided, the cause of the risk was eliminated from the project. A transferred risk was one where the risk was handled by a third party; an example from outside of this project would be the case of authentication. If single sign on were to be used, then the risk of password leakage is transferred

to the third party responsible for the authentication. If a risk was mitigated then the effects of the risks were reduced by either reducing the probability of the risk occurring, or by reducing the impact that the risk could have on the project. An accepted risk was one where no action was taken before the risk occurred, and when it did occur, it was promptly dealt with without a large impact on the project. Finally, the outcome of the risk will be discussed.

8.1 Scope Creep

Scope creep occurs when the scope of the project slowly increases over time. This occurs for one of two reasons: either a customer continuously changes their description of the final product, or the project manager adds unnecessary features to the project. As there is no customer directly involved in the development of this project, the only way in which scope creep can occur is if the project manager adds to the initial feature set.

Scope creep can be incredibly damaging to a project in one of two ways depending on whether or not the project has a predefined deadline for completion. If the project does not have a predefined deadline, and features continue to be added throughout development, then the project may never be completed. If the project is evaluated for resources at the beginning and then scope creep occurs, there may not be enough resources to complete the project. On the other hand, if the project does have a predefined deadline, adding new features means less time is spent on key features for the project. When a resource is constrained, scope creep can waste the resource and leave less of that resource for key features.

Scope creep was handled in this project by outlining the key features necessary for this application at the beginning. These key features are the requirements written in the project specification. Also, defining project success in the project specification helped to handle scope creep. The requirements outlined the key features to be developed throughout this project, and the definition of success provided a baseline for the necessary features of the application. The requirements were written using the MoSCoW method, to ensure that the key requirements had a higher priority for development, and so were implemented into the application first. This mitigated scope creep by lowering the probability that it would occur. The requirements with a higher priority were completed first and created the MVP for the project. Once this MVP was created, the 'should' requirements started to be implemented into the application. This prioritisation meant that any new features that the project manager wanted to add had to become 'could' requirements, and so, would have been added into the application after all of the key features.

This project did not suffer from scope creep. The requirements were a key reason for this, as they dictated what was completed in a specific sprint cycle.

As work was only conducted on the requirements with the highest priority, there was no ability for features which were not included in the initial requirements to be worked on, and so no time was wasted.

8.2 Personal Issues

Due to the development team consisting of only one person, who was irreplaceable to the project, any personal issues on behalf of that individual could have affected the project. Personal issues could include a high workload, family issues, or a global pandemic. Any issues which would prevent the individual from working on the project for reasons outside of the development cycle were classed as personal issues.

Personal issues are issues which cannot be prevented, as they are unpredictable and can have varying effects on a project. This meant that it had to be accepted as a risk. As personal issues were a group of issues which were unrelated to the project, but still affected it, it was possible to mitigate or transfer individual issues. For example, if the only member of the development team were to fall ill and be unable to work for two weeks, then it would have been mitigated with a deadline extension. Although personal issues as a group had to be accepted, individual issues were handled appropriately.

Accepting the personal issues with this project did not have a major effect on the outcome. Personal issues did impact the pace of development in the first academic term of this project, as the development team had other priorities to attend to. This, alongside poor project management, caused the project to fall behind schedule. This time was made up due to good project management throughout the second term of this project, and so the project was completed on time.

8.3 Poor Project Management

Poor project management is caused by a project manager who is not adequately fulfilling their role. It can appear in many forms, one of which is scope creep. It could also manifest itself as poor productivity, or disagreements within a development team. It can severely affect a project. Poor management can cause the project to fail, whereas good management can save a project.

The three main causes of poor project management are:

- an inexperienced project manager
- poor communication within a development team

- lack of sufficient planning

Each of these issues alone can slow the development process and can cause issues affecting the final product. For effective project management, these three main causes of poor project management have to be handled appropriately.

Throughout the first term and the Christmas break of this academic year, project management was not handled appropriately. This, alongside personal issues discussed earlier, caused the project to be delayed by four weeks at the beginning of the second academic term. Poor project management was initially an accepted risk, although risks were both monitored and controlled on an ongoing basis throughout the project, to ensure that the project was on track to be successfully completed. This monitoring highlighted that if the poor project management techniques were allowed to continue, then the project would be a failure. Poor project management was then changed from being an accepted risk to being mitigated.

A plan was implemented at the beginning of the second academic term to fix the poor project management from the first half of development. This plan tackled two of the three main causes of project management stated above. Poor communication within a development team was not a problem for this project since the development team consisted of one person. The problem of the inexperienced project manager was fixed by undergoing management training. This training enabled the project manager to effectively manage development time. This saved time throughout the project, enabling the completion of more requirements, and contributing to the success of the project.

