Cairo University  
Faculty of Computers and Information



**CS251**

**Software Engineering I**

Project Name

Software Requirements Specifications

Team Names

Month & Year

Contents

[Instructions [To be removed] 3](#_Toc402452668)

[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)

[Policy Regarding Plagiarism: 6](#_Toc402452682)

# Team

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Email** | **Mobile** |
| 20140097 | Hazem Mohamed Taha Fouda | hazem.fouda1@yahoo.com | 01211122871 |
| 20140293 | Noura Mohi El deen | [nouramohi712@gmail.com](mailto:nouramohi712@gmail.com) | 01005016518 |
| 20140298 | Nourhan Mahmoud Zaki | nourzaki596@gmail.com | 01152275725 |

# Document Purpose and Audience

Document Purpose: This document describes our website requirements for the creation of online educational games

Document Audience : TAs & doctors as Clients , Developers , Designers & Project Manager.

# Introduction

## Software Purpose

* Our purpose is to allow Student to find available resources of gaining more knowledge & having fun without wasting any of their time & enhance skills of students.

## Software Scope

Our software will be website , this web site divided into 3 sections ..

First section will be the user profile , it will contain his information ,profile picture and his score and the competitions which the user entered

Second section will be the gaming section in that section user can play all games in all categories

As games divided into categories such as science , math …

Third section will be the competitions , User can Enter competition between another users

1. Implementation will be handled in java for games to be user friendly.
2. Creation of website will be using HTML , CSS & java script.
3. Web application with sign in & sign up option.
4. Tutorials used for clarity of website usage.
5. Access user as Teacher/Student.
6. Group games into categories.
7. Rating & invitation options exist.
8. Exiting game & logging out of website.

## Definitions, acronyms, and abbreviations

* Abbreviations & Definitions :

Student : The target person who gains the applied knowledge.

Teacher : Organizer of a specific topic who has the access to apply new thoughts into live events.

Educational game : Game provides the student who plays it with educational concepts.

Tutorial video : A simple description of how to use the website efficiently.

Competitions : Challenges occur between students of high scores with final prizes.

Feedback : Student gives comment about the game he played & gives new ideas about the development of this game.

# Requirements

## Functional Requirements

Website user should be able to :

* Sign up (Complexity 2)

-sign up with mail , password ,username(unique), age (to clarify the best sequence of games to him ) , gender , education:

-When the user writes the URL of the website , the first page appears in front of him will be our log in at the center of the page with 2 options sign up & sign in.

-Sign up First of all , the user’s email must be valid & still active , password must be >= 10 characters of both capital & small letters , age of the user must be declared to get the suitable games for his age , also education must be declared to get categories of games near his career, then the system will send activation URL to the user’s mail to be sure that the user used his real account.

- choosing whether he Student or Teacher.

* Having User Profile (Complexity 3)

For each user he has an individual profile having his username , high scores , personal photo , his favorite games.

* Sign in as Student /Teacher.(Complexity 1)

This should be implemented for the user to choose what changes he can make to the site.

* Watch video tutorial (Complexity 1)

Our second page will include a video tutorial for description of the website.

* Creating & modifying Games.(Complexity 5)

The only who can create games is the teacher , He can’t only modify on them but also can remove them as long as he is the creator.

* Creating & Sorting Subcategories.

Games are categorized into Science , Math & Technologies & sorted into subcategories according to alphabetically , difficulty & age.

* Play game (Complexity 3)

-Selecting category of games.

-Choose game.

-Tutorials for each game :

Before the student starts any game for the first time , there would be a tutorial explaining how it’s played

-Score stored for each user.

Every time the user opens the game , his high score is stored for him + his username.

* Rating Games.(Complexity 2)

Also their must be an option for rating each game according to its creativity percentage and Students give feedback about each game they play which helps for future modifications.

* Inviting people(Complexity 2)

Each user can invite whoever he wants by his username & challenge him in a specific game.

* Exiting Game(Complexity 1)

User ends the game with storing his score & level by his username.

* Entering Competitions(Complexity 4)

Competition between users according to their high scores.

* Logging out from the whole website.(Complexity 1)

User should have the access to logging out like logging in.

Each user Student/Teacher have the log out option from his username.

## Non Functional Requirements

|  |  |
| --- | --- |
|  | **Details** |
| Usability | * Our web application is User friendly . |
| Security | * Checking for password accuracy & Sending check mails for activating new accounts of our website. |
| Simplicity | * User Interface & user interaction make the website easy to use. |
| Availability | * Website is available for all platforms browsers. |
| Manageability & Serviceability | * Must be implemented in java to gain a suitable user interface & to encourage clients to use it. |
| Flexibility | * Suitable for all ages having categorized games suitable for each age   , Sign in option will be available if user already has an account , he also can sign in with fb email , Instagram or twitter. |
| Recoverability | * Must have help Button in case of having problems & need urgent fixes to website. |
| Reliability | * Games are sorted according to their difficulty , alphabetically & age. |

