DUBLIN INSTITUTE of TECHNOLOGY
Institiúid Teicneolaíochta Bhaile Átha Cliath

Personalised Game Application for aiding Primary School Students with Math Learning

Final Year Project Report

DT228
BSc in Computer Science

Saira Javed Barlas
Dr. Pierpaolo Dondio

School of Computing
Dublin Institute of Technology

1st September 2017

Abstract

The primary goal is to create a math application for primary school students between the ages of eight and twelve. The math application would have various mathematical categories available (stored in the database of the application) and the student would be able to play the math quiz. The many categories included are addition, subtraction, multiplication, division, and a mixture. The student, after the completion of the quiz, would be able to view the result in a graphical form, which would show the percentage of CORRECT and INCORRECT answers of the quiz. The student would also be able to view the full result of the quiz. The full result would consist of the questions, the selected answers, and the actual answers of the questions. The application's interface would be developed using material design and an appropriate complimentary colour wheel. The android application would also have an Instruction screen to allow the student to be informed on how to navigate the application and the purpose of the math application.

Declaration

I hereby declare that the work described in this dissertation is, except where otherwise stated, entirely my own work and has not been submitted as an exercise for a degree at this or any other university.

Signed:

_____

Saira J. Barlas

01/09/17

Acknowledgements

I would like to thank Dr. Pierpaolo Dondio for his patience and guidance throughout this project. I am very grateful to him. I would also like to thank all the primary and secondary school teachers, especially Ms. Maria Kate Carberry, for aiding in my research and answering my questions. A great thanks to my family and friends for the encouragement and support I was given. I would also like to thank Dr. Bianca Schoen-Phelan and the staff at DT228/4 for giving me another chance at this project. Last but not the least, I would also like to thank God for helping me in the project's journey and its completion.

Table of Contents

1. Introduction

a. Project Overview

The project was inspired by the supervisor and was the lecturer's original idea. The main idea was to develop a math application for primary school students to aid them with math learning. Discussions were carried out with the supervisor and a PhD student in Dublin Institute of Technology. The project's idea was then strengthened to make it into a math application that would target certain primary school students.

The idea is to develop an android application which would be targeted towards primary school students between the ages of eight to twelve. The main purpose is to develop basic math skills of students and to allow them to become more confident and comfortable with mathematics and become more confident in using it with technology.

The application would allow the student to pick between various categories; from Addition, Subtraction, Multiplication, and Division. There's also a category which displays a Mixture of all these categories in one. The student would go through at least 10 questions which have been previously stored into the category, each. They would then be shown a graphical result of what percentage of the questions that were CORRECT and what percentage were INCORRECT. They would also be given an option to view the results and the questions individually to see which questions they answered correctly and which questions they answered incorrectly. They would also be allowed to retake the quiz to become much better at the questions they either found to be difficult or did not understand.

b. Project Objectives

There are various objectives which need to be handled in this project. The main objective is to identify the end user and the requirements need to be set out on how to develop this android application. This application is targeted towards primary school students hence the application shall be designed for them using the correct principles. Second objective is to create a database with the right set of mathematical questions in relation to the corresponding operators (addition, subtraction, multiplication, division, mixture). The level of difficulty should match the level of the primary school students between the ages of eight to twelve. Thirdly the application should be easily navigable by each student and should be easily understandable by each student. In order to enhance the math skills of students, the fourth objective is to create a graph which shows the student's percentage of CORRECT and INCORRECT mark and allows the student to view their results and see which questions they answered correctly and which questions they answered incorrectly.

c. Document Structure

i. Research

This part of the document contains all the research carried out throughout the project; background, technologies, and any other solutions that were found. This also contains other relevant research done in order to accompany this project.

ii. Design

This contains the methodology which was used for this project and the mock ups and design of the various screens of the android application. It also contains the use case of the whole application.

iii. Architecture & Development

This contains the full architecture of the gaming application and the full development components of the project. It also contains the major code classes implemented in this project and the source code layout.

iv. System Validation

This is the part of the project which contains the various test cases carried out on the android application. User Testing is also shown in this section of the project.

v. Project Plan

The various analysis and review of how the project was managed and changed from initial stages until completion. It also deals with future suggestions.

vi. Conclusion

Sums up all the key learning obtained and what the project objective was.

vii. Bibliography

Contains all the references and material which was used to generate the project and the report.

2. Research

a. Background Research

Various background research was carried out in regards to the impact of games in the classroom and education and how students learn using different learning styles. The seven learning styles, specifically, were researched on how each student learns differently. The teaching methods of teachers in regards to math was also researched. Many different user interface design principles were also researched to bring about the correct interface for the target audience. Jakob Nielsen's and Rolf Molich's Ten User Interface Guidelines, which overlap with Ben Shneiderman's 'eight golden rules', were highly researched and evaluated to create the appropriate user interface for this application. Android Design Principles developed by the Google team were also discussed. Alternative existing android math solutions were also researched. All of these were taken into consideration when developing the math application for primary school students for math learning.

i. Games and the Student Brain

An individual never stops learning, be it the age of 5 or the age of 50. Learning is a mental process which isn't optional nor is it voluntary. Hence education is an important component in everyone's lives in order for an individual to grow and reach new heights. Education is also a medium for one to practice new skills. For children, education is compulsory in more ways than one; it gives them a platform to experience the world using their own perception. It is said that infants explore their worlds by actually applying their own sight and sound (Bransford, 2000). When one is a child, and especially in a classroom environment, this is when the senses are at their peak. This is also where the concept of games arises and how they can be deployed in a primary school environment.

Digital games have been referred to as a source for learning since the day they were released. Only recently have they become prone to being utilised in the classroom and hence are referred to as "serious games" (Rooney, P, 2007). They appeal to many different learners and are successful paths of interaction for many students (Mackay,

2013). It's a learning tool for educators and game developers alike and provide a concept of active learning (Rocha, 2016). It provides a platform for one to not follow the rules but to let the mind wander and be free (Mackay, 2013). Games also give the ability to measure learning never done so before (Mackay, 2013) and have been cited as a reason for the development of positive memories towards education (Education World, 2013).

They are also responsible for the development of many skills. Social skills, being the pinnacle of any individual's lifetime, allow students to engage with the others around them (Rooney, 2007). With the growth of engaging games, collaboration between students has become the utmost importance. Considered "architectures for engagement" (Mackay, 2013), games bring humans together as humans are contemplated to be at their best through collaboration and in a network (Mackay, 2013). Human brains are considered the most effective when engaged in togetherness rather than alone (Mackay, 2013).

Cognitive skills, especially in regards to problem solving and critical thinking (Rooney, 2007) are key in regards to a person's, and a student's maturity. Students nowadays heavily rely on the presence of a different medium in which learning such skills are made easier. Games provide the perfect gateway for students to learn and practice these skills. Students are also introduced to the freedom of choice (Mackay, 2013), a concept completely foreign to them.

"*Education harnesses our natural curiosity as human beings to understand the universe and everything in it*" (Colgan and Colgan, 2017). Game development has become the forefront of education based learning. Educational games are now being developed for more difficult concepts and hence helping students to easily grasp them (Colgan and Colgan, 2017).

Hence the use of games to integrate a student into the classroom is an efficient and invaluable resource. Games have become the pinnacle of engagement and success for many students. Therefore, the focus of this project was on developing a gaming application, which specifically concentrates on math learning, for primary school

students between the ages of eight to twelve in hopes of making math more engaging and enjoyable.

ii. Teaching a Student

Each student requires different care and precision when it comes to being taught any subject matter. There are different goals which need to be met in order to insure how a student learns and different strengths and difficulties that come in doing so. There were various teaching methods that teachers carry out in the classroom that were researched in order to design the application with the correct content. In relation to maths, the following points were considered of high importance:

1. **Goals:** There should always be goals which a teacher should prioritise in order for children to get to the correct answer (Delaney, 2017). The goal should also be show them the result as well so the children is aware of what they found to be correct and what is incorrect. The content should also be appropriate for the level of the child and should help them develop.
2. **Strengths:** Showing the student the result and the incorrect answers should not be taken harshly (Delaney, 2017). Each student should be made aware of the progress they have reached and they should not be penalised for it as each student learns differently.
3. **Repetition:** Mathematics is a repetitive process. Practice makes perfect and repetition in maths is key. More similar questions need to be provided and repeated regularly to reach a certain level in order to progress. There is no other way of learning math without actually putting in work as, admittedly, mathematics is difficult to begin with (Delaney, 2017). Hence more time and effort is needed to reach the suitable goals and cater to the strengths of students when it comes to math.

Carrying out research in regards to how a teacher applies different teaching methods, the research indicated heavily on repetition and making the student aware of the questions they answer correctly and incorrectly. This research also aided in making the math application free of any time frame in order to help the student be more comfortable in using the application. There would be no external pressure and the student would learn math at their own pace. The student would also be made aware of the different questions they answer correctly and incorrectly. A graph would be an ideal visual representation of the questions that the student answers. The questions can be practiced as many times as the student wishes with a new result each time.

iii. The Seven Learning Styles

Learning style are known to group many people into a certain category depending on how they learn. Everyone has a mixture of various learning style with some people using a certain learning style more than others. The learning styles aren't fixed and depending on different situations a person may use a different learning style. It is better to utilise many different learning style in comparison to only one (Learning Styles, 2017). This is a new concept which has not been grasped as quickly as it should have. The following are the seven learning styles which are widely used:

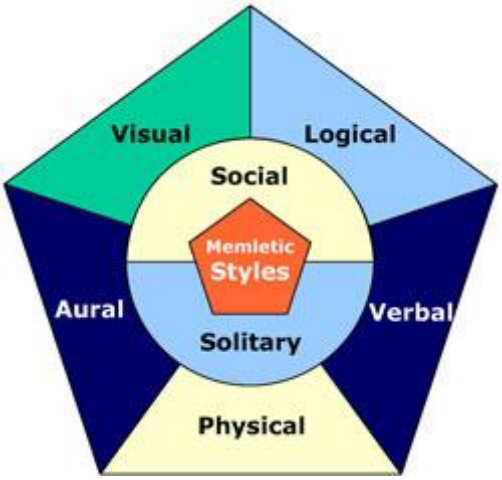


Figure 1 The Seven Styles of Learning

1. **Visual:** this is a learning style which requires the use of images, pictures, and anything visual to help in the aid of learning.
2. **Aural:** this is a style which relates to listening to music or sounds to help with learning.
3. **Verbal:** the use of words and phrases are used whether it is in speech or writing to learn.
4. **Physical:** the body is used i.e. the use of hands or touch is considered essential in order to learn.
5. **Logical:** this revolves around being able to learn through logic and reasoning and problem solving.
6. **Social:** this is a learning method which requires interacting and learning within groups and with other people.
7. **Solitary:** this learning style is used for learning by one's self and in working alone to learn.

All these various styles can be used as a guide on how one learns. The preferred style also changes the way events and experiences are represented internally for each individual and information is recalled and even down to the words that are chosen. Research has also shown that each learning style is controlled by a different part of the brain. Hence if more learning styles are utilised, the more the brain is involved during learning. It helps to remember more and to learn more (Learning Styles, 2017).

This research greatly aided in greatly enhancing the idea of the proposed project. As the android application would allow touch and problem solving, it would already be utilising the two learning styles; Physical and Logical. The graph would also represent the answers of the student and would also allow the student to review their results which would require the inclusion of Visual and Verbal learning styles. The math application is for students to learn, be it by themselves or with the help of different students in the classroom. As both utilise the styles of Solitary and Social, the application can be considered for every student present as each student learns differently. There are no sounds in this game as it was not considered appropriate to play high levels of music during a classroom session. Yet the teacher may utilise the application with her students and communicate the answers and the results in the classroom via the blackboard or just recording the results. This can indirectly be considered as the Aural learning style.

Taking the seven learning styles into account, the application shall be developed for students that use various different learning styles to learn any subject matter, in this case; mathematics.

iv. Principles for User Interface Design

It is essential to know the target users' needs and expectations when developing a software product. In regards to this project, a math application for primary school students, various heuristics and design principles are used and applied for the student to easily find the application to be navigable and usable.

Jakob Nielsen and Rolf Molich (with the addition of Ben Shneiderman's 'eight golden rules') established the 10 rules of thumb (Wong, 2017) when it came to user interface design. The following shows how these rules will be used with the project application:

1. **Visibility of system status:** the user should be informed of what is going on inside the application and every status should be displayed with enough amount of time. This shall be displayed in the project application via the correct question number being displayed and the student would be able to see what answer has been selected. The student will also be able to navigate through the questions given.
2. **Match between system and the real world:** the system should be able to communicate with the user in everyday language rather than technical jargon which the user doesn't understand. The project application will show everything in simple English and the questions shall be asked in English as well. The symbols for addition (+), subtraction (-) etc. shall be used so the student understands the question correctly as they are mathematical terms for which the application is designed for.
3. **User control and freedom:** allows the user to navigate backwards and forwards and redo and undo actions. The application will allow the user to go back and forth the questions they are answering. This application will also allow the student to retake the quiz and allow the student to use the back button on the phone to move back to the previous question if needs be.
4. **Consistency and standards:** this implies that all terminology and graphics used in the system should be consistent and should not apply various meanings throughout the system. The math application will have images which are consistent and will have individual screen titles which indicate the title for that

screen only. The buttons shall also have unique names so the student doesn't get confused on what is being asked.

