



Primary Mathletes – Primary School Mathematics Teaching Aid

Interim Report

**DT282
BSc in Computer Science International**

Paul Davis

C16324311

Cindy Liu

**School of Computer Science
Technological University, Dublin**

8th December 2019

Abstract

Area of study:

This project will be focusing on improving primary school teaching through the use of mobile apps. The project will consist of an app that is going to be developed in Kivy, using python programming, that will help primary school teachers by acting as a teaching aid for their students to improve their mathematics abilities.

The problem being tackled:

One issue that commonly occurs in classrooms, especially ones with a lot of students, is that no matter how slow the teacher goes, eventually they will have to move on so that they can appropriately tackle all elements of the curriculum before the end of the year. This project aims to aid the teacher and student by giving them the ability to teach the student more of the curriculum in a fun and interactive way, inside and outside of the classroom so that even if they fall behind there is a way of catching up without taking up valuable class time.

The app will hopefully improve the grades of students who may be struggling with maths in school and prevent them from giving up once they fall behind.

How to tackle the problem:

During the implementation of this project, research into the primary school curriculum through books and conversing with some teachers if possible to get an idea of areas where students are struggling with the topics that come up and see if it is possible to make it more accessible to a wider range of students.

Implementation of the solution:

Implementation of this project will be completed through python programming using Kivy to make a cross-platform application that will use the research conducted through books and online resources to create a fun, interactive and informative application that will aid both students and teachers in primary schools.

How to evaluate the project:

The project will, hopefully, be evaluated by getting in contact with a primary school teacher and asking them to use the app during their maths teaching/learning in-class and seeing if there is any improvement with the use of the app. If this isn't possible, an attempt will be made to get a small group of primary school students to participate in a study group to try and evaluate its usefulness.

Declaration

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

Signed:

Paul Davis

Date

Table of Contents

Table of Figures.....	5
1. Introduction	6
1.1. Project Background.....	6
1.2. Project Description.....	6
1.3. Project Aims and Objectives	9
1.4. Project Scope	9
1.5. Thesis Roadmap	9
2. Literature Review	11
2.1. Introduction	11
2.2. Alternative Existing Solutions to Your Problem	11
2.3. Technologies researched	13
2.3. Technologies Chosen	13
2.4. Other Research you've done.....	14
2.5. Existing Final Year Projects	17
2.6. Conclusions	18
3. Prototype Design.....	19
3.1 Introduction	19
3.2. Software Methodology	19
3.3. Overview of System	23
3.4. Front-End	24
3.5. Middle-Tier.....	29
3.6. Back-End.....	29
3.7. Conclusions	30
4. Prototype Development	31
4.1. Introduction	31
4.2. Front-End	31
4.3. Middle-Tier.....	35
4.4. Back-End.....	37
4.5. Conclusions	38
5. Testing and Evaluation.....	39
5.2. Plan for Testing	39
5.3. Plan for Evaluation	39

6. Issues and Future Work	40
6.1. Introduction	40
6.2. Issues and Risks.....	40
6.3. Plans and Future Work.....	40
6.3.1. GANTT Chart	40
Bibliography.....	41

Table of Figures

Figure 1 Example of a login screen created with Kivy	7
Figure 2 Comparison of easy and difficult topics on Duolingo	8
Figure 3 Paper Prototype of a minigame	8
Figure 4 AB Maths minigame.....	11
Figure 5 Minigame paper prototype.....	11
Figure 6 Minigame wireframe.....	11
Figure 7 Sprints lifecycle as part of the Scrum Methodology	19
Figure 8 Scrum lifecycle as part of the Agile Methodology	20
Figure 9 Project Trello board	21
Figure 10 Card details for Interim Report	22
Figure 11 Three-Tier System	23
Figure 12 Technical Architecture Diagram.....	24
Figure 13 First iteration use case for both students and teachers	24
Figure 14 Second iteration use case for students.....	25
Figure 15 Third iteration use case for students	26
Figure 16 Second iteration use case for teachers.....	27
Figure 17 Third iteration use case for teachers	28
Figure 18 ERD showing relationships between tables	29
Figure 19 Title Page of the app	32
Figure 20 Register Page of the app	33
Figure 21 Login Page of the app.....	34
Figure 22 Main Page of the app	35
Figure 23 Snippet of code showing Kivy libraries	35
Figure 24 Snippet of code showing firebase library being used	36
Figure 25 Snippet of code showing the workings of the log in.....	36
Figure 26 Available screens in the Kivy file.....	37
Figure 27 Example of elements found on a screen in the .kv file	37
Figure 28 Current state of the Firebase database	38
Figure 29 GANTT Chart	41

1. Introduction

1.1. Project Background

Many sources state that children even from young ages can learn a lot by using technology and can reinforce information already covered in the curriculum through gamification through apps on phones or tablets. Below are articles and documents about these sources and also includes some research done to find what the best programming language would be to write the Primary Mathletes application using:

This piece discusses the different programming environments that are available for mobile platforms such as python, c#, and java through Android Studio. The paper brought to attention the programming language python. Originally the Primary Mathletes application was going to be developed using Android Studio which, for a lot of cases is not very user-friendly in comparison to something like python which lets you do a lot with very little code. [1]

This paper discusses how the use of mobile devices such as iPads can be used by teachers to make the teaching of maths more efficient and more involved by having data stores with a piece of software on a mobile device. It also discusses several mobile learning approaches and how they can affect a student's learning positively or negatively. [2]

This paper is similar to Matthew Kearney and Damian Maher's paper but instead of focusing on iPads it is about how there is limited access to PC's in South Africa but more than 3 million teenagers have Java enabled mobile phones which leaves a market open for learning tools to be developed digitally rather than on paper in the classroom. [3]

This paper discusses the differences between e-learning (using a PC to teach students) and m-learning (using mobile devices) such as handheld phones and tablets that are internet-enabled and how m-learning is a lot more accessible because there is no hardware limitation of PC's because almost everyone nowadays already owns a mobile phone. This cuts the cost of textbooks and on building expensive computer rooms for students. [4]

1.2. Project Description

The Python [5] programming language was used for this project with heavy use of the Kivy library [6] for GUI-aspects of the application and used Google's Firebase[7] database system to store users' login information and progression through the app.

The user will need to have at least: An Android [8] device running Android 6.0 (Marshmallow) which will cover over 60% of android users.

The app will be a 3-tier system with the UI being the presentation layer, The background code of the app itself is the application layer and finally, there will be a database taking care of the data layer.

If possible, an attempt will be made to get in contact with a primary school teacher and try to get a small group of primary school students to participate in a study group to try evaluating the app's usefulness.

A lot of research was conducted around how doable this application would be to create with what was put forward in the project proposal. There was some skepticism around whether Firebase was a suitable way of handling the back-end for the application as well as the issue of having to learn a new system, using Python with the Kivy library.

Hosting the web-app section of the application was also of some concern as there was a need to research whether a domain needed to be purchased from a site such as GoDaddy [8] or if it was possible to just host it locally using Django [9] or Flask [10].

When the user first downloads the application they will be asked to create an account with a username/email and a password which will then be stored in a Firebase database (The password will be hashed to prevent users' information being stolen). After this initial set up, every time the user goes into the app it will remember their details and sign them in automatically. If ever the app cannot verify that the user came from the same device then it will ask for confirmation through username and password or just password.

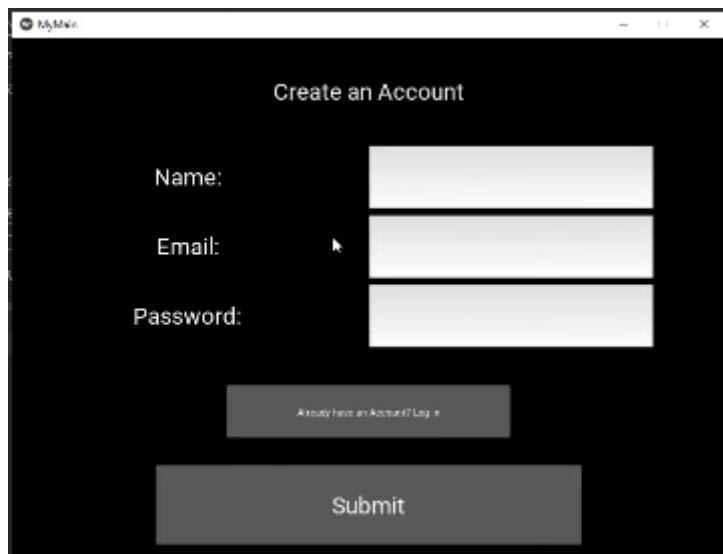


Figure 1 Example of a login screen created with Kivy

The app will follow a very similar approach to the way the Duolingo web app[11] handles progression. It will start from the very beginning of a topic and get progressively harder as you get better and better at a particular area of maths. The app will have a points/currency system that will act as a reward system for the user to improve their experience. This will hopefully be tied in with some nice sounds for correct answers, finishing a topic, etc. that will help with the user experience

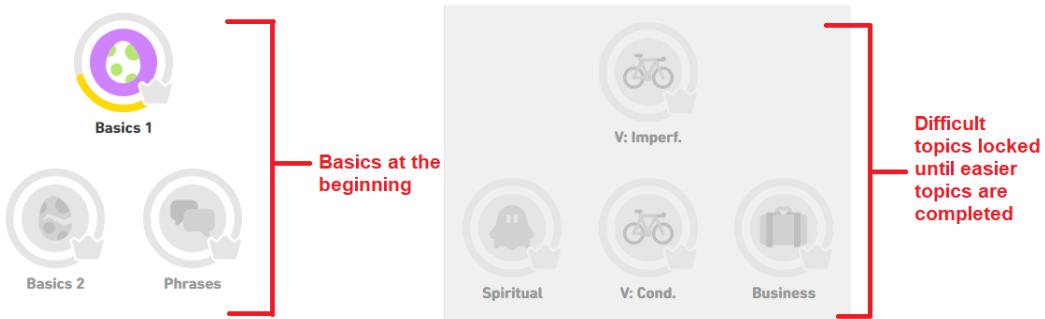


Figure 2 Comparison of easy and difficult topics on Duolingo

The content should not be too difficult to create, some research has gone into primary school books and website resources on maths and following their approach. The topics will stay quite basic as it is only for primary school students so slight progress in difficulty will ensue as the user completes more and more topics. An example of this may be going from low numbers [1-10] and gradually moving to [11-50] and then again to [50-100] but the basic idea will stay the same, such as learning addition subtraction, etc.

