Contents

[Team 2](#_Toc402452669)

[Document Purpose and Audience 2](#_Toc402452670)

[Introduction 2](#_Toc402452671)

[Software Purpose 2](#_Toc402452672)

[Software Scope 3](#_Toc402452673)

[Definitions, acronyms, and abbreviations 3](#_Toc402452674)

[Requirements 4](#_Toc402452675)

[Functional Requirements 4](#_Toc402452676)

[Non Functional Requirements 5](#_Toc402452677)

[System Models 6](#_Toc402452678)

[Use Case Model 6](#_Toc402452679)

[Use Case Tables 6](#_Toc402452680)

# Team

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Email** | **Mobile** |
| 20140247 | Mahmoud Samy | m.sami11111995@gmail.com | 01205787165 |
| 20140271 | Mustafa Muhamed | Mustafa\_xd23@yahoo.com | 01280969556 |
| 20140255 | Mahmoud Amin | Mahmoudamin502@gmail.com | 01026510565 |

# Document Purpose and Audience

***Audience***

Our audience are Project manager, CEO and developers are included in our audience and anyone who have new ideas to enhance this project.

***Purpose***

This document will illustrate our efforts to develop new websites, it will also abstract the functionalities of these websites and how it will help others to learn more about specific topics, this document is purposed to stakeholders, developers and anyone interested in this project, throughout this document we will have a brief view about some definitions, requirements and use cases of all functionalities of this project.

# Introduction

## *Software Purpose*

This software will target student from age 8 to 20 who are interested in learning new concepts throughout playing a simple game to show the core knowledge of a specific topic. First he must sign in either as a student or as a teacher then he will choose from a variety of many topic. As a teacher, he will be able to create a new game to teach students.

## *Software Scope*

This is a website application split majorly into student usage or teacher usage with the ability to search for a specific game. Any student can choose a category, join a class, play a game, add a comment on it and collect points. As a teacher, he can create a class and create a game out of

4 options:

1. Multiple choices.
2. Matching the same.
3. True & False.
4. Complete the missing.

Edit a game or remove it and comment on it. Each teacher will identify the total points for each game he will create. Each class has a certain number of games to go from a game to another in the same class you have to collect a certain number of points. For example, to go from a beginner class (Class A) to an intermediate class (Class B) created under the same category you have to collect a certain number of points and the prerequisite classes predefined by the teacher who created the (class B).

## Definitions, acronyms, and abbreviations

***Definitions***

***Use case:*** it’s a software and a system engineering that illustrate how the user can utilize the system to accomplish a specific goal.

***Use case diagram:*** it’s a graphic depiction describes the interaction between elements of a system.

## *Functional Requirements:* describe the interaction between the user the system.

***Non-Functional Requirements:***Aspects not directly related to functional behavior.

***Usability:*** the ease with which actors can use a system to perform a function

***Reliability:*** is the ability of system or component to perform its required functions under stated condition for a specific period of time.

***Performance:*** requirements are concerned with quantifiable attributes of the system.

***Supportability:*** requirements are concerned with the ease of changes to the system after deployment

# 

# Requirements

## Functional Requirements

This web application has different type of users who can utilize it with diversity in interaction and usage. Users can be either a student or teacher.

***As a user:***

* ***Create an account***

The user must have an account so he can utilize the system if he doesn’t then he can create one by signing up.

* ***Log in***

the student must sign in so he can deal with the system.

* ***Play a game***

After joining the class the user can choose a specific game to play with.

* ***Search for a game***

The user can search for a specific game in the web application.

* ***Ask for hint***

If the user faced a problem during the game he can ask for a hint to help him continue playing the game.

* ***Write a comment***

Player will be able to write comment on any game an as will replay on a comment.

***As a student:***

* ***Game rate***

The user should give a rate after playing the game.

* ***Save score***

The system will ask the player after finishing the game if he want to save his score in this game and add it to the total score of the player.

* ***Join a class***

The user can join a class which includes more than one game.

***As a teacher:***

* ***Create a game***

The user can create a game with specific rules.

* ***Edit a game***

The game can be edited by the teacher and he can change any rule inside the game.

* ***Remove a game***

The teacher has the option to remove the game he created inside the web application.

* ***Create a class***

Multiple games can be included inside a class by the teacher.

* ***Determine class prerequisites***

Teacher must determine the prerequisites for the students to join a class.

***System***

* ***calculate score***

System will calculate score according to user performance.

* ***verify the user***

System will verify the user’s data.

## *Non-Functional Requirements*

* ***Usability***

The user will use the web application in easy way. The web application will provide good interface for the user to deal with it, some of educational concepts for students to understand and the user will be able to

* ***Reusability***

The web application source code will be reused for developing a similar web application in case of the project failure.

