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 Science, Math, Engineering and Technology through simple games.

## Software Scope

The software is a web application that facilitates learning through games. Teachers and students can have accounts that give them more privileges.

A teacher can create game, edit, or remove a game he/she created before, or try any other game on the website. A student can play, rate, or comment on any game he/she played. The teacher can also reply to comments that students wrote on games he/she created. The student will be notified by the reply of the teacher.

The games on the website can be included in one of the following categories: MCQ, match, T/F, find the mistake, and run code game. “Run code game” is kind of sorting lines of code, or adding missing part of the code, so that it runs properly. Each game has a certain level of difficulty (Easy, Medium, and Hard).

When a user opens a certain game’s page, he/she can see some snapshots of the game, and tips illustrating the rules to play the game. If this user is a logged-in student he/she will get some coins. These coins can be later redeemed for hints in the games. These hints can show part of the answer, give an extra piece of information that helps in answering certain question, or show the full answer.

When the student passes certain number of levels, or reaches a certain score, he/she will get badges and achievements. Each badge/ achievement will give the student a gift. The gift can be certain number of coins, double the coins he/she has, or a certain type of hint.

A group of students can play a challenge/ competition in a certain game. The winner will be the student who gets the highest score in a fixed amount of time, or who solves a fixed number of questions in shortest time. The winner will be rewarded by a gift as the gifts mentioned before.

There will be leader boards that show the top 25 students in each game, the 25 students who won most competitions, and the 25 students who got largest number of badges.

Users will also be able to search for games by game name, teacher name, game category, field of science, or difficulty.

# Requirements

## Functional Requirements

Teacher can do the following functions:

* Create new game:
  + Game creation is done throwing registration as a teacher, viewing profile and game category to be added then choose to create a new game, chose the game category, game difficulty level (where each game is put under a certain level of difficulty),

System show category of his/her choice, teacher choose one of the built in schema to make a MCQ match, true and false, run code game (is a dynamic game that have a character to move through sorting lines of code, or adding missing part of the code, so that it runs properly) , system will show the game schema and ask for data , teacher enters game data, system will save data and display creation is done

* Try any game:
  + Teacher can try any game so that he/she could select appropriate games for the students. Teacher will register, chooses the game category and game difficulty level, system will load category, select the game (he can see game tips and snapshots), system will load the game, teacher plays it.
* Edit any game he/she created before:
  + Teacher can select the game to be edited any game, so that he/she could select appropriate games for the students, Teacher will register, chooses the game category and game difficulty level, system will load category, select the game to be edited, system will load the game, edit it then confirms his password then press save.
* Remove any game he/she created before:
  + Teacher can remove the game, Teacher will register, viewing profile, chooses the game category, game difficulty level, system will Load games in the selected category, Teacher selects the game to be removed ,system asks password confirmation , teacher inserts his/her password, System verifies password, removing game, display the message: “removing done”.
* Respond to students’ comments on games he/she created:
  + Teacher can respond students’ comments on game, Teacher will register, viewing profile, chooses notifications button (as Teacher is notified when a student comments on a game he/she created) ,system will load notifications list , teacher chooses a specific notifications then system will move hi, to a specific page to reply the comment ,press enter or reply.
* Register:
  + To a Teacher registration, open web application, chooses registration a teacher, system will show him the form, enters his/her information to create new account, System creates a new account with the given data.

Student can do the following functions:

* Play games
  + Student will log in, the student profile will be shown, student will choose the field of science, then he will choose the category depending on the game he/she has chosen, then he/she will play it.
* Rate games:
  + Student must play game before rating, after playing game the student will be asked if he/she wants to rate the game by stars (minimum 1 ,maximum 5 ).
* Write comments on the games:
  + Go to comment tab in the game page, write the comment then press Enter.
* Search:
  + Student will press search button on his/her profile; he/she can search by game category or by field of science, after choose the way of search, student will enter key word to be searched.
* Get into competition with others:
  + In the student profile, he/she can choose to get into competition with others, the competition depends on time and how many correct answers and the winner will take a gift or coins, then leader board will be updated.

Guest can do the following function:

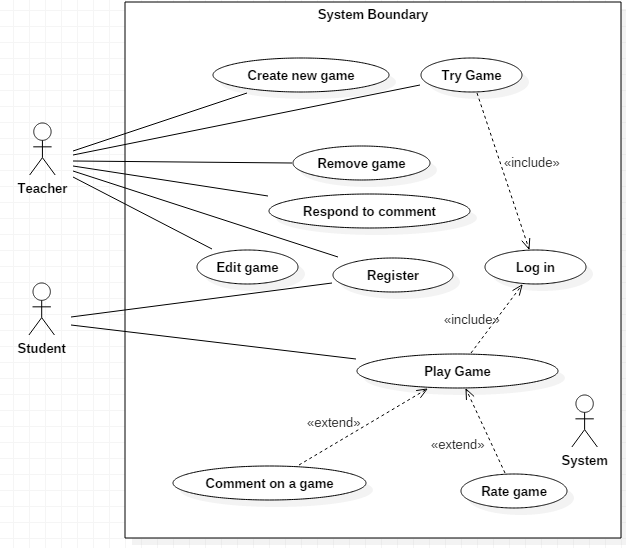
* Register:
  + Any one will be able to register to the web site as a teacher or as a student the user will choose register, he/she will choose if he/she is a student or a teacher then a registration form will be shown, depending on his/her choose.
* Try game as a guest:
  + Any guest can try game but under limitation as he/she can’t play all levels while he/she is just a guest. He/she will choose try game as a guest then choose a field of science then he/she will choose category depending on chosen game then playing game.

