The Application Development of a Relational Frame Theory Game

By

Emeka Nwaenie

Project unit: PJE
Supervisor: Farzad Arabikhan

April 2020

Abstract

The purpose of this project was to create an application that explores the exciting new realm of relational frame theory. This would be accomplished in the form of a game, that will hopefully inspire players to delve deeper into the world of relational games.

Additionally, the project details the development of the game application and the important phases involved in the project such as the gathering of requirements, designing of the application, implementing and testing of the system and finally the evaluation of the entire project.

Acknowledgements

I’m thankful for my supervisor Farzard Arabikhan for his unparalleled level of wisdom and guidance, as well as the consistent support throughout the entirety of the project. The additional deadlines set by him, allowed me to stay focused throughout the project.

Furthermore, a special thanks to my family & friends, who have consistently, rooted for me throughout the project. Specifically, thank you Robert and Jannah, for keeping me accountable whilst completing this project.

Table of Contents

Table of Figures

List of Tables

1.0 Introduction

This chapter will succinctly introduce my final year project, the introduction will include the problem statement, which is the sole purpose of this project intends to solve, As well as the project's overall aims and objectives. Also, this section of the paper will address the evaluation criteria and project assumptions and constraints.

1.1 Problem Statement

The human brain and intelligence has fascinated me since childhood, and I’ve consistently been interested in ways to improve it, which has subsequently led me to brain training and how it can impact cognition. The brain training industry has an estimated market size of over $10 billion and is bound to experience continuous growth ("Cognitive Assessment and Training Market by End-user and Geography - Forecast and Analysis 2020-2024", 2020). Brain training apps are used by people of all ages and can be used by the healthy or those with cognitive impairments. Despite the abundance of brain training/logic games on the market, many that lack scientific scrutiny. Several apps have little to no brain or cognitive benefits to the user (Kable et al., 2017).

In my personal experience many apps that I have encountered lack any real-world benefits, I get better at the game but my ability to solve problems or navigate throughout the world remains unchanged. After doing intense research many of these brain training apps are not based on any credible theory that can back up its claims of improving human cognition. Therefore, I set out to look for apps that have a theoretical and practical basis for improving human intelligence (if possible).

After discovering information about relational frame theory and how it may explain human cognition, I searched for apps that were based on this theory. I have come across a few that seem to successfully implement the theory such as RaiseYourIQ.com and Relational Frame Training on the Android App store. Many studies have supported the efficacy of relational frame theory and have demonstrated how it can improve users' overall intelligence which have been measured by multiple types of tests (Cassidy Et el 2016).

However I've encountered a few problems using these apps, firstly RaiseyourIq has a paid subscription, which creates a huge barrier of entry, it also has limited functionality, for instance, you're only allowed to play more than 5 levels a day and it’s only available via the internet. Whereas Relational training on Android is a poorly designed app, that lacks any challenging levels. So, the author thought it would be wise to develop an android based app that can be freely used, to further test the efficacy of this theory.

1.2 Solution

The fundamental problem is that there is a lack of easily accessible, well developed relational frame theory training games on mobile devices. To solve this issue, I must develop an android application, which can allow for users to test out this relational frame theory through a well-designed app.

1.3 Project Aims

The project aims to develop a high-quality Android App, that will allow the user to play a fun intuitive game based on the Relational Frame theory. The project aims Constitutes the Following:

- Allow users to test a relational frame theory game that has many progressive levels and many game modes

- Allow users to track how long they have played and see statistical analysis

- Develop a database that can store user information.

This app will serve as the perfect getaway for people to test out this theory and determine whether it has any brain-boosting benefits.

1.4 Project objectives

- Research Relational Frame Theory
- Analyse and Evaluate similar Applications
- Decide what the functional and non-functional requirements are
- Determine the usability requirements
- Develop an android application
- Test and evaluate the developed application

1.5 Project Assumptions

- Users will have an Android Phone
- Users have a touch screen device

1.6 Project Limitations

1. Time Management
The project may be severely compromised if the author does not properly manage time accomplishing this project.

2. Balancing other studies and Course Works
The author must also make sure that assignments and deadlines for other modules are met. If the author cannot balance successfully, the quality of the project will be compromised.

3. Technical Difficulties
Technical difficulties such as loss of data, programming errors, or viruses could slow down or even halt the completion of this project.

4. Lack of Mobile Development Experience
Because this is the first time the author has undertaken such a project. It means that the lack of experience could mean a lower quality app.

1.7 Project Evaluation Criteria

The evaluation will involve users testing the application. The criteria will evaluate whether the app was reliable and easy to use. Additionally, it will be evaluated against the project requirements and user requirements.

1.8 Project Report Framework

Chapter	Description
1.0 Introduction	This part of the report introduces the problem and details how the author intends to find a solution through its project aims and objectives. This section also outlines the project assumptions, limitations, and the evaluation criteria.
2.0 Literature Review	This Section covers relevant background research and how it can be utilized in the project.
3.0 Methodology	Examines different software development methodologies and which one best suits the author's project.
4.0 Requirements Analysis	This chapter contains how the author generated the requirements and what methods were used to elicit those requirements.
5.0 Design	This chapter includes information about the design of the artifact. examples include low/high fidelity prototypes, database design, logo designs.
6.0 Implementation	This section discusses the steps involved in creating the app and any problems faced during the development cycle.
7.0 Testing	This section details the results obtained in the testing process.
8.0 Evaluation	This section compares the requirements to the actualized project.
9.0 Conclusion	Review the project and to discuss potential plans.

Table 1. Report Structure

2.0 Literature Review

2.1 Introduction

The purpose of this chapter is to detail background research that was conducted, specifically research on relational frame theory. Additionally, this section will discuss relevant information about android mobile/app development.

2.2 Method

How the author acquired the relevant resources was by using a plethora of tools. for instance, the University of Portsmouth library service called Discovery and google scholar. The author was adamant in using specific key terms to avoid extraneous information, for instance, using terms such as “Relational Frame Theory” to find useful information for the artifact.

2.3 Relational Frame Theory

Relational Frame theory (RTF) was originally developed by Steven Hayes (Hayes 1991). This theory is a psychological explanation of human language and Cognition. RTF states that human cognition and our ability to communicate are contingent upon our capacity for creating and identifying relational frames or ‘links’ between stimuli. In other words, humans have the innate ability to state relationships between things. (Cullinan & Vitale, 2009).

2.3.1 Relational Frames

Three main relational frames have been identified. They can be said to be the underlying basis of all language, (Blackledge 2003).

1. Mutual entailment
Is the relational frame where stimulus A leads B, without stimulus B being explicitly stated its connected to A. Humans can imply from this relationship that A is connected to B, and b is also connected to A (Barnes - Hoems 2017). For example, Mike is older than Lucy. From this relationship, one can presume that Lucy is Younger than mike. Relation: A is Older than B, Thus B is younger than A.

2. Combinatorial entailment
Is a relational frame that states if A relates to B and B relates to C, then A and C are related (Barnes - Hoems 2017). For example, if Mike is in front of Lucy and Lucy is in front of James, then it can be derived that Mike is in front of Lucy and James is behind Lucy. Relation: A in front of B, B in front of C. Therefore, C behind B and B behind A.

3. Transformations of Functions.
This frame States that a stimulus can have multiple functions depending on the context. Additionally, the stimulus function can also change based upon random relation to that stimulus Dymond, Simon; Barnes, Dermot (1995). For example, A is less than B and B is Less than C. Despite the stimulus being neutral if we were to give a label to stimulus C, for

instance, calling it “Valuable”. Then A would be ‘less Valuable’ or ‘worse’ than C based on the relationship between the stimuli.

2.3.2 Relational Frame Theory Applied Research on Intelligence.

Studies show by intentionally testing and training individuals’ relational frames skills, a massive amount of cognitive improvement can occur. These improvements can be measured by IQ tests and increased academic performance. (Amd, M., & Roche, B. 2018).

According to another study, that had children from age 11-12 and 15-17 as participants in an intensive training intervention. By training these specific relations, the same, opposite, more, and less. Significant increases in Iq were recorded as well as improvements in numerical and verbal reasoning for almost all children (Cassidy Et el 2016). Which adds to the growing body of research that training these skills could potentially improve our underlying cognitive abilities.

2.4 Mobile Applications

Mobile applications generally are computer software designed to be used by mobile devices. Applications can come in many forms and serve many needs. Research states as of 2019 204 billion mobile apps have been downloaded (The State of Mobile 2020). This next section will talk about how the author decided what software and programs and principles were best to use to create the artefact.

2.4.1 Native

Native applications refer to apps that have been designed for a specific platform or device. For example, apps designed for the IOS will be written in Swift or Objective C, whereas apps designed for Android will be written in java (Rouse, n.d).

Additionally, because native apps are designed for a specific platform or device, it allows developers to take full advantage of the hardware, it is easier to fully access the features of the hardware, like the microphone, accelerometer etc, (Saccomani, 2020). This is accomplished by platforms providing app developers standardized SDKs (software development kits), which are specialised tools (libraries, documentation, sample code, guides and manuals) that allow developers to optimise their application on a particular platform(Khan, 2020). Due to the greater degree of optimisation compared to other types of apps, native apps will run faster and be more responsive, resulting in greater user satisfaction.

Despite the many benefits of native applications, they are susceptible to long development times, this is because it requires more time for developers to program apps for every platform. e.g. developers would have to write code for IOS, then Android. (Khan, 2020)

2.4.1.1 Web

A web application usually abbreviated as ‘Web App’ is a computer program that executes specific functions using the web browser as its client (Nations, 2019). Web apps are essentially websites and can be accessed through a browser. Web apps work by utilizing a web server, an application server, and a database (Gibb, 2016). Typically, web servers manage requests from the client, the application server completes the requested task and the database stores all relevant applications (Rouse, n.d.).

One advantage of a web server is that they can run on multiple platforms irrespective of the device or operating system, unlike native applications where developers must develop apps from scratch, to cater to individual platforms (Stevens, 2018). This potentially can reduce costs for developers because they save time and money from not developing for different platforms, additionally because the same version exists on multiple platforms, less maintenance is needed.

However, one weakness of web applications is that it always requires users to have access to an internet connection. ("Advantages and Disadvantages of Web Applications", 1999). This decreases the accessibility of the app, especially if an internet connection is imperative to use the app. Also, web apps may be less optimized for the platform because unlike Native apps, web apps are not designed for a specific platform in mind, therefore they cannot fully utilize the functions of a platform (Kumur, 2012).

2.4.1.2 Hybrid

A Hybrid app is a blend between a native application and a web application. The core of this type of application is written using HTML, CSS And JavaScript, which is generally encapsulated within a native application (Griffith, 2019). Hybrid apps work by allowing web pages to be launched within the native application's embedded browser. It accomplishes this by utilizing the mobile Webpage view object, which allows web pages to be launched via the native application ("Hybrid Apps: An overview of Advantages, Limitations & Consequences for your Testing Phases", n.d.).

A Major Advantage of Hybrid Applications is that they enjoy many benefits that both web and native applications have. For instance, native apps are written once and can run anywhere similarly to Web Applications, this consequently saves the developers time/ money resulting in faster development Cycles. (Butusov, 2019). Additionally, because Hybrid apps are also native apps, Hybrid apps can be run offline, which is beneficial for those who can't always access the internet ("Hybrid Mobile App Development - Top 9 Benefits | Redbytes", 2019). This is an advantage because this increases the accessibility of the app which will improve customer satisfaction.

However, a disadvantage of Hybrid apps is they also require internet access, this means that the full capabilities of the application can’t be available to the user if they are offline (K, 2019). Additionally, because they are written for multiple platforms, this may lead to applications, not fully optimized for specific platforms. Hybrid apps also have an extra layer between the source code that allows for hybrid mobile frameworks ("Benefits and Disadvantages of Hybrid Mobile Applications - Brooks Canavesi", n.d.). A Lack of optimization and more code consequently results in slower applications, compared to native applications.

2.4.2 Mobile Interaction Design

Human-Computer Interaction also known as HCI, is a collaborative discipline that is concerned with the design, evaluation, and implementation of interactive systems, used by humans(users) (ACM SIGCHI, 1992). The author thought it was imperative to research best practices whilst creating an app because a successful app is an app that can be intuitively used by its users. HCI's main goal is human satisfaction and it accomplishes this by drawing knowledge from a multiplicity of different fields like operating systems, computer graphics, programming languages for the machine side of things, and social sciences, communication theory, psychology for the human side of things (AGM Sigchi 2014). Merging both sides successfully results in an application or devices that can be used and understood by the users successfully.

2.4.3 Shneiderman's Eight Golden Rules

To design software, design guidelines or heuristics must be followed as this gives the developers constraints in which they can work within and this often helps mitigate common design issues developers face, following best practices can also allow users to use an application more effectively because they will be used to these standards from using other apps. Shneiderman, a prominent American computer scientist and a pioneer in design thinking presented these 8 rules to adhere to (Shneiderman, 1998).

1. Strive for Consistency
Use familiar icons, menu layouts, user flows when the user is operating an app. Standardized layouts and procedures allow the user to transfer what they learnt on one page to another.

2. Enable Frequent users to use shortcuts
Shortcuts allow users to use the application more efficiently, as it allows users to speed the process of doing a task.

3. Offer Informative feedback
The system should always indicate where the user is, for instance, if the user is completing a series of questions and they get an answer wrong, the system should alert them that they answered incorrectly.

4. Dialogue
Dialogue is used to help the user and can also be a source of closure, e.g. a “Successful” or “You are logged in” message when the user logs into their account.

5. Error Handling
When an error is made the user should be alerted swiftly, steps should then be shown to the user to rectify the problem. The errors should not be technical. as technical errors would require prerequisite knowledge to solve them.

6. Permit reversal of actions
Design the system so that it allows users to reverse their actions.

7. Support internal locus of control
Allowing users to terminate the program at will.

8.Reduce short term memory load
Design the interface with simplicity so it can allow users to better retain information, as too much information can overload user memories, making it hard for them to use the app.

2.5 Java Development Language

Java is a class-based programming language that is designed to have little implementation dependencies as possible. Its purpose is to be written once and be able to run on all java supported platforms (WORA). Java is also the most predominantly used programming language today, with most android applications being written in java. (Neil Smyth 2017). Additionally, android applications can also be developed with C or C++, but java offers many benefits such as (Mindmapped 2015):

1. Simple to use:
Java is Easy to use compared to other programming languages like C++ for instance. Writing, compiling, and debugging can be done at ease.

2. Object-Oriented:
Allows developers to utilize concepts like class, objects, abstractions inheritance etc, this allows programs to be reusable.

3. Platform - independent:
A program that is written once can be used on any java compatible software

4. Secure:
Java was created with security being integral to its design. For instance, Java virtual machines have a unique identifier that makes sure the bytecode is secure before running it.

5. Multithreading capabilities:
Java supports multithreading, which allows a program to execute man tasks concurrently.

6. Robust:
Java is a very reliable programming language; its compilers are sophisticated as they can detect errors within code easily. Java also uses exception handling and garbage collection, to allow for greater efficiency in programming and running applications.

2.5.1 Justification for Android Development

The Author has decided to develop for Android because the author has numerous android devices, such as an Android Mobile phone and tablets. Having access to a wide range of android devices would make it ideal for testing on different platforms. Testing on different platforms will give the Author a Holistic view of the app and make it easier to detect any problems the apps could cause.

2.6 Database

The author felt it was necessary to include a database in the development of the application, this is because the database will store the user's data, submitted by the user. And save any progress made by the user.

2.6.1 Firebase

Firebase is a Realtime database, which is hosted in the cloud. It is a NoSQL database and allows information to be stored and synced between users in real-time. (Firebase Realtime Database 2020). Many android applications rely on firebase because of its many benefits (Ohyver 2019):

1. No SQL
In Firebase there are no tables, data is only in the form of JSON documents. Data in the no SQL database can be put in different servers, as it is cheaper to do so than upgrading the capacity of the current server. Also, there is not an over-reliance on one server, if one server crashes then the data will be transferred to another server.

2. Realtime Database
If one user updates information on the server then, data on Google server will update instantaneously. This allows for up to date real-time information for the users and the database.

3. Simplifies Backend
Firebase Realtime database, simplifies the upkeep of the database, only the code on the client-side must be changed (Android application, web application etc). Whereas SQL demands us to create code specifically for the server-side with programming languages like Ruby And PHP. Which further complicates the maintenance of the system.

Other databases for app development exist like SQLite however, SQLite, however SQLite does require additional SQL knowledge to use it effectively. ("Firebase vs SQLite | What are the differences?", 2020).

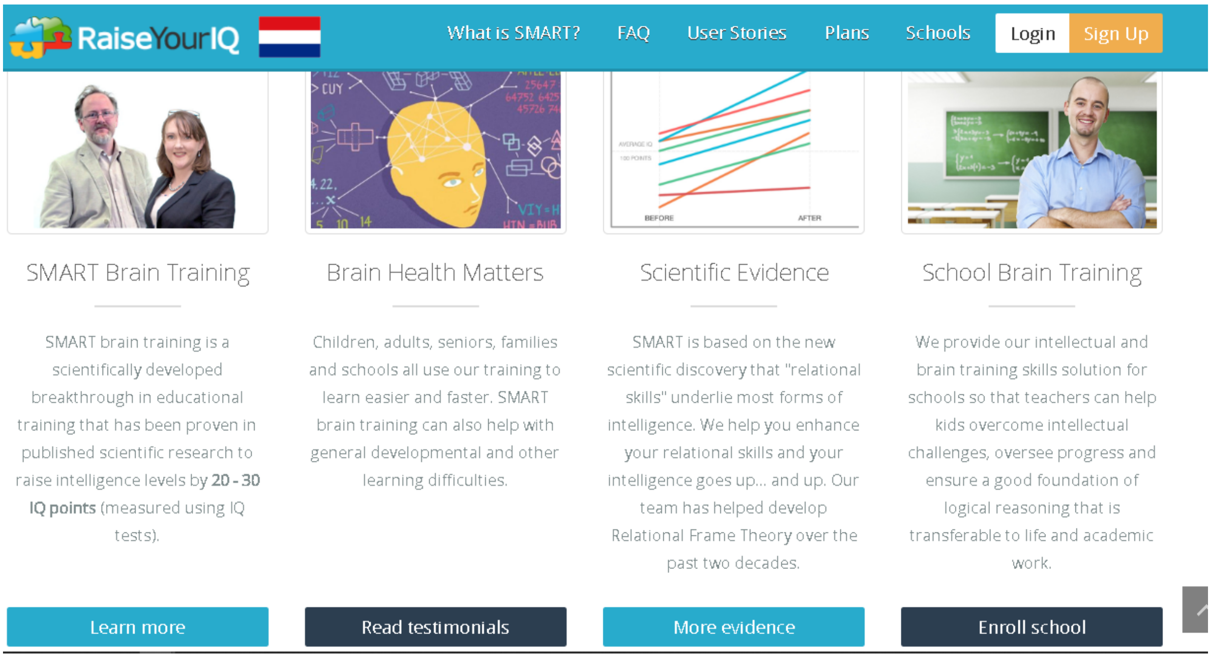
2.7 Similar applications

This section will discuss research about similar applications to the one the author intends on creating. These app are Android and web-based and will be evaluated thoroughly.

The importance of App market research was to obtain data from existing features that can then be utilized for this Application. There were not many android based apps, so the author felt it was necessary to look for web-based applications that were like the artefact.

2.7.1 RaiseyourIq

RaiseYourIq Is A brain training web-based app that is based on the scientific evidence that relational skills is the cognitive substrates of many forms of intelligence. The App also uses relational frame theory as the theoretical basis of its training system. The Game has 70 levels that progressively get harder and a variety of time-based game modes, for a further increase in difficulty. They have an old and new version, however, both versions require a paid subscription to use the full features of the product. Below is the old user home page of the Web app.



SMART Brain Training

SMART brain training is a scientifically developed breakthrough in educational training that has been proven in published scientific research to raise intelligence levels by **20 - 30 IQ points** (measured using IQ tests).

Brain Health Matters

Children, adults, seniors, families and schools all use our training to learn easier and faster. SMART brain training can also help with general developmental and other learning difficulties.

Scientific Evidence

SMART is based on the new scientific discovery that "relational skills" underlie most forms of intelligence. We help you enhance your relational skills and your intelligence goes up... and up. Our team has helped develop Relational Frame Theory over the past two decades.

School Brain Training

We provide our intellectual and brain training skills solution for schools so that teachers can help kids overcome intellectual challenges, oversee progress and ensure a good foundation of logical reasoning that is transferable to life and academic work.

Figure 1 - RaiseyourIq page

2.7.1.1 Key features

- A total of 70 levels progressively getting harder.
- Different types of levels, that train different relational Skills
- Different Timed Game modes
- Detailed Statistical Analysis
- First 15 levels free then membership to play the remaining 55

2.7.1.2 Performance

- Smooth running on Google Chrome

2.7.1.3 Interface design

Feature	Chrome
Navigation bar is fixed on every page	✓
Navigation Drawer when screen is small	✓

Table 2. Table Comparison regarding Interface Design of The RaiseyourIq web application

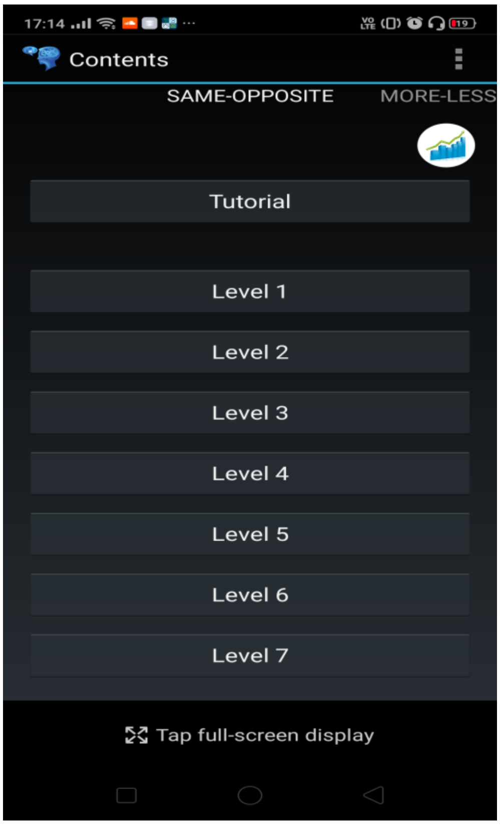
2.7.1.4 Analysis

RaiseYourIq is a great application that is cleanly designed and encapsulates what a good game is about, it does an adequate job in executing the relational frame theory in a fun and interesting way. The levels are somewhat challenging.

However, the biggest problem is the paywall. It costs 15 euros for access to all levels and extended features; this is the cheapest costing package. Additionally, it lacks native android support, meaning there are no android support applications of this app. Which reduces accessibility and portability of the app.

2.8 Relational training

Relational training is an android Mobile-based Training App, which offers 16 game levels to progress through. It has two game modes and has a simplistic statistics feature. The Application is also Free of charge. This is the home screen of the Application.



17:14

Contents

SAME-OPPOSITE MORE-LESS

Tutorial

Level 1

Level 2

Level 3

Level 4

Level 5

Level 6

Level 7

Tap full-screen display

Figure 2 - Relational Training homepage

2.8.1 Key features

Feature	Android
Allows users to play any level of their choice	✓
Can check performance statistics	✓
Provides tutorial instructions	✓

Table 3. Comparison Table regarding Key Features of Relational training

2.8.1.1 Performance

- The application is quite clunky and slow

2.8.1.2 Interface Design

- Design is bland.
- Not clear indication if the user is improving.
- Feedback Statistics are rather poor.
- Levels Are not challenging enough.
- The goals are unrealistic in the time frame given.

2.8.1.3 Analysis

The app does seem to provide a way for mobile users to play a game based on relation frames, however, the user interface is awful, the colour schemes also clash heavily. The lack of help documentation makes it hard to understand what does what in the app. The settings button is also unresponsive making the app near impossible to use for a long period.

Additionally, the in-game mechanics are quite poor, it gives you a duration to answer as many relational frames as possible, if it is incorrect it'll start the timer again. Because there is no obvious goal/target for the user to obtain it lacks any incentive to continue playing.

Moreover, because all levels are already unlocked it also comes as a drawback because there is no incentive to play for long periods if all features are already unlocked for you, This subsequently means that there are no clear indicators if users are getting better at the game.

The statistics are also very poorly designed, there is a lack of graph labelling, meaning it is hard to tell what axis represents what.

2.9 Literature review conclusion

The research conducted was used as a guideline for the project. To briefly summarize, the author reviewed existing research about HCI and why adhering to Shneiderman’s principles can offer a better experience for the user. Research into Java helped to reaffirm why it is the best for programming android apps. Additionally, researching what database would be ideal for this project helped the author have a more holistic view of the process of development.

Finally, the last section focused on app comparisons which gave the author many ideas to incorporate into the artefact. To conclude succinctly all research done has benefited the author greatly.

3.0 Methodology

This section will discuss an in-depth analysis of different software development approaches and state which approach will be best suited for this project. The author will then justify the chosen approach. Analysing and selecting the appropriate software development is imperative, as it fundamentally decides whether the project succeeds or fails. (Tshai, 2018)

3.1 Waterfall Approach

The Waterfall approach is arguably the most well-known methodology model, it was conceptualized by Winston Royce in 1970. This methodology entails that each section of the model must be completed in sequential order, thus for developers to move to the next step, it's a prerequisite to finish the stage that their currently on, once completed developers can move to the next task. There are no overlaps between sections/phases. (Royce, 1970)

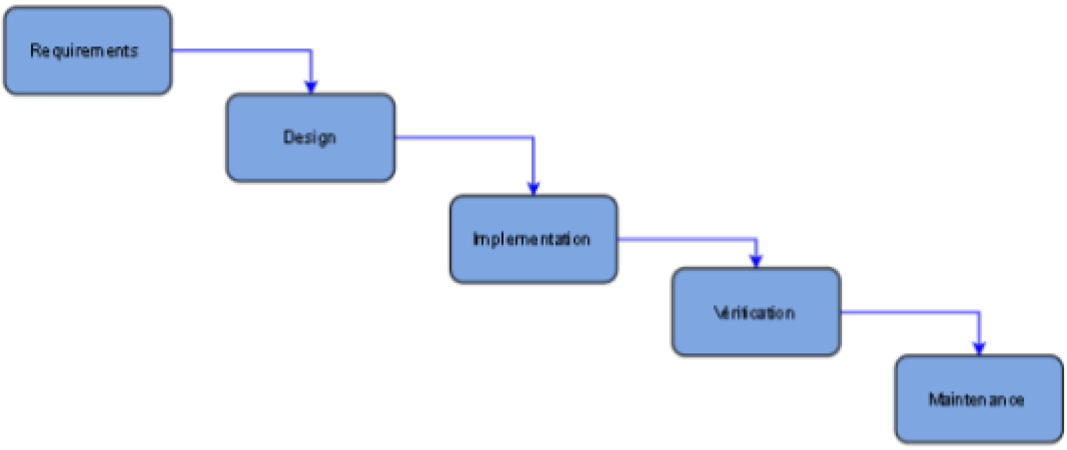


Figure 3 - Waterfall Model

Advantages

- This model is Linear therefore it is easy to implement because there are no overlapping stages. (Royce, 1970)

- A Lot of time can be allocated to each phase of development, allowing less room for error. Potentially reducing costs and saving time.

Disadvantages

- Because there's a lack of feedback from other stages, and due to the linear/sequential nature of the model, if mistakes are found during the later phases of development they cannot be rectified easily, meaning this model can consequently be very costly and time-consuming.

- New requirements/ Designs cannot be implemented in the current development process.

- Testing only occurs after the implementation stage. Meaning the system will most likely consist of many mistakes.

3.2 Prototype Model

The prototype model is a system development approach where a prototype is built and tested iteratively until an acceptable outcome is accomplished, this will be the basis of which the complete product will be developed from (Rouse, n.d.). It is best utilized in situations where the requirements are obscure ("Prototyping Model in Software Engineering: Methodology, Process, Approach", n.d.).

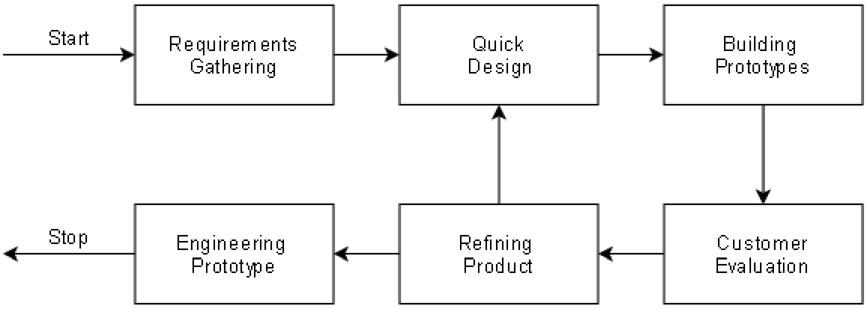


Figure 4 - Prototype Model

There are 4 main types of approaches within the Prototype model; ("Prototyping Model in Software Engineering: Methodology, Process, Approach", n.d.)

