Table of Contents

[ITC303 Assignment 1: Life Cycle Objective Milestone: Vision 4](#_Toc508278030)

[Project Introduction 4](#_Toc508278031)

[Team Introduction 4](#_Toc508278032)

[What is the app called? 4](#_Toc508278033)

[What does it do? 4](#_Toc508278034)

[Who has a use for this app? 4](#_Toc508278035)

[What will the app look like? 4](#_Toc508278036)

[What features does our app provide? 4](#_Toc508278037)

[How will users interact with Let’s Quiz 5](#_Toc508278038)

[Touch buttons 5](#_Toc508278039)

[Drop Down Menus 5](#_Toc508278040)

[Keyboard input 5](#_Toc508278041)

[What scope is there for expansion? 5](#_Toc508278042)

[Will this be infringing on anyone’s intellectual property or be in breach of copyright law? 5](#_Toc508278043)

[Let’s Quiz Features and Specifics 6](#_Toc508278044)

[Project Design 7](#_Toc508278045)

[What devices and platforms is development planned for? 7](#_Toc508278046)

[What language(s) will be used for development? 7](#_Toc508278047)

[C# 7](#_Toc508278048)

[PHP 7](#_Toc508278049)

[SQL 7](#_Toc508278050)

[What development environment will be used? 7](#_Toc508278051)

[Unity 3d 7](#_Toc508278052)

[Visual Design 8](#_Toc508278053)

[Login Scene 8](#_Toc508278054)

[Main Menu Scene 8](#_Toc508278055)

[Pregame Scene 9](#_Toc508278056)

[Game Scene 9](#_Toc508278057)

[Post Game Scene 10](#_Toc508278058)

[Use Case Realization 11](#_Toc508278059)

[Use-Case 1: Start the application Scope: End Goal 11](#_Toc508278060)

[1.1 High Level Description 11](#_Toc508278061)

[1.2 Trigger 11](#_Toc508278062)

[1.3 Actors 11](#_Toc508278063)

[1.4 Stakeholders 11](#_Toc508278064)

[1.5 Related Use Cases 11](#_Toc508278065)

[1.6 Pre-conditions 12](#_Toc508278066)

[1.7 Post-conditions 12](#_Toc508278067)

[1.8 Normal Flow 12](#_Toc508278068)

[1.9 Alternate Flows 12](#_Toc508278069)

[1.10 Key Scenarios 13](#_Toc508278070)

[1.11 Other Quality Requirements 15](#_Toc508278071)

[Use-Case 2: Start New Game: End Goal 16](#_Toc508278072)

[2.1 High Level Description 16](#_Toc508278073)

[2.2 Trigger 16](#_Toc508278074)

[2.3 Actors 16](#_Toc508278075)

[2.4 Stakeholders 16](#_Toc508278076)

[2.5 Related Use Cases 17](#_Toc508278077)

[2.6 Pre-conditions 17](#_Toc508278078)

[2.7 Post Conditions 17](#_Toc508278079)

[2.8 Normal Flow 17](#_Toc508278080)

[2.9 Alternate Flows 18](#_Toc508278081)

[2.10 Exception Flows 18](#_Toc508278082)

[2.11 Key Scenarios 19](#_Toc508278083)

[2.12 Other Quality Requirements 20](#_Toc508278084)

[Use-Case 3: Continue an existing game: End Goal 21](#_Toc508278085)

[3.1 High Level Description 21](#_Toc508278086)

[3.2 Trigger 21](#_Toc508278087)

[3.3 Actors 21](#_Toc508278088)

[3.4 Stakeholders 21](#_Toc508278089)

[3.5 Related Use Cases 22](#_Toc508278090)

[3.6 Pre-conditions 22](#_Toc508278091)

[3.7 Post Conditions 22](#_Toc508278092)

[3.8 Normal Flow 22](#_Toc508278093)

[3.9 Alternate Flows 23](#_Toc508278094)

[3.10 Exception Flows 24](#_Toc508278095)

[3.11 Key Scenarios 24](#_Toc508278096)

[3.12 Other Quality Requirements 25](#_Toc508278097)

[Use Case 4: Submit question: End Goal 25](#_Toc508278098)

[4.1 High Level Description 25](#_Toc508278099)

[4.2 Trigger 25](#_Toc508278100)

[4.3 Actors 25](#_Toc508278101)

[4.4 Stakeholders 25](#_Toc508278102)

[4.5 Related Use Cases 26](#_Toc508278103)

[4.6 Pre-conditions 26](#_Toc508278104)

[4.7 Post Conditions 26](#_Toc508278105)

[4.8 Normal Flow 26](#_Toc508278106)

[4.9 Alternate Flows 27](#_Toc508278107)

[4.10 Exception Flows 27](#_Toc508278108)

[4.11 Key Scenarios 27](#_Toc508278109)

[4.12 Other Quality Requirements 28](#_Toc508278110)

[Use Case 5: Exit the application: End Goal 28](#_Toc508278111)

# ITC303 Assignment 1: Life Cycle Objective Milestone: Vision

# Project Introduction

## Team Introduction

## What is the app called?

“Let’s Quiz”

## What does it do?

Let’s Quiz is an online mobile game where players can verse one another in a multiple-choice quiz game.