## Non Functional Requirements

* Usability
  + Users can try the games without having an account.
  + Users can see snapshots of the game and tips illustrating how to play the game before he/she starts the game.
  + Users can search for specific game, games made by specific teacher, game with specific level of difficulty, game in specific category, or game in specific field of science.
  + The website supports both Arabic and English languages.
* Reliability
  + Auto-Saving for progress in the game in case of network failure.
* Security
  + Login with name and password
  + Confirming password when editing or removing games by teacher
* Performance
  + Loading a page doesn’t take more than 4 seconds.
  + 99.9% uptime ( 8 hours 45 minutes downtime / year)
* Supportability
  + If the user forgot his/her password, a new password will be sent to his/her email within one minute.

# System Models

## Use Case Model



## Use Case Tables

|  |  |  |
| --- | --- | --- |
| Use Case ID: | REG | |
| Use Case Name: | Registration | |
| Actors: | Guest | |
| 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 make the guest to choose if he/she is student or a teacher |
|  | 3-Guest will choose |  |
|  |  | 4-System will display a form |
|  | 5-User will fill in the information required in the form and press register |  |
|  |  | 6-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: | TAG | |
| Use Case Name: | Try game as a guest | |
| Actors: | Guest | |
| Pre-conditions: | Open web site | |
| Post-conditions: | Guest exit the web | |
| Flow of events: | **User Action** | **System Action** |
| 1-guest will open the web site |  |
|  | 2-System loads the home page |
| 3-guest chooses to play as a guest |  |
|  | 4- system displays games |
| 5-guest chooses the desired game |  |
|  |  | 6-system loads game |
| Exceptions: | **User Action** | **System Action** |
| 1-guest chooses desired game |  |
|  | 2- System can’t load this game for a guest so, the guest will be asked to register before playing this game. |
| 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: | GC | |
| Use Case Name: | Get into competition with others | |
| Actors: | Student | |
| Pre-conditions: | Log in | |
| Post-conditions: | Leader board is updated | |
| Flow of  events: | **User Action** | **System Action** |
| 1- student chooses to get into competition |  |
|  | 2-system loads games |
| 3-student will choose a game (descried above) |  |
|  | 4- system loads game and displays competitors names |
| 5-students begin game together |  |
|  |  | 6-after game ends the system will determine which student is the winner through calculation  7-winner will take a gift and system will display a message for losers to try again.  8- Leader board updated. |
| Exceptions: | **User Action** | **System Action** |
|  |  |
|  |  |
| Includes: |  | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | SG | |
| Use Case Name: | Search | |
| Actors: | Teacher, Student | |
| Pre-conditions: | Log in | |
| Post-conditions: | Result is shown | |
| Flow of events: | **User Action** | **System Action** |
| 1-user press Search button |  |
|  | 2-system will ask user how he/she would like to search by any category |
| 3- user will choose the category |  |
|  | 4- search bar will be displayed to user |
| 5-user writes the key word |  |
|  |  | 6-System searching, results will be displayed to user |
| Exceptions: | **User Action** | **System Action** |
| 1- user writes key word |  |
|  | 2-System searching, no result found  3-System display message “not found try to search by another key word”. |
| Includes: |  | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| 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: |  | |

|  |  |  |
| --- | --- | --- |
| 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: | 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: | 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 and difficulty level 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 |  |
|  | 6- 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: | 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 and it's difficulty level |  |
|  | 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: | Create 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 (from built in schema) |  |
|  | 4- Loading schema and asking for game data. |
| 5-entering game name and data, press next |  |
|  |  | 6- saving data , display "creation is 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: | RSCOM | |
| Use Case Name: | Respond to comment | |
| Actors: | Teacher | |
| Pre-conditions: | Teacher log in to his/her account | |
| Post-conditions: | New reply to a comment is added to the game’s comments, and Student is notified. | |
| Flow of events: | **User Action** | **System Action** |
| 1- Viewing profile and chooses notifications button |  |
|  | 2-System loads notifications |
| 1. Teacher choose a specific notification |  |
|  |  | 4-system will move him to a specific page to replay the comment |
|  | 5- Teacher goes to comments tab on the page and writes the comment. |  |
|  |  | 6- System will add the comment to game’s comments and notify student |
| Exceptions: | **User Action** | **System Action** |
|  |  |
| Includes: |  | |
| Notes and Issues: |  | |

# Ownership Report

|  |  |
| --- | --- |
| **Item** | **Owners** |
| Document Purpose and Audience | *The whole team* |
| Software Purpose, Software Scope | *Randa Ayman* |
| Definitions, acronyms, and abbreviations | *Sara Yaser* |
| Functional Requirements | *The Whole team* |
| Non Functional Requirements | *The Whole team* |
| Use Case Model | *Amina Mahmoud* |
| Use Case Table | *The Whole team* |

# GitHub link

<https://github.com/RandaAymanAhmedElBehery/SoftwareEngineering1Project>