**Cairo University  
Faculty of Computers and Information**

**CS251**

**Software Engineering I**

Online School Requirements Analysis Document

Phase 1: Software Requirements Specifications

|  |  |  |  |
| --- | --- | --- | --- |
| Mobile | Email | Name | ID |
| 01125376468 | **amrshosny@gmail.com** | **Amr Saeed** | **20140192** |
| 01141148336 | **a.magdy7077@gmail.com** | **Amr Magdy Ibrahim** | **20140196** |
| 01114178143 | **ashrakatelnow@gmail.com** | **Ashrakat Mokhtar** | **20140064** |

**November, 2016**

**Content**

**1.0 Introduction**-------------------------------------------------- **3**

1.1 Purpose -------------------------------------------------------------- 3

1.2 Audience ------------------------------------------------------------ 3

1.3 Scope ----------------------------------------------------------------- 3

1.4 Glossary ------------------------------------------------------------- 4

**2.0 Requirements** ---------------------------------------------------- **5**

5 2.1 Functional Requirements ---------------------------------------

2.2 Non-Functional Requirement ---------------------------------- 8

**3.0 System Models** ---------------------------------------------- **10**

3.1 Use Case Model -------------------------------------------- 10

3.2 Use Case tables --------------------------------------------- 11

**4.0 Ownership** -----------------------------------------------------  **21**

**1.0 Introduction**

**1.1 Document purpose:**

Educational platform where educational games are created by teachers and played by students to enhance their skills.

**1.2 Audience that is interesting in this Document:**

1- Client: Computek Company

2- Supervisor Teacher Assistant: Mohamed Samir

**1.3 Scope:**

The software is an educational platform that provides a good communication between teacher & student for better learning.

When a user signs up for an account, he identifies which type of user he is (a student or a teacher) by simple question and a proof of his profession like professional email.

When the user enters the website he's categorized into one of two types as a teacher or a student.

After the user logins, a list of categories appears as a home page and each category contains a number of games that are related to this game category. The available categories: Math, Science, Language & Programming. Once a game is selected the game window, rate and feedback shows up.

The teacher has the privilege to create a game (Filling data of an already built-before structure), also he can edit or remove games (of his own). Beside that he can rate any game.

As a student he can play, gain score, rate and give a feedback and the game owner responds to it

**1.4 Glossary:**

|  |  |
| --- | --- |
| Term | Definition |
| **Multiuser** | The users are categorized to different types |
| **platform** | Basic hardware and software where many applications can be run |
| **Language** | Enhance any oral language skills for users.  e.g. German, English |
| **Programming** | Enhance programming language skills which means learning how to give instructions to a device |
| **Feedback** | Comments that player gives for Games he plays |

**2.0 Requirements**

**2.1 Functional Requirement:**

* Sign up for student or teacher.
* Sign in for student or teacher.
* Forget password
* List category's games
* Play certain game
* Rate the game
* Feedback certain game
* Show Scoreboard
* Exit game
* View account/score
* Edit account
* Change password
* Change User Name
* Change Email
* Save changes
* Create a game
* Edit a game
* Delete a game
* Respond to feedback (Teacher)
* View Games (Teacher)
* Sign out

**2.1.1 Sign Up**

Each User (Student or Teacher) must register in the website, the sign up form for both Teacher and Student must contain Name, Email , password, Birthday, Gander and choose their profession either Student or Teacher with mention a professional website for teachers e.g. LinkedIn.

**2.1.2 Sign In**

Each user must enter his email (the one he recorded) and password. If there is something wrong, the system will inform him.

**2.1.3 Forget Password**

The user might forget his password, so using this function we ask him to write his email (Which is recorded in the system) and sent confirmation email to it including a temporary password so he could log in and reset his password.

**2.1.4 List Category's Games**

The user has the ability to choose the category of a game he wants to play or add in case if this user is a teacher. All categories are shown up once he logged in.

