|  |
| --- |
| HCC2, HCCE2, BSHC2, BSHCE2, BSHBIS2, BSHBISE2 |
| Requirements Specification (RS) |
|  |

|  |
| --- |
| Abigail Boyle(x17323943)  Roberto Niculita(x17747341)  Tijesu Olalekan(x17347773)  Luke Sheehan (x17361401)  Wiktor Wolsza(x17444592) |

Requirements Specification (RS)

**Document Control**

**Revision History**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Date** | **Version** | **Scope of Activity** | **Prepared** | **Reviewed** | **Approved** |
| 14/10/2005 | 1 | Create | AB | X | X |
| 21/10/05 | 2 | Update | CD |  |  |

**Distribution List**

|  |  |  |
| --- | --- | --- |
| **Name** | **Title** | **Version** |
| Arghir Moldovan | Lecturer |  |
| Eamon Nolan | Lecturer |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Related Documents**

|  |  |
| --- | --- |
| **Title** | **Comments** |
| Title of Use Case Model |  |
| Title of Use Case Description |  |

**Table of Contents**

[1 Introduction](#_4d34og8) **4**

1.1 [Purpose](#_2s8eyo1) 4

1.2 [Project Scope](#_17dp8vu) 4

1.3 [Definitions, Acronyms, and Abbreviations](#_3rdcrjn) 5

2 [User Requirements Definition](#_26in1rg) **5**

[2.1 Introduction](#_7ps0cjn9jt2e) 5

[2.2 Sample Results From Survey](#_jwzc7jsmi7g) 7

[2.3 Survey Analysis](#_2etqll9pyykj) 10

[2.4 Younger User Requirements](#_pxhco059x77h) 11

[2.4.1 Age appropriate design](#_te4jzsw1ahkf) 11

[2.4.2 Feedback First](#_mrxpn1g8wj75) 11

[2.5 Functional User Requirements](#_5pyir03ad6vi) 12

3 [Requirements Specification](#_lnxbz9) **14**

3.1 [Functional requirements](#_35nkun2) 14

3.1.1 [Use Case Diagram](#_1ksv4uv) 16

3.1.2 [Requirement 1: User Account Create/ Login](#_44sinio) 17

3.1.2.1 [Description & Priority](#_is3ukzu962ay) 17

3.1.2.2 [Use Case](#_wfgxbgooaxz1) 17

3.1.3 [Requirement 2: <Category Selection>](#_2jxsxqh) 19

3.2.1.1 [Description & Priority](#_shp215f18rvn) 19

3.2.1.2 [Use Case](#_bj56q6iwhbdv) 19

3.1.4 [Requirement 3: <Provide user with questions and options with a time limit>](#_snl92q18nqra) 20

3.1.4.1 [Description & Priority](#_j1wjlgstfjgj) 20

3.1.4.2 [Use Case](#_reh7dx6xg2ly) 20

3.1.5 [Requirement 4: <Provide user with a score based on their performance>](#_s69qnmsktae1) 22

3.1.5.1 [Description & Priority](#_o7f2gkpyzxej) 22

3.1.5.2 [Use Case](#_y0vngnwvpg3o) 22

3.1.6 [Requirement 5: <Allow user to view their previous scores>](#_8dm9bnptoznq) 24

3.1.6.1 [Description & Priority](#_fbd8qhqhcktw) 24

3.1.6.2 [Use Case](#_c5p35oohex82) 24

3.1.7 [Requirement 6: <Allow user to buy features/extra categories of question>](#_vvw11130xu2x) 25

3.1.7.1 [Description & Priority](#_d1s9l5ncy8p5) 25

3.1.7.2 [Use Case](#_iil5vwihh78g) 25

3.2 [Non-Functional Requirements](#_z337ya) 27

3.2.1 [Performance/Response time requirement](#_3j2qqm3) 27

3.2.2 [Availability requirement](#_1y810tw) 28

3.2.3 [Recover requirement](#_4i7ojhp) 28

3.2.4 [Security requirement](#_1ci93xb) 28

[3.2.4.1 Confidentiality:](#_5hfvogybmnwq) 28

[3.2.4.2 Authentication:](#_dllp3iax8l0o) 29

[3.2.4.3 Error Management:](#_lcfyejnq2ai) 29

3.2.5 [Reliability requirement](#_3whwml4) 29

3.2.6 [Extendibility requirement](#_3as4poj) 29

3.2.7 [Reusability requirement](#_1pxezwc) 30

[4 GUI](#_14nlse11xx86) **30**

5 [System Architecture](#_147n2zr) **34**

6 [System Evolution](#_3o7alnk) **35**

# 1 Introduction

## Purpose

The purpose of this document is to set out the requirements for the development of a brain training application. This brain training application is to help children from ages 8 to 12 improve their mental abilities in different areas. The game will present users puzzles and challenges to complete and record their answers and progress.The system will be based on a relational database model. As users continue to play, data of the users performance in various tasks will be collected and presented in a visual format. This data can be seen by the child and the child’s parent/guardian. With this information the parent and child can use the data as a guide to see which areas they have room to improve in.

This document will outline the requirements for our application, both functional and non-functional. For the purpose of gathering requirements, we used other online materials and a survey created by a member of team.

