Cairo University  
Faculty of Computers and Information



**CS251**

**Software Engineering I**

F3A

Software Requirements Specifications

Amany AbdEl-Aal

Amira Khaled

Aya Mohammed

Fatma Ahmed

5/11/2016

Contents

[Team 3](#_Toc402452669)

[Document Purpose and Audience 3](#_Toc402452670)

scenario 3

[Introduction 4](#_Toc402452671)

[Software Purpose 4](#_Toc402452672)

[Software Scope 4](#_Toc402452673)

[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 15](#_Toc402452681)

# 

# Team

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Email** | **Mobile** |
| 20140201 | Fatma Ahmed | [0fatma.ahmed@gmail.com](mailto:0fatma.ahmed@gmail.com) | 01147084811 |
| 20140069 | Amany Abd El-Aal | [Amanya.sultan@outlook.com](mailto:Amanya.sultan@outlook.com) | 01224102660 |
| 20140075 | Amira Khaled Hassanin | [amiraKhaled35@gmail.com](mailto:amiraKhaled35@gmail.com) | 01127018549 |
| 20140089 | Aya Mohamed Mohamed | [Ayakenawy7@gmail.com](mailto:Ayakenawy7@gmail.com) | 01060928079 |

# Document Purpose and Audience

* **This document describes the software requirements for a game-based educational platform.**
* **It’s intended for the owner of the project and other developers.**

# Scenario

The system is introducing games to understand specific concepts of different levels of math i.e. (math1, math2, math3). It support to try playing game or sign in by your account that you have created as a teacher or student the system verify your entering user name and password. If you create account as a teacher you have to have certifications online. The student and teacher have this functions they can play game, rate game from 1 to 5 \*, share any data, send complain (technical complain) to system, change setting like changing of design color or update personal information, chatting with specific activate account and download game except unavailable new game (doesn’t take enough rate), if you are a teacher you have extra function like modify game belongs to another teacher if and only if he gives you the permission and develop a new game. system is Usability simple to use, Reliability support up to 4 millions user, Availability of system should working 24 per hours, Performance according to short response time, Security of every account access by only user account and password, has Constraints rules of game and has Privacy that means any teacher can modify another game if and only if owner of game give him permission.

# Introduction

## Software Purpose

**1. To let students gain more information in different, easy and funny ways about specific field they were chosen.**

**2. Supply teachers with new tools and features to help them adding their own game.**

## Software Scope

* **This is web application software for education (math).**
* **The structure of game is multiple choice, voice interaction and complete sentences.**

# Requirements

## Functional Requirements

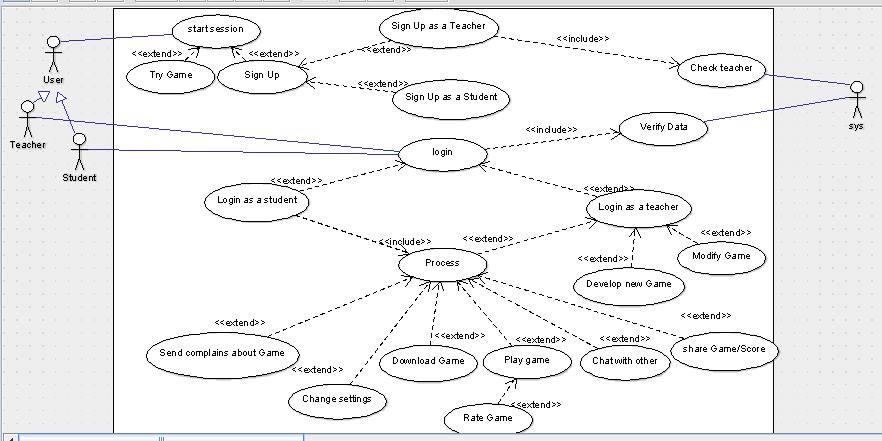
* **Try game: - The user will select which category s/he will play (Math1, Math2 and Math3), then try the game at which level s/he selects to play.**
* **Sign up: - User set personal information which is required by system to create an account for him.**
* **Check teacher: - If user signs up as a teacher, the system must check specific information (i.e. online certifications) to ensure if s/he is already a teacher.**
* **Play game: - the user will Play the game from recommended levels by the system which are determined from the evaluation test.**
* **Rate game: - To rate the performance and quality of the game depended on playing the game.**
* **Share game: - To share the game and scores of game with others users of the same game.**
* **Send complain about game: - If the user faces any problem at system, then s/he will able to send a report about it.**
* **Change setting: - the user Can control on his/her profile settings and make changes like change picture profile, user name, password or any personal information.**
* **Chat with other: -the user can chat with any user has an account on the website.**
* **Download game: -the user can download the game on his/her platform to play it easy without needing the connection of internet.**
* **Develop game: - the teacher creates game that supports a specific concept belongs to one of categories which are determined by the system.**
* **Modify game: - Teacher can modify game that s/he developed or game developed by another teacher only when he takes the permission to modify it.**
* **Verify data: - System check correctness and validation of data which is entered by the user.**