**2.1.5 Choose a Game to Play**

The user has the ability to choose any game to play through choosing the category that contains it.

**2.1.6 Play a Game**

The player can play any game by push "Play" button.

**2.1.7 Rate a Game**

The Game can be rated by the player only when he had played it or the rate will be disabled.

**2.1.8 Give Feedback**

The player can give feedback of a game he played to the teacher. If not, the comments will be disabled.

**2.1.9 Show Scoreboard**

After each game player plays, a scoreboard shows up to the user contains the highest scores achieved of the whole category.

**2.1.10 Exit Game**

The player can easily Exit the game by pressing "Exit" button.

Exit button can be on the screen or click "Esc" on the keyboard.

**2.1.11 View Account**

The user can view his profile, check his score and edit the content from "Profile" at Menu bar.

**2.1.12 Edit Account**

The user can edit his profile after though choosing "profile" then choose "Edit profile", he can change anything he wants, then click "Save".

**2.1.12 Change Password**

The user has the ability to change his password anytime he wants through "profile" at menu bar then "Edit profile".

**2.1.13 Change User Name**

The user has the ability to change his User Name (Which appear in the scoreboard, Also it's a unique name) anytime he wants through "profile" at menu bar then "Edit profile".

**2.1.14 Change Email**

Maybe the user wants to change his email because of some issues, so he can set a new one through "profile" at menu bar then "Edit profile".

**2.1.15 Save Changes**

The user can save any changes he makes by pressing "Save" button in the same window he has editing in.

**2.1.16 Create a Game**

Only teachers can create games by choosing the category that the game belongs to, and then choose "Add Game" Which through it he can fill all necessary data then press "Save" button.

**2.1.17 Edit a Game**

The teacher can edit his own games by clicking "My Games" from the menu, then choose the desired game and press "Edit" button. After editing he can save changes by pressing "Save" button.

**2.1.18 Delete a Game**

Teachers can delete their own games by clicking "My Games" from the menu, then choose the desired game and press "Delete" button. A question will appear if he really wants to remove it, He can press "Yes" to be deleted or "No" to still in his list.

**2.1.19 Respond to feedback**

Teachers can respond to students' feedback by choosing "MyGames" then open the wanted game to view feedback and respond to it in comments.

**2.1.20 View Games**

Teachers can view their own games and view total rate of each game by pressing "MyGames" from menu bar, from this page he can edit, delete games and respond to feedback.

**2.1.21 Sign Out**

The user can leave the site but still has an account on it, but to enter again he needs to sign in again.

To sign out the user press "Sign out" from menu.

**2.2 Non-Functional Requirement:**

* Usability
* Security
* Privacy
* Flexibility
* Reliability
* Modifiability
* Legal

**2.2.1 Usability**

Ease of use by users, as the number of steps to perform a specific task takes no more than 6 clicks.

**2.2.2 Security**

User email and password are required to access the user information as a kind of protection of information and using resources.

**2.2.3 Privacy**

Users have no communications at all in this site, on one know if someone else has an account on it or know if there is a particular person who achieved any achievement. No public information of any user.

**2.2.4 Flexibility**

The system can interact with different users (teachers and students) and easily responses to their requirements.

**2.2.5 Reliability**

The platform is estimated (on services used and algorithm working) to be running for 2 weeks continuously with no failure and if system goes down we are synchronizing changes so data loss is hard to happen.

**2.2.6 Modifiability**

As the scope is very simple and clear, it's very easy it makes changes in the software within this scope.