1. Rapid Throwaway Prototype:
The rapid throwaway prototype is derived from the initial requirements and is developed quickly to see how the app will look visually. The feedback gained on the prototype will help create changes to the requirement. A new prototype is then created on these newfound requirements. Additionally, the developed prototype will not be the finalised prototype. An advantage of using this technique is that it allows developers to explore novel ideas and get feedback for customer requirements instantly.

2. Evolutionary Prototyping
Evolutionary prototyping is the method where a prototype is developed and incrementally refined based upon customer feedback. The refinement process is ongoing until the prototype is finally accepted. A benefit of using this approach is that it is helpful to use when the requirements are not easily understood. Additionally, it saves time, money, and effort, because the developers are only refining one prototype, rather than starting from scratch each time.

3. Extreme Prototyping
Extreme prototyping is generally used for web development. It involves 3 sequential phases.

1. A basic prototype is developed, which includes all existing web pages in the HTML format.

2. Secondly, developers can simulate data processes using prototype services layers.

3. The services are implemented into the final prototype.

4. Incremental Prototyping

This prototyping method entails that the final product is dissected into smaller prototypes and worked on individually, the smaller prototypes are then amalgamated into a single product. The benefit of this method is it can help reduce feedback time between the users and the development team. Additionally it allows for greater meticulousness of the prototype because it's broken up into smaller pieces, allowing the development team to have a greater focus on the smaller parts, thus improving the quality of the final prototype once all parts have been combined again.

Advantages

- As prototypes are created very early in the product, it allows for instant feedback from customers. Thus, improving the quality of the requirements and specifications of the product. ("The Advantages & Disadvantages of Prototyping - Rapids Reproductions", n.d.)

- Errors can be detected much earlier; this reduces the time and costs of fixing those mistakes. ("Prototype Model with Advantages and disadvantages - All Major Seven Subjects", 2019)

- Missing functionalities can be discovered. ("The Advantages & Disadvantages of Prototyping - Rapids Reproductions", n.d.)

Disadvantages

- If a developer spends too much time creating prototypes, it could result in wasted time and become costly. Spending more time on creating complex systems means less time for refinement, a situation could occur where developers have spent all their time on a prototype but later in the development, there are disagreements of the prototype details. ("The Advantages & Disadvantages of Prototyping - Rapids Reproductions", n.d.)

- A possible problem is that there is maybe too much fluctuation of the requirements every time the prototype is being reviewed by consumers.

- Customers may mistake the prototype for the final product, this may cause a biased review because their expectations may be set too high. Additionally, users may grow fond of the prototype features that are not in the finalised system. ("The Advantages & Disadvantages of Prototyping - Rapids Reproductions", n.d.). User confusion may lead to biased or distorted feedback, which creates unreliable requirements for developers.

3.3 Agile methodology

Agile development was first conceptualized in 2001. The overarching idea of agile development is that customers are the highest priority and it is imperative to satisfy them through early and continuous iterations of the product/software, throughout the software development life cycle of the project. ("What is agile development?", 2019).

According to the agile manifesto, 4 main principles exist ("Principles behind the Agile Manifesto", n.d.).

1. The individual and team interactions are more important than the processes, tools or techniques used by developers.
2. Working software is more important than comprehensive documentation.
3. Customer collaboration is more important than contract negotiation.
4. Responding to change is more important than following a strict plan.

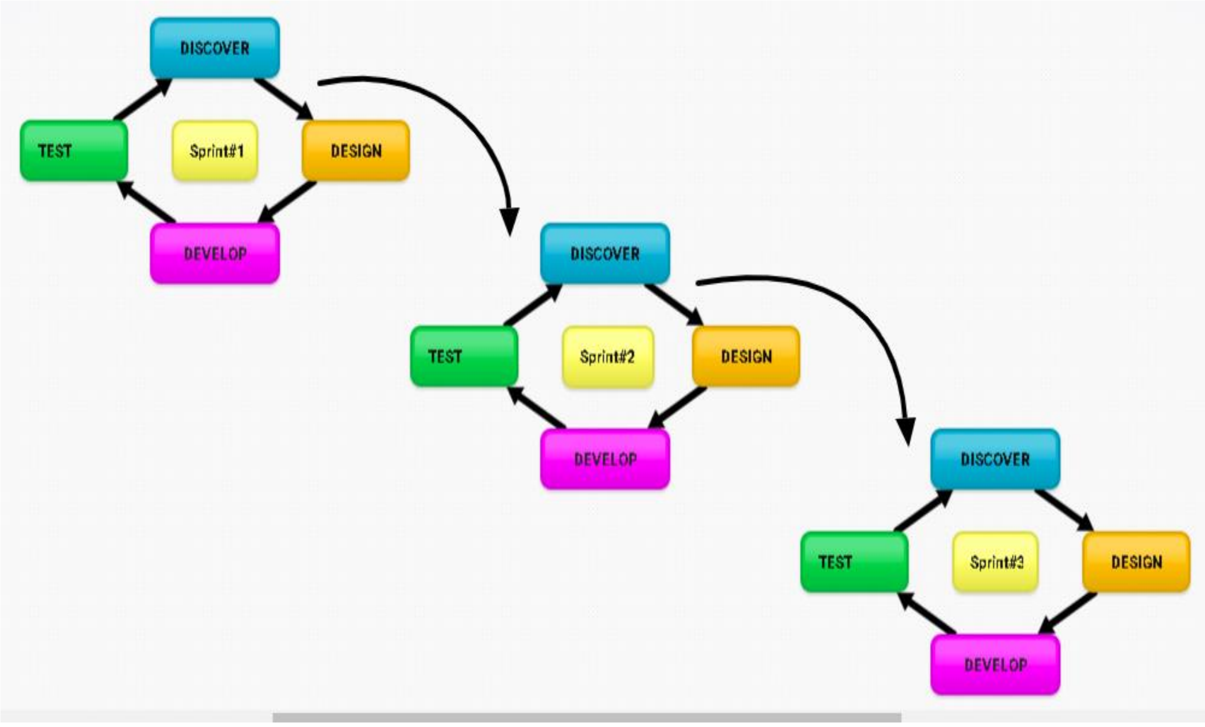


Figure 5 - Agile methodology (Reich,2013)

Advantages

- One of the biggest advantages of the Agile development model is its ability to change. Agile allows teams to focus and respond to change, it prioritizes work when it matters most.

- Additionally, developers are constantly interacting with stakeholders and consumers, this means fewer errors are made as the product is constantly being made to meet consumer expectations.

Disadvantages

- Agile development could create an indefinite loop of the project, which would result in a waste of time and money.

- Due to Agile's unpredictable nature, it's hard for developers to determine costs, how long the project will be, or resources needed to complete the project. The problem is exacerbated further when the project is more complex. (Lynn, n.d.)

- Additionally, Agile is more strenuous for developers because they constantly must communicate with each other and stakeholders (Olic, 2017). Developers must be permanently focused and flexible because they are uncertain of what will remain the same or what needs to be changed.

3.4 Justification for chosen Approach

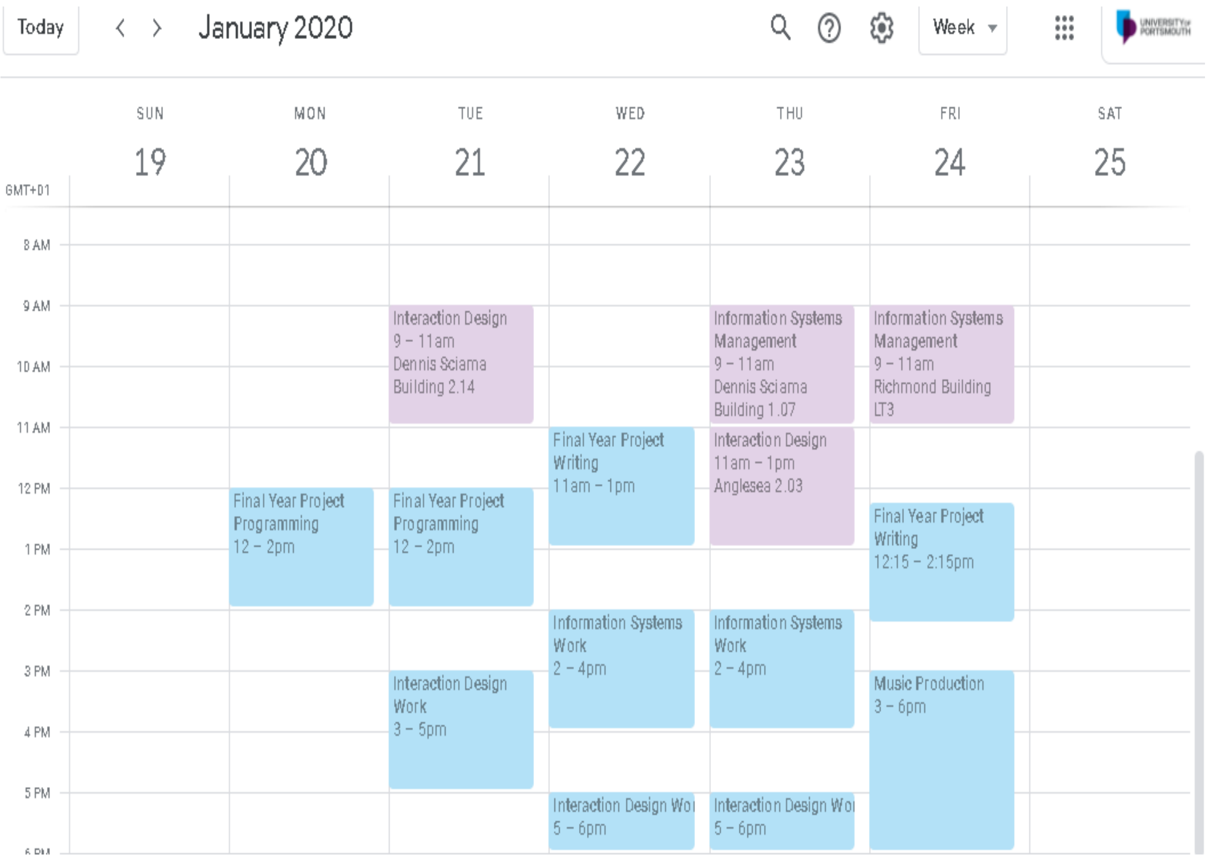
Despite the many benefits of the waterfall method, the author felt the waterfall method would be too inflexible to accommodate this artefact. Additionally, because of the author's lack of experience creating mobile applications, the lack of feedback given by the Waterfall method would cost the author a lot of time rectifying these problems later down the line. Moreover, the author felt that agile development is best suited for team-oriented projects.

After much consideration, the author has settled on using the prototype model, specifically its evolutionary variant. The guidance it offers is very relevant to this project. The author has never partaken in such a project before so, the constant feedback and guidance would help the author immeasurably. The author will be learning how to program in android, constantly improving the prototype and evolve the artefact based on new skills and techniques learned. The user feedback (from users and Supervisors) would suit the author's style of learning and complement the student and supervisor interaction.

3.5 Project planning

The final deadline to submit the final year project and the source code was set to the 7th of May 2020. However, despite the abundance of time, the author felt the need to manage time effectively to produce high-quality work. The author aimed to finish the artefact by the end of February. This will give the author ample time to focus on other aspects of the project, such as the writing of the dissertation and testing the app with users. The author created a Gantt chart at the beginning of the project (Appendix A PID), however, the Gantt chart underestimated the time it would take to complete certain tasks. So, the author revised the Gantt Chart (Appendix C Gantt chart revised)

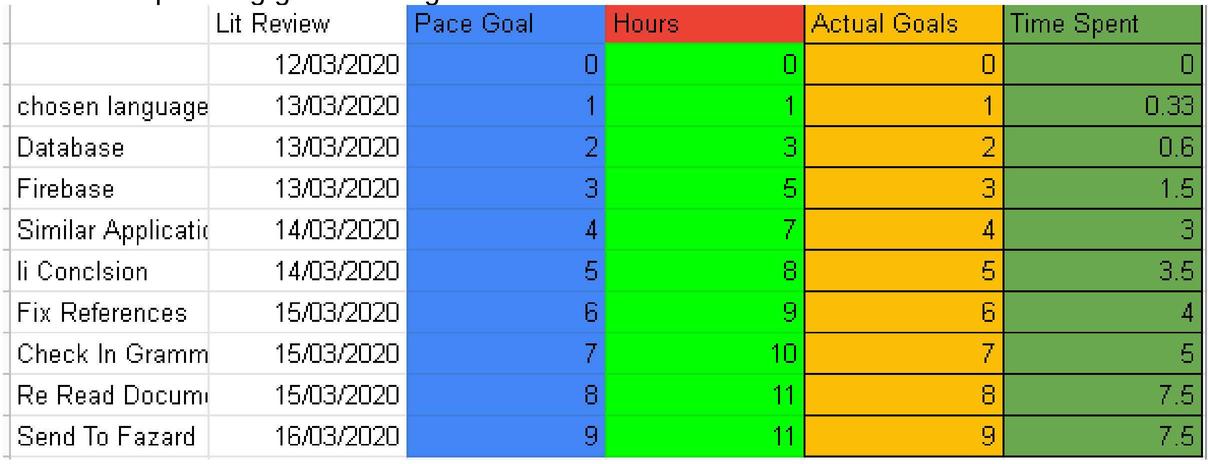
Additionally, the user used a multiplicity of time management techniques, such as creating a timetable to time box tasks and using backwards planning techniques to estimate how long a task would take.



SUN 19	MON 20	TUE 21	WED 22	THU 23	FRI 24	SAT 25
	Final Year Project Programming 12 – 2pm	Interaction Design 9 – 11am Dennis Sciama Building 2.14	Final Year Project Writing 11am – 1pm	Information Systems Management 9 – 11am Dennis Sciama Building 1.07	Information Systems Management 9 – 11am Richmond Building LT3	
		Final Year Project Programming 12 – 2pm	Information Systems Work 2 – 4pm	Interaction Design 11am – 1pm Anglesea 2.03	Final Year Project Writing 12:15 – 2:15pm	
		Interaction Design Work 3 – 5pm	Interaction Design Wo 5 – 6pm	Information Systems Work 2 – 4pm	Music Production 3 – 6pm	
				Interaction Design Wo 5 – 6pm		

Figure 6 - Personal Time management Table

Backwards planning goal tracking table.



	Lit Review	Pace Goal	Hours	Actual Goals	Time Spent
	12/03/2020	0	0	0	0
chosen language	13/03/2020	1	1	1	0.33
Database	13/03/2020	2	3	2	0.6
Firebase	13/03/2020	3	5	3	1.5
Similar Applicati	14/03/2020	4	7	4	3
Ii Conclsion	14/03/2020	5	8	5	3.5
Fix References	15/03/2020	6	9	6	4
Check In Gramm	15/03/2020	7	10	7	5
Re Read Docum	15/03/2020	8	11	8	7.5
Send To Fazard	16/03/2020	9	11	9	7.5

Figure 7 - Backwards planning goal tracking table.

.

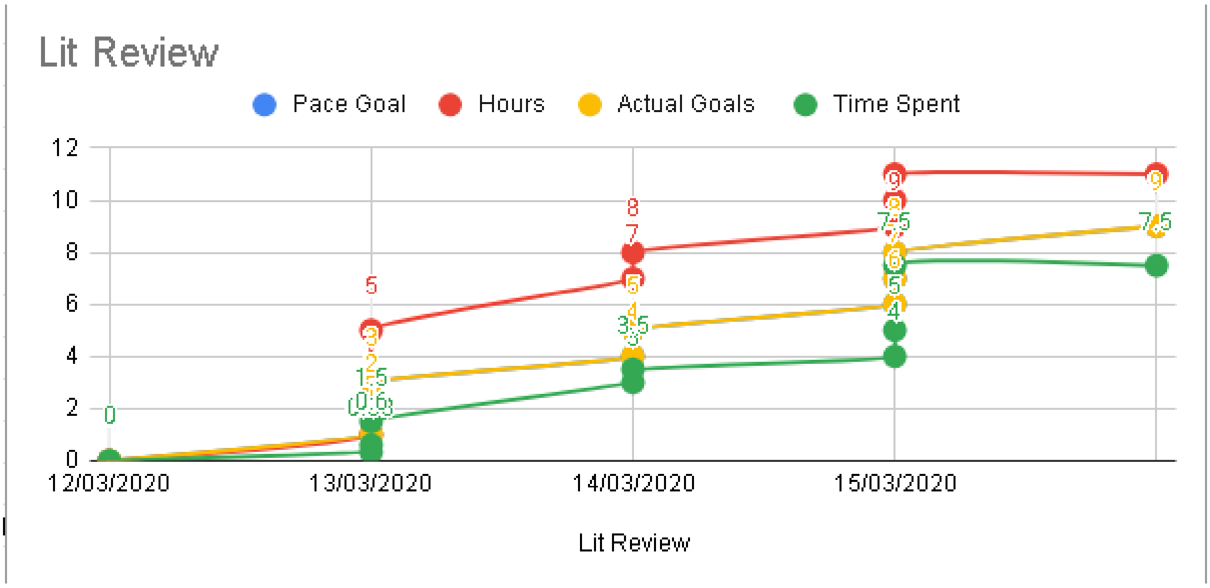


Figure 8 - Backwards planning graph

3.6 Software Development Kit (SDK)

A Software Development Kit commonly referred to as SDK is a set of programs and software tools used by developers to create applications for a specific platform (Valdellon, 2019). SDK's usually consist of; libraries, an IDE (integrated development environment), sample code and documentation that helps developers program apps more efficiently (Poetker, 2019).

Based upon the author's research Android Studio was the most recommended SDK, based upon its intuitiveness and relatively easy learning curve, which was ideal for the author, as the author lacks experience in mobile development. Additionally, Android Studio is free and has a large body of documentation and tutorials online. Additionally, Braintree is also another popular android SDK, however, this is a paid service. Eclipse is also frequently used to develop android applications; however, Android studio is packed with a plethora of benefits, making it more of an ideal choice for Android software development.

4.0 Requirement Analysis

This chapter will be outlining the functional and non-functional requirements of the project. These requirements were obtained through research using a plethora of requirement gathering techniques, such as interviews, personas, interviews. The author felt that using multiple research techniques was imperative because it will help the author create an app that caters specifically to user needs.

4.1 Stakeholders

Identifying the stakeholders is important because these are the users that will be the one using the application. so, it is important to identify them so the author can create an app that caters to their needs. A list of identified stakeholders are below:

- Children
- Students
- Adults
- Project Supervisor
- Developer (Author)

4.2 Requirement elicitation techniques

Requirement elicitation techniques is a procedure used to collect vital information from stakeholders, it is the basis in documenting the requirements of an application. (Labs, 2019).

4.2.1 Research

As stated in the literature review (chapter 2), the author did background research on similar apps to gain a better understanding of what has been already done, and what improvements needed to be made.

4.2.2 Mind map

A mind map was used to brainstorm potential requirement ideas. The benefits of using mind maps are that it helps simplify complex issues (Greene, n.d.). Simplifying the problem allowed the author to generate many ideas, which could be further developed for the requirements. The following questions helped in developing the mind map which can be found in appendix D:

- Who is most likely to use this system?
- Why would they use this system?
- When will they use the system?
- What information would the system require?
- Why would the not use such a system

4.2.3 Persona

This Chapter will discuss personas and justify why they were used. Personas are used to model, summarize, and communicate research about a group of people who have been observed or researched in some way (Goltz, 2014). Personas allow thousands of users to be represented by one character, that represents the general views and behaviour of a group. This is extremely helpful because it allows the author to have a better understanding of the user's goals and motivations. Thus, helping the developer create an app that can complement the lifestyles of the persona.

4.2.3.1 Persona one

John is a Secondary school student and goes to Bacon's College. His favourite subject is maths and Physics. he love’s strategic games like chess and Go. In his free time, he likes to solve puzzles and try out logic games online.

However as of recently logic games on android have been boring him, he finds them too easy and a lot of them are repetitive. He would usually play logic games on his phone on the way to school. However, because of the lack of diversity of logic games on android, his journey times have become dreadful and boring.

He would like an Android logic game that has a new concept to him and it's challenging. He would prefer it to be an application because it means he can play it whilst he is travelling to school. and, in class when he gets bored with the teachers.

4.2.3.2 Persona two

Mary is a Second-year Psychologist Student that attends Portsmouth University. Her model requires her to write a 2000-word research paper on any psychological theory. She has decided to base her report on Relational Frame theory and the connection of intelligence

After Researching about Relational Frame theory, a lot of papers have suggested that playing games based on the theory could have an impact on intelligence. As she is writing a paper on this, she has decided to try to find logic games based on this theory. However, there are a limited amount of games that are based on this theory. She finds a web-based game, but this requires a subscription. As she is a university student, she thought it wouldn't be wise to pay a £60 subscription just to play a game for a 2000-word research which was only worth 15% of her final grade.

Mary, however, is still very interested in getting hands-on research for her paper. She would like a game based on relational frame theory that was free to play.

4.2.3.3 Persona three

Chris is 36 years old; he is an entrepreneur and a self-proclaimed biohacker. His interests in the self-improvement of the body and the mind have led him to relational frame theory. He reads the research papers indicating that it could potentially improve intelligence, he is willing to investigate such claims.

Chris is constantly travelling due to business meetings, he does not use a desktop much, therefore he operates his business mainly from his laptop, tablet, and phone. But prefers using his tablet because it is more portable than a laptop and larger than a phone. However, he has noticed that a handful of relational frame games on the market are poorly designed and have a lot of bugs for his tablet. Making these games nearly unusable. They lack clear instruction and overtime the levels become so boring because of how easy the levels are.

He would like a well-designed game that is based on the relational frame theory that is easy to use on his tablet and phone. Additionally, he would also like a game that is engaging and challenging.

4.2.4 Questionnaire

Questionnaires were used to generate and gather requirements. The questionnaire was created and deployed towards the beginning of the requirement elicitation process. The questionnaire had a mixture of closed and open-ended questions. The questions were simply framed for users to understand them.

The questionnaire was developed online and sent to anonymous participants. 31 anonymous participants completed the questionnaire. Despite the small number of completed questionnaires, users gave consistent feedback which increased the coherence of the results. This gave the author a stable foundation to create requirements. The questionnaire can be found in appendix E

4.2.4.1 Results

Based upon the results the author discovered the two most important qualities of any application to a user with speed and offline capability being the highest demanded quality of an app (See Figure 9). This emphasizes the need for the app to be native because a native app offers the benefits of both working offline and being the fast type of application because it optimizes the phone's capabilities. Complementary to this statistic 83% of people would prefer to play a brain training game on a mobile phone rather than on a web application. This further iterates the need for a mobile developed app. (See figure 10). Finally, because most users prefer using mobile-based apps to play their brain games the author wanted to know why. The results show that the biggest reason why people prefer using their mobile to play games is due to phones being more convenient to use (See figure 11).

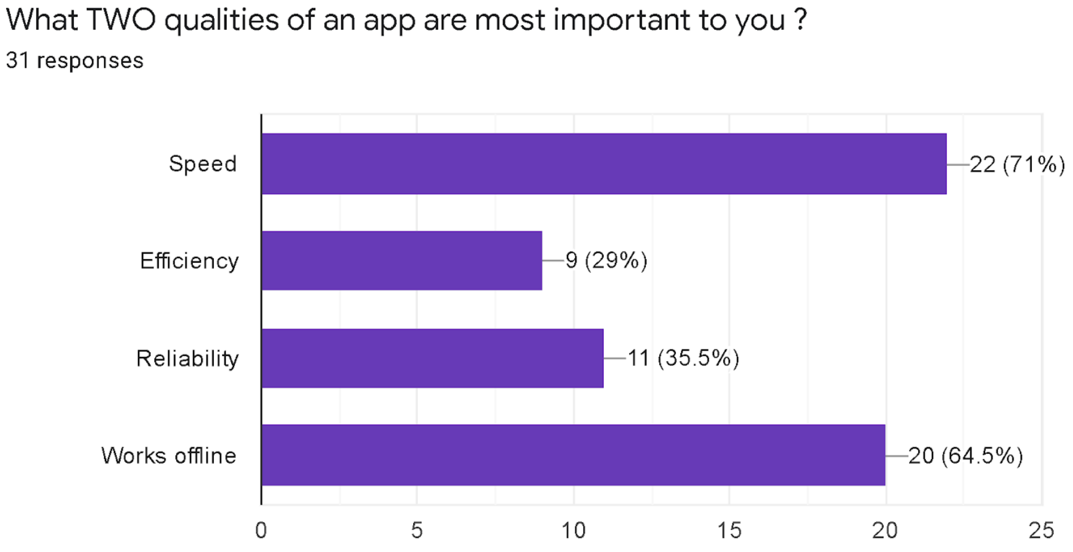


Figure 9 - Questionnaire results - What two qualities of an app are most important to you

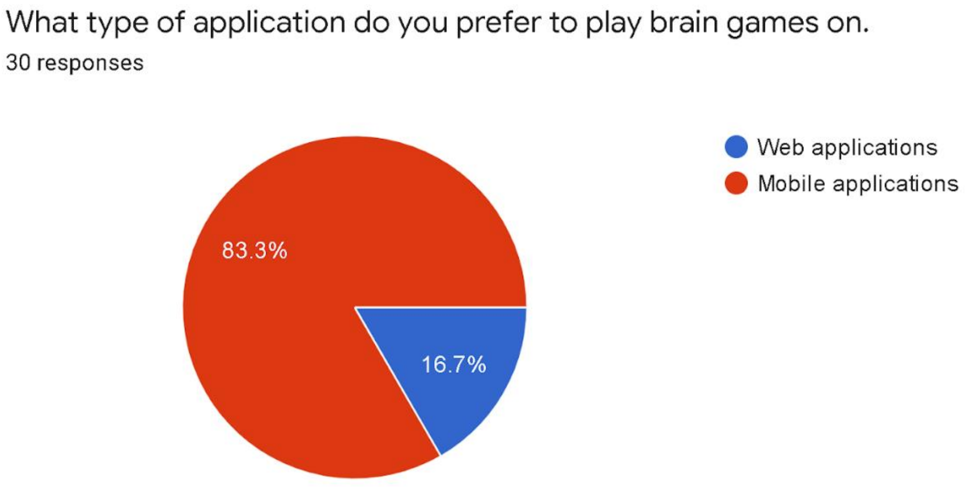


Figure 10 - Questionnaire results - What type of application do you prefer to play brain games on.

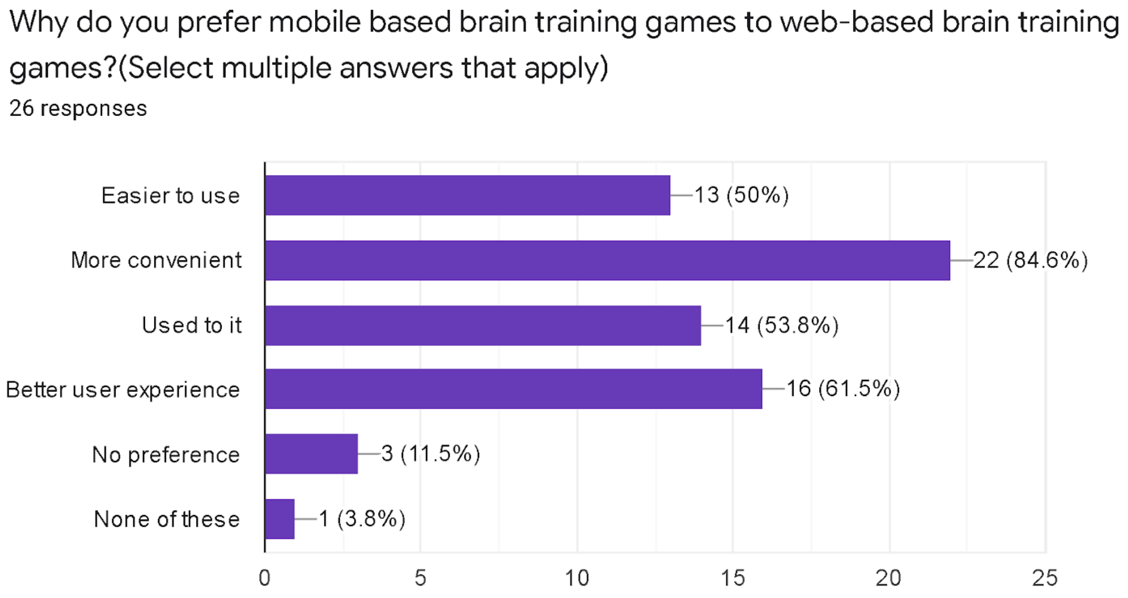


Figure 11 - Questionnaire results - Why do you prefer mobile based brain training games to web-based brain training games

4.2.5 Interviews and Observations

Additionally, 2 interviews were conducted to further gain a greater understanding of the users. The interview consisted of open-ended questions and observing the users interact with the low-fidelity prototype (See 5.2). The purpose of this was to demonstrate what the potential end design of the app may look like. Additionally, this method was used to see if further improvements to the functionality of the app can be made. The results from the interviews can be seen in appendix F of this report.

