Software Requirements Specification

for

SoftwareGenius

Version 1.0

Presented by: Maksers Group

Instructor: Lin Shang Wei

Course: Software System Analysis and Design

Table of Contents

1. Introduction

1.1. Purpose

This SRS document details requirements and use cases to build a user-interactive online educational game system. This system is created to gamify and socialize teaching and learning of software engineering courses, catering for the demand of The Teaching, Learning and Pedagogy Division (TLPD) of NTU. This document reports the requirements based on the official meetings of the members of Maksers from CZ3003 Lab Group SSP5, with Dr. Lin Shang-Wei, the instructor of CZ3003. This document will cover the purpose and complete declaration for the development of the system.

1.2. Document Conventions

In general this document follows the IEEE formatting requirements. Italic words are special terms defined in the data dictionary.

1.3. Intended Audience & Reading Suggestions

This document is intended for developers from team Maksers of CZ3003 Lab Group SSP5 as detailed project requirements, as well as for the authority of NTU TLPD and Dr. Lin Shang-Wei to understand the features of the SoftwareGenius system.

In the rest sections, system feature overall description are presented to provide functionalities and design of the game. Then the specific requirements part is to elaborate the details of functional and nonfunctional requirements and demonstrate use case models. Finally, this specification is concluded with the reference documents on which this document is based on.

1.4. Project Scope

The SoftwareGenius system, in the form of a social game, combines entertainment with education. The system is designed to encourage students taking software engineering courses to learn and explore with passion, as well as to facilitate the teaching process of software engineering course instructors.

The SoftwareGenius system is composed of two main components: a client-side application which will run on web browsers as game interface, and a server-side application which will store game-related data and answer incoming http requests.

1.5. References and Acknowledgments

IEEE SRS template 830-1988

<If you have other references, insert them here>

2. Overall Description

2.1. Product Perspective

Our product is a free educational web game which aims to teach students knowledge about software engineering. It is not only a simple educational web application, but also an exciting game where students can fight against enemies, exploit their territories and experience four different phases of software engineering including requirements engineering, architecture design, implementation and software testing. Only answer a question correctly can students generate damage to their enemies. Our product can be used after school for students to review what they have learned in class and explore new knowledge by themselves. We also allow teachers to join the game as administrators. They can manage the questions and supervise every student's performance in real time.

2.2. Product Features

1. Teacher (administrator) and students can login by gmail or facebook.
2. Users can choose game roles among four computer science perspectives: software engineer, testing etc.
3. Users answer questions with designed difficulty to trigger skills to beat monsters.
4. Different difficulty of skills will result in different amounts of damage points for monsters.
5. Users can gain experience by beating monsters and the accumulated experience will upgrade the level of the user.
6. Users explore new regions by beating monsters.
7. Students can upload self-designed questions which will be reviewed by administrator before included in the question database
8. Leaderboard contained the rankings of the top students with highest experience values or the highest accuracy.
9. Administrator can view the overall statistical report of his/her group of students and also the individual statistical report of each students on his/her group
10. Students can view his own overall statistical report.

2.3. User Classes and Characteristics

1. Students: All the students can use this application to learn after class. Their knowledge level on software engineering should not matter as the questions have three difficulty levels to choose.
2. Teaching Staff: Teaching staff will be the administrators of this application. They are supposed to manage the academic questions used in the game.

2.4. Operating Environment

It is a mobile based application using Unity for development. Unity gives users the ability to create games and experiences in both 2D and 3D, and the engine offers a primary scripting API in C#. Within 2D games, Unity allows importation of sprites and an advanced 2D world renderer.

2.5. Design and Implementation Constraints

1. The system shall use the current corporate standard MySQL Server database engine.
2. The system shall be developed using Unity.
3. As part of standard operating procedures, a testing plan will be documented during the design phase. The testing plan will be based on user roles, modules or use cases, required tasks and expected outcomes.
4. Database Design: The database structure should be as complete as possible during the design stage but there should be a room for modification without a large overhaul during later phases.