On top of the project manager gaining a greater understanding of how to effectively manage a software development project, there was an increase in planning throughout the second term of the academic year. The change which had the biggest impact on the project was creating a weekly development schedule. The first academic term consisted of working on the project whenever there was spare time. This unpredictability meant that different amounts of hours were put into the project each week. This inconsistency caused the project to fall behind, as less work was completed in the first academic term than was predicted by the initial project schedule. Giving the project specific time each week for development increased the work rate. All of Wednesday was dedicated to project work throughout the second academic term, and this enabled productivity to be greatly increased.

Eliminating the two main causes of poor project management enabled the project to be completed on time, and to a good standard. The risk of poor project management was successfully mitigated.

9 Project Management

This section of the report will begin by discussing the various details surrounding the two different schedules produced for stakeholders throughout this project. This will be followed by a discussion of the various tools used to aid this project's development process. Finally, this section will run through various miscellaneous project management procedures which aided in the development of this project, that are not visible from viewing this project's source code.

This project was a large software engineering project, especially for a team size of one. This meant that effective project management was required to ensure the success of the project.

9.1 Schedule

Two schedules were developed for stakeholders throughout this project. They were created for the two other major documents produced throughout this project, and are displayed in full within those documents. These two documents were the project specification[34], which was produced at the beginning of the project, and the progress report[35], produced at the end of the first academic term.

This subsection will discuss both of the schedules in detail, and will then evaluate the final schedule of the project.

9.1.1 Project Specification Schedule

A schedule was created at the beginning of this report. This schedule aimed to complete all requirements outlined in the project specification[34]. The schedule was created using a technique which is common in industry, known as 'planning poker'[33], which involves a set of numbered cards. The relative difficulty of each requirement was assessed compared to all of the other requirements. The difficulty was quantified using the numbered cards; the cards would follow a logarithmic pattern, doubling with difficulty each time. The lowest valued card was $1/4$ and the highest value was 16. Any requirement with a relative difficulty value greater than 16 was then broken down into sub-requirements, and these sub-requirements would then be reviewed in the same manner.

The relative difficulties in requirements gained through playing planning poker were used to create a rough schedule for the project. The schedule created for the project specification can be seen as a Gantt chart in Figure 13.

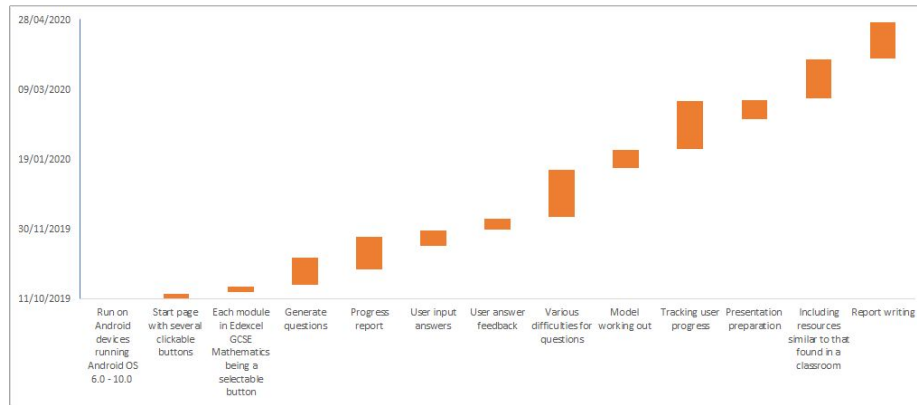


Figure 13: Gantt chart of the schedule produced for the project specification.

This schedule was an optimistic schedule. It assumed that all of the requirements would be completed by the end of the project, whereas the project specification's definition of success did not need all requirements to be completed.

9.1.2 Progress Report Schedule

The progress report submitted on 24th November 2019 updated the project specification's schedule. Development was slower than intended throughout the first term of this academic year (October 2019 - December 2019) due to personal issues, as well as poor project management, which has been discussed in detail in section 8. This updated schedule was created to account for this slow development. The new schedule, referred to from now on as the Christmas schedule, also made the assumption that every requirement would be completed before the final deadline for this project. To mitigate the slow development conducted through term one of this academic year, the Christmas schedule increased the workload of the project throughout the Christmas period (8th December 2019 - 8th January 2020) and throughout the second academic term of this project.

A Gantt chart created for this updated schedule can be viewed below as Figure 14.