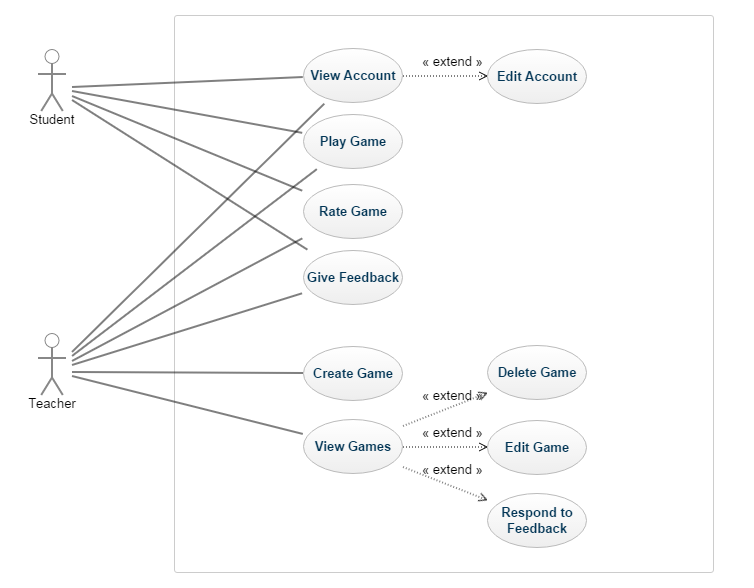
**2.2.7 Legal**

We don't allow much similarity between any games ideas as it's restricted by our copyrights terms and conditions.

**3.0 System Models**

**3.1 Use Case Model**

The following gives overview of the identified use cases.



**The following actors are identified:**

**Student**: Is able to choose any game to play, rate it & give it feedback

**Teacher**: Is able to create a game, delete a game he made or edit it besides playing any game too.

**3.2 Use Case tables**

**3.2.1 Play Game**

|  |  |  |
| --- | --- | --- |
| Use Case ID: | Play\_Case1 | |
| Use Case Name: | Play game | |
| Actors: | 1- Student  2- Teacher | |
| Pre-conditions: | The player is already logged in and list of categories shown up | |
| Post-conditions: | Game has been finished and a new score added to the player's profile. | |
| Flow of events: | User Action | System Action |
| 1- Choose specific category |  |
|  | 2- List of Games of the chosen category shown up |
| 3- Choose the desired game |  |
|  | 4- Window with play option, rate, Give feedback show up |
|  | 5- Press "play" button |  |
|  |  | 6- the system starts a new game |
|  | 7.1- If the player press Exit button |  |
|  |  | 7.1.1- The system pauses the game and asks the player if he wants to quit |
|  | 7.1.1.1- If the player wants to quit, he presses "yes" |  |
|  |  | 7.1.1.1.1 The system shows score achieved and add it to the player profile and shows the scoreboard then returns to the game window |
|  | 7.1.1.2- If the player still wants to play, he presses "No" |  |
|  |  | 7.1.1.2.1- The system resumes the game and when the game is finished, the system shows the score, scoreboard and added both score and rank to the player profile then returns to the game window |
| Exceptions: | **User Action** | **System Action** |
| - |  |
|  | - |

**3.2.2 Rate Game**

|  |  |  |
| --- | --- | --- |
| Use Case ID: | Rate\_Case2 | |
| Use Case Name: | Rate game | |
| Actors: | 1- Student  2-Teacher | |
| Pre-conditions: | The player already logged in and had played the game at least once | |
| Post-conditions: | The player profile is updated with new rate in addition to updating the total rate of the game | |
| Flow of events: | **User Action** | **System Action** |
| 1.1- choose a suitable rate for the game, then click the button "save" in case of the game was opened | 1.1.1- The system update the total rate of the game and if the user has rated the game before, the old one will be deleted |
| 1.2- If the game wasn't open, the user choose the desired category |  |
|  | 1.2.1- List of games belong to that category shows up |
| 1.2.1.1- Choose the desired game |  |
|  |  | 1.2.1.1.1- Game window appears with total rate and feedback with the ability to give a new rate |
|  | 1.2.1.1.1.1- Give a suitable rate then click "save" |  |
|  |  | 1.2.1.1.1.1.1- The system updates the total rate of the game and if the user has rated the game before, the old one will be deleted |
| Exceptions: | **User Action** | **System Action** |
| 1- Rating the game without playing it before |  |
|  | 2-The rate system is disabled for the user |
| Includes: | Updating function of updating total rate of a game | |
| Notes and Issues: | The user cannot modify his rate, if he'd like to rate a game one more time the old rate will be deleted and his new rate will be recorded | |