2.6. Documentation

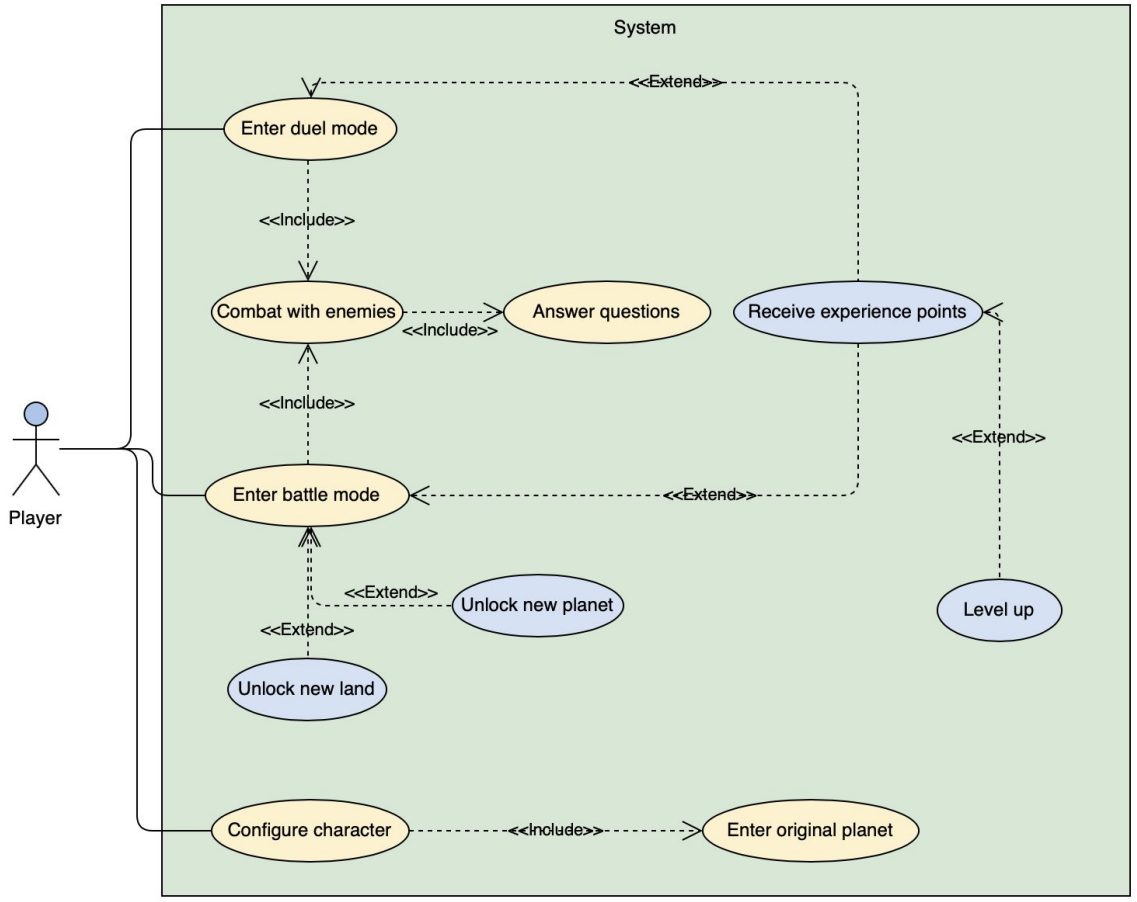
1. A tutorial will provide a quick start, a walk-thru of major system features, and further reference source for the complete system features.
2. A cross-linked help system in HTML will describe and illustrate all system functions.
3. An online form will enable users to request help. Frequently asked questions will be screened for the FAQ pages.
4. The user's guide (or user manual) will contain sufficient information and instructions required to access and use the data system. It will include:
    1) Overview of the system features and architecture.
    2) Instructions for accessing the system.
    3) Samples of screens, where appropriate.

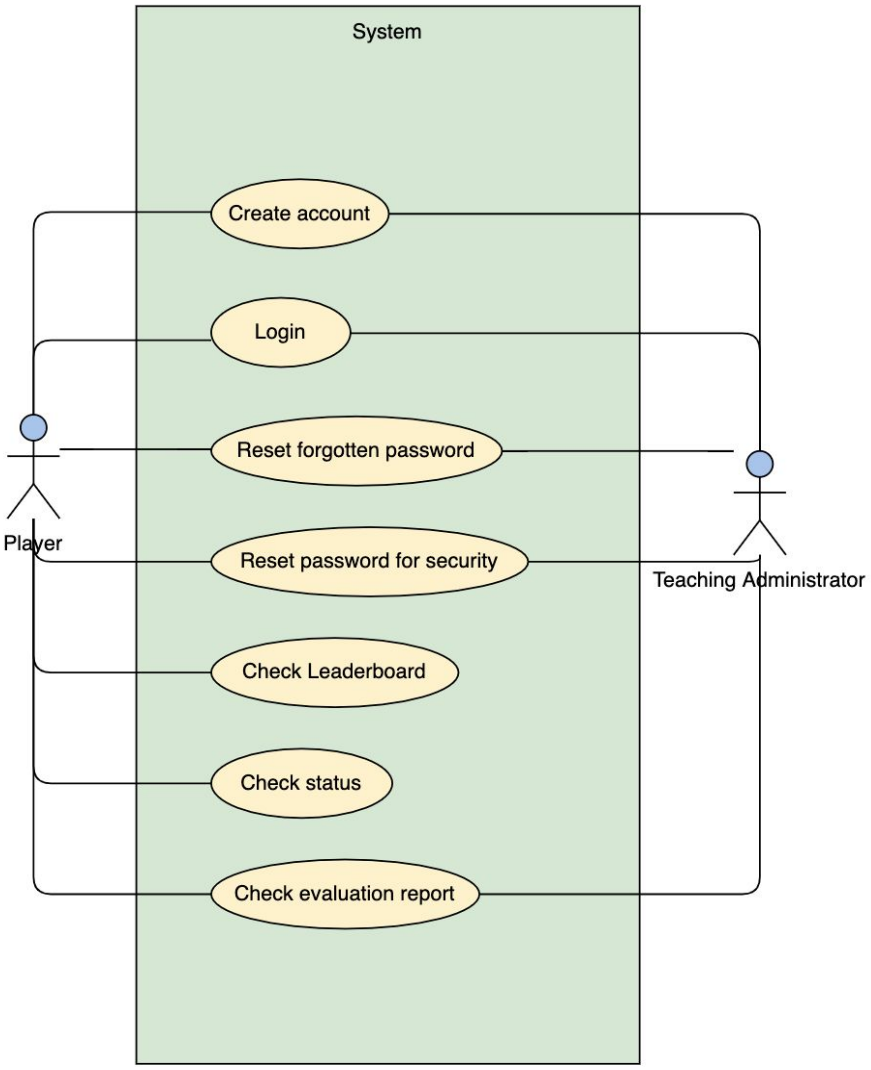
2.7. Assumptions and Dependencies

1. Since this application is a web based application game, an internet browser is required.
2. It will be assumed that the users have a good internet connection.
3. It is assumed that the user is familiar with an internet browser and has basic knowledge of handling the keyboard and mouse
4. The users know the English language, as the user interface will be provided in English.
5. Access to the application is restricted only to registered students and staff of NTU who have a pre-assigned matriculation number.