4.2.6 Task Analysis

Task analysis is a technique used to learn about users by observing them in action, to gain a better understanding of how they perform tasks and achieve their goal. ("Task Analysis | Usability.gov", n.d.). The purpose of this diagram is to show what steps users need to go through to complete a task. The main reason for the construction of this diagram was to identify the most important tasks for users. By identifying the most important tasks it allows the author to design the app more efficiently and focus on creating a smooth experience in completing tasks.

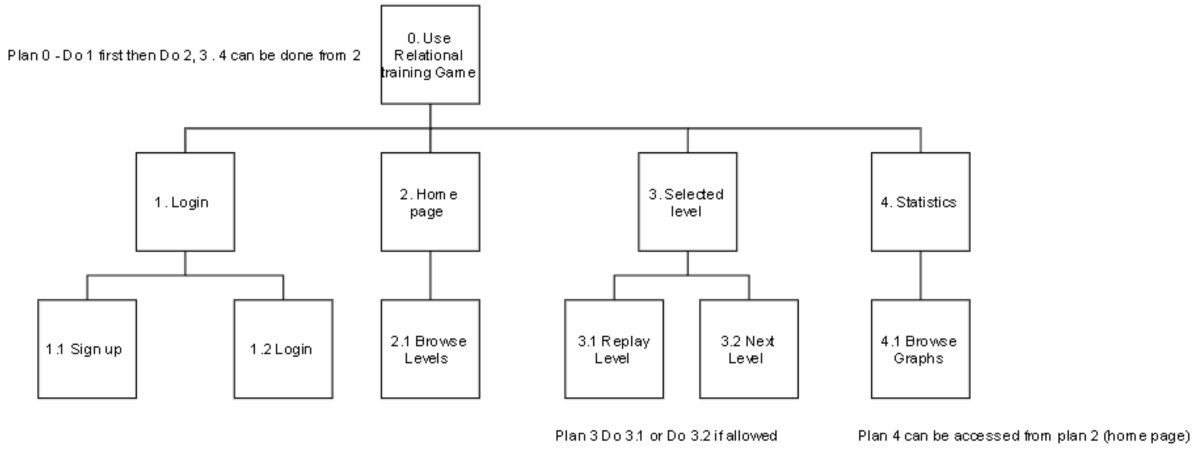

Plan 0 - Do 1 first then Do 2, 3 . 4 can be done from 2
0. Use Relational training Game
1. Login
2. Home page
3. Selected level
4. Statistics
1.1 Sign up
1.2 Login
2.1 Browse Levels
3.1 Replay Level
3.2 Next Level
4.1 Browse Graphs
Plan 3 Do 3.1 or Do 3.2 if allowed
Plan 4 can be accessed from plan 2 (home page)

Figure 12 - Hierarchical Task Analysis Diagram.

4.3 Functional Requirements

Functional requirements are requirements that describe how a product must act ("Functional and Nonfunctional Requirements: Specification and Types", 2018). Creating functional requirements is important as it outlines what the product must do. The author decided that using the MoSCoW methodology would be beneficial, it will allow the author to have a written representation of what requirements are the most important. Moscow methodology includes the following categories.

- **M** - **Must:** non-negotiable requirements.
- **S** - **Should:** requirements that are important but not vital.
- **C** - **Could:** requirements that would be desirable to have.
- **W** - **Won't:** requirements that won't be a part of the final product but could potentially be used for future work.

(Madsen, 2019)

The functional requirements were gathered from the results received from the various requirement elicitation techniques. The MoSCoW methodology allowed the author to categorize and prioritize requirements in a meaningful way. The table below shows the categorized requirements in descending order of importance.

MoSCoW	**Functional Requirements**
Must	• The application must allow users to sign up for an account. • The application must allow users to log in to account. • The application must be free to play • The application must have clear instructions to play the games. • The application must work with a database that stores user data (emails, passwords, scores). • The application must have multiple game levels that increase in difficulty. • The application must have a progress bar telling users how many questions are left. • The application must inform users when they get an incorrect answer.
Should	• The application should have a countdown feature when a level is being played. • The application should remember the user when they return to the app (no need to log in every time the user opens the app). • The application should provide a statistical analysis of the player. • The application should have different types of relational frames that can be played by a user.
Could	• The application could allow users to customize the colour schemes and themes of the App. • The App could allow players to change game settings, e.g. decreasing the countdown timer, to make it harder. • The application should allow the user to set a notification for the time they would want to play the game.
Won't	• Have an IOS counterpart • Have email sign-ups with Gmail, Facebook. • Have advertisements

Table 4. Functional requirements using MoSCoW

4.4 Non-functional Requirements

Non - Functional Requirements are requirements that discuss a system's operational capabilities and constraints/restrictions. ("What is Non-Functional Requirement? Types and Examples", n.d.). The table below presents the non-functional requirements.

MoSCoW	Non-functional requirements
Must	• The application must be able to run on Oppo reno Z and Google Nexus 7 tablet. • The application must be based on HCI principles e.g. be easy to use, clear navigation. • The application must give the user's feedback when necessary.
Should	• This app should work on any android device.
Could	N/A
Won't	N/A

Table 5. Non-Functional requirements using MoSCoW

4.5 Usability requirements

Usability requirements are requirements concerning how well a product is designed for it to be easy to use (Spacey, 2017). The importance of usability requirements ultimately decides whether an app can be usable, by creating strong usability requirements, it may guarantee high customer satisfaction. The table below discusses the usability requirements of this artefact.

MoSCoW	**Usability Requirements**
Must	• The system must be easy to navigate through. • The application design/colour schemes/fonts should be consistent throughout the application.
Should	• The application should adhere to android design standards. • The application should inform users with instructions to play the game. • The application should appear vibrant.
Could	N/A
Won't	N/A

Table 6. Usability requirements using MoSCoW

5.0 Design

This chapter will discuss the initial designs for the project and justify the design decisions that were made. This chapter will also briefly discuss how users will interact with the app and how the application will interact with a database.

5.1 Paper Designs

The first process in designing the app was to have an idea of what the main pages may look like. The justification for creating paper prototypes was to allow the user and author, have a visual representation of the application. Because the author had decided to use the Prototype Model to develop this artefact the designs were continuously changing.

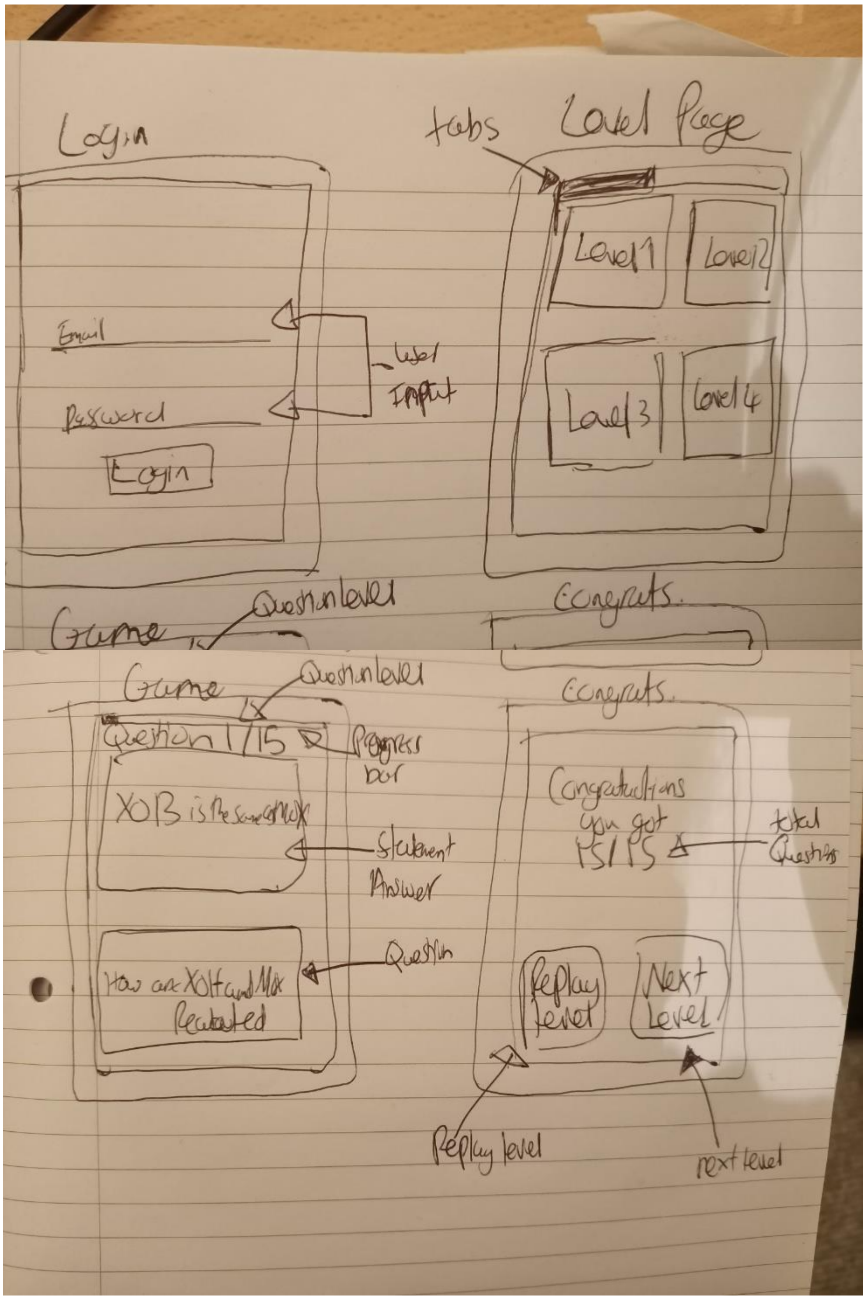


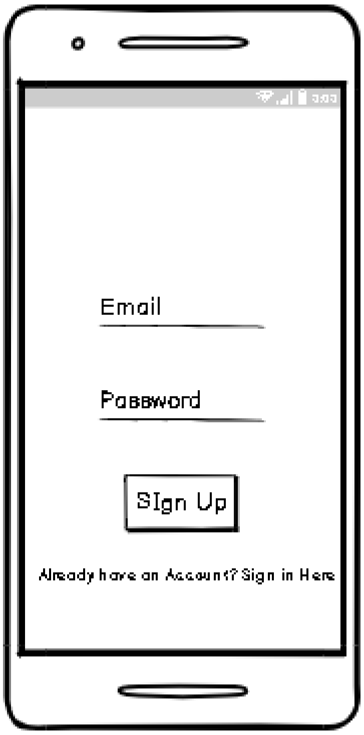
Figure 13 - Paper Prototypes

5.2 Low fidelity Designs

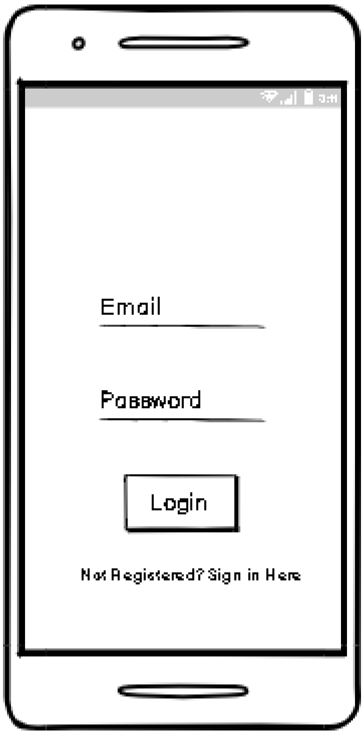
Low Fidelity prototypes are low tech designs to demonstrate how the app will behave (Esposito, 2018). The advantages of using low fidelity prototypes include instant feedback on the design, once the author has received feedback on the initial prototypes the iterative process can begin.

Additionally, the author has implemented HCI principles in the design of these low fidelity prototypes, because as discussed in the literature review, implementing HCI principles can improve overall satisfaction for users.

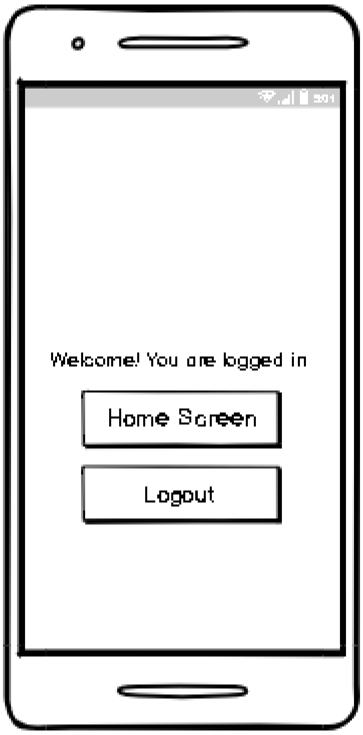
5.2.1 Registration Screen

 Email Password SIgn Up Already have an Account? Sign in Here	**Registration** - This is the sign-up page; users can sign up using a username and password. **Text field Email** - This section allows users to enter their desired email. The word count is 20. **Text field Password** - This field allows users to enter up to a 15-character password. **Sign up Button** - Once a user has entered their details, the sign-up button will add the user's information to the database. **Sign in Link** - If the user already has an account, they can select this text link at the bottom to take them to the login page.

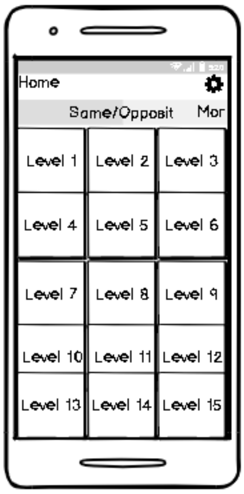
5.2.2 Login Screen

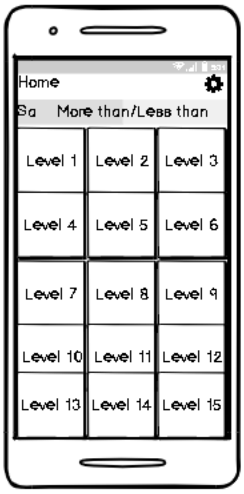
	**Login**- This is the login page; users can log in using a username and password. **Text field Email** - This section allows users to enter their desired email. The word count is 20. **Text field Password** - This field allows users to enter up to a 15-character password. **Sign up Button** - Once a user has entered their details the login button will verify the user against the database, to see if the user exists. **Sign up Link** - If the user hasn't got an account, they can select this text link at the bottom to transport them to the sign-up page.

5.2.3 Home Transitionary page

	**Home transitionary page** - is an alert page that allows users to logout the app or continue to the home screen. **Text -** Text welcomes users informing them that they've logged in. **Home screen Button** - this button is a link directly to the home page. **Logout Button** - this button allows user to log out of their account and it'll redirect them to the login page.

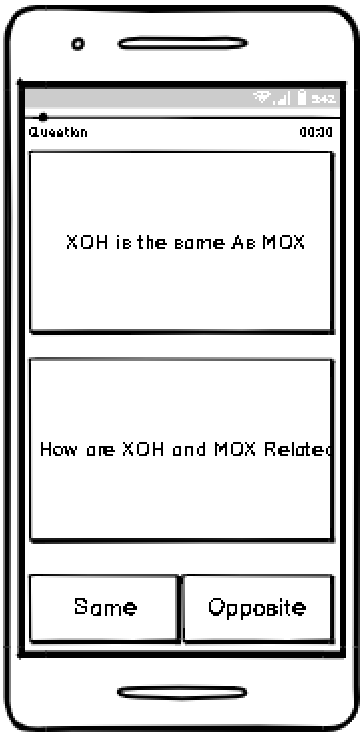
5.2.4 Level Home page

<table>
<tr>
<td>




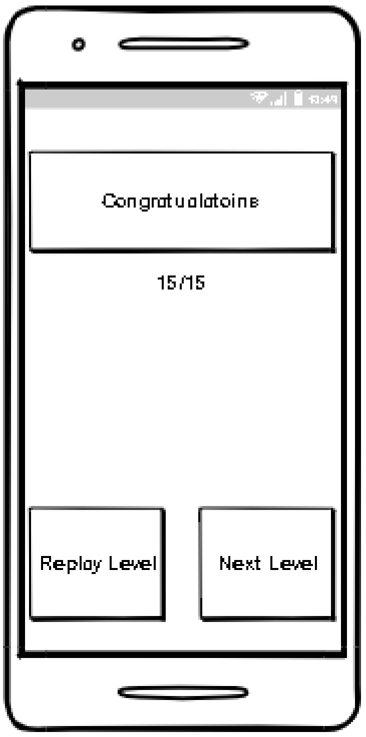
</td>
<td>
<b>Level Home page</b> - this is the homepage where all levels exist, in the left picture users are on the Same/Opposite page (Tab 1). By sliding from the right users can access the “More than less than” page (Tab 2).
<b>Levels</b> - when a user selects a level, they enter a relation frame game.
<b>Settings button</b> - the Cog button in the top left corner of the screen represents the settings, by clicking this button, user's will be transported to the settings page.
</td>
</tr>
</table>

5.2.5 Game Play

<table>
<tr>
<td>


</td>
<td>
<b>Game Play -</b> The screen showcases the play mechanics; users will select the same or opposite depending on what relation accurately answers the question.
<b>Progress bar</b> - the progress bar increases every time the user gets a question correct.
<b>Question Number</b> - Located in the top left corner, it notifies the user what question the user is on.
<b>Countdown timer</b> - each question counts down from 30 seconds.
<b>Statement</b> - this box presents the relational statement
<b>Question</b> - this box presents the question regarding the statement.
<b>Same</b> - the left button represents the same relationship
<b>Opposite</b> - the right button represents the opposite relationship
</td>
</tr>
</table>

5.2.6 Next Level Page



Next Level Page - this page is the next level page; users can see they're score and replay level or advance to the next level.

Congratulation box - this box tells users “Congratulations”

Score text - this text tells the users how many questions they got correctly

Replay level button - by selecting this button users can reply level.

Next level button - users can select this button to go to the next level.

5.3 Colour Scheme

An application must have a coherent colour scheme, colour schemes have the power to enhance the user experience and satisfaction (Babich, 2017). Thus, the author had to use colours that were simple and would induce relaxation whilst players used the app.

The author had settled on using a combination of 4 main colours. White is the primary colour of the app and some text (Figure 14), Green is the secondary colour (Figure 15), blue is the third colour, it's largely present in the gameplay(Figure 16) and black for the colour of some text (Figure 17).

Figure 14 - White

Figure 15 - Green

Figure 16 - Blue

Figure 17 - Black

The purpose of using these specific colours is, they are minimalistic, thus creating a cleaner user experience. Green adds to the playfulness of the app, emphasizing that this is a game to be enjoyed. Additionally, the contrast with the white background and the black text adds simplicity to the application. Blue was also intentionally used due to its relaxing properties (Cherry, 2018). Promoting relaxation increases the attentiveness of the user to play at the levels.

5.4 Database

The database chosen to create this application was firebase, as discussed in the literature review, firebase others a plethora of benefits that make it ideal for this type of project. The installation and setup of firebase is rather intuitive, additionally firebase requires no SQL knowledge which is an advantage for the author (Bagsariya, 2020). Figure 18 below demonstrates the system architecture.

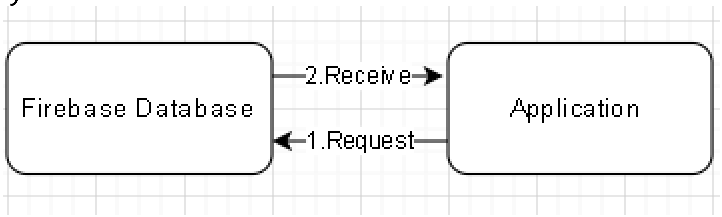


Figure 18 - System Architecture

5.5 Use Case

A use case diagram is a graphical presentation that captures the main functionalities of an app ("What is Use Case Diagram?", n.d.). Creating a use case diagram allows the author to visualize how users may interact with the application.

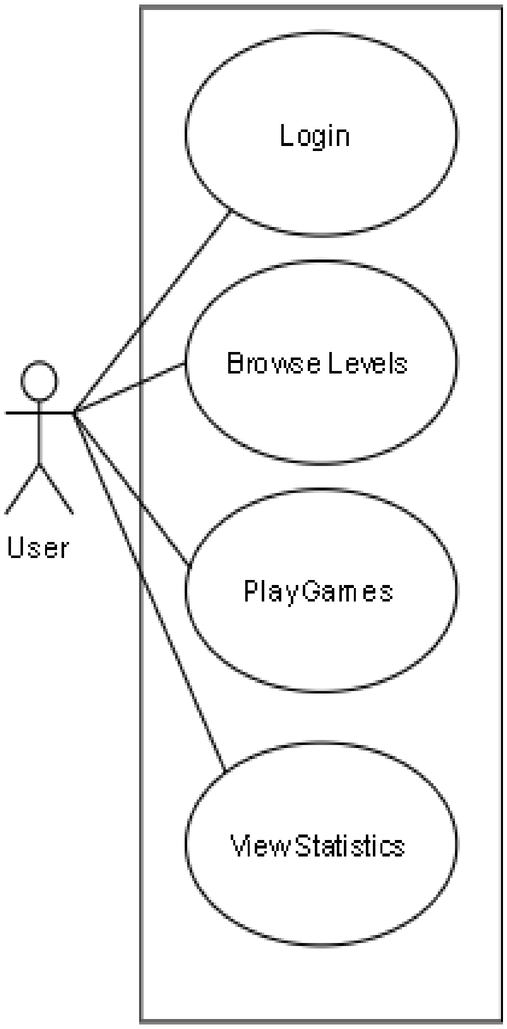


Figure 19 - Use Case

6.0 Implementation

The purpose of this chapter is to discuss the implementation of the artefact and detailing the process of how it was developed. Furthermore, this includes how the author actualized the initial designs discussed in the Design chapter. As the evolutionary model was used the author will

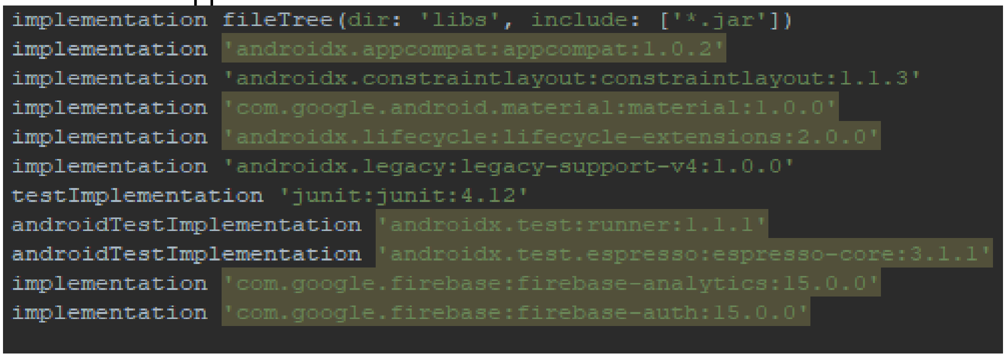
6.1 Initial Prototype

The evolutionary model entails the use of multiple models that improve iteratively, the tasks below indicate the original functions of the prototype.

- Installing Firebase
- implementing authentication
- Transition Page
- Level Home Page
- Gameplay
- Progress bar

6.1.1 Implementing Firebase

The author felt it was wise to start the app development by implementing the database first because doing so would allow for user authentication and database storage. Figure 20 below displays the code that allows Firebase libraries to be imported into the application's Gradle File. This consequently connects the application to Firebase's cloud database. Any updates made in the app would now be reflected in firebase.

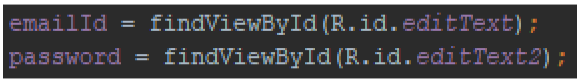


```
implementation fileTree(dir: 'libs', include: ['*.jar'])
implementation 'androidx.appcompat:appcompat:1.0.2'
implementation 'androidx.constraintlayout:constraintlayout:1.1.3'
implementation 'com.google.android.material:material:1.0.0'
implementation 'androidx.lifecycle:lifecycle-extensions:2.0.0'
implementation 'androidx.legacy:legacy-support-v4:1.0.0'
testImplementation 'junit:junit:4.12'
androidTestImplementation 'androidx.test:runner:1.1.1'
androidTestImplementation 'androidx.test.espresso:espresso-core:3.1.1'
implementation 'com.google.firebase:firebase-analytics:15.0.0'
implementation 'com.google.firebase:firebase-auth:15.0.0'
```

Figure 20 - Firebase Importation Code(implementation)

6.1.1 App Authentication

One main requirement for this application is to allow users to create an account, to save gameplay data for that user. To actualize this requirement the author had to create input fields to allow users to enter a username and password. The "**emailId**" and **"password"** field is where users can enter the username and password, these are linked to XML edit text containers which allow for text input. (See Figure 21).



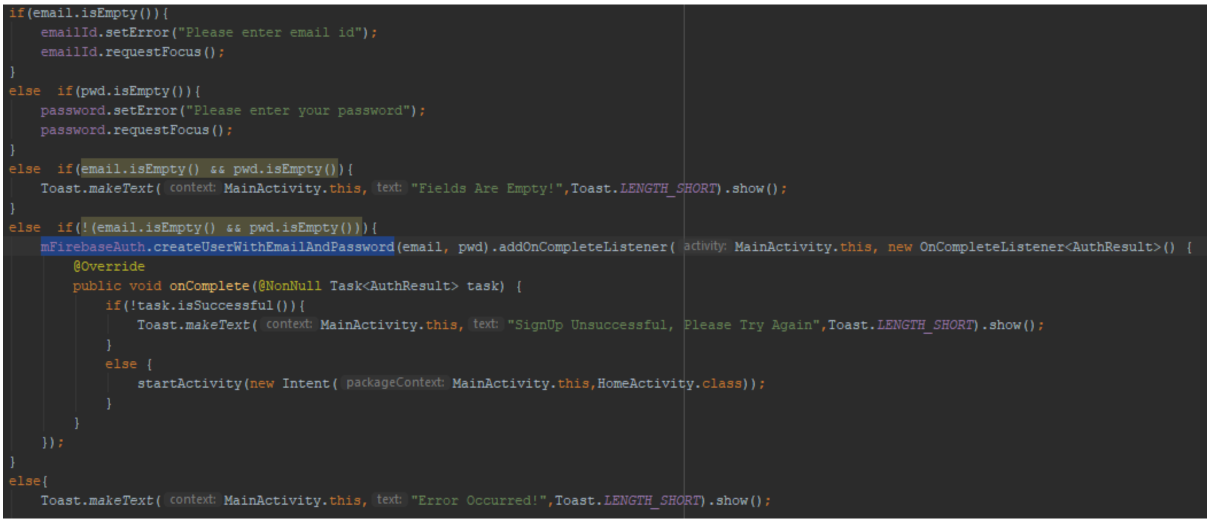
```
emailId = findViewById(R.id.editText);
password = findViewById(R.id.editText2);
```

Figure 21 - Email and Password ID

The code **“mFirebaseAuth = FirebaseAuth.getInstance();”** is where the application will call firebase to verify whether the user details can be found in the database. If The email and password fields are empty then this line of code is initialised **“else if (email.isEmpty() && pwd.isEmpty())”**, which alerts the viewer that they must enter an email and password, this idea was informed by Shneiderman's 8 golden rules, specifically - informative dialogue (Shneiderman, 1998)

"mFirebaseAuth.createUserwithEmailandPassword(email, pwd)" is the line of code that represents the creation of an account (See Figure 22), after the user has successfully made one. If successful, a new activity will start and take the user to another page. Each user that has successfully signed up will have their information stored in Firebase alongside a unique ID.

As previously stated, every time a user signs in, their details will be checked against the Firebase Database if the details do not match the **“else”** clause will kindly tell the user the information they have entered is incorrect.