**3.2.3 Give Feedback**

|  |  |  |
| --- | --- | --- |
| Use Case ID: | GiveFeedback\_Case3 | |
| Use Case Name: | Give feedback | |
| Actors: | 1- Student  2-Teacher | |
| Pre-conditions: | The player already logged in and had played the game at least once | |
| Post-conditions: | The feedback of the user is published and the page of feedbacks is updated | |
| Flow of events: | **User Action** | **System Action** |
| 1.1- Write feedback in a comment for the game, then click " Enter" in case the game was opened | 1.1.1- The system update the page that contains comments of the game |
| 1.2- If the game wasn't open, the user choose the desired category |  |
|  | 1.2.1- List of games belong to that category shows up |
| 1.2.1.1- Choose the desired game |  |
|  |  | 1.2.1.1.1- Game window appears with total rate and feedback with the ability to write feedback in comment |
|  | 1.2.1.1.1.1- Give a suitable feedback in a comment then click Enter |  |
|  |  | 1.2.1.1.1.1.1- The system updates the page to include the new comment |
| Exceptions: | **User Action** | **System Action** |
| 1- Trying to give a feedback without playing the game before |  |
|  | 2- Writing comments is disabled |
| Includes: | Updating function of updating Game window | |

**3.2.4 Create Game**

|  |  |  |
| --- | --- | --- |
| Use Case ID: | CreateGame\_Case4 | |
| Use Case Name: | Create game | |
| Actors: | 1-Teacher | |
| Pre-conditions: | A teacher has logged in into the platform | |
| Post-conditions: | New game is added to specific category and to the teacher's own games | |
| Flow of events: | **User Action** | **System Action** |
| 1- Choose specific category that his game belongs to it |  |
|  | 2- List of Games of the chosen category is shown with "Add Game" Button |
| 3- Push "Add Game" button |  |
|  | 4- A form appears asks the user for structure of the game and game name. According to the structure ,data form will be appeared right below it |
|  | 5- Fill data and press "Save" button |  |
|  |  | 6- New game is added both to the category and the teacher's own games. |
| Exceptions: | **User Action** | **System Action** |
| - |  |
|  | - |
| Includes: | Including updating data for Teacher's own games and category | |
| Notes and Issues: | The structures are already built-in and teacher selects one of them | |

**3.2.5 View Games**

|  |  |  |
| --- | --- | --- |
| Use Case ID: | ViewGames\_Case5 | |
| Use Case Name: | View Games | |
| Actors: | 1-Teacher | |
| Pre-conditions: | The teacher has already logged in | |
| Post-conditions: | Either he viewed rate of his games, responded to feedback, or edit his games | |
| Flow of events: | User Action | System Action |
| 1- Press "My Games" |  |
|  | 2- List of games created by him shows up |
| 3.1- If the teacher wants to delete, he presses "Delete" button below the desired game |  |
|  | 3.1.1- the system asks the teacher if he really wants to delete the game |
| 3.1.1.1- If yes, the teacher press" Yes" |  |
|  |  | 3.1.1.1.1 the system updates both teacher's games and the category it was belong to it |
|  | 3.1.1.2 If no, the teacher press "No" |  |
|  |  | 3.1.1.2.1 the system ignores this command |
|  | 3.2-If the teacher wants to edit, he presses "Edit" button below the desired game |  |
|  |  | 3.2.1 the system opens a form that allow the teacher to edit the content of the game |
|  | 3.2.1.1 the teacher makes his changes then presses "Save" button |  |
|  |  | 3.2.1.1.1 the system saves the changes of the game |
|  | 3.3 If the teacher wants to respond to feedback of a certain ,he opens the game |  |
|  |  | 3.3.1 the system shows the feedbacks regardless showing the game and total rate |
|  | 3.3.1.1 review comments and respond to any |  |
|  |  | 3.3.1.1.1 System updates the page. |
| Exceptions: | User Action | System Action |
| - |  |
|  | - |
| Includes: | Updating teacher's games and categories of his games | |
| Notes and Issues: | - | |