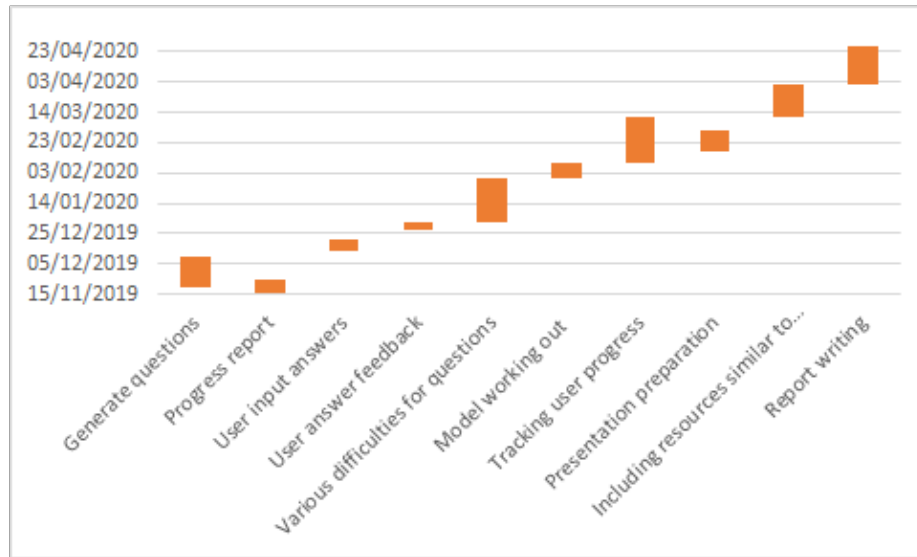


Figure 14: Gantt chart of the updated schedule produced for the project specification.

Everything before the 24th November 2019 in this Gantt chart is correct, as it was completed before this schedule was created. Anything after this date is an estimate, and comparisons for future work should be compared to this.

9.1.3 Schedule Evaluation

The image displayed below as Figure 15 shows the final schedule at the end of the project.

Objective	Start Date	End Date	Duration
Run on android between versions 6.0 and 10.0	11/10/2019	11/10/2019	0
Start page with several clickable buttons	11/10/2019	15/10/2019	4
Each module in Edexcel GCSE Mathematics being a selectable button	16/10/2019	31/10/2019	15
Generate Questions	01/11/2019	23/01/2020	83
Progress Report	16/11/2019	24/11/2019	8
User Input Answers	24/01/2020	25/01/2020	1
User Answer Feedback	26/01/2020	29/01/2020	3
Various Testing Difficulties	30/01/2020	02/03/2020	32
Model Working Out	04/03/2020	N/A	#VALUE!
Tracking User Progress	15/03/2020	N/A	#VALUE!
Presentation Preparation	17/02/2020	03/03/2020	15
Including Classroom Resources	N/A	N/A	#VALUE!
Report Writing	30/03/2020	11/05/2020	42

Figure 15: The final schedule created at the end of the project. Green requirements have been completed by the end of the project. Yellow requirements have been started by the end of the project, but not completed. Finally, red requirements were not started before the project deadline.

This figure displays information about each requirement, as well as some key deliverables produced throughout the project. It shows the start and end dates of the requirements, as well as the duration of each requirement in days. The requirements are colour coded. Requirements with a green background have been fully completed at the end of the project. A yellow background for the requirement represents a started but uncompleted requirement, and a red requirement was not started during the project.

This final schedule was behind both the project specification schedule and the progress report schedule. The reasons for this are discussed in the risks section of this report. All of the schedules were optimistic, and assumed that all of the requirements would be met. This was different to how the definition of success for this project was defined. Even though the project was behind the optimistic schedules, work was still completed at a satisfactory rate, resulting in the project being completed on time. Although, if time was not lost during the first half of development, the optimistic schedules could have been met.

9.2 Tools

This project utilized several tools to aid the development of this Android application. Two of the main tools used to add value to this project are Android Studio and Git, as outlined below.

9.2.1 Android Studio

Android Studio[22] was used as an Integrated Development Environment (IDE) throughout development, as it comes with many useful features. These features are specifically designed to help with Android application development. Examples of these features are:

- a rendering engine which converts XML into a virtual application display
- an android device emulator

The rendering engine displays the layout of any XML written in Android Studio. It enables the XML to be edited directly, with any changes made being viewable in the virtual layout in real time. It also provides the ability to edit directly into the rendered layout with a drag and drop feature. This enables quick prototyping for new pages through the use of the drag and drop feature, whilst also allowing for a much larger degree of control through the ability to directly edit XML.

The Android device emulator can simulate any Android device for which it contains files. Files for new devices can be downloaded through Android Studio, and new devices are being added regularly. This feature enabled quick and efficient testing throughout development. When a new feature was integrated into the application, it could be tested on a wide range of devices, ensuring that the application behaved consistently on all Android smartphones and tablets.

The Android Studio IDE greatly increased the efficiency of development, due to the features stated above. Also, Android Studio being built upon the IntelliJ platform meant that code editing was smooth and intuitive.

