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
1.2. Document Conventions
1.3. Intended Audience & Reading Suggestions
1.4. Project Scope
1.5. References and Acknowledgments

2. Overall Description

2.1. Product Perspective
2.2. Product Features
2.3. User Classes and Characteristics
2.4. Operating Environment
2.5. Design and Implementation Constraints
2.6. Assumptions and Dependencies

3. System Features

3.1. Battle Mode
3.2. Duel Mode
3.3. World Map
3.4. Teaching Administrator
3.5. Check profile
3.6. Report
3.7. Leaderboard
3.8. Account Management

4. External Interface Requirements

4.1. User Interfaces
4.2. Hardware Interfaces
4.3. Software Interfaces
4.4. Communications Interfaces

5. Other Non-functional Requirements

5.1. Safety Requirements
5.2. Security Requirements0000

Appendix A: Data Dictionary

Appendix B: Priority Table

Appendix C: Analysis Models

Appendix D: System Design

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

This document follows the IEEE SRS 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 mobile game which aims to teach students knowledge about software engineering. It is not only a simple educational mobile 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. Users need to sign up before they enter the main page.
2. Teachers (administrators) or students can login if they've already had an account.
3. Users can choose game roles among four computer science perspectives: software engineer, soft architecture, quality assurance and project manager.
4. Users answer questions with designed difficulty to trigger skills to beat monsters.
5. As the level of character increases, the damage points for monsters will increase accordingly.
6. Users can gain experience points by beating monsters and the accumulated experience will upgrade the level of the user.
7. Users explore new regions by beating monsters.
8. Leaderboard contained the rankings of the top students with highest experience values.
9. In Admin mode, the teaching admin can choose to sort the report list based on ascending order of student id, ascending order of accuracy, descending order of accuracy or ascending order of student name.
10. Administrators can view the overall statistical report of students and also the individual statistical report of each student.
11. 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 an Android 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.

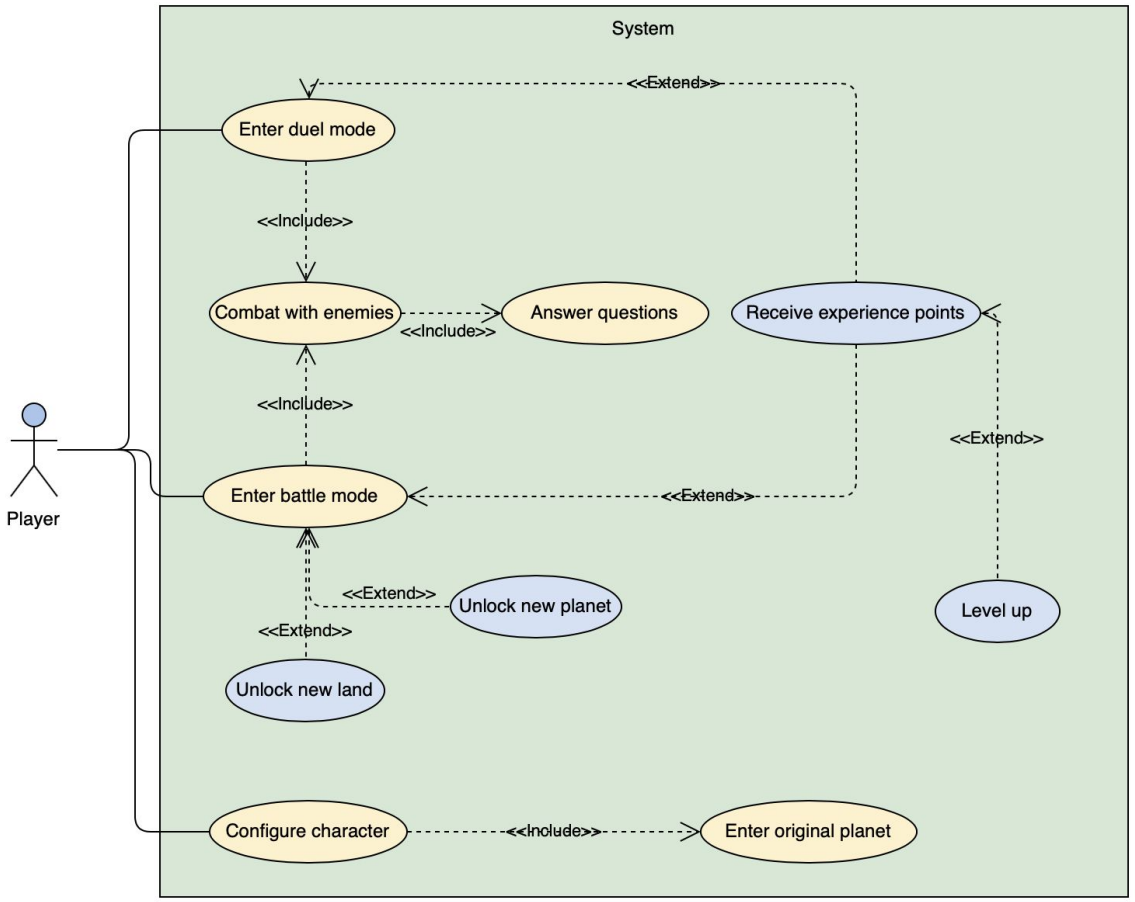
2.5. Design and Implementation Constraints

