iSNAP2Change

Internet-Based Intervention Program

Software Requirements Specification

2.0

April 22, 2016

Haokang An, Weijun Wu

Prepared for

Clinical Practice Unit at SA Health

2016

Contents

[**1. Introduction** 4](#_Toc449207397)

[**1.1 Purpose** 4](#_Toc449207398)

[**1.2 Scope** 4](#_Toc449207399)

[**1.3 Definitions, Acronyms, and Abbreviations** 4](#_Toc449207400)

[**2. General Description** 5](#_Toc449207401)

[**2.1 Product Perspective** 5](#_Toc449207402)

[**2.2 System Architecture** 5](#_Toc449207403)

[**2.3 Product Functions** 5](#_Toc449207404)

[**2.4 User Characteristics** 6](#_Toc449207405)

[**2.5 General Constraints** 6](#_Toc449207406)

[**2.6 Assumptions and Dependencies** 6](#_Toc449207407)

[**3. Specific Requirements** 6](#_Toc449207408)

[**3.1 External Interface Requirements** 6](#_Toc449207409)

[**3.1.1 User Interfaces** 6](#_Toc449207410)

[**3.1.2 Hardware Interfaces** 7](#_Toc449207411)

[**3.1.3 Software Interfaces** 7](#_Toc449207412)

[**3.2 Functional Requirements** 7](#_Toc449207413)

[**3.2.1 Login** 7](#_Toc449207414)

[**3.2.2 Signup** 8](#_Toc449207415)

[**3.2.3 Display Welcome Content** 8](#_Toc449207416)

[**3.2.4 Display Avatar, Route Map and Status** 9](#_Toc449207417)

[**3.2.5 Edit Avatar Category** 9](#_Toc449207418)

[**3.2.6 Display Weekly Content and Bonus Task** 9](#_Toc449207419)

[**3.2.7 Complete Task: Video Watching** 10](#_Toc449207420)

[**3.2.8 Complete Task: Game Playing** 10](#_Toc449207421)

[**3.2.9 Complete Task: Quiz** 11](#_Toc449207422)

[**3.2.10 Complete Task: Short Answer Question** 11](#_Toc449207423)

[**3.2.11 Complete Task: Bonus** 12](#_Toc449207424)

[**3.2.12 Popup Weekly Score Achieved** 12](#_Toc449207425)

[**3.2.13 Show Final Result and Reward** 13](#_Toc449207426)

[**3.2.14 Display Scoreboard** 13](#_Toc449207427)

[**3.2.15 Display Student Progress View** 13](#_Toc449207428)

[**3.2.16 List Short Answer Question Grading Status** 14](#_Toc449207429)

[**3.2.17 List Bonus Question Grading Status** 14](#_Toc449207430)

[**3.2.18 Grade/Regrade Students’ Short Answer Questions** 15](#_Toc449207431)

[**3.2.19 Grade/Regrade Students’ Bonus Tasks** 15](#_Toc449207432)

[**3.2.20 Discussion Board for Students** 15](#_Toc449207433)

[**3.2.21 Discussion Board for Teachers** 16](#_Toc449207434)

[**3.2.22 Customized Common Error Pages** 16](#_Toc449207435)

[**3.3 Use Cases** 17](#_Toc449207436)

[**3.3.1 Login** 17](#_Toc449207437)

[**3.3.2 Signup** 17](#_Toc449207438)

[**3.3.3 Display Welcome Content** 18](#_Toc449207439)

[**3.3.4 Display Avatar, Route Map and Status** 19](#_Toc449207440)

[**3.3.5 Edit Avatar Category** 19](#_Toc449207441)

[**3.3.6 Display Weekly Content and Bonus Task** 20](#_Toc449207442)

[**3.3.7 Complete Task: Video Watching** 21](#_Toc449207443)

[**3.3.8 Complete Task: Game Playing** 21](#_Toc449207444)

[**3.3.9 Complete Task: Quiz** 22](#_Toc449207445)

[**3.3.10 Complete Task: Short Answer Question** 23](#_Toc449207446)

[**3.3.11 Complete Task: Bonus** 23](#_Toc449207447)

[**3.3.12 View Scoreboard** 24](#_Toc449207448)

[**3.3.13 View Student Progress** 24](#_Toc449207449)

[**3.3.14 List Short Answer Question Grading Status** 25](#_Toc449207450)

[**3.3.15 Grade/Regrade Students’ Short Answer Questions** 25](#_Toc449207451)

[**3.3.16 List Bonus Question Grading Status** 26](#_Toc449207452)

[**3.3.17 Grade/Regrade Students’ Bonus Questions** 27](#_Toc449207453)

[**4. Database Design** 28](#_Toc449207454)

[**4.1 ER Diagram** 28](#_Toc449207455)

[**4.2 Schema** 29](#_Toc449207456)

# **1. Introduction**

For this section, it gives a general description and overview of everything included in SRS document. The section also describes the purpose of this requirement document and provides a list of abbreviations and definitions.

