Cairo University  
Faculty of Computers and Information



**CS251**

**Software Engineering I**

Use or Lose

**Software Requirements Specifications**

Zyad Mustafa Ghorab.

Shady Sherif.

Shrief Mohamed.

Esraa Sayed Abass.

November & 2016

Contents

[Team 3](#_Toc402452669)

[Document Purpose and Audience 3](#_Toc402452670)

[Introduction 3](#_Toc402452671)

[Software Purpose 3](#_Toc402452672)

[Software Scope 3](#_Toc402452673)

[Requirements 4](#_Toc402452675)

[Functional Requirements 4](#_Toc402452676)

[Non Functional Requirements 8](#_Toc402452677)

[System Models 9](#_Toc402452678)

[Use Case Model 9](#_Toc402452679)

[Use Case Tables 10](#_Toc402452680)

[Ownership Report 20](#_Toc402452681)

# Team

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Email** | **Mobile** |
| 20140131 | Zyad Mostafa Ghorab | zyad.ghorab97@gmail.com | 01201520595 |
| 20140148 | Shady Sherif | Shadysherif14@gmail.com | 01157701147 |
| 20140149 | Shrief Mohamed Anwar | Shriefanwer48@gmail.com | 01098237871 |
| 20140054 | Esraa Sayed Abass | Esraamagdy546@gmail.com | 01069273166 |

# 

# Document Purpose and Audience

# Purpose:

This document is about requirements specifications for developing a new educational Game-based Software called "Use or Lose", that help Students to learn basic knowledge about science, math and programming concepts, it will be available online 24/7, it includes some games (Multiple Choices game, matching pictures, Run code Game) in a specific category.

**Audience:**

The expected audience: project manager, project staff, Software Engineering TAs, developing team.

# Introduction

## 

## Software Purpose

* It is educational software that helps students, juniors and anyone interested to know the programming concepts, math formulas and science equations through playing simple games.

## Software Scope

* It is a web-based application that allows users or students to learn some concepts in a specific category like (math, science and programming concepts) through playing simple games, allow him to play games and take final score, rate the game and leave comment for the teacher.
* It will allow teachers to play game, create a new game based on some requirements made by the system, delete game with less rating , update their games with new versions and new features, and reply on student's comments on his games to provide the game supportability and each user can view his profile .

## 

# Requirements

## Functional Requirements:

When the user enters the game website, he will be asked if he want to enter as student, teacher or guest. If he selects student or teacher log in form must appear:

* Login:
  + The user enters a valid Username, and password then the system must check the validation and sends a confirmation message for authentication.
* Log out
  + The user can log out of the site at any time.
* Sign up:
  + The user should enter a valid username, password, Gender, mobile, email and Academic account for teacher to be registered on the database or he can register with Facebook account or Gmail account.
* **Playing a game**
* Choose category of games:
  + After the user logged into the system and wants to play a game, he will select the category of the game.
* Choose a game:
  + After choosing the category, he will select the game he wants to play.
* Show tips for playing:
  + Show some help for how to play (Optional) can be skipped.
* Start a game:
  + Click Start button to start playing.
* Update status of playing:
  + While playing, the system will update the status of the game periodically (Score, Level).
* Show score:
  + After finishing each level, the system will show the current user score.
* Show evaluation form:
  + After finishing 3 levels of any game, the system will show an evaluation form for the game, that the user can evaluate the game (Optional) can be skipped.
* Add comments: :
  + At any state of playing the game the user can write a comment on the game.
* Back to main menu:
  + At any state of playing the game the user can exit the current game and choose another game and the system saves the last good state.
* **Teacher Creates a new Game**

Teacher must be logged in for creating a new game.

* Choose category of games:
  + Teacher selects the category that he wants to add the game in.
* Show tips for creating the game:
  + Teacher will see some help for creating the game (Optional) can be skipped.
* Start to create the game:
  + The system will open template for creating the game, the teacher can fill the template and submit it.
* Check requirements for validation.
  + After the teacher submits the template, the system will check for the validation of the game.
* Add game to Approval Queue: :
  + After succeeding in validation, the system will add the game to the queue for the admin to approve it.

**Teacher Updates a new Game**

Teacher must be logged in for updating one of his game.

* Show my Games to be updated:
  + The system will show list of the teacher’s games.
* Choose from available games:
  + The teacher will choose one game to be updated.
* Start to update the game:
  + The teacher can update on the template and submit it.
* Check requirements for validation:
  + After the teacher submits the template, the system will check for the validation of the game.
* Add game to approval Queue as (Replacement):
  + After succeeding in validation, the system will add the game to the queue for the admin to approve it.