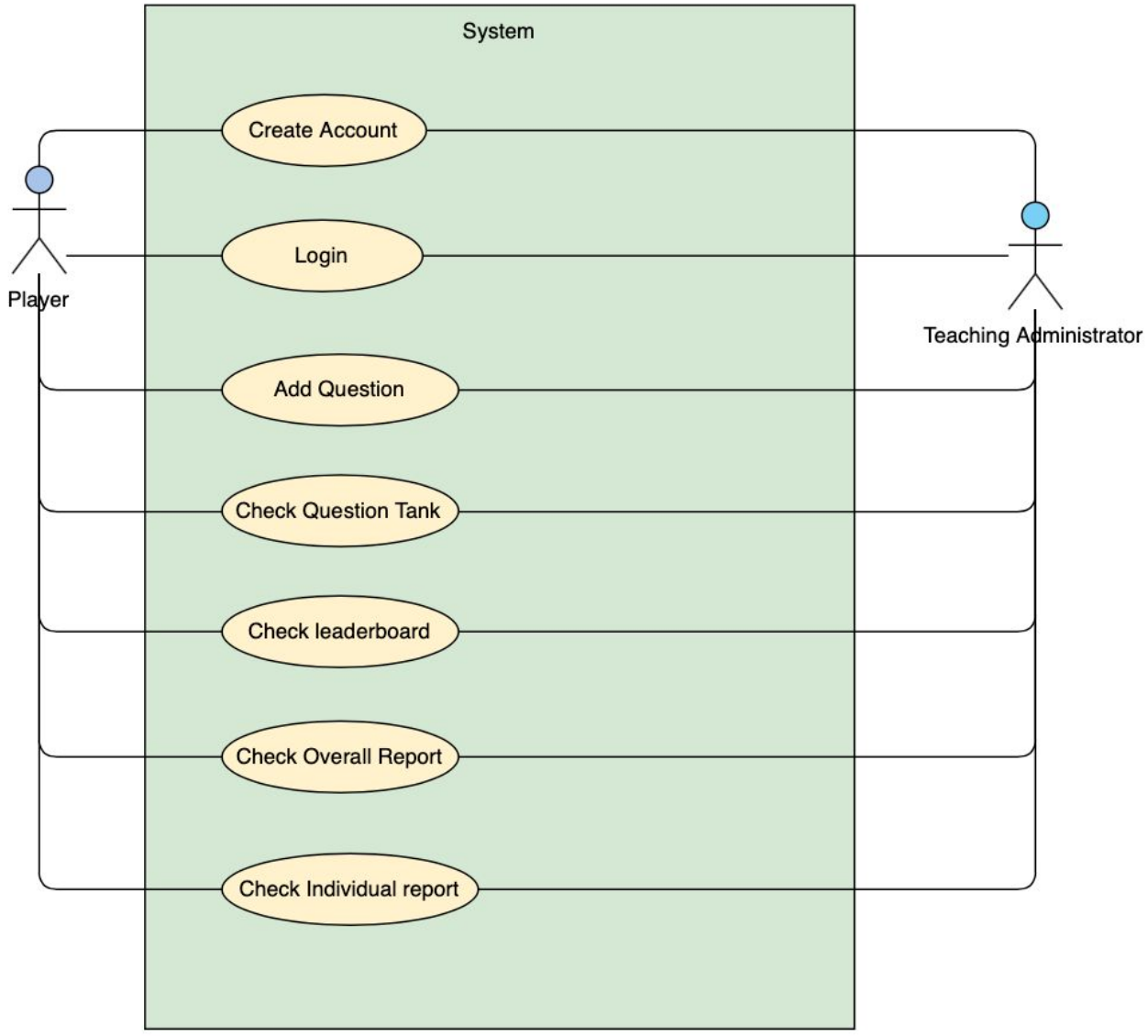
1. The system shall use a light-weight database engine like SQLite, which allows the system to be conveniently migrated on different machines.
2. The system shall be developed using Unity.
3. As part of standard operating procedures, a testing plan will be documented during the design phase. The testing plan consists of three major testing methods including unit testing, integration testing, and system testing based on features, 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. Assumptions and Dependencies