The app will allow users to participate in a quiz game made up of three rounds, in each round the user will play their turn and then the app will notify their opponent to take their turn. The app will access an online database where it will pull question data from, user will also be able to add questions to this data base. At the end of each round there will be a post game recap of the round where users can see what the correct answer was to questions they got wrong, vote on questions they like and disliked and compare their score to their opponents.

Users will be asked to log into the app the first time they open it so that their score is maintained and they can compete on a global score board. Login will be available with either Facebook, Google Play or a locally created profile.

## Who has a use for this app?

Let’s Quiz’s target audience is very broad, essentially anyone with an interest in trivia, competition and who has a smart phone can enjoy this app. One particularly nice feature of the user-maintained question list is the app itself should begin to have questions that the audience likes as the people who are playing it will be the ones add questions and voting on them.

## What will the app look like?

The app’s visual design will be quite sleek and minimalist, the specific look we are trying to avoid is cluttered and confusing. If you compare a traditional digital calculator with a TV remote it is abundantly clear that an excess of buttons and complex features do not make for a more pleasant user experience.

Due to the fun nature quiz game inherently are and because the colourful images Triva Pursuit, the worlds most popular trivia board game, brings to mind, we have elected to have a colourful and fun theme.

## What features does our app provide?

The primary feature of the app is the two player, three round, multiple choice quiz game.

Secondary features include the ability for users to add questions to the question pool, vote of their favorited questions and down vote questions they do not like. This gives a great deal of flexibility and customisation for users and hopefully ensures that the game is always changing, and the questions are always fresh.

Another secondary feature is a global score board, so users can see how good at trivia they are compared to every user of Let’s Quiz in the world. There are also push notifications, so a user will know when one of their active games has progressed and it is their turn, this feature is critical to a successful multiplayer turn based game so that users do not have to wait too long for their opponent to take their turn.

Finally, our app is compatible across both android and Apple products and allows for synchronisation with multiple social media platforms.

## How will users interact with Let’s Quiz

The application is designed with simplicity in mind. There are only three ways a user can interact with our application, touch buttons and drop down menus and keyboard input.

### Touch buttons

Touch buttons are the simplest form of input and our go to choice for how user are expected to navigate our system. It is reasonable to expect users to have familiarity with how touch buttons work so we do not have to be concerned with teaching them how to interact and operate touch buttons. Having simple buttons eliminates any chance of incorrect usage, continuing with our design goals of simplicity.

### Drop Down Menus

Where there are more options available to a user then a simple touch button can reasonably handle, like for example if the user has to select a question category we have elected to use drop down menus.

The advantage of a drop down menu is it presents the user with an array of options but limits their potential selections to those that are valid.

### Keyboard input

Where the app asks the user for input that requires creativity or personalisation for example when submitting a question, an on screen keyboard will present itself in a similar fashion to text messaging.

## What scope is there for expansion?

Currently the app is designed to work with text based questions, a future expansion may be to allow for picture based questions so for example we could show users an image of an animal and ask them to select the correct name.

Another area of expansion may be in how answers are submitted, on release Let’s Quiz presents users with 4 possible answers and the user must choose the correct one. A possible expansion might be to allow users to write their own answers or lists of answers, in this way we could ask a user to list as many countries starting the with letter C for example.

For minor tweaks and useability improvements our plan is to roll out an app in the best state possible and then listen to our users requests as to what they feel is important and beneficial and prioritise from there.

## Will this be infringing on anyone’s intellectual property or be in breach of copyright law?

No. There are many quiz games in existence and the idea of asking someone a series of questions they have to answer in a time limit is not a proprietary idea.

