Cairo University  
Faculty of Computers and Information



**CS251**

**Software Engineering I**

CodeMorning

Software Requirements Specifications

Amany Tarek

Aya Mohamed

Heba Ahmed

Sara Sabry

November 2016

Contents

[Team 3](#_Toc468213823)

[Document Purpose and Audience 4](#_Toc468213824)

[Introduction 4](#_Toc468213825)

[Software Purpose 4](#_Toc468213826)

[Software Scope 4](#_Toc468213827)

[Requirements 5](#_Toc468213828)

[Functional Requirements 5](#_Toc468213829)

[Non Functional Requirements 6](#_Toc468213830)

[System Models 7](#_Toc468213831)

[Use Case Model 7](#_Toc468213832)

[Use Case Tables 8](#_Toc468213833)

[Ownership Report 18](#_Toc468213834)

# Team

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Email** | **Mobile** |
| 20130075 | Amany Tarek Ahmed | Amany.mony\_94@yahoo.com | 01004027284 |
| 20140300 | Heba Ahmed | heba\_ahmed\_eng@yahoo.com | 01114217325 |
| 20140090 | Aya Mahmoud Mohamed | Ayaelhawary14@gmail.com | 01116591620 |
| 20140133 | Sara Sabry Saad | Sara.sabry.1924@gmail.com | 01099329814 |

# Document Purpose and Audience

This document is a web application proposal. Here we state all the needed information about our project and how it works. Our project is a web application that serves both students and teachers for educational use. Students will learn about some concepts while playing, because with us learning can be fun.

The audience we try to reach here is the CEO of Computek company, to help us carry out this project.

# Introduction

## Software Purpose

Our purpose is to create an entertaining and educational web application that will help students to understand basic educational concepts while they will be playing simple games. Teachers can use it to create their own games as well and encourage their students to play them.

## Software Scope

A Web application that contains only educational games. Games will be related to subjects like science, technology and math. This web application is to be used by students interested in learning about those three fields as well as basic programming concepts. Also, it enables teachers to create new games and try other games.

# Requirements

## Functional Requirements

1. Create accounts

* Both teachers and students can create accounts to use the web App
* Teachers can sign up as teachers and have some additional features on their accounts
* Teacher accounts have some privileges, like: creating games and editing or removing them

1. Verify the identity

* Students and teachers verify their accounts via email
* Verified accounts have access to the web app

1. Play games

* Students select a category then choose a game to play
* The system saves their progress

1. Rate games

* After playing a game, students can give a rate to this game (optional) based on their satisfaction level with it from 1 to 5 (5 being excellent, 1 being poor)

1. Write comments

* Students can write and post (optional) comments on games in the comment section below games

1. Respond to comments

* Teachers view the students comments
* Teachers can choose the reply button under each comment to respond to students comments on their own games
* Teachers can only comment on their own games

1. Manage games

* Techers can choose the manage button under their own games
* They can edit, remove or create games
* Teachers edit/remove their own games only
* The system will save these changes

1. Try any game

* Teachers find that option under every game
* Teachers can try a game without getting a score or reach any achievements

1. Choose category

* Students and teachers choose the category they want
* They browse the list of games under this category
* They may not choose a game from that category and choose to browse another or exit

1. Edit game

* Teachers can choose to edit their own games only
* They can choose to save the changes they made to the game
* They can choose not to save these changes and exits

1. Remove game

* Teachers can choose to remove their own game only

1. Create new game

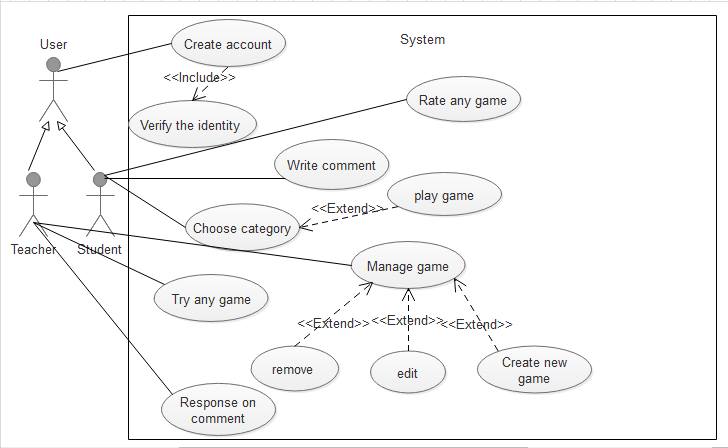
* Teachers can create new games
* New games will be saved to the list of their created games