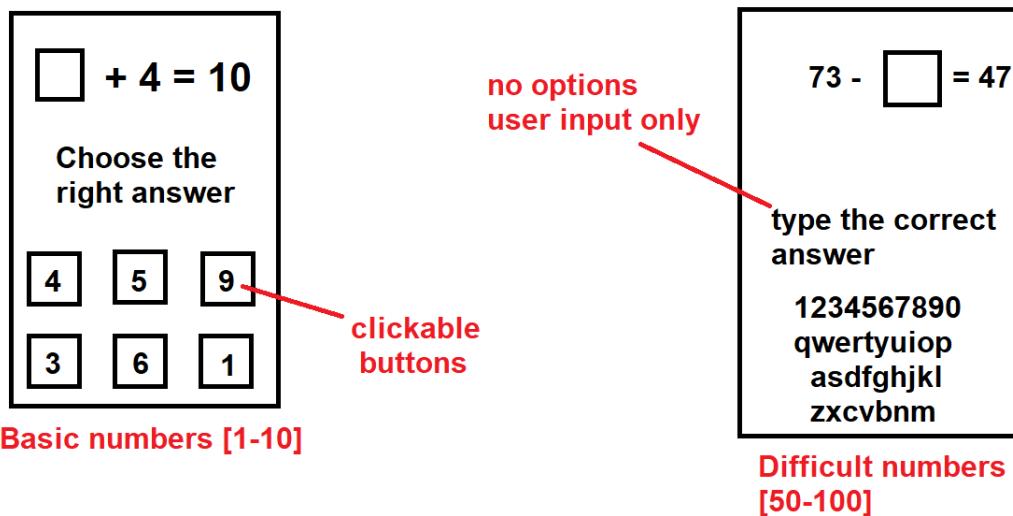


Figure 3 Paper Prototype of a minigame

1.3. Project Aims and Objectives

The overall aim of the project is to help aid primary school students and teachers to improve the understanding and abilities of the students in the maths curriculum.

The goal for the project is to create a mobile/web-based application that will be used by primary school students that will include minigames and helpful maths-related topics that will improve their overall understanding of the primary school maths curriculum.

Milestones are outlined in the GANTT chart that will be seen later in the report (Section 6). This was created to ensure that targets were set and completed (or at least attempted to be completed) within their given timeframe so that the project would be completed on or before the due date.

The purpose of the project is to see if it is possible to create an application that could actually be used by primary schools and in turn, would improve the understanding of all students that are studying maths from the age of 5 – 12. The project would be considered a success if the application was used by students and in their opinion, improved their maths abilities by some amount.

1.4. Project Scope

From research conducted, maths is one of the most hit or miss subjects that are tackled in any level of schooling whether it be primary, secondary or third-level education. The student either know the answers or they don't. So, because of this, there is a market for something that can come away from traditional learning on paper and through verbal teaching and gamify the process of learning to the point where it doesn't feel like learning anymore. With something like maths where there are definitive answers to everything, some students struggle with finding the right process to arrive at these answers. Some students don't enjoy traditional methods of learning and they may lose interest in class which will lead to them falling behind. Even though they could be a hyper-intelligent person, the method in which they learn can throw them off reaching their true level of knowledge.

1.5. Thesis Roadmap

Research

This section explores background research conducted on apps similar to the one being developed in this project and also research conducted on papers that outline positive and negative aspects of m-learning (mobile learning) and the benefits of using certain programming languages over others when developing an interactive application for younger students.

Design

This section includes use-cases and personas related to the application, a detailed description of the architecture used and testing plans for the application.

Development

This section delves deeper into the development process of the system that was outlined in the design chapter and will also include challenges that became apparent throughout the development of the application.

Testing and Evaluation

This section describes all the testing and evaluation of the system that was executed. Each part of the testing is described in detail and will have a detailed report of user feedback received during the user's evaluation of the application.

Redevelopment

This section will be included in the final dissertation and will outline some of the development steps taken as a result of the feedback gained from the user evaluation. The changes made and the importance of these changes will be examined.

Conclusions and Future Work

This section will be included in the final dissertation and will reflect on the entirety of the project and will discuss the conclusions drawn, personal reflections made, and the future work planned for the project.

2. Literature Review

2.1. Introduction

In this section, applications and websites that are similar to the application being developed in this project will be explored as well as why these other applications are not the be-all and end-all solution to the problem that this project is trying to solve. Additional research on exactly what students and teachers want and need from an app such as this will be conducted. Technologies researched and also a review of two previous computer science dissertations will be created.

2.2. Alternative Existing Solutions to Your Problem

AB maths for Apple iOS:

This app [12] wasn't developed specifically for Irish primary school maths students, but it seems to have a lot of general maths content behind paywalls for around 90% of the content. The free content that is available shows a similar structure to one of my original ideas for minigames that can be seen in the image shown below along-side a paper prototype and wireframe created for this project.

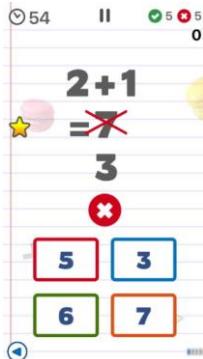


Figure 4 AB Maths minigame

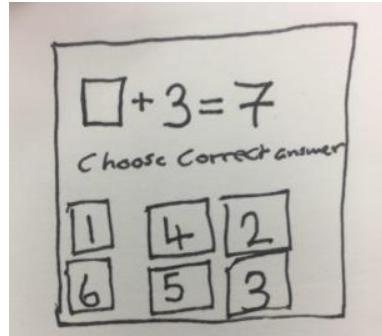


Figure 5 Minigame paper prototype

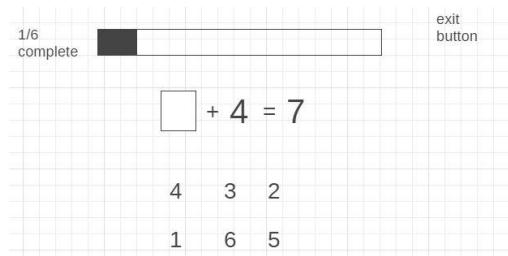


Figure 6 Minigame wireframe

Mathematics Skill Builder Primary for iOS:

This app [13] was developed for primary school maths students, but it is very outdated and cartoonish. It also seems to deal with topics that could be deemed too easy even for primary school students. Examples of this are challenges where a student needs to pick which number is bigger than the other, pick what shape is mentioned etc. Examples of this can be seen in the images below.

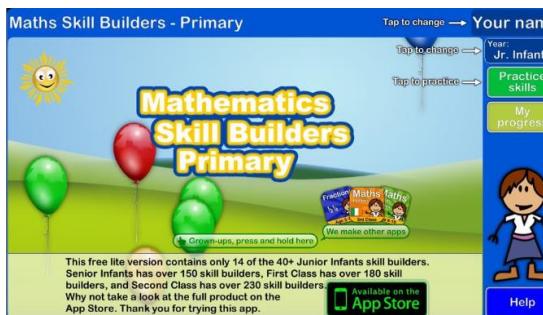


Figure #: Mathematics skill builders primary homepage

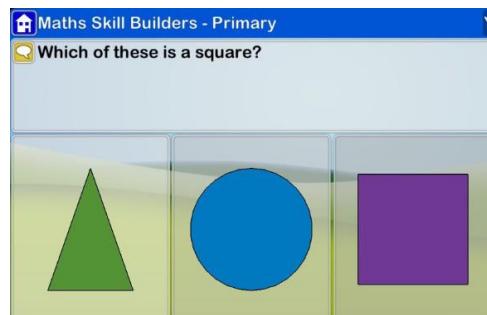


Figure #: Skill builder primary basic question example

ixl.ie website:

This website [14] has hundreds of different quizzes that are developed around the different years of curriculum for Irish primary school students.

🔊 Count **forward** by **fives** from 15.

15, , , ,

Submit

Figure #: ixl example question

All of these apps and websites are definitely in the same vein as the application that is being created for this project. Regardless of whether these apps are too basic or outdated, there seems to be room for improvement in many aspects such as content and user experience through sound and imagery.

2.3. Technologies researched

Quite a few programming languages were looked at in addition to all of the programs and languages that were studied throughout the last three years of computer science.

Firstly, some research went into web-based applications using HTML [15], CSS [16], PHP [17] and JS[18] but in the end, this was ruled out because after some conversations with primary school students and teachers it became clear that there is almost never enough desktop computers available for primary school students in their schools but most schools nowadays have access to tablets whether they're Android or iOS.

Some research also went into using Java [19] and the Android Studios IDE [20] because in 3rd year of computer science there is a module on mobile app development using both of these and creating an application using something already studied would be easier than learning a new programming language and IDE but after completing this module with a low grade it was evident that Android Studios would not be the ideal choice for this project. It is difficult to use and is not very forgiving when something goes wrong. Hard to understand error messages also makes the IDE difficult to work with.

In the 4th year of the computer science course, there are many modules that use Python. Researching Python led to the discovery of a plethora of resources for the language and this is where the Kivy library for python was found. The Kivy library can be used to create GUIs based on python code which is great because Python is very easy to use and there's a lot of great resources online for Kivy as well. Kivy also allows for cross-platform development which is perfect for this project because not every primary school student has access to predominantly iOS or predominantly Android phones or tablets.