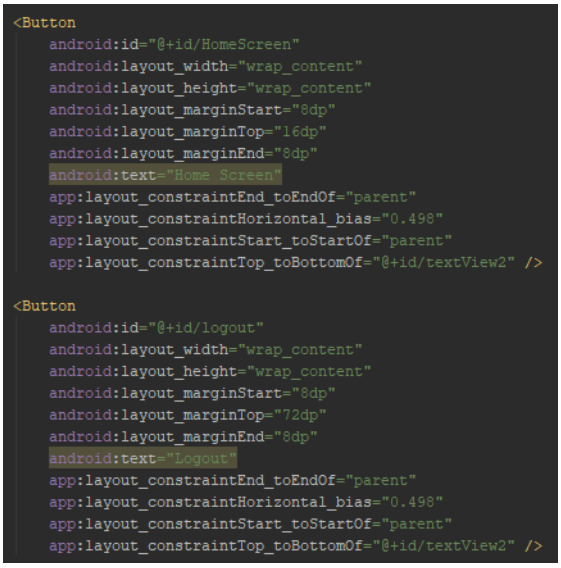
```
if(email.isEmpty()){
    emailId.setError("Please enter email id");
    emailId.requestFocus();
}
else  if(pwd.isEmpty()){
    password.setError("Please enter your password");
    password.requestFocus();
}
else  if(email.isEmpty() && pwd.isEmpty()){
    Toast.makeText( context: MainActivity.this, text: "Fields Are Empty!",Toast.LENGTH_SHORT).show();
}
else  if(!(email.isEmpty() && pwd.isEmpty())){
    mFirebaseAuth.createUserWithEmailAndPassword(email, pwd).addOnCompleteListener( activity: MainActivity.this, new OnCompleteListener<AuthResult>() {
        @Override
        public void onComplete(@NonNull Task<AuthResult> task) {
            if(!task.isSuccessful()){
                Toast.makeText( context: MainActivity.this, text: "SignUp Unsuccessful, Please Try Again",Toast.LENGTH_SHORT).show();
            }
            else {
                startActivity(new Intent( packageContext: MainActivity.this,HomeActivity.class));
            }
        }
    });
}
else{
    Toast.makeText( context: MainActivity.this, text: "Error Occurred!",Toast.LENGTH_SHORT).show();
```

Figure 22 - Firebase Authentication

6.1.2 Transition page

The transition page is the screen dedicated to informing users that they've successfully logged in and given the user the option to proceed to the home screen or log out of they’re account. The purpose of this page was to give the user the option to logout of their account. as stated in the literature review providing the user with the ability to undo actions is important in creating an efficient application (Shneiderman, 1998)

The first step in creating this page was to create an XML sheet (See Figure 23), where the option to advance to the home page or logout can exist. Each button has a unique identifier which is represented by the code **“item android:id=@+id/”.**



```
<Button
    android:id="@+id/HomeScreen"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginTop="16dp"
    android:layout_marginEnd="8dp"
    android:text="Home Screen"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView2" />

<Button
    android:id="@+id/logout"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginTop="72dp"
    android:layout_marginEnd="8dp"
    android:text="Logout"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView2" />
```

Figure 23 - Transition page Xml File

Furthermore, navigation that permits users to travel from one activity to another is called the **“onClickListener”** method. This method allows users to click on the designated buttons. As previously mentioned, each item has a designated id. Links can then be made with the designated Id and an activity.

6.1.3 Level Home Page

This page consists of multiple levels users can play, the level boxes are large rectangles, which would make users with disabilities easier to see them. As mentioned earlier an **“onClickListener”** is used on every box which will allow users to enter the game activity (See Figure 24).

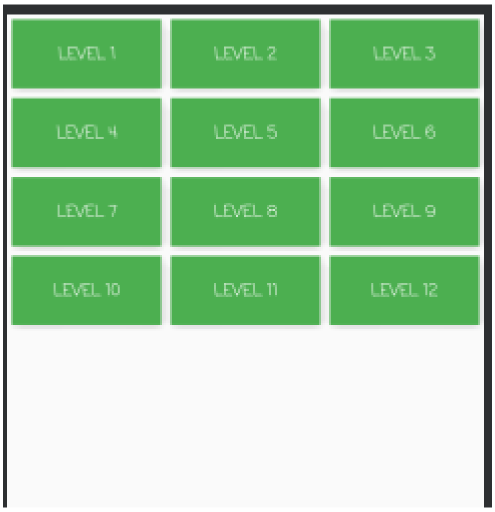
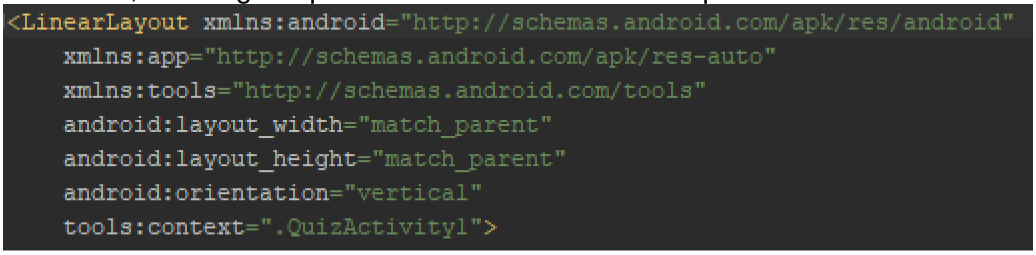


Figure 24 - Level Home Page

6.1.4 Game Mechanics

This stage pertains to the development of the game mechanics of the application. Firstly, an XML Sheet was deployed. The android layout was changed to linear layout (See Figure 25), which allows for children items to be placed on top of one another ("Linear Layout", 2019).

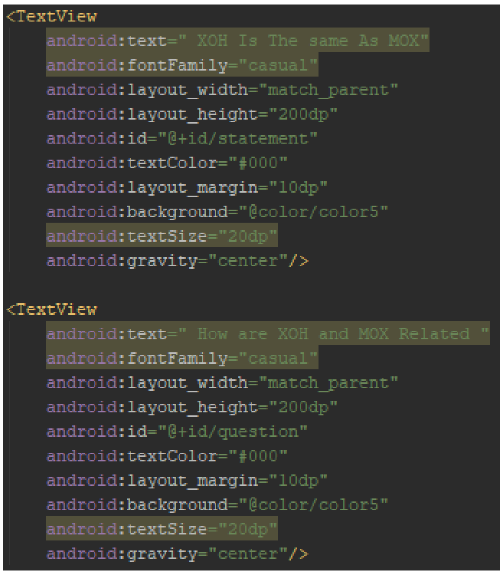
Additionally, the android orientation has been changed to vertical. This will turn the page layout into a row, allowing the placement of items to be on top and beside each other.



```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".QuizActivity1">
```

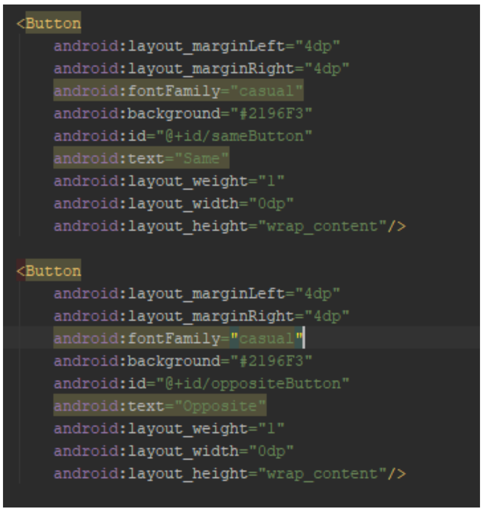
Figure 25 - Linear Layout

Within the linear layout container, two TextView containers were created; one for the question and one for the statement. Each TextView is paired with their own unique ID (See Figure 26). Two buttons were also created within another linear layout, both buttons represent the two options users can select to answer the questions **"Same"** and **"Opposite"** (See Figure 27).



```
<TextView
    android:text=" XOH Is The same As MOX"
    android:fontFamily="casual"
    android:layout_width="match_parent"
    android:layout_height="200dp"
    android:id="@+id/statement"
    android:textColor="#000"
    android:layout_margin="10dp"
    android:background="@color/color5"
    android:textSize="20dp"
    android:gravity="center"/>

<TextView
    android:text=" How are XOH and MOX Related "
    android:fontFamily="casual"
    android:layout_width="match_parent"
    android:layout_height="200dp"
    android:id="@+id/question"
    android:textColor="#000"
    android:layout_margin="10dp"
    android:background="@color/color5"
    android:textSize="20dp"
    android:gravity="center"/>
```

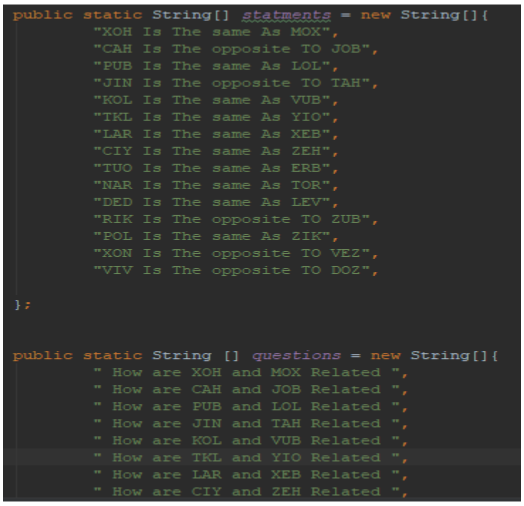
Figure 26 - Statement and Question XML text



```
<Button
    android:layout_marginLeft="4dp"
    android:layout_marginRight="4dp"
    android:fontFamily="casual"
    android:background="#2196F3"
    android:id="@+id/sameButton"
    android:text="Same"
    android:layout_weight="1"
    android:layout_width="0dp"
    android:layout_height="wrap_content"/>

<Button
    android:layout_marginLeft="4dp"
    android:layout_marginRight="4dp"
    android:fontFamily="casual"
    android:background="#2196F3"
    android:id="@+id/oppositeButton"
    android:text="Opposite"
    android:layout_weight="1"
    android:layout_width="0dp"
    android:layout_height="wrap_content"/>
```

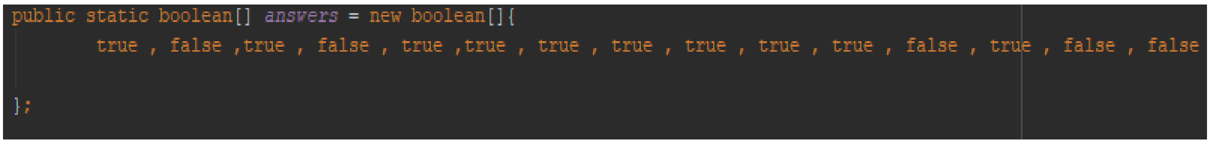
Figure 27 - Same and Opposite XML Buttons

QuizBook 1 activity refers to all the questions present in level one. “**Public Static String statements = new string”** (See Figure 28) initialises the statements inside the class. Each time a user gets a question correct the next string statement will then be shown. This mechanism is identical to how the question string operates. Additionally, the correct answers are in Boolean logic because only two variables **"Same"** or **"Opposite"** are used (See Figure 29).



```
public static String[] statments = new String[]{
        "XOH Is The same As MOX",
        "CAH Is The opposite TO JOB",
        "PUB Is The same As LOL",
        "JIN Is The opposite TO TAH",
        "KOL Is The same As VUB",
        "TKL Is The same As YIO",
        "LAR Is The same As XEB",
        "CIY Is The same As ZEH",
        "TUO Is The same As ERB",
        "NAR Is The same As TOR",
        "DED Is The same As LEV",
        "RIK Is The opposite TO ZUB",
        "POL Is The same As ZIK",
        "XON Is The opposite TO VEZ",
        "VIV Is The opposite TO DOZ",

};


public static String [] questions = new String[]{
        " How are XOH and MOX Related ",
        " How are CAH and JOB Related ",
        " How are PUB and LOL Related ",
        " How are JIN and TAH Related ",
        " How are KOL and VUB Related ",
        " How are TKL and YIO Related ",
        " How are LAR and XEB Related ",
        " How are CIY and ZEH Related ",
```

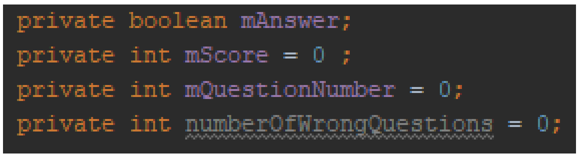
Figure 28 - Statements and questions



```
public static boolean[] answers = new boolean[]{
        true , false ,true , false , true ,true , true , true , true , true , true , false , true , false , false

};
```

Figure 29 - Boolean Logic

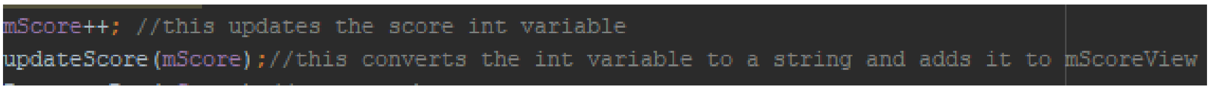
Quiz activity 1 is where the game's logic is located. the score, question number, and wrong answers are all initialised to 0 (See Figure 30)



```
private boolean mAnswer;
private int mScore = 0 ;
private int mQuestionNumber = 0;
private int numberOfWrongQuestions = 0;
```

Figure 30 - Score, question number and wrong answer = 0

As they're only two answers for the game, Boolean logic has been implemented. The correct answer will update the score, it accomplishes this by initializing the line of code **"mScore++"**, which updates the score int variable .**"Updatescore (mScore)”** converts the int variable to a string and adds it to the **"mScoreview"** variable meaning players can see the updated score when the question answered correctly. (See figure 31)

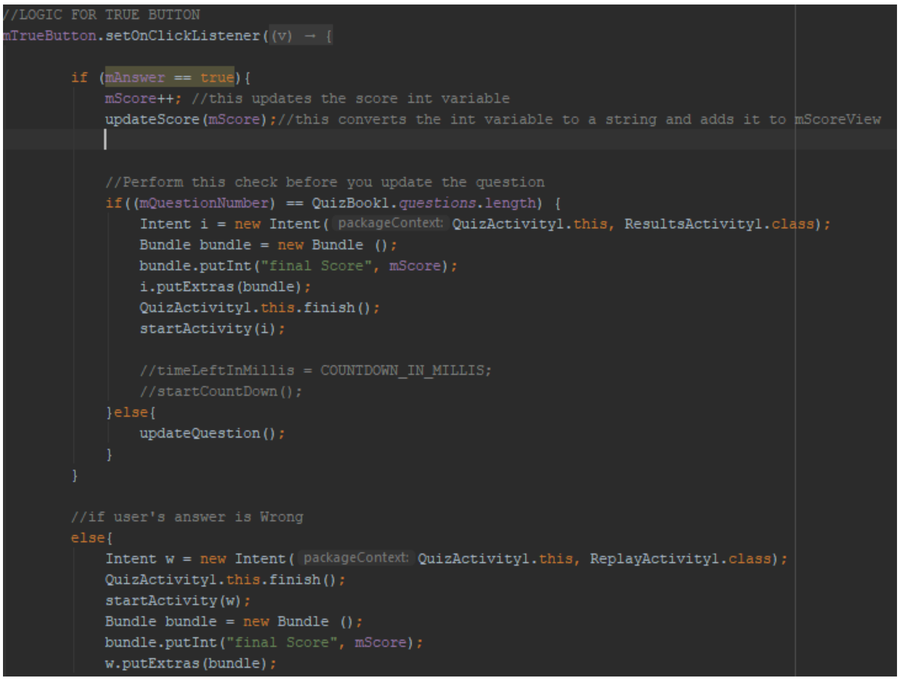


```
mScore++; //this updates the score int variable
updateScore(mScore);//this converts the int variable to a string and adds it to mScoreView
```

Figure 31 - Updating Score Logic

Concerning the true button (See Figure 32), the **“(mQuestionNumber)”** statement performs a comparison before updating the question. It verifies if the answer is correct by comparing what the user selected to the correct answer in the quiz book. If the series of questions stand incompleted, the next question will be dispensed. Once the user completes all questions, the ResultsActivity page will be launched.

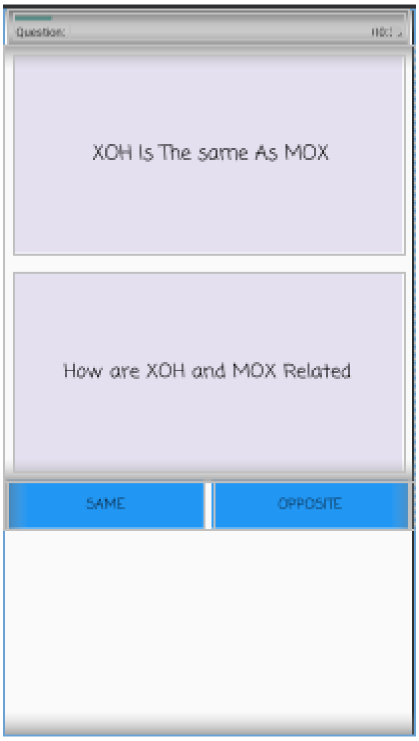
However, if the answer isn't the true button, then a new intent will be initialised, and the quiz activity will be closed meaning the game has ended.



```
//LOGIC FOR TRUE BUTTON
mTrueButton.setOnClickListener((v) → {

        if (mAnswer == true){
            mScore++; //this updates the score int variable
            updateScore(mScore);//this converts the int variable to a string and adds it to mScoreView


            //Perform this check before you update the question
            if((mQuestionNumber) == QuizBook1.questions.length) {
                Intent i = new Intent( packageContext: QuizActivity1.this, ResultsActivity1.class);
                Bundle bundle = new Bundle ();
                bundle.putInt("final Score", mScore);
                i.putExtras(bundle);
                QuizActivity1.this.finish();
                startActivity(i);

                //timeLeftInMillis = COUNTDOWN_IN_MILLIS;
                //startCountDown();
            }else{
                updateQuestion();
            }
        }

        //if user's answer is Wrong
        else{
            Intent w = new Intent( packageContext: QuizActivity1.this, ReplayActivity1.class);
            QuizActivity1.this.finish();
            startActivity(w);
            Bundle bundle = new Bundle ();
            bundle.putInt("final Score", mScore);
            w.putExtras(bundle);
```

Figure 32 - True button code

The false button operates identically to the True button. If the answer is false, the question will be updated; however, if it is not false, the activity will be stopped automatically, and the Reply activity page will be launched. See Figure 33 below for the Gameplay screen with the previously discussed game mechanics implemented.



XOH Is The same As MOX

How are XOH and MOX Related

Figure 33 - Gameplay Screen

6.1.5 First Prototype Evaluation

The initial prototype covers almost all the main requirements of the app, users can successfully login the app, and the main gameplay mechanics are working effectively. The next prototype iteration will focus on implementing the progress bar and creating more game levels.

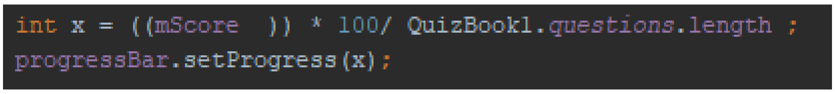
6.2 Prototype Two

Prototype two tasks mainly involved the expansion of the basic functioning game from prototype two, the expansion entails the following.

- Implementing the progress bar
- Making the G.U.I consistent through the app
- Additional Levels

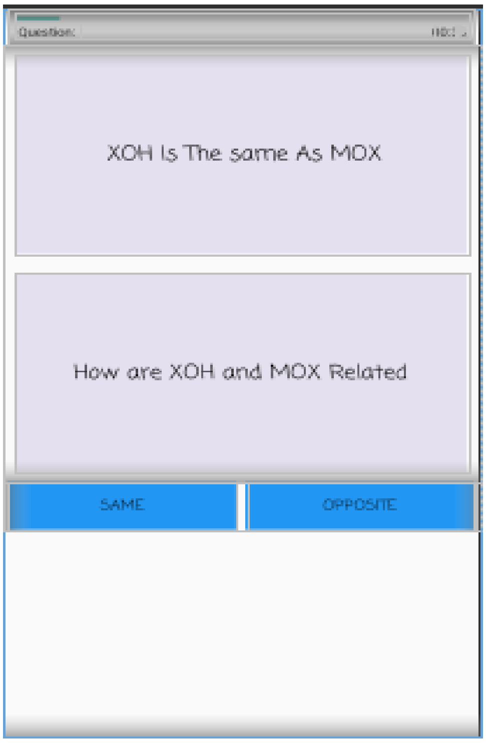
6.2.1 Progress Bar

The first step of the second prototype was to develop a progress bar that progressed every time the user got a question correct. The progress bar was set to "int X" which is set to zero and updates every time the user gets a question correct. The updated score sets the progress bar to that score (See Figure 34 & 35)



```
int x = ((mScore  )) * 100/ QuizBook1.questions.length ;
progressBar.setProgress(x);
```

Figure 34 - Progress Bar update Code



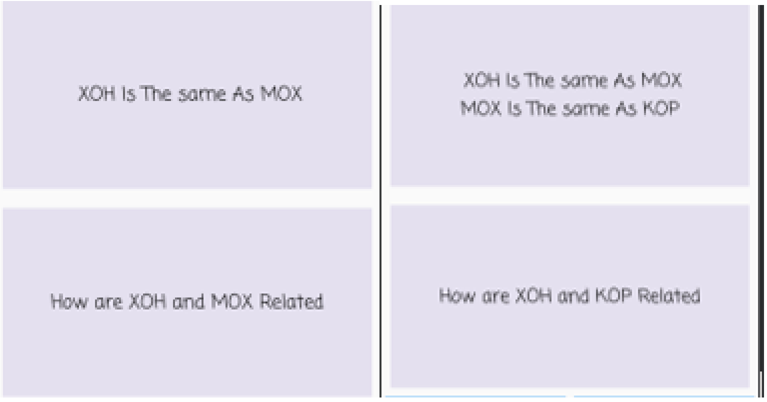
XOH Is The same As MOX

How are XOH and MOX Related

Figure 35 - Progress Bar

6.2.2 Additional levels

The author originally developed one level to test the idea and the game mechanics, further levels were added with increased difficulty, for instance, level one had 2 relational variables whereas level two has a 3 variable relationship (See Figure 36).



XOH Is The same As MOX

How are XOH and MOX Related

XOH Is The same As MOX
MOX Is The same As KOP

How are XOH and KOP Related

Figure 36 - Level 1 statement left, Level 2 statement right

6.2.3 Second Prototype Evaluation

The second prototype accomplished fulfilling some of the other main requirements and increased the longevity of the application by adding harder levels.

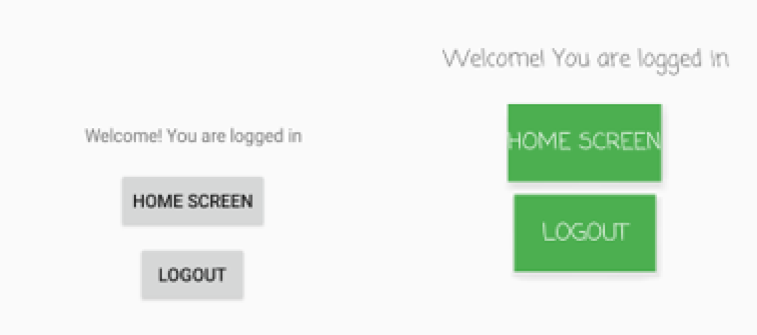
6.3 Final Prototype

This final section will discuss the implementation processes of the last prototype, this phase was largely dedicated to improving the user experience and refining details. The following tasks included.

- Update User Interface
- Adding Instruction

6.3.1 Updated user Interface

The second prototype improved the graphical user interface (GUI) of the initial prototype, the justification for doing so was that the author felt the G.U.I was bland and inconsistent. Specifically, the login activity and transition activity both had different fonts and themes compared to the rest of the application (See Figure 37).



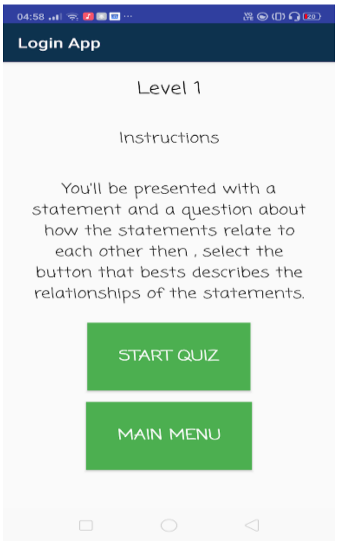
Welcome! You are logged in

Welcome! You are logged in

Figure 37 - Transition page - Old Interface (Left) New Interface (Right)

6.3.2 Instructions page

This game is quite novel, meaning the lack of exposure to relational frame theory games will make it hard for first-timers to know how to play the game, thus the author felt it was necessary to include an instructions page informing users how to play the game. The instructions page also has a main menu button to allow users to go back to the levels home page (See Figure 38).



Level 1

Instructions

You'll be presented with a statement and a question about how the statements relate to each other then , select the button that bests describes the relationships of the statements.

Figure 38 - Instructions Page

6.3.3 Evaluation of Final Prototype

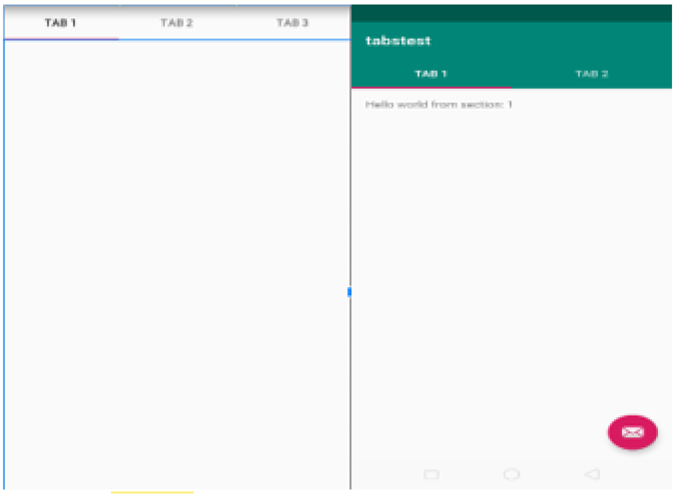
The final prototype manages to meet many of the requirements at this stage. The artefact now instructs people on how to play the game, which will remove confusion, furthermore the user interface has been updated to create a more consistent looking application. However, despite the progress made throughout the different prototype iterations, there were key features that were missing in the final prototype.

6.4 Implementation Problems

This section will discuss the numerous problems the author encountered throughout the implementation process. The issues mainly faced during this phase were errors in code, incorrect code syntax use, incorrect spelling of words. However more troublesome errors occurred which affected the author's ability to fulfil all requirements.

6.4.1 Tabs

One feature of this app was to have a tabbed view that can allow users to scroll sideways to play different types of relational frame games. However, the author contacted many errors trying to implement the tabbed activity into the app forcing the author to abandon the use of tabs. The main problem was that the author could not find a way to place a button on a fragment (tabbed page) and connect it with an activity. The author tried many tab designs but constantly received errors when trying to connect an activity page to the tabbed view. See Figure 39 for the incompatible tab designs.



Hello world from section: 1

Figure 39 - Incompatible tab Designs

6.4.2 Countdown Timer

The countdown timer is a “Should MoSCoW” requirement, however, when implementing the code, it caused the whole program to crash. The author spent a lot of time attempting to fix it however it became bothersome and was abandoned. The countdown logic seemed to interfere with the question logic meaning every time the question would initialise the app would automatically crash.

7.0 Testing

The author outlines in this chapter the testing processes used to assess the functionality of the application. The application was tested against the requirements, detailed in the requirement analysis chapter (Chapter 4). Additionally, all tests were completed on the Oppo Reno Z and the Google Nexus 7 tablet as well as the Android Studio simulator. This choice was made to test if the app could run on different devices.

7.1 Tests Results

The table below details the tests that were used during the testing phase, these tests were derived from the functional requirements in the requirement analysis chapter (Chapter 4).

Test ID	Test Description	MoSCoW Priority	Result
TF1	Sign in with a username and password.	M	SUCCESS
TF2	Inform the user if login fails.	M	SUCCESS
TF3	Stop user from signing up twice with the same account	M	SUCCESS
TF4	Logout Button Works	M	SUCCESS
TF5	The same and opposite buttons work correctly when users Press them	M	SUCCESS
TF6	The progress bar and question counter increase when the user selects the right answer	M	SUCCESS
TF7	Users can reply levels once completed	M	SUCCESS
TF8	Users can select next level when completed	M	SUCCESS

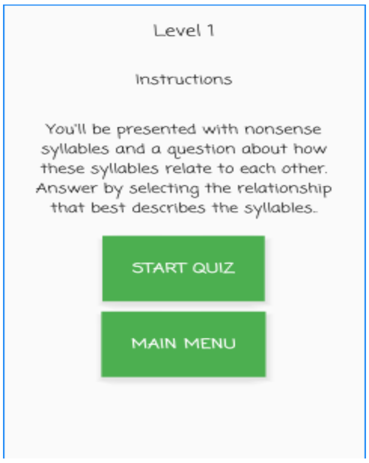
Table 7. Functional testing results

7.2 Usability Results

The overall usability results were overwhelmingly great, many participants understood the concept of the game and found the levels challenging and engaging (See appendix G). Additionally, all the usability requirements were met, users felt the app was easy to navigate, additionally they felt the app offered helpful feedback to users. However, one participant struggled with understanding the instructions of the game. they felt the instructions were not clear and it took them multiple rounds to finally understand how to play the game. Additionally, some of the colour choices may have made it hard to see texts. For instance, the “Same” and “Opposite” buttons were difficult to see especially on screens with dimmer brightness.

7.2.1 Improvements

These improvements were derived from the feedback gained from the usability tests, due to the feedback concerning the instructions on playing the game the author decided to reword the instructions on the instructions page (See Figure 40)



Level 1

Instructions

You'll be presented with nonsense syllables and a question about how these syllables relate to each other. Answer by selecting the relationship that best describes the syllables.