## Non Functional Requirements

|  |  |
| --- | --- |
|  | **Details** |
| **Performance** | * Loading games will be done within 90 seconds |
| **Scalability** | * System should be able to support up to 100 simultaneous game players |
| **Backup** | * Copy and archive users data in servers |
| **Supportability** | * Developers are ready to fix any problem or add new requested features |
| **Usability** | * An easy to use and understand system * Not too many menus just 3 clicks on the mouse to play the game they want |
| **Security** | * Data of our users are not to be seen by any other party using secured database |
| **Resilience** | * System provides and maintain an acceptable level of service |

# System Models

## Use Case Model



## Use Case Tables

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | 1 | |
| **Use Case Name:** | Create account | |
| **Actors:** | User: Student, Teacher | |
| **Pre-conditions:** | 1. Enter the website 2. Request to create an account | |
| **Post-conditions:** | System sends a confirmation mail to users | |
| **Flow of events:** | **User Action** | **System Action** |
| 1- Send a Sign Up request. |  |
|  | 2- System displays a form where the user can enter his mail, username and password. |
| 3- User enters his mail, username and password. |  |
|  | 4- System sends a confirmation mail to the user. |
| **Exceptions 1:** | **User Action** | **System Action** |
| 1- Use an already taken username. |  |
|  | 1- Reject the sign up.  2- Ask the user to use another name and display some suggested names. |
| **Exception 2:** | 1- User don’t enter the required password/username format. |  |
|  | 2- System displays the required username/password format. |
| **Include:** | 12-Verify the identity | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | 2 | |
| **Use Case Name:** | Rate games | |
| **Actors:** | Student | |
| **Pre-conditions:** | 1. log in to his account 2. Play any game on the website | |
| **Post-conditions:** | This rate will be updated to the game’s rate | |
| **Flow of events:** | **User Action** | **System Action** |
| 1- Choose the game rate from 1 (poor) to 5 (Excellent). |  |
|  | 2- System update the rate of the game. |
| **Notes and Issues:** | Some may create fake accounts to manipulate the rate of a game. | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | 3 | |
| **Use Case Name:** | Write a comment | |
| **Actors:** | Student | |
| **Pre-conditions:** | 1- log in to his account  2- Play any game on the website | |
| **Post-conditions:** | Add comment to the comment section | |
| **Flow of events:** | **User Action** | **System Action** |
| 1- Write a comment.  2- Post the comment. |  |
|  | 3- Add comment to the comment section.  4- Send a notification to the game creator. |
| **Notes and Issues:** | Some people tend to use inappropriate language, we can later handle that. | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | 4 | |
| **Use Case Name:** | Create new game | |
| **Actors:** | Teacher | |
| **Pre-conditions:** | 1- log in as a Teacher  2- Enter “Create Game” area | |
| **Post-conditions:** | Game is added to the games section in the given category | |
| **Flow of events:** | **User Action** | **System Action** |
| 1- Choose game category. |  |
|  | 2- Confirm the category choice. |
| 3- Start creating the game.  4-Finish the game creation. |  |
|  | 5- System add game to the list of games in the given category. |
| **Notes and Issues:** | Teachers may (unintentionally) create games similar to the ones that already exists. | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | 5 | |
| **Use Case Name:** | Respond to comments | |
| **Actors:** | Teacher | |
| **Pre-conditions:** | 1- Create a game 2- Student play any game and write a comment | |
| **Post-conditions:** | Comment is displayed below student’s comment | |
| **Flow of events:** | **User Action** | **System Action** |
| 1- Teacher respond to students’ comments on his own game |  |
|  | 2- System posts his reply below |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | 6 | |
| **Use Case Name:** | Try any Game | |
| **Actors:** | Teacher | |
| **Pre-conditions:** | 1- Log in as teacher  2- Choose category | |
| **Post-conditions:** | Exit game | |