9.2.2 Git

Git was used as the version control software for this project. This enabled access to the project from any device, anywhere, anytime. It also enabled the ability to revert back to an earlier version of the project if something went wrong. This meant that development could take place without fear of failure, since the project could always be reverted to the last working state, even if a change was made that broke the project.

The repository[23] for this project is public, which enables anyone to see the code that would run on their machine. Git automatically logs the time whenever a change is made to the repository. This enabled precise tracking of requirement completion, and aided project management. Git commit messages for this project were deliberately descriptive, and explained changes made to

the codebase, so they could be used as logs of when project requirements were completed. The Git commit messages for this project can be viewed, and give a detailed timeline of when requirements were completed and when features were added. This was useful because it provided updates of any changes made to the codebase, and is available to anyone who follows the repository as soon as the update is pushed. Git was used to create a timeline of requirement completion for the schedule included in this final report.

9.3 Miscellaneous Project Management

Weekly meetings were held with the project supervisor to keep key stakeholders up to date with development. Alongside weekly meetings, the project supervisor and relevant stakeholders were given access to the Git repository for this project early in development.

A document in the Git repository called notes.txt is a file which contains notes created throughout development, such as what was intended to be completed at the beginning of each development session, as well as what was actually completed by the end of the session. In early development sessions, a substantial amount of time was spent remembering what content had been added in the previous development session, the current state of the code, and what needed to be completed during this development session. Notes.txt cut down the time spent looking through previous Git commits at the beginning of each development session; overall, this file caused development to be more efficient.

10 Results and Discussion

This section of the report will evaluate the final application produced for this project and compare it to the requirements and definition of success set out at the beginning of this project. It will also contain the analysis of the project by the author.

10.1 Requirements

The definition of success for this project was defined as follows. To be deemed successful, the final product produced by this project will have completed:

- all of the ‘must’ requirements defined above
- at least one of the ‘should’ requirements defined above

- any value of ‘could’ requirements defined above.

The final application satisfies all of the ‘must’ requirements. This means that the final application can: display questions for users to answer, have an input field accepting numerical values, when an answer is submitted compare it to the pre-calculated answer, and display whether the two answers match. If the two answers do not match, then the pre-calculated answer is displayed for the user to see.

One of the three ‘should’ requirements for this project was completed outright. Requirement 7, which was defined as “having a variety of testing difficulties, ranging from easy to hard for each module” was changed between the progress report and the project presentation. It was changed to “having two separate testing difficulties of foundation and higher for each module”[34]. This change was made as the Edexcel GCSE Mathematics specification is split into two difficulty tiers of foundation and higher. The foundation tier of exams range from grades 1-5 and the higher tier exams range from grades 4-9. The higher tier covers all of the content in the foundation tier, as well as some additional content which is more complicated, to make the higher tier exams more difficult. Given that the Edexcel tier system already existed, it would only confuse students if ‘easy’, ‘medium’, and ‘hard’ testing difficulties were used instead. Because of this, the decision to switch to Edexcel’s difficulty tiers was made.

Requirements 8 and 9 were not completed by the end of this project, but were in progress. Requirement 8 was defined as “having a way of tracking user progress over a number of days or tests per module”[34], and requirement 9 as “displaying model methods for working out the correct answer for a question where the user inputted an incorrect answer”[34]. These requirements have been started and prototypes for each have been created. Model methods have been made for three different questions in the algebra topic. These model methods are displayed when the user submits an incorrect answer. A database was created to store the user’s progress, although it was never implemented.

The ‘could’ requirement has not been completed. This is because requirements 8 and 9 had to be completed before requirement 10, the ‘could’ requirement, was started. As requirements 8 and 9 were incomplete at the end of the project, development for requirement 10 never began.

As non-functional requirements are difficult to evaluate, it is a challenging task to determine in absolutes whether these requirements have been met. Any tests conducted with 11 year olds to determine their comprehension of the questions in this application would not have been representative of the population as a whole. Because of this, user testing was not conducted on 11 year olds. Instead, user testing was conducted with colleagues who work with students within the GCSE age range. Their feedback was incorporated into the development process, as is explained in section 7 of this report. Their feedback on the

final application was that the functional requirements have been completed for almost all students who are in the English Year 11, and the majority of students in the English Year 9. Because of this, these functional requirements have been deemed completed to a satisfactory standard for this project.

10.2 Discussion

By the definition of success laid out in the project specification, this project was successful. All of the ‘must’ requirements were completed, as well as one of the three ‘should’ requirements, although the ‘could’ requirement was left incomplete.

Overall, the project produced a proof of concept that the MyMaths homework functionality could be turned into a mobile application, which was the end goal of this project. The final product from this project could be released on the Google Play store in its current state. However, if this project were to be extended, then the application would not be uploaded to the Google Play store for another two to three months. This extra time would be spent completing the ‘should’ requirements, and working on the ‘could’ requirement for content which is examined on the foundation tier exams.

