Cairo University  
Faculty of Computers and Information



**CS251**

**Software Engineering I**

Software Requirements Specifications

November, 2016

Contents

[**Team** 4](#_Toc466118258)

[**Document Purpose and Audience** 4](#_Toc466118259)

[ **Purpose:** 4](#_Toc466118260)

[ **Audience:** 4](#_Toc466118261)

[**Introduction** 5](#_Toc466118262)

[**Software Purpose** 5](#_Toc466118263)

[**Software scope** 5](#_Toc466118264)

[**Definitions, acronyms, and abbreviations** 6](#_Toc466118265)

[**Requirements** 7](#_Toc466118266)

[ **Functional Requirements** 7](#_Toc466118267)

[1.Sign up 7](#_Toc466118268)

[2.Check emails’ pattern 7](#_Toc466118269)

[3.log in. 7](#_Toc466118270)

[4.Check validation 7](#_Toc466118271)

[5.Forget password 7](#_Toc466118272)

[6.Update profile 7](#_Toc466118273)

[7.Play game 7](#_Toc466118274)

[8.Write comment 8](#_Toc466118275)

[9.Reply on a comment 8](#_Toc466118276)

[10.Recap………………………………………………………………………………………………………………………..8](#_Toc466118277)

[11.learned points 8](#_Toc466118278)

[12.Search for game 8](#_Toc466118279)

[13.Add game 8](#_Toc466118280)

[14.Edit game 8](#_Toc466118281)

[15.Remove game 8](#_Toc466118282)

[16.Check students’ score 8](#_Toc466118283)

[17.Save scores 8](#_Toc466118284)

[18.Help….………………………………………………………………………………………………………………………9](#_Toc466118285)

[19.Download 9](#_Toc466118286)

[20.Sort games 9](#_Toc466118287)

[21.Give reward 9](#_Toc466118288)

[22.Send notification 9](#_Toc466118289)

[23.Rate game 9](#_Toc466118290)

[ **Non Functional Requirements** 9](#_Toc466118291)

[1. Usability 9](#_Toc466118292)

2.Availability .9

[3. Reliability 9](#_Toc466118293)

[4. Performance 10](#_Toc466118294)

5.Supportability.…………………………………………………………………………………………………………..10

[6. Implementation 10](#_Toc466118295)

[7.User Friendly 10](#_Toc466118296)

[8. Packaging 10](#_Toc466118297)

[9. Security 10](#_Toc466118298)

[10. Maintainability 11](#_Toc466118299)

[**System Models** 12](#_Toc466118300)

[ **Use Case Model** 12](#_Toc466118301)

[ **Use Case Tables** 13](#_Toc466118302)

[**Ownership Report** 25](#_Toc466118303)

# **Team**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Email** | **Mobile** |
| 20140290 | Nagham Alaa Taha | naghammalaa@gmail.com | 01124457494 |
| 20140078 | Amina Gaber Abd ELNaby | aminagaber8@gmail.com | 01140980364 |
| 20140136 | Sara Tarek Nada Habib | Sarahtarek1996@gmail.com | 01118622842 |
| 20140143 | Salma Ahmed Abd El Ghafar | salmashehabeldin@gmail.com | 01062986954 |

# **Document Purpose and Audience**

## **Purpose:**

This document will explain in details all of the needed functions and requirements to design and develop the system. This document should be readable by all of team members whom will be assigned to develop such a system.

## **Audience:**

The audience who are expected to read it are the members of the team who will develop this system. Such document should be read by the Project manager, Team leader and its members.

# **Introduction**

There has been many ways of education to enable the students to develop their ways of thinking, through past of the centenaries, education has been about presenting information in a traditional way as on a paper or in books then it developed a little bit more in the form of website’s search.

After the highly wide improvements that occurred in both the Technologies and the ways of Education, it has opened a lot of new doors and ideas to develop a new system for education around the whole world.

This new system will present Education in its new form which will bring both Entertainment and Knowledge to its users in a form of a game. This game will be divided into more than one categories which we will explain its specific details throughout the document.

# **Software Purpose**

The purpose of the system’s software is to enable its users whom will be the students to have a new way of education. There will also be teachers as a user in our system. The teachers will be able to maintain the flow of the game. They will also have the ability to answer any questions that are going to be asked by the students in a comment under each game. This kind of system should be interested to most of the students as they will be able to have more knowledge by un-traditional way.

# **Software scope**

This system is composed for 2 Actors: Teacher user, Student user and 4 main components: learning system, playing games, Notifications, and Lists.

This system contains 4 categories: Math, Science, and Programming .Each category must contain a list of games. The system enables the users to learn new concepts in each category by playing a game related to this category. If the user a student. Then he can add a comment and rate this game. Notification is send to the teacher who added this game with the Student’s comment to reply to him then notification is send to the student to see the reply.