5. **Error prevention:** the user should be prevented from making errors and all errors that due arise should be handled by the system and not the user. The math application will deal with errors by itself. Any question answered incorrectly will be shown in a graph form to the student and the student will have the option to view the results and hence the incorrect and correct answers.
6. **Recognition rather than recall:** this states that every option should be available to the user and they should not be hidden. The student will have every option at their disposal and for navigation. They shall click on buttons and can easily move from one screen to another via information already available. There will also be an instruction screen to inform the student of various actions that can be carried out.
7. **Flexibility and efficiency of use:** the user should be allowed to use shortcuts to navigate through the system. For a student, the back button can be used instead of clicking on buttons directly in the application. An example would be to use the back button to go back and view the previous questions in comparison to completing the quiz and then viewing the results and then restarting the quiz.
8. **Aesthetic and minimalistic design:** implies that the user interface should not hold any useless irrelevant information and the design should be as uncomplex as possible. The math application will be direct and concise with the right visual bright colours and minimum text. The main aim of the application is for the student to interact with the math questions and view the result and the graph. Hence the application will be simple and direct.
9. **Help users recognise, diagnose and recover from errors:** The user should be able to understand any errors that appear and all error messages should be in plain text. The application will only express everything in plain English and nothing technical at all in order for primary school students to be able to interact with it easily and confidently. If the answer is not selected and the student wishes to go to the next question, a pop up message (toast) will appear telling the student to please select an answer first.

10. **Help and documentation:** the system should have appropriate documentation available for the user to interact with it if need be. The math application will have an instruction screen which will let the student know on what the application can be used for and how it can be navigable.

The above shows the principles being utilised within the math application so the students can easily use it for their mathematical purposes. The application will be simple and concise with no presence of useless and irrelevant jargon and in only simple and plain English. The application will also be easy to navigate and the student will find the purpose of the application satisfactory for answering mathematical questions and receiving feedback.

Since the application being developed is targeted toward an Android device, the appropriate **Android Design Principles, developed by the Google team** (Android Developers, 2017), were also taken into consideration. The following shows the chosen principles in relation to the application and how they will be used with the application:

1. **Delight me in surprising ways:** the surface should be beautiful and carefully planned. The math application has been designed using various colour wheels until the appropriate colour scheme was formed. The colours chosen for the interface are complementary and essential for user interface design. The phenomenal material design with its list view and card view will also be used for the interface design.
2. **Real objects are more fun than buttons and menus:** users should be allowed to touch objects. The buttons in the application have been sufficiently provided to allow the user to easily navigate through the application and view many other features.
3. **Keep it brief:** short phrases and simple words should be used. As the math application is targeted for students, the text has been minimal for the game application. It is used only when necessary i.e. asking questions and with the instruction screen for example.

4. **I should always know where I am:** the user should know where they are present in the application. All screens have titles in the math application and hence the student shall never be lost.

The above two principles will be the key and shall be held upon as important guidelines in designing the math application of this project.

Alternate Existing Solutions

Already available math application games were researched for this project to see the various differences and essentials each required and included. The following two were considered the optimum examples to review when it came to catering to students and playing mathematical games.

1. **Kahoot**

Figure 2 Kahoot PC Interface

Kahoot is a game which was first researched as suggested by the supervisor. Kahoot is very useful for a classroom environment and optimises "*game-based approach to blended learning*" (Kahoot Journal!, 2016). A quiz is established by a teacher, she creates her own set of questions and answers, which generates a PIN and the students enter that PIN into their handheld device. The above quiz question ends up looking like the one below on a handheld device:

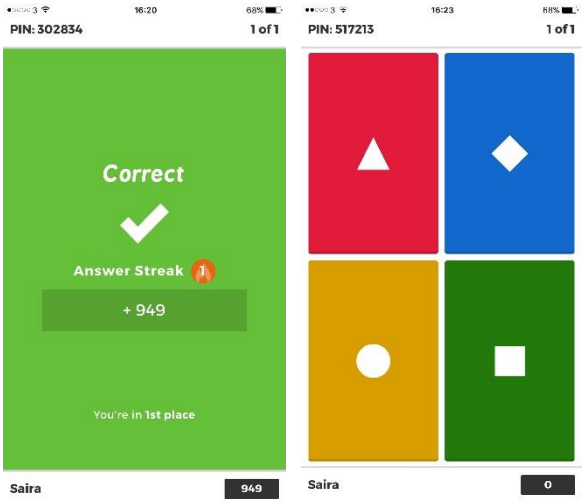


Figure 3 Kahoot Mobile Interface

2. Math Puzzle – Brain Trainer

This application is developed in regards to keeping the brain active and challenging one's self to brain puzzle games (Math Puzzle – Brain Trainer, 2016). It has a category list whereby students can choose the category they want to most practice under. There's also the Score screen which displays the highest score obtained during that exercise.

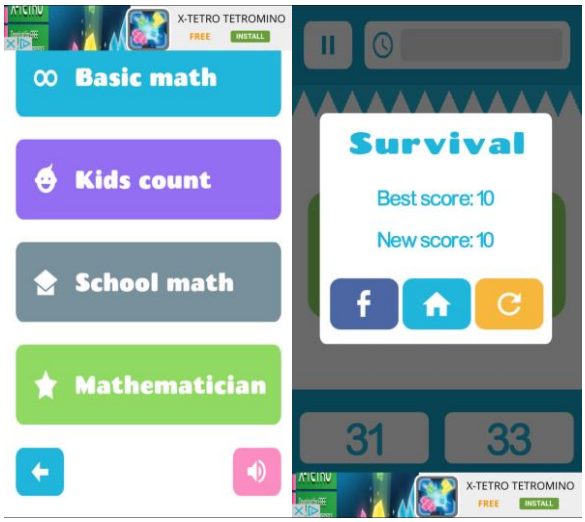


Figure 4 Math Puzzle Brain Trainer Mobile Interface

The above two applications have been very beneficial in determining what kind of application a math application for primary school students should look like. The above two vary in that they do not show the answers nor the questions to be reviewed. They also do not allow for students to pick a category i.e. addition, subtraction, multiplication etc. There is no availability for the student to practice a certain category available at all.

The applications also do not provide any feedback to the students and the students don't benefit without any feedback. Hence the math application being developed will contain a full review which would show the students the answers they picked and the correct answers that the questions entailed. It would also display all the results in an appropriate and a simple readable fashion so the students can learn from the application. The students will also be shown a graph and the percentage of how correct their answers were, which would make things visual for the student.

b. Technologies Researched

Various technologies were researched in order to finalise that Android would be the platform for this math application. As Java and XML were already languages that had been experienced before, these would be compatible with Android Studio. Git would be used for version control and Material Design of the interface shall be implanted. Many third party libraries will be used for the graph and the scaling of the application. A database shall be created to store the math questions for the application. Research was also done into designing and acquiring the essentials, such as graphics, text, colour scheme, for the user interface of the application.

i. Android /Android Studio

Android is perceived as a very popular operating system especially in regards to phone devices. It is Open Source and Linux based and is the primary choice for 87.6% of the market (Android Studio, 2016) (IDC: Smartphone OS Market Share, 2016). It's ability to be cross platform gives it a very good advantage and hence enables many phones and tablets etc. to be compatible with similar apps. These points were found to be extremely crucial in regards to the development of the math app as this means that android can reach out to more students and hence more people.

The Android Studio IDE which compliments the development of android applications is seen as a very efficient and developer friendly platform. An android project usually consists of the manifest file, java files (also contain JUnit test code), and res files (anything other than code goes here) (Studio, M, 2016). A project can easily be established and built using android. All the resources are present and third party libraries can be easily implemented for any project production. The following shows the important files in the development of an android project:

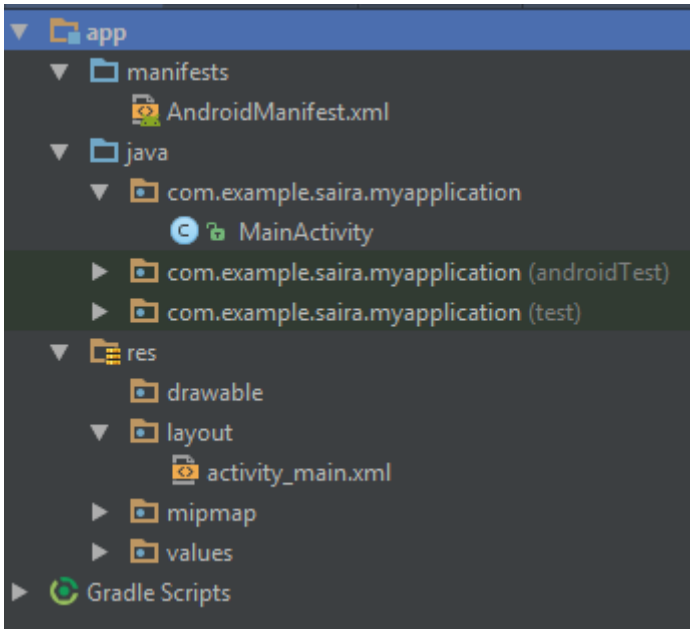


Figure 5 Android Studio Hierarchy

- *manifests folder*

AndroidManifest.xml contains the components such as activities, receivers and content providers (Manifest, 2017). The java classes are also implemented in this file and hence their capabilities can now be used. This file is extremely important as android is then aware of the classes and capabilities it can handle.

- *Java folder*

The java classes (e.g. MainActivity class) under the java folder is where the java code is written. The java code is separate from the UI implementation as java handles the behaviour of the android application (Developer.android.com, 2017). This is an advantage considering different types of code are kept separate and hence allows for greater project management of any calibre. The java code still has the ability to manipulate the UI code which is written in XML (Developer.android.com, 2017).

- *layout folder*

This folder contains all the UI elements of the application. It is implemented using the straightforward XML language with a vocabulary which is very similar to the widgets and content it corresponds to (Developer.android.com,

2017). Due to its simple nature and its powerful widgets, XML with java, complement each other in regards to the development of a perfect application.

This makes android the best option to utilise in regards for game application development. It is easily accessible to the market and can reach out to the correct audience. It keeps the behaviour and visual layers separate hence enabling better and effective code management for the proposed project.

ii. Java

Java is a programming language which was introduced during the second year of university. It was also used in the process of android development in third year. Having already been accustomed to java, this made it an excellent choice to be considered for the production of this project. It's also the primary language used in much of android development and hence can also be considered the 'go to' language for any sort of application development.

Java is an object-oriented language which comes in the form of different classes and methods and is widely used in the internet realm and beyond (venturehire, 2017). Hence this language, with the largest community online, was considered to be the language for this project (venturehire, 2017). The language is also easy to learn and if any problems are encountered, due to its popularity, can be solved seamlessly due to the vast range of helpful resources available (venturehire, 2017).

In order for the project to reach completion, a language which one is more experienced and comfortable in should be greatly considered. Java is also the primary language of android development and hence there was no reason to reject the notion of using java throughout this project.

iii. XML

It was apparent that there will be XML development in the project if android was chosen to implement it. XML is known as a markup language which is easy to read and learn as it has a hierarchical view (Programming, 2017). No knowledge of XML is necessary in regards to android development as stated by many online resources (Programming, 2017). Having two years of experience in XML already, this confirmed that android would be the obvious platform to implement this project. The

following shows an XML snippet which implements an EditText and a Button widget on the android screen:

```
<EditText
    android:id="@+id/playerName"
    android:layout_height="wrap_content"
    android:layout_width="30px"
    android:singleLine="true"
    android:layout_weight="0.6" />

<Button
    android:text="Find"
    android:layout_weight="0.2"
    android:layout_alignParentRight="true"
    android:id="@+id/findPlayerButton"
    android:layout_height="wrap_content"
    android:layout_width="wrap_content"
    android:padding="0dip" />
```

Figure 6 XMLsnippet

iv. Git

"Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency" (Git, 2016) which was first utilised in second year of university. Hence it was already understood as a tool which did not demand much difficulty. All code can be tracked and it holds the ability to undo changes or retrieve changes previously made. It also keeps a record of all the changes that had been made since the project started utilising Git. There is a version control integration feature which is available within the android IDE. Hence using Git for applying version control to this project is an extremely easy way to maintain and keep updated with the project. It also allows for one to rollback changes which are deemed not best for the project at all.