## Project Scope

The scope of the project is to develop a functional quiz application. The game will be made using Android Studio, and due to a time limit on our project will only be developed on android platform. The application will be specifically created for children between 8 and 12, however the varying types of games and questions available with the program allows users of all ages to enjoy elements of the application. Children use a large quantity of games on app-stores across both IOS and Android software. These applications all offer different types of games, with our brain training application we aim to provide a ‘one-stop for all’ and use several games differing in skill. The application will engage children by having challenging, fun and educational games with easy to use interfaces.The games will have five categories of questions, each one created in order to improve various areas of the mind.

The system shall have:

· A Login page

· Sign up page

· Home page

· 5 Categories of Quiz

· Questions for each Category

· User Profile section

· Game Store, for in-game purchases

## Definitions, Acronyms, and Abbreviations

UR - User Requirements

App- Application

# User Requirements Definition

## 2.1 Introduction

Our application is being designed for younger users aged between 8 and 12, because our target market is aimed towards individuals who are under the age of 18 we must also take into account parent/guardian's approval of the application which is fundamental to the applications success. Therefore, the user requirements will be a combination of functional and non-function requirements developed upon the design needs of the actual users (children between 8 and 12) and the needed security and data requirements of the parents/guardians of the children using the application.

Our project scope is to develop a gaming style application that will be both fun and educational. The application will comprise of multiple games, each designed to help the user develop their skill in different mental areas such as logical, analytical, memory and linguistics. After completing the games, the user can monitor and review their progress. Their performance in the games will be stored and the parent of the child playing the game can monitor the performance and set goals for their child to reach. Since the aim of our application is for it to be both educational and fun to play, we have to create an element of challenge and reward in order to encourage the child to play the game. We will introduce the idea of a points system where the user will gain points depending how accurate and quick their answers are. Points will be accumulated and stored on the players account, leveling up their account as they reach targets. The application will also constantly praise and encourage the user as they answer and level up. Points can be used to purchase extra categories of question.The system provides the user with the option to login with their account on the homepage of the application when the application is started for the first time.

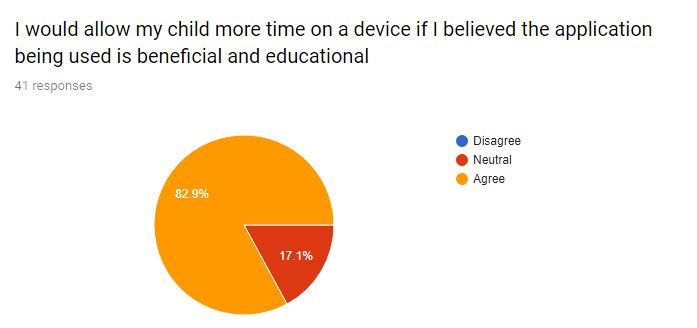
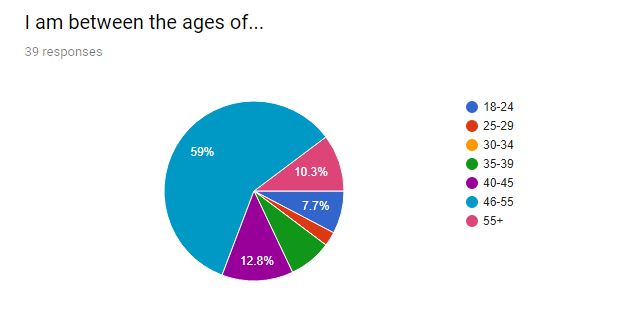
The user enters their account details and gives the application and the system logs them in.

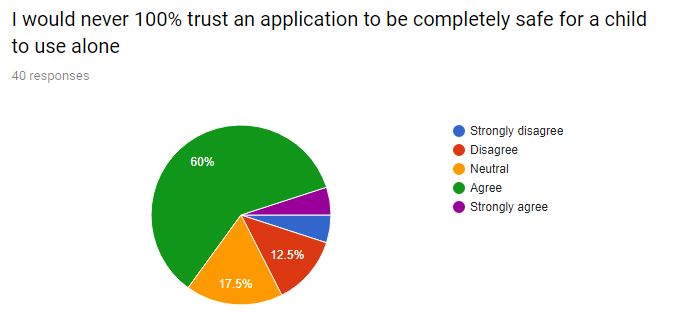
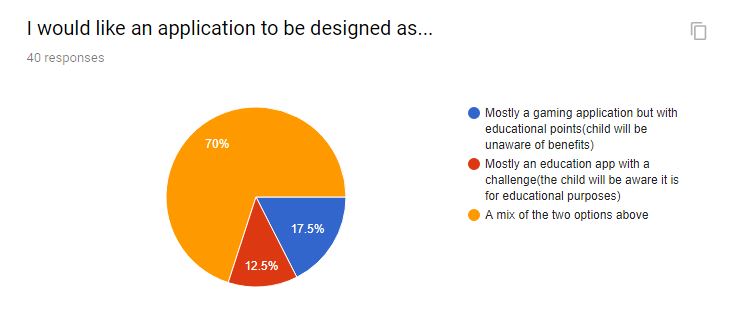
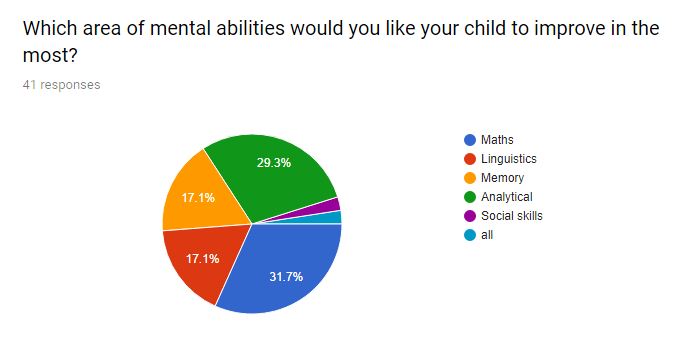
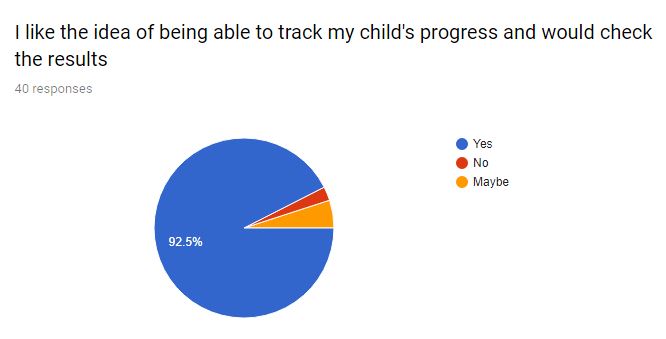
We used Google Forms and created a survey with several questions. We asked parents of all ages to complete the survey. The questions were asked in such a way that we could extract exactly what parents would require from the application in terms of child safety and making purchases.