Figure 40 - New Instructions Page

Additionally, the author decided to change the text font of the “Same” and “Opposite buttons in the gameplay, because the colour clash with the blue box background made it hard to see the text (See Figure 41)

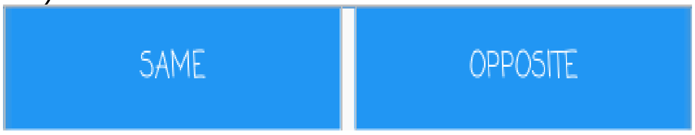


Figure 41 - Updated Same and Opposite Font.

7.3 General Criticisms

Generally, feedback on the application was positive, however constructive criticism regarding the app functionality was made. For instance, one user states “the app didn't feel like a game, it doesn't have sounds and isn't that colourful '. this sentiment was also reiterated a few times. This suggests that much more work needs to be done to make this app feel like a full-fledged game.

8.0 Evaluation

This chapter is dedicated to the evaluation of the project, specifically the requirements, the project aims and objectives and lastly, the evaluation of the project management process.

8.1 Functional requirements

Below is a table that consists of all the functional requirements, as stated in the requirements analysis (Chapter 4), the functional requirements were ranked in priority using the MoSCoW method. The completion of the artefact can now allow the author to assess whether the final artefact fulfils the initial requirements that were set near the beginning of the project.

Functional Requirements	MoSCoW Priority	Status
The application must allow users to sign up and log in to an account.	M	**Implemented**
The application must work with a database that stores user data (emails, passwords, scores)	M	**Implemented**
The application must have multiple game levels that progressively get harder.	M	**Implemented**
The application must have multiple game levels that increase in difficulty.	M	**Implemented**
The application must inform users when they get an incorrect answer.	M	**Implemented**
The application should remember the user when they return to the app (no need to log in every time the user opens the app).	M	**Implemented**
The application should have a countdown feature when a level is being played.	S	**Not Implemented** See 6.4.2 & 8.1.1.1
The application should provide a statistical analysis of the player.	S	**Not implemented**
The application should have multiple relational frames that can be played by a user.	S	**Not implemented**
The application should allow the user to set a notification for the time they would want to play the game.	C	**Not implemented**

Table 8. Evaluation of functional requirements

8.1.1 Requirements not implemented

These are requirements that did not make the final artefact; this could be due to many reasons such as errors when implementing code, its low prioritization level or project time constraints.

8.1.1.1 Countdown Timer

The countdown timer's purpose was to add urgency to the game, however, because of the errors and crashes in the app when implementing it, the author felt it was best to abandon this requirement (See Chapter 6.4.2). Despite the requirement being ranked as **“Should”**, it didn't overall affect the usability of the app, however, it does limit the game’s functionality. This comes down to the author’s lack of programming experience in mobile development. As the author didn't have the sufficient skills to implement a working countdown timer properly. Furthermore, because this is an important feature, the author decided to accomplish this requirement in future works

8.1.1.2 Statistical Analysis

The statistical requirement is also a **“Should”** requirement. Due to time constraints, unfortunately, the requirement could not be achieved, therefore this feature will be added in the future as it is still regarded as an important function to have when playing a game.

8.1.1.3 Different Types of relational Frames

This requirement of offering players different types of relational frames is a **“Should”** requirement. The process of creating a level is very time-consuming, so, unfortunately, the author did not have enough time to test users with different types of relational frames like **“More than / less than”**. Additionally, due to the failure of implementing tabs, it significantly reduced the likelihood of this requirement to be completed. The initial plan was to dedicate each type of relational frame to one tab. However, due to the failure when implementing the tabs (See Chapter 6.4.1), the author had to focus efforts elsewhere.

8.1.1.4 Notification

The notification feature was regarded as **“Could "** therefore lacked priority, the author felt that the exclusion of this feature would not affect the overall functionality or gameplay of the applications, therefore it was not executed. However, it may be considered in future developments of this application.

8.2 Non-Functional Requirements

Non-Functional Requirements	MoSCoW Priority	Status
The application must be able to run on Oppo reno Z and Google Nexus 7 tablet.	M	**Accomplished** The applications run on both Oppo Reno Z and the Google Nexus 7 successfully
The application must be based on HCI principles e.g. be easy to use, clear navigation.	M	**Accomplished** The application is easy to use according to testing feedback

The application must give the user’s feedback when necessary.	M	**Accomplished** The application notify the user when relevant.
This app should work on any android device.	S	**Accomplished** This app runs successfully on many android devices on Android Studio device emulator.

Table 9. Evaluation of Non-functional requirements

8.3 Usability Requirements

Usability Requirements	MoSCoW Priority	Status
• The system must be easy to navigate through.	M	**Accomplished**
• The application design/colour schemes/fonts should be consistent throughout the application.	M	**Accomplished**
• The application should adhere to android design standards.	S	**Accomplished**
• The application should inform users with instructions to play the game.	S	**Accomplished**
• The application should appear vibrant.	S	**Accomplished**

Table 10.Evaluation of Usability requirements

8.4 Methodology Evaluation

Extensive background research was undertaken before the author could finalize which methodology to settle on. The research helped the author identify the strengths and weaknesses of multiple software development methodologies, thus allowing the author to evaluate effectively which methodology would be the most appropriate for this project. The background research for the methodologies can be in chapter 3, where the author discusses them in depth.

The prototype model was selected to complete the application, specifically the evolutionary prototyping methodology. The justification for this was that the author lacked programming experience therefore the iterative process of this model would allow the author to grow and improve their programming skills, alongside the improvement of the prototypes. Also, the model requires feedback at every iterative stage, this means that the prototype can test out and implement new ideas easily.

To conclude, the prototype model allowed the author flexibility and the space to grow as a programmer. The constant feedback allowed the author to focus on the most important issues of the artefact.

8.5 Project management Evaluation

Time management is imperative in taking on such a huge project, thus the author decided to use tools that make the project easier to manage. The Gantt chart was one tool the author uses to manage time as discussed in chapter 3.6.

The Gantt chart allowed the author to have a holistic view of the project, enabling the author to break down the tasks into smaller assignments, and dedicate time to each assignment. The tasks that were perceived to be the most arduous, were awarded the most time. In tandem with a Gantt chart, a timetable was used to exercise more control over the week as seen in figure 6. Additionally, backwards planning (See Figure 7) was also used to break the tasks into even smaller chunks.

Despite the meticulousness of the planning, the author struggled to adhere to the Gantt chart plan, this was largely due to the underestimation of tasks, the author’s programming knowledge was so limited that tasks were extended numerous times to grant time for the author to improve their programming skills. This, unfortunately, caused the author to create a second Gantt chart to estimate the time it would need to complete the project. Additionally, major disruptions affected the author’s plans due to Covid-19, the author had to relocate back home, because universities had closed, official deadlines for the project also changed rendering my Gantt chart useless in the end.

To conclude the Gantt chart did allow the author to have a holistic overview of the project initially, however as time progressed sticking to the chart became unrealistic. The method that prevailed, in the end, was backwards planning, which allowed tasks to be constantly updated and readjusted to account for unknown variables that could hinder the project.

8.6 Evaluation of project aims and objectives

This section is dedicated to evaluating the project aims and objectives that were initially stated in chapter 1.

The project aims and objectives entail the following:

- **Allow users to test a relational frame theory game that has many progressive levels and many game modes**
  The artefact allows users to play levels that progressively get harder, however different game modes will be added in the future

- **Allow users to track how long they've played and see statistical analysis**
  The application does not currently have a statistical analysis; however, this feature will be added in future works.

- **Develop a database that can store user information.**
  The firebase database was successfully implemented.

- **Research Relational Frame Theory**
  The author extensively researched literature concerning relational frame theory to create a game based on it.

- **Analyse and Evaluate similar Applications**
  The analysed similar applications to find ways to improve the artefact.

- **Decide what the functional and non-functional requirements are**
  This was accomplished, due to the use of requirement elicitation techniques.

- **Determine the usability requirements**
  This was also accomplished, however, the system did not have considerations for those with disabilities, so testing also did not involve users with disabilities, so the extent of this accomplishment is limited.

- **Develop an android application**
  The app was successfully developed using the evolutionary prototype model.

- **Test and evaluate the developed application**
  The app was successfully tested and evaluated against the requirements.

9.0 Conclusion

To succinctly summarize the project:

Chapter 1 consisted of detailing the purpose of the project. which was to identify a problem and research the technologies that would provide a solution. The information obtained through the research allowed the author to have the tools and knowledge to solve the solution in the form of a mobile application. Chapter 2 detailed research in the form of a literature review, the author researched the necessary background information on relational frame theory, mobile technologies, and similar applications. This process allowed the author to discover the necessary features needed in the artefact.

Furthermore, chapter 3 was the stage where the author had to research and evaluate different methodologies. The author settled with using the prototyping evolutionary model because it offered the most benefits to the author. Such as greater flexibility when developing the artefact. Next, chapter 4 represents the requirement elicitation process. A multiplicity of different requirement elicitation techniques was used because the author felt it was imperative to know what the user's needs were.

Chapter 5 included the suggested design and justification of the system. Using paper prototypes and low fidelity designs allowed the author to visualize the requirements better, making the author more confident that the requirements can be actualized. Whereas, chapter 6 was the implementation stage of the various requirements with relevant screenshots detailing the processes of how and why things were coded in a specific way.

Finally, chapter 7 evaluates the completed artefact against the various functional and non-functional requirements, also the final artefact is contrasted against the project's initial aims and objectives. Consequently, the artefact accomplishes most of the requirements. However, the author identified that not all requirements were obtained and explained the reasons for their incompleteness. Moreover, room for improvement can be made which the author had stated.

9.1 General Reflection

The author will briefly reflect upon the overall project, the things that went well and the struggles that were encountered, bear in mind the author had never undertaken such a large scale project before, however, the author handled the stress and complications exceedingly well.

9.1.1 The Good

The author consistently made progress despite the issues with time management. Most tasks were accomplished in the given time. Additionally, due to the many time management techniques used, it allowed the author to always have some form of control over the project,

Furthermore, the author had zero experience in android app development, however learning how to program the application was a fun and rewarding experience. The experience gained has improved the confidence of the author to program more advanced applications in the future.

9.1.2 The Bad

Even though the author enjoyed the programming process, implementation was the most arduous phase in the entire project, the lack of programming experience meant the author struggled copiously when trying to understand new concepts or implement code. Due to the difficulty of implementing some requirements, time was lost, and some features were abandoned. Additionally, because of the Covid-19 pandemic, ordinary life was severely affected, making it even harder to focus on the completion of the project. Thankfully, the university recognised the plight of the students and extended many deadlines to attenuate the impact of Covid-19 on academia.

9.1.3 The Different

Hypothetically if the author could complete this project again, the author would most likely start the implementation stage much earlier in the year. This is because starting earlier would give the author ample time to learn more about android development and allow the author extra time to overcome programming challenges.

9.2 Future work

The app managed to execute creating a game based on relational frame theory however, as noted in previous chapters, this app has plenty of room for improvement.

9.2.1 Settings and Statistics Page

Almost all applications have a settings page where users can alter features of the applications such as, changing the themes of the application or creating notifications to remind players to play the app (See 4.3), Additionally adding statistics will increase the competitive feel of the game, it'll also allow users to analyse their progress.

9.2.2 Different types of relational frames and countdown features

As stated in the evaluation chapter both the countdown and different relational frames requirements, were not implemented due to errors and the time constraint. In the future, the author hopefully would be competent enough to include both features into the app, as it would increase the game's life span by keeping it fun and challenging. Future iterations of this app may also allow users to decrease the countdown time via settings to increase game difficulty.

9.2.3 Multiple platforms and Advertisements

Currently, this application is an android exclusive however, plans include creating an IOS version of this app, which will allow more people access to a free relational frame game. Furthermore, because this application is a game the author could potentially monetize it, by allowing small non-intrusive advertising windows in the game, this would compensate for the fact that the game is free to play.

9.2.4 More levels

Due to limited time the author could not create many of levels. However future instalments more playable levels.

9.3 Overall Conclusion

To conclude, this final year project was honestly the most challenging work I've done in my life thus far, however it was also paradoxically the most fulfilling and rewarding project I've produced. The project taught me a multiplicity of lessons, in terms of the importance of time management, the importance of perseverance and the importance of communication.

Despite the app not being as fleshed out as I'd like it to, due to my lack of programming experience, the information available on the internet was invaluable and helped me overcome the copious technical issues I faced. There were several instances where I'd get stuck on a problem for days. However, the thrill and challenge of figuring it out kept me going. Also, the desire to see an app idea in my head actualized gave me more than enough willpower to persevere. Also, the weekly meetings with supervisors were beneficial and so enlightening. I truly believe if I didn't meet my supervisor weekly and adhere to their deadlines, this project would not have been completed to this standard.

I will conclude by offering advice to future students taking on a similar project to me, I must emphasize that you "**MUST"** create something your passionate about. In my case, I managed to merge my love for human cognition with app development and produced an app that I'm proud of. Doing what you love can grant you limitless powers to overcome frustrations and failures that you will encounter in such a project. Furthermore, I'd like to highlight the importance of planning, plan when things are going right, plan when things are going wrong, always plan and break down large tasks into smaller digestible items. Doing so will alleviate unnecessary pressure and anxiety, meaning you can operate optimally. I'd also suggest that students should keep in constant contact with their supervisors because their advice and guidance is invaluable.

Lastly don't forget to live your life, schedule time for friends and family because doing so will keep you sane.

References

"Write once, run anywhere?". Computer Weekly. May 2, 2002. Retrieved July 27, 2009.

ACM SIGCHI. (1992). ACM SIGCHI Curricula for Human-Computer Interaction (pp. 5-7). New York. Retrieved from http://dl.acm.org/citation.cfm?id=2594128&preflayout=tabs

Babich, N. (2017). The Underestimated Power Of Color In Mobile App Design — Smashing Magazine. Retrieved 28 April 2020, from https://www.smashingmagazine.com/2017/01/underestimated-power-color-mobile-app-design/

Bagsariya, S. (2020). Why choose Firebase? | Features & Benefits of Firebase. Retrieved 3 May 2020, from https://kodytechnolab.com/features-benefits-firebase

Barnes-Holmes, Dermot; Barnes-Holmes, Yvonne; Luciano, Carmen; McEnteggart, Ciara (2017). "From the IRAP and REC model to a multi-dimensional multi-level framework for analyzing the dynamics of arbitrarily applicable relational responding" (PDF). Journal of Contextual Behavioral Science. 6 (4): 434–445. doi:10.1016/j.jcbs.2017.08.001. hdl:1854/LU-8547400. ISSN 2212-1447.

Blackledge, J. T. (2003). An introduction to relational frame theory: Basics and applications. *The Behavior Analyst Today, 3*(4), 421-433. http://dx.doi.org/10.1037/h0099997

Butusov, M. (2019). Native vs Hybrid apps. What to choose in 2019?. Retrieved 21 April 2020, from https://blog.techmagic.co/native-vs-hybrid-apps/

Cassidy, S., Roche, B., Colbert, D., Stewart, I., & Grey, I. (2016). A relational frame skills training intervention to increase general intelligence and scholastic aptitude. Learning and Individual Differences, 47, 222–235. https://doi.org/10.1016/j.lindif.2016.03.001.

Cherry, B. (2018). Why Are Social Media Sites Blue?. Retrieved 28 April 2020, from https://www.bluleadz.com/blog/why-are-social-media-sites-blue

Cognitive Assessment and Training Market by End-user and Geography - Forecast and Analysis 2020-2024. (2020). Retrieved 13 March 2020, from https://www.technavio.com/report/cognitive-assessment-and-training-market-industry-analysis

Cullinan, V., & Vitale, A. (2009). The contribution of Relational Frame Theory to the development of interventions for impairments of language and cognition. The Journal of Speech and Language Pathology – Applied Behavior Analysis, 4(1), 132-145.

Dymond, Simon; Barnes, Dermot (1995). "A transformation of self-discrimination response functions in accordance with the arbitrarily applicable relations of sameness, more than, and less than". Journal of the Experimental Analysis of Behavior. 64(2): 163–184. doi:10.1901/jeab.1995.64-163. ISSN 0022-5002. PMC 1350108. PMID 16812766.

Esposito, E. (2018). Low-fidelity vs. high-fidelity prototyping. Retrieved 3 May 2020, from https://www.invisionapp.com/inside-design/low-fi-vs-hi-fi-prototyping/

Firebase Realtime Database | Store and sync data in real time. (2020). Retrieved 13 March 2020, from https://firebase.google.com/products/realtime-database

Firebase vs SQLite | What are the differences?. (2020). Retrieved 13 March 2020, from https://stackshare.io/stackups/firebase-vs-sqlite

Functional and Nonfunctional Requirements: Specification and Types. (2018). Retrieved 25 April 2020, from https://www.altexsoft.com/blog/business/functional-and-non-functional-requirements-specification-and-types/

Gibb, R. (2016). What is a Web Application? | How a Web Application Works. Retrieved 13 April 2020, from https://blog.stackpath.com/web-application/

Goltz, S. (2014). A Closer Look At Personas: What They Are And How They Work | 1 — Smashing Magazine. Retrieved 24 April 2020, from https://www.smashingmagazine.com/2014/08/a-closer-look-at-personas-part-1/

Greene, J. 7 Research-Backed Benefits of Mind Mapping - Focus. Retrieved 24 April 2020, from https://www.mindmeister.com/blog/mind-mapping-benefits-who-needs-mind-maps/

Griffith, C. (2019). Ionic Article: What is Hybrid App Development?. Retrieved 21 April 2020, from https://ionicframework.com/resources/articles/what-is-hybrid-app-development

Hayes, S. C. (1991). A relational control theory of stimulus equivalence. In L. J. Hayes & P. N. Chase (Eds.), International Institute on Verbal Relations. Dialogues on verbal behavior: The First International Institute on Verbal Relations (p. 19–40). Context Press.

Hewett; Baecker; Card; Carey; Gasen; Mantei; Perlman; Strong; Verplank. "ACM SIGCHI Curricula for Human–Computer Interaction". ACM SIGCHI. Archived from the original on 17 August 2014. Retrieved 15 July 2014.

Hybrid Apps: An overview of Advantages, Limitations & Consequences for your Testing Phases. Retrieved 21 April 2020, from https://www2.stardust-testing.com/en/blog-en/hybrid-apps

Hybrid Mobile App Development - Top 9 Benefits | Redbytes. (2019). Retrieved 21 April 2020, from https://www.redbytes.in/hybrid-mobile-app-development-benefits/

Kable, J., Caulfield, M., Falcone, M., McConnell, M., Bernardo, L., & Parthasarathi, T. et al. (2017). No Effect of Commercial Cognitive Training on Brain Activity, Choice Behavior, or Cognitive Performance. The Journal Of Neuroscience, 37(31), 7390-7402. doi: 10.1523/jneurosci.2832-16.2017

Khan, U. (2020). The Pros and Cons of Native Apps | Clutch.co. Retrieved 31 March 2020, from https://clutch.co/app-developers/resources/pros-cons-native-apps

Kumur, S. (2012). Advantages and disadvantages of web application over native application. Retrieved 18 April 2020, from https://www.mobileqazone.com/profiles/blogs/advantages-and-disadvantages-of-web-application-over-native

Labs, D. (2019). Requirement Elicitation Techniques for Business Analysis. Retrieved 24 April 2020, from https://medium.com/@dltlabs/requirement-elicitation-techniques-for-business-analysis-ed0a3d405910

Linear Layout. (2019). Retrieved 4 May 2020, from https://developer.android.com/guide/topics/ui/layout/linear

Lynn, R. What are the Disadvantages of Agile? | Planview. Retrieved 10 April 2020, from https://www.planview.com/resources/articles/disadvantages-agile/

Madsen, S. (2019). Using the MoSCoW Method to Prioritize Projects - ProjectManager.com. Retrieved 25 April 2020, from https://www.projectmanager.com/training/prioritize-moscow-technique

Mindsmapped (2015). Retrieved 12 March 2020, from https://www.mindsmapped.com/java-advantages-and-disadvantages/

Nations, D. (2019). What Exactly Is a Web Application?. Retrieved 13 April 2020, from https://www.lifewire.com/what-is-a-web-application-3486637

Ohyver, M., Moniaga, J., Sungkawa, I., Subagyo, B., & Chandra, I. (2019). The Comparison Firebase Realtime Database and MySQL Database Performance using Wilcoxon Signed-Rank Test. Procedia Computer Science, 157, 396-405. doi: 10.1016/j.procs.2019.08.231

Olic, A. (2017). Advantages and disadvantages of Agile Project Management [Checklist] · Activecollab Blog. Retrieved 10 April 2020, from https://activecollab.com/blog/project-management/agile-project-management-advantages-disadvantages

Poetker, B. (2019). What Is an SDK? (+How It's Different Than an API). Retrieved 13 April 2020, from https://learn.g2.com/sdk

Principles behind the Agile Manifesto. Retrieved 9 April 2020, from https://agilemanifesto.org/principles.html

Prototype Model with Advantages and disadvantages -

All Major Seven Subjects. (2019). Retrieved 9 April 2020, from https://am7s.com/prototype-model/

Prototyping Model in Software Engineering: Methodology, Process, Approach. Retrieved 2 April 2020, from https://www.guru99.com/software-engineering-prototyping-model.html

Reich, M. (2013). Agile vs. Waterfall, Revisited. Commonplaces.com. Retrieved 10 April 2020, from http://www.commonplaces.com/blog/agile-vs-waterfall-revisited/

Rouse, M. What is native app? - Definition from WhatIs.com. (2020). Retrieved 30 March 2020, from https://searchsoftwarequality.techtarget.com/definition/native-application-native-app

Rouse, M. What is the Prototyping Model?. Retrieved 2 April 2020, from https://searchcio.techtarget.com/definition/Prototyping-Model

Rouse, M. What is Web Application (Web Apps) and its Benefits.

Retrieved 13 April 2020, from https://searchsoftwarequality.techtarget.com/definition/Web-application-Web-app

Royce, W. (1970). Retrieved 2 April 2020, from http://www-scf.usc.edu/~csci201/lectures/Lecture11/royce1970.pdf

Saccomani, P. (2020). Native, Web or Hybrid Apps? What's The Difference?. Retrieved 31 March 2020, from https://www.mobiloud.com/blog/native-web-or-hybrid-apps/#5

Shneiderman, B. (1998). Designing the user interface. Boston: Pearson

Smyth N (2017) Kotlin/Android Studio 3.0 Development Essentails , Andriod 8 editinon Retrievied From https://www.ebookfrenzy.com/pdf_previews/Kotlin30EssentialsPreview.pdf (p.71)

Software Engineering | Prototyping Model - GeeksforGeeks. Retrieved 9 April 2020, from https://www.geeksforgeeks.org/software-engineering-prototyping-model/

Spacey, J. (2017). 11 Examples of Usability Requirements. Retrieved 26 April 2020, from https://simplicable.com/new/usability-requirements.

Stevens, E. (2018). What Is The Difference Between A Mobile App And A Web App?. Retrieved 14 April 2020, from https://careerfoundry.com/en/blog/web-development/what-is-the-difference-between-a-mobile-app-and-a-web-app/

Task Analysis | Usability.gov. Retrieved 24 April 2020, from https://www.usability.gov/how-to-and-tools/methods/task-analysis.html

The Advantages & Disadvantages of Prototyping - Rapids Reproductions. Retrieved 9 April 2020, from https://rapidsrepro.com/advantages-disadvantages-prototyping/

Tshai, K. (2018). Retrieved 2 April 2020, from https://www.researchgate.net/publication/332825162_A_Systematic_Approach_to_Software_Quality_Assurance_The_Relationship_of_Project_Activities_within_Project_Life_Cycle_and_System_Development_Life_Cycle

Valdellon, L. (2019). What is an SDK? Everything You Need to Know | CleverTap. Retrieved 13 April 2020, from https://clevertap.com/blog/what-is-an-sdk/

What is agile development?. (2019). Retrieved 9 April 2020, from https://www.itpro.co.uk/agile-development/28040/what-is-agile-development

What is Non-Functional Requirement? Types and Examples. Retrieved 25 April 2020, from https://www.guru99.com/non-functional-requirement-type-example.html

What is Use Case Diagram?. Retrieved 3 May 2020, from https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/

Appendices

Appendix A - Project initiation Diagram

Project Specification (Emeka Nwaenie)

School of Computing final year project

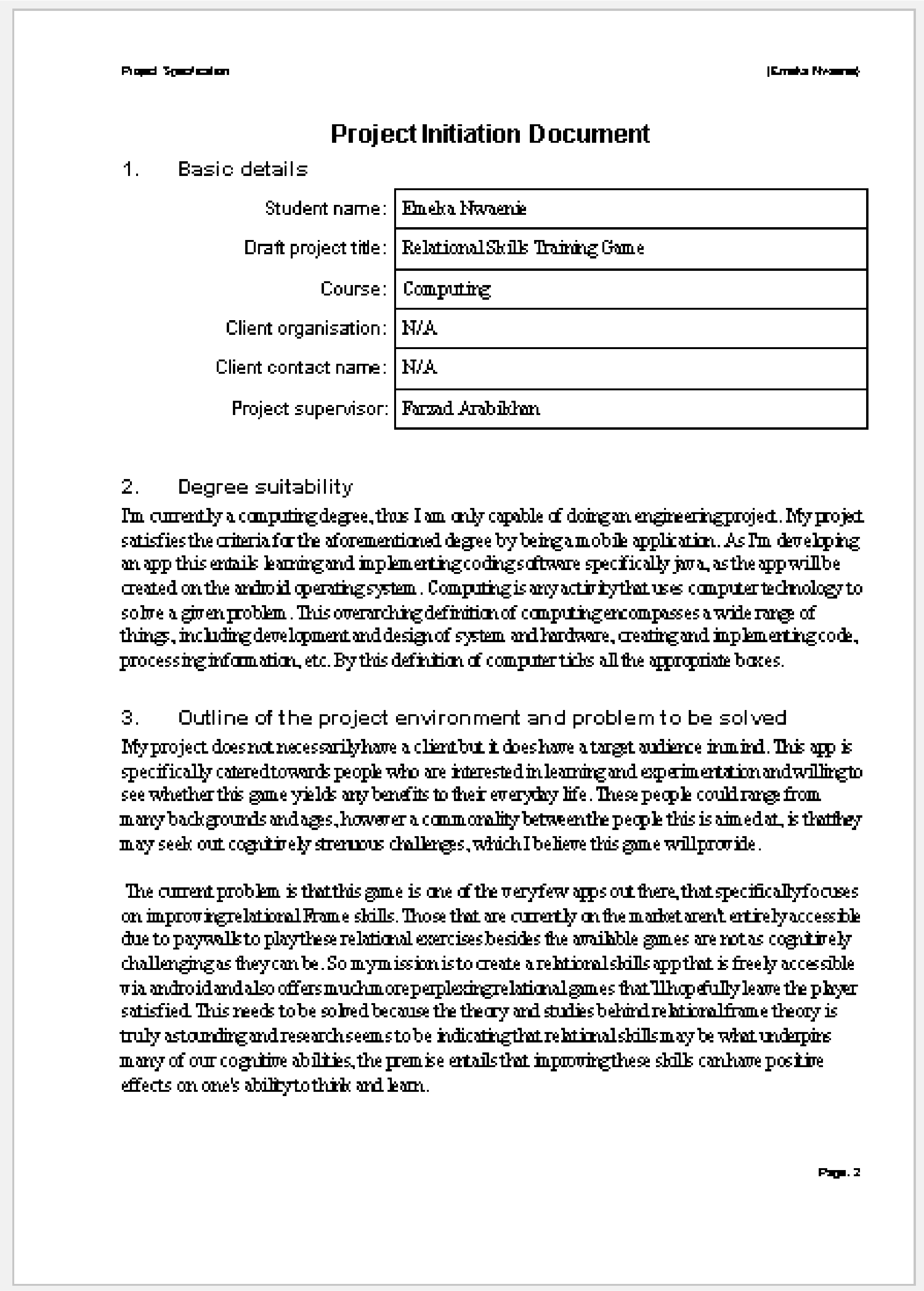
Emeka Nwaenie

PJE40

Project Initiation Document

Relational Skills Training Game

Page. 1



Project Initiation Document

1. Basic details

Student name:	Emeka Nwaenie
Draft project title:	Relational Skills Training Game
Course:	Computing
Client organisation:	N/A
Client contact name:	N/A
Project supervisor:	Farzad Arabikhan

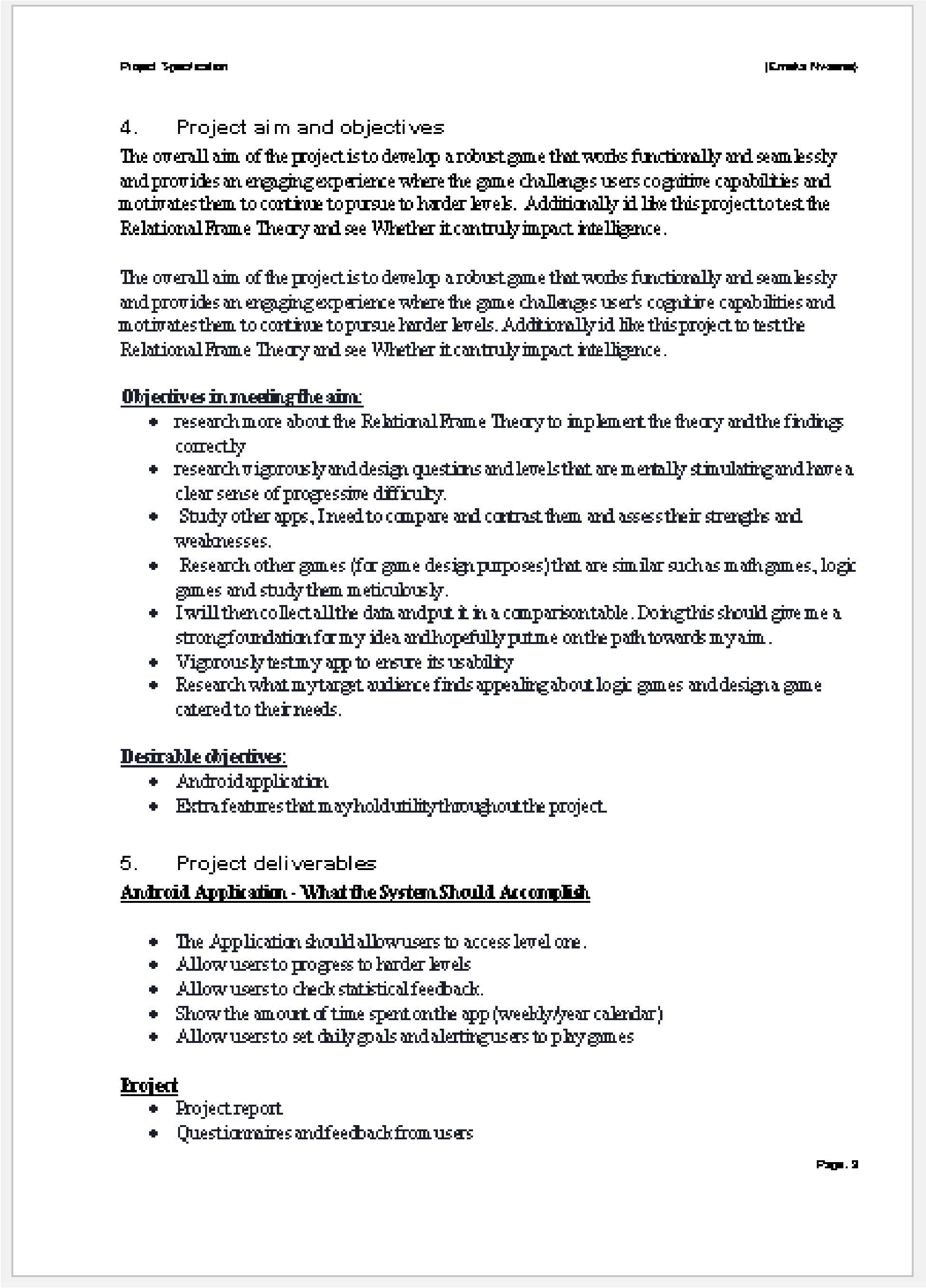
2. Degree suitability

I'm currently a computing degree, thus I am only capable of doing an engineering project. My project satisfies the criteria for the aforementioned degree by being a mobile application. As I'm developing an app this entails learning and implementing coding software specifically java, as the app will be created on the android operating system. Computing is any activity that uses computer technology to solve a given problem. This overarching definition of computing encompasses a wide range of things, including development and design of system and hardware, creating and implementing code, processing information, etc. By this definition of computer ticks all the appropriate boxes.

