Cairo University  
Faculty of Computers and Information



**CS251**

**Software Engineering I**

**Educational Entertainment Website**

Software Requirements Specifications

Names:

# Team

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Email** |  |
| 20140377 | Omar abdulazizsaeed | omer.alqubati@gmail.com | Leader |
| 20170375 | Hameed Mohammed Abdullah | Hameed203083@gmail.com |  |
| 20140367 | Asseel Ahmed mohmed | asseel7723@gmail.com |  |
| 20140339 | Wafaasamirahmedalahmer | fofoalah@gmail.com 20140339 |  |
| 20140341 | Ziadamer al-tahhan | ziadaamer 9@gmail.com |  |

# Document Purpose and Audience

**Document:**

This document contains the Functional and non-functional requirements of the software.

**The Audiences:**

The Audiences are the ones involved in the production of that software:

End users, customers, product managers, project managers,

Software engineers, designers, developers, and testers.

# Introduction

## Software Purpose

This Web Application will serve as a platform for teachers to host their educational games.

Students will be able to play those games.

## Software Scope

This software will be Web Application only.

# 

# Requirements

## Functional Requirements

1- A teacher must be able to add more than one game which is then added to his list via the system.

2- Each game a teacher adds, targets a certain level.

3- The game should have description, instruction, and level to be successfully uploaded.

4- A teacher can remove any game from his games list

5- A teacher can upgrade any game from his games list.

6- A teacher must be able to check feedbacks from his games list.

7-Ateacher can reply to student's comments.

8- A student must be able to play games.

9- A student can give his feedback to games that he already has played.

10- A student has the opportunity to see the description of a game.

11- The system should save the level and score in which a student has stopped.

12- After the student finishes playing the system adds the score to his level experience bar.

13- When the level bar is full the student is transferred to the next level.

## 

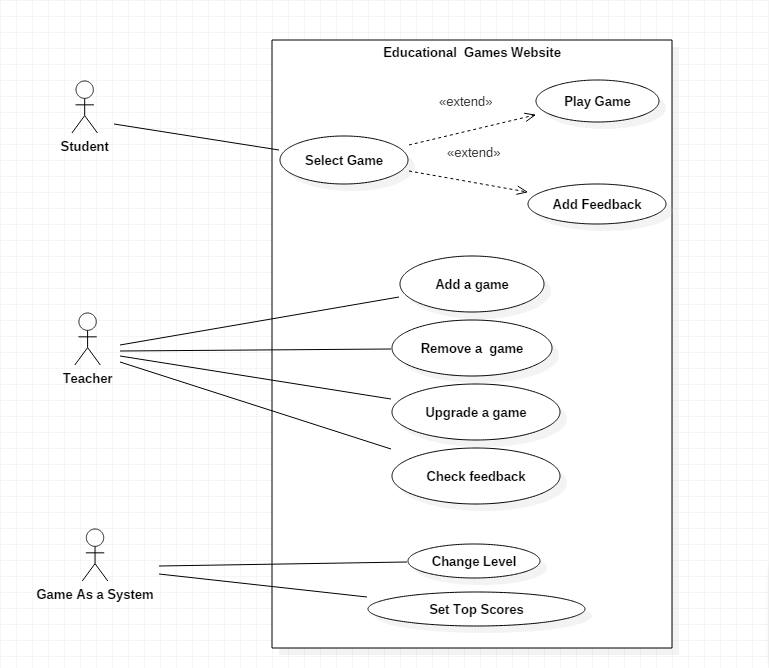
## Non Functional Requirements

|  |  |
| --- | --- |
|  | **Details** |
| Performance | 1- The system will be respond to the end user in not less than two seconds from the time of the request submittal, but it will take more time when large operations are done like add new game or delete existing one.  **2-** 1000 student should be able to open the website at the same time. |
| Security Requirement: | 1-The system uses a form of protection, a user name and password for each user, such that the user will be distinguished a student or teacher. |
| Reliability Requirement: | The system has to be 100% reliable, because it's a teaching tool for studying the basic principles of some science fields as there should be no games which are not educational. |
| Availability Requirement: | The system will be available all the time for the end users, and will be operational 24 hours a day and 7 days a week. |
| Usability Requirement: | The system uses a web browser as an interface. Since all users are familiar with the general usage of browsers, no special training is required. |