**Teacher Replies to comments**

Teacher must be logged in for replying on players’ comments.

* Show notification list:
  + After user clicks notification button, System will show notification list that have the comments on his games.
* Choose specific notification:
* Teacher will choose specific notification to go to comments and replies to this comment.
* Show the comments of the chosen notification:
* System will show this form for teacher to reply on student's comments.
* Add reply
* Teacher will add reply to the chosen comment.
* Update reply:
* Reply will be updated to website by the system.

**Teacher Deletes Game**

Teacher must be logged in for deleting one of his game.

* Press delete button:
  + Teacher will press delete button to delete one of his game.
* Show my Games to be deleted:
  + The system will show list of the teacher’s games.
* Choose and delete game:
  + Teacher will choose game from list.
* Update deletion:
  + Deletion of the game will be updated to website by the System.

**Admin filters games**

Admin must be logged in for filtering game.

* Press filter button:
* Admin will press filter button to filter any game.
* Show category list of games:
* System will show category of games.
* Choose game category:
* Admin will select the category that he wants to filter game.
* Show game list in this category:
* System will show list of category arranged by the lowest rate and number of users playing this game.
* Press delete button:
* If admin checks the validation of requirements to filter game, then press delete button.
* Remove game:
* System will filter selected game from website.

**User View Profile**

User must be logged in for viewing profile.

* Press view profile button:
* User will press view profile button to show his profile.
* Show user profile:
* The system will show user's data profile (Profile Picture, Name, Change password,…etc.).

## Non-Functional Requirements:

* + There are some important Quality attributes for this software we’ll provide them:

|  |  |
| --- | --- |
| **Function** | **Details** |
| **Performance** | * System response time will be at most 2 seconds and may be less.   -- website will be uploaded to high quality host to make sure the good performance . |
| **Scalability** | * System should be able to support up to 100 simultaneous game players.   -- high quality DBMS (Database Management System). |
| **Reliability** | * System saves the last good state for the current user if something goes wrong,   Always saves last good action in data base for this user.  -- By calling "update "function that updates score and level after each successful state.   * MTBF (mean time between failures) will be 2 months in the beginning. |
| **Usability** | * System should be easy to use by the users who are not very technical aware. * It will provide help for each part in the web app.   -- we can make good interface with simple obvious buttons and labels to describe this event and simple help, tips for interaction with the system |
| **Security** | * Validation of the user (Log in using username and password).   -- make a good design for database  -- make it easy to check validation of users and if he forgets the password we can send him an email with the password. |
| **Availability** | * System will be available online for users 24/7.   -- we can reserve alternative host and make it available if the first is unavailable. |
| **Supportability** | * System will support user by considering the comments and notify the game owner.   -- support team will notify teacher and after a specific period of time admin will checks if teacher replied. |
| **Correctness** | * System does the proper things properly, often applied especially to calculations and validating answers.   -- we call function "check requirements of the game " to check the game is running correctly or not |
| **Privacy** | * Protecting personal information and undesired access to personal space.   -- Authentication for each user to access his personal data only by username and password - Other users cannot see personal data of other users. |
| **Safety** | * not causing harm, injury or damage for users.   -- the website will be No-Advertisements site. |

# 

# System Models

## Use Case Model

## Capture1

## Use Case Tables

* Playing game:

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 1 | | |
| Use Case Name: | Playing game | | |
| Actors: | User and guest | | |
| Pre-conditions: | Login | | |
| Post-conditions: | Show final score, evaluation form | | |
| Flow of events: | **User Action** | | **System Action** |
| 1- login (Enter username, password) | |  |
|  | | 2- System Verify user data  3- show category list |
| 4- User will choose category from category list. | |  |
|  | | 5- System will retrieve the games for this category. |
| 6- Choose a game | |  |
|  | | 7- Show tips for playing |
| * + 1. Start playing | |  |
|  | | 9-Update status of a game (score, level)  10-show final score  11-Show evaluation form |
| * + 1. If user want to evaluate this game, rating and adding comments for the game | |  |
| Alternative path | | * + 1. System update rate and comments |
| 14- If user want to skip this step, Press close button. | |  |
|  | | 15- System will hide this form and tell him his greetings |
| Exceptions: | **User Action** | | **System Action** |
| 1- login (Enter username, password) | |  |
|  | | 2- user is invalid, system rejects user |
| Includes: | Playing game includes login | | |
| Extends: | Playing game extend evaluation | | |
| Notes and Issues: |  | | |
| * Create game | |  | |
| Use Case ID: | 2 | | |
| Use Case Name: | Create game | | |
| Actors: | Teacher | | |
| Pre-conditions: | Login | | |
| Post-conditions: | Availability on the website. | | |
| Flow of events: | **User Action** | | **System Action** |
| 1- login (Enter username, password) | |  |
|  | | 2- System Verify user data  3- show category list |
| 4- User choose category from categories list | |  |
|  | | 5-Show tips for creating |
| 6-create the game | |  |
|  | | 7-check requirements.  8-Add the game to approval Queue waiting admin. |
| Exceptions: | User Action | | System Action |
| 1- login (Enter username, password) | |  |
|  | | 2- user is invalid, system rejects user |
| Includes: | Creating game includes login | | |
| Exceptions: | **User Action** | | **System Action** |
| 1. Building a game | |  |
|  | | 1. game is invalid because requirements, system will tell the user it is invalid and shows retrying or cancel. |
|  | 3- if the teacher chooses retry, the system will repeat again. | |  |
| Includes: | Creating a game includes check requirements, approve new game. | | |
| Notes and Issues: |  | | |