For databases, some research went into using either a local SQL database [21] that would store the user's data locally but this led to problems such as if a student lost his or her phone then all of their progress would also be lost. There was also an issue of the teacher not being able to access the child's progress reports without going onto the student's device manually.

In the end, Google's Firebase Database System was chosen because this means that any user can access their account just by logging in on any random device, it doesn't need to be their own personal device.

2.3. Technologies Chosen

After research and contemplation on what should be used to develop the application of this project a programming language, a database system, some libraries and an IDE were decided upon.

Python

The Python programming language was chosen because of its easy usability and multitude of resources and tutorials found online.

Google's Firebase Database:

The Firebase system was chosen because of the ability to connect to the Python code using its own Python library and because it has built-in authentication and email verification for login and register.

Kivy library:

The Kivy library was chosen because it allows for easy-to-setup GUIs and it allows for cross-platform development.

Pycharm IDE:

Pycharm was chosen because it is easy to use and because it offers suggestions when coding that makes creating readable and concise code more efficient.

2.4. Other Research you've done

Below are some examples of other research conducted before beginning the development of the application:

- Twinkl: This website has a lot of resources, both for primary and secondary students as well as teachers. It allows teachers to create plans for their students and children to follow based on the curriculum given. [22]
- Transum Maths: This website has resources for both students and teachers, but it is not tailored to the Irish primary maths curriculum but instead for the British GCSE's. [23]
- NCCA Maths Development: This website provides information on the development of the new Primary Mathematics Curriculum. [24]
- NRICH Maths: This website provides problem-solving questions that help to reinforce things that students have learned after they finish a section of the curriculum. [25]

Before short document that I sent to some primary school teachers that contained a revised version of the Abstract and Section 1 of this report just to give an overview of the idea for the project, along with a short questionnaire that can be seen below:

Your Input:

1. [a] Do you think the app could be useful for you as a primary teacher?
- [b] Would students use it?

2. Do you think there are any problems with the idea and if so, what do you think could be done to correct them?
3. Do you know of any resources (websites, books or both) that I could look into to access the curriculum for primary school maths?
4. What form of devices are primary school students most comfortable with? (e.g Desktop Computer, Laptop, iPad/Tablet, mobile phone)
5. Off the top of your head do you think there is any other kind of minigame-Esque questions that would be helpful (and hopefully fun) to engage students more with the app?
6. [a] Do you think to have a section of the app dedicated to times tables would be worthwhile creating?
[b] Is there anything else in the same vein as this idea that you think could be essential to have in the app?
7. Is there a specific age the app should NOT be aimed at (junior, senior, 1st class)? Is there any point in targeting students this young?
8. Do you have any suggestions for the name of this app?
9. Can you think of any ideas that would keep the students more engaged in the app?
10. Are there any common difficulties students have with maths that I could address with the app?

Other Suggestions & Additional Feedback?

A few days after this questionnaire was sent to the teachers, responses began coming back and the collective document below was created. It gathers all the key wants and needs that application should have based on the teacher's feedback:

Things noted after speaking to primary school teachers about the app:

- Most primary school students nowadays are happy to use tablets/iPads. (steering away from desktop computers and laptops depending on schools)
- Most primary schools have poor internet connection so the ability to use the app without an internet connection is preferred but not essential.
- The app could be used for assessment as a group in the classroom after finishing a section of the curriculum. (Work on seeding so every student in the class gets the same exam questions)
- Primary school curriculum:
 - Twinkl
 - Transum Maths
 - Curriculum online [26]
 - Haylock maths for primary teachers [27]
 - Folens online [28]
 - SEN [29]
 - NCCA Mathematics curriculum online
 - Mental Maths [30]
 - NZ maths [31]
 - Bee Bots (for teaching directions) [32]
- Maths Tables section of the app would be good.
- Tooltips / Glossary for what certain keywords mean.

- App should be aimed for 2nd class and over but some have said even junior infants would benefit.

- To help with engagement:

- Bright/contrasting colours
- Badges
- Hierarchical system that gets more difficult as you go along
- Positive praise
- Good sound design for correct and incorrect answers
- Pictures
- Points system
- Allow them to create their own avatar/upload an image

- The app would benefit teachers in terms of more advanced children, the teacher can assign additional work for students that are abler when it comes to maths.

- It would also allow teachers to assign additional work to students who are struggling so that they could catch up.

- The addition of an app into the student's life would be a breath of fresh air and help with engagement in class.

- Mini-game ideas:

- Odd-One-Out
- Matching Games
- Time trial mode / Beat the clock
- NRICH maths online (problem-solving questions)
- ixl.ie (online quizzes for Irish maths curriculum)
- Word problems

- Not every student would be provided an iPad/tablet to use on their own so group activities might be necessary, but this is an issue for the school rather than for the development of the app.

- Combine all curriculum strands to make quizzes at the end of each section.

- Users should be able to see a graph of the user's progress at the end of the week/month/year/all time.

2.5. Existing Final Year Projects

Project 1

Title: Fantasy Premier League Predictive Analytics

Student: Alex Brady

Description:

This student attempted to create a web application that would be able to accurately predict how many points a player would get in the next football match based on the previous seasons. The student split the project into three sections:

The first of which was a predictive analytics system that was created in Jupyter Notebook using the Python coding language. This part of the project would do the heavy lifting in terms of scraping historical data, sorting the data and generating the predictions. The second section was a Python Flask back end web service that held all of the predictive data which was cloud-hosted.

Finally, the third section of the project was a web application that the student developed with AngularJS and NodeJS. This was the part of the app that the user would see when using the student's project. In the end, the student is quite happy with how all of the features ended up after the proposal at the start of the paper.

Although there were some features that were never implemented the core idea for the project was a success. The student was able to confidently answer the question he put forward at the beginning of the conclusion "can a machine know more than me?" and so I believe that the project was a successful one.

The student also planned to continue to update the app and add features in past the submission of the project to DIT.

Project 2

Title: Temple of Thoth

Student: Keith Mc Loughlin

Description:

The student set out to create a game environment that tracks users' behaviors and uses this information and apply logic to the data in such a way that it could predict the best way to challenge specific users. It aimed to manipulate a 3D virtual environment with the intention of leading the player into traps and to discourage them to reach the end goal. The reason for this was to see could the game itself design a challenging level purely based on the player's previous moves and decision-making patterns and use these against them.

The game aimed to see could it control user behavior by using previous data farmed from global user data and actively use player's minds against current players. The game was created inside the unity engine and was written in C++ and used a MongoDB database to store player information.

In the conclusion the student talks about how they now know how to effectively design a game, create different scenes and game objects in unity and persist data that was generated inside of the game. This student also talks about how the project could be further worked on and perhaps more on how the project can track user behavior.

2.6. Conclusions

In this section, some previous final year projects that students I previous years had written were reviewed. Some additional research conducted through questionnaires and conversations with current primary school teachers was discussed as well as a few websites that have primary school curriculum information. Applications other people have already created that attempt to help primary school students and other students of similar age with the maths curriculum were looked into. Finally some programming languages, databases and APIs were researched and decided upon that best-suited development of the application.

3. Prototype Design

3.1 Introduction

In this section, some common software methodologies are discussed, followed by a description of the one finally decided upon and why and how this decision was made. There is also a thorough overview of the system and an explanation of the front, middle and back tier of the three-tier system implemented for the design of this project.

3.2. Software Methodology

Scrum –

Scrum [33] is a methodology that is comprised of five values: commitment, courage, focus, openness, and respect. It usually involves someone taking over the role of scrum master who organizes daily stand-ups and points out places in the development team where there may be blockages and see can they be removed in one way or another. It also involves doing weekly or bi-weekly sprints which are fast and efficient pushes of work on one particular task over a short period of time. After this is finished there is a sprint review and retrospective, where lessons are created that, will ensure that sprints that follow the last will always be more efficient than the last. Scrum is a methodology that is usually used when working with a team on a project but due to the fact that this is a solo project, it did not seem like it would really be a suitable choice for this project.



Figure 7 Sprints lifecycle as part of the Scrum Methodology

Agile –

Agile [34] is one of the most famous and recognizable project management methodologies. It was developed by industry leaders that wanted to uncover better ways of developing software by having clear and measurable goals that take into consideration iterative development. It is made up of 4 fundamental values and 12 key principles. Some of the main features include having working code over comprehensive documentation, responding to change over following a set-in-stone plan and being simplistic. I decided not to use Agile but instead opted for a subsection of Agile which can be read below.

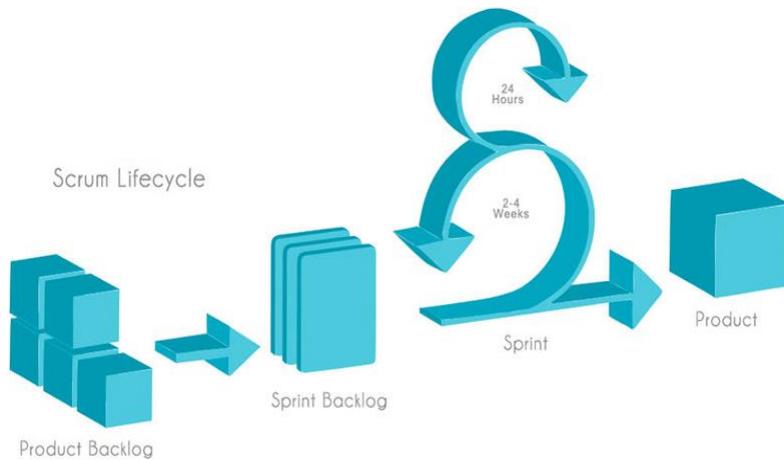


Figure 8 Scrum lifecycle as part of the Agile Methodology

Kanban Methodology –

A Kanban board is one version of the Kanban project management methodology [35] that is used to keep track of tasks to be done during a project. It is a subsection of the Agile approach to project management and they are used to visualize workflow for a given period of time and helped me a lot throughout the progression of the project as it let me add items to the backlog whenever I thought of something that I may have forgotten to add earlier or a new idea that I thought would make a good addition to the project.

