

Group 5:

Yidi Wu

Iulian Toma

Ji Nan

Want to Be a Millionaire?

* URS

Table of Contents

[1. Introduction 2](#_Toc413111904)

[2. Functional Requirements 3](#_Toc413111905)

[3. Non-Functional Requirements 3](#_Toc413111933)

[4. Use Case Diagram 4](#_Toc413111933)

5. [Use-Cases: 4](#_Toc413111906)

[1. Ask a Question. 4](#_Toc413111907)

[2. Close application 5](#_Toc413111908)

[3. Play again 5](#_Toc413111909)

[4. Ready to play. 5](#_Toc413111910)

[6. User Interface 6](#_Toc413111932)

Introduction:

In this project, we have to build small game program and the first step is to write the User Requirements Specifications. For this step we will provide the use-cases of all functionality that we can offer, a specification of user interface and also non-functional requirements. The objective of our application is to let two players against game.

In our game, two players can answer ten questions. If one of them answer the questions correctly more than the other, he will be the winner. And there is a textbox beside the question, it will update the result automatically after every question. Besides, after game is over, the players can choose to play again.

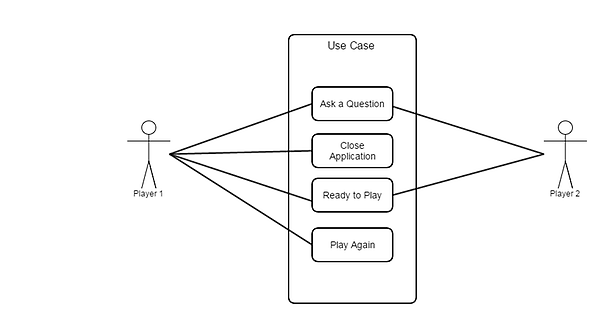
Functional Requirement:

|  |  |  |
| --- | --- | --- |
| Nr. | Description | MOSCOW |
| 1 | Playable by only 2 players | M |
| 2 | Internet connection required | M |
| 3 | Score displayed automatically after each correct answer | C |
| 4 | The game can be stopped anytime by one of the players | M |
| 5 | Answer time is maximum 1 minute otherwise system skips to the next question | C |
| 6 | If Player 1 quits , then Player 2 becomes the winner | M |

Non-Functional Requirement:

1. Only support Window 7, 8.
2. Not possible to play against computer.
3. No any update.
4. No any maintenance.
5. Game cannot be paused.

Use Case Diagram



Use Cases:

1. **Ask a Question**

**Goal:** Player can answer a question and afterwards next question shows to the other player

**Pre-Condition:** Question is shown on the screen

**Actor:** Player

**Trigger:** User clicks on the options and choose confirm, the system shows if it’s correct and the next question to the other player

**MSS:**

1. Player clicks on one of the given options.
2. Player clicks on confirm
3. System checks if the answer is correct.
4. If it’s correct, user’s score raises by 1.
5. System shows the next question to the other player

**Extension:**

4a. If it’s not correct, the score won’t be raised.

1. **Close Application**

**Goal:** Application closed.

**Pre-Condition:** Application must be started and the current game is finished or in the process

**Actor:** Player

**Trigger:** User either presses “Game” “Exit” from the top menu or uses the “X” button from the right up side

**MSS:**

1. System shows the game result on the message box.
2. System closes the running application

**Post-Condition:** Program closed.

1. **Play Again**

**Goal:** To start a new game.

**Pre-Condition:** Players finished their game an

**Actor:** Player

**MSS:**

1. Player clicks on Play Again button
2. System cleans up everything and shows the initialized screen
3. Go to use case 4

**Post-Condition:** Game starts.

1. **Ready to Play**

**Goal:** Ready to play a game.

**Pre-Condition:** Players opened the game program or finished their game.

**Actor:** Player

**MSS:**

1. Player clicks on Ready button
2. Button text changes to OK.
3. Player waits for the other player until he is ready.
4. System shows game starts.

**Extension:**

**3a.** Player clicks on OK button, button text changes to Ready.

**Post-Condition:** Game starts.

User Interface:

