CZ3003 - Software System Analysis & Design

Software Requirements Specification

Project Name: Game of Thrones
Group Name: Team TWO
Lab group: TDDP1
Date of Submission: 07/09/2021

Group Member	Matriculation Number
Chee Jia Yuan	U1921773A
Chong Jie Sheng	U1920968D
Ernest Ang Cheng Han	U1921310H
Joshua Toh Sheng Jie	U1921471F
Koh Tzi Yong	U1920076C
Lek Zhi Ying	U1922765K
Leong Hao Zhi	U1920469K
Li Zheng Jun, Jefferson	U1922429B
Lim Guo Quan	U1920769A
Loo Yi Ying Phoebe	U1921301A
Remus Neo Keng Long	U1922206F
Tan Wei Lun	U1822950L

Table of Contents

Revision History **4**

1 Introduction **1**

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Product Scope 2

1.5 References 2

2 Overall Description **3**

2.1 Product Perspective 3

2.2 Product Functions 3

2.3 User Classes and Characteristics 5

2.4 Operating Environment 5

2.5 Design and Implementation Constraints 5

2.6 User Documentation 5

2.7 Assumptions and Dependencies 5

3 External Interface Requirements **6**

3.1 User Interfaces 6

3.2 Hardware Interfaces 6

3.3 Software Interfaces 8

3.4 Communications Interfaces 10

4 System Features **11**

4.1 Use Case Diagram 11

4.2 Authentication System 13

4.3 Gameplay Mode 1 (Individual Mode) 16

4.4 Gameplay Mode 2 (PvP Mode) 20

4.5 Gameplay Mode 3 (Team Mode) 22

4.6 Leaderboard 25

4.6.1 View Leaderboard 25

4.6.2 Leaderboard System 26

4.7 Question System 28

4.7.1 Managing Question Bank 28

4.7.1.1 Filter Questions 28

4.7.1.2 Modify Question Bank 29

4.8 Assignment System **31**
4.8.1.1 Create & Modify Assignments 31
4.8.1.1.2 Social Media Sharing **32**
4.9 Report System **33**
4.10 Customization of Questions **35**
4.11 Broadcast announcement **35**

5 Other Nonfunctional Requirements **36**
5.1 Performance Requirements 37
5.2 Security Requirements 37

Appendix A: Glossary **39**
A1: Data Dictionary 39

Appendix B: To Be Determined List **41**
B1 Entity Relationship Diagram 41
B2 API Documentation (OpenAPI 3.0) 41
B3 Context Diagram 41
B4 Context Diagram (Student) 42
B5 Context Diagram (Teacher) 43
B6 Class Diagram **44**
B7 Dialog map (Teacher) 46
B8 Dialog map (Student) 47
B9 Dialog map (Individual mode) 48
B10 Dialog map (Team mode) 49
B11 Dialog map (PVP mode) 50

Revision History

Name	Date	Reason For Changes	Version
Ernest Ang	31/8/2021	Document Creation	1.0
All	1/9/2021	Document Revision	1.0

1 Introduction

1.1 Purpose

The Teaching, Learning and Pedagogy Division (TLPD) of NTU is attempting to introduce a social game to gamify and socialize teaching and learning of software engineering courses. Students can learn and compete with each other via playing the game, and teachers can assess the mastery of course via data analysis.

Hence, TeamTWO has decided to create Game of Thrones, a gamified learning platform largely based on the extremely popular American Drama Television series “Game of Thrones”, in order to facilitate a fun and interesting platform for students to learn in a competitive environment yet lacking the conventional stresses incurred from examinations and assignments,

1.2 Document Conventions

The convention used in this document is based on the IEEE Software Requirements Specification template. The document uses the Arial font. The text in the document will have a font size of 12, while formatting for headers will be as follows:

- Section Header : Font size 20, Bold
- Sub-section: Font size 16, Bold

1.3 Intended Audience and Reading Suggestions

This document is intended to be read by key stakeholders associated with the project . It covers the scope of the project, the overall description, interface requirements as well as both functional and non-functional requirements. The table below will identify the key stakeholders, and provide reading suggestions for these different personnel.

Stakeholders	**Purpose**	**Relevant sections and suggestions**
Developers	Understanding of important requirements, current capabilities/functionalities	(Section 2) Overall Description (Section 4) System Features (Section 5) Other Non-functional Requirements

Students (Users)	Understand different functions, which can allow them to have optimal user experience and learn software engineering	(Section 2.2) Product Functions (Section 3.1) User Interface
Teachers (Users)	Monitor and analyze progress of students playing the game	(Section 2.2) Product Functions (Section 3.1) User Interface
Software Test Engineers, Quality Assurance Engineers	Understand system to produce testing strategies and detect issues	(Section 2.1) Product Perspective (Section 2.5) Design and Implementation Constraints (Section 4) System Features (Section 5.2-5.3) Safety and Security Requirements

1.4 Product Scope

Game of Thrones is an application that aims to create a better learning experience for students, and to provide greater engagement in learning Software Engineering. Teachers will be able to at the same time monitor and keep track of student's progress, allowing them to tweak and adjust their teaching based on the information they can observe.

Game of Thrones comprises two main components, namely the game interface for students and also the administrative interface for teachers.

The game interface will allow students to register from login with their NTU account and play the educational game. This will test students on their knowledge of the different phases in the life cycle of software engineering, which ranges from requirements engineering to implementation and testing.

The administrative interface for teachers will allow them to manage and modify the learning materials, allowing for more effective learning for the students. The ability to track progress can allow them to have a better understanding of difficult concepts that students struggle to grasp.

1.5 References

NILL

2 Overall Description

2.1 Product Perspective

Our product, Game of Thrones, is a game that will be built to be played eventually on both mobile phones and computers and is compatible with different operating systems such as Android, iOS, Windows, and MacOS.

It is an action-adventure game where players can choose a character in which they can explore the game world with. There are 5 quests that represent the 5 phases of a software engineering lifecycle that players can go on which contain subquests. In the subquests, players can fight other game characters while answering questions to complete the quest and learn about the different phases in a software engineering cycle. For each quest the student completes, he/she will earn points which will be calculated into the global leaderboard and rank system.

Players can also issue challenges to other players, where they will be able to set the quest of the challenge along with the difficulty level of the questions to be given out in the challenge. The player which completes the challenge within the shortest amount of time will earn additional points which will be calculated into the global leaderboard and rank system.

Players can also go on quests with other players as a team or against in challenge mode with web camera and microphone function. The game aspect of point accumulation from completing quests, multiplayer interactive mode and a rank system aims to motivate the player's learning.

Teachers are admin users who are able to assign specific quests to students by providing them with a unique game code. Teachers are also able to access the analytics of the student's progress. Thus, allowing both active monitoring and assessment of the player's learning through the game in real time. Teachers will also be able to design and assign ad-hoc assignments for students to earn additional points which will be calculated into the global leaderboard and rank system.

2.2 Product Functions

1. Only students with an NTU account will be able to play our game
2. Students can use their NTU email address to login to the game
3. Students can select the character they prefer to play the game with.
4. Students can select and play different quests.

5. Each quest represents a stage of the 5 stages of the software development lifecycle.
6. Each quest will have various subquests which students must complete. Each subquest will get students to answer a series of questions.
7. Students must follow a certain order when completing subquests within a quest
8. Students must at least get 80% of the questions correct for each subquest to proceed to the next subquest.
9. Students will be awarded points for every question they answer correctly
10. Students must complete the entire quest within a certain amount of time, otherwise there will be penalties incurred on the total amount of points he will earn from the whole quest
11. Based on the total number of points a student has, students will be allocated a "rank" and be placed on a leaderboard.
12. Each quest can be attempted multiple times, with the attempt having the highest amount of points earned taken to calculate a student's rank and his/her position on the leaderboard
13. The questions given to each student when completing a quest will be adjusted based on the student's rank (i.e. students with a higher rank will be given questions of greater difficulty).
14. Students will be able to view their rank and their position on the leaderboard
15. Students will have a "PvP" mode where students will be able to challenge each other on a selected quest and set the difficulty of questions given out during the quest. Extra points will be awarded for students who play this mode in addition to the points they have already earned from the same quest used in the challenge (points are stackable). Questions to be attempted during "PvP" mode will be strictly based on the set difficulty designed by students without influence from their rank (see point 13 for more information).
16. Students will have a "Team" mode where they will attempt quizzes together as a team. In this case, all students in the team will be awarded points based on how many questions they answer correctly while completing it on time. Points earned from quests completed in "Team" mode will not be stackable (i.e. students who have already completed the quest individually will not be able to stack points earned from the same quest completed as a team).
17. Students will be able to complete assignments created by Teachers
18. There will be a separate administration interface for Teachers
19. Teachers will be able to create, read, update and delete (CRUD) questions from the question bank
20. Teachers will be able to set the difficulty and category of the questions.
21. Teachers will be able to check the progress and performance of any student

22. Teachers will be able to check the statistics of every question
23. Teachers will be able to create assignments for students to complete on an ad-hoc basis
24. Teachers will be able to share assignments on FaceBook and Twitter.

2.3 User Classes and Characteristics

1. Students: Students are the general users who play the game to learn about the lifecycle of software engineering. Students of all levels of software engineering knowledge are suitable as difficulty of questions are adjusted based on rank.
2. Teachers: Teachers are the admin users of the game. They are able to change the questions for the quests and assign students to specific quests. They also have access to the analytics of the student's progress.

2.4 Operating Environment

The game can be played on any device and operating system that Unity supports which include mobiles such as iOS, iPadOS, Android, and also laptops and desktops such as Windows or MacOS.

2.5 Design and Implementation Constraints

1. The game graphics will be developed with Unity
2. The game interface will only be in English

2.6 User Documentation

There will be an in-game tutorial explaining how the game works.

2.7 Assumptions and Dependencies

Students and teachers have at least 1 supported device.
Students and teachers have good internet connection as the game is played online.
Students and teachers know English as the user interface is provided in English.

3 External Interface Requirements

3.1 User Interfaces

The game interface will be made using the Unity game engine and the default display language will be English. The administration interface will be made using React Javascript, with Redux as our central state management tool. The backend that will be shared by both interfaces will be written using Fast API (Python) and the database will be using PostgreSQL. More details about the technology stack we intend to use for our interfaces can be found under 3.3 Software Interfaces.

With reference to Shneiderman's Eight Golden Rules of interface design, for both interfaces, we will offer simple error handling, and permit easy reversal of actions. A concise and simple interface would also be designed in order to reduce short-term memory load. There will also be prompts that signifies the completion of a task to comply with the principle of designing dialogue to yield closure.

3.2 Hardware Interfaces

To run the game, users need to own a computer/laptop that can run Windows/macOS/Linux or a mobile device that runs on a Android/iOS/tvOS operating system. The required specifications of the respective operating systems are as shown below.

Desktop/Notebook			
Operating system	***Version***	***CPU***	***Graphics Card/API***
Windows	Windows 7 (SP1+) Windows 10	x86, x64 architecture with SSE2 instruction set support	DX10, 11 & 12
Universal Windows Platform	Windows 10, Xbox One, HoloLens	x86, x64 architecture with SSE2 instruction set support ARM, ARM64	DX10, 11 & 12

macOS	High Sierra 10.13+ or newer versions	x86, x64 architecture with SSE2 instruction set support	Metal-capable Intel AMD GPUs
Linux	Ubuntu 20.04 Ubuntu 18.04 CentOS 7	x64 architecture with SSE2 instruction set support	OpenGL 3.2+ or Vulkan-capable
Mobile			
Operating system	***Version***	***CPU***	***Graphics Card/API***
Android (requires 1GB+ RAM)	4.4 (API 19)+	ARMv7 with Neon Support (32-bit) or ARM64	OpenGL ES 2.0+, OpenGL ES 3.0+, Vulkan
iOS	11+	A7 SoC+	Metal
tvOS	11+	A8 SoC+	Metal

The user is also required to have a working wifi/4G LTE connection to enable smooth gameplay. The game does not need other custom hardware protocols.