6. It is assumed that NTU database shall provide support to check the validation of User's username and matriculation number.

3. System Features





3.1. Battle Mode

- Description

  The battle mode is for players to combat against a Non-Player-Character (NPC) through answering skill-triggering questions which are composed of four different genres of software engineering topics. The goal is to exhaust the Hitpoints of the NPC by triggering multiple skills. The players will receive experience points if they win a battle.

- Response Sequence

Use Case ID:	
Use Case Name:	Battle mode use case

Actor:	Players
Description:	Players can enter the battle mode
Preconditions:	Players have logged in
Postconditions:	
Priority:	
Frequency of Use:	
Flow of Events:	Use case 1: enter battle mode 1. Player enters battle mode by clicking at one of the land units assigned to him/her. 2. The system displays 3 levels of difficulty including easy, medium, and hard on the screen for players to choose from. 3. Player selects one of the difficulty levels displayed to play. Use case 2: combat with the enemy 1. The system starts the timer for 30 seconds. 2. The system displays an enemy, a skill-triggering question, four answer options, and the remaining time on the screen. 3. Player chooses one of the four answer options. 4. The system verifies the correctness of the chosen answer. 5. The system displays “Correct answer” for 2 seconds. [Alternate Flows: Wrong answer] [Alternate Flows: Times up] 6. Player triggers his/her skill to attack the enemy. 7. The system reduces the Hit point of the enemy. 8. The enemy attacks the player causing the player to lose his/her Hit Point. Step 3~8 is repeated until the Hit Point of either the player or the enemy reduces to zero. Use case 3: Obtain the battle results 1. The system displays “Congratulations! You won the battle!” for 3 seconds. [Alternate Flows: Unsuccessful conquer] 2. Player receives the victory rewards with experience points.

	3. The system updates the total experience points of the player. 4. The system flags the target land with the player's ID. 5. The system unlocks a new land for the player. [Extension point: unlock new planet]
Alternative Flows:	Use case 2: AF-S5: If the answer is incorrect 1. The system displays "Wrong answer" for 2 seconds. 2. Jump to step 8. AF-S5: If the time runs out 3. The system displays "Times up" for 2 seconds. 4. Jump to step 8. Use case 3: AF-S1: If the Hit Point of the player reduces to zero 1. The system displays "Practice makes perfect, keep trying!" for 3 seconds. 2. The system displays two buttons with options: 2.1. "Go to study mode" 2.2. "Keep challenging " 3. The user clicks on "Go to study mode" to enter study mode. 4. The user clicks on "K 5. eep challenging" to jump to use case 2.
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

- Functional Requirement

1. The system will record experience points each player has earned in the battle mode.
   1.1. Players earn experience points through defeating enemies.
      1.1.1. The experience points earned by defeating an enemy is proportional to the overall difficulty level of the battle.

1.1.2. The experience points earned by defeating an enemy is proportional to the difficulty level of the question.
1.1.3. The experience points earned by defeating an enemy is reversely-proportional to the time used to answer the skill-triggering question.
1.2. The system will not allow players to earn any experience points if they explore the same land multiple times with the same difficulty level.
1.3. The system will record each player's total experience points across all battles.

2. The system only allows players to explore new lands if they successfully defeat all enemies on the land.
2.1. Lands adjacent to the currently explored lands will be opened.
2.2. Players defeat enemies by triggering a skill.
2.2.1. Each character has four skills to choose from.
2.2.2. Each skill has a certain property.
2.2.3. Each skill has a certain damage point.
2.2.4. A successfully triggered skill will cause the Hitpoint of enemies to drop by the same amount as the damage point.
2.3. The system requires players to answer a software engineering related question before they can trigger the skill.
2.3.1. The system will retrieve questions from a question database for players to answer.
2.3.2. The system should not display the same question for any 3 consecutive rounds.
2.3.3. Difficulty of questions depend on the battle mode's difficulty level.
2.3.4. The system requires players to choose an answer in 30 seconds.
2.3.5. The system provides at least 3 options and at most 6 options for players to choose from.
2.3.6. The system shall display the amount of time left.
2.3.7. The system will hide all options when reaching the time limit.
2.3.8. The system will collect the questions that the players get wrong for them to review.
2.3.9. Players can not trigger any skills if their answers are wrong or they don't choose the answer within the time limit.
2.4. The system triggers a skill for enemies to attack players after players answer a question.
2.5. If the Hitpoints of the player run out, the land can not be conquered.

3. The system allows players to choose from at most three difficulty levels before they enter the battle mode.
3.1. Three difficulty levels should include "easy", "medium" and "hard".
3.2. The system will recommend a difficulty level based on players' past battle performance.
3.3. The system won't allow players to choose a level easier than the recommended one.
3.4. Players can explore a land multiple times until they clear the highest difficulty level.

3.2. Duel Mode

- Description

  The duel mode is for players to compete with each other. A player can choose to enter other players' planets and take over their lands. The experience points can only be earned by the attacker if the single-battle experience points exceed that of the defender. A player can only conquer 2 lands of each friend within 24 hours.

- Response Sequences

Use Case ID:	
Use Case Name:	Duel mode design

Actor:	Players
Description:	Players can enter the duel mode
Preconditions:	Players have logged in and Players have friends connections on the system
Postconditions:	
Priority:	
Frequency of Use:	
Flow of Events:	Use case 1: enter duel mode 1. Player clicks on the friends' circle to view the list of friends. 2. The system displays a sidebar listing all connections of the player. 3. Player clicks one of the friend's name tabs to enter his/her planet. 4. The system changed the view to the target player's planet. 5. Player clicks on one of the land units that has been explored by the target player. Use case 2: combat with the enemy 1. The system displays 3 levels of difficulty including easy, medium, and hard on the screen for players to choose from. 2. Player selects one of the difficulty levels displayed to play.