# Let’s Quiz Features and Specifics

|  |  |
| --- | --- |
| **Feature** | **Specifics** |
| Login | Allows user to login to the application using either their existing Facebook account, Google Play account or by creating a profile on the Let’s Quiz app itself. The user’s profile will be saved and synced to the device so that when they are playing again they are not asked to log in again. The settings page will allow a user to make changes to their login information, for example to log out or change their Let’s Quiz public username. |
| Download living question pool | On start up the application will connect to our server where it will download the latest version of the question pool, this will be stored on the device to minimise server calls. |
| Save User Details | User information will be saved to our server and collated, this information will relate to things like, how many questions the user has submitted, how many votes those questions have received, what their accumulative score is, their rank on the global score board. |
| Save game details | At the end of each round the application will save the on going game state on our sever. This allows for the opponent to login and begin playing the game and maintaining data integrity. Some details that must be saved as part of the ongoing game are, the players name, the players score, what questions were asked and in what order and what round it is up to. |
| Global score board | All users and their accumulative scores will be kept in a table which will be the global score board so users can see how they rank compared to every other Let’s Quiz user. |
| Add questions to the question pool | Allow user to add questions to the online database. |
| Alert users when the opponent in one of their ongoing games has had their turn | Push notifications to users when the other player has finished their turn in one of the users ongoing games, this will help minimise delay between player turns and hopefully keep users coming back to Let’s Quiz. |

# Project Design

## What devices and platforms is development planned for?

Let’s Quiz is planned to be released on Android and IOS, ideally through the Play Store and the App Store.

## What language(s) will be used for development?

### C#

Our primary language will be C#. C# is an exceptionally powerful language that is very widely used, particularly in game development. Our primary reason for choosing C# is everyone in the team has at least a some experience with it and it is Unity 3d’s preferred language. For that reason there is plenty of support available should we require it.

### PHP

PHP is a general-purpose scripting language that is especially suited to server-side web development, in which case PHP generally runs on a web server.

To upload and download information from our online server and SQL databases. Not all members of the team are familiar with PHP so this will provide an excellent opportunity to learn.

### SQL

SQL stands for Structured Query Language. SQL is used to communicate with a database. It is the standard language for relational database management systems. On the sever is an SQL database where all our unit information is stored, to navigate the database we will be using SQL commands.

SQL is an incredibly simple language with its commands almost completely human readable. For the purposes of this application the SQL necessary will be quite simple.

## What development environment will be used?

### Unity 3d

Unity3d is a very powerful development tool for gaming developers. Unity allows for multiplatform development so we will not have to design the application twice for Android and IOS. Unity is a very popular game development tool meaning there is plenty of support, documentation and tutorials should we need them.

As a team some of our members have had experience with Unity while for others this will be an opportunity to learn.

# Visual Design

The following outlines the planned flow of scenes from app start up to completion and while the theme may change the functional design and feel of each scene should not alter much.

## Login Scene

Here we have a simple sign in page. The user has 4 options, they can sign in using their social media account of choice, they can elect to create username specific to Let’s Quiz or they can skip the whole sign in process.

Should a user elect to skip the sign in process their username will automatically be assigned as ‘Guest’ and their ongoing score will not be recorded in the global high scoreboard. Furthermore every time they open the Let’s Quiz app they will be asked to sign in again.

The selection of any other form of sign in will be a permeant choice, only changeable through the settings page.

## Main Menu SceneC:\Users\Collin.Mckeahnie\Pictures\screen shots of quiz game\main menu.PNG

The main menu page is fairly self-explanatory. Users who have signed in for the first time will then arrive at this page, returning users who previously signed in will immediately arrive at this scene upon opening the app.

Should the app encounter any errors the user will be returned to the Main Menu. If the user presses the back button on an android device from the Main Menu they will be asked if they wish to quit.

## Pregame SceneC:\Users\Collin.Mckeahnie\Pictures\screen shots of quiz game\pregame.PNG

The Pregame Scene is designed to be the launch pad for users to start new games, return to old games and check on progress and scores of existing games. Visible from this screen shot it is possible to see the user has four games open concurrently, the top two are waiting for the user to take his turn. The two games in red are awaiting the other player to take their turn. To enter a game the user need only press anywhere on a green banner. The final option presented to the user is at the bottom of the page where they can begin a new game.

In the top right corner is a refresh button so user can force a refresh of the page.

## Game Scene



The main game of Let’s Quiz will present as a series of questions and answers like this. Once the user has made the selection they think is correct the button will light up, green if it is right and red if it is wrong. In the event of an incorrect selection the right answer will light up green.

In the top right is the user score for the round and on the left is the remaining time for the round.

## C:\Users\Collin.Mckeahnie\Pictures\screen shots of quiz game\Post Game Scene.PNGPost Game Scene

The final scene is displayed after the round has ended, the Post Game Scene. Here the user can review the questions that were asked, red means they got the answer wrong and green for correct. A user can press a question and the text will change from the question text to the correct answer text.

# Use Case Realization

## Scope: End Goal

### 1.1 High Level Description

#### 1.1.1 User end goal story:

When the user

Wants to start the application

They click the icon on their phone to open the application

So that the application opens to the login screen

#### 1.1.2 Event-response story

When the user presses the game icon the application will start.