The image below shows how I used Trello [36] as my Kanban board for the project. Each task is given its own card and was initially placed into the backlog list. The structure of the Trello board allowed me to pick one task at a time and work on it until it was finished, then it was either added to the “testing” list or the “completed” list depending on whether it was a programming task or just a researching or writing task.

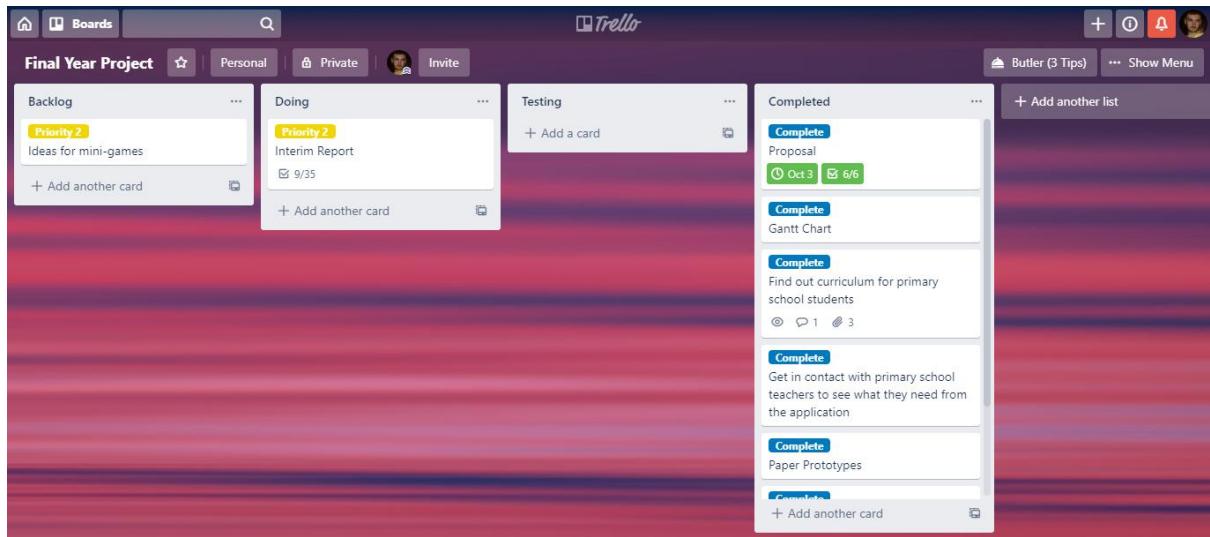


Figure 9 Project Trello board

Below, you can see what is inside of the cards themselves. Usually inside a card there would be a breakdown of all the working parts that need to be completed for the overall task to be finished. Normally I would create a checklist like you see in the image below but I also have cards where I have added useful links to pages I have researched added to the card themselves, as well as comments and descriptions that would aid me in completing the task to the best of my ability. By adding comments and checklists to the cards it helped to visualize the tasks that needed to be complete without cluttering the overall Trello board with hundreds of small tasks that could all be grouped into one card anyway.

Interim Report

in list [Doing](#)

LABELS

Priority 2 +

Description

Add a more detailed description...

Checklist

26%

[Hide completed items](#) [Delete](#)

1.1 Project Background

1.2 Project Description

1.3 Project Aims and Objectives

1.4 Project Scope

1.5 Thesis Roadmap

2.1 Introduction

2.2 Alternative Existing Solutions to Your Problem

ADD TO CARD

Members

Labels

Checklist

Due Date

Attachment

Cover

POWER-UPS

Butler Tips (3)

[Get More Power-Ups](#)

ACTIONS

Move

Copy

Make Template

The screenshot shows a Trello card for 'Interim Report'. The card has a yellow 'Priority 2' label. It includes a 'Description' section with a placeholder 'Add a more detailed description...'. A 'Checklist' section is present with a progress bar at 26%, listing items from 1.1 to 2.2. To the right, there are sections for 'ADD TO CARD' (Members, Labels, Checklist, Due Date, Attachment, Cover), 'POWER-UPS' (Butler Tips), and 'ACTIONS' (Move, Copy, Make Template).

Figure 10 Card details for Interim Report

3.3. Overview of System

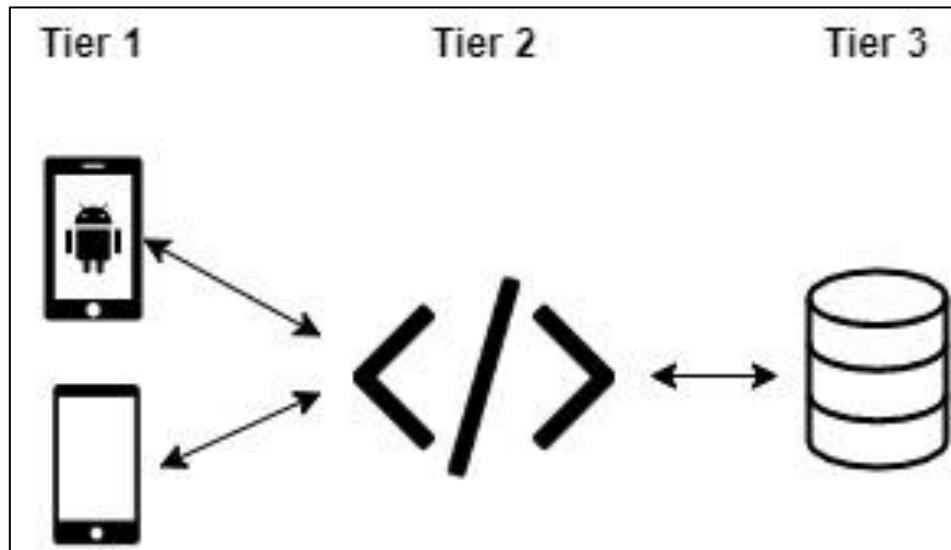


Figure 11 Three-Tier System

As shown in the above image my proposed architectural approach for the project is a basic three-tier model with the first tier separated into three separate components. An in-depth breakdown can be seen in the image below.

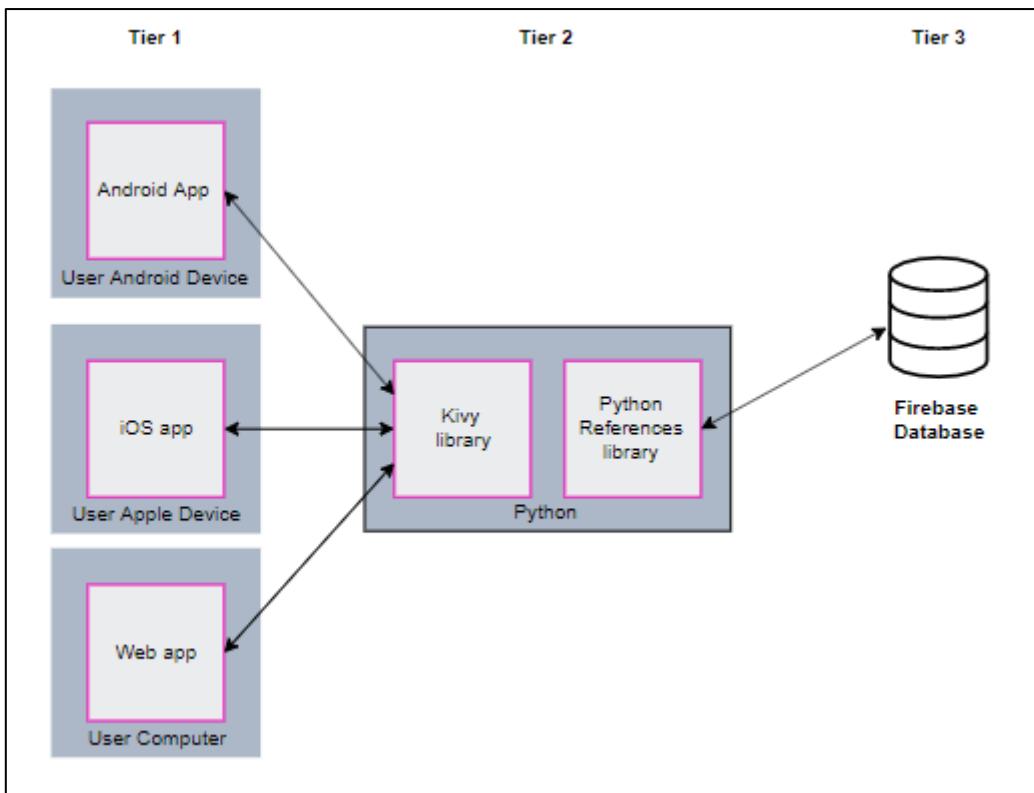


Figure 12 Technical Architecture Diagram

3.4. Front-End

Use Cases:

In the application, there are small pieces of functionality that can be accessed by both students and teachers. Then there are some that can only be accessed by students and some only by teachers. In the use case below you can see the first iteration of the use case which can be both accessed by student and teacher. The second and third iterations of the use case diagrams are labeled either as students only or teachers only.

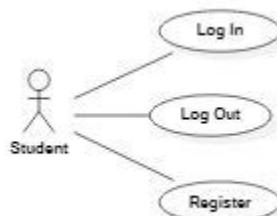


Figure 13 First iteration use case for both students and teachers

The functions shown in the diagram above will be available for both students and teachers to use. These functions will contact the Firebase Database using the firebase library and using either a get request for login or a post request for register.



Figure 14 Second iteration use case for students

The second iteration functions shown in the diagram above will be available for only students.

After logging in there will be:

- A page of playable minigames where they can select one. Each minigame will cover important topics of the primary school maths curriculum.
- An icon of their choosing that will allow them to view their profile that will be shown on another page.
- A button that will bring the user to a new page to view their progress.
- A button that will bring the user to a new page to view the classroom section of the app.