This project could have been improved with better project management throughout the first term of this academic year. The project fell behind early in development, as was discussed in the risks section of this report. Improvement was shown in project management as more time progressed. Improvements were made to the author’s time management skills, such as dedicating specific time each week to work on the project, as well as calculating the critical path for the features being implemented in this application. These improvements had a large impact on the final product, and counteracted the previous loss of development time from the first academic term.

This project was a worthwhile endeavour, as it highlights that more work can be completed on mobile GCSE revision applications. Students would benefit from an application endorsed by teachers, which aids in a student’s education in GCSE Mathematics. For many students, requiring access to a laptop or desktop computer is an insurmountable barrier between them and the revision resources they need; an application which only requires a smartphone breaks this barrier. This could increase GCSE grades, as most students have regular access to a smartphone, as is explained in the introduction of this report.

11 Future Work

The final product from this project is not perfect, and could be extended to be released onto the Google play store. This section of the report will outline key changes that would be made to the project if restarted. After, it will explain extensions which would be made to this project if it were to be continued from its current state.

11.1 Project Improvements

These improvements are changes that would be made to the application if the project was restarted from scratch.

11.1.1 Question Storage

Currently, questions are hard coded into the application, which has several disadvantages. One of the disadvantages is that any changes made to a question would only be available to users through downloading application updates. This delays question changes getting to users, and would cause lots of small updates. Another disadvantage to hard coding questions is that users cannot add their questions to the pool used within the application. This is because users do not have access to, and cannot change, the source code of the application.

There are two methods which could be implemented to store questions within the application, both with benefits and drawbacks, and both would require a change in the question display logic.

The first method of question storage would be to write the questions to a JSON file stored locally on the system. This would allow the manipulation of questions (inserting questions, updating questions, deleting questions, etc) locally. This would enable users to add questions just by appending an element to the JSON file. The small update to the current question logic would be to change from iterating through an Array list storing Java string types, to iterating through the data in a JSON object. The data would not be rewritten each update. This is because a second JSON file would be added that contains the user added questions, which would not be rewritten when there was an update overwriting the JSON file containing the developer created questions.

The second method of question storage would be to insert elements into a database, which would have the benefit of being persistent local storage. SQLite is built into Android, so it would be easy to implement. It would, however, require a complete rewriting of the question logic. This is due to the fact that it iterates through an Array list to display the question text. Instead, the question

logic would need to query the database, fetch some relevant data, find a way of displaying that data, and generate the random numbers necessary as per the requirements for this project.

If this project were to be restarted, then the method for implementing question storage would depend on the amount of question data that needs to be stored. For smaller amounts of question data the JSON file method would be used. This is because iterating through a JSON file until the right question is found would not add too much time onto question generation. The convenience of creating a working implementation of the JSON storage method would be too great to justify using the database method. However, when the JSON file gets large, iterating through questions until one specific question is found would be time consuming. In this situation, a database would be used. SQLite would be much faster at locating a specific row in the database than iterating through the JSON file.

11.1.2 Accessibility Improvements

If this project were to be restarted, then accessibility issues would be much higher on the list of priorities than it was originally. Questions could be played in audio form to aid those users who are visually impaired. Audio files could be saved to the user's device and played to explain any questions that could be generated through normal use of the application. Also, when an answer is inputted, the value could be spoken aloud; in the case of a multiple choice question, the selected answer could be read aloud. This would enable users which have some form of visual impairment to be able to use the application.

There could also be a setting which increases the font size of all text within the application, and would also increase image size. This would aid users with visual impairments, and would be built into the application from the first version. However, there is an option to increase the font size from within the Android operating system settings.

11.1.3 Content Grouping

Content within the application could be grouped into the appropriate age ranges, as well as by module. This would aid younger students, those ranging from academic Years 7 to 9, to utilise this application for revision. This would increase the usage of the application, and aid more students in their understanding of Edexcel GCSE Mathematics.

Currently, any students between the ages of 11 and 14 who wish to use the application could get questions about content that they have not yet covered

in the classroom. This is counterproductive to the aims of this application, as students would become frustrated that they cannot answer a given question, and they are less likely to continue using the application. Instead, with content grouped into appropriate academic years, as well as modules, students can be certain that the questions they are attempting are appropriate to their situation.

11.1.4 Trello

If the project was restarted, then some decisions would be changed to improve productivity. Development would be started later to allow for effective preparation of a project this size. Certain PMBOK processes would have been followed in the planning phase of this project. For example, developing a schedule should have consisted of defining activities to be completed, which produce deliverables, sequence the activities depending on their dependencies, estimate how long each activity would take, and then create the timeline for the project. Instead, this project estimated the amount of time each requirement was expected to take; estimating the completion of a task is harder than estimating the completion of a deliverable.