1. Since this application is an android application, a phone with android OS is required.
2. It will be assumed that the users have a good internet connection.
3. The users know the English language, as the user interface will be provided in English.

3. System Features





3.1. Battle Mode[Priority: High]

- 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 and Playered entered one type of world
Postconditions:	Players earned EXP points and occupied the land. OR Players earned 0 EXP points..
Priority:	
Frequency of Use:	
Flow of Events:	Use case 1: enter battle mode 1. Player enters battle mode by clicking at one of the valid land units and clicking the enter battle button. 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 25 seconds. 2. The system displays an enemy, a world-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” for 2 seconds. [Alternate Flows: Wrong] 6. Player starts 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!”. [Alternate Flows: Unsuccessful conquer] 2. Players receive 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.
Alternative Flows:	Use case 2: AF-S5: If the answer is incorrect or time's up. 1. The system displays "Wrong" for 2 seconds. 2. 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. "Back to map" 2.2. "Keep challenging " 3. The user clicks on "Back to map" to enter his map. 4. The user clicks on "Keep challenging" to jump to use case
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 number of correct questions he answered.
      1.1.2. The experience points earned by defeating an enemy is proportional to the difficulty level of the question.

   1.2. The system will not allow players to explore the same land with the same or lower 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. The system requires players to answer a question related to that world type before they can defeat enemies.
        2.1.1. The system will retrieve questions from a question database for players to answer.
        2.1.2. The system should not display the same question for any 3 consecutive rounds.
        2.1.3. Difficulty of questions depend on the battle mode's difficulty level.
        2.1.4. The system requires players to choose an answer in 25 seconds.
        2.1.5. The system provides 4 options for players to choose from.
        2.1.6. The system shall display the amount of time left.
        2.1.7. The system will hide all options when reaching the time limit.
        2.1.8. Players can not defeat enemies if their answers are wrong or they don't choose the answer within the time limit.
        2.1.9. The damage point of players should be proportional to the player's level.

    2.2. The system starts to attack players after players answer a question wrongly.
        2.2.1. The damage point of enemies should be proportional to the difficulty level of the battle that the player has chosen.
    2.3. 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 won't allow players to choose a level easier than the previous level he/she has chosen.
    3.3. Players can explore a land multiple times until they clear the highest difficulty level.

3.2. Duel Mode[Priority: Middle]

- 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 difficulty level he chooses exceeds that of the defender.
- 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 entered a type of world
Postconditions:	Players earned EXP points and occupied his friend's land. OR Players earned 0 EXP points.
Priority:	
Frequency of Use:	
Flow of Events:	Use case 1: enter duel mode 1. Player clicks on the User List button to view the list of friends in a new window. 2. Player clicks one of the friend's name tabs to enter his/her planet. 3. The system changed the view to the target player's planet. 4. Player clicks on one of the land units that has been explored by the target player to stand on. 5. Player clicks on the challenge button. Use case 2: combat with the enemy 9. The system starts the timer for 25 seconds. 10. The system displays an enemy, a world-triggering question, four answer options, and the remaining time on the screen. 11. Player chooses one of the four answer options. 12. The system verifies the correctness of the chosen answer. 13. The system displays "Correct" for 2 seconds. [Alternate Flows: Wrong] 14. Player starts to attack the enemy. 15. The system reduces the Hit point of the enemy. 16. 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 calculates and displays the experience points earned by the attacker. 2. If the enemy's HP reduces to 0, the system displays "Congratulations! You have successfully conquered the land!"