## **1.1 Purpose**

The purpose of the SRS document is to give detailed description of requirements for the “iSNAP2Change” web application. The document is intended to describe the purpose of the system and how to design and develop the system. The document will detail the analysis and design model of the system. It will help the team to implement and check the function of the system.

## **1.2 Scope**

The Software Requirements Specification describes all requirements in the document. The “iSNAP2Change” web application is a highly interactive application. It is a web application that provides the students in the school with a wide range of supplementary material including games, media content and ability to interact. Also, teachers can monitor students’ progress and grade their short answer questions and bonus tasks.

The “iSNAP2Change” web application is supposed to have following features.

1) The product is designed for the iSNAP2Change program and runs all day.

2) The system provides log on facility to the users.

3) The system allows students to complete weekly tasks including video watching, game playing, quiz and short answer questions.

4) The system allows students to complete bonus tasks.

4) The system allows students to check their avatars’ status including progress, score, rank and so on.

5) The system allows teachers to monitor students’ progresses and grade students’ short answer questions and bonus tasks.

6) The system allows teachers to answer students’ questions through the forum.

These features that are described in the SRS document are for the future phases of the software development cycle. These features meet the requirement of all users. Thus, the success criteria for the system is based in features in the document are implemented in the system.

## **1.3 Definitions, Acronyms, and Abbreviations**

|  |  |
| --- | --- |
| Term | Definition |
| iSNAP2Change | Smoking, nutrition, alcohol and physical activity to change |
| User | Someone who uses the system |
| UML | Unified Modeling Language, a standard language to model the system. |

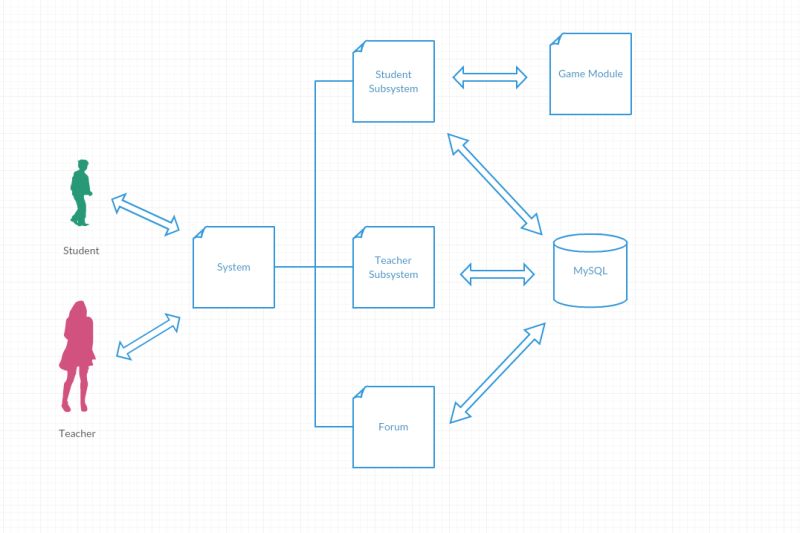
# **2. General Description**

This section will provide a blueprint of the whole system. The section will describe the basic functional of the system and how it can be used. The contexts also describe how users use the system. Finally, the constraints and other assumptions of the system will be described.

## **2.1 Product Perspective**

“iSNAP2Change” web application is to be used by students to interact with a wide range of supplementary material including games, media content and etc. Also, this web application is to be used by teachers to monitor students’ progresses and grade their short answer questions and bonus tasks. The final goal is to efficiently and effectively engage and reach the intended audience and achieve the youth health interventions.

## **2.2 System Architecture**



The system has three major modules: student subsystem, teacher subsystem and forum (discussion board). We will integrate game module as well as MySQL database with this system. Forum is a relatively independent part powered by Discuz! forum software.

## **2.3 Product Functions**

The “iSNAP2Change” web application provides a wide range of supplementary material including games, media content and ability to interact.

With the application, students will be able to log in the system to complete weekly tasks as an avatar. The route map will show the avatar’s progress. The side bar will show the avatar’s other status including score, rank and so on. As for the weekly tasks, there are mainly four types which are video watching, game playing, quiz and short answer question respectively. Also, students can complete bonus tasks.

With the application, teachers will be able to log in the system to monitor students’ progress. Also, teachers will be able to grade students’ short answer questions and bonus tasks. What’s more, teachers can answer students’ questions through the forum.

## **2.4 User Characteristics**

Student: The student can log in to the system and complete weekly tasks and bonus tasks. Also, he or she can check the avatar’s status including score, progress, rank and so on.

Teacher: The teacher can monitor students’ progresses. Also, he or she can grade students’ short answer questions. What’s more, he or she can answer questions asked by students through the forum.

## **2.5 General Constraints**

The system will use WAMP (Windows + Apache + MySQL + PHP) framework to develop and it is based on the object oriented analysis and design mode. The system shall use the MySQL database. The user information, avatar information, weekly content, bonus tasks as well as contents on discussion board will be stored in the database. The system will be constrained by the capacity of the database. Also, the system will be constrained by the compatibility and integration with games.