* Update game

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 3 | |
| Use Case Name: | Update game | |
| Actors: | Teacher | |
| Pre-conditions: | Teacher has the game to be updated | |
| Post-conditions: | Availability on the website. | |
| Flow of events: | **User Action** | **System Action** |
| 1-login (Enter username, password) |  |
|  | 2-System Verify user data  3-show teacher games list |
| 4-choose game to update  5-building game |  |
|  | 6-check requirements.  7- add the game to the queue for admin approval.  8- if admin approved Replace the current game . |
| Exception: | User Action | System Action |
| 1- login (Enter username, password) |  |
|  | 3-User is invalid, system rejects user. |
| Exception: | **User Action** | **System Action** |
| 1. Replace current game |  |
|  | 1. game is invalid because requirements, system will tell the user it is invalid and shows retrying or cancel. |
|  | 3- if the teacher chooses retry, the system will repeat again. |  |
| Includes: | Updating a game includes requirements, approve new game. | |
| Notes and Issues: |  | |

* Reply to comment

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 4 | |
| Use Case Name: | Reply to comment | |
| Actors: | Teacher | |
| Pre-conditions: | Login | |
| Post-conditions: | Reply Updated | |
| Flow of events: | **User Action** | **System Action** |
| 1-login (Enter username, password) |  |
|  | 2-System Verify user data |
| 3-press notification button. |  |
|  | 3-show notification list |
| 4-choose specific comment |  |
|  | 5- Show the comments of the chosen game |
| 6- Add reply |  |
|  | 7- update reply to website |
| Exception: | User Action | System Action |
| 1- login (Enter username, password) |  |
|  | 2- user is invalid, system rejects user |

* Delete game

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 5 | |
| Use Case Name: | Delete Game | |
| Actors: | Teacher | |
| Pre-conditions: | Log in | |
| Post-conditions: | Game deleted | |
| Flow of events: | **User Action** | **System Action** |
| 1-login (Enter username, password) |  |
|  | 2-System Verify user data |
| 3-Press delete button |  |
|  | 4-Show list of games |
| 5-Choose and delete game |  |
|  | 6-Delete game and update site |
| Exception: | User Action | System Action |
| 1- login (Enter username, password) |  |
|  | 2-user is invalid, system rejects user. |

# Filter Game

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 6 | |
| Use Case Name: | Filter game | |
| Actors: | Admin | |
| Pre-conditions: | Log in | |
| Post-conditions: | Filter Updated | |
| Flow of events: | **User Action** | **System Action** |
| 1-login (Enter username, password) |  |
|  | 2-System Verify user data |
| 3-Press filter button |  |
|  | 4-Show category list |
| 5-Choose category |  |
|  | 6-Show category list arranged |
| 7-Choose filter game  8-Check requirements  9-Press delete button |  |
|  | 10-filter select the game and update website |
| Exception: | User Action | System Action |
| 1- login (Enter username, password) |  |
|  | 2-user is invalid, system rejects user |

# View Profile

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 7 | |
| Use Case Name: | View Profile | |
| Actors: | User | |
| Pre-conditions: | Log in | |
| Post-conditions: | User has been viewed profile | |
| Flow of events: | **User Action** | **System Action** |
| 1-login (Enter username, password) |  |
|  | 2-System Verify user data |
| 3-Press view profile button |  |
|  | 4-Show data of user (Profile Picture, Name, Change password, etc.) |
| Exception: | User Action | System Action |
| 1-login (Enter username, password) |  |
|  | 2-user is invalid, system rejects user |

# GitHub Link

https://github.com/shriefanwer/Game\_educational

# Ownership Report

|  |  |
| --- | --- |
| **Item** | **Owners** |
|  |  |
|  |  |