

---

# **Software Requirements Specification**

**for**

## **Intellect**

**Prepared By GameStop**

**Version 1.0 approved**

**February 10, 2021**

# Table of Contents

<b>Introduction</b>	<b>4</b>
<b>Purpose</b>	<b>4</b>
<b>Document Conventions</b>	<b>4</b>
<b>Intended Audience and Reading Suggestions</b>	<b>5</b>
<b>Product Scope</b>	<b>5</b>
<b>References</b>	<b>6</b>
<b>Overall Description</b>	<b>7</b>
<b>Product Perspective</b>	<b>7</b>
<b>Product Functions</b>	<b>8</b>
<b>User Classes and Characteristics</b>	<b>8</b>
<b>Operating Environment</b>	<b>9</b>
<b>Design and Implementation Constraints</b>	<b>10</b>
<b>User Documentation</b>	<b>10</b>
<b>Assumptions and Dependencies</b>	<b>11</b>
<b>External Interface Requirements</b>	<b>12</b>
<b>User Interfaces</b>	<b>12</b>
<b>Hardware Interfaces</b>	<b>30</b>
<b>Software Interfaces</b>	<b>30</b>
<b>Communications Interfaces</b>	<b>31</b>
<b>Functional Requirements</b>	<b>31</b>
<b>System Functionality to be Performed</b>	<b>31</b>
<b>Account Management</b>	<b>31</b>
<b>Use Case Diagram</b>	<b>46</b>
<b>Use Case Descriptions</b>	<b>47</b>
<b>Class Diagrams</b>	<b>92</b>
<b>Context Diagram</b>	<b>100</b>
<b>Data Flow Diagram</b>	<b>101</b>
<b>Entity Relationship Diagram</b>	<b>102</b>
<b>Decisions Table</b>	<b>103</b>
<b>Dialog Map</b>	<b>107</b>
<b>Communication Diagram</b>	<b>109</b>
<b>Component Diagram</b>	<b>111</b>
<b>State Transition Diagram</b>	<b>112</b>

<b>Crudl Matrix</b>	<b>113</b>
<b>Other Nonfunctional Requirements</b>	<b>114</b>
<b>Maintainability</b>	<b>114</b>
<b>Sub-System Interface Design</b>	<b>114</b>
<b>System Architecture</b>	<b>118</b>
<b>Testing</b>	<b>120</b>
<b>Black Box Testing for Teachers</b>	<b>120</b>
<b>White Box Testing for Teachers</b>	<b>142</b>
<b>Unit Testing for Teachers</b>	<b>150</b>
<b>Black Box Testing for Students</b>	<b>151</b>
<b>White Box Testing for Students</b>	<b>163</b>
<b>Unit Testing for Students</b>	<b>179</b>
<b>Integration Testing</b>	<b>181</b>
<b>System Testing</b>	<b>181</b>
<b>Performance Testing</b>	<b>182</b>

# **1. Introduction**

## **1.1 Purpose**

This document lays out a plan for the development of Intellect, a game application by Team GameStop. The intended readers of this document are current and future developers working on Intellect. This document will comprise, but not restrict to, a summary of the system functionality, the functional and nonfunctional requirements, use-case models, User Interface Model, Data-Flow diagrams and other analysis models.

## **1.2 Document Conventions**

Described below is the style and format used for this document:

- Section Heading:
  - Font: Times New Roman
  - Face: Bold
  - Size: 18
- Subsection Heading:
  - Font: Times New Roman
  - Face: Bold
  - Size: 14
- Main Body Content:
  - Font: Times New Roman
  - Face: Normal
  - Size: 12

To number our detailed requirements and use-case, we employ a nested numbering system where a title numbered 1 will have subheadings 1.1, 1.2 and so on.

## **1.3 Intended Audience and Reading Suggestions**

This project consists of the Intellect Game Application for the Students and a Web App for the Teachers. It is made available for the Teachers and Students enrolled in Nanyang Technological University. It aims to gamify and socialize teaching and learning of software engineering courses. This document is intended for any developer, project manager, software tester or a user involved in the design, development or testing of the application Intellect.

Users of this system include the Teachers and Students. This document serves to inform users on the main functionality of the system and give a clearer overview of the software requirements. External Interface Requirements (Section 3) and System Features (Section 4) are most useful to the users.

Developers and Project managers involved in maintaining and enhancing the system will use this document to review capabilities and understand individual components of the system. This document allows developers to determine which section of the system must be focussed on for improvements or modification. It is recommended that the developers read this document from the beginning.

For software testers (Quality Assurance Engineers), this document shall act as a guideline to create relevant test cases. It shall also serve as the starting point to understand the architecture of the entire system before conducting tests. Main areas of reading shall be System Features (Section 4) and Non-Functional Requirements (Section 5) for system testing and requirements verification.

Furthermore, those who wish to visualise the working of the application can view the interface too.

## **1.4 Product Scope**

The overall objective is to build an application that will facilitate a user to learn topics of Software Engineering through a gamified and socialized approach. The application ‘Intellect’ comprises a mobile game application and a web application.

The game application is meant for Students and aims to revive the learning interests of Students by providing a fun and interactive learning platform that they can use outside the classroom. Features such as performance tracking, experience level and total points scored shall prove to be imperative in their improvement.

The web application is to be used by Teachers/instructors to track performance of Students. They can manage and supervise the learning of their Students by giving them assignments/quizzes. It provides an efficient analytical tool that allows Teachers to understand the Students' game progress and assess their overall mastery of the course. This information can be used for adjusting teaching content and key points during classroom teaching. Lastly, it allows the Teachers to make changes to the game application if any modification of the game content is necessary.