# System Models

## Use Case Model



## Use Case Tables

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 1 | |
| Use Case Name: | Creating & modifying games. | |
| Actors: | Teacher , Student, Administrator. | |
| Pre-conditions: | Creator must have an account as a teacher. | |
| Post-conditions: | -Creator can modify according to Student’s comments  -Creator can reply on Student’s comments | |
| Flow of events: | **User Action** | **System Action** |
| 1.Teacher can create game & uploads it. |  |
|  | 2.System approves upload. |
| 3.Student can write comment or feedback. |  |
|  | 4.System saves data in database & sends to creator. |
| 5.creator checks those feedbacks. |  |
|  |  | 6.System removes the checked ones to done. |
|  | 7.creator modifies game & uploads it. |  |
|  |  | 8.system approves update. |
| Exceptions: | **User Action** | **System Action** |
| 1. Student can’t write comment |  |
|  | 2.System is down. |
|  | 3.Student asks to contact administrator’s. |  |
|  |  | 4.System sends data to student. |
|  | 1.Teacher sends request to admin to upload a game. |  |
|  |  | 2.System cancels upload. |
| Includes: | Teacher sends request to admin to allow him accept modifying a game.  Administrator sends rate and feedback to Teacher. | |
| Notes and Issues: | User may not be satisfied if modification is not done. | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 2 | |
| Use Case Name: | Sign up | |
| Actors: | Students , teachers | |
| Pre-conditions: | Open web site page  User must have no past account on this website. | |
| Post-conditions: | User will have account on the web site | |
| Flow of events: | **User Action** | **System Action** |
| 1- User writes URL of website page and chooses to sign up. |  |
|  | 2-System opens a page asking him to write information like gander, mail, user name , age ,password ,etc.. |
| 3- User types the information and presses save button. |  |
|  | 4- System verifies all information & stores it in database.  5- after checking on the information , System will show the user his user name and password |
|  |  |
| Exceptions: | **User Action** | **System Action** |
|  | 1.User types password less than 10 characters |  |
|  |  | 2.System asks the user to type valid password |
|  | 1.User types duplicate user name |  |
|  |  | 2. System asks the user to type another username. |
|  | 1.User type invalid mail. |  |
|  |  | 2.System ask the user to type valid mail. |
|  | 3.User writes another mail. |  |
|  |  | 4.System checks validity of mail. |
|  | 5.User enters rest of information. |  |
|  |  | 6.System sends invitation to the valid mail. |
|  | 1.User doesn’t type any mandatory attributes |  |
|  |  | 2. System asks the user to remove these attribute. |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 3 | |
| Use Case Name: | Having User Profile | |
| Actors: | Student , teacher , Administrator | |
| Pre-conditions: | Sign up | |
| Post-conditions: | User has his private page with his info | |
| Flow of events: | **User Action** | **System Action** |
| 1- User signs up for the first time |  |
|  | 2- System creates a homepage for him with his username. |
| 3- User changes any of his data |  |
|  | 4. System saves change. |
|  |  |
|  |  |  |
| Exceptions: | **User Action** | **System Action** |
|  |  |
|  |  |
| Includes: | Administrator allows user to update his info . | |
| Notes and Issues: | User can change his name , password , photo or age. | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 4 | |
| Use Case Name: | Sign in | |
| Actors: | Student, Teacher | |
| Pre-conditions: | Sign up | |
| Post-conditions: | User opens his account | |
| Flow of events: | **User Action** | **System Action** |
| 1- User opens website page and selects sign in |  |
|  | 2- System asks him to write his username and password |
| 3- User types his name and password |  |
|  | 4- System verifies his username and password |
|  |  |
| Exceptions: | **User Action** | **System Action** |
|  | 1.User types invalid mail or password |  |
|  |  | 2.System will shows the user that password and mail is not valid & allows him to enter another mail or password. |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 5 | |
| Use Case Name: | Choosing a game | |
| Actors: | Student | |
| Pre-conditions: | Sign in | |
| Post-conditions: | User can play the chosen game | |
| Flow of events: | **User Action** | **System Action** |
| 1- User selects game category |  |
|  | 2- System shows the user the available games in that category |
| 3- User Selects a game from that category |  |
|  | 4- System opens the game the user wants. |
|  |  |
| Exceptions: | **User Action** | **System Action** |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 6 | |
| Use Case Name: | Rating a game | |
| Actors: | Students | |
| Pre-conditions: | Playing that game | |
| Post-conditions: | System saves his rate. | |
| Flow of events: | **User Action** | **System Action** |
| 1. User has just finished the game. |  |
|  | 2- System asks the user to rate the game and if he wants to write a comment & cancel if he doesn’t want. |
| 1. User rates the game and writes a comment. |  |
|  | 4- System saves the rate and the comment for that game |
|  |  |
| Exceptions: | **User Action** | **System Action** |
|  | 1.User refuses to rate |  |
|  |  | 2.Close rate option |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 7 | |
