Cairo University  
Faculty of Computers and Information



**CS251**

**Software Engineering I**

FunlerO

Software Requirements Specifications

Amina Mahmoud

Randa Ayman

Sara Yasser

November, 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** |
| 20140124 | Randa Ayman ElBehery | [Randa.6111@hotmail.com](mailto:Randa.6111@hotmail.com) | 01062756157 |
| 20140079 | Amina Mahmoud Ahmed | [amina.mahmoud282@gmail.com](mailto:amina.mahmoud282@gmail.com) | 01155986956 |
| 20140140 | Sara Yaser AbdElSamie | [sarayasser\_110@yahoo.com](mailto:sarayasser_110@yahoo.com) | 01000284179 |

# 

# Document Purpose and Audience

Purpose:

This document illustrates requirements and system models for the web application

Audience:

Client, CEO, system designer, developer

# 

# Introduction

## Software Purpose

A game based educational web application that teaches students basic concepts of some areas of science through simple games.

## Software Scope

It is a Web based educational application that has 2 types of accounts: teacher and student.

A Teacher can:

* Register to the website as teacher.
* Create games for students.
* Edit or delete games he/she created.
* Reply for students’ comments on games he/she created.
* Try any game on the website.

And Student can:

* Register to the website as student.
* Play/ Rate/ Comment on any game.

# 

# Requirements

## Functional Requirements

Teacher can do the following functions:

* Create new game:
  + Teacher will chose the game category, enter data , save.
* Try any game:
  + Teacher can try any game so that he/she could select appropriate games for the students. Teacher selects the game then plays it.
* Edit any game he/she created before:
  + Teacher selects the game to be edited, confirms the password, apply the edit he/she wants then press save.
* Remove any game he/she created before:
  + Teacher selects the game to be removed, confirms the password then press remove.
* Respond to students’ comments on games he/she created:
  + Teacher is notified when a student comments on a game he/she created then he/she will open the page where the comment is written, and writes his reply.
* Register:
  + Teacher enters his/her information to create new account.

Student can do the following functions:

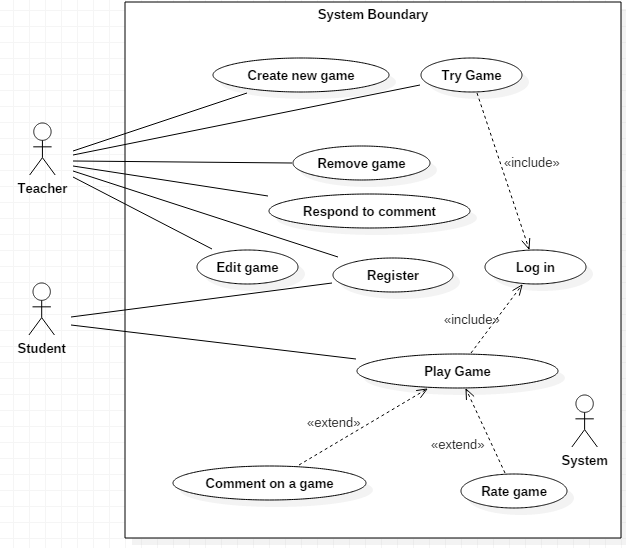
* Play games:
  + Choose the field of science from a list, then choose game category, then select the game.
* Rate games:
  + Go to rating tab in the game page, then select number of stars as rating for the game ( 1 as minimum , 5 as maximum )
* Write comments on the games:
  + Go to comment tab in the game page, write the comment then press Enter.
* Register:
  + Student enters his/her information to create new account.

## Non Functional Requirements

* Usability
  + User friendly interface
* Reliability
  + Auto-Saving for data in case of network failure.
* Security
  + Login with name and password
  + Confirming password when editing or removing games by teacher
* Performance
  + Loading a game doesn’t take more than 8 seconds.
  + System allows up to 1000 users to access the system.
* Supportability
  + If the user forgot his/her password, a new password will be sent to his/her email within 15 seconds.