# **Definitions, acronyms, and abbreviations**

|  |  |
| --- | --- |
| **Term** | **Definition** |
| 1. GLS | Games learning society the name of our system. |
| 1. Category | Each game is added to a specified category. Category is the type of science this game belongs to e.g.: math, programming, English, Science,…etc. |
| 1. Concepts | The points learned from each game for example if a game related to the category of programming after playing this game student should have been learned many concepts like conditions or loops and so on… |
| 1. Reward | A certificate a teacher can send to a student if he finished playing all the games in one category.  Student can gain only one certificate for each category. |
| 1. Recap | This feature allow the student when entering any category to know what exactly concepts he learned from this category according to games he played before in this category. |
| 1. Rate | It is a feature a student can give a game a rate from 1 to 5 according to the game performance. |
| 1. Notification | Like a message send to the teacher if a student commented on the game added by him and also is sent to the student if the teacher reply on him. |
| 1. Saved played games list | A list created for each student contain the name of the games he played before in the category he chose to play in and contain the main concepts that should have been learned from the game he played. |
| 1. Saved scores list | A list for each game contain the score gained by students who played this game teacher is allowed to see the list of the game added by him in order to enhance it. |

# **Requirements**

## **Functional Requirements**

1. Sign up -> this function allows the user to create his own account on the website by asking him/her about his/her information [e.g. Name-age-gender-email-password-phone Number-position{teacher or student}-and level{for student} ]

* **Include:** this function includes Check emails’ pattern function

1. Check emails’ pattern -> this function makes sure that the email that the new user entered is correct and was written in the required pattern [e.g. @gmai.com- @yahoo.com and so on] and make Sure that this email is new and doesn’t exist in the database. If the email wasn’t in the required pattern this function will print a massage that asking for a new email.
2. log in -> this function allows the user to login to the website and open his own profile

* **Include:**  this function includes check validation function

1. Check validation -> this function checks the validation of the user account[email and password] by searching for the entered email on the websites’ database and let the user login to website if it found this email or ask him to recheck the entered data if it didn’t find it or ask the user if he forget his password

* **Extend:** this function may call the forget password function

1. Forget password -> this functions helps the user to get back his password if he forgot it by sending him an email with a link that he can through it update his password
2. Update profile -> This Function allows the user to update his information saved in the database except his email he isn’t allowed to update it.
3. Play game -> this function allows the user to play a game according to the category that he chosen even the user was a teacher or a student but In case of the user was a student after playing the game the student’s score in this game will be increased and saved in the scores list

* **Extend:**
* this function may call the save scores function to save the student’s score
* also may call the rate function
* may call write comment function

1. Write comment -> this function allows to add a comment on any game and ask about anything related to this game after or before he play it
2. Reply on a comment-> this function allows the teacher to answer any question which was written on a comment by a student
3. Recap -> this function allows the user to show all points that he was learned from any game that he played before

* **include:** this function will call the learned points function

1. learned points -> this function determine all the points that the student should learn after he play any game according to its category
2. Search for game -> this function helps the user to find a game that he want by write the concept that he want to learn from this game

* **include:** this function will call learned points function

1. Add game -> this function is allowed for the teacher as it makes him add games to any category that he want to add on it
2. Edit game -> this function is allowed for the teacher as it makes him edit the game that he want by choose its category then choose the wanted game and edit it then this function replace the edited game with the old one
3. Remove game -> this function is allowed for the teacher as it makes him remove any game that he want by choose first the category then the game and the game will be removed from its category
4. Check students’ score -> this function allows teacher to preview the students’ scores on any game that he want to check the level of this game and test if it is need any new features

* **include:** this function will call the save scores function

1. Save scores -> this function save the students’ scores in each game in case of the teacher want to check them any time
2. Help -> this function helps students to know how they can play any game that they want
3. Download -> this function allows the user to download any game that he want on his own PC and play it at anytime
4. Sort games -> this function allow the user to sort the games when entering any category according to most played or earlier date it launched on.
5. Give reward -> if a student finished all games in any category this function allows the teacher to send a reward (certificate) to this student and save that this student gain a reward in this category.
6. Send notification -> when a user comment on a game this function send notification to the teacher who added this game telling him that someone commented on his game.
7. Rate game -> this function allow the user to give a rate to any game from 1 to 5.

## **Non Functional Requirements**

1. Usability:   
 **This system should be simple to register:** The registration rules mustn't be complex.   
 **Simple to sign in:** The login rules just include the username and the password.  
 **Simple to update:** users can change any of their information.   
 **Simple to sign out:** users can stop the application just by clicking a button.   
 **The navigation is efficient:** users can make search effectively and easily.