**3.2.6 View Account**

|  |  |  |
| --- | --- | --- |
| Use Case ID: | ViewAccount\_Case6 | |
| Use Case Name: | View account | |
| Actors: | 1-Teacher  2-Student | |
| Pre-conditions: | The user is already logged in | |
| Post-conditions: | The profile has been viewed or updated | |
| Flow of events: | **User Action** | **System Action** |
| 1- The user presses "My Account" from menu bar |  |
|  | 2- Opens user's profile with his basic info, achievements, and scores. |
| 3.1- If the user wants to edit, he presses "Edit" button. |  |
|  | 3.1.1- Opens a form that allows the user to edit his data. |
| 3.1.1.1- Updates his data then presses "save" button |  |
|  | 3.1.1.1.1- System saves changes and updates profile. |
| Exceptions: | **User Action** | **System Action** |
| - |  |
|  | - |
| Includes: | Updating user's account | |
| Notes and Issues: | - | |

**3.2.6 Delete Game**

|  |  |  |
| --- | --- | --- |
| Use Case ID: | Play\_Case6 | |
| Use Case Name: | Delete game | |
| Actors: | 1- Teacher | |
| Pre-conditions: | The player is already logged in and clicked on "My Games" | |
| Post-conditions: | Game has been deleted and teacher's games have been updated | |
| Flow of events: | User Action | System Action |
| 1-Teacher presses "Delete" button below the desired game |  |
|  | 2- the system asks the teacher if he really wants to delete the game |
| 2.1- If yes, the teacher press" Yes" |  |
|  | 2.1.1 the system updates both teacher's games and the category it was belong to it |
|  | 3.2 If no, the teacher press "No" |  |
|  |  | 3.2.1 the system ignores this command |
| Exceptions: | **User Action** | **System Action** |
| - |  |
|  | - |

**3.2.7 Edit Game**

|  |  |  |
| --- | --- | --- |
| Use Case ID: | Play\_Case7 | |
| Use Case Name: | Edit game | |
| Actors: | 1- Teacher | |
| Pre-conditions: | The player is already logged in and clicked on "My Games" | |
| Post-conditions: | Game has been Edited | |
| Flow of events: | User Action | System Action |
| 1-Teacher presses "Edit" button below the desired game |  |
|  | 2- the system opens a form that allow the teacher to edit the content of the game |
| 3- the teacher makes his changes then presses "Save" button |  |
|  | 4-the system saves the changes of the game |
| Exceptions: | **User Action** | **System Action** |
| - |  |
|  | - |

**3.2.8 Respond to Feedback**

|  |  |  |
| --- | --- | --- |
| Use Case ID: | Play\_Case8 | |
| Use Case Name: | Responding Feedback | |
| Actors: | 1- Teacher | |
| Pre-conditions: | The player is already logged in and clicked on "My Games" | |
| Post-conditions: | The respond has been posted | |
| Flow of events: | User Action | System Action |
| 1-The teacher opens the desired game |  |
|  | 2-the system shows the feedbacks regardless showing the game and total rate |
| 3-review comments and respond to any |  |
|  | 4-System updates the page. |
| Exceptions: | **User Action** | **System Action** |
| - |  |
|  | - |

# 4.0 Ownership Report

|  |  |
| --- | --- |
| Item | Owners |
| All document | Amr Saeed |
| All document | Amr Magdy |
| All document | Ashrakat Mokhtar |

Github Link : <https://github.com/MagdyA/SWE1-SRS.git>