QUIZ-GO

Who am I?

Yacquub Adan

BSc Computer Science

Aspiring Developer

Why a mobile quiz game application?

86% OF PEOPLE AGED 12-18 REGULARLY USE A MOBILE DEVICE

EDUCATIONAL QUIZ GAMES HELP STUDENTS ENGAGE BETTER IN LEARNING

Project Aim

This project aims to create an IOS mobile game that test user's knowledge on various topics in a form of multiple-choice questions

Project Objectives

- To investigate and analyse system requirements
- To research related IOS applications as a guide to designing and developing the quiz game.
- To follow the development life cycle throughout the project. (waterfall model)
- To design an interface following the IOS design principles (aesthetic integrity, consistency)
- To design the layout of the application (prototype)
- To develop the mobile application using software and swift language
- To execute a test plan and test the functionality of the application
- Maintenance, to make change of any detail in order to satisfy requirements

01

Flows in ordered stages.

02

Requirements stage is in depth

03

Focusses on planning and documentation

04

Works for small projects

Waterfall Model

Stages

Requirements – Project Scope, Aims and Objectives, Documentation, Functional requirements

Design – Design Diagrams, UI designs

Implementation – Develop software

Testing – Find errors/bugs

Maintenance – Continued support, debugging, updating.

Pros and Cons (Waterfall):

Advantages:

- Clear structure – Compared to other methodologies which are very complicated in the sense that their stages are many and can overlap, the waterfall model is very simple to understand and navigate.
- Sequential – Each stage must be completed before moving onto the next. You do not have to go back to a stage complicating things.
- Planning – All the planning and setting requirements is done during the requirements phase making it easy to understand.
- Determines the end goal early - Committing to an end goal from the beginning of the phase allows users to have a clear understanding of what must be done in order to create the software
- Information is passed on well – waterfall prioritizes accessibility towards information
- Works well with small projects (lucidchart, 2017).

Disadvantages:

- Changes become difficult – Since waterfall is an incremental step by step approach. It leaves no room for unexpected changes to specific details of the project maybe certain ideas become outdated but since you cannot work backwards with this approach it becomes hard to do.
- Risk – There can be a huge risk to this methodology for the exact reason.
- This methodology does not suit certain projects that are very complex and require constant change.
- Focuses little on the end user – If the requirements are completely met from the first stage, it doesn't matter if the user isn't completely satisfied with it.
- Delays testing after completion – leaves little room to test which can be risky.
- Documentations can be extremely long
- Not suited to huge projects (lucidchart, 2017).