| Use Case Name: | Playing game | |
| Actors: | Student , teacher | |
| Pre-conditions: | Choosing a game | |
| Post-conditions: | Rating game and save score & level | |
| Flow of events: | **User Action** | **System Action** |
| 1- User selects a game to play |  |
|  | 2- System running the game |
| 3- User playing the game |  |
|  |  |
|
| Exceptions: | **User Action** | **System Action** |
| 1.User closes the website during running of the game . |  |
|  | 2.System collects score and saves it. |
| 1.Crash occurs |  |
|  |  | 1.System saves score. |
| Includes: |  | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 8 | |
| Use Case Name: | Inviting people | |
| Actors: | Student , teacher | |
| Pre-conditions: | Sign in | |
| Post-conditions: | User can send invitation to people | |
| Flow of events: | **User Action** | **System Action** |
| 1- User clicks to send invitation . |  |
|  | 2- System asks him to choose friend already have an account |
| 3- User chooses a friend |  |
|  | 4.1- if the friend has an account System will send him invitation to him  4.2 if the friend hasn’t an account system will ask for his email |
| 5.1 User types a username for his friend  5.2 User type an email |  |
|  |  | 6.2 system will send invitation. |
| Exceptions: | **User Action** | **System Action** |
| 1.Invitation sent to mail is invalid |  |
|  | 2.Inform user by this on his mail. |
| Includes: | Checking mail | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 9 | |
| Use Case Name: | Entering competition | |
| Actors: | Students | |
| Pre-conditions: | Sign in  Students involved must have high scores on the same game. | |
| Post-conditions: | Be a member in the competition.  The winning student will get a reward. | |
| Flow of events: | **User Action** | **System Action** |
| 1- User selects competition tab |  |
|  | 2- System shows him the list of competitions |
| 3- User selects competition |  |
|  | 4. System checks for his score if it valid or not  4.1- if it’s valid , System shows time & members of the competition  4.2. if not , System will show him a massage to inform the user that his score is not valid |
| 6-if user has valid score, ask him to enter competition. |  |
|  | 7-System adds the user in the competition and increments number of members in competition by one. |
|  |  |
| Exceptions: | **User Action** | **System Action** |
| User cancels competition request. | System will remove the user from competition and decrements the number of members in the competition by one. |
| 1. there is a tie between 2 students on the final contest. | . |
|  |  | 2.System makes complex second round of competition. |
|  | 3.They choose to play or give up. |  |
|  |  | 4.Quiter is the loser. |
| Includes: | Users only with high scores can play | |
| Notes and Issues: | Competition helps them get involved more & more into the website & get the best knowledge from it. | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 10 | |
| Use Case Name: | Creating & Sorting subcategories | |
| Actors: | Teacher , Student , Administrator. | |
| Pre-conditions: | User has an active profile. | |
| Post-conditions: | User chooses a specific game. | |
| Flow of events: | **User Action** | **System Action** |
| 1.User chooses a specific category. |  |
|  | 2.System shows him subcategories. |
| 3.User chooses a way to sort games. |  |
|  | 4.System allows sorting. |
| 5.user chooses a specific game. |  |
|  |  | 6.System opens the game. |
| Exceptions: | **User Action** | **System Action** |
|  |  |
|  |  |
| Includes: |  | |
| Notes and Issues: | User can sort subcategories by age , alphabetically or according to difficulty. | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 11 | |
| Use Case Name: | Logging out. | |
| Actors: | Student , Teacher. | |
| Pre-conditions: | user exits game. | |
| Post-conditions: | website is available for another user to use. | |
| Flow of events: | **User Action** | **System Action** |
| 1.user asks to log out. |  |
|  | 2.System accepts and closes his homepage. |
|  |  |
|  |  |
|  |  |
| Exceptions: | **User Action** | **System Action** |
|  |  |
| Includes: |  | |
| Notes and Issues: | Log out will be from any page from settings in each page. | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 12 | |
| Use Case Name: | Exiting Game. | |
| Actors: | Student , Teacher. | |
| Pre-conditions: | The last update of user score must be saved. | |
| Post-conditions: | User will start from the same score & the same level afterwards. | |
| Flow of Events: | **User Action** | **System Action** |
| 1.User exits the game. |  |
|  | 2.System saves his score & level to his username. |
|
|
|
| Exceptions: | **User Action** | **System Action** |
|  |  |
|  |  |
| Includes: | -User starts from the same level he left with the same score  -System saves each score one time after another. | |
| Notes and Issues: | Game will save details of each user with username & score in history in database. | |

|  |  |
| --- | --- |
| **Item** | **Owners** |
| functional requirements , use case diagram , definitions , software purpose , abbreviation , software scope | *Hazem Mohamed Taha* |
| Functional requirements , introduction , Software audience , document purpose , use case table -> teacher , software scope | *Noura Mohi* |
| use case table -> student , non functional requirements ,software scope | *Nourhan Mahmoud Zaki* |

.