It causes the game to load the Login Scene and check for local existing user data

By checking for a userID

So that if there is a userID the app can automatically log the user in or if there is not the app can ask the user to log in for the first time

### 1.2 Trigger

The user presses the icon for Let’s Quiz from their phones apps page or home screen.

### 1.3 Actors

#### 1.3.1 User

The user starts the use case by opening the app.

#### 1.3.2 Let’s Quiz Server

Once the user has logged in whether it was done automatically or not the application will contact the online database to add the new user or pull existing user details.

### 1.4 Stakeholders

#### 1.4.1 User

The user requires the app to open without errors and minimal delay.

#### 1.4.2 Let’s Quiz Server

The device needs to be able to access the online database to check user details.

#### 1.4.3 User’s phone

The user’s phone needs to allocate memory and give the application the correct permissions to operate as required.

#### 1.4.4 The User’s Previously Provided Social Media Account

Let’s Quiz offers users a selection of ways to log in and create lasting accounts including Facebook and Google Play. Let’s Quiz will have to interact with their APIs, maintain a reputable name so as to maintain communication with each social media server and at least one member of the Let’s Quiz design team will have to developers for each social media outlet Let’s Quiz wishes to interact with.

### Related Use Cases

None

### Pre-conditions

1.6.1 User’s phone meets the minimum requirements

1.6.2 User has Let’s Quiz installed

* + 1. User press’s the Let’s Quiz icon app

1.6.4 The user has internet connectivity

### 1.7 Post-conditions

#### 1.7.1 Minimal guarantee

Give an error message to the user so they can rectify the problem, or upon an unrecoverable error the application will safely terminate and the user can restart it.

#### 1.7.2 Success guarantee

The application will start and open the Login Scene

### 1.8 Normal Flow

|  |  |
| --- | --- |
| **Actor** | **System** |
| 1. The user presses the Let’s Quiz icon 2. User selects method to log in, either Facebook, GooglePlay or Create Profile | 1. The application will open to the Login Scene 2. Device checks local data for existing userID (none present) 3. The user is asked to log in and presented with 4 options: Facebook, GooglePlay, Create Profile or Skip 4. Application verifies login with appropriate social media platform or Let’s Quiz server 5. The application creates a userID 6. Application loads Main Menu Scene |

The use case ends.

### 1.9 Alternate Flows

#### 1.9.1 Alternate flow 1: The user has logged in before and has a userID

If at step 3 of the normal flow the application already has a userID

|  |  |
| --- | --- |
| **Actor** | **System** |
|  | 3.1 Device checks local data for existing userID (userID present)  3.2 The user is logged in using the method they selected previously  3.3 A greeting message will pop display so the user knows they are logged in  Normal flow will continue from step 8 |

#### 1.9.2 Alternate flow 2: The user chooses to ‘skip’ login

If at step 4 of the normal flow the user presses skip instead of choosing a login option

|  |  |
| --- | --- |
| **Actor** | **System** |
|  | 3.1 Attempt to connect to online server times out after 30 seconds.  3.2 An error message will pop up to inform the user of the problem 3.3 The application will skip the login process and use a guest account  Normal flow will continue from step 8 |

#### 1.9.3 Alternate flow 3: There is no internet connection or the server/social media account is unavailable

If at step 6 of the normal flow the user login cannot be verified

|  |  |
| --- | --- |
| **Actor** | **System** |
|  | 6.1 Attempt to connect to online server times out after 30 seconds.  6.2 An error message will pop up to inform the user of the problem 6.3 The application will skip the login process and use a guest account  Normal flow will continue from step 8 |