	3. The system starts the timer for 30 seconds. 4. The system displays an enemy, a skill-triggering question, four answer options, and the remaining time on the screen. 5. Player chooses one of the four answer options. 6. The system verifies the correctness of the chosen answer. 7. The system displays “Correct answer” for 2 seconds. [Alternate Flows: Wrong answer] [Alternate Flows: Times up] 8. Player triggers his/her skill to attack the enemy. 9. The system reduces the Hit point of the enemy. 10. The enemy attacks the player causing the player to lose his/her Hit Point. Step 4~9 is repeated until the Hit Point of either the player or the enemy reduces to zero. Use case 3: Obtain the battle results 1. The system calculates and displays the experience points earned by the attacker for 3 seconds 2. The system displays the experience points earned by the defender for 3 seconds 3. The system compares the two experience points and displays “Congratulations! You have successfully conquered the land!” [Alternate Flows: Unsuccessful conquer] 4. Player receives the victory rewards with experience points. 5. The system updates the total sexperience points of the player. 6. The system flags the target land with the attacker’s ID. Use case 2 and 3 are repeated if the player wants to conquer more lands. One attacker is under the restriction of only conquering 2 lands from each player within 24 hours. Use case 4: Return to home planet 1. Player can click the button “Back to my planet” to go back to his/her home planet
Alternative Flows:	Use case 2: AF-S7: If the answer is incorrect 1. The system displays “Wrong answer” for 2 seconds. 2. Jump to step 10. AF-S7: If the time runs out 1. The system displays “Times up” for 2 seconds. 2. Jump to step 10. Use case 3:

	AF-S3: If the attacker fail to win the battle 1. The system displays “Practice makes perfect, keep trying!” for 3 seconds. 2. The system displays three buttons with options: 2.1. “Go to study mode” 2.2. “Keep challenging ” 2.3. “Back to my planet” 3. The player clicks on “Go to study mode” to enter study mode. 4. The player clicks on “Keep challenging” to jump to step 2. 5. The player clicks on “Back to my planet” to go back to his/her planet.
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

- Functional Requirements

1. The system shall allow players to **enter duel mode**.
   1.1. The system only allows players to conquer the land with the same level of difficulties as what the defender has already achieved.
   1.2. The system only allows players to take over lands that are already conquered by the other player.
   1.3. Attackers must achieve a higher score than the other player to take over the land.
   1.4. Attackers can only challenge a land up to 2 times within 24 hours.
   1.5. Lands taken over by attackers can be taken back by the defender only if he/she can achieve a higher score than attackers.

2. The system will grant rewards to the players if they successfully conquered other player’s lands.
   2.1. The rewards will be a skill-booster which can enhance the damage points of players in the battle with enemies.
   2.2. Players can choose which difficulty level of land they want to attack. The amount of rewards received for conquering a land depends on the level of difficulty of the land.
   2.3. The rewards are proportional to the level of difficulty.
      2.3.1. Easy-land will boost 10 damage points to the dominating skills and 5 damage points to other skills
      2.3.2. Mid-land will boost 15 damage points to the dominating skills and 20 damage points to other skills

2.3.3. Hard-land will boost 20 damage points to the dominating skills and 15 damage points to other skills

3.3. World Map

- Description

  The world map is the homepage view of the game. Every player has their own unique home planets. The players can extend his/her lands and unlock new maps by winning battles. A land unit with the flag of other users' ID indicates that the land has been taken over by someone else.

- Response Sequences

Use Case ID:	
Use Case Name:	Game World (map) design

Actor:	Players
Description:	Players can access the game world
Preconditions:	Players have logged in
Postconditions:	
Priority:	
Frequency of Use:	
Flow of Events:	Use case 1: configure the character 1. The system displays the four images of the characters on screen. 2. Player clicks on the image to choose one of the four characters. Use case 2: Obtain the initial world settings. 1. The system customizes the initial world as the corresponding planet of the chosen character 2. The system displays the map view with 3 base land units visible.

	Use case 3: Explore new lands 1. The player can enter battle mode to extend the lands. 2. The lands view is extended [Alternate Flows: fail to explore the land] 3. The land unit is lit up 4. A “next planet” button shows up when all land units on the current map are lit up Use case 4: Unlock new world 1. The player can switch to a new planet by clicking the “next planet” button. Use cases 2 ~ 4 are repeated until all the 4 planets are explored by the player
Alternative Flows:	Use case 3: AF-S2: Fail to winning the battle 1. Jump to step 1.
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

- Functional Requirement

1. The system allows players to configure characters:
   1.1. Players must configure his/her own character.
      1.1.1. Each player must choose an initial role between product managers (PM), software architects, developers, and quality assurance engineers (QA) after creating an account.
      1.1.2. Each character has its strength in his/her dominated field, which means the character will have higher damage points (10 pts) during battles in the corresponding world while the regular damage points are 5.

   1.2. The system will assign a world to the players based on their chosen characters.
      1.2.1. The system provides 4 different types of planets including requirement engineering, architectural design, implementation, and software testing.
      1.2.2. The system will assign an initial amount of land units to each player.
      1.2.3. The system will assign new planets to players when players conquer all lands in the current planets.
   1.3. Character & theme match list:

1.3.1. The dominating planet for PMs is the requirement engineering theme.
1.3.2. The dominating planet for Software architects is the architectural design theme.
1.3.3. The dominating planet for developers is the implementation theme.
1.3.4. The dominating planet for QAs is the requirement engineering theme.