Deliverables for this project would be made from each requirement, which would make estimating the time to complete this project easier. Once the deliverables for this project were created, they could then be entered into a Trello[29] board, which would have allowed for more effective management. Trello is an organisational tool, and was used by the author as a to-do list whilst writing this report. Boards can be created which contain deliverables to remind team members which items still need to be completed. Figure 16 shows the Trello board created for this report:

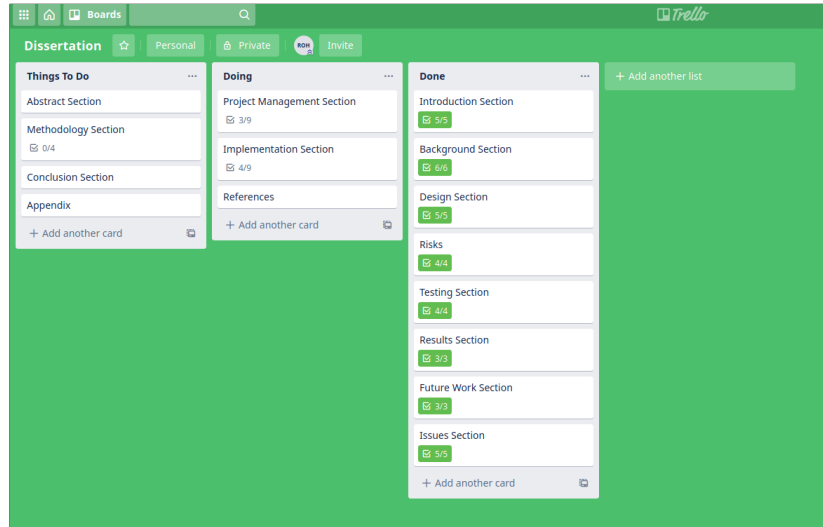


Figure 16: Trello board used as a to-do list for this report, screenshot at the completion of this section.

Figure 16 displays a to-do list for this report at the completion of this section. Using Trello as a to-do list for the deliverables of this report greatly increased productivity. This was due to increased preparation time for the final report, as opposed to the development cycle. Also, the Kanban software development methodology was used throughout the report writing process, which also contributed to the increase in productivity.

If this project was restarted from scratch, then Trello would be used, alongside the Kanban project management methodology, to improve efficiency and eliminate waste throughout this project.

11.2 Project Extensions

Project extensions are features which would be added to the application if this project continues from its current state.

11.2.1 Currently Incomplete Requirements

If this project continues, the first thing that will be completed are the three incomplete requirements. These are requirements 8, 9, and 10. This will mean that the application will track users' recent feedback, display model answers

when the user gets an answer incorrect, and there will be some kind of classroom-like learning resource.

These requirements were in the original scope of the project, but due to running out of time, they were incomplete at the end of the project. This is why they would be completed first if extra time was added onto this project, as requirements 8 and 9 are already in the prototype phase and could be fully implemented into the application given an extra month.

11.2.2 User Validation

User validation would be a key feature which would be added to the application. Two types of user accounts would be created, those available for students, and those available for teachers. Student accounts will be able to attempt any question from any module in any year group. They would have full access to all testing and learning content at any time. Teachers will have the ability to create classes, which will contain a group of students from the same academic year.

User validation was outside the scope of the project, as there are many potential legal issues with storing user data. This application would have to register under the Data Protection Act, and ensure that all data collection is compliant under the new General Data Protection Regulation, commonly referred to as the acronym GDPR, as this application would be only used within the United Kingdom. It was deemed that the potential legal issues which come with storing user data was not worth the risk for a project of this scope. This is why if the scope was increased, and time was added onto this project, then these potential legal issues would be solved.

A user would need to have the ability to register an account with an email address to utilise certain features present within the application. A single sign on (SSO) feature can also be added to allow users to sign in with other popular accounts, such as Facebook and Google. This would shift the burden of sign on security to the validating third party, although would not be practical for all cases. A user may not have a third party account, so there would need to be the ability for a user to register an account with their email address as well.

11.2.3 User Added Questions

A feature which would be useful is enabling users to add their own questions to the question pool of this application. This was deemed outside of the scope for the current project, as this is not a feature available in any of the major GCSE Mathematics revision applications. This feature would only be added after the

question storage method is changed, as currently, there is no way to automate adding a question into the question pool, due to how questions are stored.

This feature would only be available for user initiated testing. It would not be analysed by the statistical tools which would be built into the application as requirement 8. This is to prevent students from abusing the question adding process. A student could add questions which have a single, simple answer, or are much simpler than other questions in the same skill level. If the questions were analysed then a student could abuse the question adding mechanism to achieve 100% in a homework or revision set.