Currently this project is being developed for Students and Teachers of the School of Computer Science at Nanyang Technological University keeping in mind the larger goal to incorporate gamified learning in schools across many universities around the globe.

## 1.5 References

- Google Firebase is used as the database system for the application –  
<https://firebase.google.com/>
- Flutter (<https://flutter.dev/>) and Dart (<https://dart.dev/>) are used for the codebase of the application.
- Visual Paradigm UML Software (<https://www.visual-paradigm.com/>) is used to generate the various diagrams.
- Javascript is used in the web application – <https://devdocs.io/javascript/>
- HTML is used in the web application – <https://devdocs.io/html/>
- CSS is used in the web application – <https://devdocs.io/css/>

## **2. Overall Description**

### **2.1 Product Perspective**

Intellect is a game that intends to gamify and socialize teaching and learning of software engineering courses and provide Teachers with an easy and convenient way to check on their Students' progress and adapt the game to each Student's needs. It consists of two separate applications, the Intellect game for the Students and web application for the Teachers.

Intellect is a game where the subject of software engineering is split into three different worlds, each representing a concept of SDLC (Software Design Life Cycle), with each world being split into several stages, further specializing on sub-topics, followed by individual levels, containing monsters that represent different questions specifically targeting a certain aspect of software engineering.

Students would have access to the mobile app, where they would be able to play the game itself. The primary features in the game are: Solo Campaign mode, Challenge mode and Assignments. Solo Campaign mode is the main feature of Intellect, with Students aiming to defeat all the monsters in a stage within the time limit. Students would answer questions relating to software engineering correctly, thus educating them on that particular topic. Challenge mode would allow Students to design a level of their own and challenge other Students to compete with themselves in the level that they created.

Teachers would have access to the web app, where they are able to observe the status of Students currently under them, with a summary report for each world, inclusive of the number of Students who have cleared the world, currently attempting to clear the world or have not tried the world. Teachers would be able to design and send Assignments to Students which they would have to complete. Teachers would, thus, have a better understanding of the strengths and weaknesses of each Student based on which question each Student consistently got wrong. The Teachers are also in charge of account management for the Students.

## **2.2 Product Functions**

The major functions are as follows:

- Students: Play the game in 3 different modes:
  - Solo Campaign Mode: A progressive game with different worlds, stages and levels
  - Challenge mode: Students design a level to compete against other Students.
  - Assignments: Clear a stage sent by the Teacher in a predefined amount of time
- Students: Leaderboard allows them to see their rank compared to other Students for each world/stage
- Students: Choose their own character from the set of unlocked characters
- Students: Share function enables them to share their scores via social media
- Teachers: Create and manage questions in Solo Campaign Mode
- Teachers: Send assignments to Students
- Teachers: Check the progress of Students by looking at statistics
  - Check the summary for each Student in different levels and worlds
  - Check summary of entire class index
  - Check summary report of each world
- Teachers: Account management for the Students

## **2.3 User Classes and Characteristics**

The 2 main User Classes for Intellect are Students and Teachers.

Students are using Intellect to learn about software engineering in a way that is more interactive and fun as compared to the traditional education methods. This is due to intrinsic motivation via the gamification of education, allowing them to learn through a method that they would find interesting and captivating. Students will be able to play the game in three different modes (Solo Campaign Mode, Arcade Mode, Challenge Mode) allowing them to understand each concept of software engineering. The game serves as a platform for Students to practice questions, complete assignments and improve their knowledge. Moreover, the social aspect of the game allows them to

see the progress that their peers have made as well as share their own progress, providing an extrinsic motivation for them to learn about the topics

Teachers are using the web application of Intellect in order to educate their Students in a more effective and convenient manner. Through the customizability and easily viewable feedback of the web application, Teachers would be able to see which topics Students are having trouble with and easily customize the questions in order to target those sections.

## 2.4 Operating Environment

- Operating Platform for game
  - Development platform: Flutter
  - Operating system platform: Android and iOS
- Operating Platform for Web Application
  - Development Platform: Javascript, HTML and CSS
  - Operating system platform: Windows and Mac OS
- Web Browser for Teachers Web Application
  - Google chrome
  - Safari
  - Microsoft Edge
- Database Server
  - Fire Base
- Social Media API for sharing assignments and challenges
  - Facebook API
  - Twitter API

## **2.5 Design and Implementation Constraints**

1. Concurrent users: The system might not be able to accommodate a huge number of concurrent users as our system needs to communicate online to our database.
2. Lack of Assets: As this is a new system, the current assets for the game application might be inadequate and lacking in the sound effects, music, monster design and so on. Additional resources to improve the design of the game needs to be outsourced or created.
3. Timing Requirements: Assignments might prove difficult, as the quizzes will have to be time-gated, and there must be a way for Teachers to notify Students about the introduction of a new, limited-time-only level.
4. Mobile-only Application: The game application can only be run on Android and iOS mobile devices. As such, users on PC will be unable to use the game application.
5. Language support: The game is only available in English and not in any other language.
6. Security: The login system for our game uses a simple authentication and might not be the most secure way of protecting Student's information against hackers. Additional security features need to be outsourced.

## **2.6 User Documentation**

User documentation components will be delivered along with the software. Listed below are the components that will be delivered:

1. Teacher Web Application user manual
2. Student Intellect Game user manual

Separate user manuals will be given to the Teacher and Students. The user manual consists of all features of Intellect as well as the step-by-step guide on the complete configuration of the required software and hardware. FAQ sections for frequently asked questions regarding the game will also be included in the user manual.

## **2.7 Assumptions and Dependencies**

### **2.7.1 Project Schedule**

Due to the tight deadline, the team must complete the application within 13 weeks. After 13 weeks, the application must be fully functional and error-free, and must satisfy all functional and non-functional requirements.

