Cairo University  
Faculty of Computers and Information



**CS251**

**Software Engineering I**

Project Name

Software Requirements Specifications

Team names

Omar ahmed

Hoda gamal

Mostafa Mohamed

Ezzat hany

11/2016

Contents

[Team 3](#_Toc402452669)

[Document Purpose and Audience 3](#_Toc402452670)

[Introduction 3](#_Toc402452671)

[Software Purpose 3](#_Toc402452672)

[Software Scope 4](#_Toc402452673)

[Definitions, acronyms, and abbreviations 4](#_Toc402452674)

[Requirements 4](#_Toc402452675)

[Functional Requirements 4](#_Toc402452676)

[Non Functional Requirements 4](#_Toc402452677)

[System Models 5](#_Toc402452678)

[Use Case Model 5](#_Toc402452679)

[Use Case Tables 5](#_Toc402452680)

[Ownership Report 6](#_Toc402452681)

# 

# Team

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Email** | **Mobile** |
| 20140180 | Omar ahmed | Omarbadry28@gmail.com | 01151475706 |
| 20140301 | Hoda gamal | Hodagamal10.hg@gmail.com | 01151601669 |
| 20140270 | Mostafa Mohamed | Mustafalotfy48@gmail.com | 01125206371 |
| 20140173 | Ezzat hani | Ezzathany8@gmail.com | 01122148138 |

# Document Purpose and Audience

**Document purpose (what this document is)**

The purpose of this document is: describing an educational game platform, in terms of its functional and nonfunctional requirements with its use cases model and table .it serves as initial contract between customer and developer.

**Audience**

The audience includes Customers, system developers, system designers and the organization that might market for the project.

# Introduction

## Software Purpose

* + - * The purpose of this educational game is helping students to learn many sciences while they are playing such as math and simple coding and technology. This will be in a simple way and will be free. The students also can communicate and react with teachers who made this game.

## Software Scope

* + - The game is free and provides questions, student answer it in limited time.
    - It's support a specific type of games (educational games)
    - Games distribute to 4 categories according to its type.
    - Students Contact with teachers in comments.
    - Web application.
    - Accounts are classified into two types teacher or student.
    - Everything that has been mentioned in the requirements is within the scope.

## Definitions, acronyms, and abbreviations

UI: user interface.

# Requirements

Overview

The game is a one-player educational game web application. The game user interface is; upper part there is sign in, sign up buttons and searching text-box to search for a game, The game user interface displays the categories that contains lists of games that belongs to this categories, each game consists of lots of questions, the best rated games will be shown..., player choose the game he want to play, system ask him if to sign in to play the game. If user sign in , the game begin and game user interface displays at right top list contains user’s account setting , and for teacher’s account there is option to upload a game or to update a game he made before. game user cannot play the game unless have an account and sign in it.

The game consists of a questions and multiple choices to choose the right answer from them .questions difficulty and the numbers of multiple choices set according to student’s previous score. The student wins when he chooses the right choice, his score increase and level of difficulty increase. He loses when he chooses wrong answer or when time is up.

## Functional Requirements

The player should be able to:

* + - Sign up as student or teacher.
    - Sign in to user account.
    - Edit profile.
    - Choose the game.
    - Play the game.
    - Rate the game.
    - Make comments and responds.
    - Check current score.
    - Save highest score.
    - Exit current game.
    - Upload a game.
    - Approve uploading a game.
    - Update a game.
    - Sign out from account.

**Describing functional requirements:**

**Sign up an account***.*

Student or teacher can make account to be able to play the games ,save their previous score, to make comments in case of teacher’s account allow him to upload or update the game .Account name should be unique, User enter his real name, age, gender email, etc...Choose if he is a teacher or student. If he is a teacher he should fill more things like his experience, where is he works etc…

**Sign in to user account.**

The user should have an account to be allowed to enter the game by entering user name and password.

**Edit profile.**

User can change the information of his profile such account name, password, etc…

**Choose the game.**

As the system is: an educational web platform containslists of categories, each one has many games in it .so the user should choose a game from this lists to play it.

**Play the game.**

The user student/teacher will see a lot of questions. He will try to answer these questions by choosing the correct answer from multiple choices (choose only one answer).each question has limited time. After answering the question, the game will evaluate the user according to his / her answer.

**Rate the game.**

After answering the questions, the user will evaluate the game from 1 to 5, 1 is the less degree and 5 is the highest and make any feedback, ask or response to a question.

**Check current score.**

System checks the current score of the user and compares it with the highest score.

**Save current score.**

System saves the current score after checking the current score.

**Make comments, responds or communicate.**

The student can connect with his / her teacher by making comments and the teacher responds in a comment.

**Stop current game.**

The user can stop a game by pressing the stop button.