This feature would be implemented by adding an activity which would allow the user to input details for a new question. Users would utilise radio buttons to select the module and submodule for the specific question, as well as the year if that feature had been implemented. There would be a text box where users could type out the question text, and a specific set of characters would be used to represent variables which the application has to generate. This set of characters would then be found using regular expressions.

11.2.4 MyMaths Homework Feature

If the scope of this project was increased, then a homework feature, similar to that of MyMaths homework feature, could be implemented. This would allow teachers to set their classes revision tests to be completed by a given date. Students would require a certain percentage score in these tests to pass the homework, but they have no attempt limit. The students would then be notified of the homework via push notifications, as well as something visual within the application, preferably on the homepage so that students do not miss it.

Students could then attempt the specific homework, with their highest score being sent to the teacher to analyse. If a student achieves a score below the passing grade for that homework, then they could reattempt the homework, or look at some revision materials for that specific topic. These homework questions would be separate from the user added questions, for reasons of integrity discussed above.

Teachers could also be provided with the ability to select specific questions for a topic, and change the appearance probability. This would grant teachers the flexibility to customise a homework specifically for their class.

12 Legal, Social, Ethical, and Environmental Issues

This project did not face many legal, social, ethical, or environmental issues throughout development. There were no environmental or social issues faced for this project. There were limited legal and ethical issues which will be discussed in this section of the report under their respective subheadings.

12.1 Legal Issues

There was one legal issue faced throughout this project. This issue was related to the tools which were used. As the final source code was not wholly developed by the development team, there may have been some legal issues regarding licensing. Some of the code was generated through the tools used to aid this project, for example, any code pre-generated through the Android Studio IDE which was included in the final source code could create legal issues due to copyright laws.

This is why the licensing for all of the tools used is linked below as a reference, and relevant sections are quoted to ensure that there are no legal issues caused by this project.

12.1.1 Android Studio

Android Studio is built on top of the IntelliJ platform. It shares the same license as IntelliJ, as is shown in this comment on a blog post by the creator of IntelliJ, Dmitry Jemerov (Yole in the blog post)[24]. The IntelliJ IDEA[25] (the IntelliJ IDE) is licensed under two licenses, but as is explained in the blog post, the licensed version of IntelliJ IDEA that is used as the basis of Android Studio has the Apache 2 license. The relevant sections of the Apache 2 license are as follows: “You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form”[26]. This allows for the use of the IntelliJ and Android Studio IDEs for commercial and independent development, which this project falls under, and so there are no legal issues with using Android Studio as the IDE for this project.

12.1.2 Git

The majority of Git is released under the GNU public license[27], which enables copying, modifying, or distributing Git under any circumstances. The relevant sections of the GNU licence (from the preamble): “By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software—to make sure the software is free for all its users”[27]. This enables free use of Git for this project.

Small sections of Git are not licensed under the GNU public license, but under the GNU Lesser Public License (GPL)[28]. Relevant sections from the GPL preamble: “By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software—to make sure the software is free for all its users”. This section of the license refers to the GNU General Public Licenses, which the GPL is included as a part of. This enables the free use of Git for this project.

Since these are the only two external tools used throughout this project, and the only tools which could have impact on the source code of this project, then there are no legal issues caused by this project. Any other materials were produced by the author specifically for this project, and so there are no legal problems with using these materials.

12.2 Ethical Issues

This project avoided one ethical issue to do with the testing of the final product. Testing was conducted on colleagues who gave verbal consent to participate in the testing of the application. This testing method makes this project one which does not require ethical review under the departmental guidelines. “Student projects with primarily an educational purpose, for example asking peers or family members to test software as part of your course. Such projects should not include any form of deception or coercion or involve vulnerable groups (e.g. schoolchildren)”[29]. Testing was not conducted on vulnerable groups for this project, as user testing was conducted on colleagues over the age of 18, this project had no ethical issues.

13 Conclusion

In conclusion, this project was a success. It has produced a proof of concept application, which highlights a gap in the GCSE Mathematics revision market. On top of this, the final application produced by this project could be used by GCSE students to aid their revision of GCSE Mathematics, which was the

problem that this project aimed to solve.

A GCSE Mathematics revision application endorsed by schools will most likely be produced in the future. Throughout the development of this project, MyMaths have released an application which is accessible through tablets. It currently has a small amount of content available, with more being added throughout the coming months[30]. MyMaths have created this tablet application because all of their content before November 2019 was written in the Flash multimedia software platform. Adobe announced in July 2017 that it was going to “stop updating and distributing the Flash Player at the end of 2020 and encourages content creators to migrate any existing Flash content to new open formats”[31]. This means that MyMaths is migrating all of their content away from Flash, and in doing so, they are producing an application available on tablets.