### **2.7.2 Connectivity**

Internet access is required when playing the game since Students must log into their accounts. Sharing scores and uploading any posts on social media will also require internet access.

### **2.7.3 Users' Hardware Capability**

Students should have an Android or iOS mobile phone capable of running the game.

### **2.7.4 Authentication**

Our game will require authentication in the form of user ID and passwords. Authentication is also needed as a form of OTP when a Student wants to change his or her password.

### **2.7.5 Assumptions**

1. Users are aware how to use web and mobile applications
2. Users of the mobile application know how to navigate using the touchscreen.

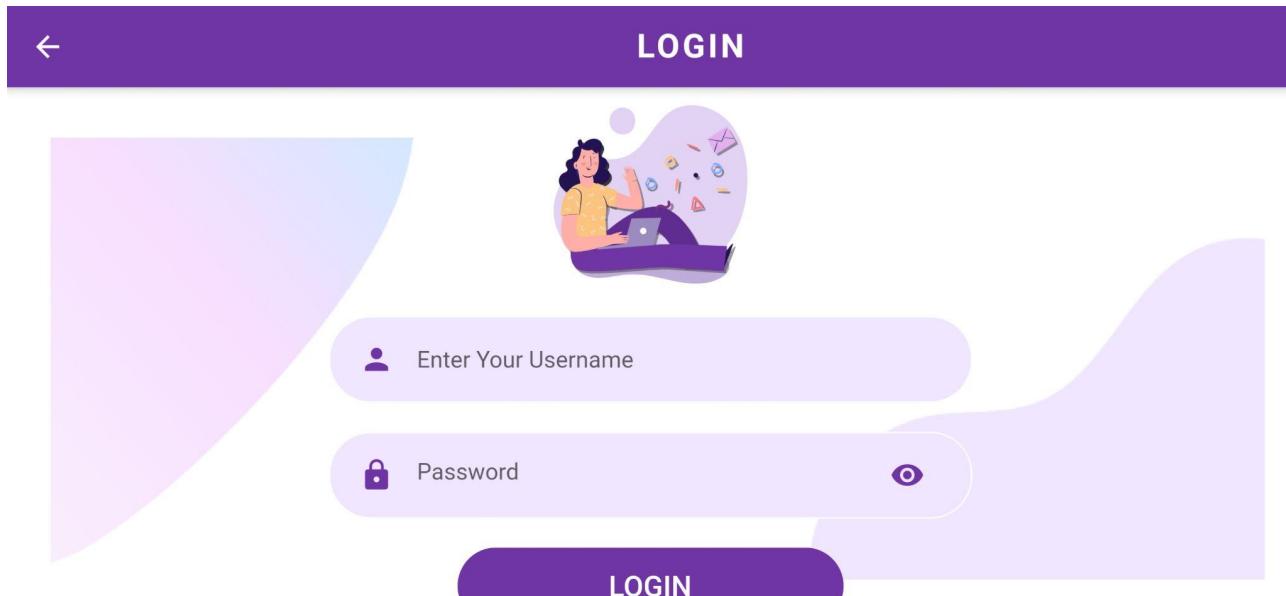
3. Users will have good internet connectivity.
4. Teachers' accounts are already registered in the database.
5. Students check their social media regularly and derive the code on their own for assignments.
6. Each assignment can have a maximum of 8 questions.
7. To use a particular world in arcade or challenge mode, Students have to first complete that world in the Solo Campaign mode. Otherwise they will not be allowed to use that world.
8. Users will check their emails and enter OTP manually.

## 3. External Interface Requirements

### 3.1 User Interfaces

#### 3.1.1 Login Page

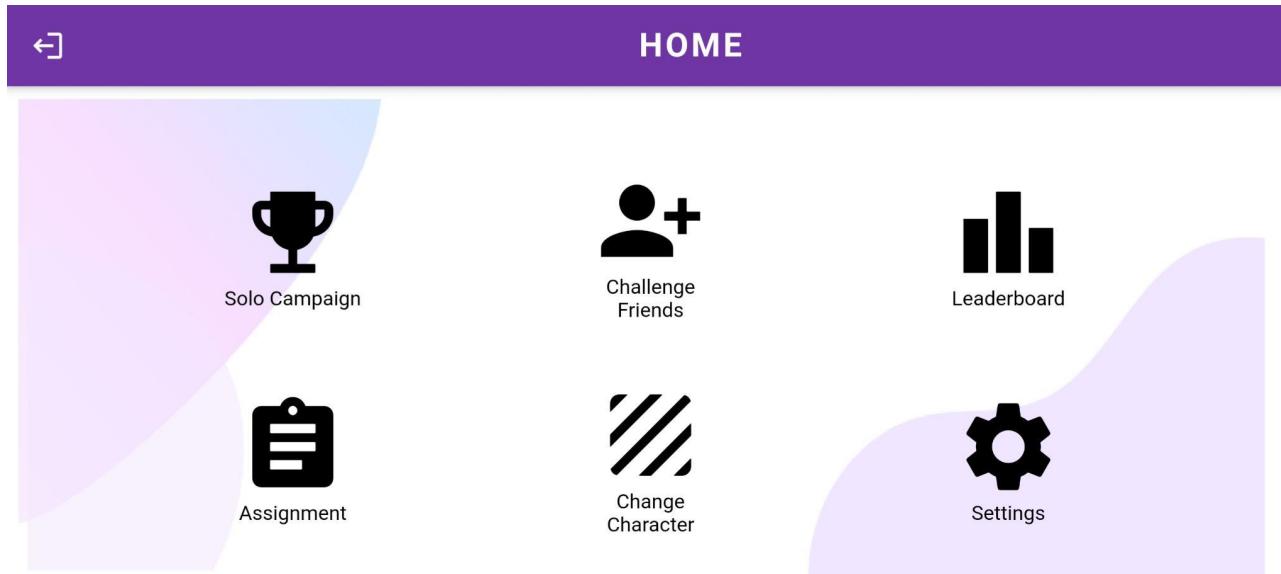
The Student must log in at the login page to enter the game.