3.3 Software Interfaces

Overall Architecture - the backend is separated into 2 main services, each adopting its own data schema, database, application server, and API endpoints. The respective software versions and communications are illustrated in the table below

Software Used	Version	Description
PostgreSQL	13.4	**PostgreSQL** is a powerful, open source object-relational database system with over 30 years of active development that has earned it a strong reputation for reliability, feature robustness, and performance. We are using it for relational storage of resources managed by the admin UI service.
Firebase Realtime Database	8.4.1	**Firebase** is a cloud storage server solution which enables services to connect via HTTPS requests. We are using it for object storage of resources managed by the game service.
psycopg	2.9.1	**Psycopg** is a PostgreSQL database adapter for the Python programming language, designed for heavily multi-threaded applications that create and destroy lots of cursors and make a large number of concurrent INSERTs or UPDATEs. We are using it to enable communication between our database and application server in the admin UI service.
React	17.02	**React** is a JavaScript library for building out user interface and state management, and adopts a component-based architecture. We are using it for our front-end admin UI.
Redux	4.1.1	**Redux** is a JavaScript library for managing state at an application level, instead of at a component-level supported by React.
Unity	2020.1	**Unity** is an open-source cross-platform game engine in C#. We are using it for developing the front-end of our game.
Facebook	10.0	**Facebook** is an online social media platform that

Pages API		can be used to share user posts and other content. We are enabling it as one of the platforms that our players can send out challenge invites.
Twitter Standard API	1.1	**Twitter** is another popular online social media platform with a greater focus on short textual content, allowing users to share their thoughts and events in their lives. We are also enabling it as one of the platforms that our players can use to send out challenge invites.

Communication Protocols - Between the backend service and their respective front-ends, communication via HTTP/1 is used, as the services each expose a RESTful API consisting of CRUD operations against the resources they handle. Between the admin UI service and game service, communication via gRPC is used, with the added benefits of HTTP/2 including binary framing for data compression, and request multiplexing to eliminate head-of-line blocking at the application layer

Data Sharing - by design, the resources managed by each service is decoupled, and any sharing of data between the services should be done via their respective APIs. In the event the request-response cycle takes too long (poor performance), we can adopt a pub-sub architecture using Kafka (implementation of a message queue), where all requests are pushed to a central message queue, and each service reacts only to requests that require its input (via subscribing to topics within the queue)

3.4 Communications Interfaces

The following table illustrates the communication protocols and message formats used in the communications between our components.

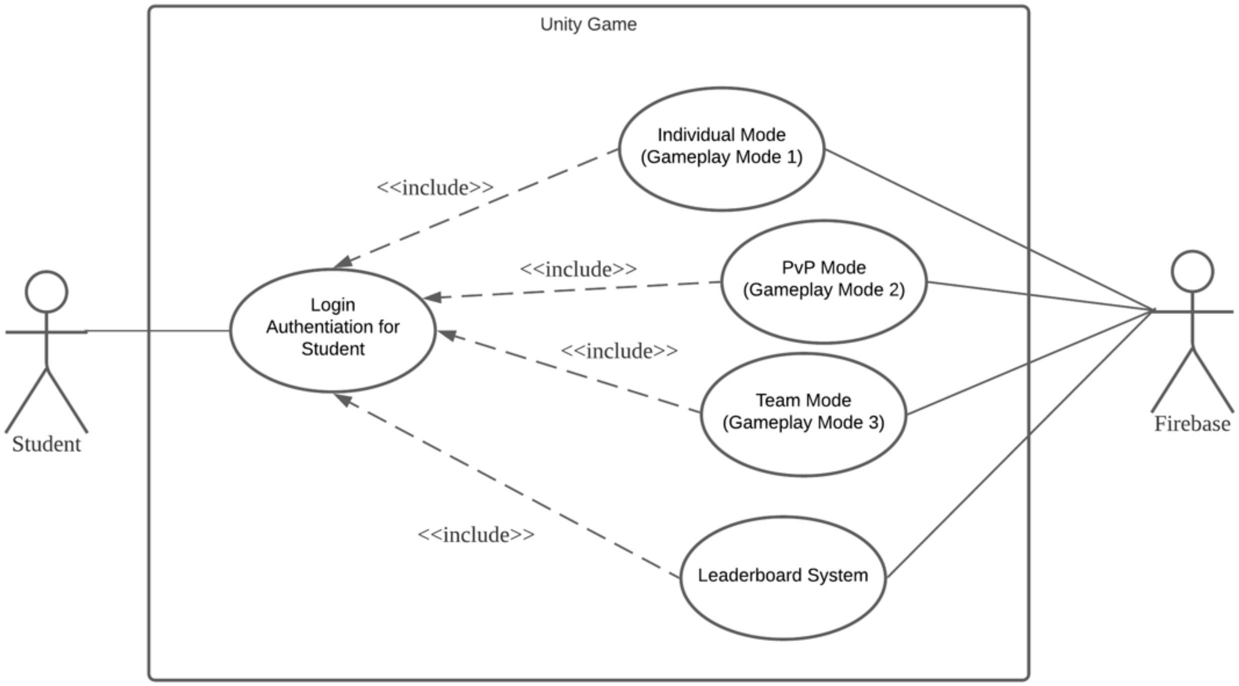
Component	Component	Protocol	Message Format
Admin UI Service	Admin UI Frontend	HTTP/1	Plaintext
Game Service	Game Frontend	HTTP/1	Plaintext
Admin UI Service	Game Service	gRPC, over HTTP/2	User-defined via protocol buffers, and binary at the application layer

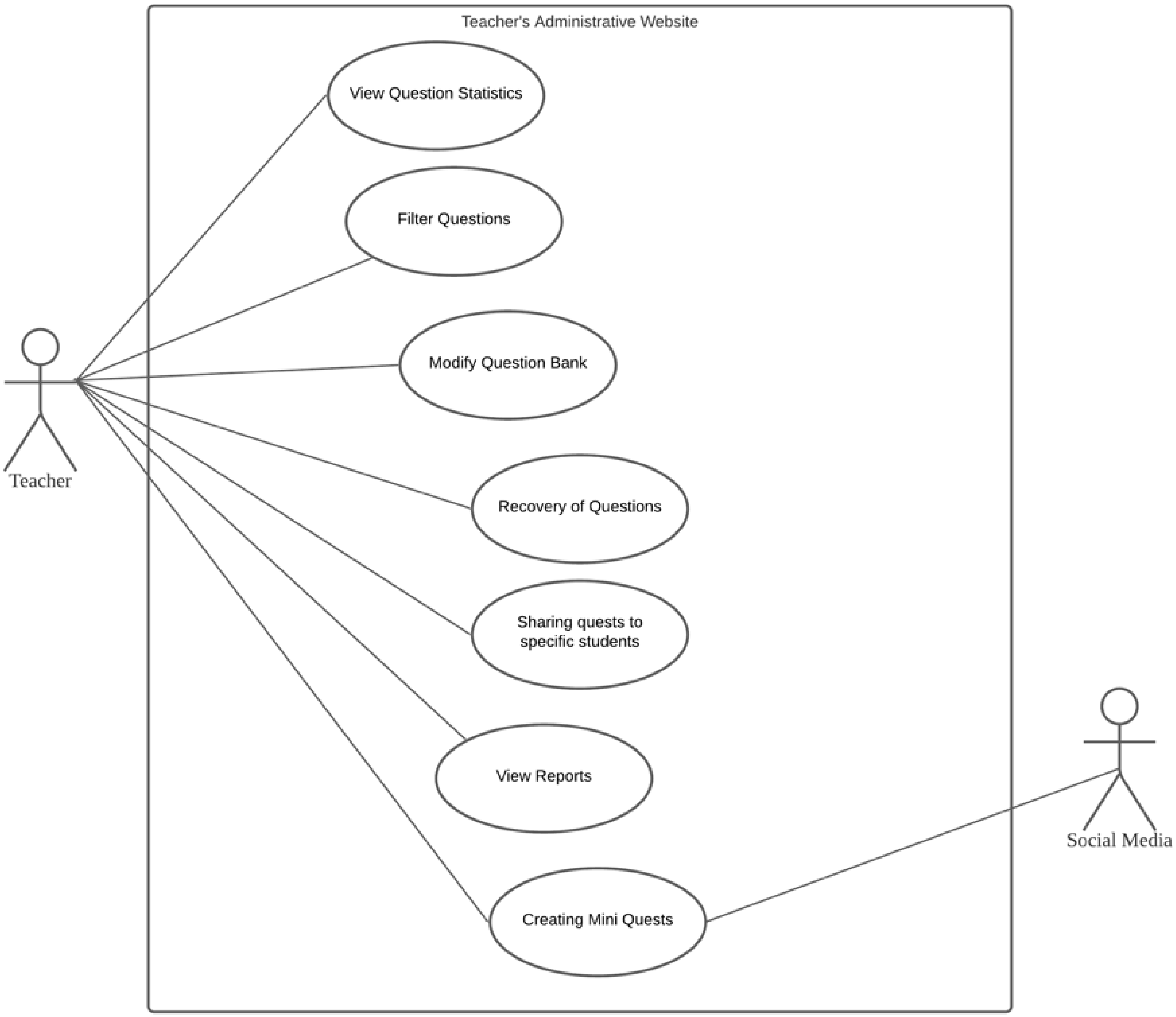
Security Issues - by default, our application servers are configured to only receive and send requests and responses respectively over HTTPS. In other words, TLS encryption is used for all communications above.

Data Transfer Rate - gRPC relies on HTTP/2 at the application layer, which adopts additional features over HTTP/1 such as binary framing (smaller size, higher performance compared to plaintext request/responses in HTTP/1) and request and response multiplexing (multiple request-response cycles can happen in parallel over a single HTTP connection) which eliminates HTTP head-of-line blocking, at the application layer. Unfortunately, TCP head-of-line blocking, at the transport layer, cannot be resolved, and hence only a slight performance boost is expected over HTTP/1

4 System Features

4.1 Use Case Diagram


Unity Game
Individual Mode (Gameplay Mode 1)
PvP Mode (Gameplay Mode 2)
Team Mode (Gameplay Mode 3)
Leaderboard System
Login Authentiation for Student
<<include>>
<<include>>
<<include>>
<<include>>
Student
Firebase


Teacher's Administrative Website
View Question Statistics
Filter Questions
Modify Question Bank
Recovery of Questions
Sharing quests to specific students
View Reports
Creating Mini Quests
Teacher
Social Media

4.2 Authentication System

Use Case ID:	U001		
Use Case Name:	Login Authentication for Student		
Created By:	Tzi Yong	Last Updated By:	Tzi Yong
Date Created:	4th Sept 2021	Date Last Updated:	4th Sept 2021

Actor:	Student
Description:	The Student must be able to login with their valid verification code at the login page to be able to use the application
Preconditions:	1. The system must be active 2. Student must have a NTU Network User Name
Postconditions:	Students can enter the game
Priority:	High Priority (Functional Requirement)
Frequency of Use:	Everytime Student wants to access the game.
Flow of Events:	1. Student enters their NTU Network User Name and click "enter". 2. System checks whether Student is signed up as a player. 3. An email containing a 6-digit verification pin will be sent to the Student's NTU email address. 4. The system will prompt the Student to enter the verification pin. 5. Student enters verification pin at the login page. 6. Login page changes to character customization page. 7. Student selects 1 of the available characters to enter the game with. 8. Student enters the lobby.
Alternative Flows:	AF-S1: NTU Network User Name not valid 1. The system will request Student for a valid NTU Network User Name again. 2. The Student enters their NTU Network User Name and clicks enter. 3. If NTU Network User Name is valid, move to Step 2. 4. If NTU Network User Name is not valid, move to AF-S1.