* ***Capacity***

Capacity of the database shall be large enough to hold 500,000 user as a begging and it will be extended if the number of users exceeded more than 500,000 users.

* ***Recover and Backup***

The application will be able to recover its state in case of any emergency breakdowns, also it will use hot backup technique.

* ***Security***

The user’s data will be secured with the user’s account name and password and it will be done using a specific criteria for the password.

* ***Reliability & Availability***

Teacher should test his game before he launch it so the game’s result is reliable. The system shall has his own server which connected to the internet 24 hours a day by a high speed connection up to 20 mbps.

* ***Performance***

Using a high speed connection the response time will also be very fast.

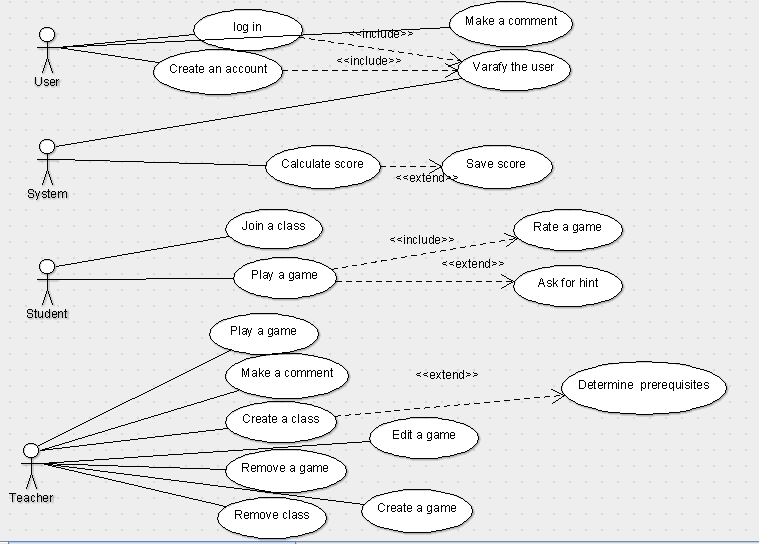
* ***Supportability***

When a player faces a problem with a game he will make a comment about it and any user will reply to it

Within 24 hours.