The front end is under the context of English. The development of the front end will use HTML, CSS, JavaScript and Bootstrap.

## **2.6 Assumptions and Dependencies**

The students and teachers have the experience on using computers and web applications. The system can interact with the game database.

# **3. Specific Requirements**

## **3.1 External Interface Requirements**

### **3.1.1 User Interfaces**

For every page, its layout and style must be very explicit, user-friendly and attractive to users, especially to students. When the user visits the website, there will be a welcome page with login button which linked to login page. On the login page, the user will be asked to input his or her username and password.

For students, if the username and password is verified, the interface will be turned into the core page with avatar, route map and status where the user can finish the weekly task. For teachers, if the username and password is verified, the interface will be turned into the page with students’ progress view.

There will be buttons in sidebar so that the user can navigate through the website. A collection of GUI interface will be well designed to engage users. If the user input anything malformed, the interface will display an error message and ask the user to retry.

For every GUI interface, all the color and design will be very colorful, engaging and interactive. The font style as well as the graphics will be in a well-organized manner that is suitable for students.

Main page can only be accessed after verifying the registered user’s password and username, it provides a better security control.

The system be simple and plain to understand and operate for users, especially for students.

### **3.1.2 Hardware Interfaces**

MySQL server will be used for storing data, which supports relational association and ACID. PDO will be used to connect data to the system to improve compatibility or scalability since this project is expected to expanded nationally.

### **3.1.3 Software Interfaces**

The system can be divided into three major function modules:

**Student subsystem** for weekly tasks **–** implemented using MySQL as storage, which can be easily synchronized and expanded in the future.

**Teacher** **subsystem** to grade students’ tasks and monitor students’ progress – use MySQL to support ACID.

**Discussion board –** use MySQL which is one of the two databases officially supported by Discuz! forum software.

There are also some additional parts of the system:

**Scoreboard** – show the scores and ranks among students in each class

**Welcome content –** default welcome webpage which is accessible for students, teachers as well as other visitors

## **3.2 Functional Requirements**

### **3.2.1 Login**

#### 3.2.1.1 Introduction

The system shall allow students/teachers to log in based on username, password and token.

#### 3.2.1.2 Inputs

Username and password.

#### 3.2.1.3 Processing

The system checks the validity.

#### 3.2.1.4 Outputs

If the user logs in successfully, the system will display main page for students or main page for teachers accordingly. Otherwise, the system will display a message which indicates the failure of log in operation.

#### 3.2.1.5 Error Handling

If the user name or password is not correct, the system will display an error message. Then the system will give the user two options to input again or cancel the login operation.

### **3.2.2 Signup**

#### 3.2.2.1 Introduction

The system shall allow students/teachers to register.

#### 3.2.2.2 Inputs

Username, password, token, real name, age, gender, class, school. For students, they will also choose avatar category they prefer.

#### 3.2.2.3 Processing

The system validates users’ input and register for different types of users (students/teachers).

#### 3.2.2.4 Outputs

The system displays a message indicating whether the operation of signup is successful or not.

#### 3.2.2.5 Error Handling

The system will display an error message if the signup operation fails due to incomplete information or wrong token. Then the system will give the user two options to try again or cancel the signup operation.

### **3.2.3 Display Welcome Content**

#### 3.2.3.1 Introduction

The system shall allow students, teachers and other visitors to see the welcome contents.

#### 3.2.3.2 Input

None.

#### 3.2.3.3 Processing

The system retrieves welcome contents.

#### 3.2.3.4 Outputs

The system displays the welcome contents (videos, images and short messages).

#### 3.2.3.5 Error handling

The system displays an error message if the display of welcome contents is abnormal.

### **3.2.4 Display Avatar, Route Map and Status**

#### 3.2.4.1 Introduction

The student subsystem shall allow students to see an avatar, a route map and the avatar’s status (current progress, score, rank and etc.).

#### 3.2.4.2 Inputs

None.

#### 3.2.4.3 Processing

The system retrieves avatar’s status including current progress, score, rank and etc.

#### 3.2.4.4 Outputs

The system displays an avatar, a route map and the avatar’s status (current progress, score, rank and etc.).

#### 3.2.4.5 Error Handling

The system displays an error message if the display of avatar, route map and status is abnormal.

### **3.2.5 Edit Avatar Category**

#### 3.2.5.1 Introduction

The student subsystem shall allow students to change his/her avatar category as they wish.

#### 3.2.5.2 Input

The new avatar category the student chooses.

#### 3.2.5.3 Processing

The system updated the avatar category for the student.

#### 3.2.5.4 Outputs

The avatar category and the current avatar image gets updated.

#### 3.2.5.5 Error Handling

The system will display an error message if the request failed.