### 3.1.2 Home Page

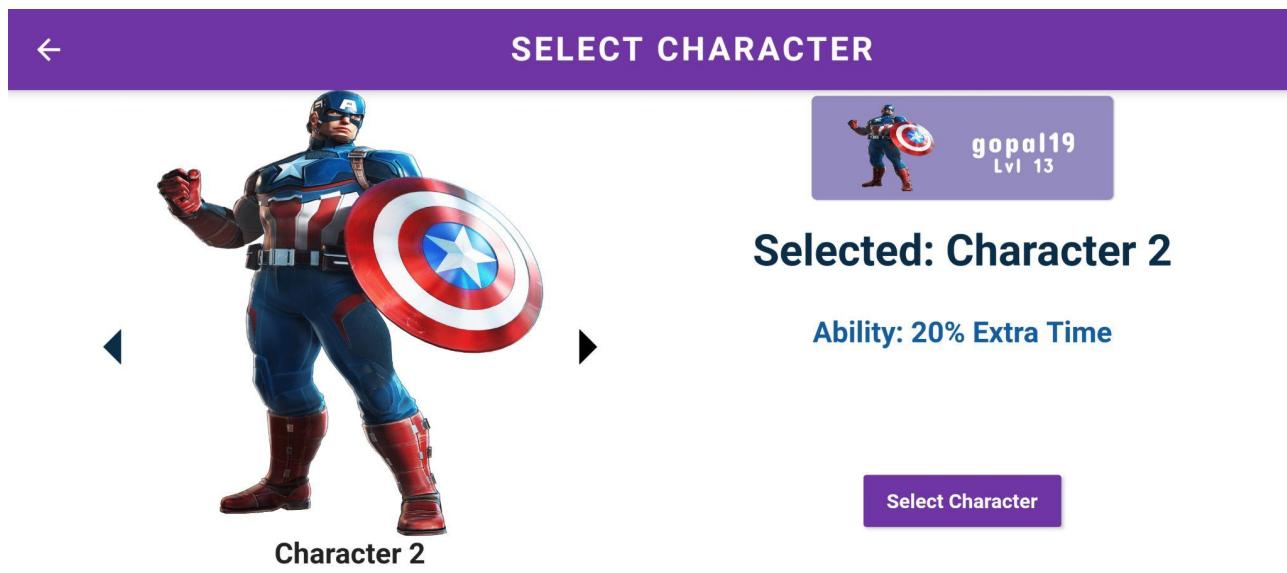
Users can access the various contents and features of the game through the main page.

- Solo Campaign Mode: Each world corresponds to a subject. Selecting an unlocked world will bring students to the stage selection page.
- Challenge Friends: Create a challenge for their peers or attempt challenges sent by their peers.
- Leaderboard: Allow students to view their ranking amongst all students or amongst their classmates.
- Assignment: Carry out the assignment stage distributed by their teacher.
- Change Character: Allow students to change their character.
- Settings: Change in game settings.



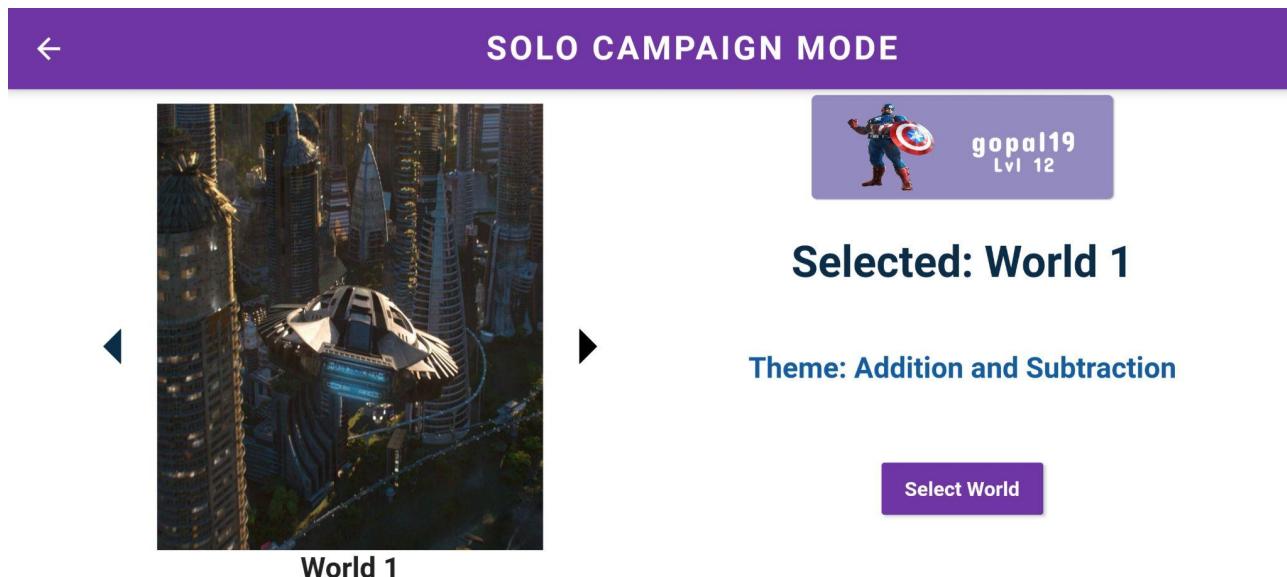
### 3.1.3 Character Selection Page

Students can select a character from among those that are unlocked. Each character has unique abilities that provide different advantages during battle.