	AF-S2: The User Account has yet to be signed up by the system automatically 1. The system will determine whether it has received a valid NTU Network User Name 2. The system sends an email to the Teacher to approve sign up for Student 3. The system creates a game account for the Student using the Student's NTU email address. AF-S4: Verification pin is not valid 1. System will inform Student that verification pin is not valid 2. The Student can request for another verification pin to be sent to the email address. 3. Student enters the verification pin again at the login page. 4. If the verification pin is valid, move to Step 5. 5. If the verification pin is not valid, move to AF-S3. AF-S6: Student enters the game for the first time 1. System will prompt the Student to enter their preferred name to be displayed. 2. Move to Step 6.
Exceptions:	N.A
Includes:	N.A
Special Requirements:	N.A
Assumptions:	N.A
Notes and Issues:	N.A

4.2.1 The system is accessible by two types of users: Students and Teachers.

4.2.1.1 The webapp can only be accessed using the Teacher's login credentials.

4.2.1.2 The game application can only be accessed using the Student's login credentials.

4.2.2 Student Login

4.2.2.1 The system will require the user's Network User Name.

4.2.2.2 The user will receive an email with a 6-digit verification pin.

4.2.2.2.1 The user can request for another verification email after 30 seconds.

4.2.2.3 The system will prompt the user to use the 6-digit pin to enter the game.

4.2.2.3.1 User will remain on the login page if the verification code provided and the verification code entered does not match.

4.2.2.4 The system will check if the user is entering the game for the first time.

4.2.2.4.1 The system will prompt the user for their preferred name to be displayed if the user is entering the game for the first time.

4.2.2.4.2 The system will save the new preferred name into the database.

4.2.2.5 The user will be redirected to the customize characterization screen upon successful login verification.

4.2.2.6 The user can choose 1 of the many available characters to enter the game with.

4.2.2.7 The user will enter the game.

4.3 Gameplay Mode 1 (Individual Mode)

4.3.1 Students will be able to teleport and move in the game to complete a total of 5 quests from 5 different worlds.

- 4.3.1.1 Each quest corresponds to one of the 5 stages of the software development lifecycle.
- 4.3.1.2 For each quest will have various subquests which correspond to the specific topics of each phase in the SLDC.
  - 4.3.1.2.1 Subquests must be unlocked in a specific order.
  - 4.3.1.2.2 Students must get at least 80% of the questions correct for each subquest to proceed to the next subquest.

4.3.2 Students will be able to complete subquests which are in the form of quizzes.

- 4.3.2.1 Subquests (Quizzes) will have 3 difficulty levels.
  - 4.3.2.1.1 Students will unlock the next difficulty level only if the Students get 80% on the previous difficulty level.
  - 4.3.2.1.2 Level 1 subquests have questions with difficulty rating 1 and 2. Level 2 subquests have questions with difficulty rating 3 and 4. Level 3 subquests have questions with difficulty rating 3 and 4
  - 4.3.2.1.3 Subquests at each difficulty level when completed, will give different point multipliers to the Student.
- 4.3.2.2 Subquests consist of a series of 5 to 10 questions.
  - 4.3.2.2.1 Questions will be a mixture of different formats: Multiple Choice Questions, True/False Questions and Multiple Select Questions.
    - 4.3.2.2.1.1 Multiple Choice Questions will have 1 correct answer out of 4.
    - 4.3.2.2.1.2 True/False Questions will have 1 correct answer out of 2 (T/F).
    - 4.3.2.2.1.3 Multiple Select Questions will have 1 or more correct answers out of 4.
  - 4.3.2.2.2 Questions within each difficulty level will vary by difficulty rating. Questions will be selected based on the Student's current rank.
  - 4.3.2.2.3 Students will get 1 point * (difficulty multiplier) for each correct question answered.
  - 4.3.2.2.4 Subquests must be completed within a certain amount of time otherwise penalties will be incurred on the total amount of points earned from the subquest
- 4.3.2.3 Subquests can be completed multiple times; the attempt with the highest amount of points will be stored and taken into account when calculating the Students total points (and rank).

4.3.3 Students will be able to view the subquest screen to do the subquest (take the quiz).

- 4.3.3.1 Verification will be done to ensure the Student has completed prerequisite subquests.
- 4.3.3.2 The subquest screen will display the subquest information and allow for difficulty selection.
- 4.3.3.3 During the quiz, the subquest screen will also display questions and options to choose from.
- 4.3.3.4 After a question is answered, the subquest screen will display the correct answer.

4.3.3.5 At the end of the quiz, scores will be counted and saved.

Use Case ID:	U002		
Use Case Name:	Navigating Worlds (Individual Mode)		
Created By:	Chong Jie Sheng	Last Updated By:	Chong Jie Sheng
Date Created:	2th Sept 2021	Date Last Updated:	5th Sept 2021

Actor:	Student
Description:	The Student must be able to navigate the quests in different worlds. Following which they will be able to access and complete the subquests within each world.
Preconditions:	1. Students are logged in and playing in the game.
Postconditions:	1. Students are teleported to the World of their choice.
Priority:	High Priority (Functional Requirement)
Frequency of Use:	High. Since navigating to quests is required for the Student.
Flow of Events:	1. The Student approaches a teleportation device in the World. 2. System displays an 'interact' icon on the teleportation device. 3. System will verify the Student's progress and based on the criteria display World's that the Student has access to. 4. Student selects a World and clicks the travel button. 5. The Student is transported to the World and may start completing Quests.
Alternative Flows:	AF-S4: Student has not met the criteria to travel to the World. 1. System will display "You have not unlocked this World yet. Please complete the Quests in the previous World."
Exceptions:	N.A
Includes:	N.A
Special Requirements:	N.A
Assumptions:	N.A

Notes and Issues:	N.A

Use Case ID:	U003		
Use Case Name:	Completing a Subquest (Individual Mode)		
Created By:	Chong Jie Sheng	Last Updated By:	Chong Jie Sheng
Date Created:	3th Sept 2021	Date Last Updated:	5th Sept 2021

Actor:	Student
Description:	The Student must be able to access the quests in different worlds. Following which they must be able to access and complete the subquests within each world. To complete the subquests (quizzes), they will have to complete a series of questions with varying difficulty levels to get points.
Preconditions:	1. Students are logged in and playing in the game. 2. Student is currently in a Quest world
Postconditions:	1. Students complete a subquest. 2. Students' total scores are tallied and saved in the Database.
Priority:	High Priority (Functional Requirement)
Frequency of Use:	High. Since completing quests and sub quests are objectives of the same
Flow of Events:	1. The Student approaches an NPC in the World. 2. System displays an “interact” icon on the NPC. 3. System will verify that the Student has met the criteria to start the subquest. 4. Subquest screen would be shown to the Student displaying subquest information and the available difficulty levels based on Student’s progress. 5. Student selects a difficulty level and clicks the start button. 6. System retrieves a set of random questions from the Database based on the Student’s chosen difficulty level and leaderboard rank. 7. System displays the questions and answer options. 8. The Student can select an option from the answers.

	9. System verifies whether the answer is correct and displays the correct answer. 10. System tallies the total points earned for the subquest by the Student (Correct Answers * Difficulty Level). 11. Subquest completed and total scores are updated in the Database.
Alternative Flows:	AF-S3: Student has not met the criteria to start the Subquest. 1. System will display “You have not unlocked this subquest yet. Please complete the previous ones in this world.” 2. System exits the Student from the NPC interaction.
Exceptions:	N.A
Includes:	N.A
Special Requirements:	N.A
Assumptions:	N.A
Notes and Issues:	N.A

4.4 Gameplay Mode 2 (PvP Mode)

4.4.1 Players will be able to send challenge invitations to other players.

4.4.1.1 Players will be able to design challenges by choosing the SDLC stage, topic and overall difficulty.

4.4.1.2 Players will be able to reject or accept challenge invitations.

4.4.1.3 Players will receive points for participating in this player versus player game mode based on their performance in the challenge quiz.

4.4.1.4 Players' points and related statistics will be stored in the database.

Use Case ID:	U005		
Use Case Name:	PvP Mode		
Created By:	Jefferson Li	Last Updated By:	Jefferson Li
Date Created:	2th Sept 2021	Date Last Updated:	6th Sept 2021

Actor:	Student
Description:	Students can enter the challenge mode to compete with other players
Preconditions:	1. Students are logged in and playing in the game. 2. Two students want to challenge another player at the same time.
Postconditions:	Students earned points from completing the challenge.
Priority:	High Priority (Functional Requirement)
Frequency of Use:	Frequent
Flow of Events:	1. The Student approaches an NPC in the World. 2. System displays an "interact" icon on the NPC. 3. System will verify that the Student has met the criteria to start the subquest. 4. Students choose to enter challenge mode by selecting the "challenge mode" button. 5. Student waits for another student to join up. Other students must have selected the challenge mode button as well. 6. The students will create challenges by choosing the difficulty level and question type. The difficulty levels available to choose are the common levels between both players.

	7. Student selects the "start" button and waits for the other student to also select the "start" button. Once both students have selected the "start" button, the subquest begins. 8. System retrieves a set of random questions from the Database based on the Student's chosen difficulty level and leaderboard rank. 9. System displays the questions and answer options. An additional progress bar is available for students to view other student's progress 10. The Student can select an option from the answers. 11. System verifies whether the answer is correct and displays the correct answer. 12. System tallies the total points earned for the subquest by the Student (Correct Answers * Difficulty Level * Time Taken). The winner is determined by the one with the higher score. 13. System will display "You won the challenge" 14. Subquest completed and total scores (Correct Answers * Difficulty Level) are updated in the Database.
Alternative Flows:	AF-S4: No other students selected "challenge mode". 1. System automatically time-outs after 10 minutes and brings the student back to the previous screen. 2. Return to step 3. AF-S12: Student lost the challenge 1. System will display "You lost the challenge".
Exceptions:	N.A
Includes:	N.A
Special Requirements:	N.A
Assumptions:	N.A
Notes and Issues:	N.A

4.5 Gameplay Mode 3 (Team Mode)

4.5.1 Student will be able to create a team with invitation code

- 4.5.1.1 Student will be able to join a team using the invitation code
- 4.5.1.2 Student will be able to leave a team
- 4.5.1.3 Each team can have a maximum of 4 students
- 4.5.1.4 Students in the same team should be able to see each other on the interface
- 4.5.1.5 Students in the same team should be able to start the quest
    - 4.5.1.4.1 Any single student can have the power to start the quest
- 4.5.1.6 Students in the same team should be able to attempt subquests together
    - 4.5.1.6.1 Students must be within a metre from each other for them to start the quiz
    - 4.5.1.6.2 Students in the same team should see the same displayed question on the interface
    - 4.5.1.6.3 Any single student can have the power to answer the question from the subquest
- 4.5.1.7 Students in the same team will all end the quest together once all of the subquest has been completed

Use Case ID:	U006		
Use Case Name:	Creating room in Team Mode		
Created By:	Ernest Ang	Last Updated By:	Ernest Ang
Date Created:	2th Sept 2021	Date Last Updated:	6th Sept 2021

Actor:	Student
Description:	Students can enter the challenge mode to compete with other players
Preconditions:	1. Students are logged in and playing in the game. 2. At least 2 students want to form a team and attempt a quest together
Postconditions:	All students in the party earned the same amount of points from the quest. Points will be added to a player's total points earned if the quest has not been att
Priority:	High Priority (Functional Requirement)
Frequency of Use:	Frequent
Flow of Events:	1. The Student approaches an NPC in the World.