This project developed software that could be released as an application onto the Google Play store, and would be useful for students sitting their GCSE Mathematics examinations from November 2021 onwards. All of the questions in the application have been taken from non-calculator past papers, so no calculator is necessary. However, the questions could be adapted to include more difficult numbers, which would require a calculator. This application could also be updated with new questions after each exam cycle is complete.

The development of this application could continue into a commercial product. If the improvements in the future work section of this report are completed, then the application would be brought up to paid quality, and so could be released for a small price on the Google Play store. It could also be continuously updated with new features, and the scope of the project increased.

14 References

[1] - <https://www.gov.uk/government/speeches/reformed-gcses-in-english-and-mathematics> (Accessed 29th March 2020)

[2] - <https://qualifications.pearson.com/en/qualifications/edexcel-gcses/mathematics-2015.html> (Accessed 29th March 2020)

[3] - <https://www.pewresearch.org/internet/2018/05/31/teens-social-media-technology-2018/> (Accessed 30th March 2020)

[4] - <https://www.comscore.com/Insights/Presentations-and-Whitepapers/2017/2017-US-Cross-Platform-Future-in-Focus> (Accessed 30th March 2020)

[5] - <https://gs.statcounter.com/os-market-share/mobile/worldwide> (Accessed

30th March 2020)

[6] - <https://qualifications.pearson.com/content/dam/pdf/GCSE/mathematics/2015/specification-and-sample-assessment/gcse-maths-2015-specification.pdf> (Accessed: 6th October 2019)

[7] - <https://www.mymaths.co.uk/> (Accessed 22nd March 2020)

[8] - <https://www.mymaths.co.uk/help.html> (Accessed 22nd March 2020)

[9] - <https://www.khanacademy.org/> (Accessed 22nd March 2020)

[10] - <https://www.khanacademy.org/math/trigonometry/trig-with-general-triangles/law-of-sines/v/law-of-sines> (Accessed 22nd March 2020)

[11] - <https://apps.apple.com/gb/app/khan-academy/id469863705> (Accessed 22nd March 2020)

[12] - <https://play.google.com/store/apps/details?id=org.khanacademy.android> (Accessed 22nd March 2020)

[13] - <https://mathswatch.co.uk/> (Accessed 25th March 2020)

[14] - <http://www.mathswatch.co.uk/subscription/4594745491> (Accessed 25th March 2020)

[15] - <http://coolt.ch/notizen/wp-content/uploads/2016/02/Head-First-Android-Development-2015.pdf> (Accessed 31st March 2020)

[16] - <https://developer.android.com/docs> (Accessed 31st March 2020)

[17] - <https://github.com/AlphaMustang> (Accessed 31st March 2020)

[18] - <https://trello.com/> (Accessed 8th March 2020)

[19] - <https://developer.android.com/guide/components/activities/intro-activities> (Accessed 28th March 2020)

[20] - <https://developer.android.com/guide/components/fragments> (Accessed 28th March 2020)

[21] - <https://developer.android.com/studio/run/emulator> (Accessed 26th March 2020)

[22] - <https://developer.android.com/studio> (Accessed 17th April 2020)

[23] - <https://github.com/AlphaMustang/thirdYearProject> (Accessed 28th March 2020)

- [24] - <https://blog.jetbrains.com/idea/2013/05/intellij-idea-and-android-studio-faq/#comment-4939> (Accessed 29th March 2020)
- [25] - <https://www.jetbrains.com/idea/features/> (Accessed 29th March 2020)
- [26] - <https://www.apache.org/licenses/LICENSE-2.0> (Accessed 29th March 2020)
- [27] - <https://github.com/git/git/blob/master/COPYING> (Accessed 22nd March 2020)
- [28] - <https://github.com/git/git/blob/master/LGPL-2.1> (Accessed 22nd March 2020)
- [29] - <https://warwick.ac.uk/fac/sci/dcs/teaching/ethics> (Accessed 22nd March 2020)
- [30] - <https://support.mymaths.co.uk/technical-support/what-mymaths-content-has-been-updated/> (Accessed 16th April 2020)
- [31] - <https://theblog.adobe.com/adobe-flash-update/> (Accessed 16th April 2020)
- [32] - Head first android development pg 312
- [33] - <https://www.agilealliance.org/glossary/poker/> (Accessed 20th April 2020)
- [34] - <https://drive.google.com/open?id=1SOxvimR9xaiPEVLAIH3B9jKxaiNNzVrV> (Accessed 1st May 2020)
- [35] - <https://drive.google.com/open?id=1BtAMqo6Dd-cObO9DbjG9RZZmNwRaOx2g> (Accessed 1st May 2020)
- [36] - <https://developer.android.com/reference/android/app/ListFragment> (Accessed 4th May 2020)
- [37] - <https://developer.android.com/reference/android/app/ListActivity> (Accessed 4th May 2020)