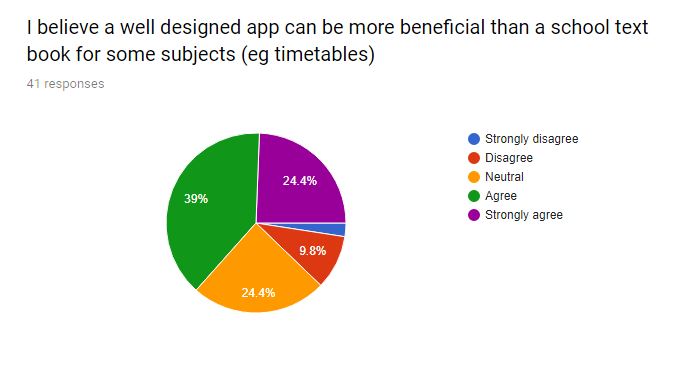
For the requirements of the younger users, we took into account what the parents had answered in the survey and also researched into applications that are popular today with youth.

(Duolingo, The Brain, CogniFit, BrainWell, Brain Trainer, Lumosity, NeuroNation)

## 2.2 Sample Results From Survey

****

****

****

## 2.3 Survey Analysis

Our survey was used to find the requirements of parents. As explained above, it was essential to outline the requirements of parents since they usually finance their children’s applications and also their approval of an application will determine whether or not a child will use it or not in most cases.

63.4% either agreed or strongly agreed that an application can be more beneficial than a school textbook.

⅓ of responses said that they would like their child to improve the most in math ability.

82.9% said that they would allow their child more time on a device if they believed the application to be educational and beneficial.

92.5% they would track their child’s progress.

A huge 70% of responders indicated that they would never totally trust an application.

From these results we could see that there is a demand for an application like ours and that parents would be eager to review their child's progress in categories of mental ability. However, we can also see that security and child safety are still huge factors to consider in requirements and that it is essential to program our application to be suitable for the ages of our users and not allow easy access to purchasing in app extras etc.

The fact that most responders indicated that maths was the biggest area they would like their child to improve in suggested to the team that we perhaps must develop the maths questions and puzzles in detail. We figured that a more developed maths section will most likely contribute highly to the applications success.

## 2.4 Younger User Requirements

### 2.4.1 Age appropriate design

Children are typically consciously aware of age design.(Sherwin, 2019). New research performed with users aged 3–12 shows that children have obtained substantial proficiency in working with websites and apps in modern times, though many designs are still not optimized for younger users. Designing for children is a difficult procedure that must use distinct usability approaches, including targeting content narrowly for children of different ages. For example, an application designed for a four year old would differ a lot when compared with an application designed for an eight year old. With adults, an application can be designed for a much larger age group but with children, an efficient divide in user base is essential.

Although it is true that children and adults differ vastly in reasons to use a website or application and that children prefer animations and sounds being used, both groups prefer high quality design and are inclined to leave a website or application behind if the GUI is of low quality or difficult to use.

From reading this study, the team decided that our targeted users require a high quality age appropriate GUI.

### 2.4.2 Feedback First

According to a study done by Sean McGowan on Child UX Design, younger users require constant feedback on everything, it's how children learn about and understand the world around them. Therefore, our application will be sure to include dialog boxes with encouraging messages to the users on their progress. For example, “You are getting much faster at answering subtraction!”.

## 2.5 Functional User Requirements

UR1: Performance will be stored on the players account and accessed at any time after correct login details are given.

UR2: Player has free will in what games they can play.

UR3: The games will run smoothly and when the player answers a question the system must recognize that answer immediately.

UR4: The applications interface shall be menu driven. It will provide dialog boxes and help screens to navigate the user through all the features.

UR5: Players can adjust the level settings in order to obtain harder or easier levels.

UR6: The games must be simple to navigate but tougher in solving the answer.

UR7: The games must give constant feedback and encouragement to the user.

UR8: The games must differ in challenge and be related to the category of mental ability they are assigned to.

UR9: The games must be suitable for ages 8 to 12.

UR10: The applications content must be only 100% child friendly.

UR11: The avatar’s must be suitable for children.

UR12: The quiz questions must be random and not repetitive.

UR13: The applications layout must be consistent.

UR14: The games must be engaging so that the younger user is entertained and motivated to progress through the application.

UR15: The games must be a mixture of visual and auditory challenges in order to suit the learning styles of our all users.