	2. System displays an "Create Team" icon on the NPC. 3. System will allocate student into a room with a 6 digit code 4. Student will wait for at least one more student to join room using 6 digit code 5. Any student can pick a quest and the entire team will start the quest 6. System proceeds to generate the quest, subquests and the relevant questions from the database for the team to complete
Alternative Flows:	AF-S5: No other student join 1. System will display "Team must have at least 2 students" 2. System returns to step 4 AF-S12: Student lost the challenge 1. System will display "You lost the challenge".
Exceptions:	N.A
Includes:	N.A
Special Requirements:	N.A
Assumptions:	N.A
Notes and Issues:	N.A

Use Case ID:	U007		
Use Case Name:	Joining room in Team Mode		
Created By:	Ernest Ang	Last Updated By:	Ernest Ang
Date Created:	2th Sept 2021	Date Last Updated:	6th Sept 2021

Actor:	Student
Description:	Students can enter the challenge mode to compete with other players
Preconditions:	1. Students are logged in and playing in the game.

	2. At least 2 students want to form a team and attempt a quest together
Postconditions:	All students in the party earned the same amount of points from the quest. Points will be added to a player's total points earned if the quest has not been att
Priority:	High Priority (Functional Requirement)
Frequency of Use:	Frequent
Flow of Events:	1. The Student approaches an NPC in the World. 2. System displays an "Join Team" icon on the NPC. 3. Student will input a 6 digit code given by another student 4. Student will join the room on success 5. Any student can pick a quest and the entire team will start the quest 6. System proceeds to generate the quest, subquests and the relevant questions from the database for the team to complete
Alternative Flows:	AF-S5: Invalid 6 digit code 1. System will display "Invalid 6 digit code, no such room exist" 2. System returns to step 4 AF-S12: Student lost the challenge 1. System will display "You lost the challenge".
Exceptions:	N.A
Includes:	N.A
Special Requirements:	N.A
Assumptions:	N.A
Notes and Issues:	N.A

4.6 Leaderboard

4.6.1 View Leaderboard

4.6.1 Students can view both their overall score and other players' overall score

- 4.6.1.1 Students can filter the leaderboard by selecting a particular quest or subquest.
- 4.6.1.2 Students can search players' ranks and score information based on User ID.
- 4.6.1.3 Students can view their ranks with respect to other students.
    - 4.6.1.3.1 Students can view their overall rank.
    - 4.6.1.3.2 Students can view their rank for individual quests or subquests
- 4.6.1.4 Students can refresh the leaderboard by clicking the refresh icon

Use Case ID:	U008		
Use Case Name:	View Leaderboard		
Created By:	Lek Zhi Ying	Last Updated By:	Lek Zhi Ying
Date Created:	5th September 2021	Date Last Updated:	5th September 2021

Actor:	Student
Description:	The leaderboard will be displayed on click of the leaderboard icon. Students will be able to view their individual score and ranking. They will also be able to view the scores and rankings of other students in the game. This will allow the student to compare their result with their peers and encourage them to acquire more points
Preconditions:	Students have an account with the game. Students are logged into the game.
Postconditions:	Leaderboard View is opened with all the rankings of the students shown.
Priority:	High Priority
Frequency of Use:	Frequent
Flow of Events:	1. Students click on the leaderboard icon on the game page. 2. System displays the overall top 50 students' ranking and score information onto the leaderboard.
Alternative Flows:	AF-S2: Student input a UserID into the search input box

	1. System displays the ranking and score information of that particular UserID. AF-S2: Student selects a subquest or quest. 1. System displays the top 50 students' ranking and score information of selected quest or subquest. AF-S2: Student selects the refresh button. 1. System goes to Step 2.
Exceptions:	N.A
Includes:	Leaderboard System
Special Requirements:	N.A
Assumptions:	N.A
Notes and Issues:	N.A

4.6.2 Leaderboard System

4.6.2 System will fetch students information from the database.

4.6.2.1 System will calculate and sort the data into a rank list.

4.6.2.1.1 System will sum up students' score from all quests and subquests into an overall score.

4.6.2.1.2 System will sort the list based on the calculated score.

4.6.2.2 System can filter the data by UserID input.

4.6.2.3 System can select and filter the leaderboard by individual quests or subquests.

4.6.2.4 System will pass the data back to the GUI, which will be used by the "View Leaderboard" use case.

Use Case ID:	U009		
Use Case Name:	Leaderboard System		
Created By:	Lek Zhi Ying	Last Updated By:	Lek Zhi Ying
Date Created:	5th September 2021	Date Last Updated:	5th September 2021

Actor:	Student

Description:	The leaderboard system executes fetching, calculation and filtering of data from the database. The information will then be passed to the leaderboard GUI as explicit in the "View Leaderboard" use case.
Trigger:	View Leaderboard
Preconditions:	Students have an account with the game. Students are logged into the game. Students have the leaderboard view opened.
Postconditions:	Leaderboard for the selected quest and subquest is displayed according to filters and input where applicable.
Priority:	High Priority
Frequency of Use:	Frequent
Flow of Events:	1. System fetches all students' game result information from the database. 2. System passes the information for calculation and sorting. 3. System passes the information to the GUI.
Alternative Flows:	AF-S1: Student selects the refresh button. 1. System goes to Step 1. AF-S2: System receive UserID input 1. System pick out the specific UserID's information 2. System goes to Step 3. AF-S2: System receive a filter request 1. System filter from the fetched information based on filter condition 2. System goes to Step 3.
Exceptions:	N.A
Includes:	N.A
Special Requirements:	N.A
Assumptions:	N.A
Notes and Issues:	N.A

4.7 Question System

4.7.1 Managing Question Bank

4.7.1 Teachers will be able to manage the questions in the game stored in a question bank.

> 4.7.1.1 A filter system will allow the teacher to sort and filter the questions based on the difficulty, sub-quest and points.
> 4.7.1.2 Teachers will be able to create, read, update and delete questions from the question bank.

4.7.1.1 Filter Questions

Use Case ID:	U011		
Use Case Name:	Filter questions according to difficulty, sub-quest, points.		
Created By:	Loo Yi Ying Phoebe	Last Updated By:	Joshua Toh
Date Created:	4th Sept 2021	Date Last Updated:	12th Nov 2021

Actors:	Teacher
Description:	The Teacher must be able to filter questions according to difficulty, sub-quest, points.
Pre-conditions:	Teacher has a working internet connection.
Post-conditions:	Teacher is able to see the list of sorted questions.
Normal Flow:	1. Teacher selects the “Question Bank” tab on the website. 2. Teacher selects on the column dropdown of difficulty, sub-quest or points. 3. Teacher chooses the dropdown option that they want to filter. 4. System displays a list of questions sorted.
Alternative Flow:	N.A.
Exceptions:	N.A.
Includes:	N.A.
Priority:	Medium

Frequency of Use:	Medium
Special Requirements:	N.A.
Assumptions:	N.A.
Notes and Issues:	N.A.

4.7.1.2 Modify Question Bank

Use Case ID:	U012		
Use Case Name:	Modifying question bank		
Created By:	Loo Yi Ying Phoebe	Last Updated By:	Remus Neo
Date Created:	4th Sept 2021	Date Last Updated:	12th Nov 2021

Actors:	Teacher
Description:	The Teacher must be able to create, edit and delete questions from the question bank for the game.
Pre-conditions:	Teacher has a working internet connection.
Post-conditions:	Teacher modifies the question bank successfully and the modified list of questions are shown.
Normal Flow:	1. Teacher selects the “Create” tab on the website. 2. Teacher keys in “Category” followed by “Quest” and then “Subquest”. 3. Teacher selects the difficulty level for the question. 4. Teacher keys in answer options for the question. 5. Teacher selects “Save and add another question”. 6. System shows a confirmation message and the question is added successfully.
Alternative Flow:	AF-S2: Teacher chooses to edit questions. 1. Teacher selects the “Question Bank” tab. 2. Teacher selects a question of choice. 3. Teacher clicks on the “Edit” icon (pencil icon). 4. Teacher keys in updated details. 5. Teacher clicks on the “Tick” icon. 6. Question details updated successfully

Exceptions:	EX1: Teacher selects “Cancel” for confirmation prompts. Modification is incomplete and the original list of questions will be shown.
Includes:	N.A.
Priority:	High
Frequency of Use:	High
Special Requirements:	N.A.
Assumptions:	N.A.
Notes and Issues:	N.A.

4.8 Assignment System

4.8.1 Teachers will be able to assign students a specific assignment.

4.8.1.1 Teachers are able to create a custom assignment which will be assigned to specific students

4.8.1.1.1 Teachers are able to modify & delete question, or delete entire assignment

4.8.1.1.2 Teachers can share updates or announcements through social media (Facebook, Twitter)

4.8.1.1 Create & Modify Assignments

Use Case ID:	U014		
Use Case Name:	Modify assignments		
Created By:	Loo Yi Ying Phoebe	Last Updated By:	Loo Yi Ying Phoebe
Date Created:	4th Sept 2021	Date Last Updated:	19th Sept 2021

Actors:	Teacher
Description:	This feature allows the Teacher to create, edit and delete assignments for students to complete in their own time.
Pre-conditions:	Teacher has an account on the administrative website. Teacher is logged onto the administrative website. Teacher has a working internet connection.
Post-conditions:	The Teacher modifies assignment(s) successfully and can view the list of assignments created.
Normal Flow:	1. Teacher selects “Assign” on the navigation bar. 2. Teacher selects the “Assign” tab. 3. Teacher creates the assignment name and description. 4. Teacher selects the “Add” tab. 5. Teacher selects the assignment of choice to add questions to and keys in the details. 6. Teacher selects “Save and add another question”. 7. System displays an empty question template for Teacher to continue adding questions.
Alternative Flow:	AF-S2: Teacher chooses to remove assignment. 1. Teacher selects “Modify”. 2. Teacher selects the assignment of choice.

	3. Teacher selects “Delete Assignment”. 4. System prompts the Teacher for confirmation. 5. Teacher selects “Delete”. AF-S2: Teacher chooses to edit assignment. 7. Teacher selects the assignment of choice. 8. Teacher selects “Edit Questions”. 9. System displays a list of questions. 10. Teacher edits question details. 11. Teacher selects a tick icon. 12. System displays an updated list of questions.
Exceptions:	EX1: Teacher selects “Cancel” for confirmation prompt. Creation is incomplete and the original list of assignments will be shown.
Includes:	N.A.
Priority:	High
Frequency of Use:	Low
Special Requirements:	N.A.
Assumptions:	N.A.
Notes and Issues:	N.A.

4.8.1.1.2 Social Media Sharing

Use Case ID:	U015		
Use Case Name:	Sharing Quests to specific students		
Created By:	Loo Yi Ying Phoebe	Last Updated By:	Remus Neo
Date Created:	4th Sept 2021	Date Last Updated:	12th Nov 2021

Actors:	Teacher
Description:	This feature allows the Teacher to share updates on assignments through social media to notify students about new assignments available.
Pre-conditions:	Teacher has an account on the administrative website.

	Teacher is logged onto the administrative website. Teacher has a working internet connection. Teacher has already created the quest on the website.
Post-conditions:	Updates or announcement is successfully posted onto the social media post and students will be made known of new assignment
Normal Flow:	1. Teacher selects “Share” under “Assign” tab 2. Teacher can select from 2 social media options: a. Facebook b. Twitter 3. Social media post with the new update/announcement are posted successfully.
Alternative Flow:	N.A.
Exceptions:	N.A.
Includes:	N.A.
Priority:	Low
Frequency of Use:	Low
Special Requirements:	N.A.
Assumptions:	N.A.
Notes and Issues:	N.A.

4.9 Report System

4.9.1 Teachers will be able to view students’ overall performance.

4.9.1.1 Teachers will be able to retrieve an overall cohort performance report which details the best performing student, rank, position, points earned and percentage correct of all students in a page.