### **3.2.6 Display Weekly Content and Bonus Task**

#### 3.2.6.1 Introduction

The student subsystem shall allow students to see the weekly content including video watching task, game playing task, quiz task, short answer question task as well as bonus task.

#### 3.2.6.2 Input

None.

#### 3.2.6.3 Processing

The system retrieves different types of tasks (video watching, game playing, quiz, short answer question as well as bonus task).

#### 3.2.6.4 Outputs

The system displays the weekly content including video watching task, game playing task, quiz task and short answer question task as well as bonus task.

#### 3.2.6.5 Error handling

The system will display an error message if the display of weekly content is abnormal.

### **3.2.7 Complete Task: Video Watching**

#### 3.2.7.1 Introduction

The student subsystem shall allow students to complete the video watching task.

#### 3.2.7.2 Inputs

Students’ clicks on the button which indicate the beginning of the video watching task.

Students’ clicks on the button which indicate the completion of the video watching task.

#### 3.2.7.3 Processing

The system captures students’ clicks which indicate the beginning of video watching task and the system displays the video. Next, the system captures students’ clicks which indicate the completion of video watching task. Finally, the systems updates students’ score, rank, avatar and other information if needed.

#### 3.2.7.4 Outputs

The system displays a video. The system displays the result of the video watching task as well.

#### 3.2.7.5 Error Handling

The system will display an error message if the video cannot be displayed normally.

### **3.2.8 Complete Task: Game Playing**

#### 3.2.8.1 Introduction

The student subsystem shall allow students to complete the game playing task.

#### 3.2.8.2 Input

Students’ clicks on the button which indicate the beginning of the game playing task.

Students’ clicks on the button which indicate the completion of the game playing task.

#### 3.2.8.3 Processing

The system captures students’ clicks which indicate the beginning of game playing task. Then the system displays the game. Next, the system captures students’ clicks which indicate the completion of the game playing task. In addition, the system captures the result of the game playing. Finally, the systems updates students’ score, rank, avatar and other information if needed.

#### 3.2.7.4 Outputs

The system displays the game. Also, the system displays the result of the game playing task.

#### 3.2.8.5 Error handling

The system will display an error message if the game cannot be displayed normally or there is some problem to upload the score.

### **3.2.9 Complete Task: Quiz**

#### 3.2.8.1 Introduction

The student subsystem shall allow students to complete the quiz task.

#### 3.2.9.2 Inputs

Students’ clicks on the button which indicate the beginning of the quiz task.

Students’ clicks on the button which indicate the completion of the quiz task.

Students’ choices on all the questions.

#### 3.2.9.3 Processing

The system captures students’ clicks which indicate the beginning of the quiz task. Then the system displays the quiz. Next, the system captures students’ clicks which indicate the completion of the quiz task. What’s more, the system reads students’ answers for the quiz and automatically grades them. Finally, the system updates students’ score, rank, avatar and other information if needed.

#### 3.2.9.4 Outputs

The system displays the quiz. Also, the system displays the result of the quiz task.

#### 3.2.9.5 Error handling

The system will display an error message if the quiz cannot be displayed normally.

### **3.2.10 Complete Task: Short Answer Question**

#### 3.2.10.1 Introduction

The student subsystem shall allow students to answer short answer questions and submit answers.

#### 3.2.10.2 Inputs

Answers for each short answer questions.

#### 3.2.10.3 Processing

Validate that all the questions are answered and the minimum answer length is met. Answers are saved into database and wait for teacher’s grading and feedback. The system updates students’ score, rank, avatar and other information if needed.

#### 3.2.10.4 Outputs

The system displays a message indicates the answers are submitted and user will receive full score (it might get deducted after teacher’s grading).

#### 3.2.10.5 Error Handling

The system displays error message if any question has not been answered or any answer is too short, and user should revise answers and resubmit them.

The system will display an error message if the short answer section cannot be displayed normally.

### **3.2.11 Complete Task: Bonus**

#### 3.2.11.1 Introduction

The student subsystem shall allow students to finish bonus tasks to get extra points after class.

#### 3.2.11.2 Inputs

Check the checkbox indicating that the student have finished this bonus task.

#### 3.2.11.3 Processing

The system updates students’ score, rank, avatar and other information if needed.

#### 3.2.11.4 Outputs

The system displays a message indicates the submission is successful and user will receive full score (it might get deducted after teacher’s grading).

#### 3.2.11.5 Error Handling

The system will display an error message if the bonus cannot be displayed normally.

### **3.2.12 Popup Weekly Score Achieved**

#### 3.2.12.1 Introduction

When students finish all the tasks during the current weekly phase, the student subsystem shall show a popup to congratulate their achievement.

#### 3.2.12.2 Inputs

No inputs involved. It is automatically triggered when all the tasks finished.

#### 3.2.12.3 Processing

Validate that all the tasks are finished with full score. If so, show the popup window.

#### 3.2.12.4 Outputs

A popup to congratulate their achievement.

#### 3.2.12.5 Error Handling

No error handling involved.