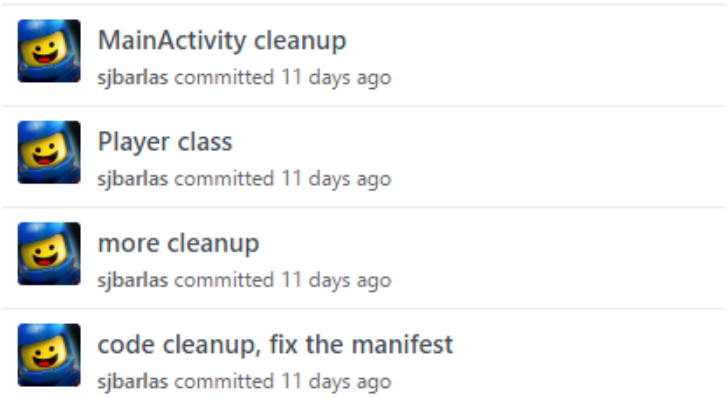


Figure 7 Git Commits

v. SDP (Scalable DP)

This is a third party library developed in regards to screen resizing. It's an android SDK which allows a new size unit called sdp (scalable dp) (SDP, 2017). This size unit then scales with the screen size and hence can support multiple screens. This was chosen to allow the math application to work without breaking any other screen resolutions. The math application would appear to the correct resolution of the device, be it tablet or phone.

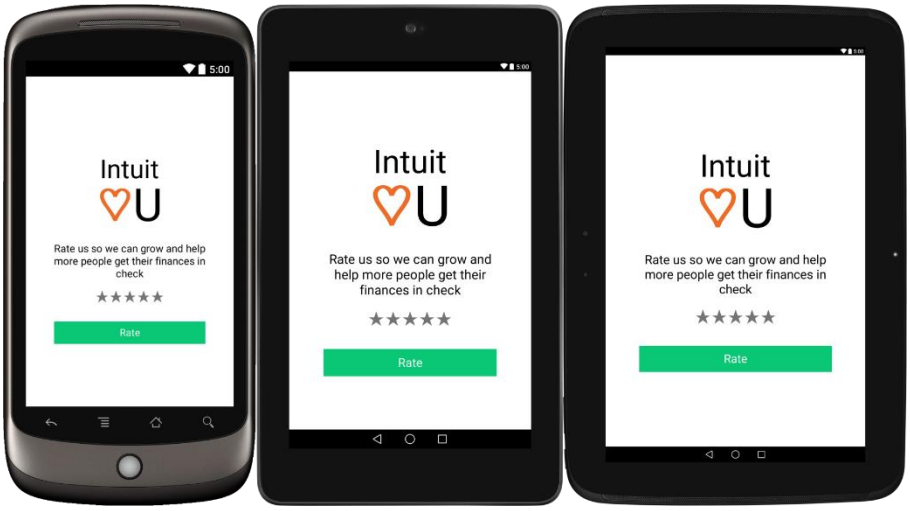


Figure 8 Automatically scaled resolutions using SDP

vi. CircleView

This third party library creates a circle view which allows the developer to add a title and a subtitle to the circle (CircleView, 2017). This was picked especially in regards to creating a graph which would then display the percentage of correct and incorrect answers in the math application. The questions answered correctly would appear in a separate circle view called CORRECT with a corresponding percentage. Another circle view would be generated where the incorrect answers would be displayed under the title of INCORRECT with a corresponding percentage.

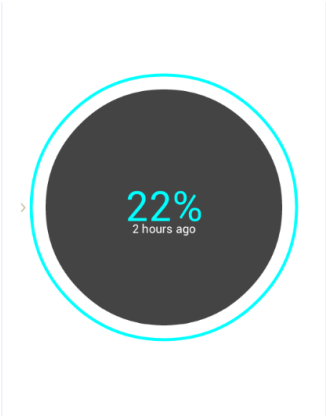


Figure 9 Circle View with a Title and a Subtitle

vii. Material Design

Material Design (Android, 2017) is a user interface design developed by Google which is currently very popular for Android. It's based on the theme and influenced by cards. It provides a simple and clear interface and looks as if it is well stacked and minimalistic. This design was chosen especially to create a math application which would appeal to all the students and would allow for easy navigating. Material Design allows for:

1. **Lists with RecyclerView:** this shall be used to create clear and concise lists in the math application.
2. **CardView:** this allows content to be displayed as cards. This shall be used to display the results of the student's questions.

The whole math application shall be designed using Material Design.

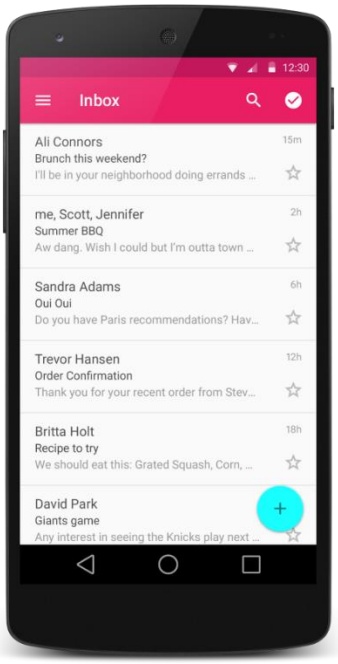


Figure 10 List View

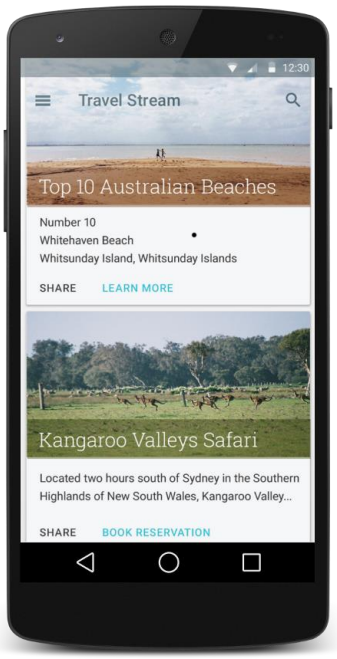


Figure 11 Card View

viii. SQLite and SQLiteAssetOpenHelper

SQLite database shall be used to store and populate the various questions in the math application. SQLiteAssetOpenHelper (Android SQLite Asset Helper, 2017) is a helper class which will manage the creation of the database and the version control using the application's raw asset files. The math application will come with the database and the questions already implemented in the app.

SQLiteAssetOpenHelper is implemented as an extension of the SQLiteOpenHelper. Rather than creating onCreate() and onUpgrade() methods for SQL execution, the asset files will contain all the statements already in the assets directory. Hence these contain creation and upgrading scripts already.

ix. DB Browser for SQLite

This is an open source tool which was used to create and edit the database compatible with SQLite for the math application (DB Browser for SQLite, 2017).

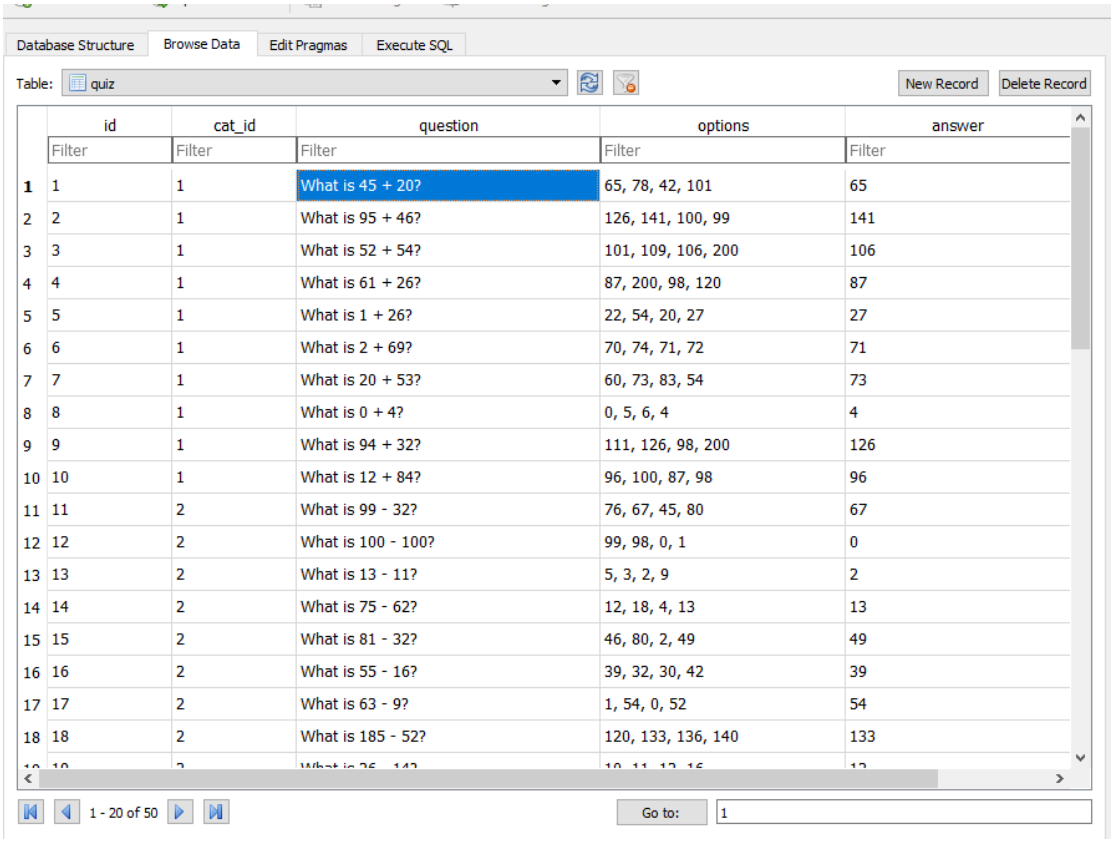


Figure 12 Application's database (quiz table) as seen in DB Browser for SQLite

c. Other Relevant Research done

i. Images – Pixabay

Images allow the sense of sight to become engaged with memory and learning. Images form a canvas of pictures and aid in generating learning experiences with the task. Pixabay was chosen for the source of these images as they are readily available for inclusion into any project by anyone (Pixabay.com, 2017). The sense of sight being the most important sight of a student; the application would be much more effective with students if it includes different images. The categories for various quizzes will include various images to represent the different category available. The images shall also be used to show the correct and incorrect questions answered by the student. This site was also used to acquire the icon launcher(s) for the application.

Figure 13 Launcher icon of the application Appy

Figure 14 Various images used for the math categories

ii. Application Name – Cooltext

This site was used to generate the application name which would then be displayed on the app in the main activity screen (Cooltext, 2017). The title of the application is "Appy".

Figure 15 Application name designed by cooltext

iii. Colour Scheme – Session College

The development of the application interface required looking at the colour schemes and how it would be visually aesthetic in the application for the target audience. Since the application is for students, the appropriate bright colour scheme was generated and identified and utilised (Colour Calculator, 2017). These are complimentary colours. The following is the scheme which has been generally used in the design and development of this application:

HEX	RGB			CMYK			
HEX: **#4fc0e7**	R: **79**	G: **192**	B: **231**	C: **66**	M: **17**	Y: **0**	K: **9**
HEX: **#49cdb6**	R: **73**	G: **205**	B: **182**	C: **64**	M: **0**	Y: **11**	K: **20**
HEX: **#e77852**	R: **231**	G: **120**	B: **82**	C: **0**	M: **48**	Y: **65**	K: **9**
HEX: **#e74f68**	R: **231**	G: **79**	B: **104**	C: **0**	M: **66**	Y: **55**	K: **9**

Figure 16 Colour Scheme of the application Appy

3. Design

a. RAP Waterfall Methodology

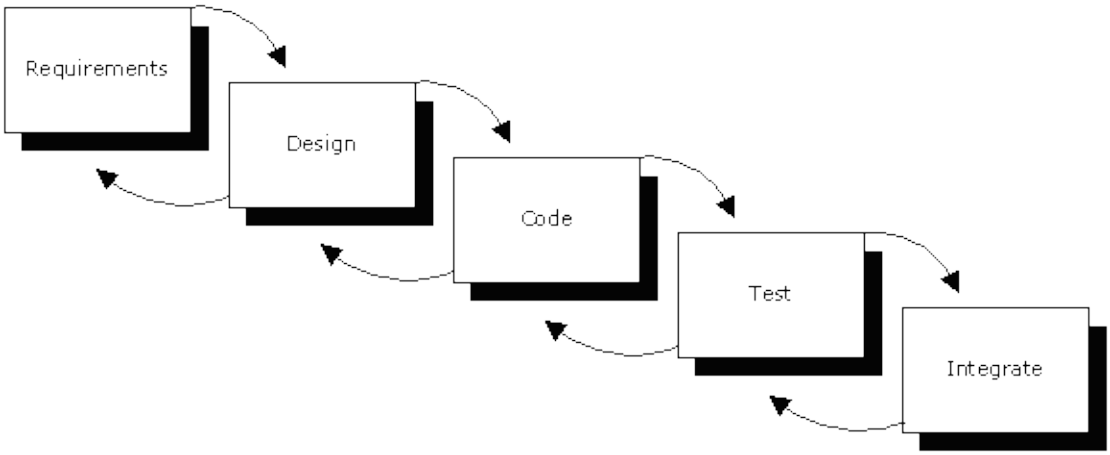


Figure 17 Different Phases of the RAP Waterfall Model

The RAP Waterfall Model was selected because it met the needs of how this project was to be implemented. The key advantage of the RAP Waterfall model is that it consists of phases which cannot be touched unless the previous phase has been completed. It also allows for an individual to return to a phase if an error or compatibility issues have arisen. This suited the project more so considering the project was first approached by the need for the presence of requirements. The Android operating system was the first to be considered the essential requirement for this project. As the project progressed and ideas were changed, the advantage of choosing this model became more apparent.