# System Models

## Use Case Model



## Use Case Tables

|  |  |  |
| --- | --- | --- |
| Use Case ID: | SI101 | |
| Use Case Name: | Log in | |
| Actors: | Student, Teacher | |
| Pre-conditions: | Available Internet Connection | |
| Post-conditions: | System is available for using | |
| Flow of events: | **User Action** | **System Action** |
| 1- The User opens the website application. |  |
|  | 2- System asks for Username and Password |
| 3- The User Enters Username and Password. |  |
|  | 4- System Verify user data.  5-System open the user account. |
| Exceptions: | **User Action** | **System Action** |
| 1- The User opens the website application. |  |
|  | 2- System asks for Username and Password |
| 3- The User Enters Username and Password. |  |
|  | 4- Username or password is invalid  5- System display un error message for user. |
| Include: | Verification | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | SU102 | |
| Use Case Name: | Create an account | |
| Actors: | Student, teacher | |
| Pre-conditions: | Available Internet | |
| Post-conditions: | Actors can use the system | |
| Flow of events: | **User Action** | **System Action** |
| 1- The User opens the website application. |  |
|  | 2- System asked for new username, first name, last name, gender, birthday, email and password |
| 3- User Enter new all requirements |  |
|  | 4- System check if the Username is available  6-System check if password is identify requirements  7- System add the user to the system database  8-System open user page |
| Exceptions: | **User Action** | **System Action** |
| 1- The User opens the website application.  2- The User Enter New Username or Password |  |
|  | 2- Username is unavailable or password is incorrect  4- System show massage to user ”invalid username or password” |
| Includes: | Verification | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | CM103 | |
| Use Case Name: | Write a Comment | |
| Actors: | Student, Teacher | |
| Pre-conditions: | Inside the game page | |
| Post-conditions: | The comment is done | |
| Flow of events: | **User Action** | **System Action** |
| 1- The User opens the website application. |  |
|  | 2- System asks for Username and Password |
| 3- The User Enters Username and Password. |  |
|  | 4- System Verify user data. |
| 5-The Student or The Teacher chooses a game to play |  |
|  | 6-The system starts the game. |
| 7- The Student or The Teacher wants to add comment to the game.  8- The Student or The Teacher presses the comment area.  9- The Student or The Teacher starts writing a comment. |  |
|  | 10- System save the comment. |
| Include: |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| Use Case ID: | PG104 | |
| Use Case Name: | Play a Game | |
| Actors: | Student, teacher | |
| Pre-conditions: | User make sign in | |
| Post-conditions: | Student can take points of this game and rate it | |
| Flow of events: | **User Action** | **System Action** |
| 1- The User opens the website application. |  |
|  | 2- System asks for Username and Password |
| 3- The User Enters Username and Password. |  |
|  | 4- System Verify user data. |
| 3- Teacher or Student chooses category |  |
|  | 4- System retrieve the category |
| 5- Teacher or Student chooses game |  |
|  | 6- System retrieve the game to be played |
| Exceptions: | **User Action** | **System Action** |
| 1- The User Enters Username and Password. |  |
|  | 2-System shows error message informing the user the incorrect Username or ID. |
| Include: | Rate, calculate score | |
| Extend: | Comment, Ask for hint | |
|  |  |  |
| Use Case ID: | CG301 | |
| Use Case Name: | Create a Game | |
| Actors: | Teacher | |
| Pre-conditions: | Login as a teacher | |
| Post-conditions: | The game will be available for students | |
| Flow of events: | **User Action** | **System Action** |
| 1- The User log in as a Teacher. |  |
|  | 2- The System retrieve The Teacher account. |
| 3-The Teacher determines to create a game. |  |
|  | 4-The System asks for the category of the game |
| 5- Teacher choose the category of the game (math, chemistry, code...)  6-Teacher choose classification of the game. |  |
|  | 7- System add the game to the system |
| Exceptions: | **User Action** | **System Action** |
| Includes: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | RM302 | |
| Use Case Name: | Remove a game | |
| Actors: | Teacher | |
| Pre-conditions: | Sign in as a teacher , Teacher in the game page | |
| Post-conditions: | The game will not be available to be played | |
| Flow of events: | **User Action** | **System Action** |
| 1- The User log in as a Teacher. |  |
|  | 2- The System retrieve The Teacher account. |
| 3-The Teacher determines to remove a game. |  |
|  | 4-The System asks for the category of the game |
| 5- Teacher choose the category of the game (math, chemistry, code...)  6-Teacher choose the game.  7-Teacher presses the Remove button. |  |
|  | 8- System remove the game  9-System show message “game is removed” |
| Exceptions: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | ED303 | |
| Use Case Name: | Edit a game | |
| Actors: | Teacher | |
| Pre-conditions: | Login as a teacher, Teacher in the game page | |
| Post-conditions: | The game will be edit | |
| Flow of events: | **User Action** | **System Action** |
| 1- The User log in as a Teacher. |  |
|  | 2- The System retrieve The Teacher account. |
| 3-The Teacher determines to edit a game. |  |
|  | 4-The System asks for the category of the game |
| 5- Teacher choose the category of the game (math, chemistry, code...)  6-Teacher choose the game.  7-Teacher starts editing. |  |
|  | 8-System save the game.  9-System show message “game is edit” |
| Exceptions: |  |  |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | CC304 | |
| Use Case Name: | Create a class | |
| Actors: | Teacher | |
| Pre-conditions: | Sign in as teacher | |
| Post-conditions: | There is a new class for students | |
| Flow of events: | **User Action** | **System Action** |
| 1- The User log in as a Teacher. |  |
|  | 2- The System retrieve The Teacher account. |
| 3-The Teacher determines to create a class. |  |
|  | 4-The System asks The Teacher for Class Name and ID. |