4.9.1.2 Teachers will be able to view individual performance reports of each student.

4.9.1.2.1 Teachers will be able view the overall correct rate of the student and the quests attempted.

Use Case ID:	U016		
Use Case Name:	View reports		
Created By:	Loo Yi Ying Phoebe	Last Updated By:	Loo Yi Ying Phoebe

Date Created:	4th Sept 2021	Date Last Updated:	19th Sept 2021

Actors:	Teacher
Description:	The Teacher must be able to view reports detailing the overall performance of the cohort and of each individual student.
Pre-conditions:	Teacher has an account on the administrative website. Teacher is logged onto the administrative website. Teacher has a working internet connection.
Post-conditions:	Teacher is able to view the report.
Normal Flow:	1. Teacher selects “View Report”. 2. Teacher selects “Individual Performance Score”. 3. Teacher selects a student of choice. 4. System displays the report.
Alternative Flow:	AF-S2: Teacher chooses to view report on cohort performance. 1. Teacher selects “Cohort Performance Score”. 2. System returns to step 4.
Exceptions:	N.A.
Includes:	N.A.
Priority:	Low
Frequency of Use:	Low
Special Requirements:	N.A.
Assumptions:	N.A.
Notes and Issues:	N.A.

4.10 Customization of Questions

4.10.1 The difficulty of sub-quests is customized based on the student's past performance, which will be measured by their rank.

Use Case ID:	U017		
Use Case Name:	Customizing difficulty level of questions		
Created By:	Loo Yi Ying Phoebe	Last Updated By:	Loo Yi Ying Phoebe
Date Created:	4th Sept 2021	Date Last Updated:	4th Sept 2021

Actors:	Student
Description:	System customizes subsequent quest levels based on the student's rank, which is indicative of the student's performance.
Pre-conditions:	Student has an account and is logged onto the game. Student has a working internet connection.
Post-conditions:	Student attempts future questions at the adjusted difficulty level.
Normal Flow:	1. Student starts the game. 2. System increases the quest difficulty level based on the Student's rank.
Alternative Flow:	N.A.
Exceptions:	N.A.
Includes:	4.3 Gameplay Mode 1 (Individual Mode)
Priority:	High
Frequency of Use:	High
Special Requirements:	N.A.
Assumptions:	N.A.
Notes and Issues:	N.A.

4.11 Broadcast announcement

4.11.1 Teachers can broadcast announcements to students through email.

Use Case ID:	U018		
Use Case Name:	Broadcast Announcement		
Created By:	Joshua Toh	Last Updated By:	Joshua Toh
Date Created:	12th Nov 2021	Date Last Updated:	12th Nov 2021

Actors:	Teacher
Description:	Teachers can broadcast announcements to students through email.
Pre-conditions:	Teacher has a working internet connection. Teacher has a working email address. Students has a working email address.
Post-conditions:	Students are able to view the email
Normal Flow:	1. Teacher clicks on the home button. 2. Teacher selects the sender address. 3. Teachers select one or more recipients. 4. Teacher enters the subject, header and announcement text. 5. Teacher presses the Post button and send out the email to students.
Alternative Flow:	N.A.
Exceptions:	N.A.
Includes:	N.A.
Priority:	High
Frequency of Use:	High
Special Requirements:	N.A.
Assumptions:	N.A.
Notes and Issues:	N.A.

5 Other Nonfunctional Requirements

We will focus on performance for our game interface and security for the administration interface.

It is crucial that our game interface performs well under high load due to the fact that realistically, hundreds of students may be playing our game simultaneously within a period of time.

For the administration interface, performance is not as important given that the number of teachers are far less than the number of students, and it is unlikely that multiple teachers will be using the interface within a period of time. Hence, the priority should be on security given that the administration interface will have access to the student data which are sensitive in nature.

5.1 Performance Requirements

5.1.1 The game interface should be able to efficiently access data on the student's particulars and questions.

5.1.1.1 The game server must be able to query the database within 0.5 seconds.

5.1.1.2 The game server must be able to authenticate students within 1 second

5.1.1.3 The game server must be able to register students into the database within 1 second

5.1.1.4 The game server must be able to load questions based on the quest within 5 seconds

5.1.2 The game interface should be able to update and send data to students

5.1.2.1 The game server must be able to verify player's answer within 1 second

5.1.2.2 The game server must be able to update the database within 5 seconds after the quest has been completed

5.1.2.3 The game servers must be able to update student's profile within 2 second2

5.1.2.4 The game servers must be able to update leaderboard within 5 seconds

5.1.2.5 The game server must be able to send a "PvP" request within 1 second

5.1.2.6 The game server must be able to add "Mini-Quest" within 1 second

5.1.3 The game server must be able to serve 100 students at any point of time.

5.2 Security Requirements

5.2.1 The administration interface must be free of web-related security flaws

5.2.1.1 The administration interface must not be susceptible to SQL Injections

5.2.1.2 The administration interface must not be susceptible to Cross-Site Scripting

5.2.1.3 The administration interface must not be susceptible to parameter tampering

5.2.1.4 The administration interface must not be susceptible to cookie/token stealing

5.2.1.5 The administration interface must not be susceptible to session hijacking

5.2.1.6 Authentication for administration interface must not be broken

5.2.1.7 The administration interface must not be susceptible to Cross Site Request Forgery

5.2.1.8 The administration interface must enforce protective routing

5.2.1.9 The administration interface must restrict url access

Appendix A: Glossary

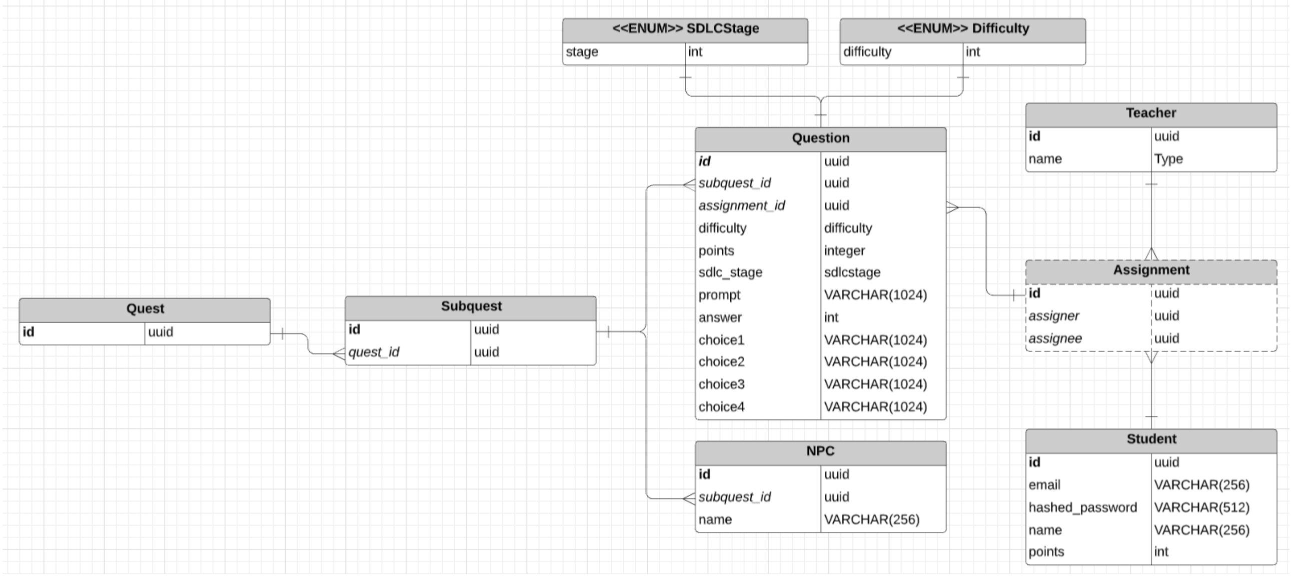
A1: Data Dictionary

Terms	Description
Game Interface	Refers to the game portion of our entire system where students will play and learn from
Administration Interface	Refers to the interface where teachers will be able to add questions to the question bank.
Student	Refers to students who are enrolled in the course and will go on and play the game
Teacher	Refers to teachers who are lecturers, tutors, or teacher assistants involved in teaching the course
User	Refers to both students and teachers, utilized in use cases involving both students and teachers
Quest	Refers to the virtual world which the student will enter in the game. Within this virtual world, there will be multiple subquests for students to complete. Each quest will have subquests which will quiz students on each stage of the software development lifecycle.
Subquest	Refers to a subquest within a quest. Each subquest will hold a task for students to complete (e.g. find a landmark and interact with a NPC, who will ask you 10 questions).
Assignments	Refers to the ad-hoc assignments which Teachers can create for students to attempt. Teachers can also share assignments on social media.
NPCs	Non-player character (NPC) which will be situated around the quest. Students will interact with these NPCs as part of their subquest in order to complete the entire quest
Rank	Students will be allocated ranks based on the number of points he/she has earned so far: 0 - 100 points: Herald 100 - 200 points: Guardian 200 - 300 points: Crusader 300 - 400 points: Archon 400 - 500 points: Legend 500 - 600 points: Ancient

	600 - 700 points: Divine 700 - 800 points: Immortal 800 - 900 points: Genesis 900 - 1000 points: Challenger 1000 points < : Platinum
Difficulties	Questions are categorized into 3 difficulty levels: Easy, Medium, and Hard. Questions given across the subquests are adjusted according to the student's rank
Question	Each question that a student answers correctly within a quest earns 10 points. The amount of points earned from a quest will only be added to the student's account if he completes the entire quest (i.e. if student gives up halfway, points he/she has earned so far from answering questions correctly within subquest will be voided and lost)
Question bank	Refers to the database which will contain all questions added by teachers, along with the possible options, the correct answer, the difficulty category, and the topic.
Individual report	Refers to a summary of all of the attempts made by a student for each of the quest, the highest attempt scored by the student for a quest, along with the detailed performance by the student on the quest (e.g. questions that were given, answers given by student, correct/incorrect attempt made by the student)
Question Report	Refers to the summary of all of the questions, along with the list of students who have attempted the same question and the percentage of these students who attempted it correctly.