The major phases of this project has been Requirements and Design respectively. One cannot start building a game application without realising the criteria nor the needs that should be met in order for the game application to deliver what is promised at the start. The project was heavily researched at the start in order to understand the background of game applications and game design. This allowed for the appropriate and crucial technologies to be investigated before the Design phase could be concentrated on. Once the project reached the Coding phase, there were different aspects of the project that now needed to be either implemented or researched again,

this model had the capability to allow for both cases to be furthered and allowed the flexibility for the idea to be changed.

The Design phase involved the design of the game application in general and building a prototype. Since waterfall allows an individual to return to a phase, this suited the project even more so since while designing. The Code phase was the next phase which was started when the Requirement and Design phases were considered complete. Code phase was where most of the game design and requirements were implemented to show the physical results and outcomes.

As the idea started to be implemented there were indeed inconsistencies and hiccups that arose. The next phase, the Testing phase was the pinnacle phase in determining the future of the game application. Testing was carried out in order to ensure the code, however much written and whichever part of it that worked, wasn't buggy and nothing was being broken within the code. Any issues that arose, one could visit the previous phases to rectify the situation if one's own ability allowed it.

Last phase was the Integrate phase, this is where all feature work and any separate code can be combined to form the full final product. This is quite advantageous as there were many different features in this game application which were aimed at being designed/coded independently and separate from the main code. After everything passes, the application can then be deployed and would just involve maintenance.

This model is hence considered excellent and perfect for the development of a mathematical game application. It was primarily chosen for its phases which suited the needs of the game. It allows for one to backtrack and rectify and then to move forward again. This way the game application was allowed to be experimented with. It allowed more flexibility in order for the product to reach completion at the aimed time. Any ideas/approach that do not work nor are they suitable with a game application, they can be easily removed or tweaked to suit the performance of the application if there was more time. If there are any ideas in the future which would be considered for implementation, this model would be the perfect tool to do so especially in regards to multiplatform capability.

b. Use Case and Mock Ups

The following shows the features of the application being displayed as a complete use case with the database as an ERD and the mock ups with the correct and finalised user interface of the application.

i. Complete Use Case of the Math Application

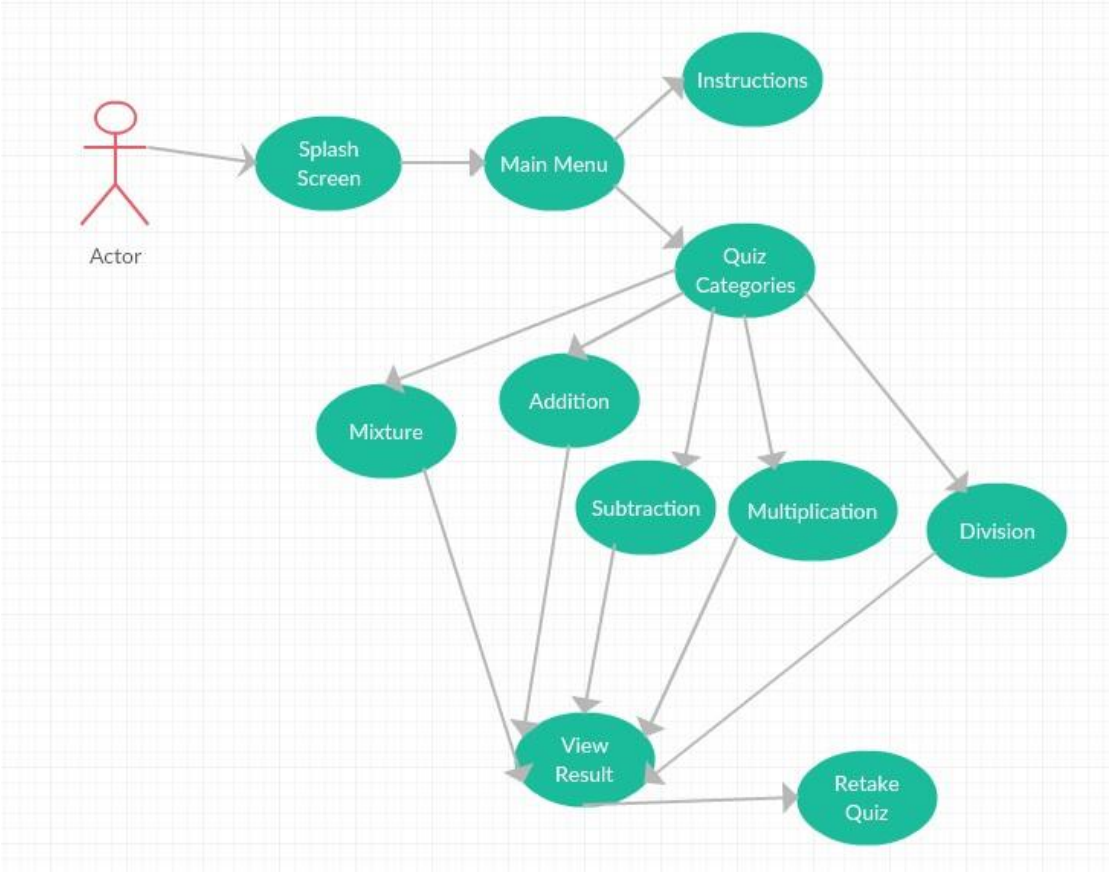


Figure 18 Math Application's Use Case

The above use case shows the overall functionality of the math application and how the application is navigable. The student is greeted with a splash screen and then with the main menu. On the main menu the student will be given the option to access the instructions screen or the quiz categories made available. The instructions screen shows the text on how to use this application. The quiz categories screen will show the various categories the student can choose to play the quiz game. After completion of the quiz, the student will be shown the result screen with the graph and will also have the option of looking at the result individually with each question display with

the correct answer picked and the answer the student picked. Afterwards the student will also be given to retake the quiz and will be brought to the main menu where s/he can select which category to play from.

ii. Application Mock Ups and User Interface

The following shows the rough mock ups that were created during the initial stages of the math application. The images on the right hand side display the finalised user interface of the math application, using Material Design and the correct complimentary colour schemes.

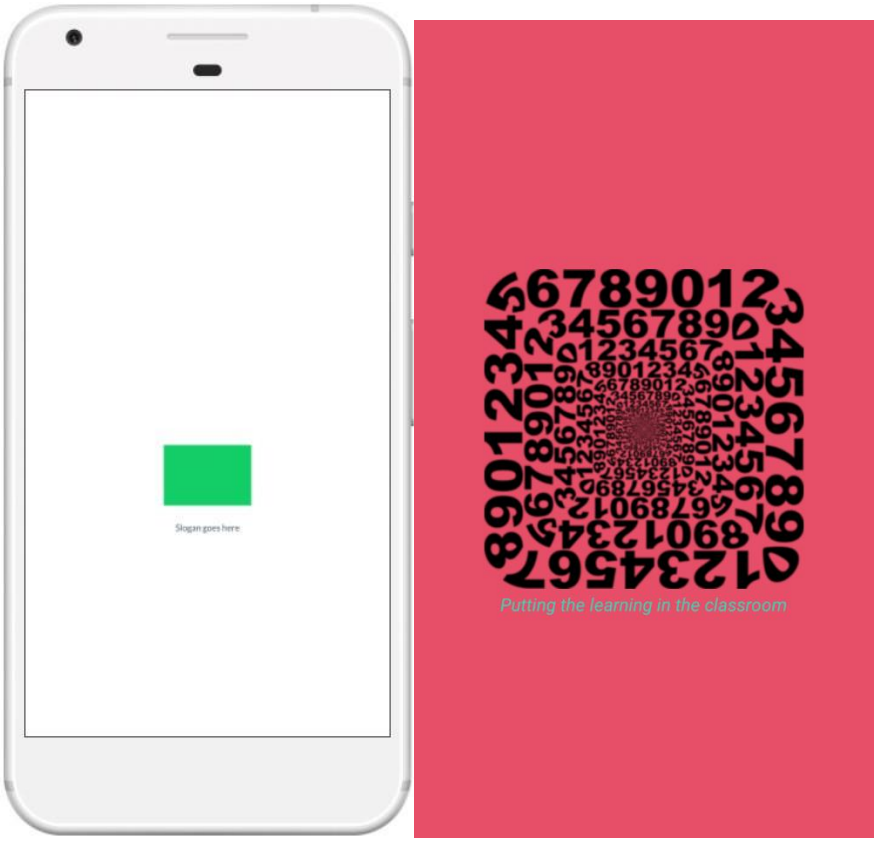


Figure 19 Splash Screen

On the left is the mock up of the splash screen of the math application. On the right is the finalised splash screen of the math application. This shall be displayed when the student opens up the math application for a few seconds before disappearing. The splash screen shows the slogan of the math application.

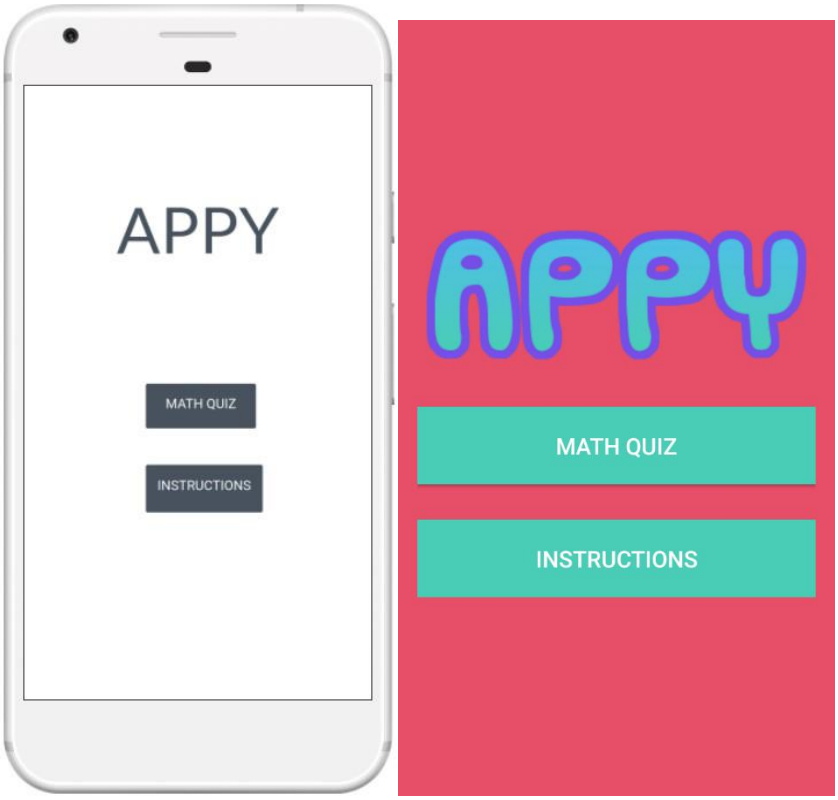


Figure 20 Main Menu Screen

The above shows the mock up of the menu as it was designed in the early stages. The one on the right is the finalised design. It shows the menu screen whereby the student can pick the Math Quiz or can look at the Instructions screen. The bright colours have been used as how they were mentioned beforehand when planning was being carried out for this application.

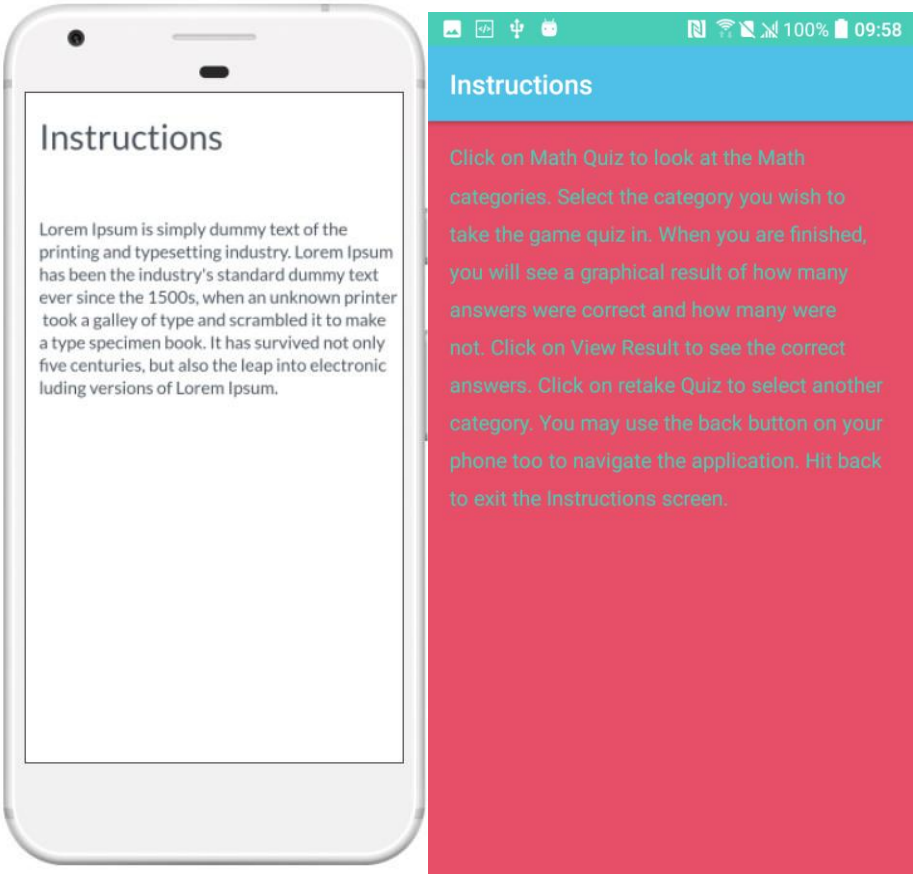


Figure 21 Instructions Screen

The above again shows the mock up and the correct user interface of the math application. The user interface shows the instructions screen and it tells the student on what to expect and how they can navigate the math application. It also lets them know on what they can expect and what the back button on the phone can do.