2. World design
   2.1. The planet is designed as a map composed of multiple hexagon units which represent the lands of the planet.
      2.1.1. Each planet will have its own representative color.
      2.1.2. The level of difficulties will be reflected in the darkness of the land colors.
   2.2. The player will start from the left bottom corner of the map with the vision of its adjacent lands. With more lands explored, the vision will be expanded.
   2.3. To take over someone else's land, the player can click the friend list to enter others' planet.

3.4. Teaching Administrator

- Data Dictionary

Term	Definition
Question id	Each question has a unique id.
Question details	includes its id, description, correct answer, category, difficulty level, accuracy, time of creation and rating
Time of creation	the time when the question was added to the system
Rating	the average rating from users on a certain question.
Pending self-designed question	a submitted self-designed question waiting for approval from the system administrator
Overall performance report	indicates the overall performance of all users
Personal performance report	indicates the personal performance of a certain user
User's report (may have a better name)	user's questions and comments on the description, answer or difficulty level of a question

- Description

  The Administrator mode is for university staff to login as administrator, allowing them to access and modify the questions database and the students' performances reports. The administrator can view, modify, add, or delete the questions categorized by the three different difficulty levels: easy, medium, and hard.

- Response Sequences

Use Case ID:	
Use Case Name:	Administrator mode design

Actor:	Administrator
Description:	Administrator can access the backend of the system
Preconditions:	Administrator have logged in
Postconditions:	
Priority:	
Frequency of Use:	
Flow of Events:	Use case 1: administrator login 1. The administrator logs in as administrator Use case 2: view questions 1. The administrator requests to view questions. 2. The system displays all the questions currently in use categorized by their difficulty levels. 3. The administrator requests to search for questions. 4. The system requests question id. 5. The administrator inputs question id. 6. The system displays the results. 7. The administrator requests to filter the questions by their difficulty level. 8. The system displays the result. 9. The administrator requests to view the details of a question. 10. The system displays its details. Use case 3: modify question details 1. The administrator requests to modify the question.

	2. The system turns to the interface of modifying question description. 3. The administrator modifies the question. 4. The system saves the modification. 5. The system displays “Modification Succeed”. Use case 4: add question 1. The administrator requests to add a question. 2. The system turns to the interface of creating a question. 3. The administrator inputs all the information. 4. The administrator clicks “add”. 5. The system adds the question to the database. 6. The system displays “Adding Succeed”. Use case 5: delete question 1. The administrator requests to delete a question. 2. The system requests for question id. 3. The administrator inputs the id. if the question id exists 4. The system displays “Are you sure to delete it?” and “Yes”,”No” buttons. If the administrator chooses “Yes” 5. The system removes the question from the database. 6. The system displays “Delete Succeed”. Use case 6: obtain performance reports 1. The administrator requests for a performance report. 2. The system displays the overall performance report. 3. The administrator inputs a user id to obtain his personal performance report. if the user id exists: 4. The system displays his personal performance report.
Alternative Flows:	Use case 5: AF-S4: if the question id doesn’t exist 1. The system displays “Question id does not exist” and turns back to the previous state. AF-S5: if the administrator chooses “No” 2. The system turns back to the previous state. Use case 6: AF-S4: if the user id doesn’t exist 4. The system displays “User id does not exist” and turns back to the previous state.

Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

- Functional Requirement

1. A teaching staff can login to the game as an administrator.

2. The administrator should be able to view questions currently used in the game.
   2.1. The administrator should be able to search for questions by id.
   2.2. The administrator can view the questions categorized by different difficulty levels.
   2.3. The administrator can view the question details.

3. The administrator should be able to manage questions.
   3.1. The administrator should be able to modify questions including contents, correct answer, difficulty level, etc.
   3.2. The administrator should be able to add new system questions.
   3.3. The administrator should be able to delete certain questions.

4. The administrator should be able to view/supervise users' performance.
   4.1. The administrator can obtain an overall performance report which shows the overall mastery of the course in real time.
   4.2. The administrator can access the leaderboard.
      4.2.1. The administrator can access the complete leaderboard.
   4.3. The administrator can obtain the personal performance report of any users by their id or name.

3.5. Check status

- Description and Priority

  Every user has a "bag" which records the current gaming parameters and territory information. The user can click and check for status updates.

- Use Case Description

Use Case ID:	

Use Case Name:	Check status use case

Actor:	Players,
Description:	Players check status
Preconditions:	Players clicks bag button
Postconditions:	Players clicks close button
Priority:	
Frequency of Use:	
Flow of Events:	1. Users click the “bag” button on the homepage. 2. The system queries the gaming parameters from the database using the player’s id. 3. Users can check the gaming parameters displayed on the bag pop-up window. 4. Users can click on the close button on the top right corner to return to the homepage.
Alternative Flows:	
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

- Functional Requirements

1. The system shall track the gaming progress of each user
   1.1. The system shall track the character that the user is using
   1.2. The system shall track the current level of the character which ranges from 1 to 30
   1.3. The system shall track the multiple skills of the character
   1.4. The system shall track the the damage point of a specific skill which ranges from 0 to 100
   1.5. The system shall track the experience point of user which ranges from 0 to infinity
   1.6. The system shall track the territory information including the number of territories the user has explored and the number of grids in a territory.
2. There should be a “bag” button on the homepage.
3. All players should be able to check their status by pressing the button anytime.