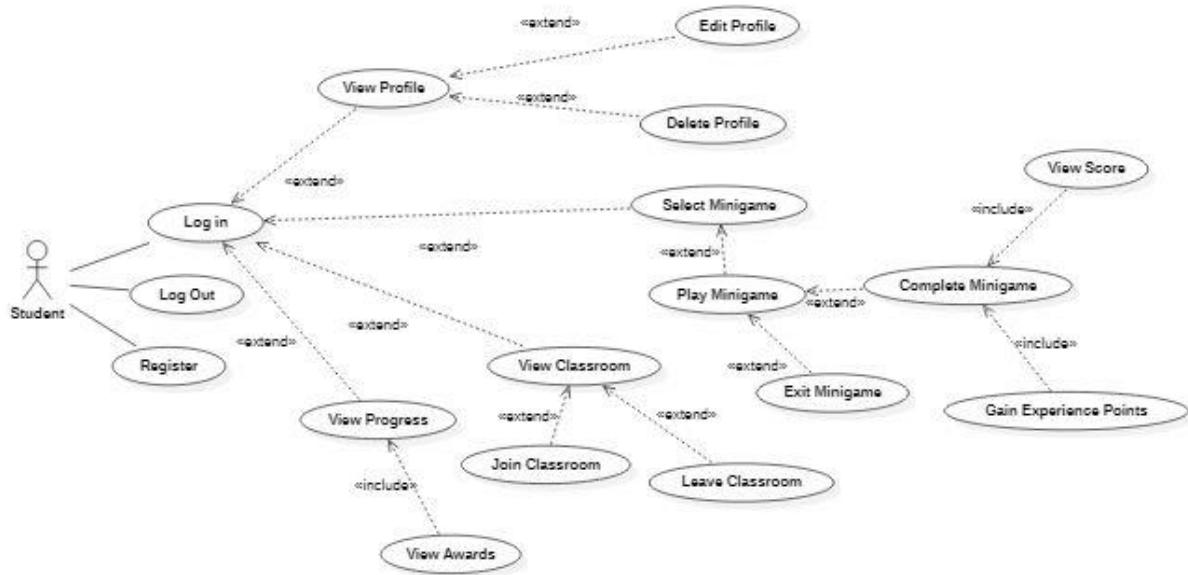


Figure 15 Third iteration use case for students

The third iteration functions shown in the diagram above will be available for only students.

If the user decides to view their profile, they have the following options:

- Edit their profile, including updating their username, updating their email address and updating their password.
- Delete their profile.

If the user decides to select a minigame they will be able to play the minigame itself. When playing the minigame the user will be able to either exit the minigame or complete the minigame. After completing the minigame the user can view their score and in turn gain some experience points as well.

If the user decides to view the classroom, they will have the following options:

- Join a classroom if they have not joined a classroom yet.
- Leave a classroom if they are already part of one.

If the user decides to view progress, they will be able to view their awards.

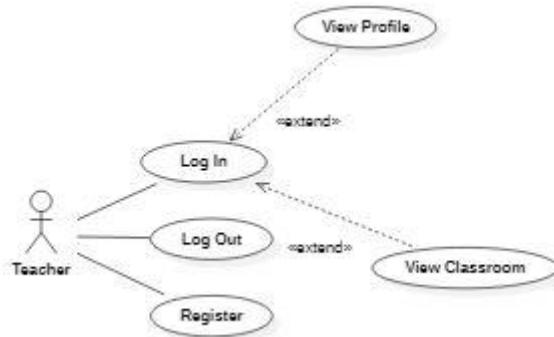


Figure 16 Second iteration use case for teachers

The second iteration functions shown in the diagram above will be available for only teachers.

After logging in there will be:

- An icon of their choosing that will allow them to view their profile that will be shown on another page.
- A button that will bring the user to a new page to view the classroom section of the app.

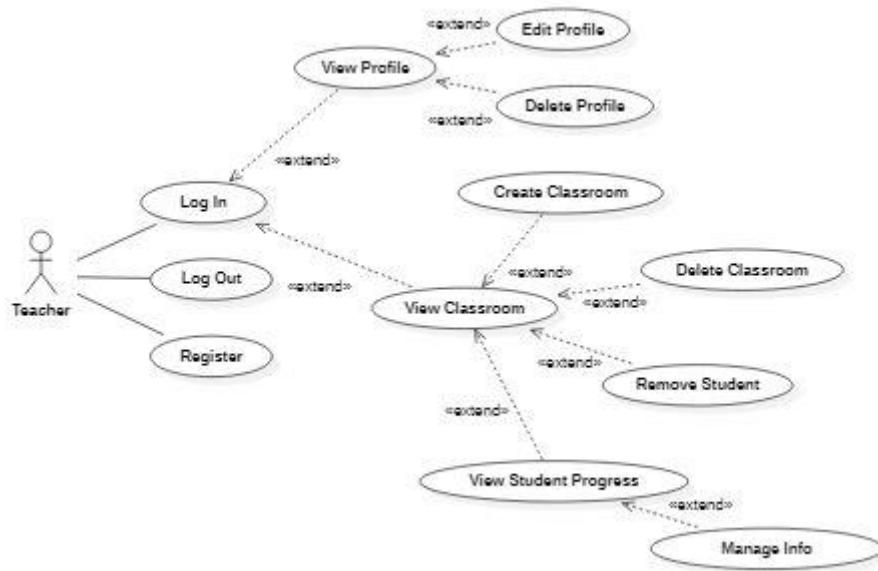


Figure 17 Third iteration use case for teachers

If the teacher decides to view their profile, they have the following options:

- Edit their profile, including updating their username, updating their email address and updating their password.
- Delete their profile.

If the teacher decides to view the classroom, they have the following options:

- Create a classroom for students to join if they have not created one yet.
- Delete the classroom if they have already created a classroom.
- Remove students if they no longer want them to be there.
- View individual student's progress and manage the student's information.

3.5. Middle-Tier

The middle tier contains all of the functions in the code that drive the application's capabilities. It will allow the users to input data to the front-end of the app and the middle tier will take this information and interact with the database. This will happen on almost every page where the user is doing anything such as:

- Logging in
- Registering for the app
- Editing their personal information on the app.
- Playing minigames and gaining experience points.

3.6. Back-End

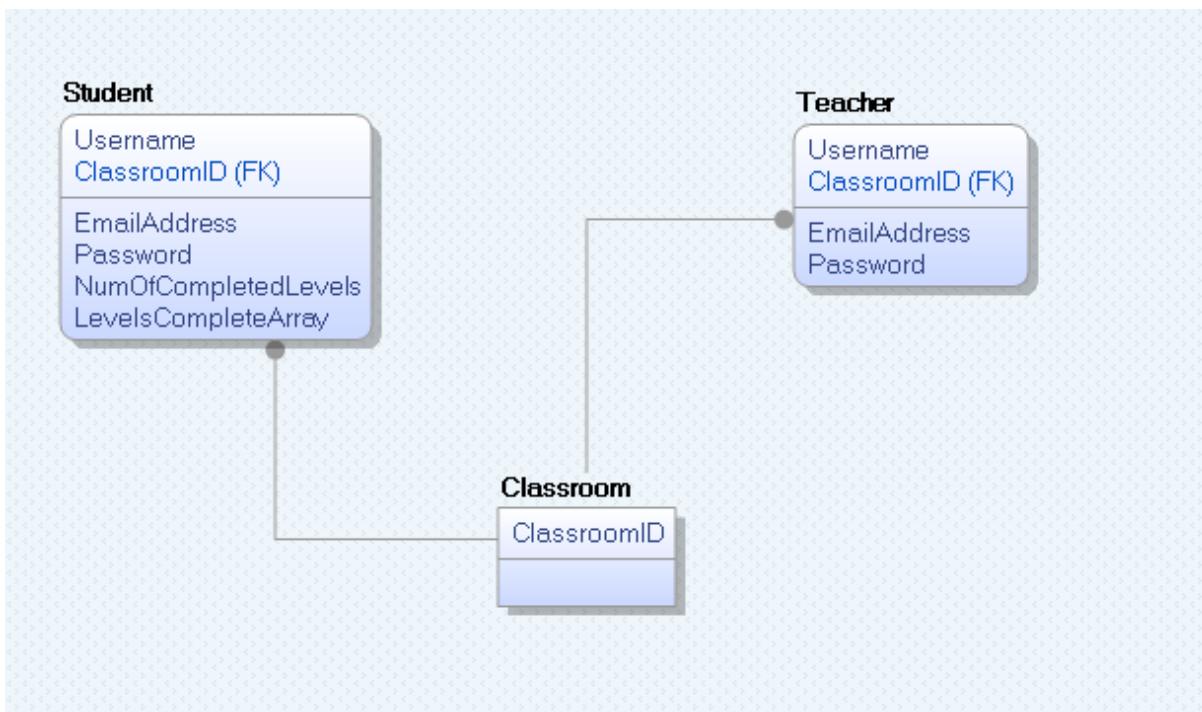


Figure 18 ERD showing relationships between tables

The above image shows the Entity-Relationship Diagram mock-up that was first developed to display the relationships between the tables that would be found later in the Firebase database. The student and teacher will both share similar attributes such as a unique username, a unique email address and a password. They will also be able to share one classroom by using the ClassroomID foreign key. Each classroom will be able to have many students and one teacher.

3.7. Conclusions

To conclude this section, many things were touched on, an overview of the system was described that touched on the front, middle and back ends of the system which led to the development of the prototypes (Section 4). There was a discussion on software and development methodologies which helped to structure prototype development in a precise and efficient coming up to the interim deadlines.

4. Prototype Development

4.1. Introduction

In this section, there will be a demonstration of the working prototypes that have been created so far for the front and back end of the application as well as how the middle tier helps the front end communicate with the back end for data retrieval from the database. There will be a breakdown of the front-end pages that are shown to the user and why they were chosen, code samples of the middle tier that show how the front end is created as well as code samples on how the front end communicates with the database in the back-end and finally there will be some screenshots of the database/back-end itself along with a description of its purpose and details on why everything is where they are.