	[Alternate Flows: Unsuccessful conquer] 3. The system updates the total experience points of the player. 4. 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. Use case 4: Return to his/her friend's 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 or time's up. 3. The system displays "Wrong" for 2 seconds. 4. Jump to step 8. Use case 3: AF-S3: If the attacker fail to win the battle 1. The system displays "Practice makes perfect, keep trying!" . 2. The system displays three buttons with options: 2.1. "Keep challenging " 2.2. "Back to Map" 3. The player clicks on "Keep challenging" to jump to use case 2. 4. The player clicks on "Back to Map" to go back to his/her friend's 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 a higher level of difficulty as what the defender has already achieved: If a player has conquered the land with difficult levels, then other players cannot challenge on that land.
   1.2. The system only allows players to take over lands that are already conquered by the other player.

1.3. Lands taken over by attackers can be taken back by the defender only if he/she can successfully defeat a higher level enemy than attackers. If an attacker has conquered the land with difficult levels, then the original owner cannot take back that land.

2. The system will grant experiences to the players if they successfully conquered other player's lands.
   2.1. The rewards are proportional to the level of difficulty.

3.3. World Map[Priority: High]

- 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 the other user's name 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:	Players can view how a world have been explored in the map
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 24 bare land units visible. Use case 3: Explore new lands 1. The player can enter battle mode to extend the lands. 2. The land unit is lit up [Alternative flow: monster isn't defeated.] Use case 4: Unlock new world 1. The player can switch to a new planet only if he has explored equal or over 6 lands of each of his unlocked world 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. Land is not conquered. 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(software engineers), and quality assurance engineers (QA) after creating an account.
        1.1.2. Each character can accumulate experience in his/her dominated field, which means the character will have higher damage points during battles in the corresponding world according to his/her Level.
    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 software engineering, software architecture, quality assurance, project manager.

1.2.2. Players may choose to enter a new world when they have conquered ¼ of all lands in the current planets.

2. World design
   2.1. The planet is designed as a map composed of multiple square units which represent the lands of the planet. Each planet will have its own representative color.
   2.2. The player may start from any of the unexplored lands.
   2.3. To take over someone else's land, the player can click the user list to enter others' planet.

3.4. Teaching Administrator[Priority: Low]

- Data Dictionary

Term	Definition
Question id	Each question has a unique id.
Question details	includes its id, description, all choices, correct answer, category, difficulty level, accuracy
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 or add 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 click on ‘Question Tank’ 2. The system displays all the questions currently in use categorized by their difficulty levels. 3. The administrator is able to filter the questions by Question ID, Accuracy(low to high), Accuracy(high to low) and Title. 4. The system displays the result according to sorting method 5. The administrator click on any question to view question details 6. The system displays question details. 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 “Question added” and the first 20 characters for that problem. Use case 5: obtain performance reports 1. The administrator clicks on the ‘ overall report’ button. 2. The system displays the overall performance report.
Alternative Flows:	Use case 5: AF-S1: if the administrator requests for checking individual reports. 1. The administrator clicks on the ‘Individual Report’ button. 2. The system displays performance reports by individuals.

	3. The administrator is able to sort individual reports by Student ID, Accuracy(low to high), Accuracy(high to low) and student Name. 4. The system displays performance according to its sorting method. 5. The administrator is able to click on any individuals to check on performance details.
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 sort the questions by Question ID, Accuracy(low to high), Accuracy(high to low) and Title.
   2.2. The administrator can view the question details.

3. The administrator should be able to manage questions.
   3.1. The administrator should be able to add new system questions.
   3.2. The administrator should be able to check all questions.
   3.3. The administrator should be able to check the details of each question.

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.
      4.1.1. The administrator can also view performance reports by individual.
   4.2. The administrator can access the leaderboard.
      4.2.1. The administrator can access the complete leaderboard.

