Let’s Quiz

Initial Requirement Model

20/03/2018

Table of Contents

[1. Introduction 2](#_Toc509066669)

[1. Purpose 3](#_Toc509066670)

[2. Scope 3](#_Toc509066671)

[2. System-Wide Functional Requirements 3](#_Toc509066672)

[4. System Qualities 4](#_Toc509066673)

[1. External interface requirements (Non-functional requirements) – 4](#_Toc509066674)

[i) Usability 4](#_Toc509066675)

[ii) Reliability 4](#_Toc509066676)

[(1) Availability 4](#_Toc509066677)

[(2) Accessibility 4](#_Toc509066678)

[iii) Performance 4](#_Toc509066679)

[(1) Response Times 4](#_Toc509066680)

[(2) Capacity 4](#_Toc509066681)

[(3) Throughput 4](#_Toc509066682)

[iv) Supportability 4](#_Toc509066683)

[(1) Compatibility 4](#_Toc509066684)

[(2) Maintainability 4](#_Toc509066685)

[(3) Documentation Requirements 4](#_Toc509066686)

[v) Security 5](#_Toc509066687)

[4. System Constraints 5](#_Toc509066688)

[5. Assumptions and dependencies 5](#_Toc509066689)

[6. Domain Diagram 5](#_Toc509066690)

[7. Functional requirements 6](#_Toc509066691)

[1. Use cases 6](#_Toc509066692)

[(1) Use case diagrams 9](#_Toc509066693)

VERSION INFORMATION

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author(s) | Description |
| 0.1 | 17 March 2006 | Michelle Vinall | Initial draft |
|  |  |  |  |
|  |  |  |  |

# 1. Introduction

The aim of the Let’s Quiz project is to design an online, multi-player, trivia question game, that allows players to register or login using social media accounts or as a guest. It will allow social interactions such as Invite friends and share. The game will allow highest score recording and meaningful, fun game play.

We are going to create the game with the game engine Unity 3D, using C# as the primary programing language as Unity has many advantages for game making. Some of these are extended platform support of 27 platforms, Unity has a game engine, an IDE and a user interface application all geared towards game development.

## Purpose

This Initial Requirements Model document will describe the requirements and specifications of the Let’s Quiz online trivia game. We will use this document to set expectations of for the development of this project A requirements documentation is needed to guide the developers through to completion and should assist, our developers to define the intended functionality and parameters needed to develop this project.

## Scope

* Produce an on-line single/multiplayer trivia game
* Incorporate Facebook and Google play services SDK’s into game
  + Facebook login
  + Facebook sharing
  + Facebook inviting
  + Google Play services Leader board
  + Google Play services Achievements
  + Google Play services Login
* Database integration
  + Global high score
  + Player statistics
  + Managing game states of open games between players
* Server integration
* Finalize all implementations of game to be playable
* Complete all tests to remove a bugs

# 2. System-Wide Functional Requirements

* F-1-Security Services
  + Security services will be needed as we will be needing to authenticate users logins and will assist in these processes
* F-2-Persistence Services
  + This will be done with a SQL database to store data in a non-volatile manner
* F-3-Language Services
  + Language services will be integrated as English to start off with a hope of extending in future updates to support other languages
  + Services that allow players with a disability would be beneficial and will be implemented wherever possible
* F-4 Networking Services
  + Networking services will be required for multi-player playing or to login with Facebook or Google Services, and for the majority of interactions available in the game.

# 4. System Qualities

The system Qualities are known as non-functional requirements. These are attributes of the system and describe how the software will do a required task, not what the task entails. The FURPS+ acronym sums up the requirements F-Functional, and the URPS are the non-functional requirements

## External interface requirements (Non-functional requirements) –

### Usability

* The user interface should be easy to use
* Interface should be compatible with mobile device screens
* The game mechanisms should be easy to learn, and navigate around.
* The users should be able to compete tasks in a reasonable amount of time

### Reliability

#### Availability

* The game should be available to players on request at least 99% of the time
* The application should have no more than 1 hour down time in any 2 month period
* The application should save progress at set stages so that if a failure occurs the user does not need to repeat extreme amounts of gameplay

#### Accessibility

* Once the game is installed the user should have 24/7 availability of use.

### Performance

#### Response Times

* The users should be able to see a response from their interactions within 5 seconds optimum tine 2 seconds
* For game launch the game should be playable within 30 seconds of launching optimum 10 seconds

#### Capacity

* The system should be capable of handling 100 users at any one time

#### Throughput

* The throughput will be dependent on the number of players at any one time

### Supportability

#### Compatibility

* The game will need to be compatible with both Android and IOS devices

#### (2) Maintainability

* The Game may wish to be added to in future updates so it will be advisable to begin with refactorable, clean code and thorough documentation

#### (3) Documentation Requirements

* All required documentation will follow version control and all reporting protocols. We will supply all documentation necessary for the project

### v) Security

* + As the program will have a database that contains users details the system will need to be secure.
  + Authentication of users can be done by the system for registered users and the individual SDK’s for Facebook and Google play services users