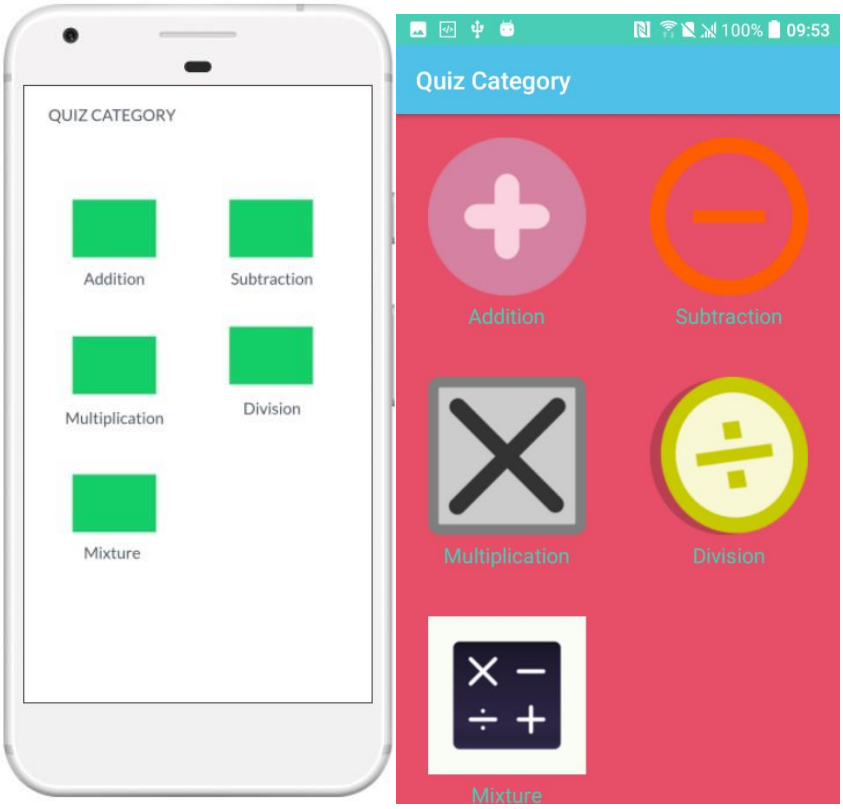


Figure 22 Quiz Category Screen

The mock up looks exactly like the Quiz Category as it was proposed. This screen allows the students to pick the category they wish to take the game quiz in. This screen is accessed when the Math Quiz button on the main screen is hit.

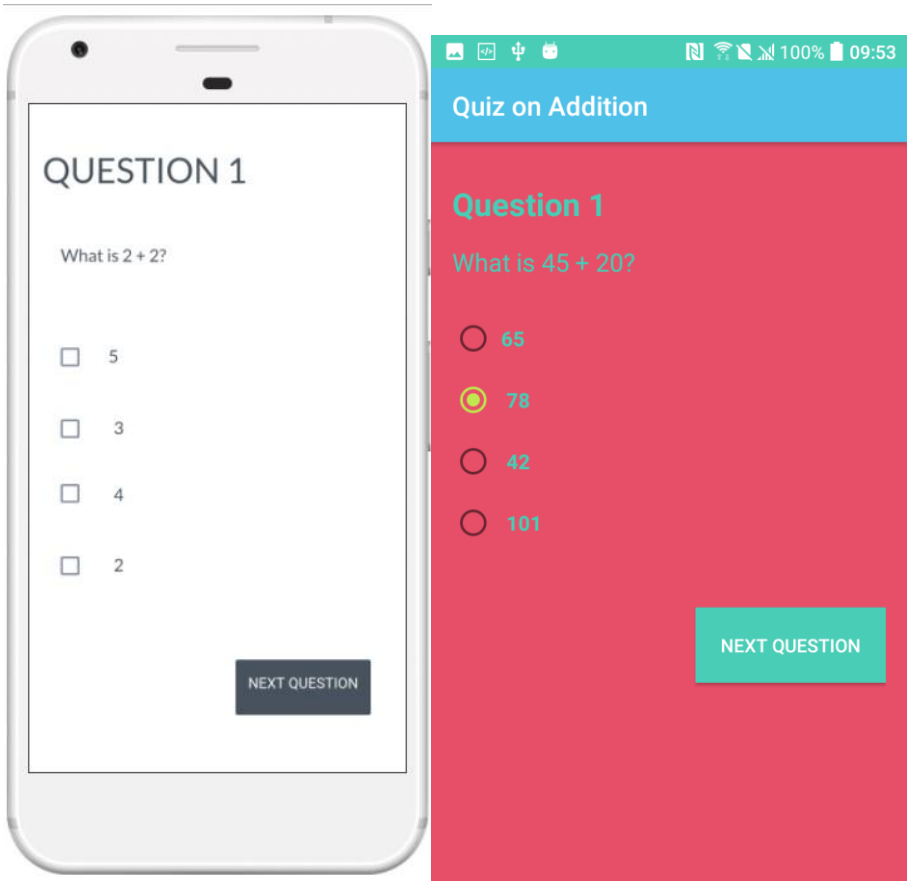


Figure 23 Main Quiz Screen for all categories

This is the quiz screen the student sees when it comes to playing the quiz game. The student can select the answer and then navigate to the next question easily as shown and stated. All the other categories have similar number of questions which are of 10.

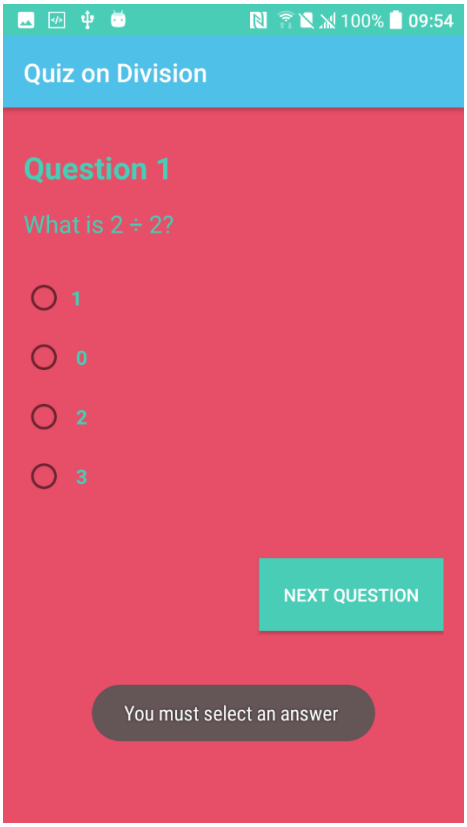


Figure 24 Pop Up on the Quiz Screen

If the answer is not selected then a pop up (toast) will appear telling the student to select an answer before allowing to go to the next question.

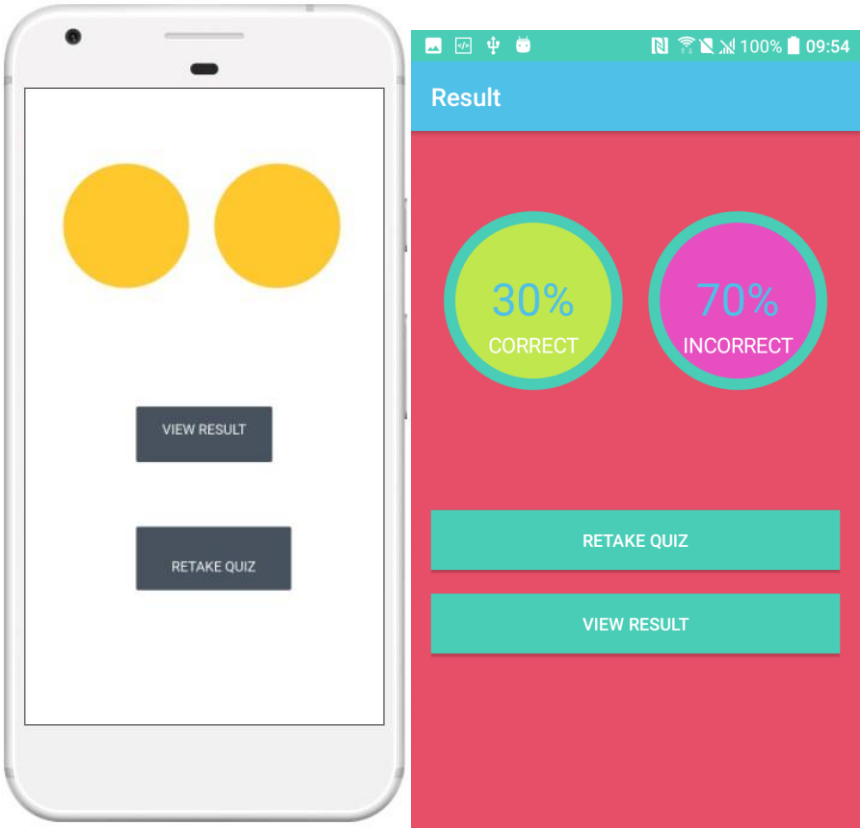


Figure 25 Result Screen showing the Graph

The above shows the result the student will receive after the completion of the quiz. The student will be shown a graph of the result of how much percentage correct the answers were and how much the incorrect answers were. The student would also be allowed to retake the quiz which would lead them to the menu screen and allow them to either pick the same category or different categories or they can decide to view the result in detail.

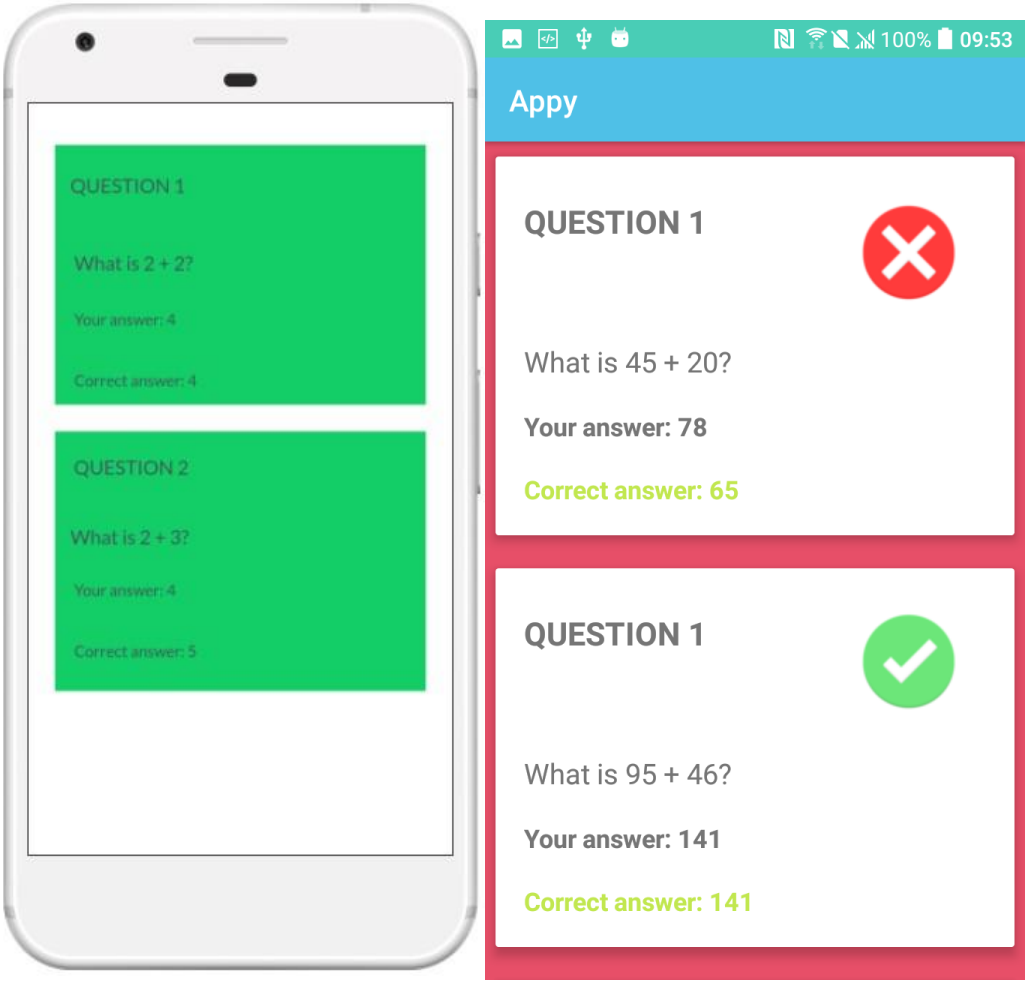


Figure 26 Full Result Screen with the correct answers against the answers chosen

The above mock up shows the card view of the Material Design. The right hand side is the correct interface of the results showing them in a card view. This way the student can see what s/he got correct and what the answer they put in as well. They can also see exactly the correct answer of the question and quiz as well. They can exit the screen using the back button of the phone.