3.5. Check Profile[Priority: Middle]

- Description and Priority

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

- Use Case Description

Use Case ID:	
Use Case Name:	Check profile use case

Actor:	Players
Description:	Players check profile
Preconditions:	Players clicks profile button
Postconditions:	Players clicks close button
Priority:	
Frequency of Use:	
Flow of Events:	Use case 1: 1. Users click the “Profile” button on the homepage. 2. The system queries the gaming parameters from the database using the player’s id. 3. Users check the overall gaming parameters displayed on the profile page. 4. Users click the bars or the character images to check parameters of other worlds 5. Users can click on the return button on the top left corner to return to the homepage.
Alternative Flows:	Use case 2: AF-S3: Users can click the column chart for Software Engineer Experience Point to switch to the Profile for Software Engineer world. Use case 3: AF-S3: Users can click the column chart for Software Architecture Experience Point to switch to the Profile for Software Architecture world. Use case 4: AF-S3: Users can click the column chart for Product Manager Experience Point to switch to the Profile for Product Manager world. Use case 5: AF-S3: Users can click the column chart for Quality Assurance Experience Point to switch to the Profile for Quality Assurance World.
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 current level of the character which ranges from 1 to infinity
    1.2. The system shall track the experience point of user which ranges from 0 to infinity
    1.3. The system shall track the the damage point of the user which ranges from 20 to infinity
    1.4. The system shall track the number of questions the user has answered.
    1.5. The system shall track the accuracy of answered questions.
2. There should be a “Profile” button on the homepage.
3. All players should be able to check their status by pressing the button anytime.
4. The status is displayed by changing to another scene with a return button on the top left corner.

3.6. Report[Priority: High]

- 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 “Report” button
Postconditions:	Players/Teaching administrators clicks return button
Priority:	
Frequency of Use:	
Flow of Events:	Use Case 1: Player check the individual report 1. Players clicks on the ‘Report’ button on homepage 2. Report-generating-system queries the accuracy of each world and total accuracy, which are displayed on the left side. 3. Report-generating-system queries the email, ranking and online time of the user, which are displayed on the right side. Use Case 2: Teaching administrators check the list of individual by ranking ID/Name/Accuracy AF-S1: 1. Teaching administrators click on the ‘Individual Report’ button on the teacher's home page. 2. Report-generating-system queries list of students and display them with ascending order of ID/Accuracy/Name. 3. Teaching administrators click on the entry of the report list. 4. The system queries the detailed report of the selected user and display the report to the teaching administrator 5. The teaching administrator clicks the return button to return to the home page . Use Case 3: Teaching administrators check the overall report 1. Teaching administrators click on the ‘Overall Report’ button on the teacher's home page. 2. Report-generating-system queries overall accuracy, the accuracy rate of each world, and total online time.
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 with student-id, email, ranking of that student(based on experience point), experience point, overall accuracy, accuracy of each world, and length of online period.
3. The database keeps the statistics of all worlds.
4. A button named "Report"' should be placed on the student's homepage
5. A button named "Overall Report" should be placed on the teacher's homepage.
6. A button named "Individual Report" should be placed on the teacher's homepage
7. All players should be able to check their reports by pressing the button anytime.
8. The individual report or the overall report is displayed in a window with a return button on the top left corner.
9. The list of users' accuracy will be displayed with a scroll down bar and a return button on the top left corner.
10. The report can be closed by the close button (or by clicking outside the window).
11. In Student mode, total accuracy, accuracy for each world, email, ranking of experience points and online time are displayed in the report.
12. In Administrator mode, the average of all students' accuracy, each world accuracy and length of online period is shown.
13. In Admin mode, a list of students is displayed in a scrolling-enabled scene with descending order of ID in default.
14. In Admin mode, the teaching admin can choose to sort the report list based on ascending order of student id, ascending order of accuracy, descending order of accuracy or ascending order of student name
15. In Admin mode, administrators can access each student's information by clicking on the student in the list.

3.7. Leaderboard[Priority: Low]

- Description

  A leaderboard is a scoreboard showing the names and current experience points 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:	The Student/Teacher clicks the return button on the top left corner
Priority:	
Frequency of Use:	
Flow of Events:	Use case 1: 1. The Student/Teacher clicks on the 'Leaderboard' button on homepage 2. The Leaderboard Displaying System queries the top 10 users' names and overall experience points.
Alternative Flows:	Use case 2: AF-S2: Users can click a button to switch to the Software Engineer World Leader board. Use case 3: AF-S2: Users can click a button to switch to the Software Architecture World Leader board. Use case 4: AF-S2: Users can click a button to switch to the Product Manager World Leader board. Use case 5: AF-S2: Users can click a button to switch to the Quality Assurance World Leader board.
Exceptions:	EX-S2: 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. There should be a leaderboard for each world and for all four worlds.
2. The leaderboard contains three columns, named 'Rank', 'Name', and 'Experience Points'' 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. A button named ‘Leaderboard’ should be placed on the homepage
   5.1. The gamers should be able to check the scoreboard by pressing the button anytime
6. The leaderboard is displayed with a return button on the top left corner.