4. The status is displayed in a pop-up window with a close button on the top right corner.

3.6. Report

- Description

  A report includes different sets of data about a player, which is stored in the system backend. Not all datasets are visible to users. This data shows what knowledge the player has learned and describes how well the player understands the knowledge

- Use cases

Use Case ID:	
Use Case Name:	Check evaluation report

Actor:	Players, Teaching administrators
Description:	Evaluation report evaluation report status
Preconditions:	Players/Teaching administrators click the “Generate report” button
Postconditions:	Players/Teaching administrators clicks return button Or Players/Teaching administrators clicks outside the pop-up window
Priority:	
Frequency of Use:	
Flow of Events:	1. Players/Teaching administrators clicks on the ‘Generate report’ button on homepage 2. The Report-generating-system checks if the user is a student or teacher 3. If it is a student, the Report-generating-system queries the experience point of each field and total accuracy using the player’s id from the database. 4. Advice is generated by the system based on the student’s information. The statistics as well as the advice will be displayed on the student's report in a pop-up window. 5. If it is a teacher, the Report-generating-system queries experience points of each field, total accuracy and length of online period from the database.

	6. The average experience point world by world will be displayed on admin's report in a pop-up window. If the admin clicks on "details", the whole list of student information will be shown, sorted by student-id by default.
Alternative Flows:	
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

- Functional Requirements

1. The system keeps the record of each player's information in a database.
2. The database concludes of student-id, current character name, current world name, experience point, total accuracy and length of online period.
3. The database keeps the statistics of all worlds.
4. A button named 'Generate report' should be placed on the homepage.
5. All players should be able to check their reports by pressing the button anytime.
6. The report is displayed in a scrolling-abled pop-up window with a close button on the top right corner.
7. The report can be closed by the close button (or by clicking outside the window).
8. In Student mode, experience point and total accuracy are displayed in the report. Additionally, the system will generate a piece of advice for students to guide them further.
9. In Administrator mode, the average of all students' experience point, total accuracy and length of online period is shown world by world.
10. In Admin mode, a "detail" button is on the bottom of the report window.
11. In Admin mode, administrators can access each student's information by clicking on the "detail" button.

3.7. Leaderboard

- Description

A leaderboard is a scoreboard showing the names and current scores of the leading gamers, sorting by score in descending order.

- Use cases

Use Case ID:	
Use Case Name:	Check leaderboard

Actor:	The logged-in user
Description:	The logged-in user can check the current leaderboard
Preconditions:	1. The Student/Teacher has internet connection And 2. The Student/Teacher is on homepage
Postconditions:	1. The Student/Teacher clicks the close button Or 2. The Student/Teacher clicks outside the pop-up window
Priority:	
Frequency of Use:	
Flow of Events:	1. The Student/Teacher clicks on the 'Check Leaderboard' button on homepage 2. The Leaderboard Displaying System checks if the currently logged-in account is a student account or an admin account. 3. The Leaderboard Displaying System queries the top 10 users' names and scores from the database. 4. If it is a student account, the Leaderboard Displaying System checks if he/she is in top 10 gamers. If not, the Leaderboard-displaying-system queries the student's score and rank from the database 5. If it is a teacher, the Leaderboard-displaying-system queries the top 10 users' names and scores from the database 6. The Leaderboard-displaying-system displays the queried ranks, names and scores in a pop-up window
Alternative Flows:	
Exceptions:	EX-S3: If there are less than 10 records in the database 1. database return all records found
Includes:	

Special Requirements:	
Assumptions:	
Notes and Issues:	

- Functional requirements

1. Each world should have their own leaderboard
2. The leaderboard contains three columns, named ‘Rank’, ‘Name’, and ‘Score’ respectively.
3. Top 10 leading gamers and their scores are shown on the scoreboard
4. If the total number of records in the database is less than 10, then all records are shown.
5. If the currently logged-in account is a student account
   5.1. If the account’s own rank is not top 10, the account’s rank, name and score shall be displayed at the bottom of the leaderboard as the last record.
   5.2. The account’s own record shall be highlighted on the board.
6. A button named ‘Check Leaderboard’ should be placed on the homepage
   6.1. The gamers should be able to check the scoreboard by pressing the button anytime
7. The leaderboard is displayed in a scrolling-abled pop-up window with a close button on the top right corner.
8. The leaderboard can be closed by the close button (or by clicking outside the window).

3.8. Account Management

- Description
  The Account Management system allows current users to enter the game and new users to register an account

- Use cases

Use Case ID:	
Use Case Name:	Create account

Actor:	User

Description:	A new user to the game may create a new account to enter the game or to perform admin duties.
Preconditions:	1. User is an existing student or staff in NTU AND 2. User has a valid NTU email account and matric/staff card AND 3. The user has a network connection
Postconditions:	1. User's gaming profile is created and saved in the database of the game OR 2. The game has failed to create an account for the user
Priority:	
Frequency of Use:	
Flow of Events:	1. The user selects the type of account to create, i.e., student account or admin account 2. The user keys in his matriculation number 3. The user keys in a username, which is unique and does not include unsupported characters 4. The user keys in his NTU email account 5. The user keys in his real name 6. The user sets his password, which satisfies the requirements 7. The user reenters and successfully confirms his password 8. The user chooses his security question 9. The user answers his security question 10. The system verifies the email account, name, matriculation number with NTU database 11. The user clicks "Create Account". Subsequently, all inputs are successfully saved by the game system and the account is created
Alternative Flows:	AF-S1 if the user selected "Admin Account" 1. The user keys in his staff number 2. The flow continues from step 3
Exceptions:	EX-S3 If the chosen username is already taken 1. The input username is cleared 2. The system displays the message "Username already taken. Please select another one" 3. The system continues from step 3 EX-S6 if the entered password does not satisfy the requirements 1. The input username is cleared 2. The system displays the message "Password must be alphanumeric combined with special characters" 3. User is prompted to reselect a password