### **3.2.13 Show Final Result and Reward**

#### 3.2.13.1 Introduction

When students finish all the weekly phases, there will be a popup to congratulate their achievement. The student subsystem shall show a final evaluation and result, along with some sweet reward.

#### 3.2.13.2 Inputs

No inputs involved. It is automatically triggered when all the tasks in the last week finished.

#### 3.2.13.3 Processing

Validate that weekly phases are finished. If so, show the popup window.

#### 3.2.13.4 Outputs

A popup to show final evaluation and result, along with some sweet reward.

#### 3.2.13.5 Error Handling

No error handling involved.

### **3.2.14 Display Scoreboard**

#### 3.2.14.1 Introduction

Students can see the scoreboard with everyone’s score and ranks in the same class by clicking “Scoreboard” button within sidebar. We only show top 10 student’s score and ranks.

#### 3.2.14.2 Inputs

No inputs involved.

#### 3.2.14.3 Processing

Retrieve ordered data from database and list them ordered by score.

#### 3.2.14.4 Outputs

Students’ score and ranks in the same class.

#### 3.2.14.5 Error Handling

No error handling involved.

### **3.2.15 Display Student Progress View**

#### 3.2.15.1 Introduction

Teacher can view all the students’ progress in the class by teacher subsystem. It is the default view after the teacher log in or the teacher click “Progress” button in sidebar. Teacher can sort or filter the result by student, week, and finished status.

#### 3.2.15.2 Inputs

No inputs involved.

#### 3.2.15.3 Processing

Retrieve ordered data from database and list them ordered by student name.

#### 3.2.15.4 Outputs

Each student’s progress and score in the class.

#### 3.2.15.5 Error Handling

No error handling involved.

### **3.2.16 List Short Answer Question Grading Status**

#### 3.2.16.1 Introduction

Teacher can view all the short answer questions to grade by clicking “Grading” button in sidebar by teacher subsystem, graded question are still listed but will be put in the tail. Teacher can sort or filter the result by student, week, and graded/finished status.

#### 3.2.16.2 Inputs

No inputs involved.

#### 3.2.16.3 Processing

Retrieve data from database and list them ordered by graded/finished status, time and date.

#### 3.2.16.4 Outputs

All the short answer questions to grade as well as graded questions in the tail.

#### 3.2.16.5 Error Handling

No error handling involved.

### **3.2.17 List Bonus Question Grading Status**

#### 3.2.17.1 Introduction

Teacher can view all the bonus questions to grade by clicking “Bonus” button in sidebar by teacher subsystem, graded question are still listed but will be put in the tail. Teacher can sort or filter the result by student, week, and graded/finished status.

#### 3.2.17.2 Inputs

No inputs involved.

#### 3.2.17.3 Processing

Retrieve data from database and list them ordered by graded/finished status, time and date.

#### 3.2.17.4 Outputs

All the bonus questions to grade as well as graded questions in the tail.

#### 3.2.17.5 Error Handling

No error handling involved.

### **3.2.18 Grade/Regrade Students’ Short Answer Questions**

#### 3.2.18.1 Introduction

Teacher can click any row in the short answer question grading status list to grade certain submission, and it can be regraded when needed. This view is showed by a popup window.

#### 3.2.18.2 Inputs

Score for each question and overall feedback.

#### 3.2.18.3 Processing

Validate the input. Saved score and feedback to database and updates students’ score, rank, avatar and other information if needed.

#### 3.2.18.4 Outputs

Updated short answer question grading status.

#### 3.2.18.5 Error Handling

If any input is malformed, there will be an error message to ask the teacher to modify the inputs and resubmit the grading.

### **3.2.19 Grade/Regrade Students’ Bonus Tasks**

#### 3.2.19.1 Introduction

Teacher will grade students’ bonus tasks, and it can be regraded when needed. This view is showed by a popup window.

#### 3.2.19.2 Inputs

Score and overall feedback.

#### 3.2.19.3 Processing

Validate the input. Saved score and feedback to database and updates student’s score, rank, avatar and other information if needed.

#### 3.2.19.4 Outputs

Updated short answer question grading status.

#### 3.2.19.5 Error Handling

If any input is malformed, there will be an error message to ask the teacher to modify the inputs and resubmit the grading.

### **3.2.20 Discussion Board for Students**

#### 3.2.20.1 Introduction

Students can click “Discussion” to visit the forum, in which they can post question and discuss with each other.

#### 3.2.20.2 Inputs

Title, text, image or other attachment for the post or follow-up discussion.

#### 3.2.20.3 Processing

Validate the input, save the data into database and add the new thread or follow-up discussion in the forum.

#### 3.2.20.4 Outputs

The newly posted or updated thread.

#### 3.2.20.5 Error Handling

If any input is malformed, there will be an error message to ask students to check and revise the content.

### **3.2.21 Discussion Board for Teachers**

#### 3.2.21.1 Introduction

Teachers can click “Discussion” to visit the forum, in which they can post notes, manage and answer students’ posts.

