**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 ----------------------------------------- 11

**3.0 System Models** ---------------------------------------------- **13**

3.1 Use Case Model ------------------------------------------------------------- 13

3.2 Use Case tables -------------------------------------------------------------- 14

**4.0 Ownership Report and GitHub Link** --------------------  **29**

4.1 Ownership Report ---------------------------------------------------------- 29

4.2 GitHub Link ------------------------------------------------------------------- 29

**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 interested 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 and 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 a 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 are Math, Science, Language and Programming. Once a game is selected from one of the available categories 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 into 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 a player gives for a game he has played |

**2.0 Requirements**

**2.1 Functional Requirement:**

* Sign up
* Sign in
* Forget password
* List category's games
* Play a certain game
* Hint
* Help
* Rate a game
* Feedback a certain game
* Show scoreboard
* Join a tournament
* Exit game
* View account
* Edit account
* Change profile picture
* Change password
* Change username
* Change email
* Save changes
* Create a game (Teacher)
* Create a tournament (Teacher)
* Edit a game (Teacher)
* Delete a game (Teacher)
* 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, Birthdate, Gender and choose their profession either Student or Teacher with a mention to a professional reference for teachers e.g. LinkedIn.

**2.1.2 Sign In**

Each user must enter his registered email and password. If there is something wrong, the system will inform him.

**2.1.3 Forget Password**

The user may forget his password, so he we will be asked to write his registered email. After that, a confirmation email will be sent to his registered email including a temporary password so he could login and reset his password.

**2.1.4 List Category's Games**

The user has the ability to choose the desired category, list its games and choose a game to play, or add a new game in case of the user was a teacher.

**2.1.5 Play a Certain Game**

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

**2.1.6 Hint**

If the user faces a difficult question while playing a game and needs a hint, he has the ability to check one which is added by the teacher, but this will affect his score.

**2.1.7 Help**

If the user has any questions about how to play a particular game or about the score system, the help buttons contains a clear description of the game.

**2.1.8 Rate a Game**

The game can be rated by the player only if he had previously played it or the rate will be disabled.

**2.1.9 Feedback a Certain Game**

The player can give a feedback of a particular game to the teacher only if he had previously played it. If not, the comments will be disabled.

**2.1.10 Show Scoreboard**

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

**2.1.11 Join a Tournament**

Anyone registered can participate in tournaments either official or for training but teachers will be joined as unofficial (no rating).

**2.1.12 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.

**2.1.13 View Account**

The user can view his profile, check his score and edit the content from "Profile" at Menu bar. He will be able to view his score in each game he plays, his basic info. The user's profile also have a diagram shows his activity on the site.

**2.1.14 Edit Account**

The user can edit his profile through choosing "Profile" then click on "Edit Profile". There is some information that is allowed to be changed and others cannot be changed. The user has the ability to change the allowed parts, then click "Save".

**2.1.15 Edit Profile Picture**

The user can update his profile picture anytime he wants

**2.1.16 Change Password**

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

**2.1.17 Change Username**

The user has the ability to change his username (Which appears in the scoreboard, also it's a unique name) anytime he wants through "Profile" at menu bar then "Edit Profile".

**2.1.18 Change Email**

The user may want to change his email because of some issues, so he can set a new one through "Profile" at menu bar then "Edit profile".

**2.1.19 Save Changes**

The user can save any changes he makes by pressing "Save" button in the same window where he performs a modification in.

**2.1.20 Create a Game**

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

**2.1.21 Create a Tournament**

Each teacher is allowed to create tournaments for other user to enroll in; His tournaments will be viewed in his profile. He can create it through "Create Tournament" from menu bar. Tournaments can be accessed by anyone as what happened in rating and score changes for any participant of it.

**2.1.22 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.

**2.1.23 Delete a Game**

Teacher can delete his 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 has the choice to press "Yes" if he really wants to delete it or "No" to cancel the deletion process.

**2.1.24 Respond to Feedback**

Teachers can respond to students' feedback by choosing "My Games" then open the desired game to view the feedback and respond to it.

**2.1.25 View Games**

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

**2.1.26 Sign Out**

The user has the ability to sign out from his account and leave the site, but to enter again he needs to sign in one more time with his already existed account. The user can sign out by pressing "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**

No sensitive information will be public (e.g. Email) but anyone can access others' pages to see their achievements.

**2.2.4 Flexibility**

The system can interact with different users (teachers and students) and easily responds 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.

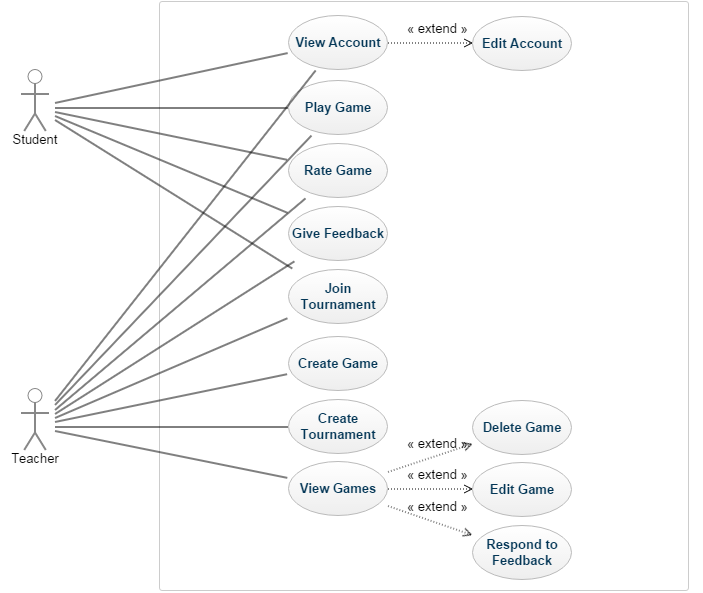
**2.2.7 Legal**

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 and give it feedback.

**Teacher**: Is able to create, delete or edit a game he made 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 |
| 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 |

**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** |
| - |  |
|  | - |
| 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 |
| - |  |
|  | - |
| 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** |
| - |  |
|  | - |
| 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** |
| - |  |
|  | - |

**3.2.9 Create Tournament**

|  |  |  |
| --- | --- | --- |
| Use Case ID: | Play\_Case9 | |
| Use Case Name: | Create Tournament | |
| Actors: | 1- Teacher | |
| Pre-conditions: | The player is already logged in | |
| Post-conditions: | The tournament has been published and added to teacher's profile | |
| Flow of events: | User Action | System Action |
| 1- Teacher clicks on "Create tournament" from menu bar |  |
|  | 2- the system shows a form that allows the teacher to choose the date, duration, structure of game and fill data according to each structure |
| 3- Teacher fills data then press "publish" |  |
|  | 4-The system publishes the tournament. |
| Exceptions: | **User Action** | **System Action** |
| - |  |
|  | - |

**3.2.10 Join Tournament**

|  |  |  |
| --- | --- | --- |
| Use Case ID: | Play\_Case10 | |
| Use Case Name: | Join tournament | |
| Actors: | 1- Teacher  2- Student | |
| Pre-conditions: | The player is already logged in | |
| Post-conditions: | Participant is registered in the tournament | |
| Flow of events: | User Action | System Action |
| 1-Participant presses "Register" to join the tournament |  |
|  | 2- The System records his name |
| Exceptions: | **User Action** | **System Action** |
| - |  |
|  | - |

**4.0 Ownership Report and GitHub Link**

**4.1 Ownership Report**

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

**4.2 GitHub Link**

https://github.com/MagdyA/SWE1-SRS