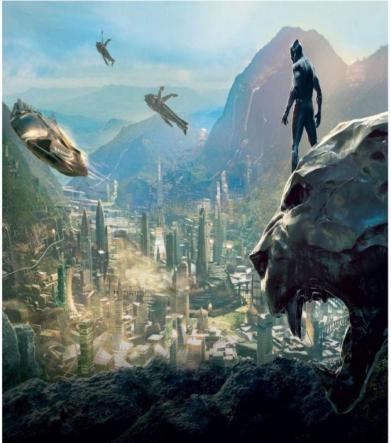


### 3.1.4 Stage Selection Page

Every world has 3 stages corresponding to a sub-topic of the subject and each stage has 3 levels of difficulty (easy, medium, hard). Students start off with only the first stage unlocked and will have to clear stages on at least easy difficulty to unlock the next stages.



[←](#) **SOLO CAMPAIGN MODE**



Stage 1



**Selected: Stage 1**

Topic: 3 Operand Equations

[Select Stage](#)

[←](#) **SOLO CAMPAIGN MODE**



Easy



Medium

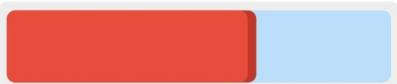


Hard

[PLAY](#)

### 3.1.5 Monster Battle

In each level, students will fight monsters to score points under a time limit. Students defeat monsters when they answer questions correctly and take damage when they answer wrongly.



60

29

9 / 10

Question:  $85 + 25 - 11?$ 

a.	101
b.	92
c.	96
d.	99



### 3.1.6 Results

After clearing a level, the final score will be displayed and students have the options to share their results on social media, retry the level, return to the main page or move on to the next level.

**RESULTS**

**LEVEL CLEARED**

Score: 60

Number of correct questions: 6 / 10

Time taken: 40 seconds

Remaining health: 20 / 100

**Share**    **Retry**    **Next**    **Exit**

### 3.1.7 Challenge Friends

Students can send a challenge to another student through Create Challenges. They can check if they have any pending challenges through Pending Challenges. They can also check their sent and received challenges records through Sent History and Received History respectively.

**Player v/s Player**

- Create Challenges
- Pending Challenges
- Sent History
- Received History

**CREATE A CHALLENGE**

Select a World	Select a Stage	Select a Level	Select a student
World 1 ▾	Stage 1 ▾	Easy ▾	glenn131 ▾

**ATTEMPT**

NOTE: You have to attempt the challenge before sending it



## PENDING CHALLENGES

Received from	World	Stage	Level	Opponent score	ATTEMPT?
ming_98	World 1	Stage 2	Easy	30	<a href="#">play</a>



## CHALLENGES RECEIVED HISTORY

Received on	Received from	World	Stage	Level	Opponent score	Your score
2021-04-14	glen131	World 2	Stage 2	Easy	70	0
2021-04-15	ming_98	World 1	Stage 2	Easy	30	0

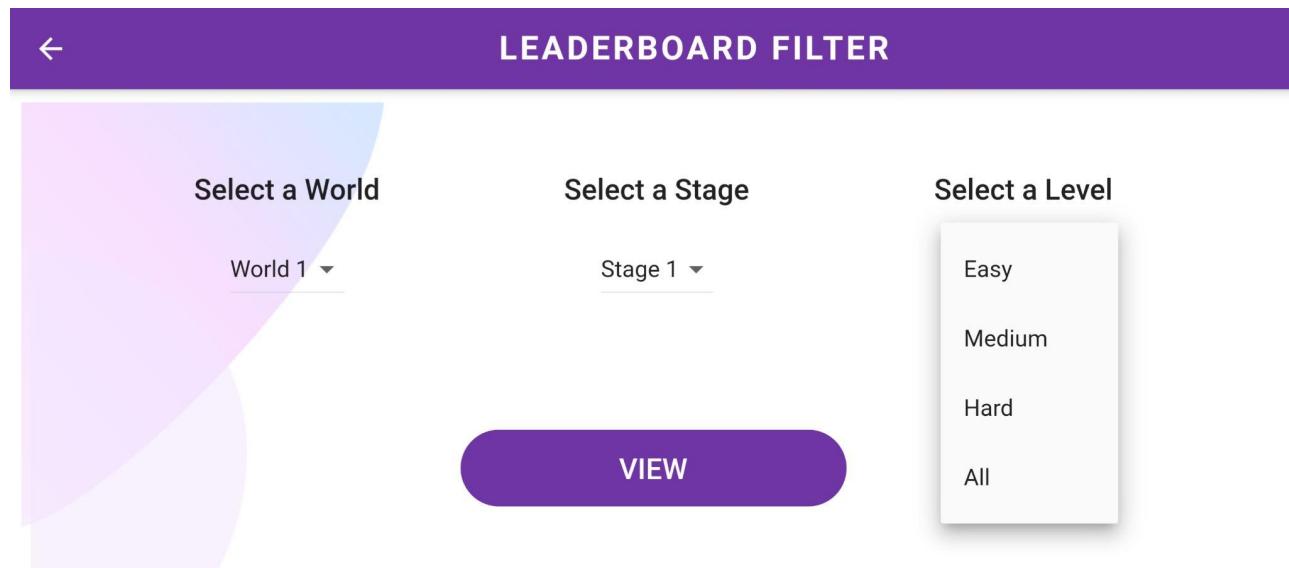


## CHALLENGES SENT HISTORY

Sent on	Sent to	World	Stage	Level	Your score	Opponent score
2021-04-15	glenn131	World 1	Stage 1	Easy	60	0

### 3.1.8 Leaderboard

Students can check their ranking among their friends according to the selected world, stage and level.