**Upload a game**

This feature is For the teachers only. The teacher can upload a game from his account, put it in its category and write a description about it then the system should verify and confirm this.

**Approve uploading the game.**

System checks the game quality before approving the game and uploads it to the site.

**Update a game**

The teacher can update a game that he has uploaded before from his account, this update is according to the student’s feedback or any developments the teacher wants to make.

**Sign out from account**.

The user can exit his account by doing sign out.

## 

## Non Functional Requirements

* **Non-functional requirements describe how the system works**

|  |  |
| --- | --- |
| **Name** | **Details** |
| **Performance** | * The Time for the user to sign in doesn’t exceed 2 seconds * The game starts 2 seconds after the user clicks play game, so the startup time is very low. * It takes 1 second from the system to change the question after the user chooses the correct answer. |
| **Maintainability** | * Checking well the games uploaded in fit category with high quality and fast to upload any game |
| **Portability** | * This platform is for web applications. |
| **Human Factors** | * The UI must be user-friendly. * The UI must be responsive. * All categories menus must have a consistent format. |
| **Reliability** | * It's non-stop and free system that available at any time |
| **Security/privacy** | * The system protects the user personal information. * Protect the user account from hacking. * Usable only in authorized ways by authorized users; as the user must has an account to play the games. * Protect the user rights in owning his uploaded game. |
| **Usability** | * The game is easy to use and simple to understand; for every game there is a description to describe how to play the game. * A user-friendly interface |
| **Scalability** | * Used by varying numbers of users, there is no limitation on the number of users. |

**Scenarios**

**Start the game**

User interface(starting page) displays the categories, which consists of many games belongs to this category, at the top of the website there is sign up ,sign in buttons and text-box search. At the middle of the page; the most rated games appear. To play a game; the user must has an account and sign in with it, after the user signs in he can choose the game he wants to play by clicking on its icon.

**Play the game**

After the game is ready ,many questions with multiple choices will appear to the user one by one with limited time for each question to answer, the user tries to solve this questions by clicking on the right choice ,as he chooses the correct answer his score and the level of difficulty increase. The student also can rate the game he playing. if user needs any query about this game he can make a comment and teacher responds to the comment. if user wants to exit the game there is an exit button at the top of the game.

**Correct choice**

The user click on a right choice before limited time exceeded, the score will increase and difficulty of the questions increase and next question appears. After the user finishes all questions the score appears.

**Wrong choice**

If The user clicks on a wrong choice or limited time exceeded, the score will not change, difficulty of the questions increase and next question appears. After the user finishes all questions the score appears.

**Win the game**

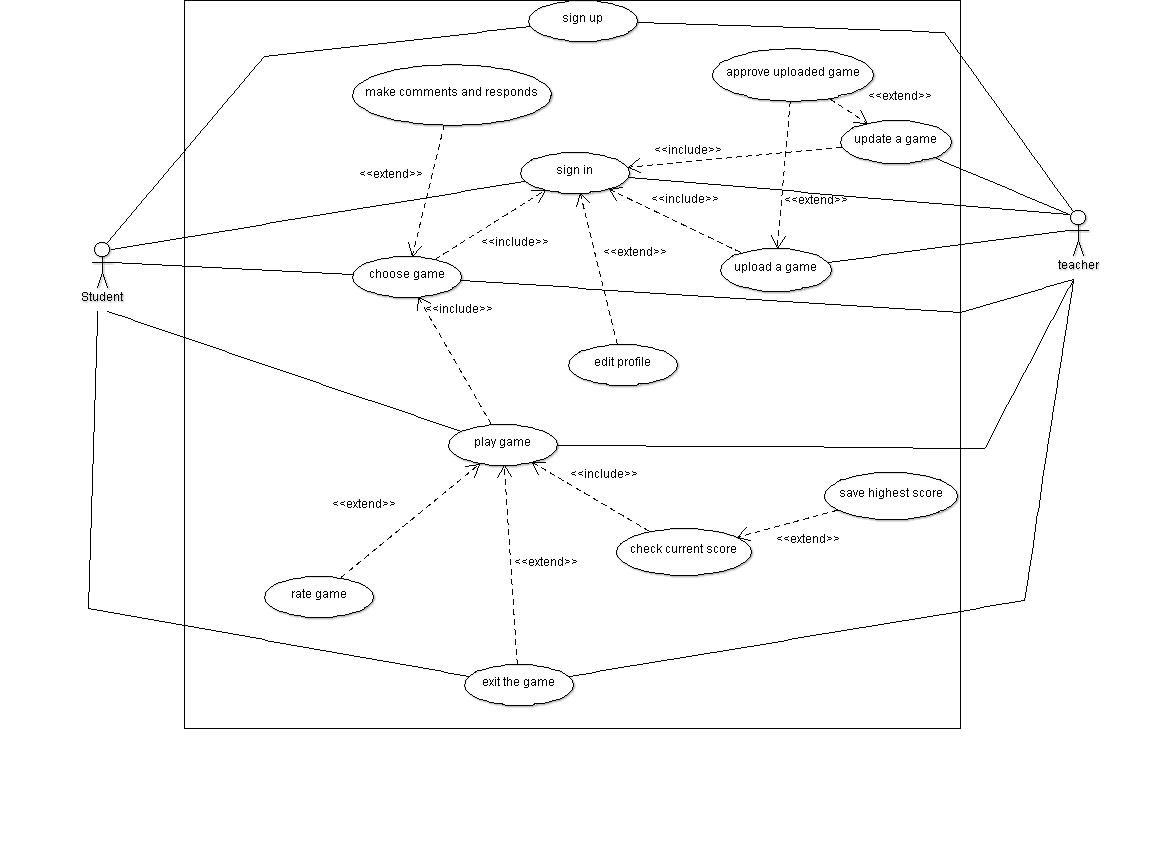
If the score is greater than highest score, the highest score will be updated.