#### 3.2.21.2 Inputs

Title, text, image or other attachment for the post or follow-up discussion.

#### 3.2.21.3 Processing

Validate the input, save the data into database and add the new thread or follow-up discussion in the forum.

#### 3.2.21.4 Outputs

The newly posted or updated thread.

#### 3.2.21.5 Error Handling

If any input is malformed, there will be an error message to ask students to check and revise the content.

### **3.2.22 Customized Common Error Pages**

#### 3.2.22.1 Introduction

Common error pages when there are problems to visit any page.

#### 3.2.22.2 Inputs

No input involved.

#### 3.2.22.3 Processing

Retrieve customized corresponding error page.

#### 3.2.22.4 Outputs

Corresponding error page.

#### 3.2.22.5 Error Handling

No error handling involved.

## **3.3 Use Cases**

### **3.3.1 Login**

|  |
| --- |
| Use case: Login |
| ID:1 |
| Brief description:  Students or Teachers log in the system. |
| Primary actors:  Students or Teachers |
| Secondary actors:  None |
| Preconditions:  1.The system runs normally. |
| Main flow:  1. The use case starts when the system runs normally.  2. The system asks the student or teacher to input the username, password and token.  3. The system displays main page for students or main page for teachers accordingly if username and password match. |
| Post Conditions:  The student or teacher enters different main pages accordingly. |
| Alternative flows:  3.1. If the username or password is not correct  3.1.a. The system asks the student or teacher to enter his or her details comprising username, password and token again for verifying.  3.1.b The system validates the student or teacher details. |

### **3.3.2 Signup**

|  |
| --- |
| Use case: Signup |
| ID:2 |
| Brief description:  The student or teacher inputs his or her username, password, token, real name, age, gender, class, and school. |
| Primary actors:  Students or Teachers |
| Secondary actors:  None |
| Preconditions:  The student or teacher enters the signup page. |
| Main flow:  1. The use case starts when the student or teacher enters the sign up page.  2. The system asks the student or teacher to input his or her username, password, token, real name, age, gender, class and school.  3. The student or teacher inputs his or her username, password, token, real name, age, gender, class and school.  4. The system validates the input information.  5. If signup is successful, the system displays main page for students or main page for teachers accordingly. |
| Post Conditions:  The student or teacher enters different main pages accordingly. |
| Alternative flows:  5.1. If the input information is not complete or token is wrong,  5.1.a. The system asks the student or teacher to input his or her username, password, token, real name, age, gender, class and school again.  5.1.b. The system checks whether the input information is not complete or token is wrong. |

### **3.3.3 Display Welcome Content**

|  |
| --- |
| Use case: Display Welcome Content |
| ID:3 |
| Brief description:  Students or teachers see the welcome content. |
| Primary actors:  Students or Teachers |
| Secondary actors:  None |
| Preconditions:  The system runs normally. |
| Main flow:  1. The use case starts when the student or teacher visit the website with correct URL.  2. The system displays the welcome content (videos, images and short messages). |
| Post Conditions:  Welcome content displayed. |
| Alternative flows:  None |

### **3.3.4 Display Avatar, Route Map and Status**

|  |
| --- |
| Use case: Display Avatar, Route map and Status |
| ID:4 |
| Brief description:  Students see an avatar, a route map and the avatar’s status including current progress, score, rank and so on. |
| Primary actors:  Students |
| Secondary actors:  None |
| Preconditions:  The student enters the main page for students. |
| Main flow:  1. The use case starts when the student enters the main page.  2. The system displays an avatar, a route map and the avatar’s status including current progress, score, rank and so on. |
| Post Conditions:  The student sees an avatar, a route map and the avatar’s status including current progress, score, rank and so on. |
| Alternative flows:  None |

### **3.3.5 Edit Avatar Category**

|  |
| --- |
| Use case: Edit Avatar Category |
| ID:5 |
| Brief description:  Students can change his/her avatar category as they wish. |
| Primary actors:  Students |
| Secondary actors:  None |
| Preconditions:  The student enters the main page for students. |
| Main flow:  1. The use case starts when the student click “Edit Avatar” button.  2. The system displays all avatar categories.  3. The student choose the preferred categories and click “Submit”. |
| Post Conditions:  The avatar category and image get updated. |
| Alternative flows:  None |

### **3.3.6 Display Weekly Content and Bonus Task**

|  |
| --- |
| Use case: Display Weekly Content and Bonus Task |
| ID:6 |
| Brief description:  Students see the weekly content including video watching task, game playing task, quiz task and short answer question task as well as bonus task. |
| Primary actors:  Students |
| Secondary actors:  None |
| Preconditions:  The student enters the weekly content page. |
| Main flow:  1. The use case starts when the student enters the weekly content page.  2. The system displays the weekly content including video watching task, game playing task, quiz task and short answer question task as well as bonus task. |
| Post Conditions:  The student sees the weekly content including video watching task, game playing task, quiz task and short answer question task as well as bonus task. |
| Alternative flows:  None |