### 1.10 Key Scenarios

#### 1.10.1 Open application new user logs

1.10.1.1 The application opens the Login Scene

1.10.1.2 Application checks local data for userID

1.10.1.3 No userID is found

1.10.1.4 The user is given a choice of ways to login to Let’s Quiz

1.10.1.5 The user selects either Facebook, Google Play or Create a Profile

1.10.1.6 The application verifies the user details with either the Let’s Quiz server or the social media platform

1.10.1.7 The verification is successful and the application creates a userID

1.10.1.8 The application loads the Main Menu Scene

#### 1.10.2 The user has logged in before and has a userID

1.10.2.1 The application opens the Login Scene

1.10.2.2 Application checks local data for userID

1.10.2.3 userID is found on the device

1.10.2.4 The user is logged in using the previously saved login details

1.10.2.5 The application contacts the Let’s Quiz server for stored user details including existing open games

1.10.2.6 A personalised greeting message is displayed to the user so they know they have been logged in

1.10.2.7 The application loads the Main Menu Scene

#### 1.10.3 The user chooses to skip the login step

1.10.3.1 The application opens the Login Scene

1.10.3.2 Application checks local data for userID

1.10.3.3 No userID is found

1.10.3.4 The user is given a choice of ways to login to Let’s Quiz

1.10.3.5 The user chooses ‘Skip’ from the login options

1.10.3.6 The user name is set to ‘Guest’

1.10.3.7 No userID is created

1.10.3.8 The application loads the Main Menu Scene

#### 1.10.4 The server or social media platform is unavailable or verification otherwise fails

1.10.4.1 The application opens the Login Scene

1.10.4.2 Application checks local data for userID

1.10.4.3 No userID is found

1.10.4.4 The user is given a choice of ways to login to Let’s Quiz

1.10.4.5 The user selects either Facebook, Google Play or Create a Profile

1.10.4.6 The application verifies the user details with either the Let’s Quiz server or the social media platform

1.10.4.7 the server or social media platform is unavailable, or verification otherwise fails

1.10.4.8 The pop up message is displayed to the user saying what has happened

1.10.4.9 The user name is set to ‘Guest’

1.10.4.10 No userID is created

1.10.4.11 The application loads the Main Menu Scene

### 1.11 Other Quality Requirements

#### 1.11.1 Internet connection

To connect to the online server or social media account the application requires an active internet connection.

## Start New Game: End Goal

### 2.1 High Level Description

#### 2.1.1 User end goal story:

When the user

Wants to start a new game

They press the ‘start new game’ button in the Pregame Scene

So that the application will start a new game or join an existing game without an opponent

#### 2.1.2 Event-response story

When the user presses ‘start new game’

It causes the application to ask the server if there are any games without opponents, if there are the user will join that game as player2 if not the application will start a new game and the user will be player1

So that a new game is either started or an existing game’s round 1 is completed

### 2.2 Trigger

The user presses ‘start new game’ from the Pregame Scene

### 2.3 Actors

#### 2.3.1 User

The user presses the ‘start new game’ button.

#### 2.3.2 Let’s Quiz Server – Ongoing games table

The application will ask the server if there is an ongoing game that needs an opponent and if so add the user to the game. If there is no ongoing games then the user’s game will be added to the Ongoing Games table and listed as ‘needing an opponent’

#### 2.3.3 Let’s Quiz Server – Questions table

The application will ask the user questions and check their answer

### 2.4 Stakeholders

#### 2.4.1 User

The user requires the app to run the game as expected

#### 2.4.2 Let’s Quiz Server

The device needs to be able to access the online database to check user details and current game state.

#### 2.4.3 User’s phone

The user’s phone needs to allocate memory and give the application the correct permissions to operate as required.

#### 2.4.4 Game Opponent

The other player, whether they have joined the game yet or not, is a stake holder. The user’s score, questions the user has been asked and what answers they gave will all effect the opponent’s game.

Whether or not the game was already ongoing or not there will always be an opponent who needs to compete against the user

### 2.5 Related Use Cases

#### 2.5.1 Start application

In this use case the application is already started, the user has logged in not using a guest account. This use case is dependent on the Let’s Quiz Server being in a useable state.

### 2.6 Pre-conditions

2.6.1 The user has started the application

* + 1. The has selected Play Game from the Main Menu

2.6.3 There is an open internet connection

### 2.7 Post Conditions

#### 2.7.1 Minimal guarantee

Give an error message to the user so they can rectify the problem, or upon an unrecoverable error the application will safely terminate and the user can restart it.

#### 2.7.2 Success guarantee

The application creates a new game or joins an existing game without an opponent, the user plays the round and the use case ends in the Post Game Scene.

### 2.8 Normal Flow

The use case begins when the user presses ‘Start New Game” in the Pregame Scene

|  |  |
| --- | --- |
| Actor | System |
| 1. The user presses ‘Start New Game’ button | 1. The app will connect to the Let’s Quiz server and check if there are any ongoing games without two players 2. The server response FALSE 3. The application starts a new game setting the user to Player1 4. The application asks the user a random question from the unasked question pool and provides 4 possible answers |
| 1. The user chooses the answer they think is correct | 1. The selected answer is checked to see if it is correct (TRUE) 2. The selected answer is highlighted in green, 10 points are added to the users score 3. The round timer reaches zero 4. The application ends the round 5. The Ongoing games table is updated with the results of the game 6. The Post Game Scene is loaded |

The use case ends.