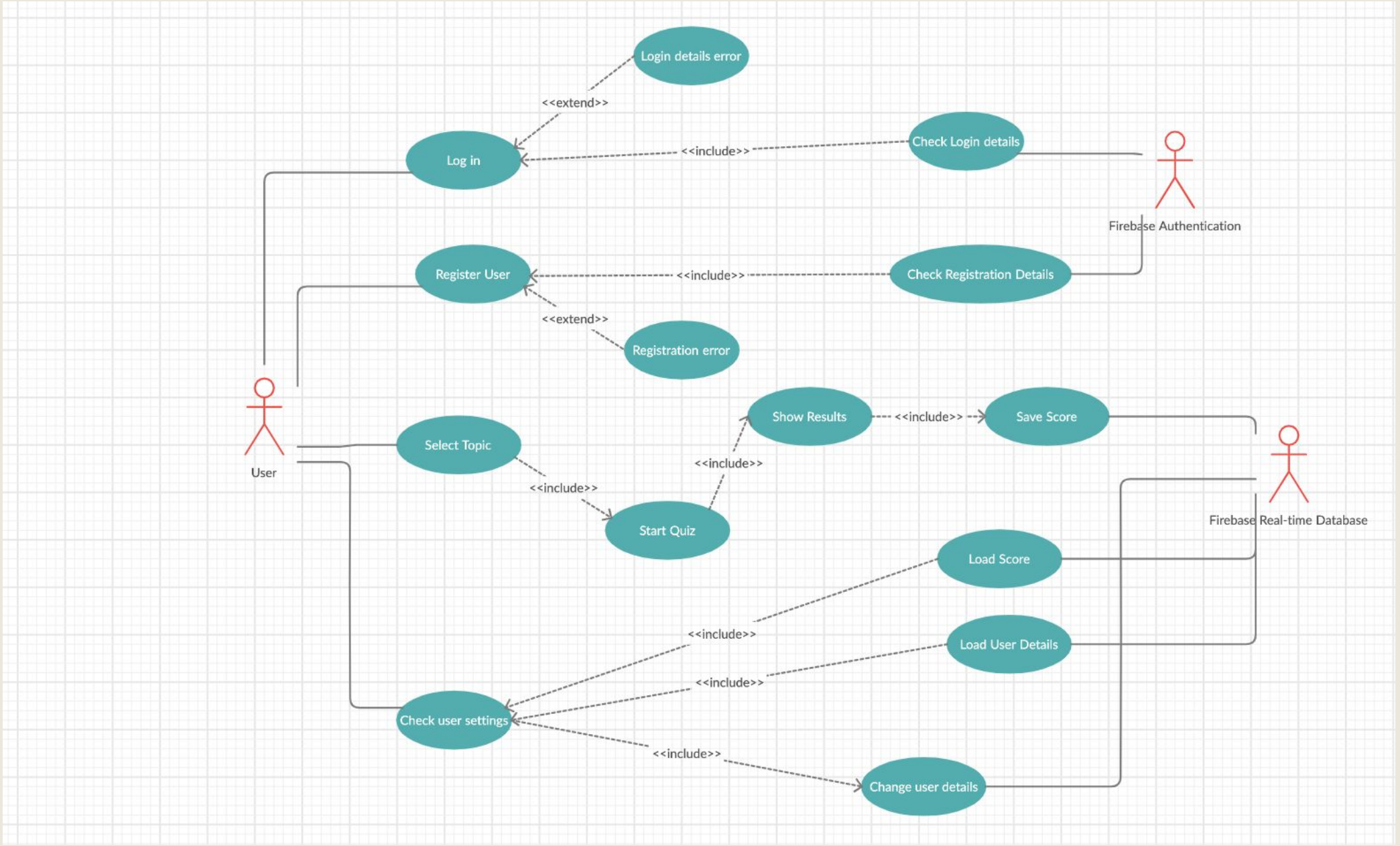
Firebase Real Time Database

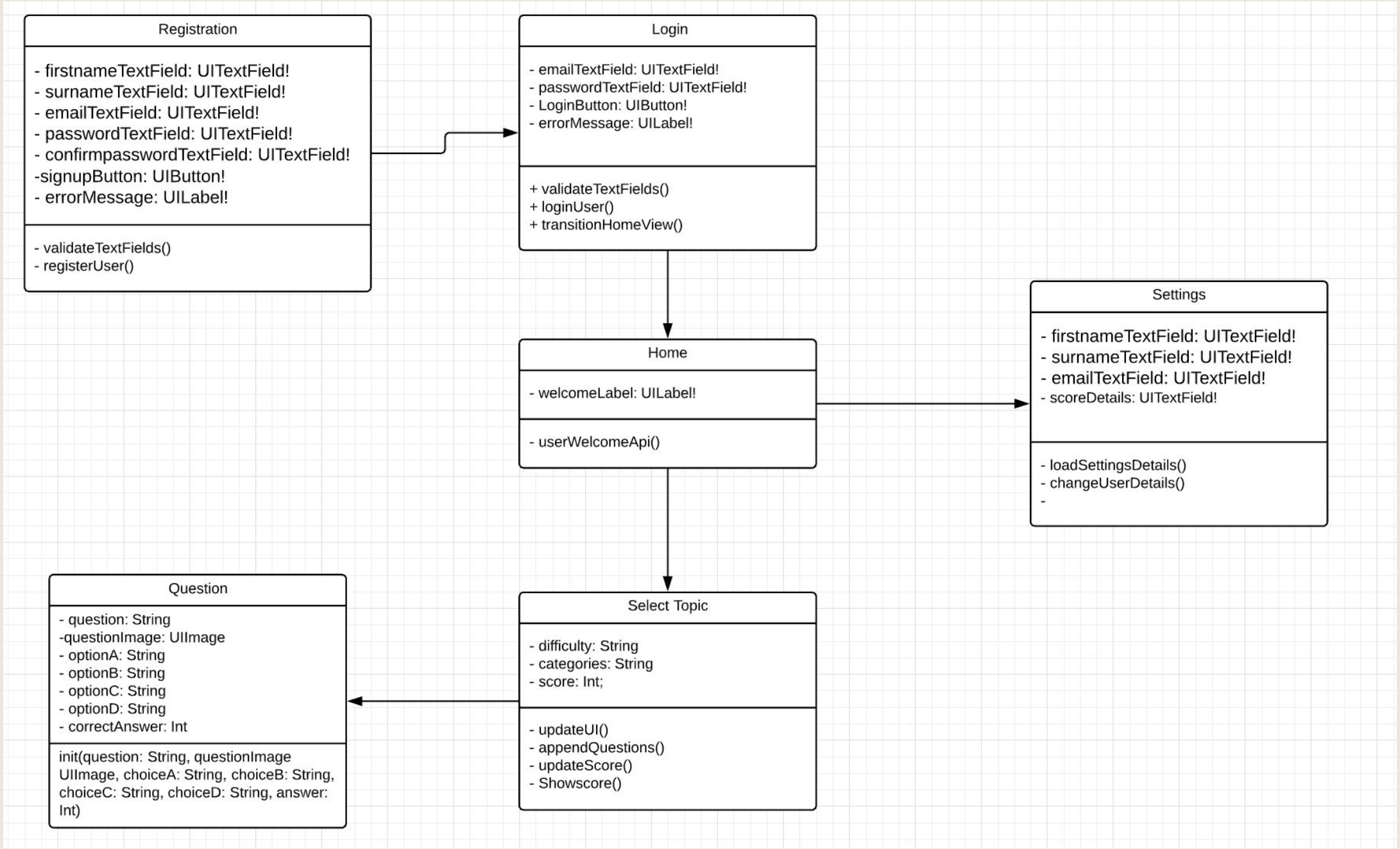
Firebase Authentication

Firebase

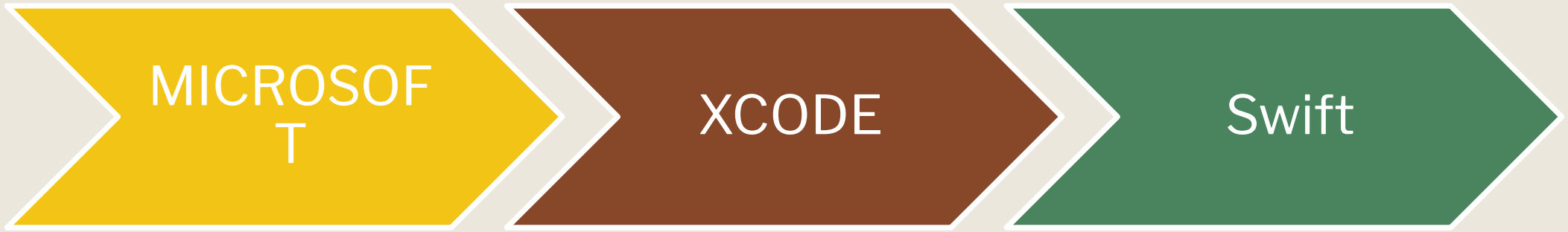
The Firebase Realtime Database is a NoSQL Database which has a lot of optimizations and features compared with most of relational databases. ... Firebase is a Database stored as JSON objects, which is easier to use than some SQL databases for the way to handle the data like a tree

Firebase Authentication supports authentication using passwords, phone numbers, popular federated identity providers like Google, Facebook and Twitter, and more.


Login details error
<<extend>>
Log in
<<include>>
Check Login details
Firebase Authentication
Register User
<<include>>
Check Registration Details
<<extend>>
Registration error
User
Show Results
<<include>>
Save Score
Select Topic
<<include>>
<<include>>
Start Quiz
Firebase Real-time Database
Load Score
<<include>>
Load User Details
<<include>>
Check user settings
<<include>>
Change user details


Registration
- firstnameTextField: UITextField!
- surnameTextField: UITextField!
- emailTextField: UITextField!
- passwordTextField: UITextField!
- confirmpasswordTextField: UITextField!
-signupButton: UIButton!
- errorMessage: UILabel!
- validateTextFields()
- registerUser()
Login
- emailTextField: UITextField!
- passwordTextField: UITextField!
- LoginButton: UIButton!
- errorMessage: UILabel!
+ validateTextFields()
+ loginUser()
+ transitionHomeView()
Home
- welcomeLabel: UILabel!
- userWelcomeApi()
Settings
- firstnameTextField: UITextField!
- surnameTextField: UITextField!
- emailTextField: UITextField!
- scoreDetails: UITextField!
- loadSettingsDetails()
- changeUserDetails()
-
Question
- question: String
-questionImage: UIImage
- optionA: String
- optionB: String
- optionC: String
- optionD: String
- correctAnswer: Int
init(question: String, questionImage UIImage, choiceA: String, choiceB: String, choiceC: String, choiceD: String, answer: Int)
Select Topic
- difficulty: String
- categories: String
- score: Int;
- updateUI()
- appendQuestions()
- updateScore()
- Showscore()