Appendix B: To Be Determined List

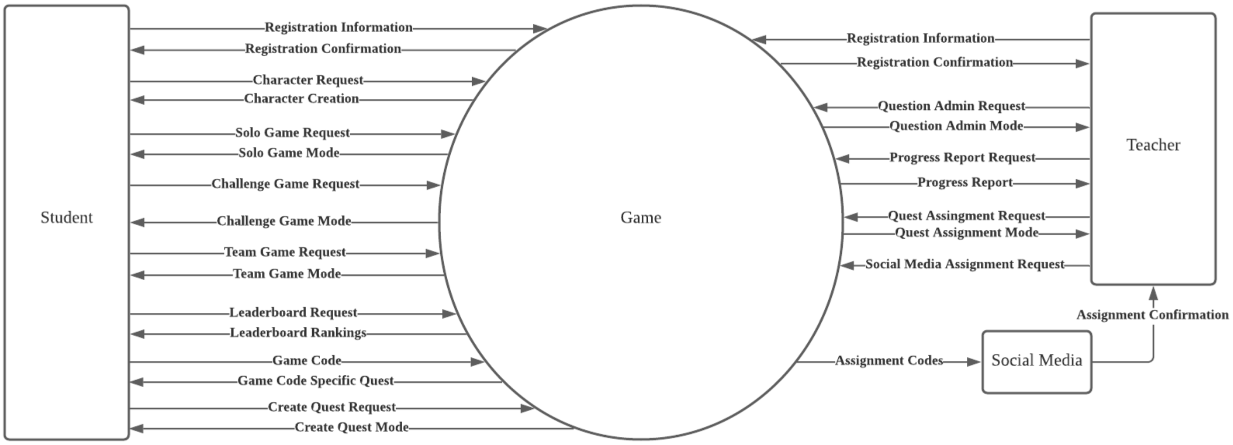
B1 Entity Relationship Diagram



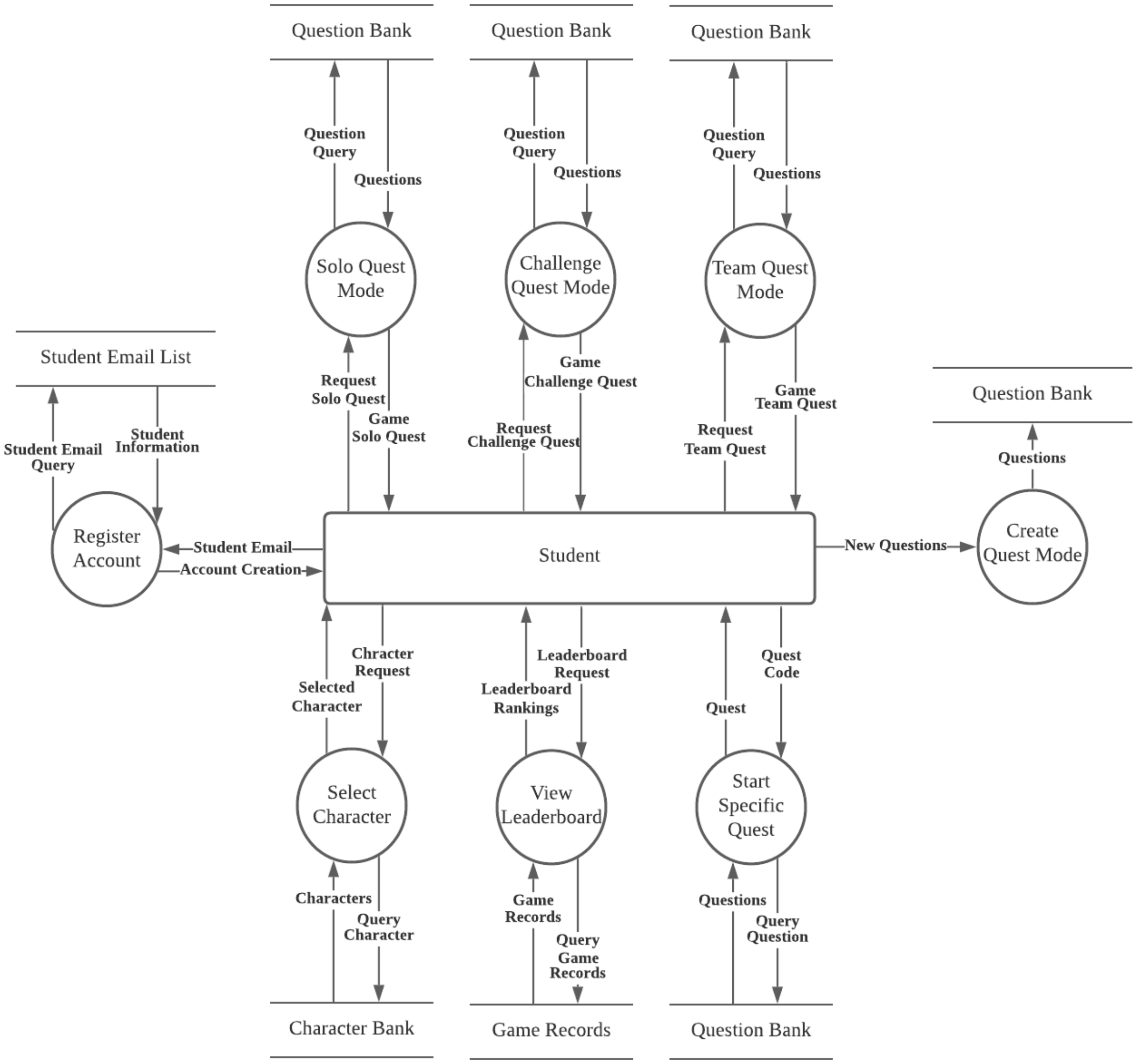
B2 API Documentation (OpenAPI 3.0)

https://fastapi-ernestang98.cloud.okteto.net/docs

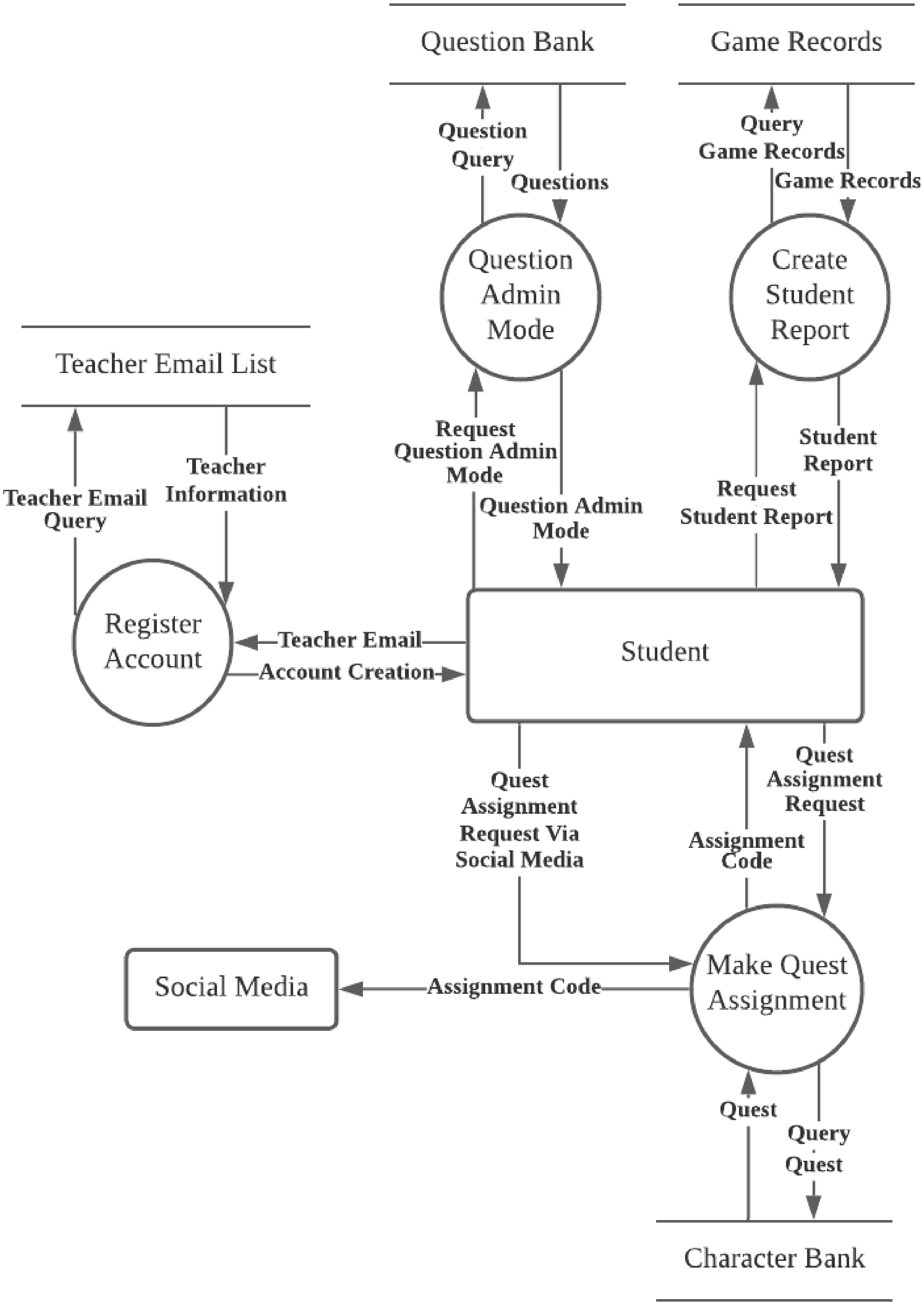
B3 Context Diagram



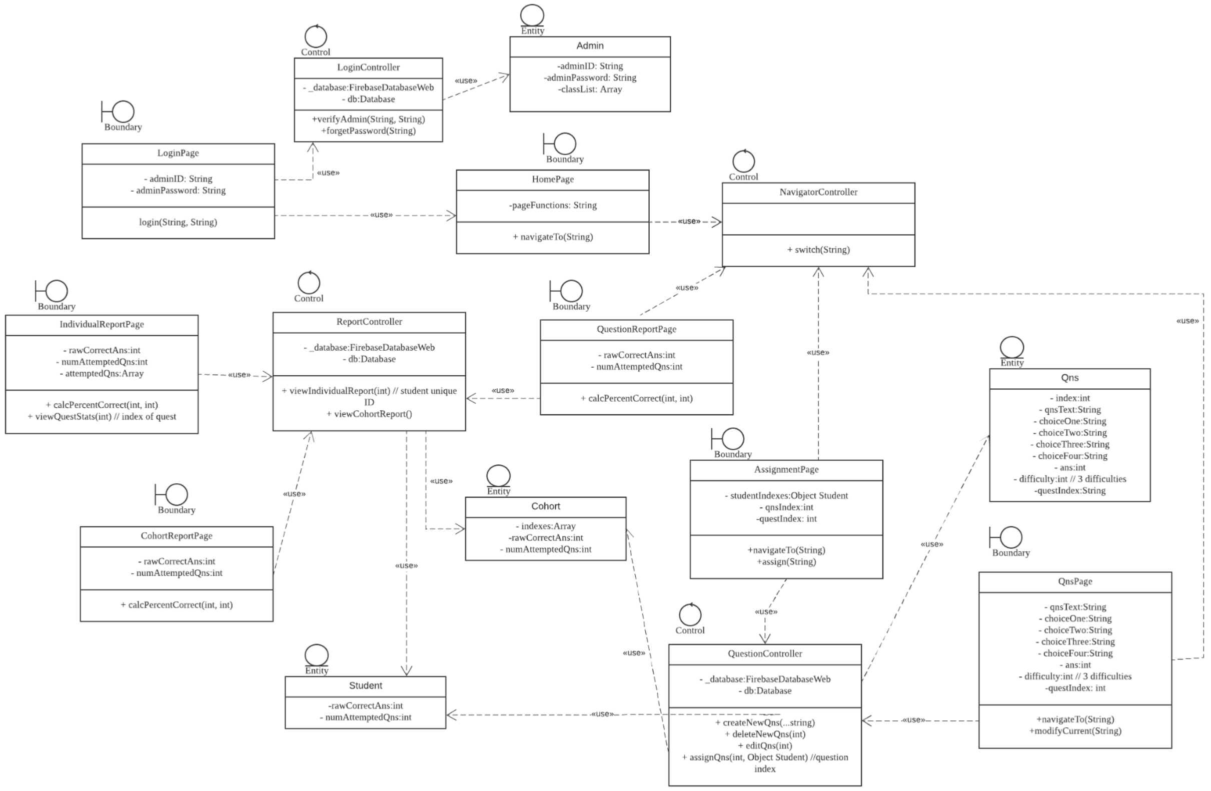
B4 Context Diagram (Student)