### **3.3.7 Complete Task: Video Watching**

|  |
| --- |
| Use case: Complete Task: Video Watching |
| ID:7 |
| Brief description:  Students complete the video watching task. |
| Primary actors:  Students |
| Secondary actors:  None |
| Preconditions:  Students has logged in and entered the current weekly phase. |
| Main flow:  1. The use case starts when the student clicks on the button which indicates the beginning of the video watching task.  2. The system displays the video content.  3. The student watches the video.  4. The student clicks on the button which indicates the completion of the video watching task.  5. The student sees a message which indicates the result of the video watching task. |
| Post Conditions:  Students’ score, rank, avatar and other information are updated. |
| Alternative flows:  4.1. If the interval is shorter than the minimum interval demanded between two button clicking  4.1.a. the submission fails and the student is asked to watch the video again. |

### **3.3.8 Complete Task: Game Playing**

|  |
| --- |
| Use case: Complete Task: Game Playing |
| ID: 8 |
| Brief description:  Students complete the game playing task. |
| Primary actors:  Students |
| Secondary actors:  None |
| Preconditions:  Students has logged in and entered the current weekly phase. |
| Main flow:  1. The use case starts when the student clicks on the button which indicates the beginning of the game playing task.  2. The system displays the game.  3. The student plays the game.  4. The student clicks on the button which indicates the completion of the game playing task.  5. The student sees a message which indicates the result of the game playing task. |
| Post Conditions:  Students’ score, rank, avatar and other information are updated. |
| Alternative flows:  None |

### **3.3.9 Complete Task: Quiz**

|  |
| --- |
| Use case: Complete Task: Quiz |
| ID: 9 |
| Brief description:  The student subsystem should allow students to complete the quiz task. |
| Primary actors:  Students |
| Secondary actors:  None |
| Preconditions:  Students has logged in and entered the current weekly phase. |
| Main flow:  1. The use cases starts when the student clicks on the button which indicates the beginning of the quiz task.  2.The system displays the quiz.  3. The student finishes the quiz.  4. The student clicks on the button which indicates the completion of the quiz task.  5. The answers are submitted and graded automatically.  6. If passed, the student will see a message which indicates the result of the quiz task. |
| Post Conditions:  Students’ score, rank, avatar and other information are updated. |
| Alternative flows:  5.1. If the minimum points are not met, the student has to try the quiz again. |

### **3.3.10 Complete Task: Short Answer Question**

|  |
| --- |
| Use case: Complete Task: Short Answer Question |
| ID: 10 |
| Brief description:  Students finish and submit all the short answer questions which teacher will grade later and give them feedback. |
| Primary actors:  Students |
| Preconditions:  Students has logged in and entered the current weekly phase. |
| Main flow:  1. The use case starts when the student clicks on the button which indicates the beginning of the short answer question task.  2. Students click “Start” button to start the section.  3. Students filled all the questions.  4. Students click “Submit” button to submit answers.  5. Students get full score with successful submission and system automatically jumps back into the current weekly phase page. |
| Post Conditions:  All the questions are correctly finished and submitted. Students’ score, rank, avatar and other information are updated. |
| Alternative flows:  5.1. The system displays error message if any question has not been answered or any answer is too short, and user should revise answers and resubmit them. |

### **3.3.11 Complete Task: Bonus**

|  |
| --- |
| Use case: Complete Task: Bonus |
| ID: 11 |
| Brief description:  Students finish and submit bonus tasks which teacher will grade later and give them feedback. |
| Primary actors:  Students |
| Preconditions:  Student has logged in and entered the current bonus phase. |
| Main flow:  1. The use case starts when the student clicks on the button which shows all the bonus tasks.  2. Students finish one or more bonus tasks.  3. Students click “Submit” button which indicates the completion of bonus tasks.  4. Students get full score with successful submission and system automatically jumps back into the route map page. |
| Post Conditions:  Bonus tasks are correctly finished and submitted. Students’ score, rank, avatar and other information are updated. |
| Alternative flows:  None. |

### **3.3.12 View Scoreboard**

|  |
| --- |
| Use case: View Scoreboard |
| ID: 12 |
| Brief description:  Students can see the scoreboard with everyone’s score and top 10 ranks in the same class by clicking “Scoreboard” button within sidebar. |
| Primary actors:  Students |
| Preconditions:  1. Students has logged in and entered the roadmap. |
| Main flow:  1. Students click “Scoreboard” button in the sidebar. |
| Post Conditions:  Students’ score and top 10 ranks in the same class shows. |
| Alternative flows:  None |

### **3.3.13 View Student Progress**

|  |
| --- |
| Use case: View Student Progress |
| ID: 13 |
| Brief description:  Teacher can view all the students’ progress in the class. It is the default view after the teacher log in or the teacher click “Progress” button in sidebar |
| Primary actors:  Teacher |
| Preconditions:  1.Teacher has logged in. |
| Main flow:  1. Teacher can sort or filter the result by student, week, and finished status. |
| Post Conditions:  Corresponding students’ progress and score in the class shows. |
| Alternative flows:  None |