4. Architecture & Development

a. Technical Architecture

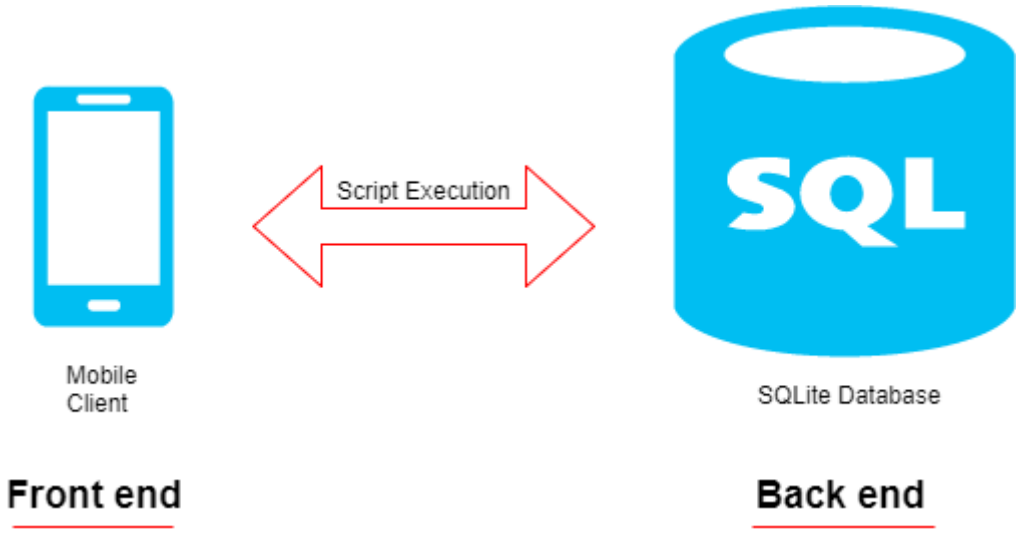
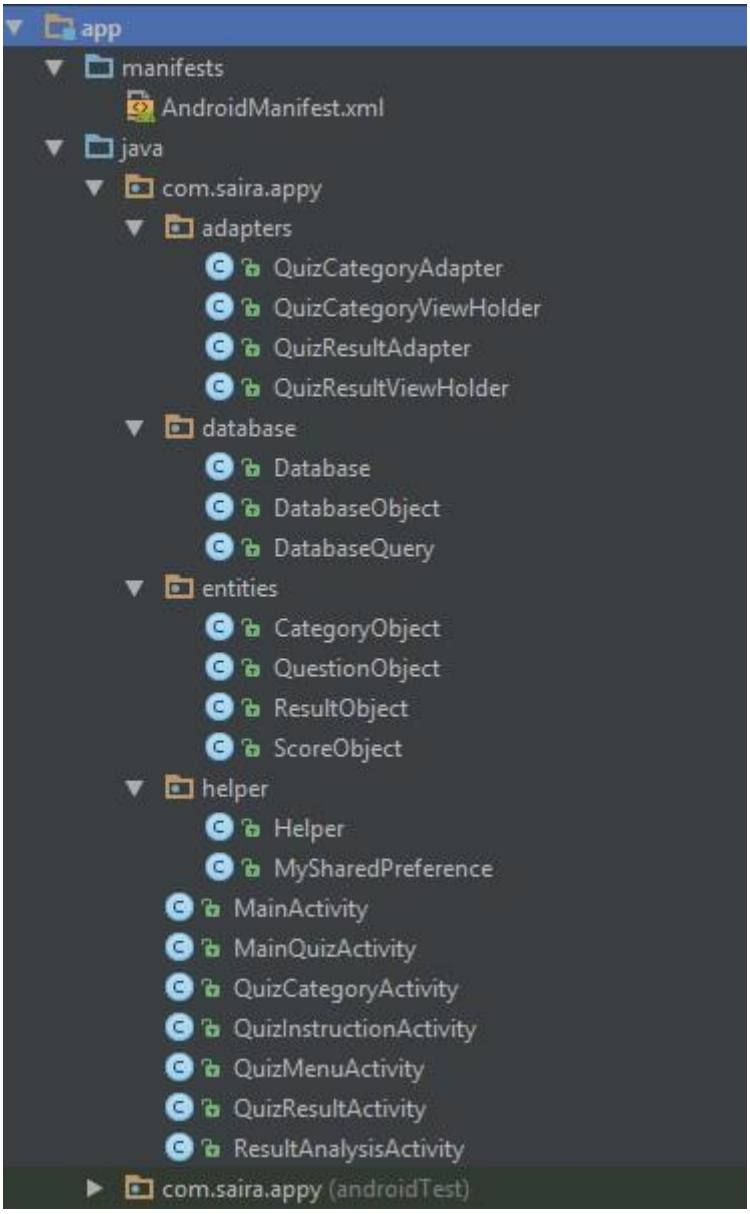


Figure 27 Two tier Architecture

The technical architecture used for this application is a two tier architecture. This was considered appropriate as mobile client would have an SQLite database at the backend. As the math application would come with a database already i.e. math questions already stored and implemented into the application hence this was considered to be the appropriate architecture. The appropriate scripts would be executed when the questions would be displayed and selected and the results would then be displayed.

b. Development and Code Implementation

The following shows the explanation of chosen code for understanding of the math application. The list is also displayed here of all the code and assets that have been put into the math application:



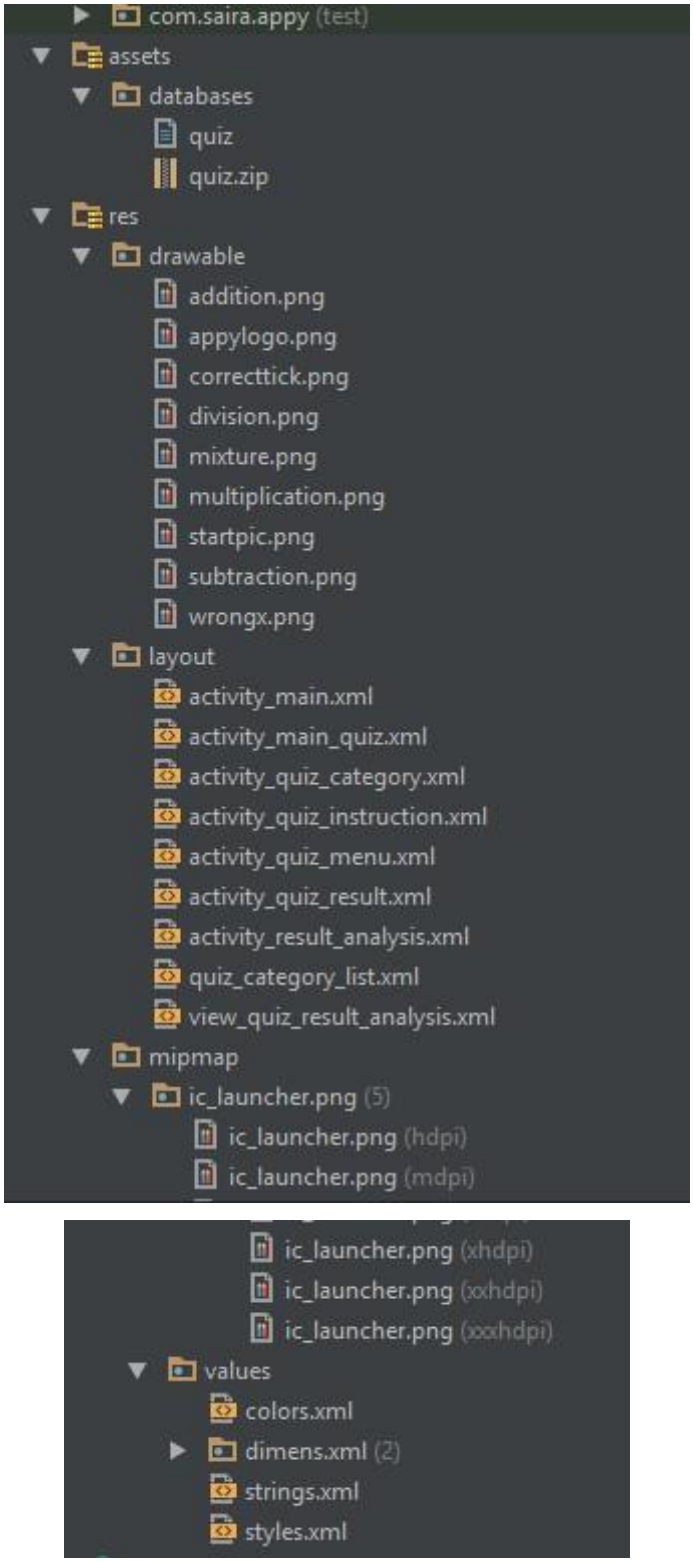


Figure 28 List of the source code

All the adapter files follow the RecyclerView layout. The QuizCategoryAdapter is used to retrieve and display all the different quiz categories available while the QuizCategoryViewHolder holds the name and image of the category. The

QuizResultAdapter will interact with deal with displaying the result screen where the student can see what questions were correct/incorrect. The various images (tick/cross) will be displayed beside the questions to indicate if they were right or not. The QuizResultViewHolder will retrieve the visual aspects of the Result screen i.e. images for ticks/crosses and the layout of the screen.

The database folder holds the SQLite database of the application. The SQLiteAssetHelper is an extension which manipulates and keeps version control of the database which will be exported with the application.

The entities folder holds the various objects which will be used for the result and the score of the student. It shall also be used to display the category and the question to the student.

The java files deal with the running and the execution of the application and the XML files deal with the user interface and the display of the app. The drawables all contain images which shall be used in the application.

The following shows chosen code of the math application which include most of the logic:

MainActivity – this file contains the logic to display the splash screen of the application with the app's slogan. It shall be displayed for a few seconds and contains no title.

```
public class MainActivity extends AppCompatActivity {

    private static final String TAG = MainActivity.class.getSimpleName();

    private final int SPLASH_DISPLAY_LENGTH = 3000;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        requestWindowFeature(Window.FEATURE_NO_TITLE);
        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN, WindowManager.LayoutParams.FLAG_FULLSCREEN);
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);

        ActionBar actionBar = getSupportActionBar();
        if(null != actionBar){
            actionBar.hide();
        }

        new Handler().postDelayed(new Runnable(){
            @Override
            public void run(){
                Intent startActivityIntent = new Intent(MainActivity.this, QuizMenuActivity.class);
                startActivity(startActivityIntent);
                MainActivity.this.finish();
            }
        }, SPLASH_DISPLAY_LENGTH);
    }
}
```