UR16: The report of the user's progress must be accurate and *only* related to that users progress.

UR17: The report on player progress can only be accessed by the account the report is assigned to. The application must not feature an ability to check other players reports.

UR18: The application must not grant children using the application access to in-app purchases without parents consent.

UR19: The games must have a wide range of questions that will shuffle each time a round of that particular game is loaded.

UR20: The questions asked by the games must have a large enough range so that players cannot become familiar with questions at ease and score higher due to completing that exact question only a short time before hand.

UR21: The application must have the ability to disable features/games that require the use of a speaker.

UR22: The application must be accessible from both mobile and tablet devices.

UR23: The application must give constant feedback to the user and encourage them to continue on in the puzzles.

**2.6 Non-Functional User Requirements**

UR23: The application will load a loading page as soon as a player clicks into it.

UR24: The application will be totally child friendly, no inappropriate advertisements or adult content.

UR25: The system will be able hold up to 10,000 user’s information initially

UR26: The system will store data for at least one year

UR27: The system will be available 24/7

UR28: The application will ask for permission of speaker use.

UR29: The application will have security measures in place in order to protect confidential data and player reports being accessed by unauthorised individuals.

UR30: Each category of games will hold at least 50 different questions.

UR31: The application will be capable of shuffling questions in order to make the game difficulty harder.

UR32: The application will not use the ‘pay by fingerprint’ feature that android provides on various devices.

UR33: The system will only accept the user’s play store password as confirmation of payment.

UR34: The application will always ask for users password if a request is made for in app purchases.

UR35: The application will feature games that include the use of a speaker.

UR36: The application will feature the ability to play games without a speaker.

UR37:The application will feature three levels of difficulty: beginner,intermediate,advanced.

# Requirements Specification

All requirements should be verifiable. For example, experienced controllers shall be able to use all the system functions after a total of two hours training. After this training, the average number of errors made by experienced users shall not exceed two per day.

## Functional requirements

This section lists the functional requirements in **ranked order**. Functional requirements describe the possible effects of a software system, in other words, *what* the system must accomplish. Other kinds of requirements (such as interface requirements, performance requirements, or reliability requirements) describe *how* the system accomplishes its functional requirements. Each functional requirement should be specified in a format similar to the following:

Short, imperative sentence stating highest ranked functional requirement.

Functional requirements:

1. Allow user to create account

2. Allow user to choose category of question

3. Provide user with questions and options with a time limit

4. Provide user with a score on how they did

5. Allow user to view their previous scores

6. Allow user to buy features/extra categories of question

### Use Case Diagram

### Requirement 1: User Account Create/ Login

#### Description & Priority

This purpose of this requirement is to give users the ability to create an account when they enter the app and let them log in with that account. The users can then log in with their created accounts when they enter the game. It is important for users to have accounts in our application so that the users results and scores can be saved in their profile. This feature is important since we want users to be able to track their results and progress.

#### Use Case

Tag: FRlog

**Scope**

The scope of this use case is to give users the ability to create an account when they enter the app and then let them log in with their created account. The users can the log in with their created accounts when they enter the game. It is important for users to have accounts in our application so that the users results and scores can be saved in their profile. This feature is important since we want users to be able to track their results and progress.

**Description**

This use case describes the creation of an account when the user first enters the application.

**Flow Description**

**Precondition**

The user enters the application for the first time.

**Activation**

This use case starts when an <Actor> selects the create account option on the homepage of the app.

**Main flow**

1. The system displays the user with the text boxes: “Enter username”, “Enter Password” and “Re-enter password”. The system also provides the user with an enter button to confirm their username and password and a back button which returns the user to the home page.
2. The user enters a username and the enters a password.
3. The user re-types their password to avoid mistakes when entering it.
4. The users account is created.

**Alternate flow**

A1 : <login>

1. The system provides the user with the option to login with their account on the homepage of the application when the application is started for the first time.
2. The user enters their account details and gives the application and the system logs them in.

**Exceptional flow**

E1 : <login>

1. The system provides the user with the option to login with their account on the homepage of the application when the application is started for the first time.
2. The user enters their account details and gives the application and the system logs them in.

**Termination**

The users clicks enter on the create account page of the application.

**Post condition**

The system proceeds to the select question category stage of the application.

### Requirement 2: <Category Selection>

#### Description & Priority

This requirement is needed so that the user can select what category of question they will be quizzed on. This feature is important since we want the users to be able to find out what areas of question they are strongest/weakest on.

#### Use Case

FRcat

**Scope**

The scope of this use case is to have different categories of questions available for the user to be quizzed on.

**Description**

This use case describes the interaction between the user and the ‘choose category’ screen of the application.

**Flow Description**

**Precondition**

The user has signed into their account and is on the ‘select category’ of question

page’

**Activation**

This use case starts when the user has signed into their account.

**Main flow**

1. The system displays five categories of questions for the user

2. The user selects what category of question they want to be tested on

3. Once selected, the system provides the user with options: “Start Quiz” or Cancel

4. Selecting “Start Quiz” will start the quiz for the user

**Alternate flow**

A1 : <Not selecting a category>

1.The system provides the user with categories of question.

2.The user selects what category of question they want to be tested on and is presented with the options: “Start Quiz” and “Cancel”

3.Selecting “Cancel” will return the user to the ‘select category’ page of the application.