# 4. System Constraints

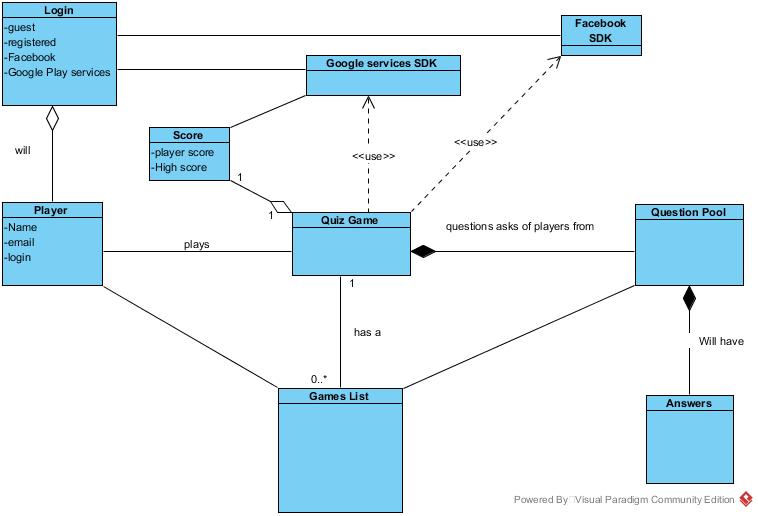
Constraints are the plus in FURPS+ and include

* The game will require an internet connection
* Platform dependent on features
* Most of the coding will be done with C#, PHP
* The database will be in SQL.
* The user interface will have different levels of authentication handled by separate SDKs

# 5. Assumptions and dependencies

* That the mobile device the application is installed on has adequate performance abilities.
* That the user of the application has minimal technology knowledge
* That all developers have knowledge of the required IDE and other necessary aspects including Unity 3d, Facebook SKD, Google Play SKD.
* I will be assumed that the of Facebook and Google Play services servers are available

# 6. Domain Diagram



# 7. Functional requirements

## 1. Use cases

Use Case: End Goal: Launching Game

When the user

Wants to start the application they click the application icon on their device

So that the application opens to show the login screen

Use Case: End Goal: Login

When the user

Wants to login, they must choose login option

So that that the application allows login and displays pregame screen

Use Case: Login Registered users-Sub-function End Goal: Login

When the user

Wants to play by logging in, they then press login

So that that the application opens to the login screen

Use Case: Login with Facebook-Sub-function End Goal: Login

When the user

Wants to play by logging in with Facebook, they then press Facebook login

So that that the application connects to the Facebook Authentication server and allows login

Use Case: Login with Google play services-Sub-function End Goal: Login

When the user

Wants to play by logging in with Google play services, they then press Google play services login

So that that the application connects to the Google play services Authentication server and allows login

Use Case: Play as Guest-Sub-function End Goal: Login

When the user

Wants to play without logging in or first registering they press play as guest

So that that the application opens to the pre-game screen

Use Case: End Goal: Start a Game

When the user

Wants to start a new game they will press the start new game button

So that the application will either start a new game or join an existing game

Use Case: Choose Game Mode-Sub-function End Goal: Start a Game

When the user

Wants to Choose game mode they close the game mode by pressing Corresponding mode

So that the application opens the correct game state

Use Case: Continue an existing game-Sub-function End Goal: Start a Game

When the user

Wants to take their turn in a previously started game they press the games description

So that the application opens the correct game state

Use Case: End Goal: Answer question

When the user

Wants to answer a question they select the correct answer

So that the game can check the answer for correctness

Use Case: End Goal: Facebook share

When the user

Wants to share game they click share on face book button/link

So that the application connects to the Facebook server and allows sharing

Use Case: End Goal: Facebook Challenge/Invite

When the user

Wants to Challenge/Invite they click the challenge button

So that So that the application connects to the Facebook server and sends invitation

Use Case: End Goal: Check Leader board

When the user

Wants to check the leader board scores they press the leader board button

So that the application connects to the Google play Services server to display the leader board

Use Case: End Goal: Check Achievements

When the user

Wants to check their achievements they press the achievements button

So that the application connects to the Google play Services server to display their achievements

Use Case: End Goal: Exit application

When the user

Wants to exit the application they press the back button on android and home button on IOS

So that the application closes down

#### Use case diagrams

Basic outline use case application individual use cases underneath

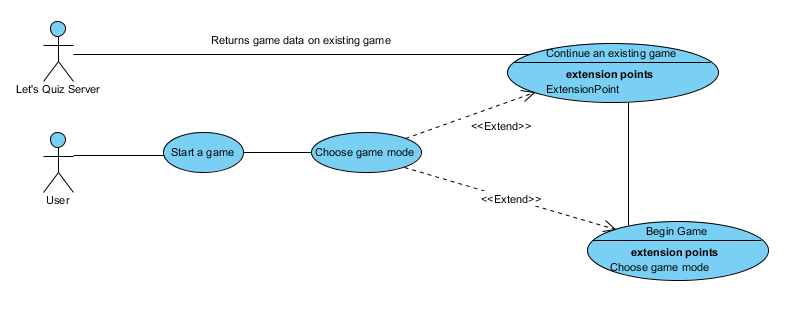


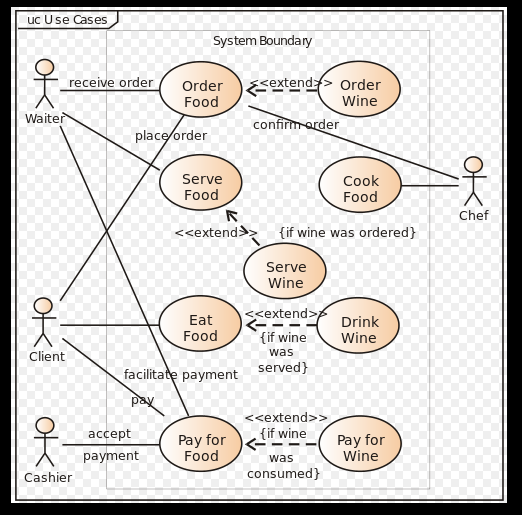
Use Case: End Goal: Login



Use Case: End Goal: Start a Game







Use Case: End Goal: Answer question



Use Case: End Goal: Facebook Share and Challenge/Invite



Use Case: End Goal: Check Leader board/achievements