4.2. Front-End

The prototypes created for the front-end prototypes are very minimalistic. They were created with the barebones, default appearances of the Kivy library, purely because there will be a big emphasis leading towards the end of the project on the user experience, including colors, images, sounds, and touch interaction, but for the development of small prototypes before the Winter break there was a need to first research and practice using the library before trying to make anything look and feel fancy.

For these prototypes, two types of files were used: Python (.py) files and Kivy (.kv) files. (Discussed in Section 2.3)

Title Page

The title page is the page that the user is brought to when they open the app for the first time. It allows the user to decide whether they would like to register if they have never used the application before or if they want to log in if they already have an account. This first page was created for the register and login functions, but it was also created to test the ability to create buttons as well as swapping between screens which is a big part of the Kivy library.

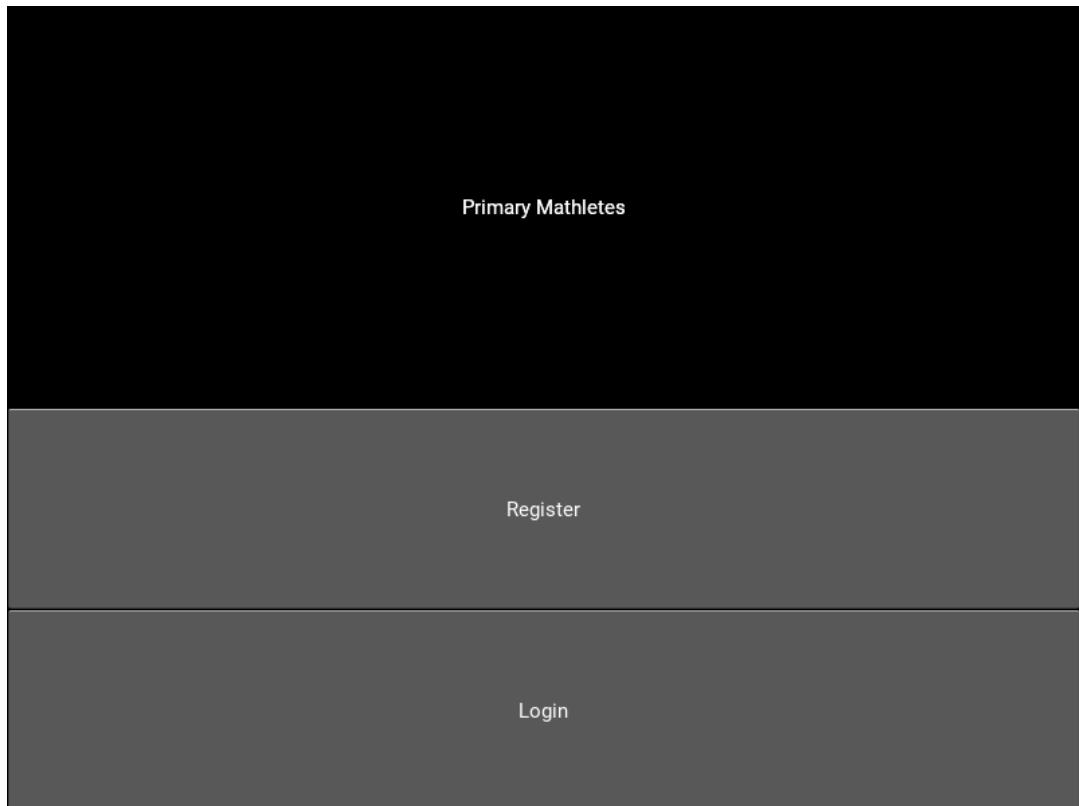


Figure 19 Title Page of the app

Register Page

The register page was created to again, try out the screen swapping capabilities of the Kivy library. It allows the user to press a button and be brought to the login screen. The register page was also created to try out the text input option that Kivy offers. Some labels were created to let the user know what information was being asked for and then text boxes were included underneath them for the user to fill out. Once the information was all there, the user can press the register button. The middle tier will then check the database to see if another user has already used this information to create an account. (This will be discussed in more detail in the next section)

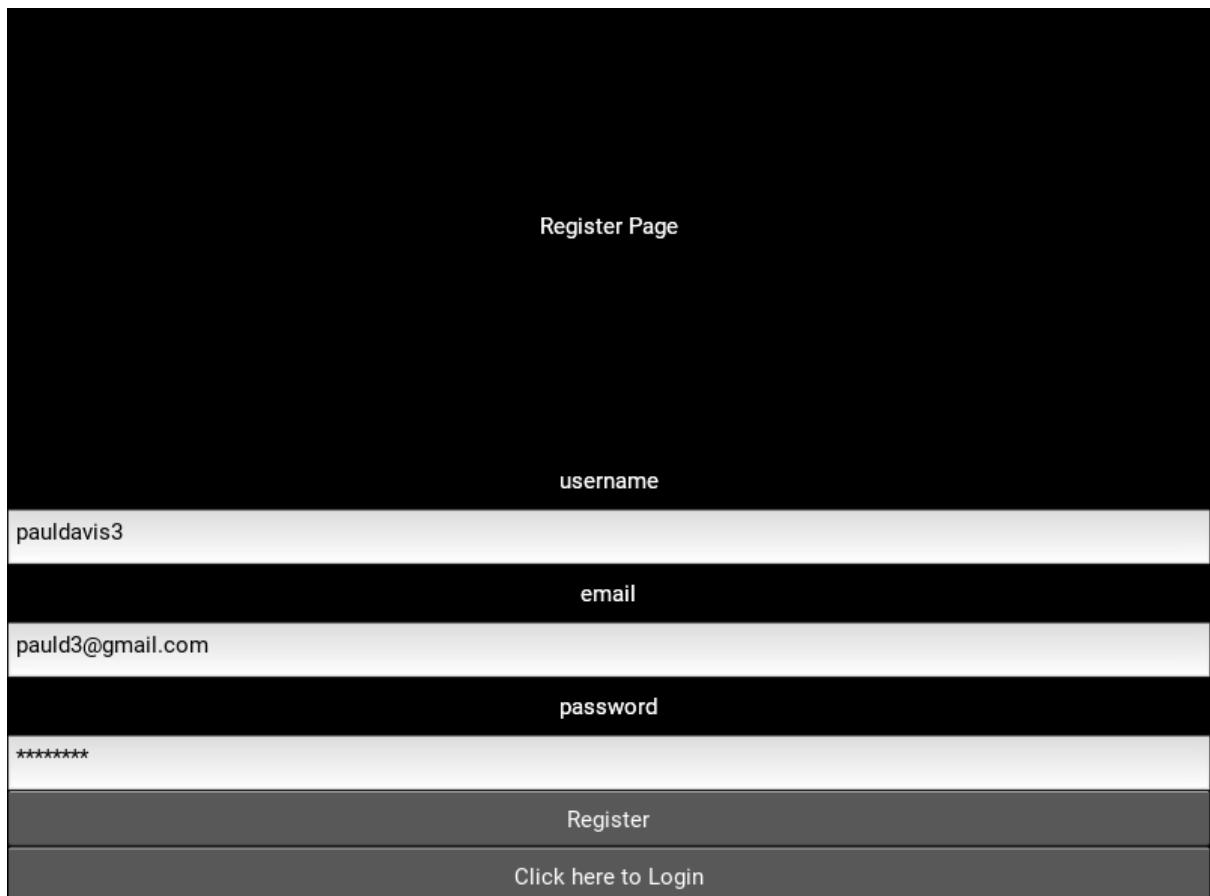


Figure 20 Register Page of the app

Login Page

The login page is very similar to the register page except that there is no text box for the email as this information should already be in the database. The user can enter their details on this page and then press the log in button the be brought to the main page of the app provided that the details they entered match a pair found in the database, otherwise they will stay on the same page.