**Upload a game**

To upload a game and update it; the user account must be a teacher ,he can upload a game to the site and choose its category the ,system checks the quality of the game before other users can play it . If system confirms the uploading; the game owner the only one can update, develop, delete it. And notify for any comment or feedback about his game.

# System Models

## Use Case Model

Figure 1: Overview of identified use cases (UML use case diagram).

## Use Case Tables

The following system actors are identified:

**Student**

The student is able to play games. He can win or lose a game, he can also rate any game and makes comments.

**Teacher**

The Teacher is able to play games. He can win or lose a game; he can also rate any game, write comments, respond to students' comments, upload a game and update it at any time.

|  |  |  |
| --- | --- | --- |
| Use case id | 1 |  |
| Use Case Name: | **Sing up** |  |
| Actors: | Student , teacher |  |
| Pre-conditions: | Open the web-site |  |
| Post-conditions | User will have an account |  |
| Flow of events: | **Actor steps**  2-actors fill information as user-name, password, email, age etc… in case of teacher account will fill more fields as his experience etc… | **System steps**  1- ask user for enter the type of the account  3-system will check validation of the information that user enters. |
| Exceptions: | **User action**  1-user enters email or name is already taken before by another user or invalid information. | **System action**  2- Displays a message that email and password are wrong.  3-account won’t open. |
| Includes: | none |  |
| Notes and Issues: | none |  |

|  |  |  |
| --- | --- | --- |
| Use case id | 2 |  |
| Use Case Name: | **Sign in** |  |
| Actors: | Student , teacher |  |
| Pre-conditions: | User has account |  |
| Post-conditions | Account has opened |  |
| Flow of events: | **Actor steps**  1-user enters user-name and password | **System steps**  2-system check validation |
| Exceptions: | **User action**  1-user enters invalid user-name or password.  3-user edits his profile information | **System action**  2- displays a message to make the user enter information again |
| Includes: | 3,11,12 |  |
| Notes and Issues: | None |  |

|  |  |  |
| --- | --- | --- |
| Use case id | 3 |  |
| Use Case Name: | **choose game** |  |
| Actors: | Student , teacher |  |
| Pre-conditions: | This use case is including in sign in case. |  |
| Post-condition | Game begins. |  |
| Flow of events: | **Actor steps**  1-user chooses the category of the game he wants to play.  3-user chooses the game. | **System steps**  2-system shows the games that exist in the category the user has chosen. |
| Exceptions: | none | none |
| Includes: | 4 |  |
| Notes and Issues: | none | none |

|  |  |  |
| --- | --- | --- |
| Use case id | 4 |  |
| Use Case Name: | **Play game.** |  |
| Actors: | Student , teacher |  |
| Pre-conditions: | This use case is include in choose game. |  |
| Post-conditions | The game begins. |  |
| Flow of events: | **Actor steps**  2-user chooses one of these choices. | **System steps**  1-system begins the game and shows a multi choice question.  3-system check if the user’s choice right or not.  4-if it is right score and difficulty will increase.  5-if answer is wrong no change in the score will happen.  6- system shows the score. |
| Exceptions: | **User action**  1-user exits the game.  3-student rate the game. | **System action**  2-score won’t saved |
| Includes: | 7 |  |
| Notes and Issues: | none |  |

|  |  |  |
| --- | --- | --- |
| Use case id | 5 |  |
| Use Case Name: | **Rate game.** |  |
| Actors: | Student , teacher |  |
| Pre-conditions: | user play the game |  |
| Post-conditions | A new rate has added. |  |
| Flow of events: | **Actor steps**  2-user chooses one of these choices. | **System steps**  1-system shows a box has 5 choices from 1 to 5 .  3-system saves this choice and send it to the game owner. |
| Exceptions: | **User action**  none | **System action**  none |
| Includes: | none |  |
| Notes and Issues: | none |  |