| 4-The Teacher enters Class Name and ID. |  |
|  | 5-System show message “You have create class successfully”  6-System open the created class. |
| Exceptions: | **User Action** | **System Action** |
| 1- Teacher enter Class Name and ID. |  |
|  | 2- Name or ID are unavailable  3- System show message” unavailable Name or ID ”. |
| Extend: | Add students, determine prerequisites | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | DCP305 | |
| Use Case Name: | Determine class prerequisites | |
| Actors: | Teacher | |
| Pre-conditions: | Sign in as teacher | |
| Post-conditions: | Class has prerequisites | |
| Flow of events: | **User Action** | **System Action** |
| 1- Teacher enters the Class Name and ID. |  |
|  | 2-System verify the input data. |
|  | 3-Teacher Determines class prerequisites |  |
|  |  | 4-System add class prerequisites.  5-System save this changes in database. |
| Exceptions: | **User Action** | **System Action** |
| 1- Teacher enters the Class Name and ID. |  |
|  | 2-System verify the input data.  3-Class Name or ID is incorrect.  4-System shows error message. |
| Includes: | Create a class | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | RG201 | |
| Use Case Name: | Game rate | |
| Actors: | Student | |
| Pre-conditions: | Student already play the game | |
| Post-conditions: | The game will be rated | |
| Flow of events: | **User Action** | **System Action** |
| 1- The User log in as a Student |  |
|  | 2- System retrieve The Student account. |
| 3- The Student choose a game to play. |  |
|  | 4- System starts the game. |
| 5-The Student decides to rate the game.  6-The Student presses the ‘Rata the game’ button. |  |
|  | 7-System shows the rate to The Student. |
| 8-The Student rates the game. |  |
|  | 9-System saves the rate. |
| Exceptions: |  |  |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | JC202 | |
| Use Case Name: | Join a class | |
| Actors: | Student | |
| Pre-conditions: | Sign in as student | |
| Post-conditions: | Students are joined to a class | |
| Flow of events: | **User Action** | **System Action** |
| 1- The user log in as a Student. |  |
|  | 2- System retrieve The Student account. |
| 3-The Student searches for the class by Name OR ID. |  |
|  | 4- System display a menu by every class has this name or the class that has the unique ID. |
| 5-The Student presses ‘Join’ button |  |
|  | 6-System add the student into the class.  7-System save the update in database. |
| Exceptions: | **User Action** | **System Action** |
| 1- Student search for class by name or ID. |  |
|  | 2-System show message” There is no class by this name”. |
| Includes: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | SV203 | |
| Use Case Name: | Save score | |
| Actors: | Student | |
| Pre-conditions: | Sign in as student, student already play the game | |
| Post-conditions: | Student has score for this game | |
| Flow of events: | **User Action** | **System Action** |
| 1-The User log in as a Student. |  |
|  | 2- System retrieves The Student account. |
| 3-The Student choose a game to play. |  |
|  | 4-System starts the game. |
| 5-The Student starts playing and scoring points.  6-The Student determines to save the score.  7-The Student presses “Save” button. |  |
|  | 8-System saves the score. |
| Exceptions: | **User Action** | **System Action** |
|  |  |
|  |  |
| Includes: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | AH204 | |
| Use Case Name: | Ask for hint | |
| Actors: | Student | |
| Pre-conditions: | Sign in as student, student is play the game now | |
| Post-conditions: | Student new how to play this game | |
| Flow of events: | **User Action** | **System Action** |
| 1-The User log in as a Student. |  |
|  | 2- System retrieves The Student account. |
| 3-The Student choose a game to play. |  |
|  | 4-System starts the game. |
| 5-The Student has obstacle inside the game and wants to pass it.  6-The Student wants a hint for the game.  7-The Student presses ‘Hint’ button. |  |
|  | 8-System receives The Student request for a hint.  9-System shows a hint for the game to The Student. |
| 10-The Student continues playing the game with the hint. |  |
| Exceptions: | **User Action** | **System Action** |
| 1- The Student presses ‘Hint’ button. |  |
|  | 2-System shows message:  “You already received a hint in this level, you can’t ask for a hint again in this level”. |
| Includes: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 401 | |
| Use Case Name: | Verify the user | |
| Actors: | System | |
| Pre-conditions: | Sign in, sign up | |
| Post-conditions: | Let User access his account | |
| Flow of events: | **User Action** | **System Action** |
| 1- The User opens the website application. |  |
|  | 2- System asks for Username and Password |
| 3- The User Enters Username and Password. |  |
|  | 4- System Verify user data. |
| 5-The Student choose a game to play. |  |
|  | 6-System starts the game. |
| 7-System starts playing the game. |  |
|  | 8-System will compare user data to the database if he is a member or not.  9-System retrieve the user account as a Teacher or Student.  10-System will open user account. |
| Exceptions: | **User Action** | **System Action** |
| 1- The User Enters Username and Password. |  |
|  | 2-System shows error message  informing incorrect Username or ID. |
| Includes: |  | |
|  |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 402 | |
| Use Case Name: | Calculate score | |
| Actors: | System | |
| Pre-conditions: | User already play the game | |
| Post-conditions: | Game is marked as played | |
| Flow of events: | **User Action** | **System Action** |
| 1- The User opens the website application. |  |
|  | 2- System asks for Username and Password |
| 3- The User Enters Username and Password. |  |
|  | 4- System Verify user data. |
| 4-The Student choose a game to play. |  |
|  | 5-System starts the game. |
| 6-System starts playing the game. |  |
|  | 1-System will calculate every correct move in the game until The Student terminates the game.  2-System will extend save score. |
| Exceptions: | **User Action** | **System Action** |
|  |  |
| Extend: | Save score | |