# 

# System Models

## Use Case Model



## 

## Use Case Tables

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 1 | |
| Use Case Name: | Select Game | |
| Actors: | Student | |
| Pre-conditions: | 1-Logged in  2-Select certain game | |
| Post-conditions: | 1-Start to play the game.  2-Add feedback to the game. | |
| Flow of events: | **User Action** | **System Action** |
| 1- Student log in. |  |
|  | 2- System Displays all the games |
| 3- Student choose the game |  |
|  | 4- System Displays information of game. |
|  | 5- Student will choose either to play or add a feedback. |  |
| Exceptions: | **User Action** | **System Action** |
| No exception |  |
| Includes: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 2 | |
| Use Case Name: | Play game | |
| Actors: | Student | |
| Pre-conditions: | 1- Student selects a game.  2- Student's level fits the game's level. | |
| Post-conditions: | 1- Lose the game.  2- End of the game's Level. | |
| Flow of events: | **User Action** | **System Action** |
| 1-Student Press "start play button" |  |
|  | 2- System Load the game. |
| 3- Student Choose whether to start new game or continue. |  |
| 4- Student Start playing the game. |  |
|  | 5- After finishing a level. The systemsaves the current level as played level. |
| Exceptions: | **User Action** | **System Action** |
| No Exceptions. |  |
| Includes: |  | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 3 | |
| Use Case Name: | Add Feedback | |
| Actors: | Student | |
| Pre-conditions: | 1- Student selects a game.  2- Student already played the game. | |
| Post-conditions: | 1-Uploading the feedback. | |
| Flow of events: | **User Action** | **System Action** |
| 1-Student selects a certain game. |  |
| 1-Student Press the "Add feedback button". |  |
|  | 2- System Prints the ways of submitting the feedback. |
| 3- Student Choose a way of submitting the feedback. |  |
|  | 4- System Displays average rate and the comments. |
| 5- Student Adds his comment or gives a rate. |  |
| 6- Student submits the feedback |  |
|  | 5- System updates and re-calculates the new average. |
| Exceptions: | **User Action** | **System Action** |
| 1-Student press Add feedback and he didn't play the game before. |  |
|  |  | 2-System prints error. |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 4 | |
| Use Case Name: | Add a game | |
| Actors: | Teacher | |
| Pre-conditions: | 1- Logged in account as a teacher. | |
| Post-conditions: | 1- Game finished uploading. | |
| Flow of events: | **User Action** | **System Action** |
| 1- Teacher Choose name and icon of the game and its category. |  |
|  | 2- System verifies the name of the game. |
| 3- Teacher adds description of the game. |  |
| 4-Teacher uploads the game file and its instruction. |  |
|  | 5- System adds the game to teacher's list. |
|  |  | 6- System Displays the new game in the home page. |
| Exceptions: | **User Action** | **System Action** |
| 1- Teacher Chooses name for the game and the name already exists. |  |
|  | 2- System rejects the addition process. |
| Use Case ID: | 5 | |
| Use Case Name: | Remove a game | |
| Actors: | Teacher | |
| Pre-conditions: | 1- The teacher owns the game. | |
| Post-conditions: | 1- The game is deleted. | |
| Flow of events: | **User Action** | **System Action** |
| 1- Teacher enters the list of the games that he uploaded. |  |
| 2- select a game to be removed |  |
|  | 3- System removes the game from1 the home page and teachers list. |
| Exceptions: | **User Action** | **System Action** |
| 1- Teacher removes the game while there are students playing it. |  |
|  | 2-System directly returns Students to the home page. |
| Includes: |  | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 6 | |
| Use Case Name: | Upgrade a game. | |
| Actors: | Teacher | |
| Pre-conditions: | 1- The teacher owns the game. | |
| Post-conditions: | 1- The new version of the game is added. | |
| Flow of events: | **User Action** | **System Action** |
| 1- Teacher enters the list of the game that he uploaded. |  |
| 2- Teacher Select a game to be updated. |  |
| 3- Teacher removes and add the new version. |  |
|  | 4- System adds the game to home page and teachers list again. |
| Exceptions: | **User Action** | **System Action** |
| 1- Teacher upgrading the game while there are students playing it. |  |
|  | 2-System directly returns Students to the home page. |
| Includes: |  | |
| Use Case ID: | 7 | |
| Use Case Name: | Check feedback. | |
| Actors: | Teacher | |
| Pre-conditions: | The game developed by this teacher. | |
| Post-conditions: | Finish all feedback and reply to the students comments . | |
| Flow of events: | **User Action** | **System Action** |
| 1- Teacher Enters the game list he added. |  |
| 2- Teacher Select a game to check its feedbacks. |  |
|  | 3-System opens the games feedback. |
| 4- Teacher responds to the students comment. |  |
|  |  | 5-System uploads the replied message to the students comment. |
| Exceptions: | **User Action** | **System Action** |
| No exception. |  |
| Includes: |  | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 8 | |
| Use Case Name: | Change level. | |
| Actors: | Game as system. | |
| Pre-conditions: | 1- Student succeeds in a game given in his level. | |
| Post-conditions: | 1- The Level experience bar of the Student being filled. | |
| Flow of events: | **User Action** | **System Action** |
| 1- After the student finish playing, the score is calculated. |  |
| 2- The students experience bar gets increased. |  |
|  | 3- System displays the new bar. |
| Exceptions: | **User Action** | **System Action** |
| No exception. |  |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 9 | |
| Use Case Name: | Choose top score players. | |
| Actors: | Game as system. | |
| Pre-conditions: | 1- Student finishes playing a game. | |
| Post-conditions: | 1- Top score list is updated. | |
| Flow of events: | **User Action** | **System Action** |
| 1- Game as a system saves the score of the player. |  |
| 2- Add the score and the students name to the top score players. |  |
| 3- Sort the players depending on the score and choose the best ten. |  |
|  | 4- System displays the top 10 players. |
| Exceptions: | **User Action** | **System Action** |
| No exception |  |

# Ownership Report

|  |  |
| --- | --- |
| **Item** | **Owners** |
|  |  |
|  |  |