**Exceptional flow**

E1 : <title of E1>

1. The system displays five categories of questions for the user

2. The user selects what category of question they want to be tested on

3. Once selected, the system provides the user with options: “Start Quiz” or Cancel

4. Selecting “Start Quiz” will start the quiz for the user.

**Termination**

The user selects “Start Quiz” or “Cancel”.

**Post condition**

The system begins the quiz section of the application.

### Requirement 3: <Provide user with questions and options with a time limit>

#### Description & Priority

This requirement is for the quiz section of the application. It has priority since the application is built around users answering questions under a time limit. The time limit is a feature since we want to reward the users with more points based on how fast they answer questions correctly.

#### Use Case

FRquiz

**Scope**

The scope of this use case is to provide users with questions and options under a time limit.

**Description**

This use case describes how the the user would interact with the quiz section of

the application.

**Flow Description**

**Precondition**

The system is in the quiz game mode.

**Activation**

This use case starts when the user has selected the category of question and the

AAselects: “Start Quiz”.

**Main flow**

1. The system provides the user with a short 5 second countdown and the the quiz begins
2. The system provides the user with their first out of ten questions
3. Each question has a ten second time limit for the user to pick and answer from four options.
4. When the user chooses an option the system will tell the if they’re correct or incorrect and the move onto the next question unless they have answered all ten questions
5. When all ten questions are answered the system provides the user with their score. The user can then choose to replay the quiz or exit the quiz back to the “select category” screen of the application.

**Alternate flow**

A1 : <exiting quiz mid-game>

1. The system starts the quiz

2. The user exits the quiz mid-game

3. The game provides the user with their score. The user receives no points for leaving the quiz incomplete.

4. The user is returned to the “select category” screen of the application

**Exceptional flow**

E1 : <exiting the quiz mid-game>

1. The system starts the quiz

2. The user exits the quiz mid-game

3. The game provides the user with their score. The user receives no points for leaving the quiz incomplete.

4. The user is returned to the “select category” screen of the application

**Termination**

The user exits the quiz after the quiz or during the quiz.

**Post condition**

The system returns to the “select category” screen of the application.

### Requirement 4: <Provide user with a score based on their performance>

#### Description & Priority

This requirement is needed for the user to receive a score on the quiz so they can see how well they did.

#### Use Case

FRscore

**Scope**

The scope of this use case is to provide the user with a score after taking a quiz on the application

**Description**

This use case describes how the system provides the user with points and also

how the point system works in our game

**Flow Description**

**Precondition**

The user has completed a quiz.

**Activation**

This use case starts when the user exits the quiz mid-game or exits the game

after completion.

**Main flow**

1. The system provides the user with 10 points for a correct answer plus one point for every second spare they had left to answer. E.g the user answers a question correctly in 7 seconds. The user receives 13 points.
2. The user receives 0 points for an incorrect answer regardless on how fast they answered the question.
3. The user also gets 20 points for completing the quiz
4. The points are totalled and the system displays the users total score once they have answered all ten questions.
5. The users score is saved in their profile.
6. The system will return the user to the “Select Category” screen of the application

**Alternate flow**

A1 : <incomplete score>

1. If the user leaves the quiz mid-game then the user will only receive points for the questions they have answered but will receive no completion points.
2. The users score will be saved under their profile
3. The system will return the user to the “Select Category” screen of the application

**Exceptional flow**

E1 : <incomplete score>

1. If the user leaves the quiz mid-game then the user will only receive points for the questions they have answered but will receive no completion points.
2. The users score will be saved under their profile
3. The system will return the user to the “Select Category” screen of the application

**Termination**

The user exits the quiz after they are provided with their score

**Post condition**

The system returns the user to the “Select Category” screen of the application.

### Requirement 5: <Allow user to view their previous scores>

#### Description & Priority

This is a requirement for our application since we want users to be able to monitor their progress by comparing their scores from previous quizzes.

#### Use Case

FRprog

**Scope**

The scope of this use case is to have a feature in the application where users

can look at their progress.

**Description**

This use case describes the interaction between the user of the application and

the “progress” feature of the application.

**Flow Description**

**Precondition**

The user selects their profile and then selects the “My progress” option in their

profile.

**Activation**

This use case starts when the user selects “My progress” in their profile.

**Main flow**

1. The user enters the “My progress” section in their profile
2. The user can view their total points as well as their results from previous quizzes.
3. The system provides the user with a stat pentagon which displays their average score in each of the games question categories. This stat pentagon will show the user which area they are strongest/weakest in.
4. The user can also view a chart of their scores to see if their results are improving and to see if they are making progress.

**Alternate flow**

A1 : <Not viewing>

1. The user enters the “My progress” section in their profile
2. They can leave this section by selecting the back option
3. Once this option is selected they are returned to their profile.

**Exceptional flow**

E1 : <Not viewing>

1. The user enters the “My progress” section in their profile
2. They can leave this section by selecting the back option
3. Once this option is selected they are returned to their profile.

**Termination**

The user exits the The user enters the “My progress” by selecting the back

button

**Post condition**

The system returns the user to their profile.

### Requirement 6: <Allow user to buy features/extra categories of question>

#### Description & Priority

This is a requirement for our application since wanted to have a feature where users can pay for extra features in the application to make it monetizable.

#### Use Case

Each requirement should be uniquely identified with a sequence number or a

meaningful tag of some kind.

**Scope**

The scope of this use case is to give users the ability to pay for extra

features/categories of question in our application.

**Description**

This use case describes the interaction between the user and the “buy extras”

option in their profile.

**Flow Description**

**Precondition**

The user is in their profile

**Activation**

This use case starts when the user selects the “buy extras” option in their profile.