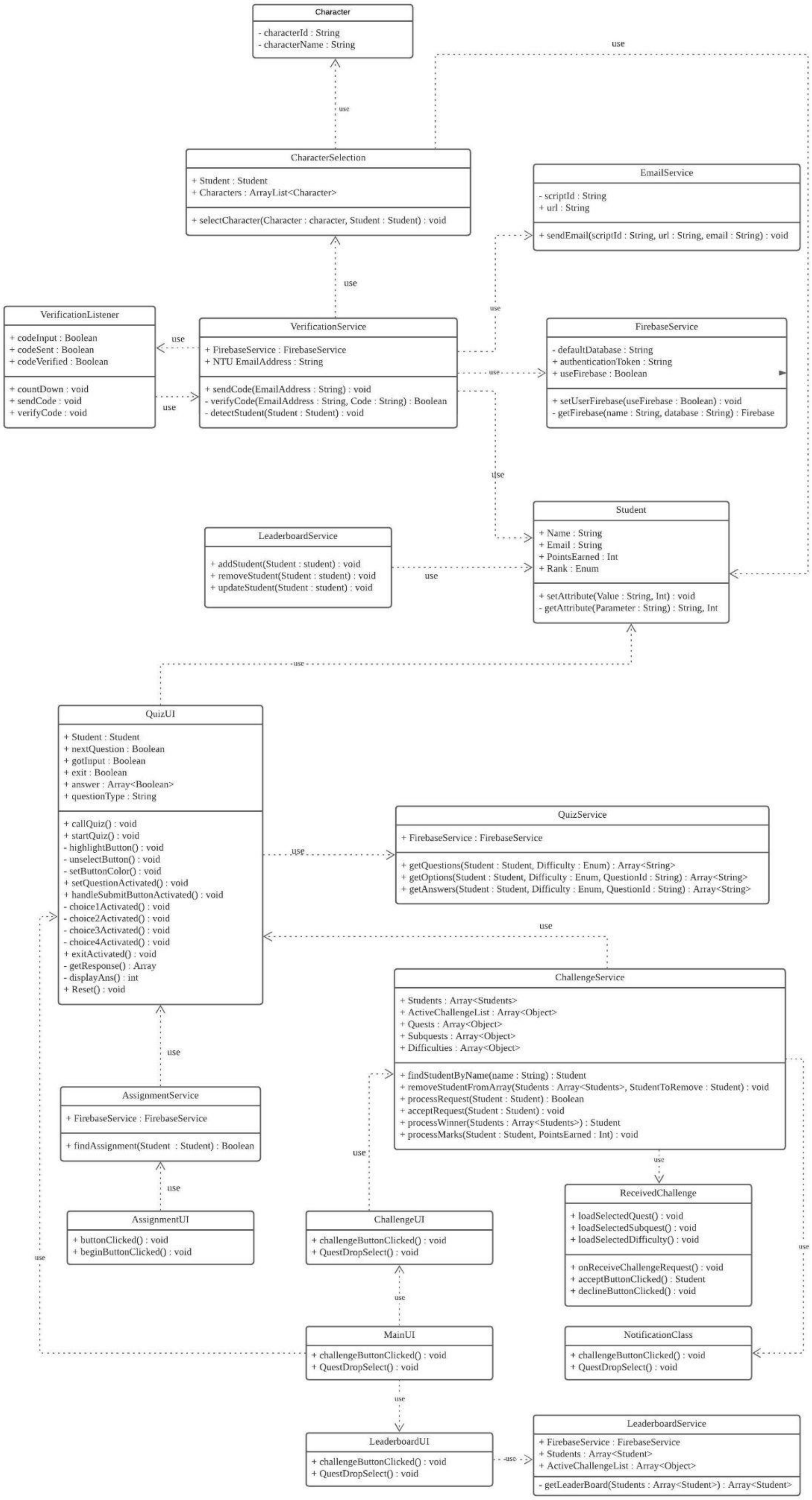


B5 Context Diagram (Teacher)

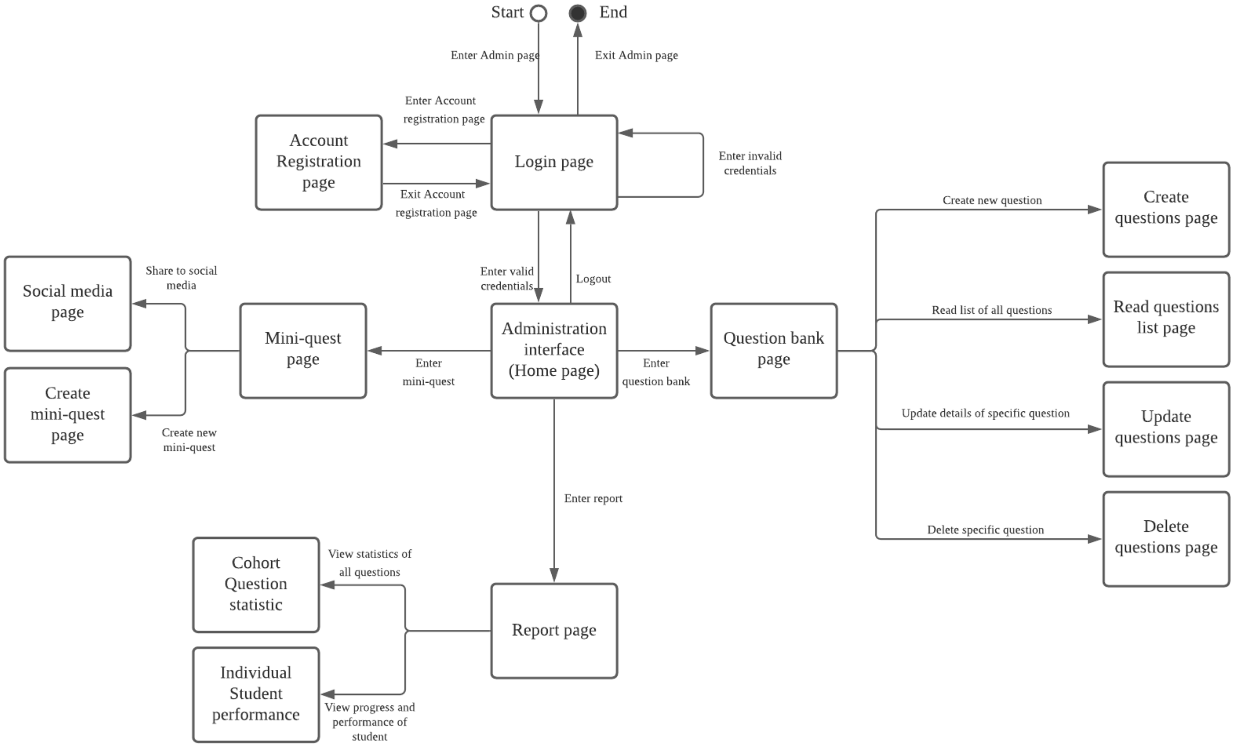


B6 Class Diagram

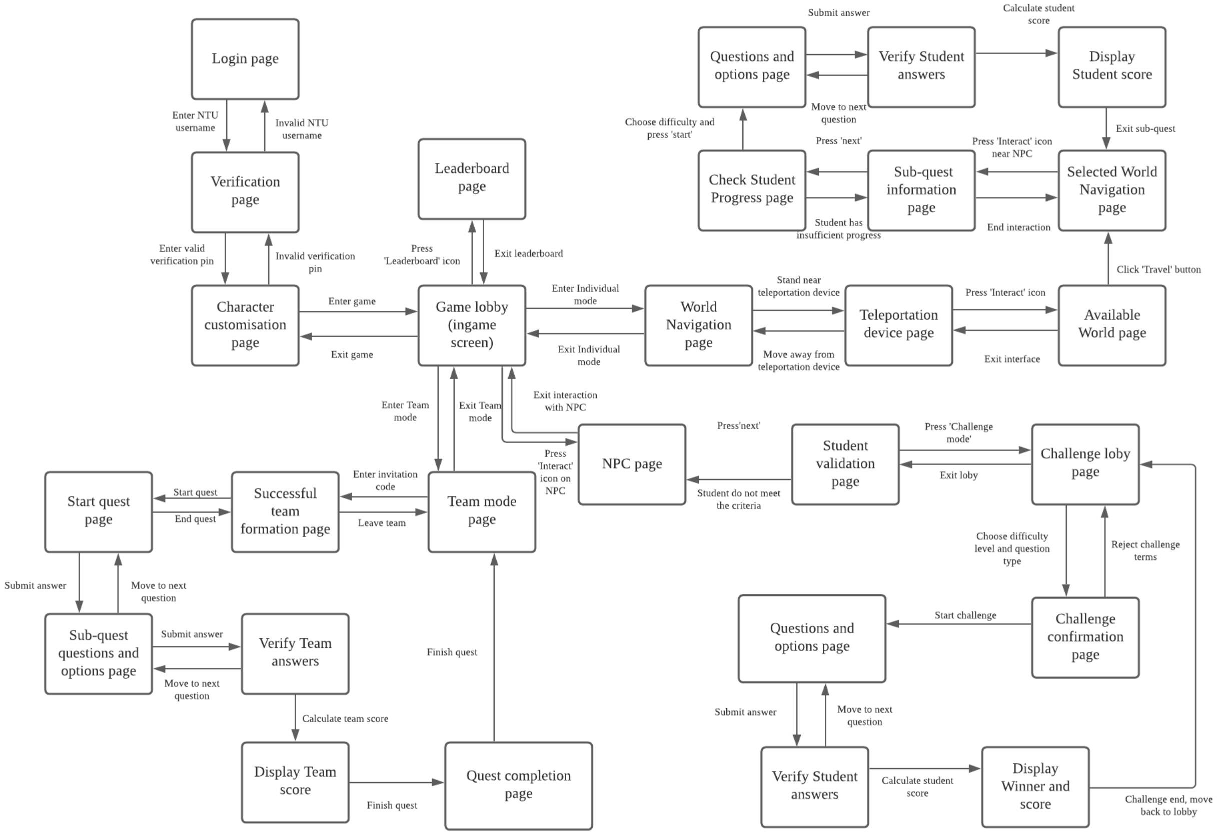



Character
- characterId : String
- characterName : String
use
use
CharacterSelection
+ Student : Student
+ Characters : ArrayList<Character>
+ selectCharacter(Character : character, Student : Student) : void
EmailService
- scriptId : String
+ url : String
+ sendEmail(scriptId : String, url : String, email : String) : void
use
use
VerificationListener
+ codeInput : Boolean
+ codeSent : Boolean
+ codeVerified : Boolean
+ countDown : void
+ sendCode : void
+ verifyCode : void
use
use
VerificationService
+ FirebaseService : FirebaseService
+ NTU EmailAddress : String
+ sendCode(EmailAddress : String) : void
- verifyCode(EmailAddress : String, Code : String) : Boolean
- detectStudent(Student : Student) : void
use
FirebaseService
- defaultDatabase : String
+ authenticationToken : String
+ useFirebase : Boolean
+ setUserFirebase(useFirebase : Boolean) : void
- getFirebase(name : String, database : String) : Firebase
use
Student
+ Name : String
+ Email : String
+ PointsEarned : Int
+ Rank : Enum
+ setAttribute(Value : String, Int) : void
- getAttribute(Parameter : String) : String, Int
LeaderboardService
+ addStudent(Student : student) : void
+ removeStudent(Student : student) : void
+ updateStudent(Student : student) : void
use
use
QuizUI
+ Student : Student
+ nextQuestion : Boolean
+ gotInput : Boolean
+ exit : Boolean
+ answer : Array<Boolean>
+ questionType : String
+ callQuiz() : void
+ startQuiz() : void
- highlightButton() : void
- unselectButton() : void
- setButtonColor() : void
+ setQuestionActivated() : void
+ handleSubmitButtonActivated() : void
- choice1Activated() : void
- choice2Activated() : void
- choice3Activated() : void
- choice4Activated() : void
+ exitActivated() : void
- getResponse() : Array
- displayAns() : int
+ Reset() : void
use
QuizService
+ FirebaseService : FirebaseService
+ getQuestions(Student : Student, Difficulty : Enum) : Array<String>
+ getOptions(Student : Student, Difficulty : Enum, QuestionId : String) : Array<String>
+ getAnswers(Student : Student, Difficulty : Enum, QuestionId : String) : Array<String>
use
ChallengeService
+ Students : Array<Students>
+ ActiveChallengeList : Array<Object>
+ Quests : Array<Object>
+ Subquests : Array<Object>
+ Difficulties : Array<Object>
+ findStudentByName(name : String) : Student
+ removeStudentFromArray(Students : Array<Students>, StudentToRemove : Student) : void
+ processRequest(Student : Student) : Boolean
+ acceptRequest(Student : Student) : void
+ processWinner(Students : Array<Students>) : Student
+ processMarks(Student : Student, PointsEarned : Int) : void
use
AssignmentService
+ FirebaseService : FirebaseService
+ findAssignment(Student : Student) : Boolean
use
use
use
ReceivedChallenge
+ loadSelectedQuest() : void
+ loadSelectedSubquest() : void
+ loadSelectedDifficulty() : void
+ onReceiveChallengeRequest() : void
+ acceptButtonClicked() : Student
+ declineButtonClicked() : void
use
AssignmentUI
+ buttonClicked() : void
+ beginButtonClicked() : void
ChallengeUI
+ challengeButtonClicked() : void
+ QuestDropSelect() : void
use
use
MainUI
+ challengeButtonClicked() : void
+ QuestDropSelect() : void
NotificationClass
+ challengeButtonClicked() : void
+ QuestDropSelect() : void
use
LeaderboardUI
+ challengeButtonClicked() : void
+ QuestDropSelect() : void
use
LeaderboardService
+ FirebaseService : FirebaseService
+ Students : Array<Student>
+ ActiveChallengeList : Array<Object>
- getLeaderBoard(Students : Array<Student>) : Array<Student>