|  |  |  |
| --- | --- | --- |
| Use case id | 6 |  |
| Use Case Name: | **Exit the game.** |  |
| Actors: | Student, teacher. |  |
| Pre-conditions: | User plays a game. |  |
| Post-conditions | The game has exited. |  |
| Flow of events: | **Actor steps**  1-user presses the exit button.  3-user confirms or rejects exiting. | **System steps**  2-ask user to confirm exit.  4-if user confirms, back to user interface that contains categories of the games.  5-if user rejects game continues. |
| Exceptions: | **User action**  none | **System action**  none |
| Includes: | none |  |
| Notes and Issues: | none |  |

|  |  |  |
| --- | --- | --- |
| Use case id | 7 |  |
| Use Case Name: | **Check current score.** |  |
| Actors: |  |  |
| Pre-conditions: | The user play a game |  |
| Post-conditions | System checking the current score. |  |
| Flow of events: | **Actor steps**  1-user finishes all the questions or he losses the game. | **System steps**  2-system compares his score with the highest score. |
| Exceptions: | **User action**  none | **System action**  none |
| Includes: | none |  |
| Notes and Issues: | none |  |

|  |  |  |
| --- | --- | --- |
| Use case id | 8 |  |
| Use Case Name: | **Save highest score** |  |
| Actors: |  |  |
| Pre-conditions: | System checks current score. |  |
| Post-conditions | System saving the highest score. |  |
| Flow of events: | **Actor steps** | **System steps**  1-if current score is greater than the highest score , system save current score as the highest score. |
| Exceptions: | **User action**  none | **System action**  none |
| Includes: | none |  |
| Notes and Issues: | none |  |

|  |  |  |
| --- | --- | --- |
| Use case id | 9 |  |
| Use Case Name: | **Make comments and responds.** |  |
| Actors: | Student , teacher |  |
| Pre-conditions: | Sign in to an account and choose a game. |  |
| Post-conditions | none |  |
| Flow of events: | **Actor steps**  1-if user is a student and he wants to ask or make a query then he makes a comment.  3-owner’s game will contact with the user that makes the comment. | **System steps**  2-system will notify the owner of the game. |
| Exceptions: | **User action**  none | **System action**  none |
| Includes: | none |  |
| Notes and Issues: | none |  |

|  |  |  |
| --- | --- | --- |
| Use case id | 10 |  |
| Use Case Name: | **Edit profile.** |  |
| Actors: | Student , teacher |  |
| Pre-conditions: | Sign in to an account. |  |
| Post-conditions | Profile has edited. |  |
| Flow of events: | **Actor steps**  1-user changes the information he wants like name , email, password, etc…  2-user clicks save information. | **System steps**  3-system checks validations. |
| Exceptions: | **User action**  1- User enters invalid information. | **System action**  2-system asks user to enter the invalid information again and make it valid. |
| Includes: | none |  |
| Notes and Issues: | none |  |

|  |  |  |
| --- | --- | --- |
| Use case id | 11 |  |
| Use Case Name: | **Upload a game** |  |
| Actors: | Teacher |  |
| Pre-conditions: | Sign in by teacher’s account. |  |
| Post-conditions | Game uploaded. |  |
| Flow of events: | **Actor steps**  1-upload the game he wants.  2-choose the category that the game belongs to and put it in. | **System steps**  3-check the quality of the game.  4- If system accepted the game it will make the game for all users who had an account. |
| Exceptions: | **User action**  2-user updates his game according to the feedback from the system and can upload it again. | **System action**  1-if system refuses the game it will contact with the user who wants to upload it and told him the missing rules. |
| Includes: | None. |  |
| Notes and Issues: | None. |  |

|  |  |  |
| --- | --- | --- |
| Use case id | 12 |  |
| Use Case Name: | **Update a game** |  |
| Actors: | Teacher |  |
| Pre-conditions: | User has an uploaded game. |  |
| Post-conditions | The game has updated. |  |
| Flow of events: | **Actor steps**  1-choose the game that he wants to update from his uploaded games. | **System steps**  2-check the quality of the game.  3- If system accepted the updates the game will update. |
| Exceptions: | **User action**  2-user changes in the updates according to the feedback from the system. | **System action**  1-if system refuses the game updates; system will contact with the game owner who wants to update it and told him the missing rules. |
| Includes: | None. |  |
| Notes and Issues: | None. |  |

# Ownership Report

|  |
| --- |
| we have participated all together in every point in this document. |