### 2.9 Alternate Flows

#### 2.9.1 No connectivity to the Let's Quiz Server

If at step 2 of the normal flow the app cannot connect to the Let’s Quiz Server

|  |  |
| --- | --- |
| **Actor** | **System** |
|  | 2.1 A popup message alerts the user to the error  2.2 The application gracefully exits to the main menu |
|  |  |
|  |  |

#### 2.9.2 The serve’s response TRUE to the question ‘is there ongoing games without two players

If at step 3 of the normal flow when asked if there are ongoing games that need an opponent the server returns TRUE

|  |  |
| --- | --- |
| **Actor** | **System** |
|  | 3.1 The app asks the server for the details of the game requiring an opponent  3.2 The application starts a new game setting the user to Player2  3.3 The user is asked questions in order from the asked questions pool and 4 possible answers are presented |
|  |  |
|  | Normal flow will continue from step 6 |

#### 2.9.3 The user selects an incorrect answer

If at step 7 the selected answer is incorrect

|  |  |
| --- | --- |
| **Actor** | **System** |
|  | 8.1 The selected answer is highlighted in red, the correct answer is highlighted in green, 5 points are subtracted from the users score |
|  |  |
|  | Normal flow will continue from step 9 |

#### 2.9.4 The user selects an incorrect answer

If at step 9 the round timer has not reached zero

|  |  |
| --- | --- |
| **Actor** | **System** |
|  |  |
|  |  |
|  | Normal flow will continue from step 5 or 3.3 depending if the user is Player1 or Player2 |

### 2.10 Exception Flows

None

### 2.11 Key Scenarios

#### 2.11.1 Start a new game

2.11.1.1 The user selects ‘Start New Game’

2.11.1.2 The application queries the server if there are any ongoing games that do not have two players

2.11.1.3 The server responds FALSE

2.11.1.4 The application starts a new game marking the user as Player1 and Player2 as null

2.11.1.5 The user is asked random questions from the remaining question pool

2.11.1.6 The user answers the questions and is given a score as they answer correctly or incorrectly

2.11.1.7 The questions are stored as asked questions, the questions are removed from the remaining question pool

2.11.1.8 The round timer runs out

2.11.1.9 The application ends the round

2.11.1.10 The Post Game Scene is loaded

2.11.1.11 The game data is uploaded to the server

#### 2.11.2 Join an existing game that needs an opponent

2.11.2.1 The user selects ‘Start New Game’

2.11.2.2 The application queries the server if there are any ongoing games that do not have two players

2.11.2.3 The server responds TRUE

2.11.2.4 The server sends the game data for the ongoing game needing an opponent

2.11.2.5 The application starts a new quiz

2.11.2.6 The user is asked the questions saved in the game data as asked questions in order until they run out at which time the user will be asked random questions from the remaining questions pool

2.11.2.7 The user answers the questions and is given a score as they answer correctly or incorrectly

2.11.2.8 The round timer runs out

2.11.2.9 The application ends the round

2.11.2.10 The Post Game Scene is loaded

* + - 1. The game data on the server is updated

### 2.12 Other Quality Requirements

#### 2.12.1 Internet connection

The application needs to be able to communicate with the Let’s Play Server

## Continue an existing game: End Goal

### 3.1 High Level Description

#### 3.1.1 User end goal story:

When the user

Wants to take their turn on a game they started previously

They press on the game description in the Pregame Scene

So that the application will open the game in the correct state

#### 3.1.2 Event-response story

When the user presses on the game description in the Pregame Scene

It causes the application to ask the server for the current game data for the game number corresponding to the button they pressed.

The game data is passed to the application

So that a new round can be started for the player where questions are asked from either the remaining question pool or the started or an existing game’s round 1 is completed

### 3.2 Trigger

The user presses on the game description in the Pregame Scene

### 3.3 Actors

#### 3.3.1 User

The user presses on the game description in the Pregame Scene, then plays the round

#### 3.3.2 Let’s Quiz Server – Ongoing games table

The application will ask the server for the game data relating to the game description that was pressed. The game data will include the question pool to draw questions from and information on how the users opponent did during the round.

### 3.4 Stakeholders

#### 3.4.1 User

The user requires the app to run the game as expected and in the event of an error gracefully back out

#### 3.4.2 Let’s Quiz Server

The device needs to be able to access the online database to check user details and game data.

#### 3.4.3 User’s phone

The user’s phone needs to allocate memory and give the application the correct permissions to operate as required.

#### 3.4.4 Game Opponent

The other player, is a stake holder. The user’s score, questions the user has been asked and what answers they gave will all affect the opponent’s game.

### 3.5 Related Use Cases

#### 3.5.1 Start application

In this use case the application is already started, the user has logged in not using a guest account. This use case is dependent on the Let’s Quiz Server being in a useable state.

### 3.6 Pre-conditions

3.6.1 The user has started the application

* + 1. The user has logged in not using a guest account

3.6.3 The has selected Play Game from the Main Menu

3.6.4 There is an open internet connection

### 3.7 Post Conditions

#### 3.7.1 Minimal guarantee

Give an error message to the user so they can rectify the problem, or upon an unrecoverable error the application will safely terminate and the user can restart it.

#### 3.7.2 Success guarantee

The application retrieves the relevant game data from the server and the user plays the appropriate round. The use case ends in the Post Game Scene.

### 3.8 Normal Flow

The use case begins when the user presses on the game description for the game they wish to return to in the Pregame Scene

|  |  |
| --- | --- |
| Actor | System |
| 1. The user presses on the game description they wish to return to | 1. The app will connect to the Let’s Quiz server and retrieves the game data attached to the game number they just pressed 2. A new round is started 3. The user is Player1 4. The application asks the user a random question from the unasked question pool and provides 4 possible answers |
| 1. The user chooses the answer they think is correct | 1. The selected answer is checked to see if it is correct (TRUE) 2. The selected answer is highlighted in green, 10 points are added to the users score 3. The round timer reaches zero 4. The application ends the round 5. The Ongoing games table is updated with the results of the game 6. The Post Game Scene is loaded |

The use case ends.

### 3.9 Alternate Flows

#### 3.9.1 No connectivity to the Let's Quiz Server

If at step 1 of the normal flow it is not the users turn

|  |  |
| --- | --- |
| **Actor** | **System** |
|  | 1.1 The game description will be in red and the button it represents will not be intractable. Nothing will happen |
|  |  |
|  | Normal Flow will continue from step 1 |

#### 3.9.2 No connectivity to the Let's Quiz Server

If at step 2 of the normal flow the app cannot connect to the Let’s Quiz Server

|  |  |
| --- | --- |
| **Actor** | **System** |
|  | 2.1 A popup message alerts the user to the error  2.2 The application gracefully exits to the main menu |
|  |  |
|  |  |

#### 3.9.3 The user is Player2

If at step 4 of the normal flow the user is Player2

|  |  |
| --- | --- |
| **Actor** | **System** |
|  | 4.1 User is Player 2  4.2 The user is asked questions in order from the asked questions pool and 4 possible answers are presented |
|  |  |
|  | Normal flow will continue from step 6 |

#### 3.9.4 The user selects an incorrect answer

If at step 7 the selected answer is incorrect

|  |  |
| --- | --- |
| **Actor** | **System** |
|  | 8.1 The selected answer is highlighted in red, the correct answer is highlighted in green, 5 points are subtracted from the users score |
|  |  |
|  | Normal flow will continue from step 9 |

#### 3.9.5 The user selects an incorrect answer

If at step 9 the round timer has not reached zero

|  |  |
| --- | --- |
| **Actor** | **System** |
|  |  |
|  |  |
|  | Normal flow will continue from step 5 or 3.3 depending if the user is Player1 or Player2 |

### 3.10 Exception Flows

None

### 3.11 Key Scenarios

#### 3.11.1 Return to existing game as Player1

3.11.1.1 The user presses on the game description they want to return to

3.11.1.2 The application queries the server for game data related to the game description the user pressed

3.11.1.3 The application opens a new round

3.11.1.4 The user is asked random questions from the remaining question pool

3.11.1.5 The user answers the questions and is given a score as they answer correctly or incorrectly

3.11.1.6 The questions are stored as asked questions; the questions are removed from the remaining question pool

3.11.1.7 The round timer runs out

3.11.1.8 The application ends the round

3.11.1.11 The Post Game Scene is loaded

3.11.1.11 The game data is uploaded to the server

3.11.1.12 Push notification is sent to the other player

#### 2.11.2 Join an existing game that needs an opponent

3.11.2.1 The user presses on the game description they want to return to

3.11.2.2 The application queries the server for game data related to the game description the user pressed

3.11.2.3 The application opens a new round

3.11.2.4 The user is asked the questions saved in the game data as asked questions in order until they run out at which time the user will be asked random questions from the remaining questions pool

3.11.2.5 The user answers the questions and is given a score as they answer correctly or incorrectly

3.11.2.6 The round timer runs out

* + - 1. The application ends the round

3.11.2.8 The Post Game Scene is loaded

3.11.2.9 The game data on the server is updated

### 3.12 Other Quality Requirements

#### 3.12.1 Internet connection

The application needs to be able to communicate with the Let’s Play Server

## Use Case 4: Submit question: End Goal

### 4.1 High Level Description

#### 4.1.1 User end goal story:

When the user

Wants to add a question to the Let’s Quiz question pool

They select Submit Question from the Main Menu

So that the application will open the Submit Question Scene

#### 4.1.2 Event-response story

When the user selects Submit Question

It causes the application to load the Submit Question scene

The user then adds data in the order asked

So that the serialised data can then be sent to the Let’s Quiz server to be added to the Questions SQL table