### 2. Availability:

Users can use this application 24 hours per day.   
3. Reliability:

**Number of users:** millions of users may use this application at the same time.

4. Performance:   
 **Short response time:** The loading time of the application must be smaller than 10 seconds. All other response times must be below 2 seconds.   
 **Storage:** using very huge memory for saving all users' data.  
  
5. Supportability:   
 The system should provide multi language support.   
 The code developed for this system should be open source.

6. Implementation:

This Application will be implemented in Java.

7. User friendly:   
 **Simple user interface:** The user interface of the application should be understandable to the user on the first view and fulfill ergonomic requirements.

The user interface is based on a main window, which includes requirements for either registration or login.   
 The system should provide a uniform look.

8. Packaging:

The executable Application must be available over the web within one download.

9. Security:

The server on which the Application resides will have its own security to save the information of users from hacking.   
 The system should use secure sockets in all transactions that include any confidential customer information.   
 The system should not leave any cookies on the customer’s computer containing the user’s password.   
   
 The customer’s web browser should never display a customer’s password.   
 When the user enters an incorrect password several times or login from a different browser, the application sends a verification message to user’s mobile number or email address.

10. Maintainability:   
 **Mean Time to Repair:** it is the amount of time required to repair a system and bring it back online, so the system allowed being out of operation maximum 2 hours after it has failed.   
 **Mean Time to Failure:** it is the time that a system is not failed, or is available and it's about one  
year.