## Non Functional Requirements

* **Usability: - Simple to use.**
* **Reliability: - Accepts up to 4 million users.**
* **Supportability: - System can solve problems that the user sends complain about it.**
* **Availability: - system is working 24 hours.**
* **Performance: - short response time, not crash more than 5 minutes per week.**
* **Security: - every account access by only user account and password.**
* **Constraints: - Rules of game.**
* **Privacy: - Any teacher can modify another game if and only if owner of game give him permission.**

# System Models

## Use Case Model



## Use Case Tables

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 1 | |
| Use Case Name: | Try game. | |
| Actors: | User (Teacher, Student, Guest) | |
| Pre-conditions: | Start session. | |
| Post-conditions: | Get score. | |
| Flow of events: | **User Action** | **System Action** |
| 1- Select game. |  |
|  | 2- Open available levels of selected game. |
| 3- Select desired level. |  |
|  | 4- Open selected level. |
| 5- Play game |  |
|  |  | 6- Calculate score. |
| Exceptions: | **User Action** | **System Action** |
| 1- Select unavailable level to play. |  |
|  | 2- Unavailable to play right now. |
| Includes: |  | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 2 | |
| Use Case Name: | Sign up as a student. | |
| Actors: | User (New visitor). | |
| Pre-conditions: | User wants to create new student account. | |
| Post-conditions: | Student account created. | |
| Flow of events: | **User Action** | **System Action** |
| 1- Choose sign up as a student. |  |
|  | 2- Open form of student creation. |
| 3- Fill his/her personal information that is required. |  |
|  | 4- Submit. |  |
|  |  | 5- Account created. |
| Exceptions: | **User Action** | **System Action** |
| 1- User enter invalid field. |  |
|  | 2- Display error message. |
| Includes: |  | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 3 | |
| Use Case Name: | Sign up as a teacher. | |
| Actors: | User (New visitor). | |
| Pre-conditions: | User wants to create teacher account. | |
| Post-conditions: | Teacher account created. | |
| Flow of events: | **User Action** | **System Action** |
| 1- Choose sign up as a teacher. |  |
|  | 2- Open form of teacher creation. |
|  | 3- Fill his/her personal information that is required. |  |
|  | 4- Submit. |  |
|  |  | 5- System check entered information as if s/he is already a teacher or not. |
|  |  | 6- Account created. |
| Exceptions: | **User Action** | **System Action** |
| 1- User data entered as a teacher isn’t correct. |  |
|  | 2- Data entered is invalid and not identified.  3- System rejects the registered teacher. |
| Includes: | Check teacher. | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 4 | |
| Use Case Name: | Login. | |
| Actors: | User (Teacher, Student). | |
| Pre-conditions: | Has an account. | |
| Post-conditions: | Account opened. | |
| Flow of events: | **User Action** | **System Action** |
| 1- User Enter username and Password. |  |
|  | 2- System Verifies user data |
| Exceptions: | **User Action** | **System Action** |
| 1- User Enter invalid username or Password. |  |
|  | 2- Entered data is invalid and not identified.  3- System rejects login. |
| Includes: | Verify entered data. | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 5 | |
| Use Case Name: | Share game/score. | |
| Actors: | User(Student, Teacher). | |
| Pre-conditions: | Login. | |
| Post-conditions: | Data shared. | |
| Flow of events: | **User Action** | **System Action** |
| 1- User Share data (Game link, Game level). |  |
|  | 2- List web sites that can share data above. |
|  | 3- Select the desired web site. |  |
|  |  | 4. Data shared. |
| Includes: |  | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 7 | |
| Use Case Name: | Chat with other. | |