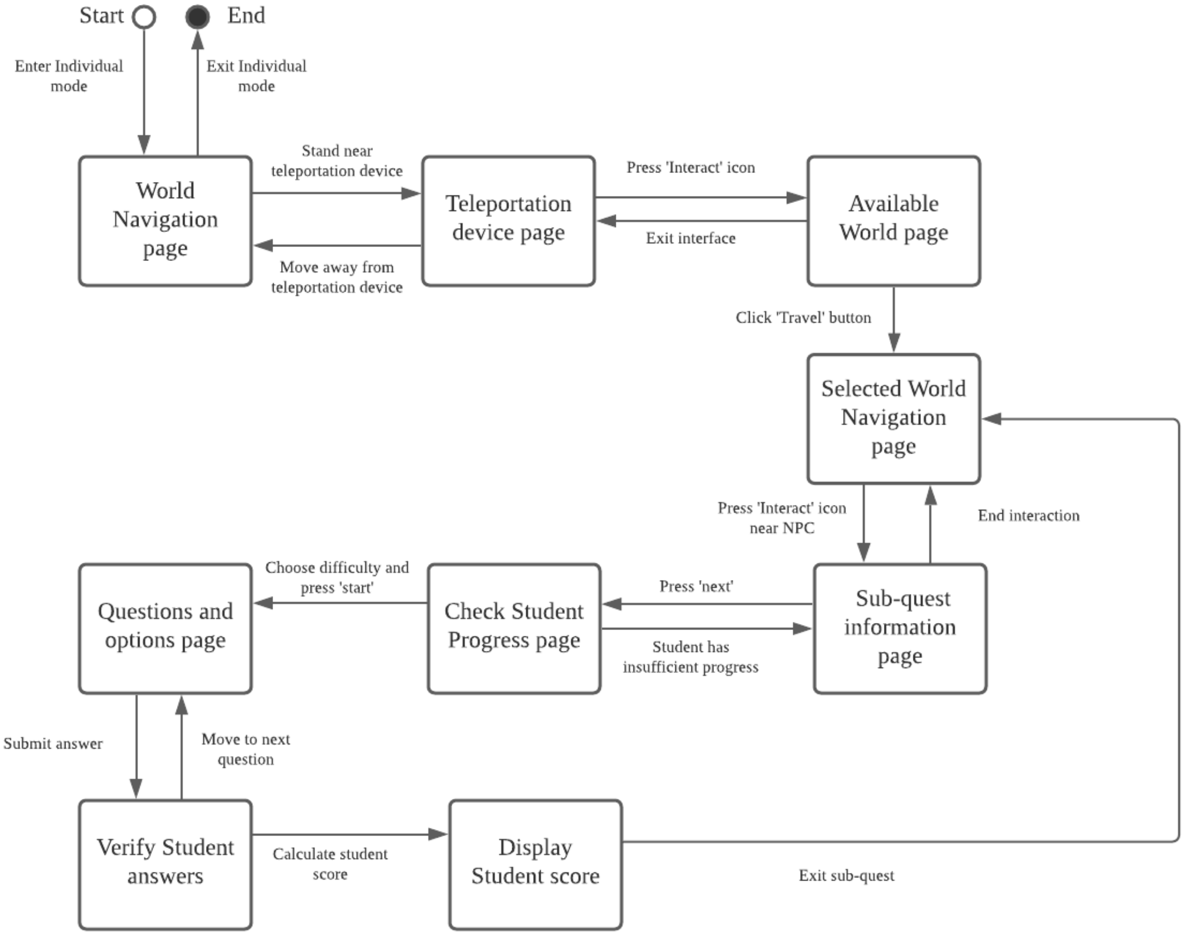
B7 Dialog map (Teacher)



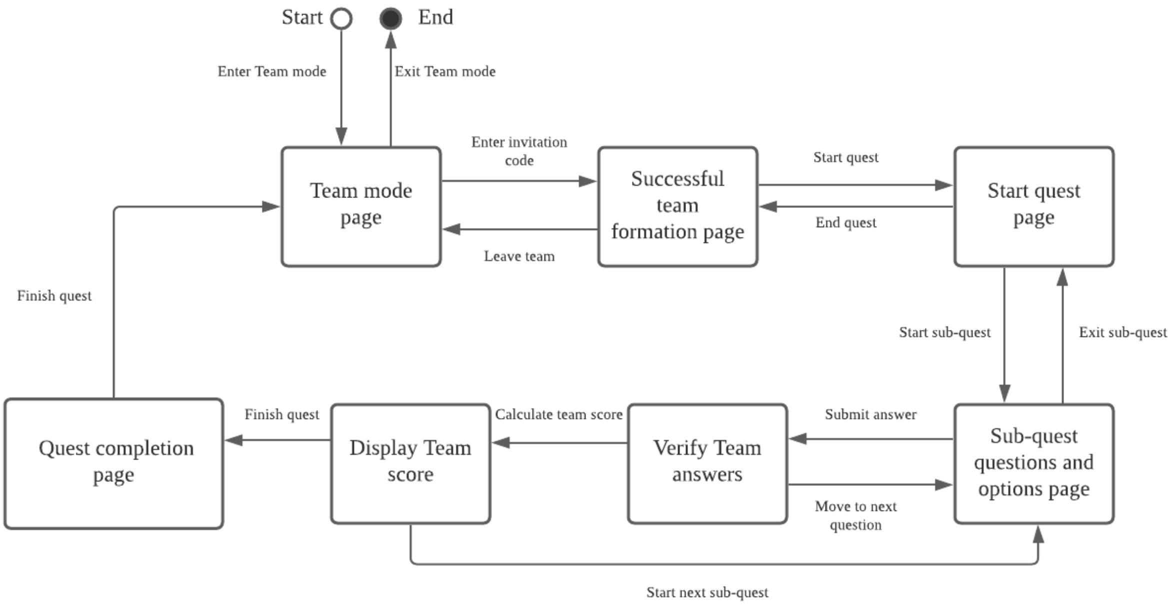
B8 Dialog map (Student)



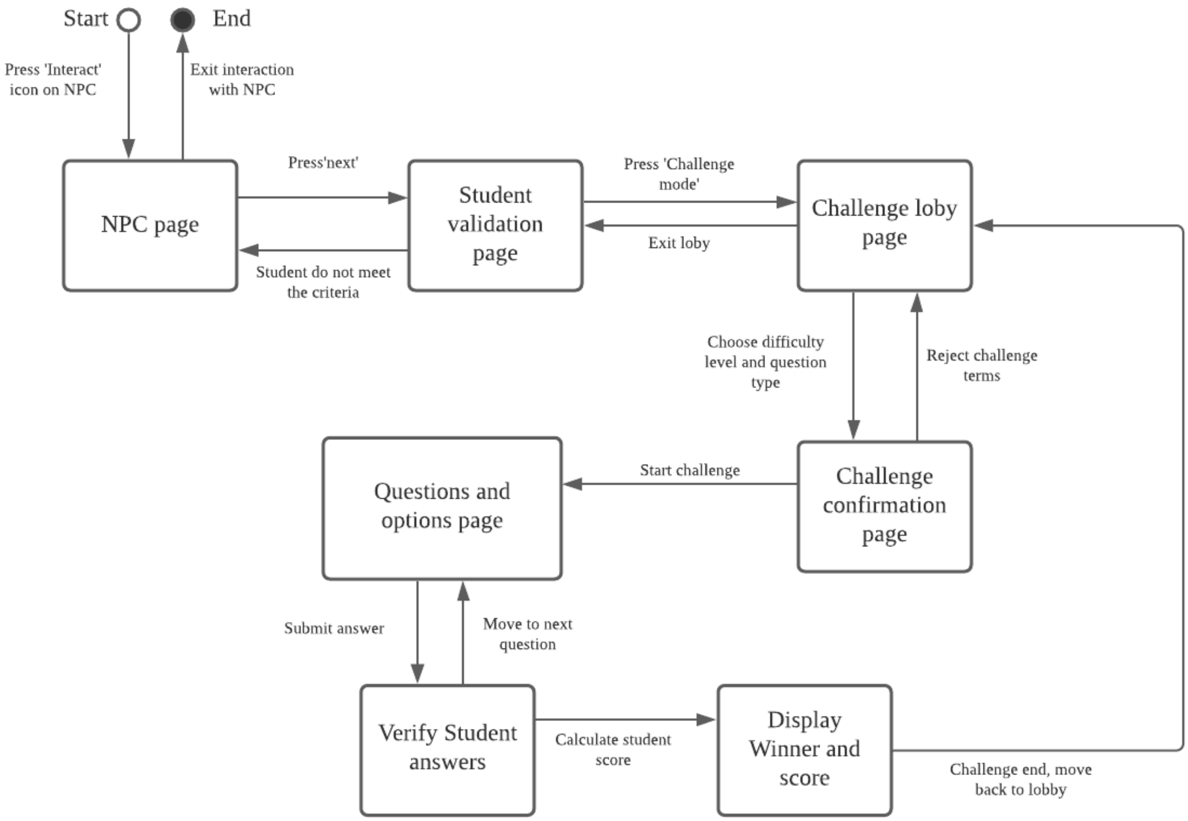
B9 Dialog map (Individual mode)


Start
End
Enter Individual mode
Exit Individual mode
World Navigation page
Stand near teleportation device
Move away from teleportation device
Teleportation device page
Press 'Interact' icon
Exit interface
Available World page
Click 'Travel' button
Selected World Navigation page
Press 'Interact' icon near NPC
End interaction
Sub-quest information page
Press 'next'
Student has insufficient progress
Check Student Progress page
Choose difficulty and press 'start'
Questions and options page
Submit answer
Move to next question
Verify Student answers
Calculate student score
Display Student score
Exit sub-quest

B10 Dialog map (Team mode)



B11 Dialog map (PVP mode)


Start
End
Press 'Interact' icon on NPC
Exit interaction with NPC
NPC page
Press'next'
Student validation page
Student do not meet the criteria
Press 'Challenge mode'
Exit loby
Challenge loby page
Choose difficulty level and question type
Reject challenge terms
Challenge confirmation page
Start challenge
Questions and options page
Submit answer
Move to next question
Verify Student answers
Calculate student score
Display Winner and score
Challenge end, move back to lobby