3. Outline of the project environment and problem to be solved

My project does not necessarily have a client but it does have a target audience in mind. This app is specifically catered towards people who are interested in learning and experimentation and willing to see whether this game yields any benefits to their everyday life. These people could range from many backgrounds and ages, however a commonality between the people this is aimed at, is that they may seek out cognitively strenuous challenges, which I believe this game will provide.

The current problem is that this game is one of the very few apps out there, that specifically focuses on improving relational Frame skills. Those that are currently on the market aren't entirely accessible due to paywalls to play these relational exercises besides the availble games are not as cognitively challenging as they can be. So my mission is to create a relational skills app that is freely accessible via android and also offers much more perplexing relational games that'll hopefully leave the player satisfied. This needs to be solved because the theory and studies behind relational frame theory is truly astounding and research seems to be indicating that relational skills may be what underpins many of our cognitive abilities, the premise entails that improving these skills can have positive effects on one's ability to think and learn.



4. Project aim and objectives

The overall aim of the project is to develop a robust game that works functionally and seamlessly and provides an engaging experience where the game challenges users cognitive capabilities and motivates them to continue to pursue to harder levels. Additionally id like this project to test the Relational Frame Theory and see Whether it can truly impact intelligence.

The overall aim of the project is to develop a robust game that works functionally and seamlessly and provides an engaging experience where the game challenges user's cognitive capabilities and motivates them to continue to pursue harder levels. Additionally id like this project to test the Relational Frame Theory and see Whether it can truly impact intelligence.

Objectives in meeting the aim:

- research more about the Relational Frame Theory to implement the theory and the findings correctly
- research vigorously and design questions and levels that are mentally stimulating and have a clear sense of progressive difficulty.
- Study other apps, I need to compare and contrast them and asses their strengths and weaknesses.
- Research other games (for game design purposes) that are similar such as math games, logic games and study them meticulously.
- I will then collect all the data and put it in a comparison table. Doing this should give me a strong foundation for my idea and hopefully put me on the path towards my aim.
- Vigorously test my app to ensure its usability
- Research what my target audience finds appealing about logic games and design a game catered to their needs.

Desirable objectives:

- Android application
- Extra features that may hold utility throughout the project.

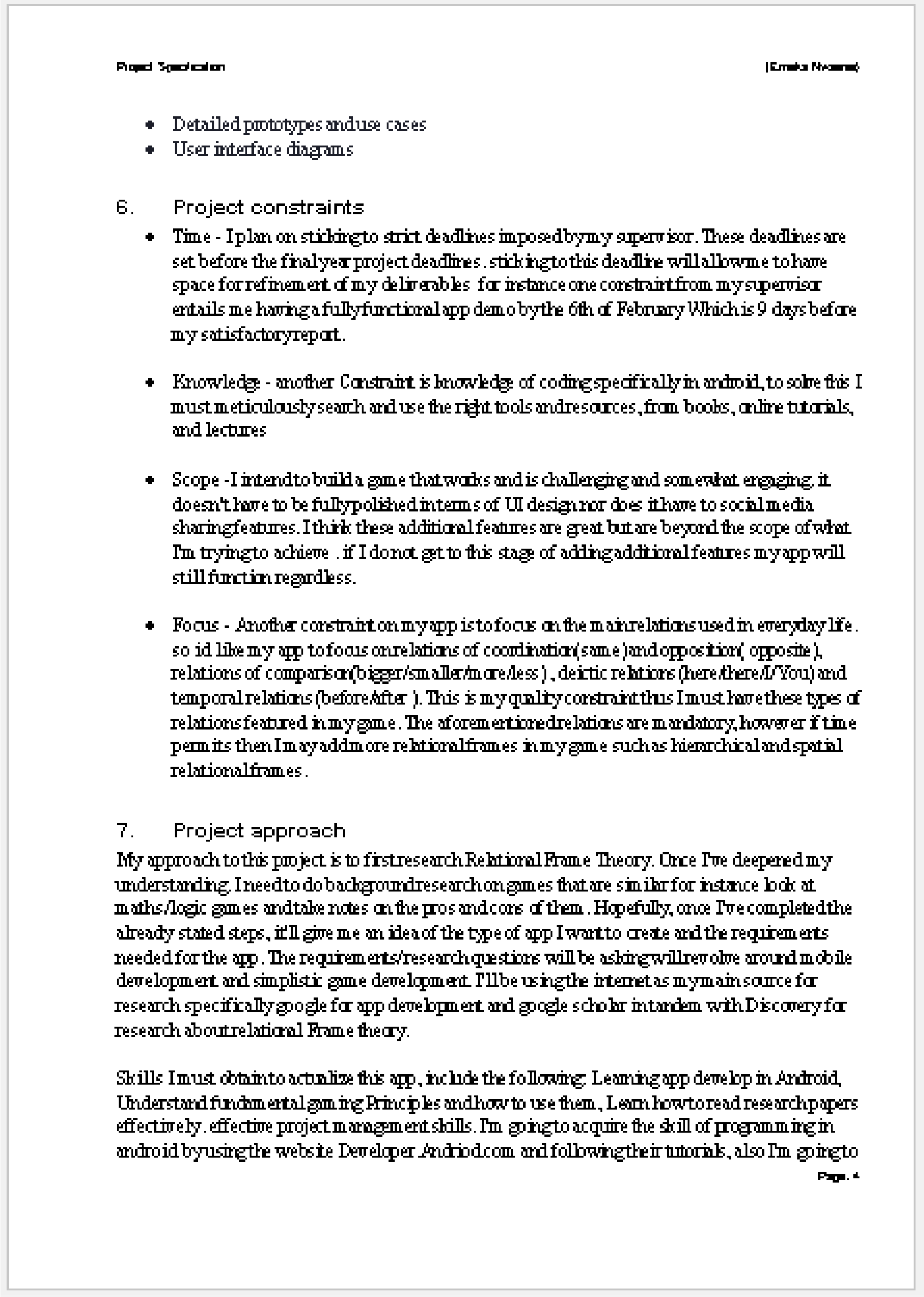
5. Project deliverables

Android Application - What the System Should Accomplish

- The Application should allow users to access level one.
- Allow users to progress to harder levels
- Allow users to check statistical feedback.
- Show the amount of time spent on the app (weekly/year calendar)
- Allow users to set daily goals and alerting users to play games

Project

- Project report
- Questionnaires and feedback from users



- Detailed prototypes and use cases
- User interface diagrams

6. Project constraints

- Time - I plan on sticking to strict deadlines imposed by my supervisor. These deadlines are set before the final year project deadlines . sticking to this deadline will allow me to have space for refinement of my deliverables for instance one constraint from my supervisor entails me having a fully functional app demo by the 6th of February Which is 9 days before my satisfactory report.

- Knowledge - another Constraint is knowledge of coding specifically in android, to solve this I must meticulously search and use the right tools and resources, from books, online tutorials, and lectures

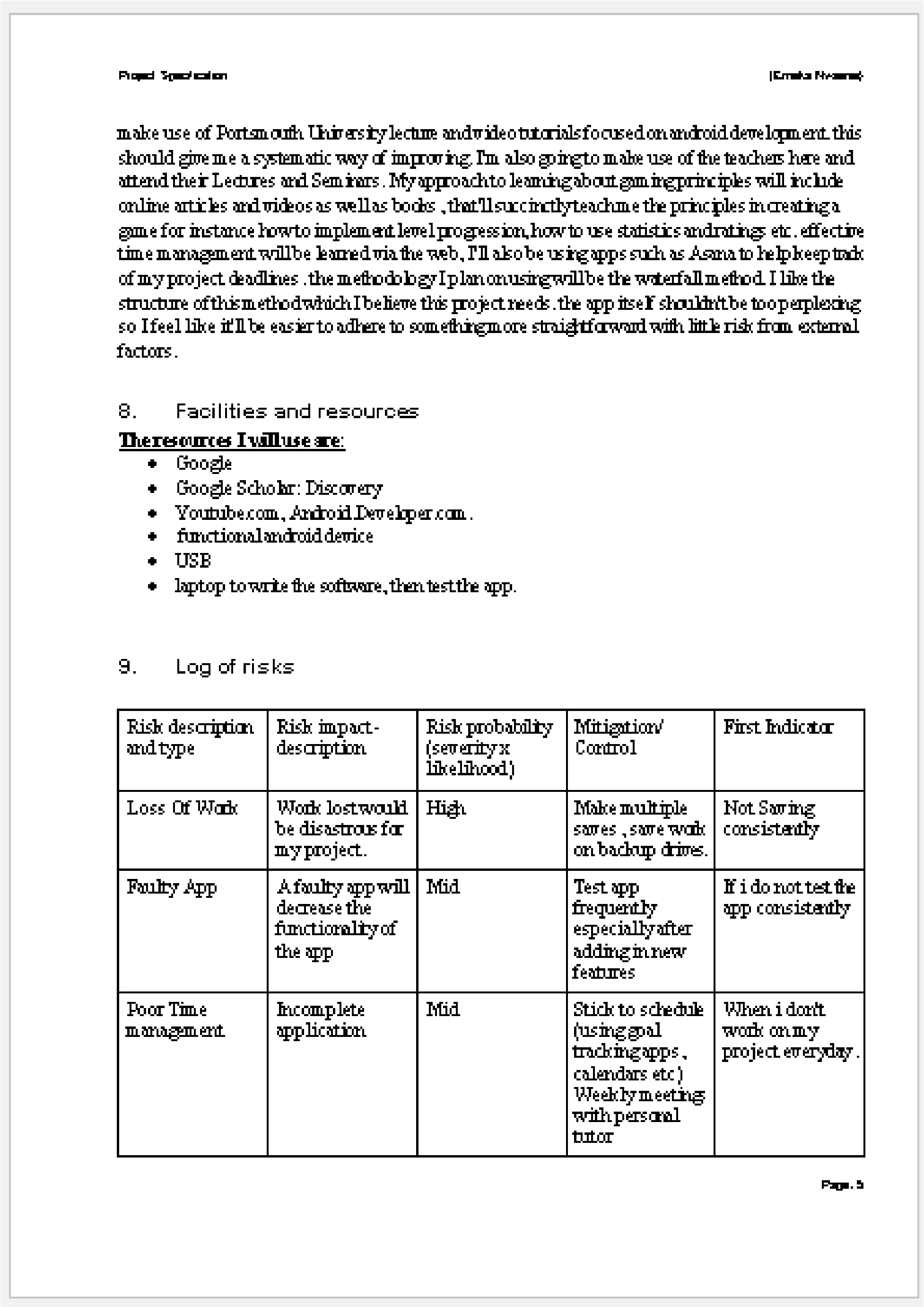
- Scope -I intend to build a game that works and is challenging and somewhat engaging. it doesn't have to be fully polished in terms of UI design nor does it have to social media sharing features. I think these additional features are great but are beyond the scope of what I'm trying to achieve . if I do not get to this stage of adding additional features my app will still function regardless.

- Focus - Another constraint on my app is to focus on the main relations used in everyday life . so id like my app to focus on relations of coordination(same)and opposition(opposite), relations of comparison(bigger/smaller/more/less) , deictic relations (here/there/I/You) and temporal relations (before/After). This is my quality constraint thus I must have these types of relations featured in my game . The aforementioned relations are mandatory, however if time permits then I may add more relational frames in my game such as hierarchical and spatial relational frames .

7. Project approach

My approach to this project is to first research Relational Frame Theory. Once I've deepened my understanding. I need to do background research on games that are similar for instance look at maths/logic games and take notes on the pros and cons of them . Hopefully, once I've completed the already stated steps, it'll give me an idea of the type of app I want to create and the requirements needed for the app . The requirements/research questions will be asking will revolve around mobile development and simplistic game development. I'll be using the internet as my main source for research specifically google for app development and google scholar in tandem with Discovery for research about relational Frame theory.

Skills I must obtain to actualize this app, include the following: Learning app develop in Android, Understand fundamental gaming Principles and how to use them., Learn how to read research papers effectively. effective project management skills. I'm going to acquire the skill of programming in android by using the website Developer.Andriod.com and following their tutorials, also I'm going to



make use of Portsmouth University lecture and video tutorials focused on android development. this should give me a systematic way of improving. I'm also going to make use of the teachers here and attend their Lectures and Seminars . My approach to learning about gaming principles will include online articles and videos as well as books , that'll succinctly teach me the principles in creating a game for instance how to implement level progression, how to use statistics and ratings etc . effective time management will be learned via the web, I'll also be using apps such as Asana to help keep track of my project deadlines . the methodology I plan on using will be the waterfall method. I like the structure of this method which I believe this project needs . the app itself shouldn't be too perplexing so I feel like it'll be easier to adhere to something more straightforward with little risk from external factors .

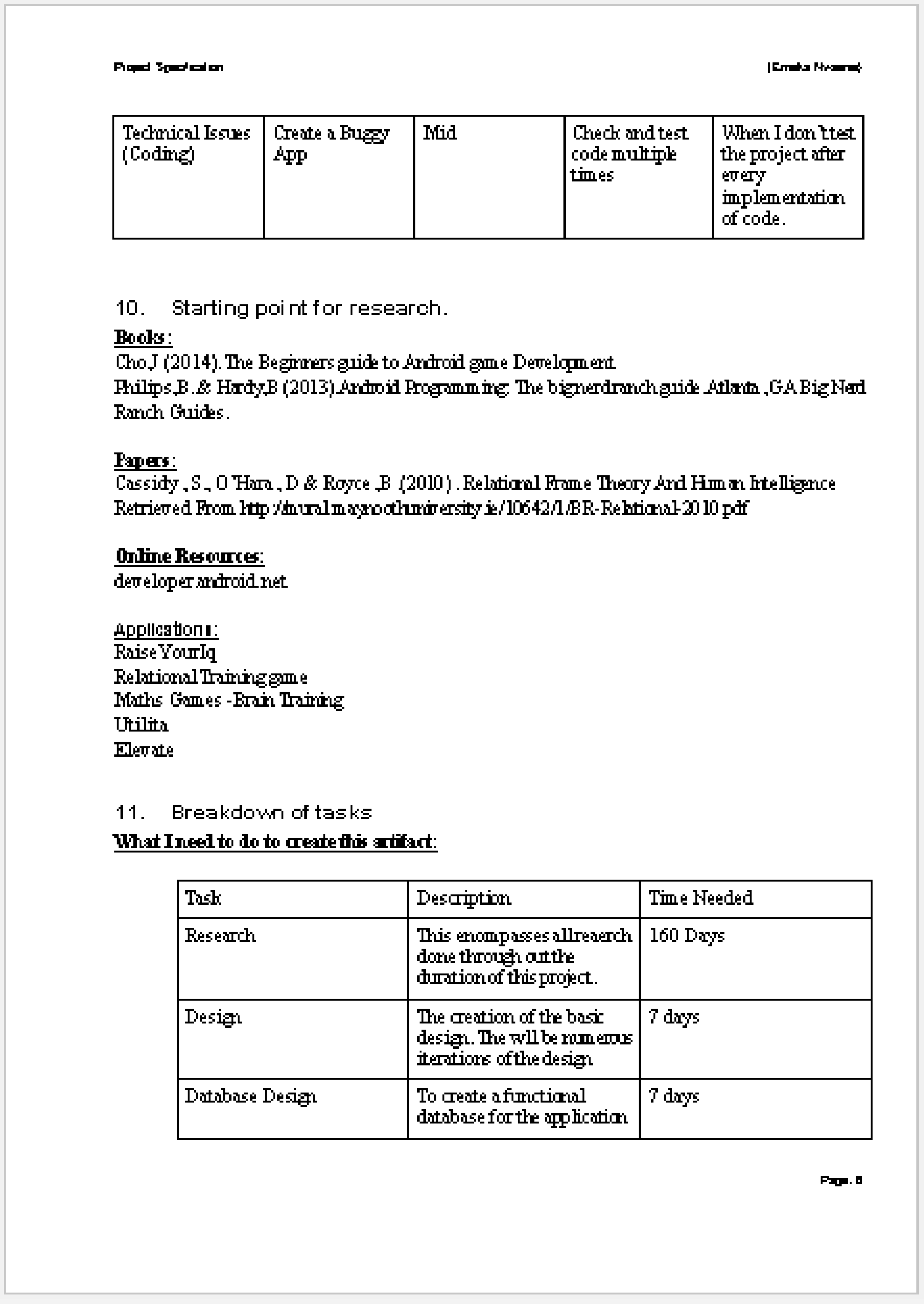
8. Facilities and resources

<u>The resources I will use are:</u>

- Google
- Google Scholar: Discovery
- Youtube.com , Android Developer.com .
- functional android device
- USB
- laptop to write the software, then test the app.

9. Log of risks

Risk description and type	Risk impact- description	Risk probability (severity x likelihood)	Mitigation/ Control	First Indicator
Loss Of Work	Work lost would be disastrous for my project.	High	Make multiple saves , save work on backup drives.	Not Saving consistently
Faulty App	A faulty app will decrease the functionality of the app	Mid	Test app frequently especially after adding in new features	If i do not test the app consistently
Poor Time management	Incomplete application	Mid	Stick to schedule (using goal tracking apps , calendars etc) Weekly meeting with personal tutor	When i don't work on my project everyday .



Technical Issues (Coding)	Create a Buggy App	Mid	Check and test code multiple times	When I don't test the project after every implementation of code.

10. Starting point for research.

Books:

Cho,J (2014). The Beginners guide to Android game Development
Philips,B .& Hardy,B (2013).Android Programming: The bignerdranch guide .Atlanta ,GA:Big Nerd Ranch Guides .

Papers:

Cassidy , S., O'Hara , D & Royce ,B (2010) . Relational Frame Theory And Human Intelligence Retrieved From http://mural.maynoothuniversity.ie/10642/1/BR-Relational-2010.pdf

Online Resources:

developer.android.net

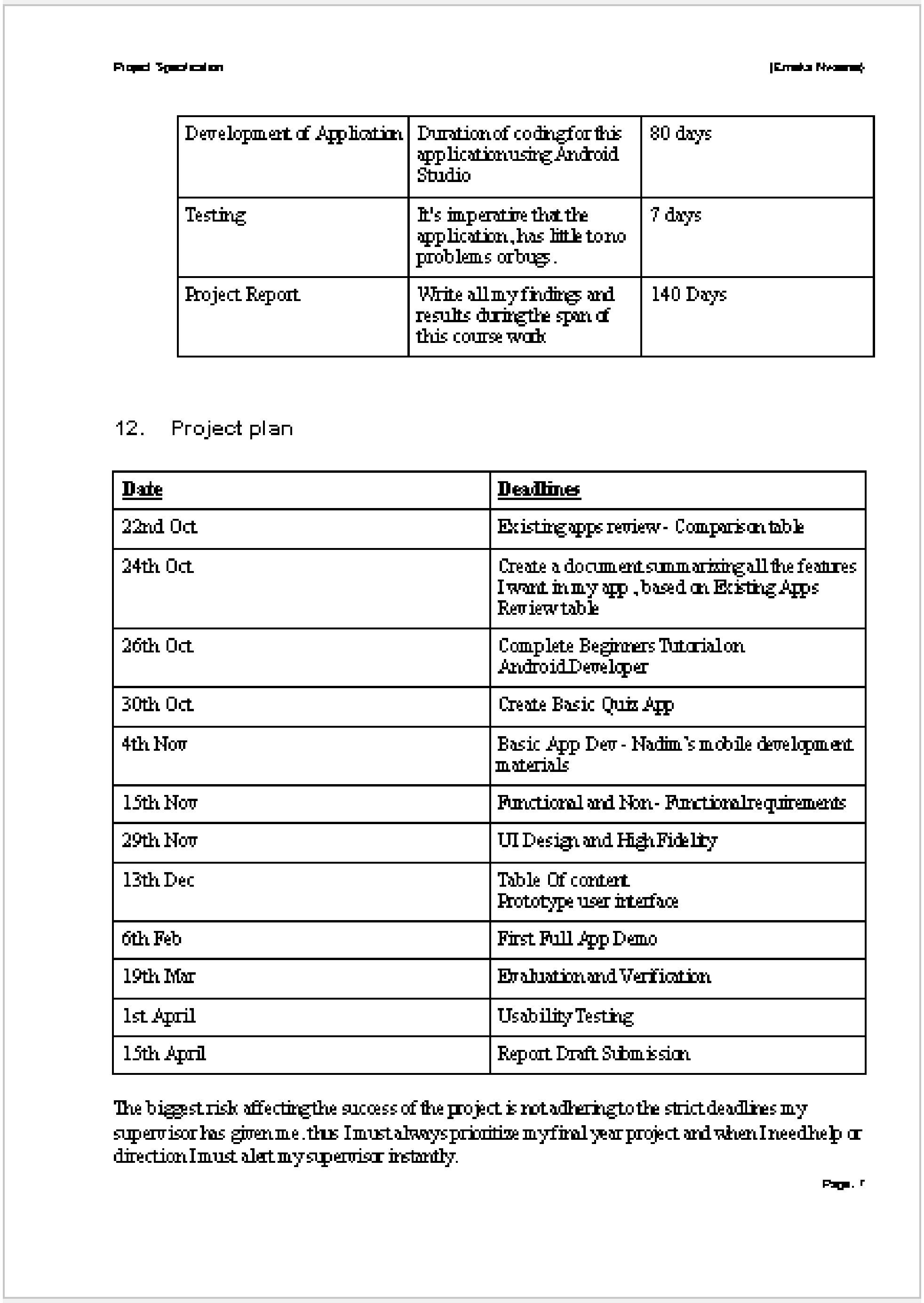
Applications:

Raise YourIq
Relational Training game
Maths Games -Brain Training
Utilita
Elevate

11. Breakdown of tasks

What I need to do to create this artifact:

Task	Description	Time Needed
Research	This enompasses all reearch done through out the duration of this project.	160 Days
Design	The creation of the basic design. The wll be numerous iterations of the design	7 days
Database Design	To create a functional database for the application	7 days



Development of Application	Duration of coding for this application using Android Studio	80 days
Testing	It's imperative that the application, has little to no problems or bugs.	7 days
Project Report	Write all my findings and results during the span of this course work	140 Days

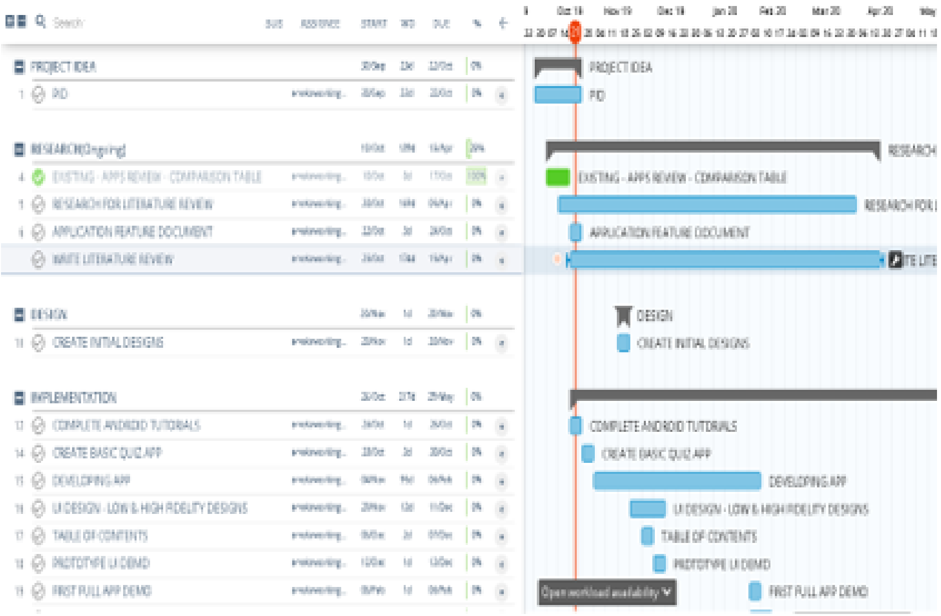
12. Project plan

Date	**Deadlines**
22nd Oct	Existing apps review - Comparison table
24th Oct	Create a document summarizing all the features I want in my app , based on Existing Apps Review table
26th Oct	Complete Beginners Tutorial on Android Developer
30th Oct	Create Basic Quiz App
4th Nov	Basic App Dev - Nadim's mobile development materials
15th Nov	Functional and Non - Functional requirements
29th Nov	UI Design and High Fidelity
13th Dec	Table Of content Prototype user interface
6th Feb	First Full App Demo
19th Mar	Evaluation and Verification
1st April	Usability Testing
15th April	Report Draft Submission

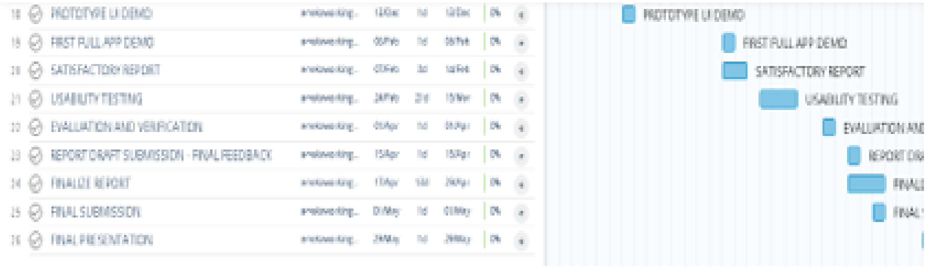
The biggest risk affecting the success of the project is not adhering to the strict deadlines my supervisor has given me. thus I must always prioritize my final year project and when I need help or direction I must alert my supervisor instantly.