| Actors: | User (Teacher, Student). | |
| Pre-conditions: | Login. | |
| Post-conditions: | Successful Communication. | |
| Flow of events: | **User Action** | **System Action** |
| 1- User press chat. |  |
|  | 2- List available people within her/his network. |
| 3- Choose the one s/he wanted. |  |
|  | 4- Open chat window of selected person. |
| Exceptions: | **User Action** | **System Action** |
| 1- User select deactivated account. |  |
|  | 2- The selected account had deactivated. |
| Includes: |  | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 8 | |
| Use Case Name: | Play game. | |
| Actors: | User (Teacher, student). | |
| Pre-conditions: | Login. | |
| Post-conditions: | Change in score. | |
| Flow of events: | **User Action** | **System Action** |
| 1- Play game. |  |
|  | 2- System Save score. |
| Exceptions: | **User Action** | **System Action** |
| 1- User exit before end of game. |  |
|  | 2- Score won’t change. |
| Includes: |  | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 9 | |
| Use Case Name: | Download game. | |
| Actors: | User (Teacher, Student). | |
| Pre-conditions: | Login. | |
| Post-conditions: | Game downloaded. | |
| Flow of events: | **User Action** | **System Action** |
| 1- User selects game. |  |
|  | 2- Downloading.  3- Game downloaded. |
| Exceptions: | **User Action** | **System Action** |
| 1- Select non available game to download. |  |
|  | 2- Unavailable to download. |
| Includes: |  | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 10 | |
| Use Case Name: | Change setting. | |
| Actors: | User(student, teacher). | |
| Pre-conditions: | Login. | |
| Post-conditions: | Information changed. | |
| Flow of events: | **User Action** | **System Action** |
| 1- Select the field that wants to change. |  |
| 2- Fill new data. |  |
| 3- Save. |  |
|  | 4- changed done. |
|
| Exceptions: | **User Action** | **System Action** |
| 1- Fill invalid data according to fields’ constraints. |  |
|  | 2-Not valid. |
| Includes: |  | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 11 | |
| Use Case Name: | Rate game. | |
| Actors: | User(student, teacher). | |
| Pre-conditions: | Play game. | |
| Post-conditions: | Rate changed. | |
| Flow of events: | **User Action** | **System Action** |
| 1-Enter rate. |  |
|  | 2- Change rate of game. |
| Includes: |  | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 12 | |
| Use Case Name: | Send complain about game. | |
| Actors: | User(student, teacher). | |
| Pre-conditions: | Login. | |
| Post-conditions: | Complain send to system. | |
| Flow of events: | **User Action** | **System Action** |
| 1- S/he enters complain. |  |
|  | 2- Receive complain. |
| Includes: |  | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 13 | |
| Use Case Name: | Modify game. | |
| Actors: | User(teacher). | |
| Pre-conditions: | Login as a teacher. | |
| Post-conditions: | Game modified. | |
| Flow of events: | **User Action** | **System Action** |
| 1- Select game. |  |
|  | 2- Accepted to modify. |
| 3-teacher starts to modify. |  |
|  | 4- Game modified. |
| Includes: |  | |
| Notes and Issues: | If owner of game not accept to modify the system will prevent modifying. | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 14 | |
| Use Case Name: | Develop game. | |
| Actors: | User(teacher) | |
| Pre-conditions: | Login as a teacher. | |
| Post-conditions: | Game created. | |
| Flow of events: | **User Action** | **System Action** |
| 1- Select tool to create game. |  |
|  | 2- Make this tool available. |
| 3- Create game. |  |
|  | 4- Save game. |
| Exceptions: | **User Action** | **System Action** |
| 1- Use memory higher than determined size. |  |
|  | 2- Not valid. |
| Includes: |  | |
| Notes and Issues: |  | |

# Ownership Report

All team members think together.

GitHub Link

<https://github.com/Monmon96/phase1>