**Main flow**

1. The system presents the user with the options to buy two extra categories of question.
2. The user selects the option and selects “pay now”.
3. The user enters their card details and purchases the extra categories of question.
4. The extra categories of question will be unlocked and will appear in the “Select category” page of the application.

**Alternate flow**

A1 : <Pay with points>

1. The system presents the user with the options to buy two extra categories of question.
2. The user selects the option “pay with points”.
3. The user can buy the extra categories of question if they have at least 500 in game points.

**Exceptional flow**

E1 : <Pay with points>

1. The system presents the user with the options to buy two extra categories of question.
2. The user selects the option “pay with points”.
3. The user can buy the extra categories of question if they have at least 500 in game points.

**Termination**

The user leaves “buy extras” option by selecting the back button..

**Post condition**

The system returns the user to their profile. The paid for categories of question

are now available in the “select categories” section of the application.

## Non-Functional Requirements

### Performance/Response time requirement

The performance/response time is a very important part of our application. The user will be put under pressure for time in the quiz section of the app. This means that our app must have as low of a response time as possible. A slow response time may even make the user think their click was not registered at all and click again meaning that the user unintentionally clicks twice potentially even failing the game.

When the user first clicks then app icon the first in app screen should be shown within a standard 1 to 2 seconds which will vary from device to device. However once the app opens and the game is played the response time should be as low as possible for the reasons outlined above the goal would be around 500 ms or less.

We have tried to test the response time of the current games on the market (while in game, not load speeds) however the results were very inconsistent and too difficult to measure correctly. The first app we tested for loading time to the home screen was NeuroNation by NeuroNation (Play.google.com, 2019), this app on average took 9 seconds to get to the homescreen. The next app was Brain Training by App holdings (Play.google.com, 2019) which took on average 7 seconds to ge to the homescreen. The last app tested was Lumosity by Lumos labs (Play.google.com, 2019),inc which required an average of 2 seconds to get to the home screen. All of the apps displayed their logo nearly instantly after pressing the app in order to launch it. Based of the results achieved by the competition we would like to target below 15 seconds to get to the homescreen. All tests were done with the Samsung galaxy S7 with high performance mode on.

### Availability requirement

The availability of most android games is typically quite high however as the development of the game continues the potential for bugs also exponentially increases and as more and more total time is spent playing the game more bugs can be found which may be minor or major which could reduce the games availability however the aim will be to reach an availability of around 95%. This app could reach a wide range of audience through the app store.

### Recover requirement

One of the key features of the application is the ability for users to check their stats and progression in the app in order to keep the users motivated and engaged in playing the game as a result it is very important to have a strong recovery process in place as it would not be tolerated by players if as a result of a fault/break all of their progress was reset. Furthermore the application will include paid features such as extra categories of questions which means that a strong emphasis needs to be placed on recovering and saving the users account details so that they don’t lose their progress or the features they have purchased.

### Security requirement

#### **3.2.**4**.1 Confidentiality:**

Due to our application requiring users to make an account with the application, we must ensure that any personal data collected from the user is protected against theft or disclosure of the data to unauthorised sources. We need to use methods such as encryption or masking etc to ensure that data is secure at any point. In accordance with General Data Protection Regulation we will perform an audit (DPIA) of our procedures and measure how they will affect the privacy rights of the individuals whose data we store.The DPIA will achieve three things:  
  
1. Compliance with applicable legal, regulatory and policy requirements regarding privacy  
  
2. Determine risks and effects  
  
3. Evaluate protections and alternative processes to mitigate potential privacy risks

(Grenacher, 2019)

We will also respect the right to be forgotten and only hold on to users data and keep it in our system so long as they are still using our services. If data is not needed for any intended purpose or duration it should be forgotten.

#### 3.2.4.2 Authentication:

These requirements are all about ensuring legitimacy and validity of identity. Users will be required to enter a password each time they login to the system. Our application will complete a password check. If the password is incorrectly entered the user will be alerted. A request for password change will consist of the user entering the email address they signed up with and they will be sent an email containing a link to the password change process.

#### 3.2.4.3 Error Management:

In our application, we will design the system to alert the user of any error that does occur. The system will manage errors in a format which will save any in app process and display appropriate information to the user detailing exactly which error has occurred. The information displayed will be brief in all scenarios and will not disclose any of the systems architecture or design to the user.

### Reliability requirement

At the moment the first step is to have the application connected to a simple relational database, however there are plans to at some point in the future go to cloud base database which would be very beneficial as automatic back ups would be included, There are some potential provider he have looked at such as Azure and Amazon Web Services.