# System Models

## Use Case Model



## Use Case Tables

|  |  |  |
| --- | --- | --- |
| Use Case ID: | REG | |
| Use Case Name: | Registration | |
| Actors: | Teacher , Student | |
| Pre-conditions: | User wants to have an account, opens the website. | |
| Post-conditions: | User has an account. | |
| Flow of events: | **User Action** | **System Action** |
| 1- Open registration page. |  |
|  | 2-System will show the registration form. |
|  | 3- User will fill in the information required in the form and press register. |  |
|  |  | 4- System creates a new account with the given data. |
| Exceptions: | **User Action** | **System Action** |
| 1-user fill in the information |  |
|  | 2- System check if this email exists if exists it will display “email already used”. |
|  | **User Action** | **System Action** |
|  | 1- User fill in the information |  |
|  |  | 2- System checks if the password is secured enough (has more than 8 characters). If it’s not secured system will display “please make your password greater than 8 characters”. |
| Includes: |  | |
| Notes and Issues: |  | |
|  |  | |
| Use Case ID: | PG-S | |
| Use Case Name: | Play game | |
| Actors: | Students | |
| Pre-conditions: | User login to the system | |
| Post-conditions: | Game over or student end the game | |
| Flow of events: | **User Action** | **System Action** |
| 1- Student selects the field of science |  |
|  | 2- System show category of his/her choice |
| 3-Student chooses the category |  |
|  |  | 4- system load games |
|  | 5- Student selects desired game |  |
|  |  | 6-System load selected game |
|  | 7-Student play game |  |
|  |  | 8-System will calculate student score and save it then ask student if he want to play again. |
| Exceptions: | **User Action** | **System Action** |
|  | 1. User asks for help to play game |  |
|  |  | 1. System will show set of instructions to help student in playing. |
| Includes: | Login | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | RM | |
| Use Case Name: | Remove game | |
| Actors: | Teacher | |
| Pre-conditions: | Log in | |
| Post-conditions: | Remove done | |
| Flow of events: | **User Action** | **System Action** |
| 1- Viewing profile and choose category of the game to be removed |  |
|  | 2-Load games in the selected category. |
| 3- choose the game to be deleted |  |
|  | 4-system asks password confirmation |
| 5- teacher inserts his/her password |  |
|  | 4- System verifies password, removing game, display the message: “removing done”. |
| Exceptions: | **User Action** | **System Action** |
| 1- choose the game to be deleted and delete it |  |
|  | 2- Someone is playing the game.  3- System rejects to remove. |
|  | Teacher inserts his/her password |  |
|  |  | System checks and found password invalid. |
| Includes: |  | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | LGIN | |
| Use Case Name: | Log in | |
| Actors: | Teacher , Student | |
| Pre-conditions: | User wants to play | |
| Post-conditions: | Users already logged in | |
| Flow of events: | **User Action** | **System Action** |
| 1- User enters his/her user name and password |  |
|  | 2-System Verifies user data |
| Exceptions: | **User Action** | **System Action** |
| 1-user enters user name and password |  |
|  | 2-System will show “invalid” message and ask user to try again or to get new password |
|  | 3-user asks for password because he/she forgets it |  |
|  |  | 4- system ask user of his/her email |
|  | 5- User writes his/her email |  |
|  |  | 6-System will check if this email is exists in the database if exist system sends new password  Else system will send invalid mail |
| Includes: |  | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | EG | |
| Use Case Name: | Edit game | |
| Actors: | Teacher | |
| Pre-conditions: | Log in | |
| Post-conditions: | Edit done | |
| Flow of events: | **User Action** | **System Action** |
| 1- Viewing profile and choose category of the game to be edited |  |
|  | 2-load games category |
| 3- choose the game to be edited |  |
|  | 4-System loads game data |
| 5-teacher will edit the game and press edit button |  |
|  | 6- system asks password confirmation |
| 7- teacher insert his/her password |  |
|  | 8- System verifies password, save editing and show a message “editing done”. |
| Exceptions: | **User Action** | **System Action** |
| 1- Select Edit |  |
|  | 2-Someone is playing the game.  3- System rejects to Edit. |
|  | Teacher inserts password |  |
|  |  | System checks password and it’s not valid |
| Includes: |  | |
| Notes and Issues: |  | |
|  |  | |
| Use Case ID: | TG-T | |
| Use Case Name: | Try game | |
| Actors: | Teacher | |
| Pre-conditions: | User logged in to the system | |
| Post-conditions: | Game over or teacher stop the game | |
| Flow of events: | **User Action** | **System Action** |
| 1- Teacher selects the field of science |  |
|  | 2- System shows category of his/her choice |
| 3-Teacher chooses the category |  |
|  |  | 4- system loads games |
|  | 5- Teacher selects desired game |  |
|  |  | 6-System loads selected game |
|  | 7-Teacher plays game |  |
| Exceptions: | **User Action** | **System Action** |
|  |  |
| Includes: | Login | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | AG | |
| Use Case Name: | Add Game | |
| Actors: | Teacher | |
| Pre-conditions: | Log in | |
| Post-conditions: | Adding done | |
| Flow of events: | **User Action** | **System Action** |
| 1- Viewing profile and choose game category to be added |  |
|  | 2-load games category |
| 3- choose the game to be created |  |
|  | 4- Asking for game data. |
| 5-entering game name and data, press save |  |
|  |  | 6- saving data, adding done |
| Exceptions: | **User Action** | **System Action** |
| 1- Entering game name |  |
|  | 2-check game name.  3- System rejects to add , as game name is already existing |
| Includes: |  | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | RG | |
| Use Case Name: | Rate Games | |
| Actors: | Student | |
| Pre-conditions: | Student log in to his/her account. | |
| Post-conditions: | Total rating of the game is updated. | |
| Flow of events: | **User Action** | **System Action** |
| 1. Student selects the game to be rated. |  |
|  | 1. System loads game page |
| 1. Student highlights number of starts to rate the game |  |
|  | 1. System displays the message: “Thanks for your rating”. |
|  | 1. System calculates the new average rating for the game. |
| Exceptions: | **User Action** | **System Action** |
|  |  |
|  |  |
| Includes: |  | |
| Notes and Issues: |  | |
|  |  | |
| Use Case ID: | WCONG | |
| Use Case Name: | Write comment on a game | |
| Actors: | Student | |
| Pre-conditions: | Student log in to his/her account | |
| Post-conditions: | New comment is added to this game’s comments, and teacher is notified. | |
| Flow of events: | **User Action** | **System Action** |
| 1. Student opens the page of the game he/she wants to comment on. |  |
|  | 1. System loads game page |
| 1. Student goes to comments tab on the page and writes the comment. |  |
|  |  | 1. System will add the comment to game’s comments and notify teacher |
| Exceptions: | **User Action** | **System Action** |
|  |  |
| Includes: |  | |
| Notes and Issues: |  | |

# 