3.8. Account Management[Priority: Low]

- 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 has a valid email account and matric/staff card AND 2. 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/her username, which is valid and limited by 15 characters 3. The user keys in his email account 4. The user keys in his/her real name as Matric/Admin card, which is limited by 25 characters

	5. The user sets his password, which satisfies the requirements 6. The user reenters and successfully confirms his password 7. The user clicks “Sign Up”. Subsequently, all inputs are successfully saved by the game system and the account is created
Alternative Flows:	AF-S4 if the user selected “Sign up as Admin” 1. The user keys in his Admin card number 2. The flow continues from step 5
Exceptions:	EX-S6 if the reentered password does not match the previously entered password 1. The system displays the message “Invalid input” 2. The user is taken back to step 5
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 due to wrong input
Priority:	
Frequency of Use:	
Flow of Events:	1. The user keys in the email 2. The user keys in the password

	3. The user clicks the Login button 4. The system captures the input and checks if the input email is in email format 5. The system checks the captured input with the database 6. The input matches one of the entries in the database and the user successfully logs in.
Alternative Flows:	
Exceptions:	EX-S3 The email is in wrong format 1. The system displays the message “Invalid input” 2. The user clicks the ”OK” button of the alert window 3. The system returns to step one and waits for new input EX-S2 The password is incorrect or the email is not found in the database 1. The system displays the message “user not found / password wrong!” 2. The input fields are cleared 3. The user clicks the “OK” button of the alert window 4. The system returns 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 limited to 15 characters only. Underscore and digit are allowed
    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
        1.2.2. 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
    1.3. Users are required to enter his/her 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. Users are required to confirm his/her password by entering the password again