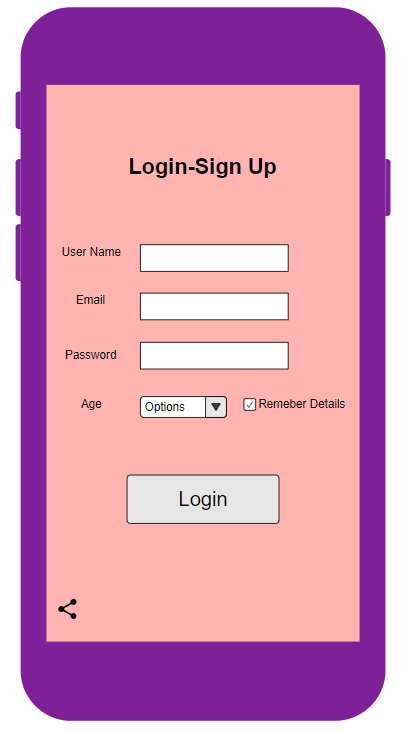
### Extendibility requirement

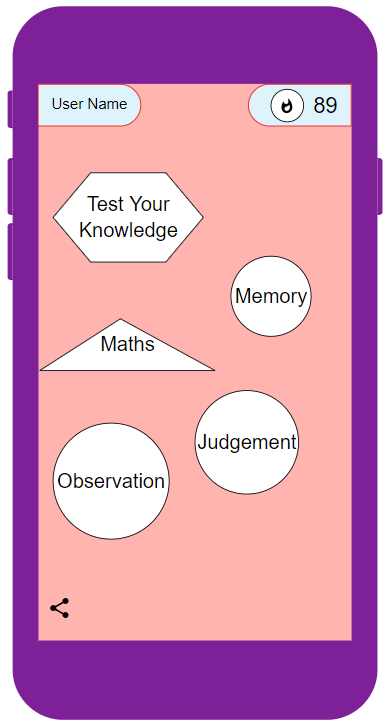
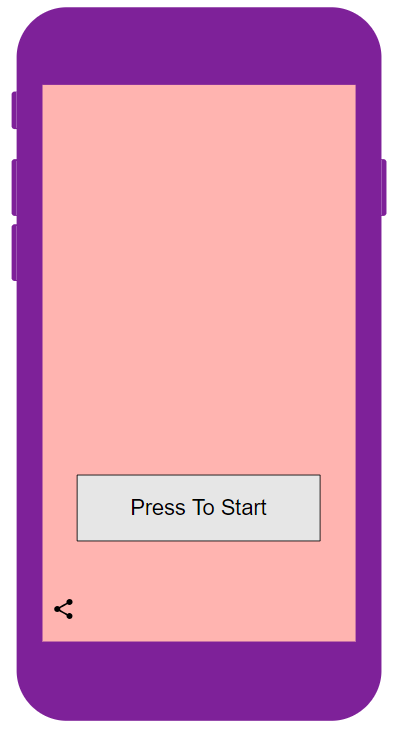
In the case of this application’s extendibility is a very important aspect. The key way in which this application will make money is via the users buying extra categories in the form of DLC. This means that the application needs to be easily extendable in the form of extra sections being continuously added to the application. The extra categories in the form of paid DLC would be along the lines of for example a geography section.

### Reusability requirement

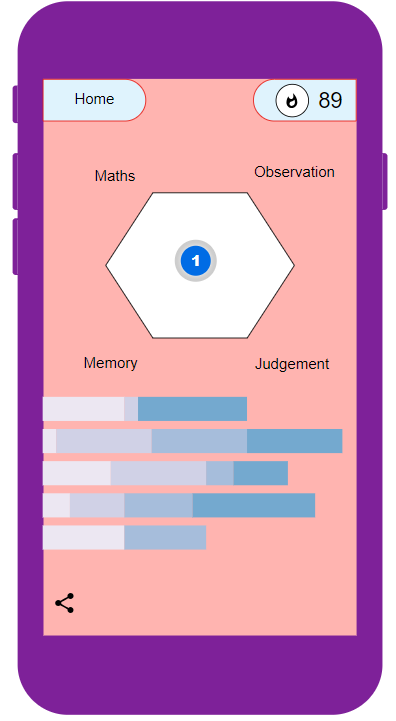
Reusability is an important factor for our project as this is one of the ways in which our app will aspire to be better than other apps on the market. After the market research it became clear what were the main drawbacks of the majority of the apps on the market, I have found that most apps on the market lack reusability. The current apps lack in reusability is due to their very linear categories, each question was always just a harder version of the previous question making each category feeling very repetitive and boring to play after not a lot of time on each section making the user very disinterested and unmotivated to go back to the game making most of the games reusability very low.

# 4 GUI

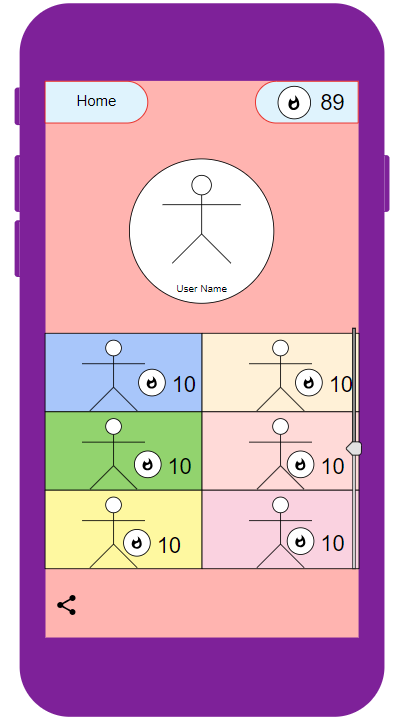
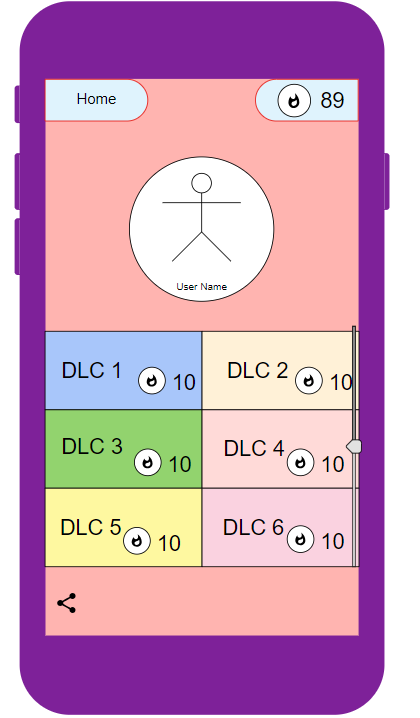
When creating the mockup, the team thought that it would be best to have the login page as the first screen that comes up when loading the application for the first time. This way the user can make an account straight away, so their details and progress are being saved and kept safe. Once the user has entered all the details on the login page and pressed the button to login they will be met by a screen where they can press a start button and go on to the applications homepage. 