The image shows the 'LEADERBOARD' screen. At the top, there is a purple header bar with a back arrow icon and the text 'LEADERBOARD'. Below the header, there is a table with the following data:

Rank	Username	Score
1	suhana01	130
2	remus_14	130
3	glenn131	130
4	khush_156	80
5	gopal19	80
6	guatkwan_21	70

**gopal19**  
**Rank: 5**  
**Score: 80**

### 3.1.9 Assignments

Students can check their current assignments as well as records of past assignments under Pending Assignments and Past Assignments.

The image displays two screenshots of a mobile application interface. The top screenshot shows a purple header bar with a back arrow icon and the text "ASSIGNMENTS". Below the header is a large white area containing two rounded rectangular buttons: "Pending Assignments" (purple) and "Past Assignments" (light blue). The bottom screenshot shows a purple header bar with a back arrow icon and the text "PAST ASSIGNMENT". Below the header is a table with five columns: "Code", "Title", "World", "Result", and "Score". The table contains two rows of data:

Code	Title	World	Result	Score
CZ3003	SSAD	World 1	FAIL	N/A
CZ2002	OODP	World 1	PASS	70



## PENDING ASSIGNMENTS

Code	Title	World	Due on :	ATTEMPT?
CZ3003	SSAD	World 1	17-Apr-2021	<a href="#">play</a>
CZ2001	Algorithms	World 1	17-Apr-2021	<a href="#">play</a>
HE9092	Economic Theory	World 1	17-Apr-2021	<a href="#">play</a>
CZ2003	CGV	World 2	17-Apr-2021	<a href="#">play</a>
CZ2004	HCI	World 2	18-Apr-2021	<a href="#">play</a>

### 3.1.10 Settings

Students can turn off background music, change password or logout in the settings screen.



## SETTINGS



Background Music

[Change Password](#)

[Logout](#)



## FORGOT PASSWORD

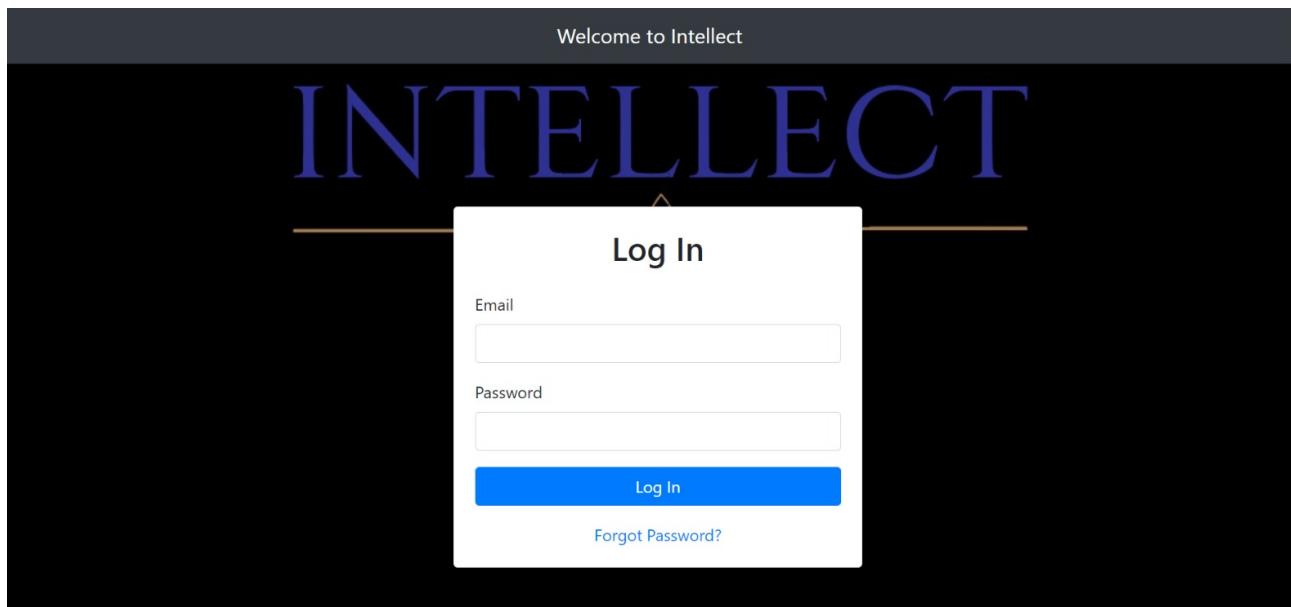


Your Email

Request OTP

### **3.1.11 Teacher Login Page**

The teacher must log in at the login page to enter the web application.



### **3.1.12 Teacher Main Page**

Teachers can access the various features of the application through the main page.

- Add New Index: Add a new class index.
- Send New Assignment: Allow teachers to create and send assignments to students and share on social media.
- Manage Students: Allows teachers to create or delete students' accounts.
- View Progress: Allow teachers to check on the progress of the class or the sent assignments.
- Edit Question Bank: Allow teachers to add or delete questions in the database of the game.

# Welcome, Teacher!

What do you want to do today?

Log Out

### 3.1.13 Add New Student Page

Teachers are able to enter the username, first name, last name and the class index of the student to create a new student account. Default passwords will be assigned to new students upon the creation of their accounts.

