## 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