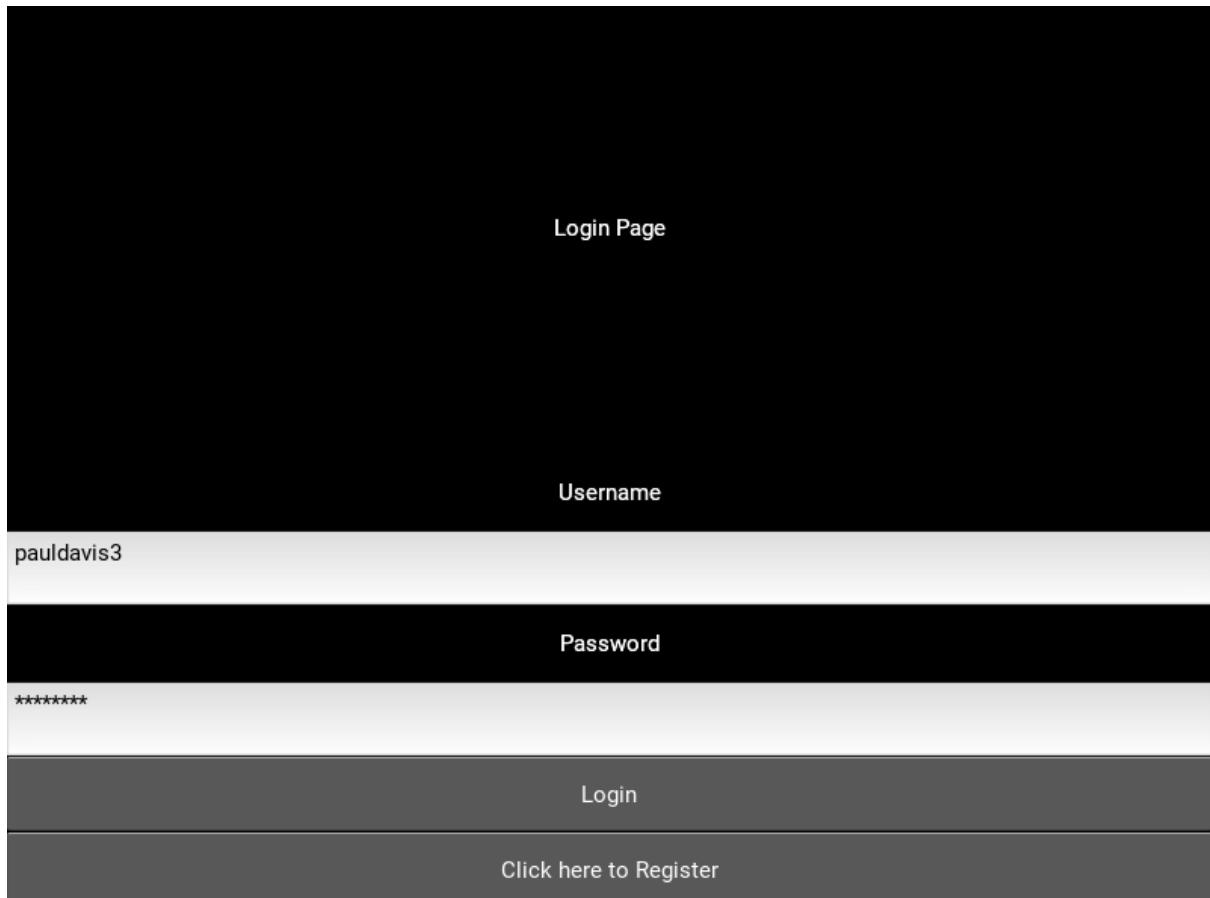


Figure 21 Login Page of the app

Main Page

The main page, at the moment, does not let the user do anything else. The main reason for the creation of this page was to visualize to anyone using the app that their login attempt was a success and in future versions of the application they will be brought to this page where they can do a number of different things.

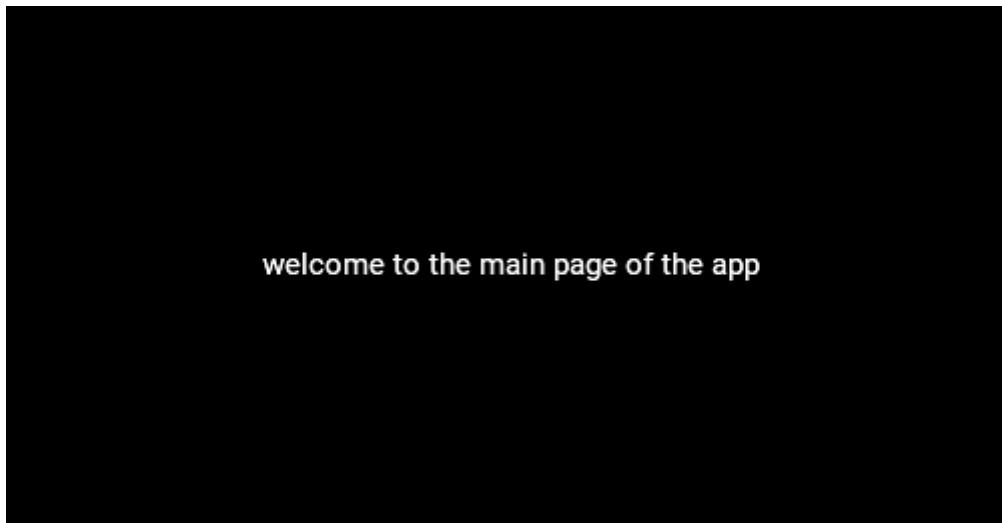


Figure 22 Main Page of the app

4.3. Middle-Tier

Python File

Inside the python file, there are a few pieces that are crucial to make the app run. The Kivy library along with its screen manager sub-library is responsible for most of the interactive features and functions from the front-end like buttons, labels, text input boxes, etc. as well as allowing the user to traverse between screens, like going from the title page to the login page and from the login page to the main page. The builder library allows for the python file to communicate with the .kv file.

```
1  from kivy.app import App  
2  from kivy.uix.screenmanager import ScreenManager, Screen  
3  from kivy.lang.builder import Builder
```

Figure 23 Snippet of code showing Kivy libraries

The other import library is the firebase library, this allows for items in the database to be created, read, updated and deleted (C.R.U.D). For the current prototype, there are only two firebase functions being used: post and get. Get requests to the database are used twice within the python file: once to ensure that the user has not entered information that has already been used to register their account and then again when the user is logging in to ensure that their username and password pair matches in the database. Below you can see a code snippet showing the importing of the firebase library and below it is where the library is used to reference the Firebase database in the code.

```

4     from firebase import firebase
5
6     firebase = firebase.FirebaseApplication('https://c16324311fyp.firebaseio.com/')

```

Figure 24 Snippet of code showing firebase library being used

When the user attempts to login in the piece of code below gets run. The code takes the username and password that the user typed and loops through all of the data in the user's table of the database. It checks to see if the username and password pair match anything that's found and if it finds something then it prints a message to the console saying that the user has logged in successfully before changing the screen to bring the user to the main page. If the username and password pair do not match anything found in the database then a different message, explaining that nothing in the database matched the user's input, and the user is kept on the log in page where they can attempt to log in again or choose to register a new account.

```

24     class LoginPage(Screen):
25         def checkLogin(self, uname, pword):
26             if self.loginLoop(uname, pword) == -1:
27                 print('this username and password do not match anything in the database')
28             else:
29                 print('log in successful')
30                 self.manager.current = 'main'
31
32         def loginLoop(self, uname, pword):
33             while True:
34                 results = firebase.get('/users/', None)
35
36                 for index in results:
37                     if results[index]['username'] == uname and results[index]['password'] == pword:
38                         return 1
39
40             return -1

```

Figure 25 Snippet of code showing the workings of the log in

Kivy File

The Kivy (.kv) file is usually used when programming in python with the Kivy library imported because it is very powerful. It is possible to do everything that a .kv file can do inside a python file but it is more long-winded and inconvenient than doing them inside a separate file. The fact the buttons and labels and input boxes etc. are kept in a separate file also means that there is a lot less clutter than having everything in one big python file.

```

1  ScreenManagement:
2      TitlePage:
3      LoginPage:
4      RegisterPage:
5      MainPage:

```

Figure 26 Available screens in the Kivy file

Most of the screens created in the .kv file follow a similar pattern. They have a name, a label and then some sort of grid layout to keep everything on the page in line. Some examples of the elements found in the grid are: buttons that bring the user to various parts of the app, text input boxes for the user to input information about themselves or labels that tell the user what they are seeing on the screen.

```

8 <TitlePage>:
9     name: "title"
10    GridLayout:
11        cols:1
12        Label:
13            text: "Primary Mathletes"
14            GridLayout:
15                cols: 1
16                Button:
17                    text: "Register"
18                    on_release: app.root.current = "register"
19                Button:
20                    text: "Login"
21                    on_release: app.root.current = "login"

```

Figure 27 Example of elements found on a screen in the .kv file

4.4. Back-End

For the current prototype the back-end Firebase database is only dealing with one, limited table, the users table. In future iterations of the application, the user table will be expanded to include the user's current experience level, their profile image, the number of levels which they have completed among other things. The database will also hold two more tables: one for teachers and one for classrooms. The database at the moment does not have much to do other than reading through information when the user attempts to register and log in and occasionally adding some information in when the user register successfully. A screenshot of some of the data in the user table can be seen in the image below.

The screenshot shows the Firebase Realtime Database interface. At the top, there's a navigation bar with 'Final Year Project' and 'Database'. Below that is a sub-navigation bar for 'Database' with tabs for 'Data', 'Rules', 'Backups', and 'Usage'. The main area displays a hierarchical database structure under the project name 'c16324311fyp'. The 'users' node contains several child nodes, each representing a user account. The structure is as follows:

```

c16324311fyp
  users
    -Lv6AIAPIS-N6zpL9TaE
    -Lv6Ath4pXlbILbNZHm8
    -Lv6Bh9gdsmV0o1hClp
      email: "paul@email.com"
      password: "password"
      username: "paul"
    -LvCHjnQXWsV_CWnOfaj
      email: "test@test.test"
      password: "12345"
      username: "testymctestface"
    -LvCHs2uNYULS2TTR7EF
      email: "craigisemail@email.co"
      password: "craig"
      username: "craig"
    -LvHVBU1goYmLxk0Gr8Y
    -LvHeaLkr3ZaPZGRc9Sw

```

Figure 28 Current state of the Firebase database

4.5. Conclusions

This section looked at how the prototype design was used as reference for the prototype development discussed in this section. The user's table in the Firebase database only contained a section of the table discussed in the ERD from Section 3 but the ground-work is there for the development of the other two tables as well as the additional information that should be a part of the users/student table.

5. Testing and Evaluation

5.2. Plan for Testing

At the moment, it is too early in development to know exactly what the plan is for testing. This will be constructed after the winter break and exams are finished.

5.3. Plan for Evaluation

The evaluation plan will also be constructed at the same time as the test plan, right now it is too early to decide what would be successful development of the application.

6. Issues and Future Work

6.1. Introduction

In this section, a list of issues and risks during the development of the app will be created. This list will be updated along the way as there is no way of knowing all issues that will arise before the proper development of the application has even begun. There is also a subsection discussing plans for the future development of the application that goes hand-in-hand with a Gantt chart that was discussed in Section 3.

6.2. Issues and Risks

6.3. Plans and Future Work

Future work will be to develop the application itself as well as write the dissertation alongside the application.

6.3.1. GANTT Chart

At the beginning of the academic year, when the deadline for the written proposal was over, this GANTT chart was developed. Since then it has been updated with new deadlines as they were introduced over the weeks that followed. After the Winter break, there will certainly be more updates to the GANTT chart to accommodate new deadlines. The GANTT chart and the Trello board that was mentioned previously in the report (Section 3) go hand in hand to ensure all tasks and deadlines are kept on top of.