Once the user is on the homepage of the application from here, they will be able to access all the other features. The features that can be accessed here are the categories which contain games related to that category, you can access your personal profile from the main page by pressing on your avatar found in top left corner of the screen. The user will be able to check their progress in the “My Profile” section, this section will show them statistic on how their performing in that specific category, how many levels they have completed and check their scoreboard or league score. Under the “my profile” section the user will also be able to go to the store where they can check out the extra content and characters that they can buy for in game currency or real money.

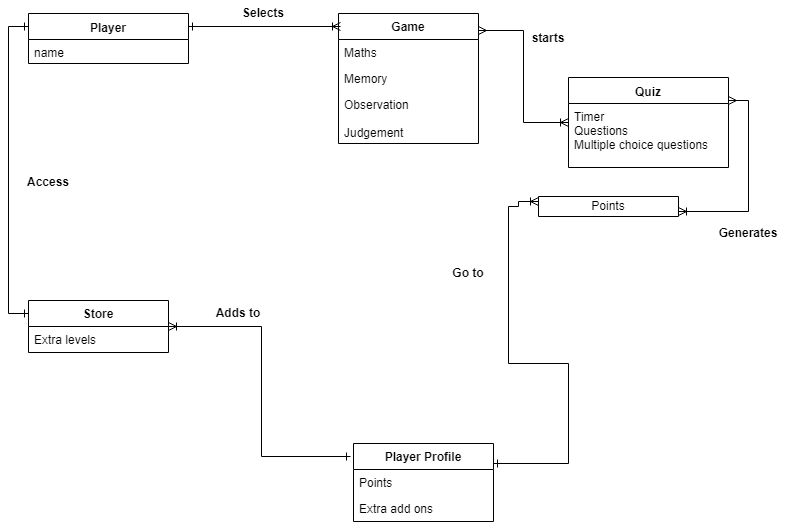


Under my progress section the users statistics can be visualised. The first graph will show how much skill the user has in any of the categories, underneath can be found your progress within the category by telling you how many of the questions have been answered correctly.



This design has been chosen for our game to help engage kids, young adults and parents. Kids will be able to get started with playing the game as soon as they login, due to the easy design of the game users don't have to go and look around for where things are . Depending of what the user is in the mood to play he can easily just choose a category straight from our home page, there are five four distinct ones such as maths, observation, judgement and memory. The fifth choice it's a mixture of all the above categories where the user can test their overall knowledge instead of a specific kind.Besides the different levels on the main page the user can visualise their score which is part of the tournament system that we will be implementing in the future.

# 5 System Architecture



# 6. System Evolution

There are many areas for the system to develop and evolve in our application. If the application gets more users, then we will need a larger database to store the users login details.

The choose category/home page can be evolved with the addition of more categories of question as well as giving users the option to play their most played category of question right away on the home screen. We plan on increasing the pool of questions that will be used in the different categories of quizzes in the application so that there is more variety each time the user takes a quiz and this will also help to avoid repeated questions. The application could also be designed to be available on the ios store as well as the play store.

The point and time system can be evolved to be altered to provide the users with less time to answer questions in the higher leagues of question. The leagues of question that the user is in is based on their point total which can be found in their profile. We want to have this feature since it will keep the users wanting to reach the higher leagues of questions with increased difficulty but with more points awarded for correct answers. We also want to add to the amount of features that can be paid for in the game.

We also hope that with the development of the application that the teams skill level will grow. With a greater skill level, the quality of design and gaming will improve vastly.

**7 Bibliography**

**Sherwin, K. (2019). *Children’s UX: Usability Issues in Designing for Young People*. [online] Nielsen Norman Group. Available at: https://www.nngroup.com/articles/childrens-websites-usability-issues/ [Accessed 15 Feb. 2019].**

**Grenacher, M. (2019). *GDPR, The Checklist For Compliance*. [online] Forbes.com. Available at: https://www.forbes.com/sites/forbestechcouncil/2018/06/04/gdpr-the-checklist-for-compliance/#1ec9753d5bec [Accessed 15 Feb. 2019].**

**McGowan, S. (2019). *What Can Children Teach Us About UX Design? - Usability Geek*. [online] Usability Geek. Available at: https://usabilitygeek.com/what-can-children-teach-us-about-ux-design/ [Accessed 15 Feb. 2019].**

**Play.google.com. (2019). [online] Available at: https://play.google.com/store/apps/details?id=air.nn.mobile.app.main&hl=en\_IE [Accessed 17 Feb. 2019].**

**Play.google.com. (2019). [online] Available at: https://play.google.com/store/apps/details?id=com.raghu.braingame&hl=en\_US [Accessed 17 Feb. 2019].**

**Play.google.com. (2019). [online] Available at: https://play.google.com/store/apps/details?id=com.lumoslabs.lumosity&hl=en\_IE [Accessed 17 Feb. 2019].**