	EX-S7 if the reentered password does not match the previously entered password 1. The input for password and password confirmation sections are cleared 2. The system displays the message “Passwords do not match. Please re-enter” 3. The user is taken back to step 6 EX-S10 if the input cannot be verified from the NTU database 1. The system displays the message “One or more fields from your personal details might not be valid” 2. The user continues from step 2 EX-S11 if the input are not successfully saved due to reasons such as bad network connection 1. The system displays the message “Sorry, account not created due to bad network connection” 2. The user continues from step 11
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	
Use Case Name:	Login to the game

Actor:	User
Description:	An existing user may use his/her credentials to login to the game
Preconditions:	1. User’s profile is existent in the game’s database AND 2. User remembers his login credentials AND 3. The user has a network connection
Postconditions:	1. User successfully logins to the game OR 2. User fails to login with 3 attempts and his account is frozen for 1 day
Priority:	
Frequency of Use:	

Flow of Events:	1. The user keys in the username 2. The user keys in the password
Alternative Flows:	
Exceptions:	EX-S1 The username does not exist in the game system 1. The system displays the message “Username does not exist” EX-S2 The password is incorrect 1. The inputs are cleared 2. The system displays the message “Wrong password” 3. The account is frozen if the password is keyed in wrongly for 3 times in a row
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	
Use Case Name:	Reset password for security

Actor:	User
Description:	A user may reset his/her password when he/she remembers his/her password but wishes to make an update
Preconditions:	1. User’s profile is existent in the game’s database AND 2. User remembers the old password AND 3. The user has a network connection
Postconditions:	1. User successfully resets his password OR 2. User fails to reset his password
Priority:	
Frequency of Use:	
Flow of Events:	1. The user click “Forgot password” 2. The user chooses the security question 3. The user answers the security question

	4. The system verifies the answer to the security question 5. The user enters new password 6. The system verifies that the password satisfies the requirements 7. The user is prompted to re-enter the password 8. The system verifies whether the two passwords entered are the same 9. The system saves the new password
Alternative Flows:	
Exceptions:	EX-S4 The user keys in the wrong answer 1. The input is cleared 2. The system displays the message “Wrong answer. Please try again” 3. The user is taken back to step 2 EX-S6 if the entered password does not satisfy the requirements 1. The input is cleared 2. The system displays the message “Password must be alphanumeric combined with special characters” 3. User is taken back to step 5 EX-AS8 if the re-entered password does not match the previously entered password 1. The input for password and password confirmation sections are cleared 2. The system displays the message “Passwords do not match. Please re-enter” 3. The user is taken back to step 5 EX-AS9 if the input are not successfully saved due to reasons such as bad network connection 1. All the previous input are cleared 2. The system displays the message “Sorry, account not created due to bad network connection” 3. The user is taken back to step 1
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

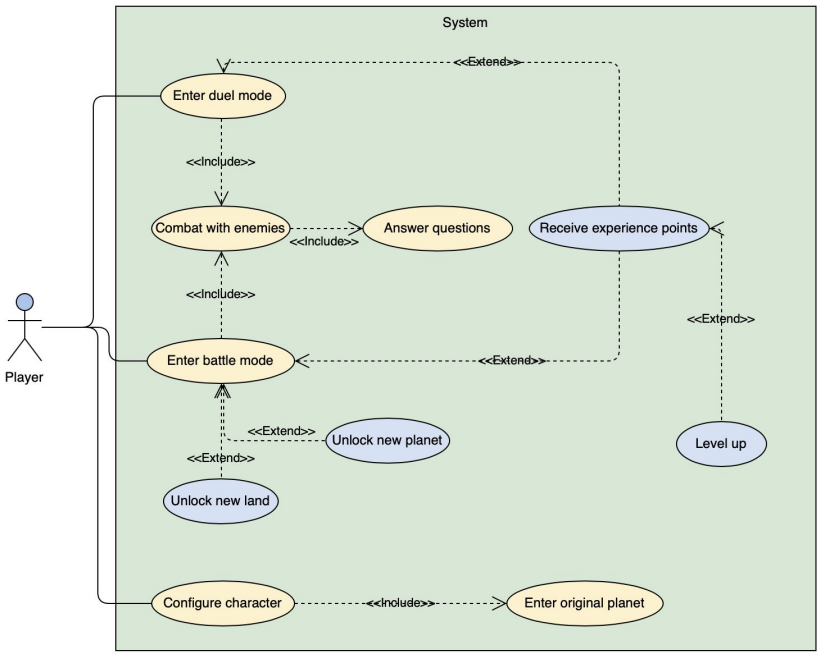
Use Case ID:	
Use Case Name:	Reset forgotten password

Actor:	User
Description:	An existing user forgot his password and chooses to reset the password
Preconditions:	1. User's profile is existent in the game's database AND 2. User remembers the answer to his security question
Postconditions:	1. User successfully resets his password OR 2. User fails to reset his password
Priority:	
Frequency of Use:	
Flow of Events:	1. The user enters the old password 2. The system verified the password 10. The user enters the new password 11. The system verifies that the password satisfies the requirements 12. The user is prompted to reenter the password 13. The system verifies whether the two passwords entered are the same 14. The system saves the new password
Alternative Flows:	
Exceptions:	EX-S2 The password is wrong 1. The input is cleared 2. The system displays the message "Wrong password. Please try again" EX-S4 if the entered password does not satisfy the requirements 1. The input is cleared 2. The system displays the message "Password must be alphanumeric combined with special characters" 3. User is prompted to reselect a password EX-S6 if the reentered password does not match the previously entered password 1. The input for password and password confirmation sections are cleared 2. The system displays the message "Passwords do not match. Please re-enter" 3. The user is taken back to step 4 EX-S7 if the input are not successfully saved due to reasons such as bad network connection 1. All the previous input are cleared 2. The system displays the message "Sorry, account not created due to bad network connection" 3. The user is taken back to step 1

Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

- Functional requirements

1. Users should be able to create an account and choose the type of account if they are new to the game
    1.1. All users must create a username
        1.1.1. The username must be unique and alphanumeric only. Underscore is allowed
        1.1.2. The username is case-insensitive
    1.2. Users may choose to create a student account or an admin account
        1.2.1. If the user creates a student account
            1.2.1.1. he/she is required to enter his/her full name as appeared on the matric card
            1.2.1.2. he/she is required to enter his/her matriculation number as appeared on the matric card
        1.2.2. If the user creates an admin account
            1.2.2.1. he/she is required to enter his/her full name as appeared on the staff card
            1.2.2.2. he/she is required to enter his/her staff number as appear on the staff card
    1.3. Users are required to enter his/her NTU email address
        1.3.1. The system shall check whether the email address is valid
    1.4. Users are required to create his/her password
        1.4.1. Passwords are case sensitive and must include special characters
        1.4.2. Users are required to confirm his/her password by entering the password again
    1.5. Users are required to select a security question and provide the answer
    1.6. Users are required to connect his/her account with his/her social media account, such as Facebook, Twitter.
2. Users should be able to login at the entry page of the game
    2.1. The user needs to enter his/her username or email account
        2.1.1. The entry is case-insensitive
        2.1.2. If the username/email entered does not exist, the message”invalid user” would be displayed
    2.2. The user needs to enter his/her password
        2.2.1. Password will be masked by the system
        2.2.2. Users have only 3 chances to enter the passwords
        2.2.3. If all failed, account would be frozen for 1 day
    2.3. The user may answer his/her security question to reset the password
        2.3.1. The user is required to select and answer his/her security question
        2.3.2. The user is required to enter a new password and confirm the password by re-entering it
        2.3.3. Passwords are case sensitive and must include special characters



4. External Interface Requirements

4.1. User Interfaces

The user interface will be simple and consistent, using terminology commonly understood by the intended users of the system. The system will have a simple interface, consistent with industry standard interfaces, to eliminate the need for user training of infrequent users. The Maksers team will evaluate the user interface of similar systems and apply appropriately. User testing will be used to ensure the user interface is clear (simple,commonly understood vocabulary, intuitive to use without training), complete (users can perform all functions from the interface), and consistent(buttons and wording are the same throughout the system).

4.2. Hardware Interfaces

No extra hardware interfaces are needed. The system will use the standard hardware and data communications resources. This includes, but is not limited to, the general

Ethernet network/T1 connection at the server/hosting site, network servers, and network management tools.

4.3. Software Interfaces

The system will use the standard software resources available on the Internet. This includes, but is not limited to Unity game engine, Java, PHP, Firebase and also if there is another necessary resource needed by the mobile application.

4.4. Communication Interfaces

The system will use common communication resources. This includes, but is not limited to, HTTP protocol for communication with the web browser and the web server. TCP/IP network protocol with HTTP protocol. This is done for compatibility and stability purposes. APIs from Facebook and Google shall also be involved.

5. Other Non-functional Requirements

5.1. Safety Requirements

SR-1: The SoftwareGenius system shall be able to preserve user data when unexpected damage, such as sudden power shortage, occurs to the main server.

5.2. Security Requirements

SCR-1: The SoftwareGenius system will follow industry best practices for authentication. Authentication addresses security requirements to ensure those using system are who they say they are. This is of greatest concern when the system contributes to the university course grading. This is primarily done through login IDs and passwords.

Appendix A: Data Dictionary

Term	Definition
Battle	A battle is a competition between players and Non-player-characters (NPC). In each battle, players need to trigger skills and attack the opponents. Each battle has three different levels players can choose from.
Non-Player-Character	NPC is a system controlled character that will compete with players in each battle.
Skill-triggering question	A skill-triggering question is a software-engineering related question that players must answer correctly to trigger a skill to attack enemies in a battle.
Hit point	The amount of damage a character can withstand before it is defeated.
Skill	Skills are special abilities to attack enemies with a certain damage point. Each character has 4 skills
Experience point	Experience points are earned after players win a battle. It can be accumulated to level up a character.
Damage point	Damage point is a value related to a skill that is used to attack other users. Damage point refers to the amount of hit points a skill can reduce on the opponents.
Accuracy	Accuracy is a value related to a certain question. For a certain question, it is calculated as the times it is correctly answered divided by the times it is answered.
Land	There are 20 pieces of explorable lands on each planet. Players explore and conquer new lands by entering battle mode and beat all enemies on the land.
Level of character	There are 30 levels of a character. Players enter battle or duel mode to earn experience points and level up their characters.
Planet	A planet refers to a world. There are 4 planets/worlds in the game, namly, Software Engineering World, Software Architecture World, Project Management World, and Quality Assurance World.

Admin account	Accounts for teachers at the university so that they can track student's learning status, check reports and update question tank
Student account	Accounts for students at the university so that they can play the game and allow the system to track their progress
Security question	A question the user selects which has the answer that is private and only known to the user himself/herself. The question should be a short answer question and the answer should have no more than 10 words
Leaderboard	A leaderboard is a scoreboard showing the names and current scores of the leading gamers, sorted by score in descending order.
Friend	Players can connect with other players and add them as friends.