### 4.2 Trigger

The user presses Submit Question from the Main Menu

### 4.3 Actors

#### 4.3.1 User

The user presses Submit Question and then follows the prompts to add their question to the server

#### 4.3.2 Let’s Quiz Server – Questions table

The application will append the question data to the SQL table containing the question pool for Let’s Quiz

### 4.4 Stakeholders

#### 4.4.1 User

The user is trying to add their own question to the question pool for future Let’s Quiz games

#### 4.4.2 Let’s Quiz Server

The device needs to be able to access the server to add the question to the Question table

#### 4.4.3 User’s phone

The user’s phone needs to allocate memory and give the application the correct permissions to operate as required.

#### 4.4.4 All Let’s Quiz players

Due to the communal nature of all the questions every user is a stakeholder in the quality of the questions being submitted.

### 4.5 Related Use Cases

#### 4.5.1 Start application

In this use case the application is already started, the user has logged in not using a guest account. This use case is dependent on the Let’s Quiz Server being in a useable state.

### 4.6 Pre-conditions

* + 1. The user has started the application
    2. The user has logged in not using a guest account

4.6.3 There is an open internet connection

### 4.7 Post Conditions

#### 4.7.1 Minimal guarantee

Give an error message to the user so they can rectify the problem, or upon an unrecoverable error the application will safely terminate and the user can restart it.

#### 4.7.2 Success guarantee

The user enters a new question that is submitted to the Question Table on the Let’s Quiz server.

### 4.8 Normal Flow

The use case begins when the user presses on the game description for the game they wish to return to in the Pregame Scene

|  |  |
| --- | --- |
| Actor | System |
| 1. The user presses Submit New Question button from the Main Menu 2. The user enters the Question text 3. The user presses Next button | 1. The Submit Question scene is loaded 2. A brief check of the question is made to ensure it is an acceptable length, ends with a question mark and starts with a capital letter 3. The user is asked to enter the correct answer |
| 1. The user enter the correct answer 2. The user presses the Next button 3. The user enters an incorrect answer 4. The user presses the Next button 5. The user enters an incorrect answer 6. The user presses the Next button 7. The user enters an incorrect answer 8. The user presses the Finish button | 1. The user is asked to enter an incorrect answer 2. The user is asked to enter an incorrect answer 3. The user is asked to enter an incorrect answer 4. The serialised data is added to the Questions Table on the Let’s Play server via a PHP script also stored on the server 5. A success message is presented to the user 6. The Main Menu scene is loaded |

The use case ends.

### 4.9 Alternate Flows

#### 4.9.1 The question the user enters is too short or too long

If at step 4 of the normal flow it is not the users turn

|  |  |
| --- | --- |
| **Actor** | **System** |
|  | 4.1 The question is considered too small or too long  4.2 The user is presented with a pop up message explaining the error  4.3 The scene does not advance any further |
|  |  |
|  | Normal Flow will continue from step 3 |

#### 4.9.2 No connectivity to the Let's Quiz Server

If at step 18 of the normal flow the app cannot connect to the Let’s Quiz Server

|  |  |
| --- | --- |
| **Actor** | **System** |
|  | 18.1 A popup message alerts the user to the error  18.2 The serialised data is stored locally and sent to the server the next time the application has internet connectivity |
|  |  |
|  | Normal Flow continues from step 20 |

### 4.10 Exception Flows

None

### 4.11 Key Scenarios

#### 4.11.1 Add question to the Question Table on the Let’s Play Server

4.11.1.1 User selects ‘Submit Question’ from the Main Menu

4.11.1.2 The Submit Question Scene is loaded

4.11.1.3 The user is asked to enter the question text

4.11.1.4 The user presses Next button

4.11.1.5 The user is asked to enter the correct answer

4.11.1.6 The user presses Next button

4.11.1.7 The user is asked to enter incorrect answer 1

4.11.1.8 The user presses Next button

4.11.1.11 The user is asked to enter incorrect answer 2

4.11.1.11 The user presses Next button

4.11.1.12 The user is asked to enter incorrect answer 3

4.11.1.13 The user presses Finish button

4.11.1.14 The system serialises the question data and enters it to the Questions Table on the Let’s Play server via a PHP script also stored on the Let’s Play server

4.11.1.15 A success message is presented to the user and the Main Menu Scene is loaded

### 4.12 Other Quality Requirements

#### 4.12.1 Internet connection

The application needs to be able to communicate with the Let’s Play Server

## Use Case 5: Exit the application: End Goal