## Add New Student

Enter Full name

Enter email address

Enter username

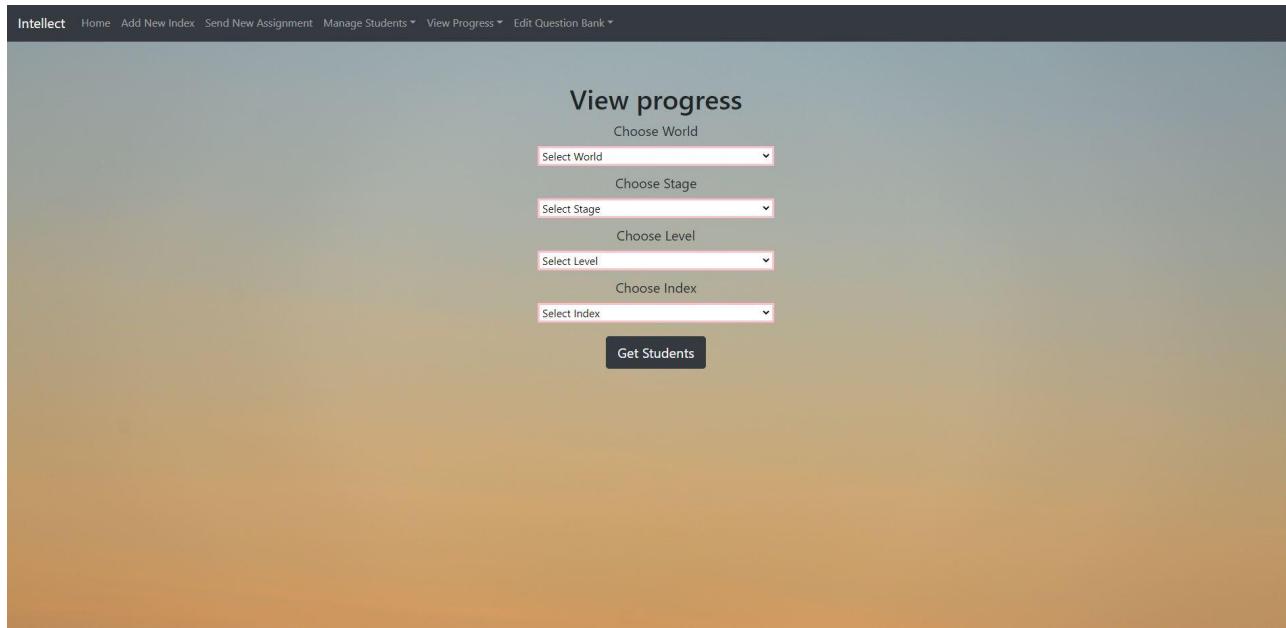
## Choose Class Index

Select Index ▾

Submit

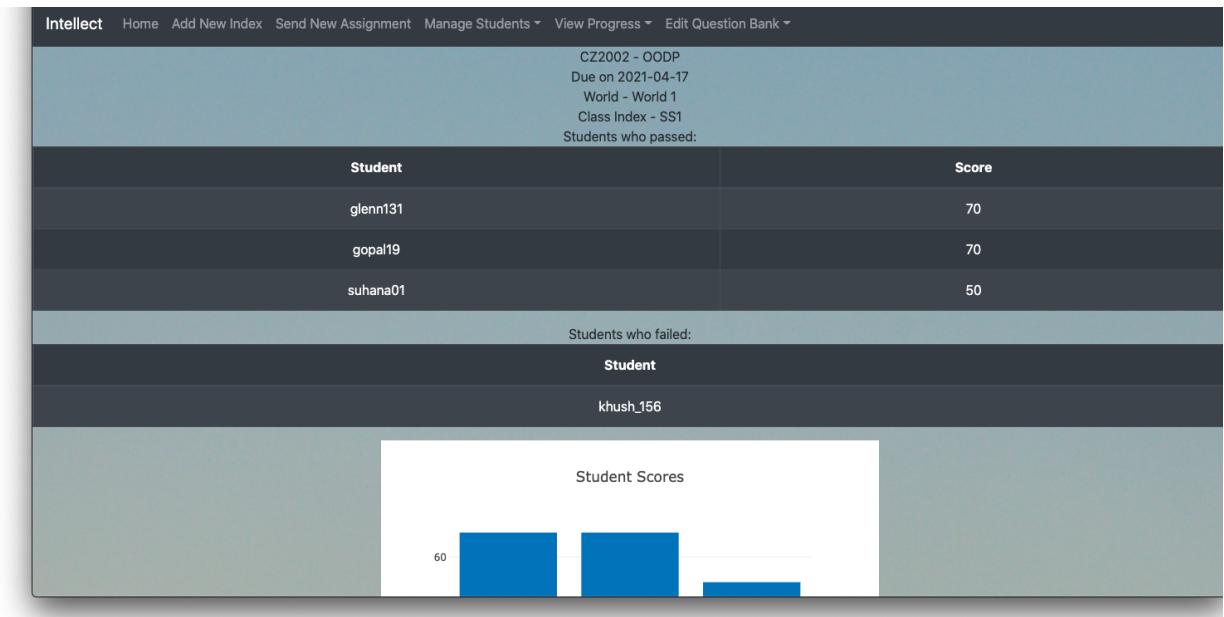
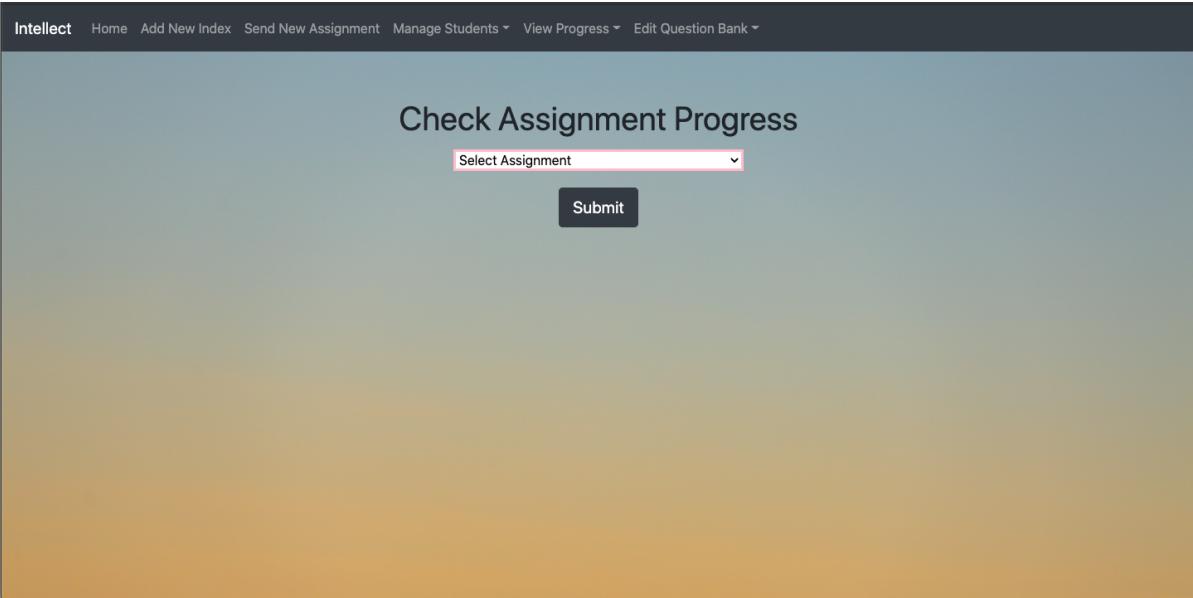
### **3.1.14 View Class Progress**