# **System Models**

## **Use Case Model**

## **Use Case Tables**

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 1 | |
| Use Case Name: | Write a comment | |
| Actors: | User(student) | |
| Pre-conditions: | Person logged in successfully | |
| Post-conditions: | * Comment posted successfully on a game. * Number of comments on this game increased. | |
| Flow of events: | **User Action** | **System Action** |
| 1- User writes a comment on a game. |  |
|  | 2- System post the comment on this game. |
|  | 3-Increase the number of comments on this game. |
|  |  | 4-send notification to the teacher who added this game with the comment added if the user is a student or another teacher. |
| Exceptions: | **User Action** | **System Action** |
| NULL. |  |
| Includes: | This Function includes send notification function. | |
| Notes and Issues: | NULL. | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 2 | |
| Use Case Name: | Add game | |
| Actors: | User(teacher) | |
| Pre-conditions: | Teacher logged in successfully and choose the category he wants to add the game to. | |
| Post-conditions: | New game is added successfully to the chosen category. | |
| Flow of events: | **User Action** | **System Action** |
| 1- User choose the category he wants to add the game to. |  |
|  | 2- System display the chosen category. |
| 3-User adds his game. |  |
|  |  | 4-system post the game to the category. |
| Exceptions: | **User Action** | **System Action** |
| NULL. |  |
| Includes: | NULL. | |
| Notes and Issues: | The teacher is the only User who can add a game. | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 3 | |
| Use Case Name: | Recap | |
| Actors: | User(student) | |
| Pre-conditions: | Student logged in successfully and chose the category he wants to play in. | |
| Post-conditions: | Student knew the concepts he learned in this category. | |
| Flow of events: | **User Action** | **System Action** |
| 1- Student choose the category. |  |
|  | 2-System display the saved played games list for this Category. |
| Exceptions: | **User Action** | **System Action** |
| NULL. |  |
| Includes: | This function will include learned points function | |
| Notes and Issues: | NULL. | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 4 | |
| Use Case Name: | Edit a game | |
| Actors: | User(teacher) | |
| Pre-conditions: | Teacher logged in successfully | |
| Post-conditions: | The game has edited with new features | |
| Flow of events: | **User Action** | **System Action** |
| 1- Teacher choose the category then choose from it the game that he wants to edit. |  |
| 2-teacher start to edit the chosen game and then press save. |  |
|  | 3-system will save what the teacher edited on the game |
|  |  | 4-system will replace the edited game with the old one |
| Exceptions: | **User Action** | **System Action** |
| 1-teacher forget to press save and return to the previous page. |  |
|  |  | 2-system will not edit the game. |
| Includes: | NULL | |
| Notes and Issues: | The teacher is the only User that he can edit a game | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 5 | |
| Use Case Name: | remove a game | |
| Actors: | User(teacher) | |
| Pre-conditions: | Teacher logged in successfully | |
| Post-conditions: | The game has removed from its category | |
| Flow of events: | **User Action** | **System Action** |
| 1- teacher choose the category then choose from it the game that he want to remove |  |
| 2-teacher remove the chosen game and press save. |  |
|  | 3-system will remove this game from its category |
| Exceptions: | **User Action** | **System Action** |
| 1-teacher forget to press save and return to the previous page. |  |
|  |  | 2-System will not remove the chosen game. |
| Includes: | NULL | |
| Notes and Issues: | The teacher is the only User that he can remove a game | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 6 | |
| Use Case Name: | Check students’ score | |
| Actors: | User(teacher) | |
| Pre-conditions: | Teacher logged in successfully | |
| Post-conditions: | Teacher has information about the level of each game (easy-medium-hard) | |
| Flow of events: | **User Action** | **System Action** |
| 1- teacher choose the game that he want to checks its scores |  |
|  | 2-system will display list of students‘ scores of this game |
| 3-teacher show this list and can determine the level of this game |  |
| Exceptions: | **User Action** | **System Action** |
| NULL. |  |
| Includes: | The save scores function | |
| Notes and Issues: | The teacher is the only User that he can check the students’ scores | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 7 | |
| Use Case Name: | Reply on a comment | |
| Actors: | User(teacher) | |
| Pre-conditions: | Student makes a comment on a game | |
| Post-conditions: | * The reply posted successfully on the comment on the game. * Number of comments on this game increased. | |
| Flow of events: | **User Action** | **System Action** |
| 1- Teacher will mark the comment that he wants to reply on. |  |
|  | 2- System will post the reply on the marked comment |
|  | 3-Increse the number of comments on the game |
|  |  | 4-send notification to the student who added this comment with the reply added. |
| Exceptions: | **User Action** | **System Action** |
| NULL. |  |
| Includes: | This Function includes send notification function. | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 8 | |
| Use Case Name: | Play a game | |
| Actors: | User(Both teacher and student) | |
| Pre-conditions: | User must login successfully | |
| Post-conditions: | Played the game successfully and his score was increased to this game | |
| Flow of events: | **User Action** | **System Action** |
| 1- User choose the game he wants to play |  |
|  | 2- System will open the game chosen by the user |
| 3-User will begin to play the game according to the instructions of the game |  |
|  |  | 4-System will manage the score and game’s instructions |
| Exceptions: | **User Action** | **System Action** |
| 1-The user could comment or give a rate on the game |  |
|  |  | 2-System may use the save scores function if the user was a student to save his score |
| Includes: | NULL. | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 9 | |
| Use Case Name: | Help | |
| Actors: | User(student) | |
| Pre-conditions: | Student logged in successfully and chose the game he wants to play. | |
| Post-conditions: | Student will play the game under the instructions that he had read | |
| Flow of events: | **User Action** | **System Action** |
| 1- Student will choose the game he wants to know the instructions about. |  |
|  | 2-System display the instructions for the chosen game |
| Exceptions: | **User Action** | **System Action** |
| NULL. |  |
| Includes: | NULL | |
| Notes and Issues: | NULL. | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 10 | |
| Use Case Name: | Download Game | |
| Actors: | User(teacher, student) | |
| Pre-conditions: | User logged in successfully | |
| Post-conditions: | Downloaded the game successfully | |
| Flow of events: | **User Action** | **System Action** |
| 1- User will choose the game he wants to download |  |
|  | 2- System download the chosen game. |
| Exceptions: | **User Action** | **System Action** |
| NULL. |  |
| Includes: | NULL. | |
| Notes and Issues: | NULL. | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 11 | |
| Use Case Name: | Search for game | |
| Actors: | User(teacher, student) | |
| Pre-conditions: | User logged in successfully | |
| Post-conditions: | User found the game easily and successfully | |
| Flow of events: | **User Action** | **System Action** |
| 1- User will choose the concept he wants to search about. |  |
|  | 2- System display all games related to this concept. |
| 3- User will choose the game he wants. |  |
|  |  | 4-System will manage the game’s instructions. |
| Exceptions: | **User Action** | **System Action** |
| NULL. |  |
| Includes: | Learned points function | |
| Notes and Issues: | NULL. | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 12 | |
| Use Case Name: | Reward | |
| Actors: | System | |
| Pre-conditions: | * Student logged in successfully * Student finished all games in a category | |
| Post-conditions: | User(student) finished category and rewarded about it | |
| Flow of events: | **User Action** | **System Action** |
| 1- User plays all games in a category. |  |
|  | 2- System will save his all scores about this games. |
|  | 3-System will reward the student if he exceeded the minimum determined score. |
| Exceptions: | **User Action** | **System Action** |
| NULL. |  |
| Includes: | This Function includes play a game function. | |
| Notes and Issues: | NULL. | |

# 

# **Ownership Report**

|  |  |
| --- | --- |
| **Item** | **Owners** |
| Document Purpose and Audience | Sara Tarek. |
| Introduction | Sara Tarek. |
| Software Purpose | Salma Ahmed. |
| Software scope | Amina Gaber. |
| Definitions, acronyms, and abbreviations | Nagham Alaa. |
| Functional Requirements | Salma Ahmed. |
| Non Functional Requirements | Amina Gaber. |
| Use Case Model | Nagham Alaa. |
| Use Case Tables | Nagham Alaa, Sara Tarek, Amina Gaber, Salma Ahmed. |