Final Year Project

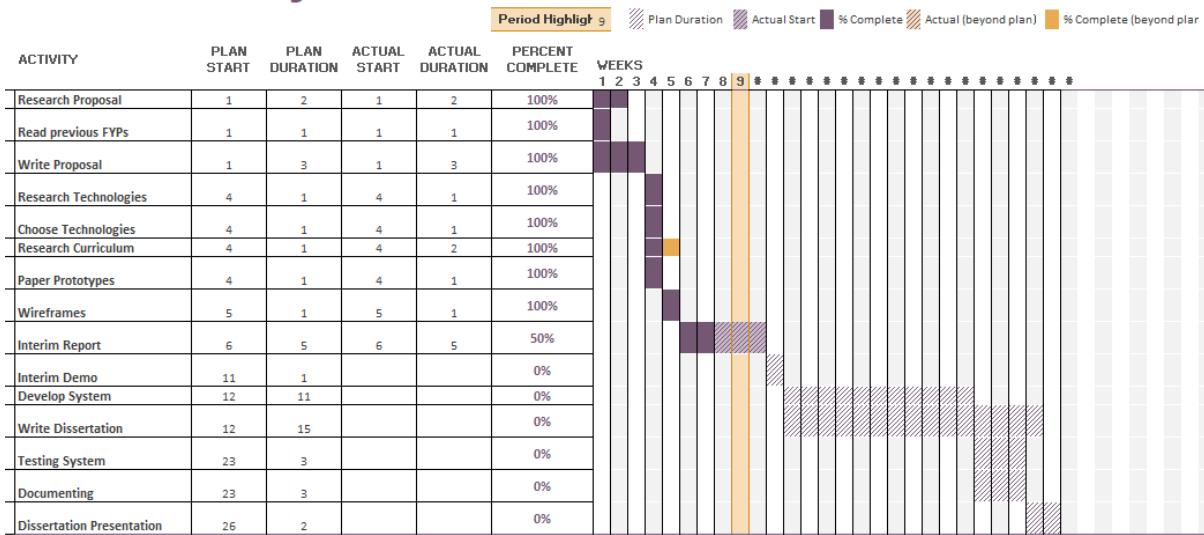


Figure 29 GANTT Chart

Bibliography

- [1] Emeteru, M., Oluwafemi, T., Akinlusi, T., Allen, M. and Ama, E. (2019). Developing a Mathematical Mobile App: A Case Study of an Environmental Model. [online] iaeng.org. Available at: http://www.iaeng.org/publication/WCECS2016/WCECS2016_pp150-153.pdf [Accessed 8 Dec. 2019].
 - [2] Kearney, M. and Maher, D. (2019). Mobile learning in maths teacher education: Using ipads to support preservice teachers' professional development. [online] researchgate.net. Available at: https://www.researchgate.net/profile/Matthew_Kearney/publication/285751515_Mobile_learning_in_maths_teacher_education_Using_ipads_to_support_preservice_teachers%27_professional_development/links/568c55ca08ae153299b66628/Mobile-learning-in-maths-teacher-education-Using-ipads-to-support-pre-service-teachers-professional-development.pdf [Accessed 8 Dec. 2019].
 - [3] Machdel Matthee, Jacobus - Mathematics on the Move: Supporting Mathematics Learners through Mobile Technology in South Africa. Available at: https://www.researchgate.net/profile/Machdel_Matthee/publication/238749401_Mathematics_on_the_Move_Supporting_Mathematics_Learners_through_Mobile_Technology/links/568c55ca08ae153299b66628/Mathematics_on_the_Move_Supporting_Mathematics_Learners_through_Mobile_Technology.pdf

[gy_in_South/links/0a85e53c38801e8c9d00000/Mathematics-on-the-MoveSupporting-Mathematics-Learners-through-Mobile-Technology-in-South.pdf](https://www.researchgate.net/profile/Youcef_Baghdadi/publication/236310002_Webenabled_services_development_with_respect_to_serviceorientation_paradigm/links/5528c2640cf29b22c9bcb350.pdf#page=185) [Accessed 8 Dec. 2019].

- [4] Al-Zeidi, A, Al-Kindi, K, Al-Khanjari, Z. A. - SQU Future: From E-Learning to MLearning Application Development:
https://www.researchgate.net/profile/Youcef_Baghdadi/publication/236310002_Webenabled_services_development_with_respect_to_serviceorientation_paradigm/links/5528c2640cf29b22c9bcb350.pdf#page=185
- [5] Python.org. (2019). Welcome to Python.org. [online] Available at: <https://www.python.org/> [Accessed 8 Dec. 2019].
- [6] Kivy.org. (2019). Kivy: Cross-platform Python Framework for NUI. [online] Available at: <https://kivy.org/#home> [Accessed 8 Dec. 2019].
- [7] Firebase. (2019). Firebase. [online] Available at: <https://firebase.google.com/> [Accessed 8 Dec. 2019].
- [8] Android Developers. (2019). Documentation | Android Developers. [online] Available at: <https://developer.android.com/docs> [Accessed 8 Dec. 2019].
- [9] Djangoproject.com. (2019). The Web framework for perfectionists with deadlines | Django. [online] Available at: <https://www.djangoproject.com/> [Accessed 8 Dec. 2019].
- [10] Pallets. (2019). Flask. [online] Available at: <https://www.palletsprojects.com/p/flask/> [Accessed 8 Dec. 2019].
- [11] Duolingo. (2019). Learn a language for free. [online] Available at: <https://www.duolingo.com/learn> [Accessed 8 Dec. 2019].
- [12] Lehovetzki, N. (2019). AB Math games - fun educational apps for iOS and Android. [online] Ab-math-games.com. Available at: <http://www.ab-math-games.com/> [Accessed 8 Dec. 2019].
- [13] App Store. (2019). Maths Skill Builders - Ireland. [online] Available at: <https://apps.apple.com/ie/app/math-skill-builders-ireland/id550663008> [Accessed 8 Dec. 2019].
- [14] IXL Learning. (2019). IXL | Online maths and English practice.. [online] Available at: <https://ie.ixl.com/> [Accessed 8 Dec. 2019].
- [15] Devdocs.io. (2019). DevDocs — HTML documentation. [online] Available at: <https://devdocs.io/html/> [Accessed 8 Dec. 2019].
- [16] Devdocs.io. (2019). DevDocs — CSS documentation. [online] Available at: <https://devdocs.io/css/> [Accessed 8 Dec. 2019].
- [17] Php.net. (2019). PHP: Documentation. [online] Available at: <https://www.php.net/docs.php> [Accessed 8 Dec. 2019].
- [18] Devdocs.io. (2019). DevDocs — JavaScript documentation. [online] Available at: <https://devdocs.io/javascript/> [Accessed 8 Dec. 2019].

- [19] Oracle Help Center. (2019). Java Documentation - Get Started. [online] Available at: <https://docs.oracle.com/en/java/> [Accessed 8 Dec. 2019].
- [20] Android Developers. (2019). Download Android Studio and SDK tools | Android Developers. [online] Available at: <https://developer.android.com/studio> [Accessed 8 Dec. 2019].
- [21] Docs.microsoft.com. (2019). Microsoft SQL documentation - SQL Server. [online] Available at: <https://docs.microsoft.com/en-us/sql/?view=sql-server-ver15> [Accessed 8 Dec. 2019].
- [22] Twinkl. (2019). ROI Teaching Resources - Infants, Juniors, Seniors. [online] Available at: <https://www.twinkl.ie/> [Accessed 8 Dec. 2019].
- [23] Mathematics, T. (2019). Transum. [online] Transum. Available at: <https://www.transum.org/> [Accessed 8 Dec. 2019].
- [24] NCCA. (2019). Maths Curriculum | NCCA. [online] Available at: <https://www.ncca.ie/en/primary/primary-developments/maths-curriculum> [Accessed 8 Dec. 2019].
- [25] Nrich.maths.org. (2019). NRICH - Mathematics Resources for Teachers, Parents and Students to Enrich Learning. [online] Available at: <https://nrich.maths.org/> [Accessed 8 Dec. 2019].
- [26] Curriculum. (2019). Home. [online] Available at: <https://www.curriculumonline.ie/Home/> [Accessed 8 Dec. 2019].
- [27] Haylock, D. and Thangata, F. (2007). Key concepts in teaching primary mathematics. Los Angeles: SAGE.
- [28] Folensonline.ie. (2019). Welcome - FolensOnline. [online] Available at: <https://www.folensonline.ie/> [Accessed 8 Dec. 2019].
- [29] Irish National Teachers' Organisation. (2019). The Irish National Teachers' Organisation - The Oldest and Largest Teachers' Trade Union in Ireland. [online] Available at: <https://www.into.ie/ni/help-advice/> [Accessed 8 Dec. 2019].
- [30] Pdst.ie. (2019). Mental Maths | PDST. [online] Available at: <https://pdst.ie/mentalmaths> [Accessed 8 Dec. 2019].
- [31] Nzmaths.co.nz. (2019). Homepage | NZ Maths. [online] Available at: <https://nzmaths.co.nz/> [Accessed 8 Dec. 2019].
- [32] Thinkingtoys.ie. (2019). Bee Bot - Rechargeable. [online] Available at: https://www.thinkingtoys.ie/index.php?route=product/product&product_id=138 [Accessed 8 Dec. 2019].
- [33] Agilemodeling.com. (2019). Core Practices for Agile/Lean Documentation. [online] Available at: <http://www.agilemodeling.com/essays/agileDocumentationBestPractices.htm> [Accessed 8 Dec. 2019].
- [34] ReQtest. (2019). Agile Documentation - 6 Hacks for Documenting in Agile Projects. [online] Available at: <https://reqtest.com/agile-blog/agile-documentation/> [Accessed 8 Dec. 2019].
- [35] Atlassian. (2019). Kanban - A brief introduction | Atlassian. [online] Available at: <https://www.atlassian.com/agile/kanban> [Accessed 8 Dec. 2019].

[36] Trello.com. (2019). Trello Tour. [online] Available at: <https://trello.com/en/tour> [Accessed 8 Dec. 2019].