2. Users should be able to login at the entry page of the game

2.1. The user needs to enter his/her email account

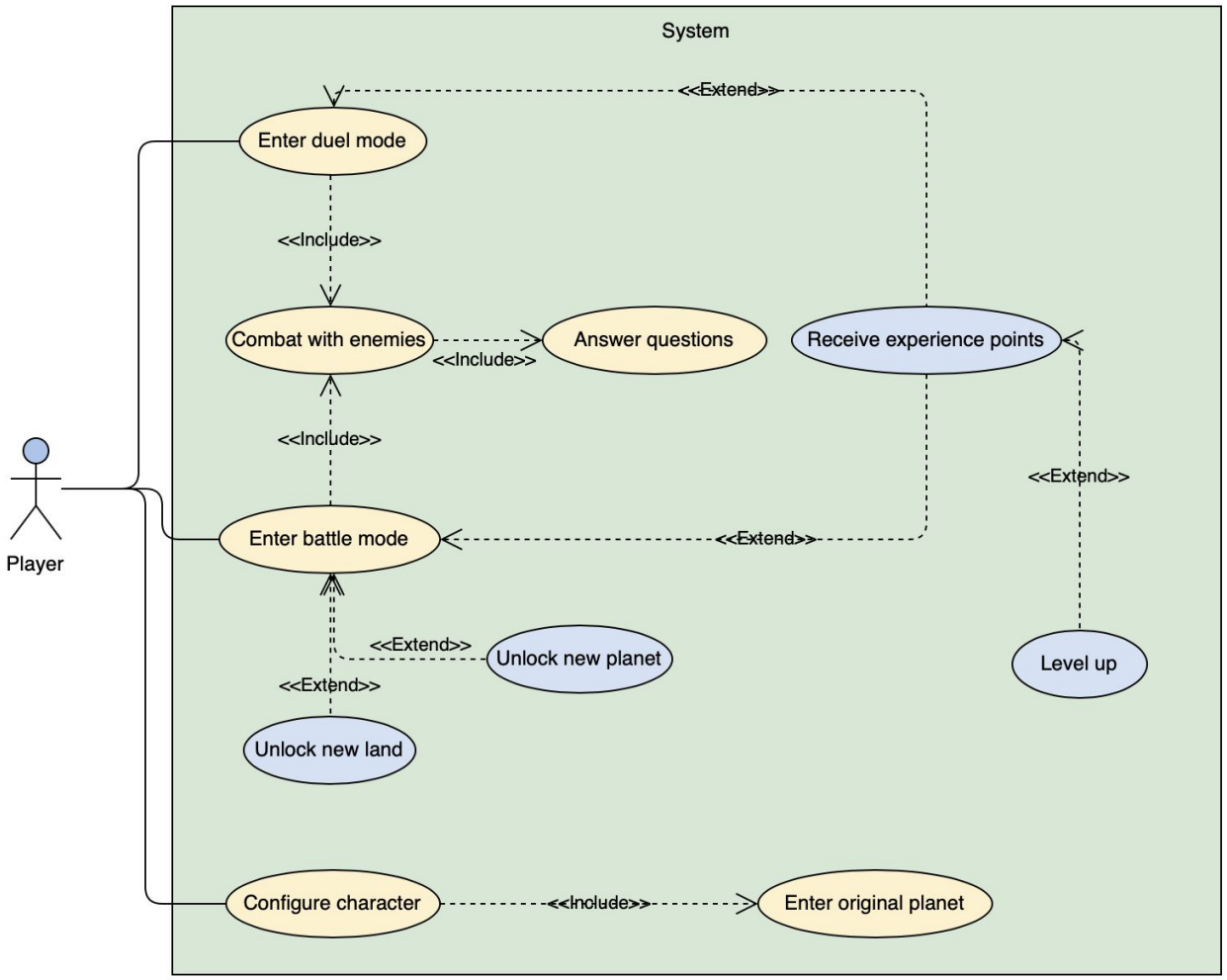
2.1.1. The entry is case-insensitive

2.1.2. if the email format is wrong, the alert window with the message “Invalid Input” 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. If the username/email combination entered does not exist, the message ”user not found / wrong password ” would be displayed



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. System testing will be used to ensure the user interface is bug free, 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) with proper alerts and notifications when users perform inappropriate activities.

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, Spring Boot 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. Viability Requirements

SR-1: The SoftwareGenius system shall be able to handle all the exceptions created by the inappropriate operations of users, including fail to fulfill database constraints, fail

to follow game rules, or fail to meet the authentication standards. The system will operate as usual to keep rebost and provide the users with proper alerts or notifications.

5.2. Security Requirements

SCR-1: The SoftwareGenius system will follow industry best practices for authentication. Authentication addresses security requirements to ensure those using the 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 answer questions 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.
World-triggering question	A world-triggering question is a world category related question that players must answer correctly to attack enemies in a battle.
Hit point	The amount of damage a character can withstand before it is defeated. Each player and enemy has 100 Hit points initially.
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 proportional to Level of character. Damage point refers to the amount of hit points a skill can reduce on the opponents. The minimum damage point is 20.
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 24 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 infinite 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 leaderboard, check reports, check question tanks, and update question tanks.
Student account	Accounts for students at the university so that they can play the game and allow the system to track their progress
Leaderboard	A leaderboard is a scoreboard showing the names and current experience points of the leading gamers, sorted by score in descending order.
Friend	Players can connect with other players and view them as friends.

Appendix B: Priority Table

Priority Table									
Feature	**Benefit (1-9)**	**Penalty (1-9)**	**Total value**	**value %**	**cost**	**cost %**	**risk**	**risk %**	**priority**
Account management	6	7	13	13.68%	6	13.64%	6	19.35%	41.48%
World map	5	5	10	10.53%	4	9.09%	3	9.68%	56.09%
Battle mode	9	9	18	18.95%	7	15.91%	5	16.13%	59.14%
Duel mode	8	8	16	16.84%	8	18.18%	4	12.90%	54.18%
Check player status	5	3	8	8.42%	5	11.36%	2	6.45%	47.27%
Leaderboard	5	3	8	8.42%	4	9.09%	3	9.68%	44.87%
Generate report	7	5	12	12.63%	4	9.09%	3	9.68%	67.30%
Teaching administrator	5	5	10	10.53%	6	13.64%	5	16.13%	35.36%
Total	50	45	95	100.00%	44	100.00%	31	100.00%	