Software and Programming Languages



APPLICATION DEMO LIVE

No	Test Case	Expected Result	Actual Result
1	Loading page	While the application is loading the loading, page appears with the logo in the middle and white background	Pass
2	Login	User types in his email and password and logs in successfully	Pass
3	“Don’t have an account? Sign up” button	When clicked it should lead to the registration view for users to sign in.	
4	Registration field validation	If users do not type in anything in a field, then the error label will instruct users to fill them in	Pass
5	Registration Email validation	If users type in the wrong email address combination e.g. with the wrong combination, then the error label will read “Invalid Email Address”	Pass
6	Registration Password Validation	If the user types a password shorter than 8 characters without any special characters or numbers then the error label will read “please make sure that password is 8 characters long, includes a special character and a number”	Pass
7	Registration Confirm Password Validation	If the confirm password text field doesn’t match the password text field, then the label will state “Make sure that passwords are the same”	Pass

8	Login button	Considering the user has registered it should lead to home page	Pass
9	Sign up button	Once the user has successfully typed in all the fields with the right information then the database adds that user.	Pass
10	Home page play button	When click it should lead to the select topic page	Pass
11	Profile icon button (home page)	When pressed it should lead to the profile/settings page	Pass
12	Profile back button	When pressed it should lead to home page screen	Pass
13	Edit details button (in profile page)	When pressed it should lead to the edit details page	Pass
14	Edit details back button	When pressed it should lead to the profile/settings page	Pass
15	Edit details save changes button	When pressed it should save new user details if they have typed in both first name and last name fields (save to firebase)	Pass
16	Edit details Validation	If users do not fill in both fields then they will be instructed to do so via error message	Pass
17	Logout button (settings page)	When pressed it should log out the user and direct them back to the login page	Pass
18	Select view back button	Once click it should lead to the previous page which is home page	Pass
19	Select View topic	Once clicking a topic, it should present a popup that highlights the topic name and description with a back	Pass

19	Select View topic buttons	Once clicking a topic, it should present a popup that highlights the topic name and description with a back button and a start button	Pass
20	Select View topic pop-up back button	When pressed should lead back to select view	Pass
21	Select view topic play button	When pressed should enter the quiz view	Pass
22	Quiz view finished back button	When pressed should lead back into the select view and save the quiz score	Pass
23	Quiz option buttons	When pressed should lead to the next question	Pass
24	Quiz questions	Tests to see if the quiz questions are loaded, successfully incremented.	Pass
25	Updating score	Checks to see if the scores are updated after each quiz in the settings page	Pass
26	Save Details	Checks to see if detail changes are correctly saved in the real time database and in the settings page	Pass
27	Correct Answer	Checks to see if the user has pressed the correct answer that the background colour changes to green	Pass
28	Wrong Answer	Checks to see if the user has pressed the wrong button that the background colour changes to purple	Pass

21	Select view topic play button	When pressed should enter the quiz view	Pass
22	Quiz view finished back button	When pressed should lead back into the select view and save the quiz score	Pass
23	Quiz option buttons	When pressed should lead to the next question	Pass
24	Quiz questions	Tests to see if the quiz questions are loaded, successfully incremented.	Pass
25	Updating score	Checks to see if the scores are updated after each quiz in the settings page	Pass
26	Save Details	Checks to see if detail changes are correctly saved in the real time database and in the settings page	Pass
27	Correct Answer	Checks to see if the user has pressed the correct answer that the background colour changes to green	Pass
28	Wrong Answer	Checks to see if the user has pressed the wrong button that the background colour changes to purple	Pass
29	Button functionality	All the buttons must function as specified.	Pass
30	Keyboard	Users should be able to use their device keyboards to input text into the fields required	Pass
31	Censored text fields	Passwords and sensitive information should be censored when typed	Pass

How I could’ve improved the project

- Added a timer in-between the questions (more competitive)
- Added different difficulty levels
- Created a learning environment

Skills acquired

- Understanding of Firebase database and how it works
- The waterfall Model – experience with it
- New language learnt (swift)
- How to develop an IOS app