# Quiz Application

## Revision History

|  |  |  |
| --- | --- | --- |
| Date | Author | Description of change |
| 3/6/19 | JSW | Initial draft |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Use Case:** Plays Game

**Id**: UC- 113

**Description**

Player starts game and is prompted to select a category. After choosing from the list, the player is presented with a question which was randomly selected from the category. The player selects an answer and is immediately shown the correct answer along with the updated running score. The player then chooses whether to answer another question or end the game.

**Level:** High Level

**Primary Actor**

Player

**Supporting Actors**

None

**Stakeholders and Interests**

Player: Plays the game for entertainment and to increase knowledge

**Pre-Conditions**

Remote database containing questions and running score must be available

**Post Conditions**

Success end condition

At least one question asked and answered. Running score accurately reflects results of game(s)

Failure end condition:

Answer not accepted or score does not reflect game results

Minimal Guarantee

Game will end cleanly and user will know whether or not question was scored

**Trigger**

Player launches application

## Main Success Scenario

1. Player launches application
2. Player selects a category
3. Game presents a random question and possible answers, one of which is correct
4. Player chooses an answer
5. Game presents results and updated score
6. Player selects whether to end the game or see another question

## Extensions

3a. In step 3, if randomly selected question has been asked during the current session, a new question is presented to the player

1. Game compares random question to list of questions shown during current session
2. Game selects another random question from the same category
3. Process repeated until a valid question is ready to present to the user

## Variations

1. In step 1, the Player may abort the game before selecting a category.
2. In step 3, the Player may abort the game before answering a question. The score is not updated.

**Frequency:** Daily

**Assumptions**

The database includes enough questions to keep the Player engaged over many sessions and to avoid having to select alternate questions too often.

## Special Requirements

Performance

1. The backend database shall return a session-unique random question within 500ms

User Interface

1. The Player should be able to play the game using one hand

Security

1. No personally identifiable information shall be stored in the client or server

## Issues

1. How should the application respond if a question is not returned in time or if the connection is broken?

## To do

1. Design UX

To learn more about “TopTeam Analyst for Use Cases” visit [www.technosolutions.com](http://www.technosolutions.com)