| **Flow of events:** | **User Action** | **System Action** |
| 1- Teacher choose to play a game |  |
|  | 2- System loads the game |
| 3- Teacher play game. |  |
|  | 4- System doesn’t save his progress |
| 5- Teacher exists the game |  |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | 7 | |
| **Use Case Name:** | Choose Category | |
| **Actors:** | Student, Teachers | |
| **Pre-conditions:** | Log in  Choose “Play Game” | |
| **Post-conditions:** | Browse through the games that fall under the chosen category | |
| **Flow of events:** | **User Action** | **System Action** |
| 1- Users enter gaming area |  |
|  | 2- System displays a list of categories |
| 3- User Select a certain category |  |
|  | 4- System shows games under this category |
| 5- User choose a game to play |  |
| **Exceptions:** | **User Action** | **System Action** |
| 1- Users choose a category |  |
|  | 2- System displays a list of categories |
| 3- User Select a certain category |  |
|  | 4- System shows games under this category |
| 5- User exits this category |  |
|  | 6- System displays back the list of category |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | 8 | |
| **Use Case Name:** | Play game | |
| **Actors:** | Student | |
| **Pre-conditions:** | Log in | |
| **Post-conditions:** | Student exits the game | |
| **Flow of events:** | **User Action** | **System Action** |
| 1- Student choose a category |  |
|  | 2- System lists games under this category |
| 3- Student chooses a game |  |
|  |  | 4- System displays loading menu  5- Shows game interface |
|  | 6- Student starts playing  7- Student finish playing |  |
|  |  | 8- System saves student’s score and achievements |
| **Exceptions:** | **User Action** | **System Action** |
| 1- User Enter Card and Password. |  |
|  | 2- Card is invalid and unreadable.  3- System rejects cars. |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | 9 | |
| **Use Case Name:** | Manage games | |
| **Actors:** | Teacher | |
| **Pre-conditions:** | Log in as a teacher | |
| **Post-conditions:** | Teacher click done managing | |
| **Flow of events:** | **User Action** | **System Action** |
| 1- log in as teacher |  |
|  | 2- System Verify username and password from database. |
| 3- Choose manage game. |  |
|  | 4- System displays “manage game” menu. |
| 5- Teacher starts editing/removing/creating. |  |
|  | 6- System saves the changes. |
| **Exceptions:** | **User Action** | **System Action** |
| 1- Teacher choose game he didn’t created. |  |
|  | 2- Display massage: ”you cannot manage this game”. |
| **Includes:** | Remove game, edit game or create new game. | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | 10 | |
| **Use Case Name:** | Edit games | |
| **Actors:** | Teacher | |
| **Pre-conditions:** | Choose the game he wants to edit | |
| **Post-conditions:** | Done editing | |
| **Flow of events:** | **User Action** | **System Action** |
| 1- Choose a game to edit. |  |
|  | 2- Displays the tools to edit the games. |
| 3- Teacher edit the game. |  |
|  |  | 4- Save the game that user is done editing. |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | 11 | |
| **Use Case Name:** | Remove game | |
| **Actors:** | Teacher | |
| **Pre-conditions:** | Choose manage games | |
| **Post-conditions:** | The game is removed from the system | |
| **Flow of events:** | **User Action** | **System Action** |
| 1- User choose to remove the game |  |
|  | 2- System removes it from the system |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | 12 | |
| **Use Case Name:** | Verify the identity | |
| **Actors:** | User: Teacher, Student | |
| **Pre-conditions:** | Request to create account | |
| **Post-conditions:** | Users have their account ready to use | |
| **Flow of events:** | **User Action** | **System Action** |
| 1- User checks his mail  2- Open the link in the confirmation mail |  |
|  | 2- Add user’s username and password to the database.  3- Create an account on the system for the user with his information |
| **Exceptions:** | **User Action** | **System Action** |
| 1- User doesn’t confirm his mail. |  |
|  | 2- System rejects the sign up. |

# Ownership Report

|  |  |
| --- | --- |
| **Item** | **Owners** |
| Use Cases | *Aya-Sara-Amany* |
| Use Case Diagram | *Heba* |
| Requirements | *Amany* |