MainQuizActivity – this is the quiz screen where the quiz questions are displayed. This shows the layout of the quiz screen with the options and the questions with the question numbers. This also deals with logic of calculating the score and the result of the quiz and checking if there are any more questions left or not. If not, it finishes. If yes, it displays the next question. It also compares the questions answered correct against the questions incorrectly.

```
public class MainQuizActivity extends AppCompatActivity {

    private static final String TAG = MainActivity.class.getSimpleName();

    private TextView questionNumber, question;

    private RadioGroup radioGroup;

    private RadioButton optionOne, optionTwo, optionThree, optionFour;

    private List<QuestionObject> quizObject;

    private QuestionObject allQuestions;

    private int totalQuizCount;

    private int questionCount = 0;

    private ScoreObject mScore;

    private Button nextQuestionButton;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main_quiz);

        String categoryName = getIntent().getExtras().getString("QUIZ_CATEGORY_NAME");
        if(!TextUtils.isEmpty(categoryName)){
            setTitle("Quiz on " + categoryName);
        }

        mScore = new ScoreObject();
```

```
61          questionNumber = (TextView)findViewById(R.id.question_number);
62          question = (TextView)findViewById(R.id.question);
63
64          radioGroup = (RadioGroup)findViewById(R.id.answer_options);
65
66          optionOne = (RadioButton)findViewById(R.id.answer_one);
67          optionTwo = (RadioButton)findViewById(R.id.answer_two);
68          optionThree = (RadioButton)findViewById(R.id.answer_three);
69          optionFour = (RadioButton)findViewById(R.id.answer_four);
70
71          int quizCategoryId = getIntent().getExtras().getInt("QUIZ_CATEGORY_ID");
72          DatabaseQuery query = new DatabaseQuery(MainQuizActivity.this);
73
74          quizObject = query.getQuizQuestionsById(quizCategoryId);
75
76          nextQuestionButton = (Button)findViewById(R.id.next_quiz);
77
78          if(quizObject.size() > 0){
79              totalQuizCount = quizObject.size();
80              allQuestions = quizObject.get(questionCount);
81              displayQuizQuestions();
82
83              assert nextQuestionButton != null;
84              nextQuestionButton.setOnClickListener(new View.OnClickListener() {
85                  @Override
86                  public void onClick(View view) {
87                      int radioButtonId = radioGroup.getCheckedRadioButtonId();
88                      String userSelectedAnswer = selectedAnswerOption(radioButtonId);
89
90                      if(userSelectedAnswer.equals("")){
91                          Toast.makeText(MainQuizActivity.this, "You must select an answer " + userSelectedAnswer, Toast.LENGTH_LONG).show()
92                      }else{
93                          //check for the correct answer
94                          Log.d(TAG, "Match answers " + allQuestions.getAnswer() + " select " + userSelectedAnswer);
95                          if(allQuestions.getAnswer().trim().equals(userSelectedAnswer.trim())){
```

```
    //set new score
    mScore.setScore(1);
    //set the result
    mScore.addNewQuizResult(new ResultObject(""+allQuestions.getId(), allQuestions.getQuestion(), userSelectedAnswer, allQuestions.getAnswer(), true
}else{
    mScore.addNewQuizResult(new ResultObject(""+allQuestions.getId(), allQuestions.getQuestion(), userSelectedAnswer, allQuestions.getAnswer(), fals
}
Log.d(TAG, "Quiz Result " + mScore.getQuizResultObject().size());
questionCount++;

// check if there is more question
if(questionCount >= totalQuizCount){
    // Quiz over
    Intent quizOverIntent = new Intent(MainQuizActivity.this, QuizResultActivity.class);

    GsonBuilder builder = new GsonBuilder();
    Gson gson = builder.create();
    final String scoreString = gson.toJson(mScore);
    quizOverIntent.putExtra("RESULT_OBJECT", scoreString);

    double percentageScore = (mScore.getScore() * 100) / totalQuizCount ;
    quizOverIntent.putExtra("TOTAL_SCORE", String.valueOf(percentageScore));

    // compare score and save
    MySharedPreference sharedPreference = new MySharedPreference(MainQuizActivity.this);
    Double mDouble = new Double(percentageScore);
    int presentScore = mDouble.intValue();
    if(!sharedPreference.isHighestScore(presentScore)){
        sharedPreference.saveQuizHighestQuizScore(presentScore);
    }

    startActivity(quizOverIntent);

}else{
    // display new questions
```

```
                            allQuestions = quizObject.get(questionCount);
                            displayQuizQuestions();
                        }
                    }
                }
            });
        }else{
            optionOne.setVisibility(View.GONE);
            optionTwo.setVisibility(View.GONE);
            optionThree.setVisibility(View.GONE);
            optionFour.setVisibility(View.GONE);
            nextQuestionButton.setVisibility(View.GONE);
            Toast.makeText(MainQuizActivity.this, getString(R.string.no_quiz_in_category), Toast.LENGTH_LONG).show();
        }
    }

    private void displayQuizQuestions(){
        if(allQuestions != null){
            unsetRadioButton();
            int currentQuestion = questionCount + 1;
            questionNumber.setText("Question " + currentQuestion);
            question.setText(allQuestions.getQuestion());

            String answerOption = allQuestions.getOptions();
            String[] allAnswerOptions = allQuestions.convertOptionsToStringArray(answerOption);

            optionOne.setText(allAnswerOptions[0]);
            optionTwo.setText(allAnswerOptions[1]);
            optionThree.setText(allAnswerOptions[2]);
            optionFour.setText(allAnswerOptions[3]);
        }
    }

    private void unsetRadioButton(){
        optionOne.setChecked(false);
        optionOne.setChecked(false);
        optionTwo.setChecked(false);
        optionThree.setChecked(false);
        optionFour.setChecked(false);
    }

    private String selectedAnswerOption(int id){
        String textContent = "";
        if(id == R.id.answer_one){
            textContent = optionOne.getText().toString();
        }
        if(id == R.id.answer_two){
            textContent = optionTwo.getText().toString();
        }
        if(id == R.id.answer_three){
            textContent = optionThree.getText().toString();
        }
        if(id == R.id.answer_four){
            textContent = optionFour.getText().toString();
        }
        return textContent;
    }
}
```

QuizCategoryActivity – this displays the categories of the math quiz i.e. addition, subtraction etc.

```
public class QuizCategoryActivity extends AppCompatActivity {

    private static final String TAG = QuizCategoryActivity.class.getSimpleName();

    private RecyclerView quizRecyclerView;

    private static int quizType = 1;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_quiz_category);

        setTitle(getString(R.string.quiz_category));

        DatabaseQuery dbQuery = new DatabaseQuery(QuizCategoryActivity.this);
        List<CategoryObject> categoryData = dbQuery.getAllQuizCategory(0);

        quizRecyclerView = (RecyclerView)findViewById(R.id.quiz_category);
        GridLayoutManager mGrid = new GridLayoutManager(this, 2);
        quizRecyclerView.setLayoutManager(mGrid);
        quizRecyclerView.setHasFixedSize(true);

        QuizCategoryAdapter mAdapter = new QuizCategoryAdapter(QuizCategoryActivity.this, categoryData, quizType);
        quizRecyclerView.setAdapter(mAdapter);
    }

    @Override
    public void onBackPressed() {
        Intent backIntent = new Intent(QuizCategoryActivity.this, QuizMenuActivity.class);
        startActivity(backIntent);
    }
}
```

QuizInstructionActivity – this class is used to display the Instructions screen.

```
public class QuizInstructionActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_quiz_instruction);

        setTitle(getString(R.string.quiz_information_notice));

        TextView instructionText = (TextView)findViewById(R.id.quiz_instruction);
        instructionText.setText(Helper.instruction);
    }
}
```

QuizMenuActivity – this displays the main menu screen where the Instructions button and the Quiz button are displayed. The student picks the correct button depending on what they intend to do.

```
public class QuizMenuActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        requestWindowFeature(Window.FEATURE_NO_TITLE);
        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN, WindowManager.LayoutParams.FLAG_FULLSCREEN);
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_quiz_menu);

        setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);

        ActionBar actionBar = getSupportActionBar();
        if(null != actionBar){
            actionBar.hide();
        }

        Button selectQuiz = (Button)findViewById(R.id.text_quiz_button);
        assert selectQuiz != null;
        selectQuiz.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent quizCategoryIntent = new Intent(QuizMenuActivity.this, QuizCategoryActivity.class);
                quizCategoryIntent.putExtra("QUIZ_TYPE", 1);
                startActivity(quizCategoryIntent);
            }
        });

        Button quizInstruction = (Button)findViewById(R.id.quiz_instruction_button);
        assert quizInstruction != null;
        quizInstruction.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent instructionIntent = new Intent(QuizMenuActivity.this, QuizInstructionActivity.class);
                startActivity(instructionIntent);
            }
```

QuizResultActivity – this is the class that shows the result screen whereby the graph is shown with circles. The userPassScore shows the CORRECT percentage of questions in the circle and the userFailedScore shows the INCORRECT percentage of questions in the circle as a graph.

```
public class QuizResultActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_quiz_result);

        setTitle(getString(R.string.quiz_result));

        final String resultString = getIntent().getExtras().getString("RESULT_OBJECT");
        String userScore = getIntent().getExtras().getString("TOTAL_SCORE");

        double passResult = Double.parseDouble(userScore);
        int userPassMark = new Double(passResult).intValue();

        CircleView userPassScore = (CircleView)findViewById(R.id.pass);
        CircleView userFailedScore = (CircleView)findViewById(R.id.fail);

        int failedScore = 100 - userPassMark;

        userPassScore.setTitleText(String.valueOf(userPassMark) + "%");
        userFailedScore.setTitleText(String.valueOf(failedScore) + "%");

        Button retakeQuizButton = (Button)findViewById(R.id.retake_quiz);
        assert retakeQuizButton != null;
        retakeQuizButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent retakeQuizIntent = new Intent(QuizResultActivity.this, QuizMenuActivity.class);
                startActivity(retakeQuizIntent);
            }
        });

        Button viewQuizResultButton = (Button)findViewById(R.id.view_result);
        assert viewQuizResultButton != null;
        viewQuizResultButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent resultIntent = new Intent(QuizResultActivity.this, ResultAnalysisActivity.class);
                resultIntent.putExtra("RESULT", resultString);
                startActivity(resultIntent);
            }
        });
          }
       });
    }

    @Override
    public void onBackPressed() {
    }
}
```

ResultAnalysisActivity – this class is used to display the result analysis and to display the result score of the game.

```
public class ResultAnalysisActivity extends AppCompatActivity {

    private static final String TAG = ResultAnalysisActivity.class.getSimpleName();

    private RecyclerView resultRecyclerView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_result_analysis);

        resultRecyclerView = (RecyclerView)findViewById(R.id.quiz_result_analysis);
        LinearLayoutManager linearLayoutManager = new LinearLayoutManager(ResultAnalysisActivity.this);
        resultRecyclerView.setLayoutManager(linearLayoutManager);
        resultRecyclerView.setHasFixedSize(true);

        String mQuizResult = getIntent().getExtras().getString("RESULT");
        GsonBuilder builder = new GsonBuilder();
        Gson gson = builder.create();

        ScoreObject storedResult = gson.fromJson(mQuizResult, ScoreObject.class);
        List<ResultObject> getStoredResults = storedResult.getQuizResultObject();
        if(getStoredResults != null){
            Log.d(TAG, " Result score " + getStoredResults.size());
        }

        QuizResultAdapter resultAdapter = new QuizResultAdapter(ResultAnalysisActivity.this, getStoredResults);
        resultRecyclerView.setAdapter(resultAdapter);
    }
}
```

5. System Validation

Software testing is extremely crucial in regards to development to ensure that the system is delivered with the functionality that it is intended for. A report of the following test cases and the outcome of the proposed system will be shown below.

a. Black Box Testing

Testing phase involved looking at all the components of this system and making sure everything was working properly. Black box testing was used for this feat as black box testing allows for the test of the overall functionality of the project more achievable and hence black box testing has been given more priority. The following shows the various tests carried out on the application:

i. Splash Screen

Test Name:	Splash Screen is displayed
Steps:	Clicking on the application icon
Expected Outcome:	Splash Screen is displayed
Actual Outcome:	Splash Screen is displayed
Result:	Passed

ii. Categories and Category Images

Test Name:	Quiz Categories are retrieved from the database and are displayed with the correct images from drawables
Steps:	1. Clicking on the application icon 2. Click on the Math Quiz Button
Expected Outcome:	Correct images with the correct categories are displayed
Actual Outcome:	Correct images with the correct categories are displayed
Result:	Passed

iii. Categorical Questions from the Database

Test Name:	Correct questions are retrieved for the corresponding categories from the database
Steps:	1. Clicking on the application icon 2. Click on the Math Quiz Button 3. Click on the category of choice
Expected Outcome:	Correct questions are displayed for the category
Actual Outcome:	Correct questions are displayed for the category
Result:	Passed

iv. Graph shows Incorrect and Correct Answers

Test Name:	The Result Screen shows the graphs for the Incorrect and the Correct answers
Steps:	1. Clicking on the application icon 2. Click on the Math Quiz Button 3. Click on the desired category 4. Complete the quiz
Expected Outcome:	Graphs are displayed for the Correct and Incorrect answers
Actual Outcome:	Graphs are displayed for the Correct and Incorrect answers
Result:	Passed

v. “Your Answer” and “Correct Answer”

Test Name:	Result Analysis shows the “Your Answer” and the “Correct Answer” on the screen to the student
Steps:	1. Clicking on the application icon 2. Click on the Math Quiz Button 3. Click on the desired category 4. Complete the quiz 5. Click on View Results
Expected Outcome:	Your Answer and Correct Answer display the correct values
Actual Outcome:	Your Answer and Correct Answer display the correct values

Result:	Passed

vi. Question Numbers in the Result Analysis

Test Name:	Result Analysis shows the Question numbers correctly
Steps:	1. Clicking on the application icon 2. Click on the Math Quiz Button 3. Click on the desired category 4. Complete the quiz 5. Click on View Results
Expected Outcome:	The numbers of the questions should be displayed correctly
Actual Outcome:	The numbers do not correspond to the question number
Result:	Failed

Only one test failed which was to show that in the result analysis that the question number is the same as the question answered, yet for some reason this problem was not solvable.

b. User Testing

The math application was tested by Ms. Maria Kate Carberry, a secondary school Math teacher. The aim of the meeting with Ms. Carberry was to display the application to her and to receive feedback. She tested each component of the application from the perspective of a teacher and a student. The feedback from the teacher was very positive and she was delighted with the user interface. She found the clear and concise design to be very neat and effective and found the questions to be suitable for primacy school students. She also found the questions asked to be satisfactory and the visual graph to be very effective for primary school students.