13. Legal, ethical, professional, social issues (mandatory)

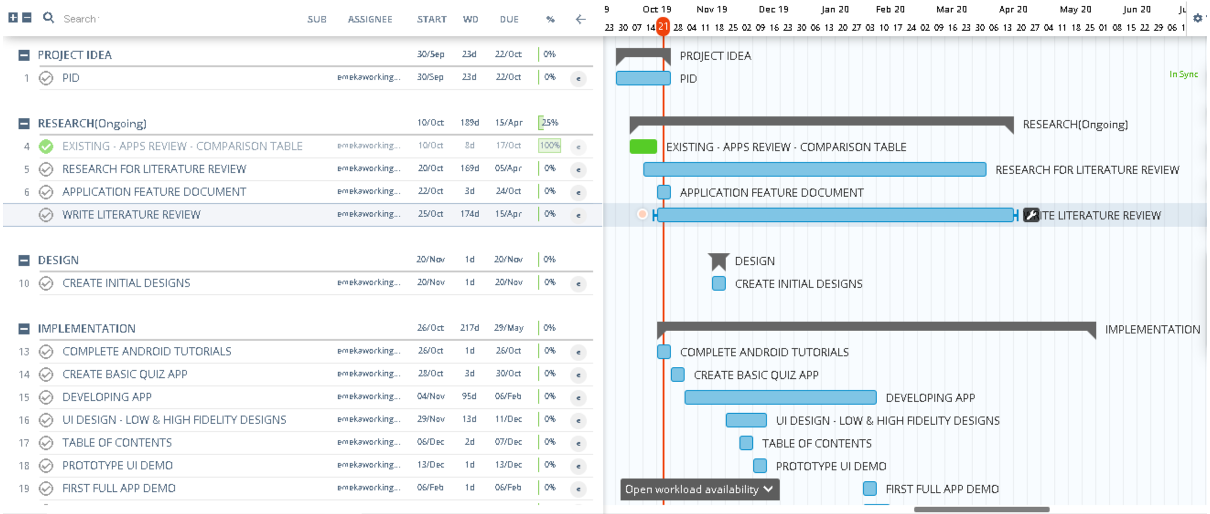
Due to the nature of this app, the only legal, ethical and professional issue with this app, is the data that will be stored in a database. I will comply with the Data Protection Act 1998 throughout the entirety of this project.



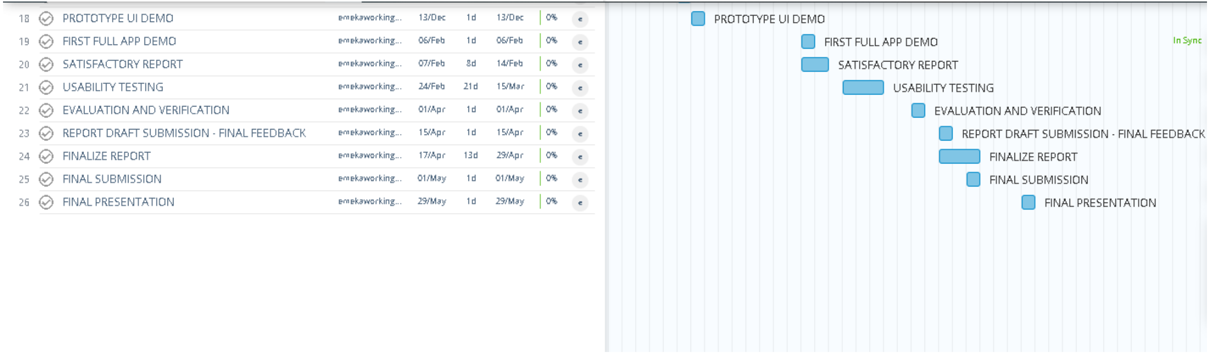
			SUB	ASSIGNEE	START	WD	DUE	%
	PROJECT IDEA				30/Sep	13d	12/Oct	0%
1	PID			emekaworking...	30/Sep	13d	12/Oct	0%
	RESEARCH(Ongoing)				10/Oct	129d	14/Apr	20%
4	EXISTING - APPS REVIEW - COMPARISON TABLE			emekaworking...	10/Oct	5d	17/Oct	100%
7	RESEARCH FOR LITERATURE REVIEW			emekaworking...	23/Oct	118d	06/Apr	0%
8	APPLICATION FEATURE DOCUMENT			emekaworking...	22/Oct	3d	24/Oct	0%
	WRITE LITERATURE REVIEW			emekaworking...	24/Oct	124d	14/Apr	0%
	DESIGN				20/Nov	1d	20/Nov	0%
11	CREATE INITIAL DESIGNS			emekaworking...	20/Nov	1d	20/Nov	0%
	IMPLEMENTATION				24/Oct	217d	25/May	0%
13	COMPLETE ANDROID TUTORIALS			emekaworking...	24/Oct	1d	24/Oct	0%
14	CREATE BASIC QUIZ APP			emekaworking...	28/Oct	2d	29/Oct	0%
15	DEVELOPING APP			emekaworking...	04/Nov	94d	06/Feb	0%
16	UI DESIGN - LOW & HIGH FIDELITY DESIGNS			emekaworking...	20/Nov	13d	11/Dec	0%
17	TABLE OF CONTENTS			emekaworking...	06/Dec	2d	07/Dec	0%
18	PROTOTYPE UI DEMO			emekaworking...	12/Dec	1d	12/Dec	0%
19	FIRST FULL APP DEMO			emekaworking...	06/Feb	1d	06/Feb	0%



18	PROTOTYPE UI DEMO	emekaworking...	12 Dec	1d	12 Dec	0%	+
19	FIRST FULL APP DEMO	emekaworking...	05 Feb	1d	05 Feb	0%	+
20	SATISFACTORY REPORT	emekaworking...	07 Feb	3d	12 Feb	0%	+
21	USABILITY TESTING	emekaworking...	26 Feb	21d	15 Mar	0%	+
22	EVALUATION AND VERIFICATION	emekaworking...	05 Apr	1d	05 Apr	0%	+
23	REPORT DRAFT SUBMISSION - FINAL FEEDBACK	emekaworking...	15 Apr	1d	15 Apr	0%	+
24	FINALIZE REPORT	emekaworking...	17 Apr	10d	26 Apr	0%	+
25	FINAL SUBMISSION	emekaworking...	01 May	1d	01 May	0%	+
26	FINAL PRESENTATION	emekaworking...	28 May	1d	28 May	0%	+

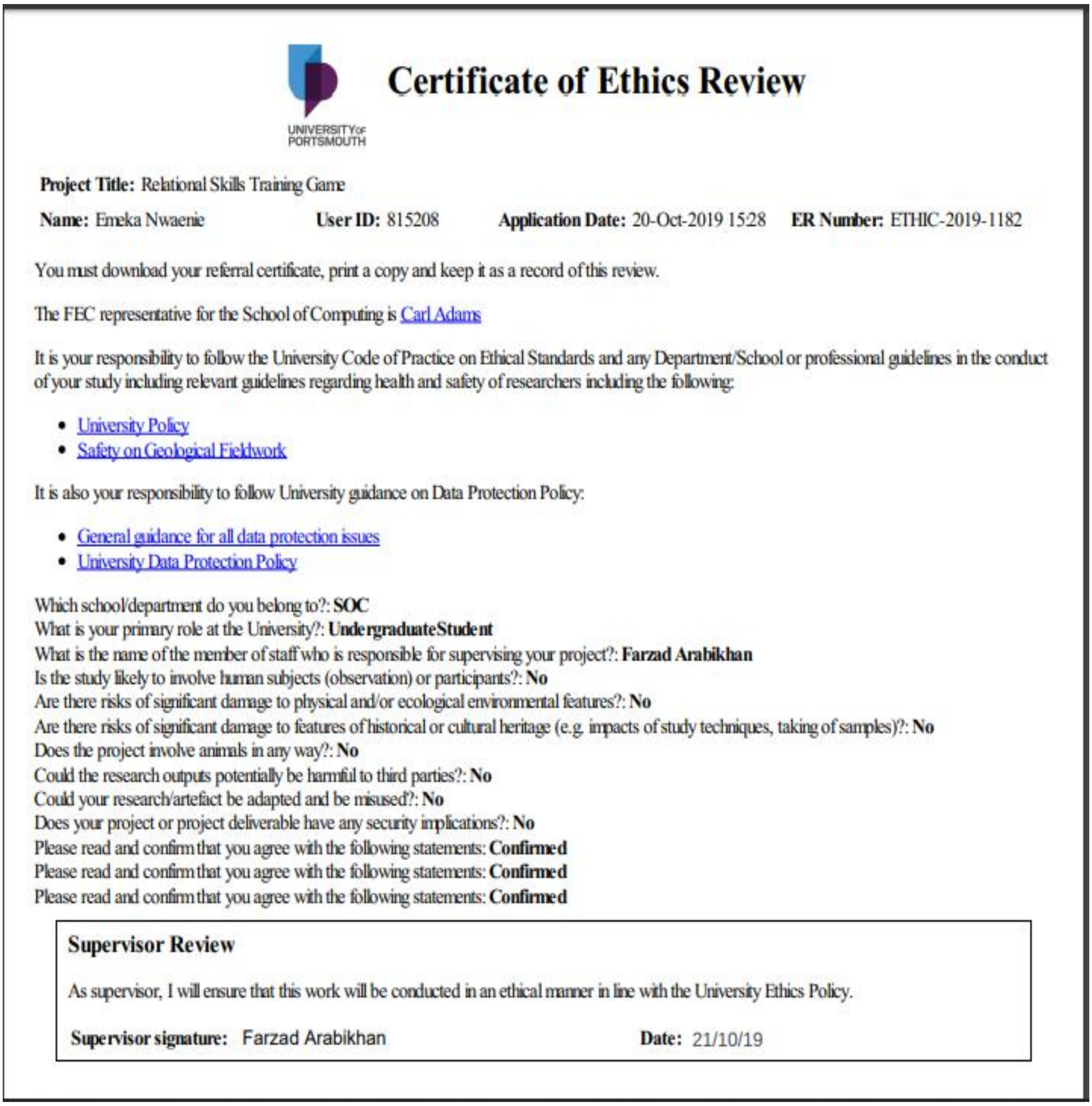


	Search	SUB	ASSIGNEE	START	WD	DUE	%
	PROJECT IDEA			30/Sep	23d	22/Oct	0%
1	PID		emekaworking...	30/Sep	23d	22/Oct	0%
	RESEARCH(Ongoing)			10/Oct	189d	15/Apr	25%
4	EXISTING - APPS REVIEW - COMPARISON TABLE		emekaworking...	10/Oct	8d	17/Oct	100%
5	RESEARCH FOR LITERATURE REVIEW		emekaworking...	20/Oct	169d	05/Apr	0%
6	APPLICATION FEATURE DOCUMENT		emekaworking...	22/Oct	3d	24/Oct	0%
	WRITE LITERATURE REVIEW		emekaworking...	25/Oct	174d	15/Apr	0%
	DESIGN			20/Nov	1d	20/Nov	0%
10	CREATE INITIAL DESIGNS		emekaworking...	20/Nov	1d	20/Nov	0%
	IMPLEMENTATION			26/Oct	217d	29/May	0%
13	COMPLETE ANDROID TUTORIALS		emekaworking...	26/Oct	1d	26/Oct	0%
14	CREATE BASIC QUIZ APP		emekaworking...	28/Oct	3d	30/Oct	0%
15	DEVELOPING APP		emekaworking...	04/Nov	95d	06/Feb	0%
16	UI DESIGN - LOW & HIGH FIDELITY DESIGNS		emekaworking...	29/Nov	13d	11/Dec	0%
17	TABLE OF CONTENTS		emekaworking...	06/Dec	2d	07/Dec	0%
18	PROTOTYPE UI DEMO		emekaworking...	13/Dec	1d	13/Dec	0%
19	FIRST FULL APP DEMO		emekaworking...	06/Feb	1d	06/Feb	0%



18	PROTOTYPE UI DEMO		emekaworking...	13/Dec	1d	13/Dec	0%
19	FIRST FULL APP DEMO		emekaworking...	06/Feb	1d	06/Feb	0%
20	SATISFACTORY REPORT		emekaworking...	07/Feb	8d	14/Feb	0%
21	USABILITY TESTING		emekaworking...	24/Feb	21d	15/Mar	0%
22	EVALUATION AND VERIFICATION		emekaworking...	01/Apr	1d	01/Apr	0%
23	REPORT DRAFT SUBMISSION - FINAL FEEDBACK		emekaworking...	15/Apr	1d	15/Apr	0%
24	FINALIZE REPORT		emekaworking...	17/Apr	13d	29/Apr	0%
25	FINAL SUBMISSION		emekaworking...	01/May	1d	01/May	0%
26	FINAL PRESENTATION		emekaworking...	29/May	1d	29/May	0%

Appendix B - Ethics form



UNIVERSITY OF PORTSMOUTH

Certificate of Ethics Review

Project Title: Relational Skills Training Game

Name: Emeka Nwaenie **User ID:** 815208 **Application Date:** 20-Oct-2019 15:28 **ER Number:** ETHIC-2019-1182

You must download your referral certificate, print a copy and keep it as a record of this review.

The FEC representative for the School of Computing is Carl Adams

It is your responsibility to follow the University Code of Practice on Ethical Standards and any Department/School or professional guidelines in the conduct of your study including relevant guidelines regarding health and safety of researchers including the following:

- University Policy
- Safety on Geological Fieldwork

It is also your responsibility to follow University guidance on Data Protection Policy:

- General guidance for all data protection issues
- University Data Protection Policy

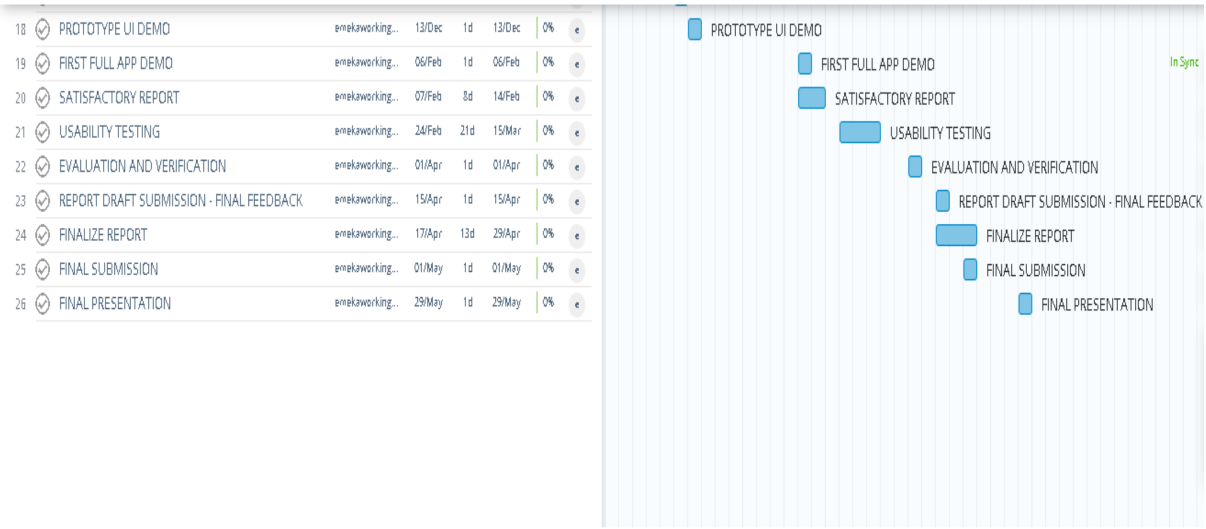
Which school/department do you belong to?: **SOC**
What is your primary role at the University?: **UndergraduateStudent**
What is the name of the member of staff who is responsible for supervising your project?: **Farzad Arabikhan**
Is the study likely to involve human subjects (observation) or participants?: **No**
Are there risks of significant damage to physical and/or ecological environmental features?: **No**
Are there risks of significant damage to features of historical or cultural heritage (e.g. impacts of study techniques, taking of samples)?: **No**
Does the project involve animals in any way?: **No**
Could the research outputs potentially be harmful to third parties?: **No**
Could your research/artefact be adapted and be misused?: **No**
Does your project or project deliverable have any security implications?: **No**
Please read and confirm that you agree with the following statements: **Confirmed**
Please read and confirm that you agree with the following statements: **Confirmed**
Please read and confirm that you agree with the following statements: **Confirmed**

Supervisor Review

As supervisor, I will ensure that this work will be conducted in an ethical manner in line with the University Ethics Policy.

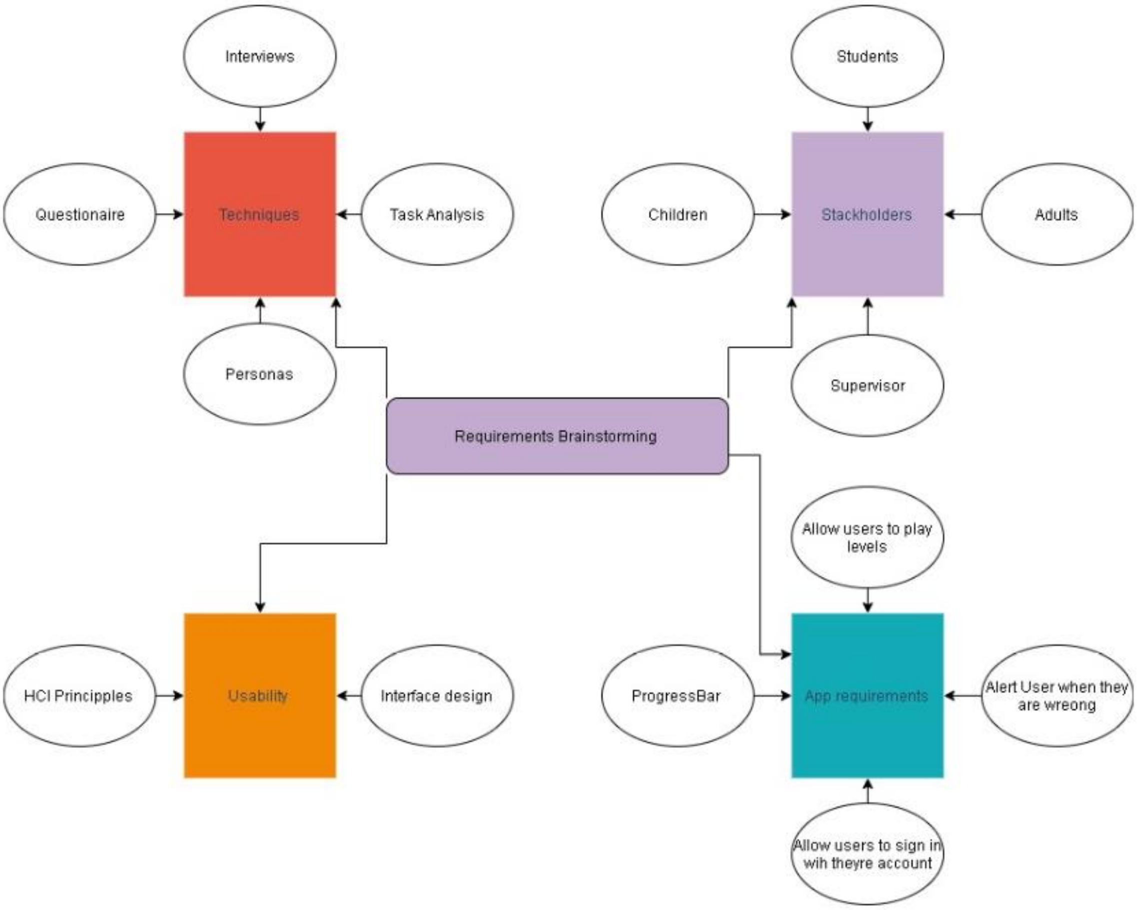
Supervisor signature: Farzad Arabikhan **Date:** 21/10/19

Appendix C - Revised Gantt Chart

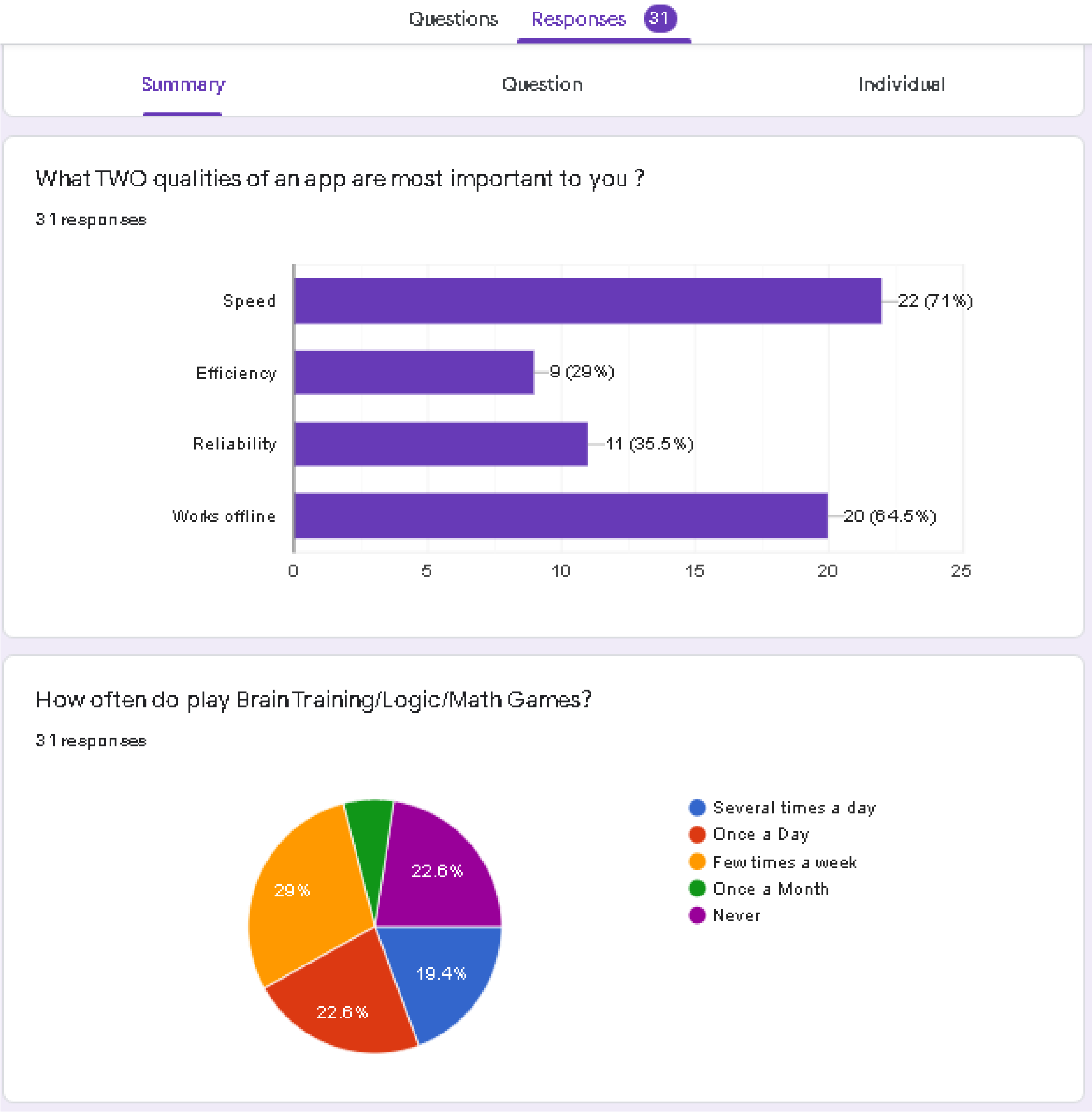


18	✓	PROTOTYPE UI DEMO	emekaworking...	13/Dec	1d	13/Dec	0%	e
19	✓	FIRST FULL APP DEMO	emekaworking...	06/Feb	1d	06/Feb	0%	e
20	✓	SATISFACTORY REPORT	emekaworking...	07/Feb	8d	14/Feb	0%	e
21	✓	USABILITY TESTING	emekaworking...	24/Feb	21d	15/Mar	0%	e
22	✓	EVALUATION AND VERIFICATION	emekaworking...	01/Apr	1d	01/Apr	0%	e
23	✓	REPORT DRAFT SUBMISSION - FINAL FEEDBACK	emekaworking...	15/Apr	1d	15/Apr	0%	e
24	✓	FINALIZE REPORT	emekaworking...	17/Apr	13d	29/Apr	0%	e
25	✓	FINAL SUBMISSION	emekaworking...	01/May	1d	01/May	0%	e
26	✓	FINAL PRESENTATION	emekaworking...	29/May	1d	29/May	0%	e

Appendix D - Mind Map Requirements



Appendix E - Questionnaire

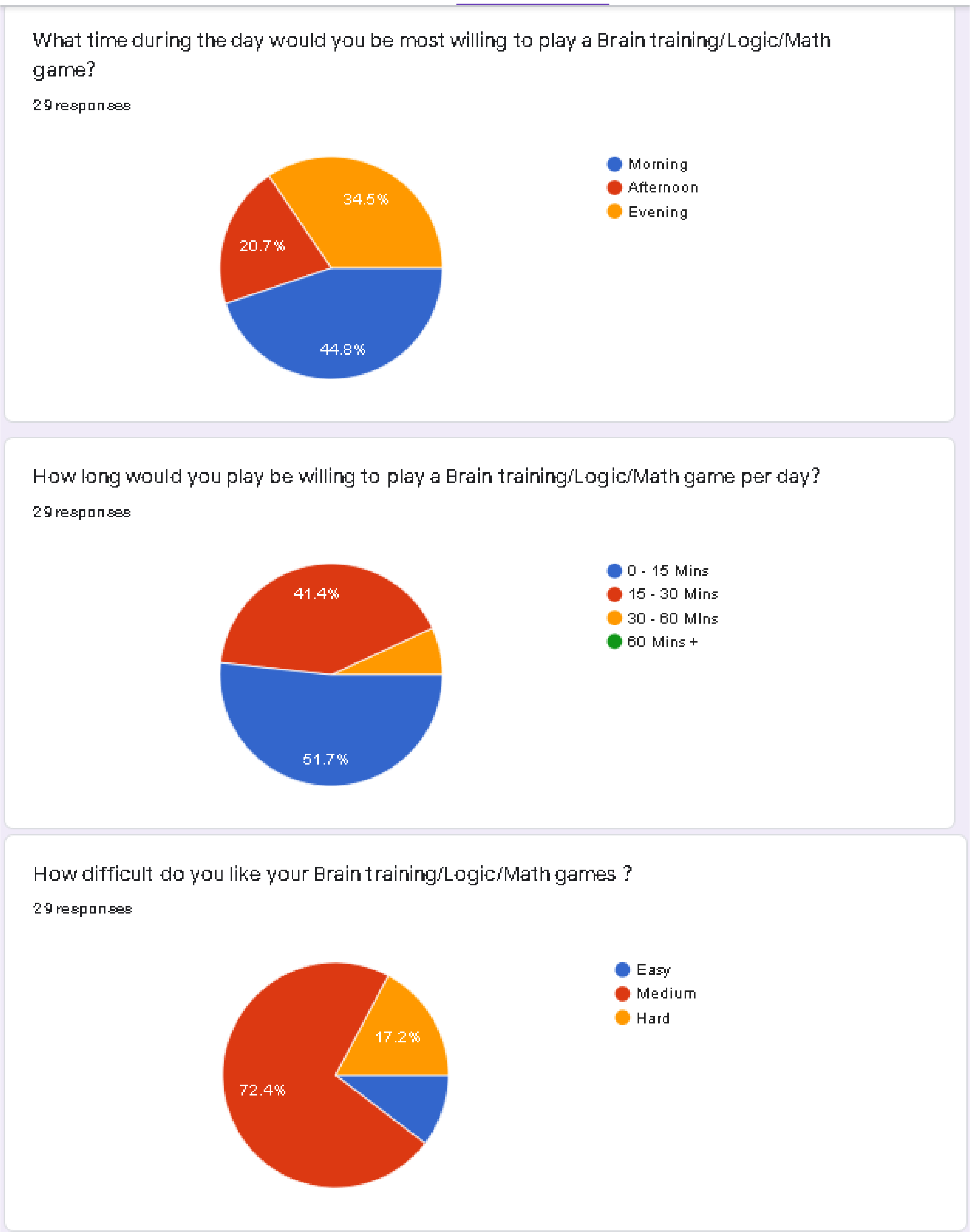


What TWO qualities of an app are most important to you ?

31 responses

How often do play Brain Training/Logic/Math Games?

31 responses



What time during the day would you be most willing to play a Brain training/Logic/Math game?