### **3.3.14 List Short Answer Question Grading Status**

|  |
| --- |
| Use case: List Short Answer Question Grading Status |
| ID: 14 |
| Brief description:  Teacher can view all the short answer questions to grade by clicking “Grading” button in sidebar, graded question are still listed but will be put in the tail. |
| Primary actors:  Teacher |
| Preconditions:  1.Teacher has logged in. |
| Main flow:  1. Teacher click “Grading” button in the sidebar.  2. Teacher can sort or filter the result by student, week, and finished/grading status. |
| Post Conditions:  Filtered/sorted short answer questions show. |
| Alternative flows:  None |

### **3.3.15 Grade/Regrade Students’ Short Answer Questions**

|  |
| --- |
| Use case: Grade/Regrade Students’ Short Answer Questions |
| ID: 15 |
| Brief description:  Teacher will grade students’ answers for short answers section, and it can be regraded when needed. |
| Primary actors:  Teacher |
| Preconditions:  1.Teacher has logged in. |
| Main flow:  1.Teacher has logged in.  2.Teacher click “Grading”  3.Teacher choose “Short Answer” subcategory  4.Teacher chooses which row to grade  5.Teacher review the answers and give the marks and feedback  6.Teacher click “Submit” to confirm grading |
| Post Conditions:  Updated short answer question grading status and updated students score. |
| Alternative flows:  5.1. If any input is malformed, there will be an error message to ask the teacher to modify the inputs and resubmit the grading. |

### **3.3.16 List Bonus Question Grading Status**

|  |
| --- |
| Use case: List Bonus Question Grading Status |
| ID: 16 |
| Brief description:  Teacher can view all the bonus questions to grade by clicking “Bonus Grading” button in sidebar, graded question are still listed but will be put in the tail. |
| Primary actors:  Teacher |
| Preconditions:  1.Teacher has logged in. |
| Main flow:  1. Students click “Grading” button in the sidebar.  2. Teacher can sort or filter the result by student, week, and finished/grading status. |
| Post Conditions:  Filtered/sorted bonus questions to grade as well as graded questions show. |
| Alternative flows:  None |

### **3.3.17 Grade/Regrade Students’ Bonus Questions**

|  |
| --- |
| Use case: Grade/Regrade Students’ Bonus Questions |
| ID: 17 |
| Brief description:  Teacher will grade students’ bonus questions, and it can be regraded when needed. |
| Primary actors:  Teacher |
| Preconditions:  1.Teacher has logged in. |
| Main flow:  1. Teacher has logged in.  2.Teacher click “Grading”  3.Teacher choose “Bonus” subcategory  3.Teacher chooses which row to grade  4.Teacher review the answers and give the marks and feedback  5.Teacher click “Submit” to confirm grading |
| Post Conditions:  Updated bonus question grading status and updated students’ score. |
| Alternative flows:  5.1. If any input is malformed, there will be an error message to ask the teacher to modify the inputs and resubmit the grading. |

# **4. Database Design**

## C:\Users\ABS\AppData\Local\Temp\WeChat Files\852919927665240242.jpg**4.1 ER Diagram**

## **4.2 Schema**

SCHOOL (SchoolID, SchoolName)

CLASS (ClassID, ClassName, SchoolID@)

TOKEN (TokenID, Type, TokenString, ClassID@)

STUDENT (StudentID, Username, Password, FName, LName, Gender, Score, ClassID@)

TEACHER (TeacherID, Username, Password, FName, LName, ClassID@)

VIDEO (VideoID, TimeThreshold, URL, Week, Points)

VIDEO\_ RECORD (VideoID@, StudentID@, Finished)

GAME (GameID, Description, Week, Points)

GAME\_ RECORD (GameID@, StudentID@, Finished)

QUIZ (QuizID, Week, Points)

QUIZ\_RECORD (QuizID@, StudentID@, Finished)

QUIZ\_QUESTION (QuestionID, Question, CorrectChoice, QuizID@)

OPTION (OptionID, Content, QuestionID@)

QUIZ\_QUESTION\_RECORD (StudentID@, QuestionID@, Choice)

SHORT\_ANSWER\_SECTION (ShortAnswerID, Week)

SHORT\_ANSWER\_SECTION\_RECORD (ShortAnswerID@, StudentID@, Finished)

SHORT\_ANSWER\_QUESTION (SAQID, Question, Points, ShortAnswerID@)

SHORT\_ANSWER\_QUESTION\_RECORD (StudentID@, SAQID@, Answer, Feedback, Grading)

BONUS (BonusID, Week)

BONUS\_RECORD (BonusID@, StudentID@, Finished)

BONUS\_TASK (BonusQuestionID, Question, Points, BonusID@)

BONUS\_TASK\_RECORD (StudentID@, BonusQuestionID@, Answer, Feedback, Grading)