A student was also given the application who is a family friend. She is 13 years old and went through the application by herself. She found the application to be fun and enjoyable and very straightforward. She also found the graph and the viewing of the results extremely different from the other math applications she had experience with.

Overall the application has been found to be quite engaging and effective from the point of view of the teacher and the student. The application was considered extremely clear and straightforward and the graph is a new and gorgeous functionality which the users found to be different from the other applications they have knowledge of.

c. Demonstration

The demonstration of the app can be found on this link: https://youtu.be/Bvkc9eb4JLU

6. Project Plan

a. Overview

The project began in October 2016 and was completed in April 2017 yet it needed to still be continued until August 2017. In October, the project's idea was developed with the supervisor and it was submitted with the functionality and the background research. As it was approved, many individuals were contacted from DIT in regards to the future and requirements of the project. A school had been visited and there was a meeting with a teacher, Ms. Maria Kate Carberry, to understand what the math application requirements and functionalities can be.

As November approached, the application's functionality and the requirements of the end user were determined. This month was heavily spent on research and developing mock ups and use cases. These initial mock ups and use cases were also used as guidelines on what the application would eventually end up looking like and what changes can be included. These mock ups and use cases would be kept and changed and edited. These were shown to the supervisor and feedback was received on whether these were considered suitable or were there additions that needed to be made.

December approached and it was time to submit the interim report. A short presentation, where all the findings were brought together, was presented in front of the supervisor and the second reader. The feedback from this was very useful as this paved a path for the future work that needed to be put into the application.

During January 2017 and onwards, the right technologies were assembled and first of all tested to make sure that they are suitable for the project at hand. Android Studio development started and code was written in Java and XML. The user interface started to be designed and the logic started to be coded. This was the time when the implementation process was initially started.

February and March were spent on the developing the functionality of the project. A lot of time was spent on developing the features and correcting any bugs that arose. The final submission and meeting in April really aided in continuing the development of this project until August. During that meeting, it was suggested to not include a timer at all, and this was a key point that was taken into consideration for this project.

During June and July, most of the time was spent on correcting the features and the user interface of the project. This time was highly spent on correcting and fixing bugs and implementing all the features necessary to fully complete this project altogether. During the start of August, testing was carried out, especially Black Box Testing to insure all the components and the features handled by the application is satisfactory and does not fail. Ms. User testing was also carried out to make sure that the application appealed to the target audience. A meeting was held with the math teacher Ms. Maria Kate Carberry to make sure the math application was sufficient for primary school students. Feedback received from this was very positive. Another meeting was held with a school student to identify the feelings of the student for this application. Feedback from the student was also positive and encouraging.

During August was also the time the proposed application was finalised and the final report was fully completed for submission.

b. Future Suggestions

i. Fixing Bug

All of the features of the application work perfectly. There is one bug that exists in the application which does not display the correct number of the questions when checking the full result. This is a bug which, no matter how much effort was put in, did not seem to be correctable. If more time was given, this bug would be the one thing that would definitely be targeted in the future.

7. Conclusion

The research stage was extremely crucial and a vast amount of research was performed for the user interface and the end user. Many technologies had to be researched as well as the alternative existing solutions. While researching, it was important to understand the needs and requirements for the end user and the technologies that could satisfy the requirements. The designing stage was when the correct RAP Waterfall Methodology was selected. The use cases and the mock ups of the math application were designed and the user interface was fully finalised. This stage brought together the full understanding of the math application and really brought together the actual understanding and the application of the Waterfall methodology.

The development stage was when the features were implemented and developed. The architecture of the application was clarified and the features were then coded and implemented. As the testing and the validation stage started, the math application was tested using various test cases. The math application was also tested by a real teacher and was provided with feedback. The application was also tested by the student and the appropriate feedback received was taken into consideration. The testing phase was considered highly important as it determined that the application was effective and performed the aim it was intended for.

a. Objective

The main aim of this project was to develop a math application for primary school students between the ages of eight and twelve. The current math application generates math quizzes for various categories, from addition, subtraction, multiplication, division, and micture. As a quiz is completed, a result is generated. The student can view a graphic representation of the CORRECT and INCORRECT answers, in the form of a percentage, for their result. If they wish to see the full result, they can see that too. The result will show the question, the answer picked by the student and the correct answer of the question with an image of a tick or a cross to symbolise if it is correct or not. The application’s interface design was also designed using Material

Design and the colour schemes were taken into account in order to produce a suitable math application for primary school students i.e. the end user.

The above was achieved by immense research on what the requirement would suit the students and many technologies that can support the project idea. Feedback from the teacher and the student were vital in regards to the completion of this project.

b. Learning

Much was learning during the execution of this project. The learning curve has been vast and tackling the end user requirements for this project has been a great journey. Using a methodology to reach the completion of this project has been a great learning experience in regards to game development, application development, and software development. The importance of feedback and the importance of time management are key for a product to be fully finalised and completed.

The development of this application also brought about the needs of a primary school student in regards to math into light and the various teaching techniques teachers use in the classroom. The various technologies used and researched also greatly heightened the knowledge of the technologies available and research skills. As the project reached completion, programming skills were greatly improved and communication skills.

Overall, the development of this application has been a very cherish able experience and it has brought about many changes in the everyday software development viewpoint.

8. Bibliography

- www.tutorialspoint.com. (2016). Android Studio. [online] Available at: https://www.tutorialspoint.com/android/android_studio.htm [Accessed 6 Nov. 2016].
- Bransford, J., Cocking, R. and Brown, A. (2000). How People Learn: Brain, Mind, Experience, and School. Expanded Edition. 1st ed. National Academies Press, p.79.
- Rooney, P., (2007). Students @ play: serious games for learning in higher education. [online] Dublin: Dublin Institute of Technology. Available at: http://arrow.dit.ie/cgi/viewcontent.cgi?article=1026&context=scschcomcon [Accessed 6 Nov. 2016].
- Educationworld.com. (2013). Education World: Why Use Games in Your Classroom? [online] Available at: http://www.educationworld.com/a_curr/reasons-to-play-games-in-the-classroom.shtml [Accessed 6 Nov. 2016].
- ES, D. (2016). Displaying Graphics with OpenGL ES | Android Developers. [online] Developer.android.com. Available at: https://developer.android.com/training/graphics/opengl/index.html [Accessed 30 Nov. 2016].
- www.idc.com. (2016). IDC: Smartphone OS Market Share. [online] Available at: https://www.idc.com/prodserv/smartphone-os-market-share.jsp [Accessed 28 Nov. 2016].
- Google App Play Store. (2016). Math Puzzle - Brain Trainer. [online] Available at: https://play.google.com/store/apps/details?id=com.peafone.mathpuzzle&hl=en [Accessed 29 Oct. 2016].
- Google App Play Store. (2016). Math Workout. [online] Available at: https://play.google.com/store/apps/details?id=com.akbur.mathsworkout [Accessed 29 Oct. 2016].
- Rocha, M., Tangney, B. and Dondio, P. (2016). A Survey of Digital Games Used in Irish Schools: the Drill and Practice has Turned Digital. [online] Dublin: Dublin Institute of Technology. Available at:

http://arrow.dit.ie/cgi/viewcontent.cgi?article=1204&context=scschcomcon [Accessed 2 Nov. 2016].

- Studio, M. (2016). Meet Android Studio | Android Studio. [online] Developer.android.com. Available at: https://developer.android.com/studio/intro/index.html [Accessed 6 Nov. 2016].
- Swarmconnect.com. (2015). Swarm - Leaderboards, Achievements, Social Features, and Monetization for Android Mobile Games and Apps. [online] Available at: http://swarmconnect.com/ [Accessed 1 Nov. 2016].
- Kahoot! Journal. (2013). What is a Kahoot?. [online] Available at: http://blog.getkahoot.com/post/58906886260/what-is-a-kahoot [Accessed 28 Oct. 2016].
- Git-scm.com. (2016). Git. [online] Available at: https://git-scm.com/ [Accessed 15 Nov. 2016].
- Mackay, R.F. (2013). Using games as an educational tool provides opportunities for deeper learning, panelists at Stanford event say. [online] Stanford News. Available at: http://news.stanford.edu/2013/03/01/games-education-tool-030113/ [Accessed 23 Nov. 2016].
- Colgan, A. and Colgan, A. (2017). 8 Things Every Educational Game Developer Needs to Know. [online] Leap Motion Blog. Available at: http://blog.leapmotion.com/8-things-every-educational-game-developer-needs-know/ [Accessed 2 Mar. 2017].
- Manifest, A. (2017). App Manifest | Android Developers. [online] Developer.android.com. Available at: https://developer.android.com/guide/topics/manifest/manifest-intro.html [Accessed 23 Feb. 2017].
- Developer.android.com. (2017). Layouts | Android Developers. [online] Available at: https://developer.android.com/guide/topics/ui/declaring-layout.html [Accessed 23 Feb. 2017].
- venturehire, (2017). Advantages of learning android programming in java. [online] Slideshare.net. Available at: https://www.slideshare.net/venturehire/advantages-of-learning-android-programming-in-java [Accessed 23 Feb. 2017].

- Programming, X. (2017). XML for Android Programming. [online] Stackoverflow.com. Available at: http://stackoverflow.com/questions/8639074/xml-for-android-programming [Accessed 23 Feb. 2017].
- Lifehacker.com. (2017). Cite a Website - Cite This For Me. [online] Available at: http://lifehacker.com/why-google-play-services-are-now-more-important-than-an-975970197 [Accessed 28 Feb. 2017].
- Science, L. (2017). Brain's Link Between Sounds, Smells and Memory Revealed. [online] Live Science. Available at: http://www.livescience.com/8426-brain-link-sounds-smells-memory-revealed.html [Accessed 23 Feb. 2017].
- Bensound.com. (2017). Royalty Free Music by Bensound | Creative Commons Music. [online] Available at: http://www.bensound.com/ [Accessed 5 Mar. 2017].
- Neuroscience News. (2017). Neuroscience of Music – How Music Enhances Learning Through Neuroplasticity. [online] Available at: http://neurosciencenews.com/neuroscience-music-enchances-learning-neuroplasticity/ [Accessed 2 Mar. 2017].
- Pixabay.com. (2017). Free Images - Pixabay. [online] Available at: https://pixabay.com/ [Accessed 10 Jan. 2017].
- Envato Tuts+. (2017). Free How-To Tutorials & Online Courses by Envato Tuts+. [online] Available at: https://tutsplus.com/ [Accessed 11 Feb. 2017].
- Viral Android – Tutorials, Examples, UX/UI Design. (2017). Viral Android – Tutorials, Examples, UX/UI Design. [online] Available at: http://www.viralandroid.com/ [Accessed 22 Feb. 2017].
- Grec, V. (2017). Introduction to Android Espresso. [online] Android Research Blog. Available at: https://androidresearch.wordpress.com/2015/04/04/an-introduction-to-espresso/ [Accessed 14 Mar. 2017].
- Delaney, S. (2017). Learning to Teach Primary School Mathematics. [online] Available at: http://seandelaney.com/wp-content/uploads/2011/09/Learning-to-Teach-Primary-School-Math-SCoTENS-30-9-11.pdf [Accessed 18 Jun. 2017].

- Learning Styles. (2017) The Seven Learning Styles. [online] Available at: https://www.learning-styles-online.com/overview/ [Accessed 21 Jun. 2017].
- Wong, E. (2017) User Interface Design Guidelines: 10 Rules of Thumb. [online] Available at: https://www.interaction-design.org/literature/article/user-interface-design-guidelines-10-rules-of-thumb [Accessed at 21 Jun. 2017].
- Android Developers. (2017) Android Design Principles. [online] Available at: http://www.designprinciplesftw.com/collections/android-design-principles [Accessed 29 Jun. 2017].
- Cooltext. (2017). Cooltext. [online]. Available at: https://cooltext.com/ [Accessed 21 Jun. 2017].
- CircleView. (2017). CircleView. [online]. Available at: https://github.com/pavlospt/CircleView [Accessed 11 Jul. 2017].
- SDP. (2017). SDP. [online]. Available at: https://github.com/intuit/sdp [Accessed 11 Jul. 2017].
- DB Browser for SQLite. (2017). DB Browser for SQLite. [online]. Available at: http://sqlitebrowser.org/ [Accessed 11 Jul. 2017].
- Colour Calculator. (2017). Colour Calculator. [online]. Available at: https://www.sessions.edu/color-calculator/ [Accessed 11 Jul. 2017].
- Android. (2017). Material Design. [online]. Available at: https://developer.android.com/design/material/index.html [Accessed 11 Jul. 2017].
- Android SQLite Asset Helper. (2017). Android SQLite Asset Helper. [online]. Available at: https://github.com/jgilfelt/android-sqlite-asset-helper [Accessed 18 Jul. 2017].