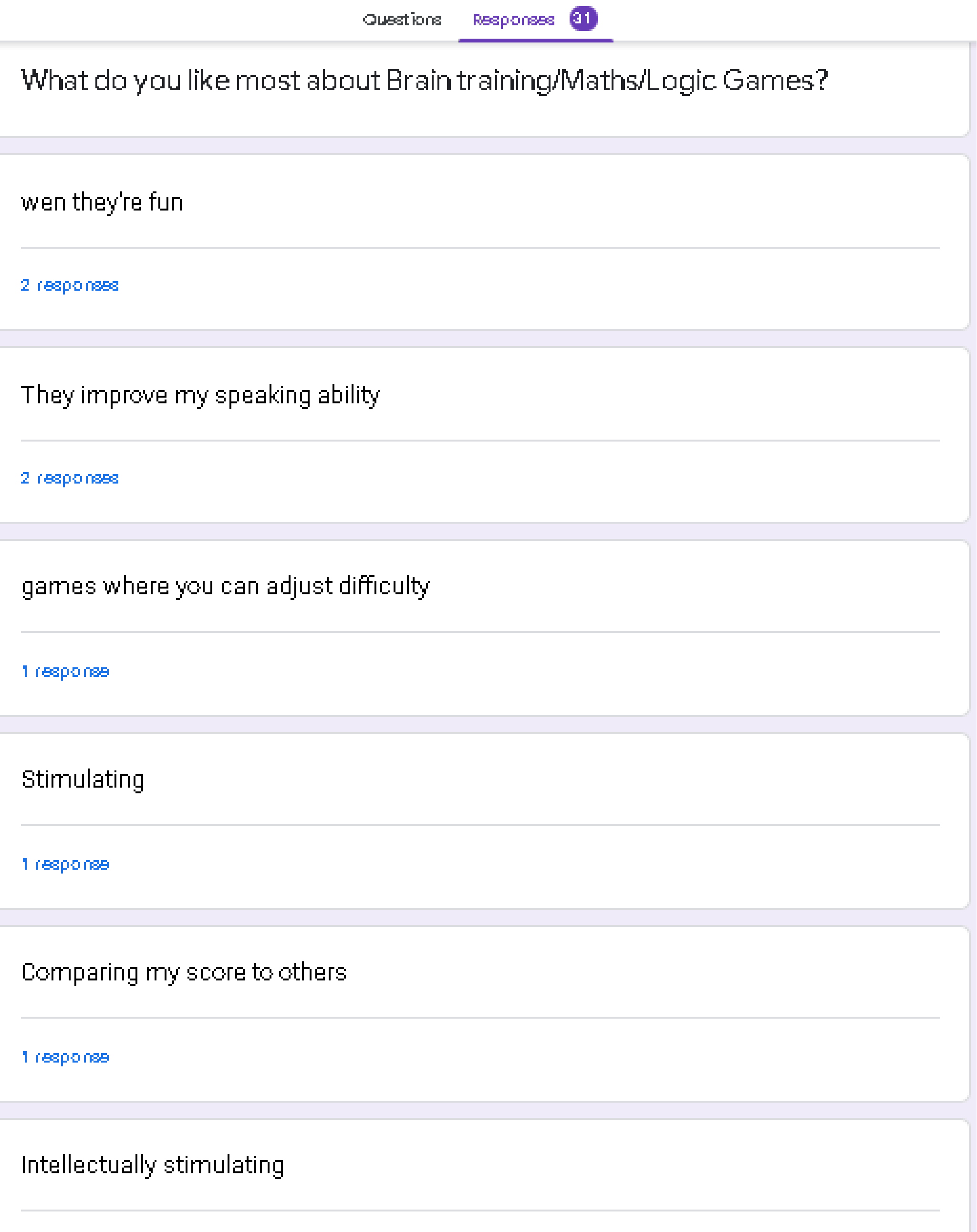
29 responses

How long would you play be willing to play a Brain training/Logic/Math game per day?

29 responses

How difficult do you like your Brain training/Logic/Math games ?

29 responses



What do you like most about Brain training/Maths/Logic Games?

wen they're fun

2 responses

They improve my speaking ability

2 responses

games where you can adjust difficulty

1 response

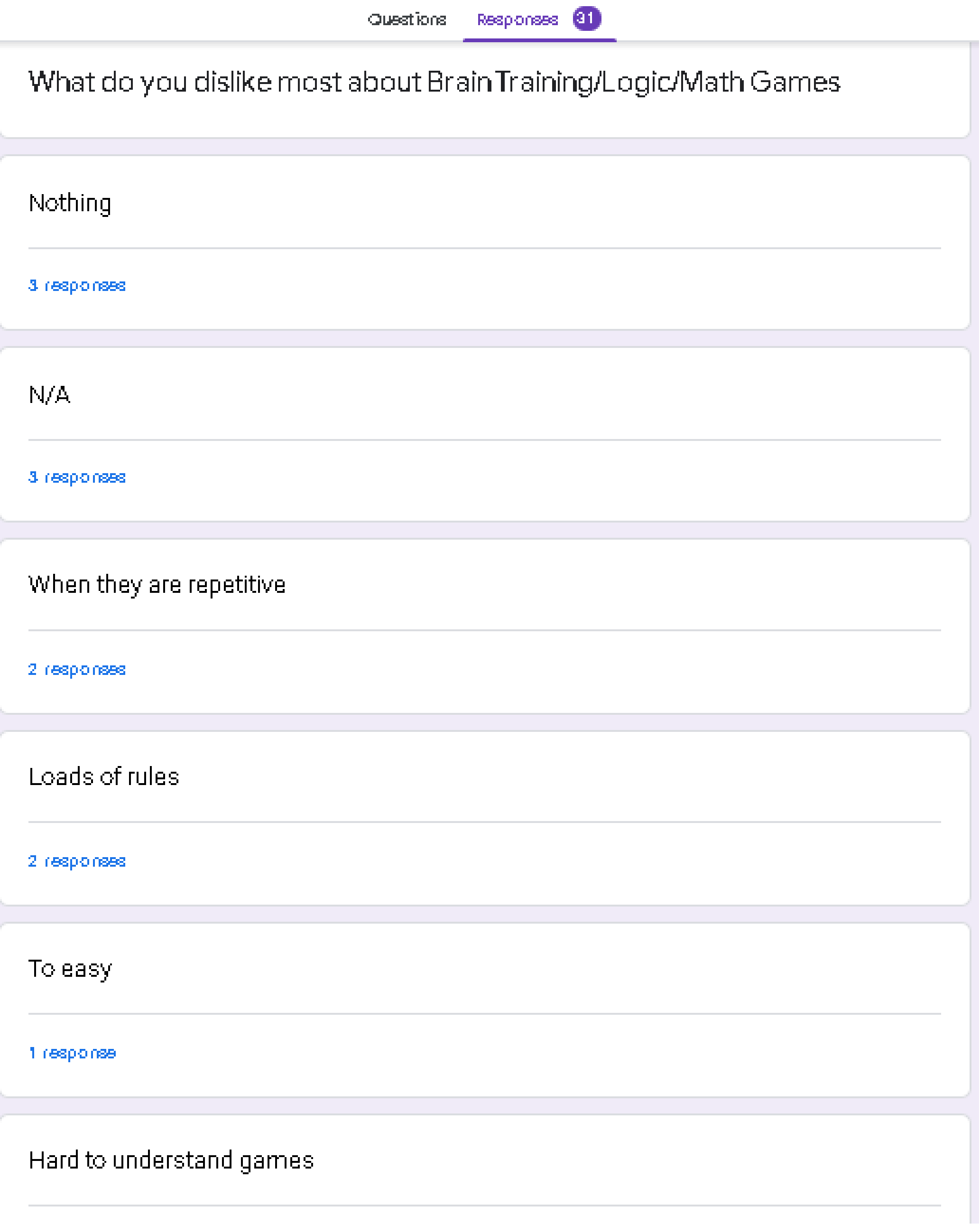
Stimulating

1 response

Comparing my score to others

1 response

Intellectually stimulating



What do you dislike most about Brain Training/Logic/Math Games

Nothing

3 responses

N/A

3 responses

When they are repetitive

2 responses

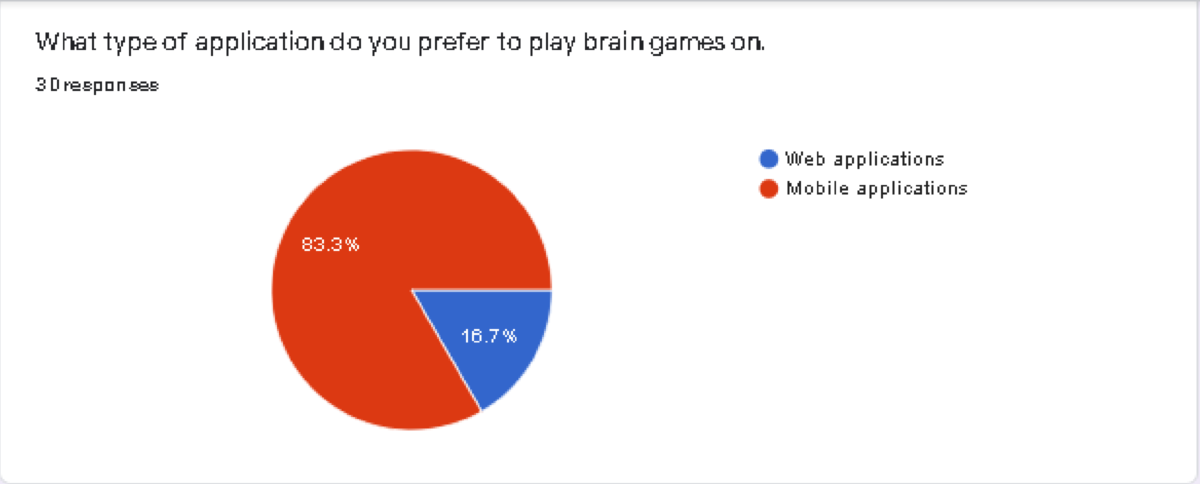
Loads of rules

2 responses

To easy

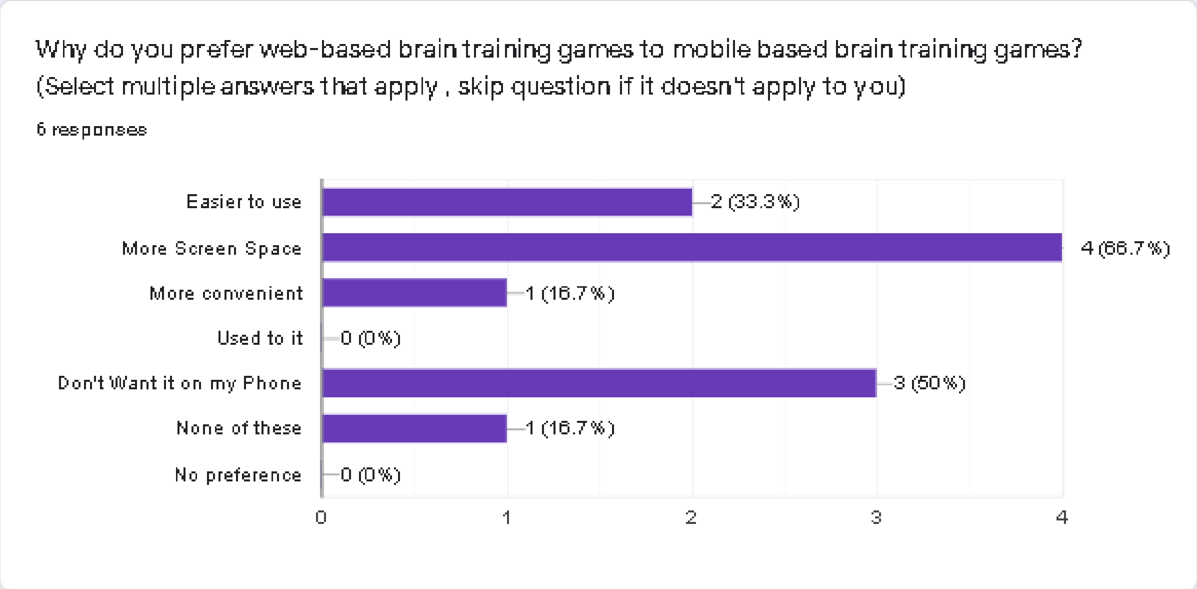
1 response

Hard to understand games



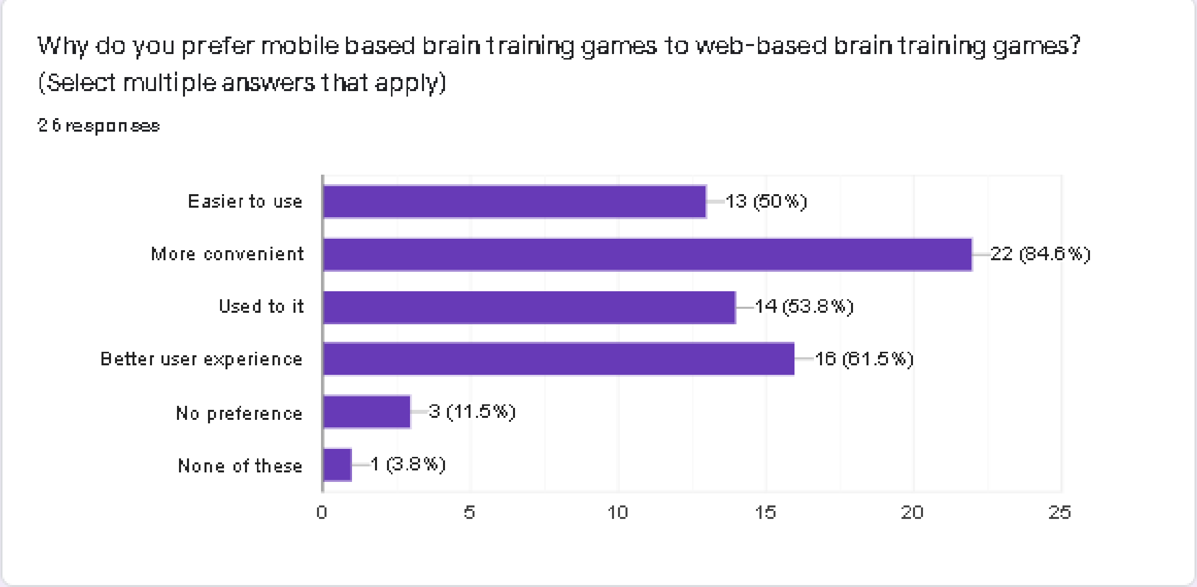
What type of application do you prefer to play brain games on.

30 responses



Why do you prefer web-based brain training games to mobile based brain training games? (Select multiple answers that apply , skip question if it doesn't apply to you)

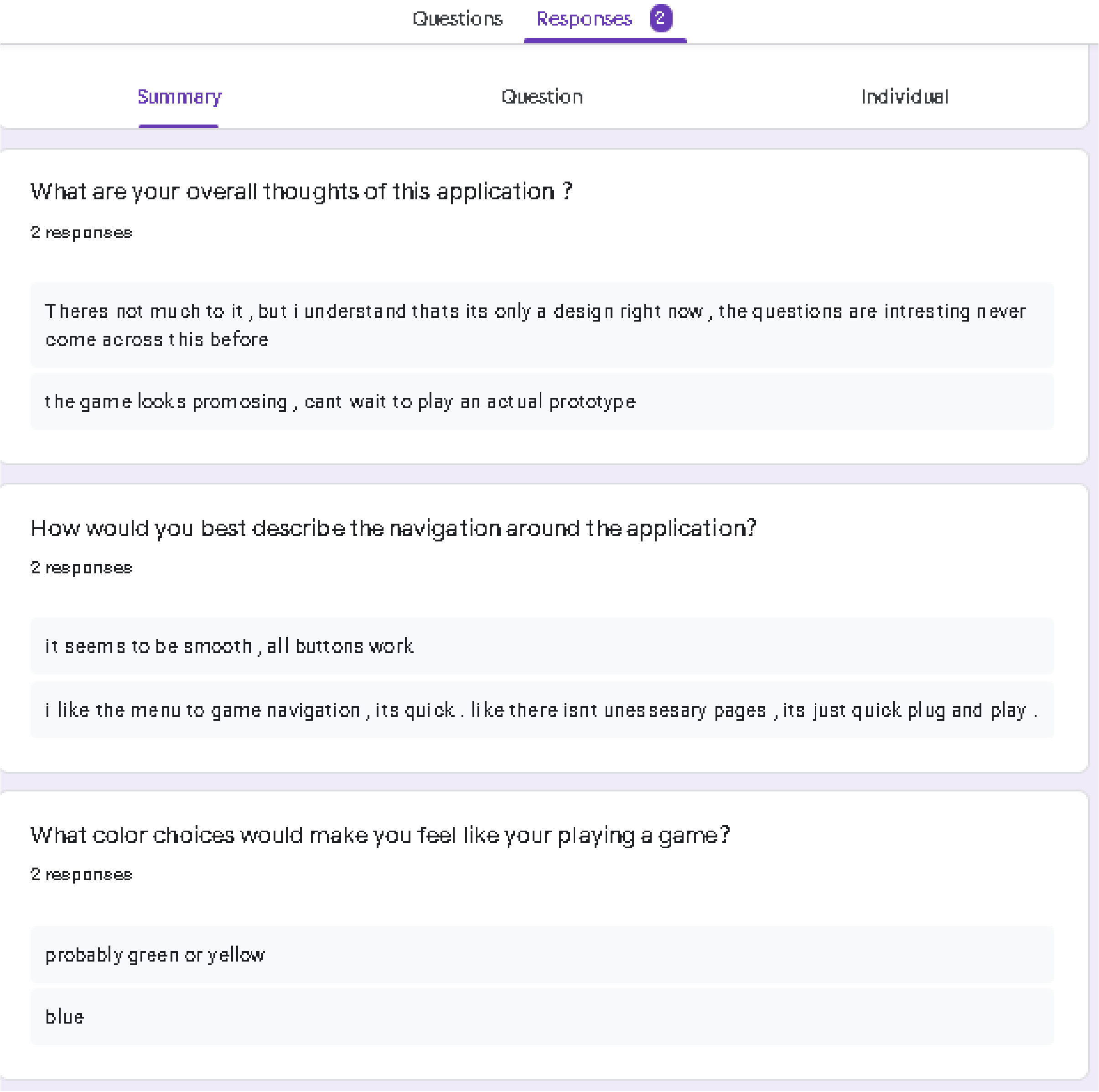
6 responses



Why do you prefer mobile based brain training games to web-based brain training games? (Select multiple answers that apply)

26 responses

Appendix F - Low fidelity prototype interview



What are your overall thoughts of this application ?

2 responses

Theres not much to it , but i understand thats its only a design right now , the questions are intresting never come across this before

the game looks promosing , cant wait to play an actual prototype

How would you best describe the navigation around the application?

2 responses

it seems to be smooth , all buttons work

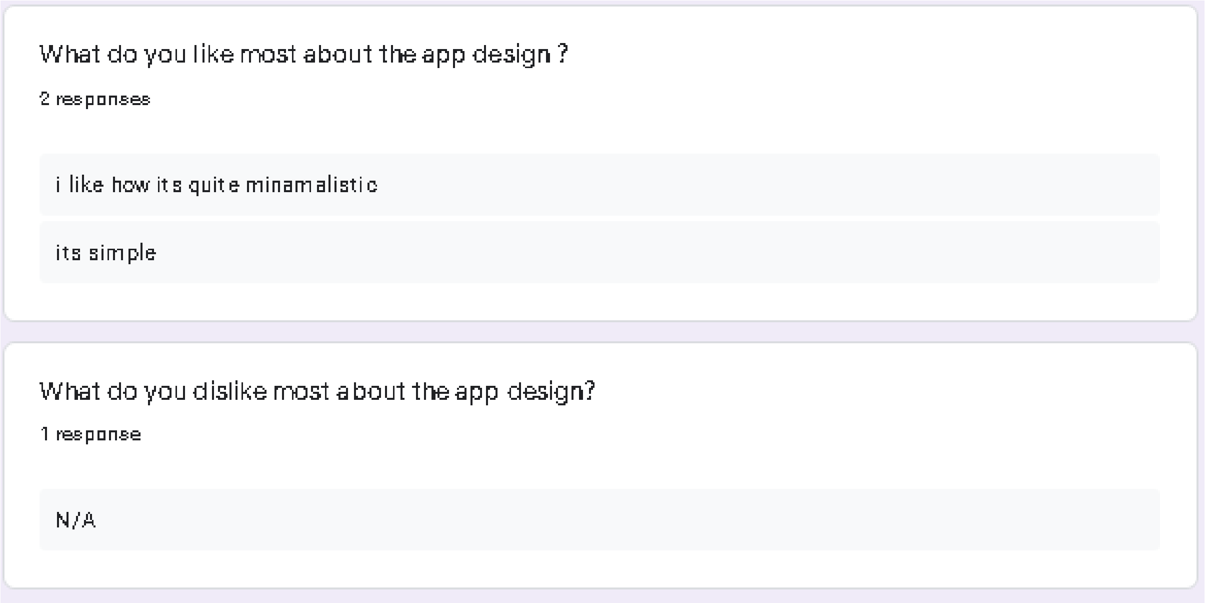
i like the menu to game navigation , its quick . like there isnt unessesary pages , its just quick plug and play .

What color choices would make you feel like your playing a game?

2 responses

probably green or yellow

blue



What do you like most about the app design ?

2 responses

i like how its quite minamalistic

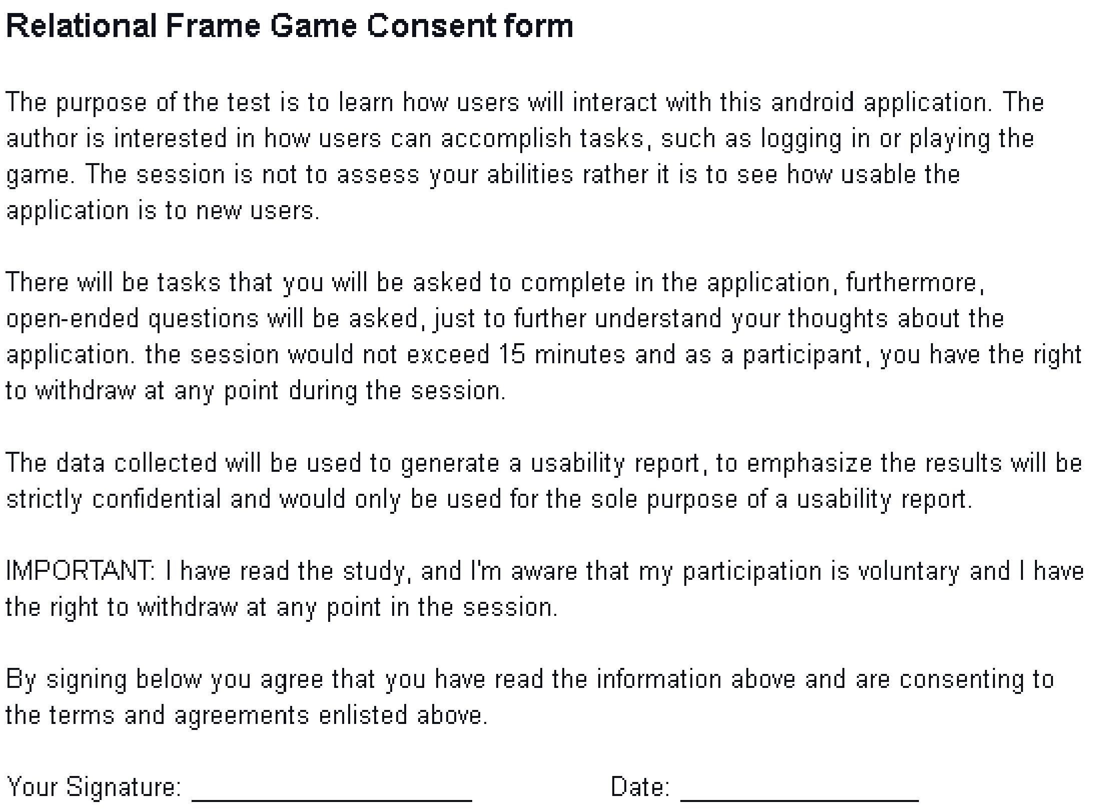
its simple

What do you dislike most about the app design?

1 response

N/A

Appendix G - Usability Tests



Relational Frame Game Consent form

The purpose of the test is to learn how users will interact with this android application. The author is interested in how users can accomplish tasks, such as logging in or playing the game. The session is not to assess your abilities rather it is to see how usable the application is to new users.

There will be tasks that you will be asked to complete in the application, furthermore, open-ended questions will be asked, just to further understand your thoughts about the application. the session would not exceed 15 minutes and as a participant, you have the right to withdraw at any point during the session.

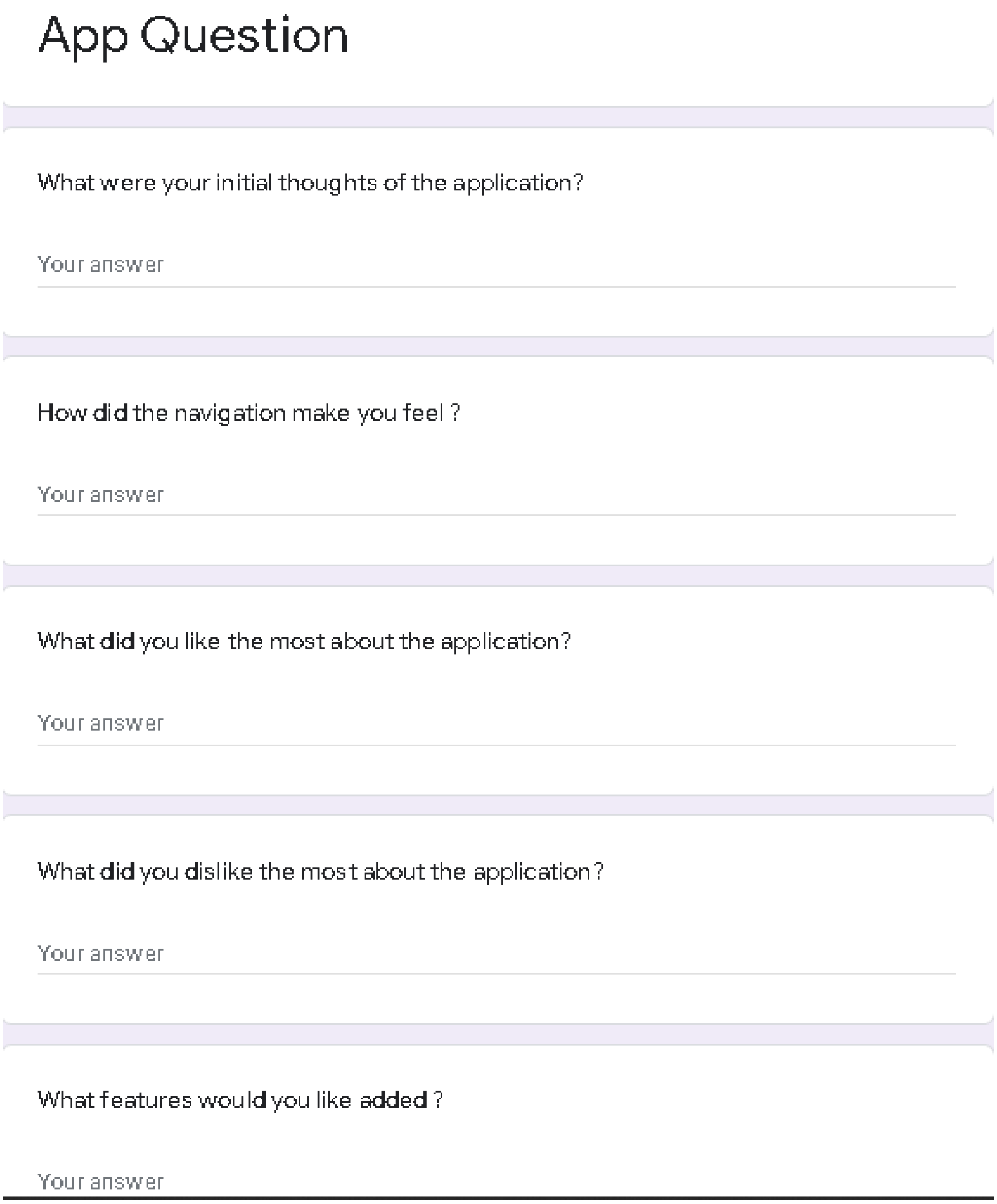
The data collected will be used to generate a usability report, to emphasize the results will be strictly confidential and would only be used for the sole purpose of a usability report.

IMPORTANT: I have read the study, and I'm aware that my participation is voluntary and I have the right to withdraw at any point in the session.

By signing below you agree that you have read the information above and are consenting to the terms and agreements enlisted above.

Your Signature: ____________________ Date: _________________

Scenario	Task
You're interested in playing this new logic game you've never seen before however it requires you to sign up.	Sign in and login
You've successfully logged in but don’t really understand the game	Read the instructions page to understand the game



App Question

What were your initial thoughts of the application?

Your answer

How did the navigation make you feel ?

Your answer

What did you like the most about the application?

Your answer

What did you dislike the most about the application?

Your answer

What features would you like added ?

Your answer