Teachers are able to check the progress of the class index as well as individual students. The system will display all relevant progresses of individual student and class indexes.



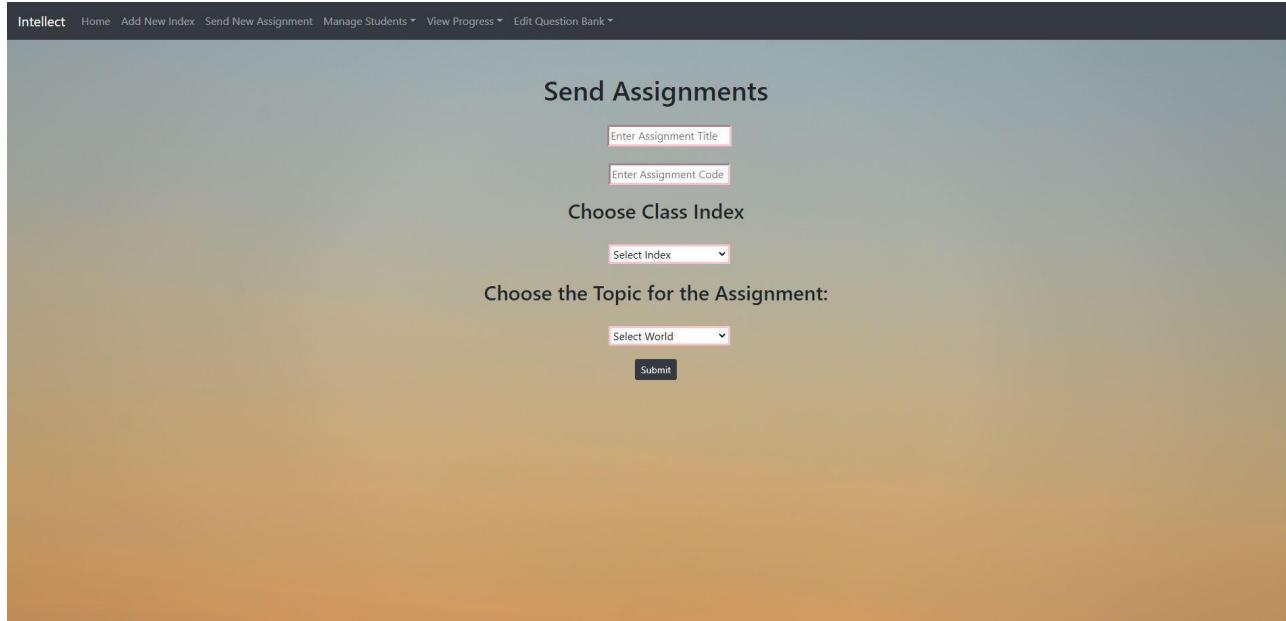
### **3.1.15 View Assignment Progress**

Teachers can track the relevant statistics of the assignment assigned to the students. For example, teachers can see the highest score obtained for the assignment and the number of students who completed or failed the assignment.



### 3.1.16 Send Assignment

After creating an assignment, teachers can enter an assignment code together with the title of the assignment. They will then be able to share on the social media such that students can retrieve the assignment code and attempt the assignment in the game.



### 3.1.17 Add Question

Teachers can add questions to the database of the game. To add a question, teachers will have to first select the world, the stage, followed by the difficulty level. Afterwhich, they will input the question and the options.

Intellect Home Add New Index Send New Assignment Manage Students View Progress Edit Question Bank

### Add Questions

Choose World  
Select World ▾ Choose Stage  
Select Stage ▾ Choose Level  
Select Level ▾ Question  
Enter the question

Answer  
Enter the answer

Wrong Option 1  
Enter wrong option 1

Wrong Option 2  
Enter wrong option 2

Wrong Option 3  
Enter wrong option 3

Add Question

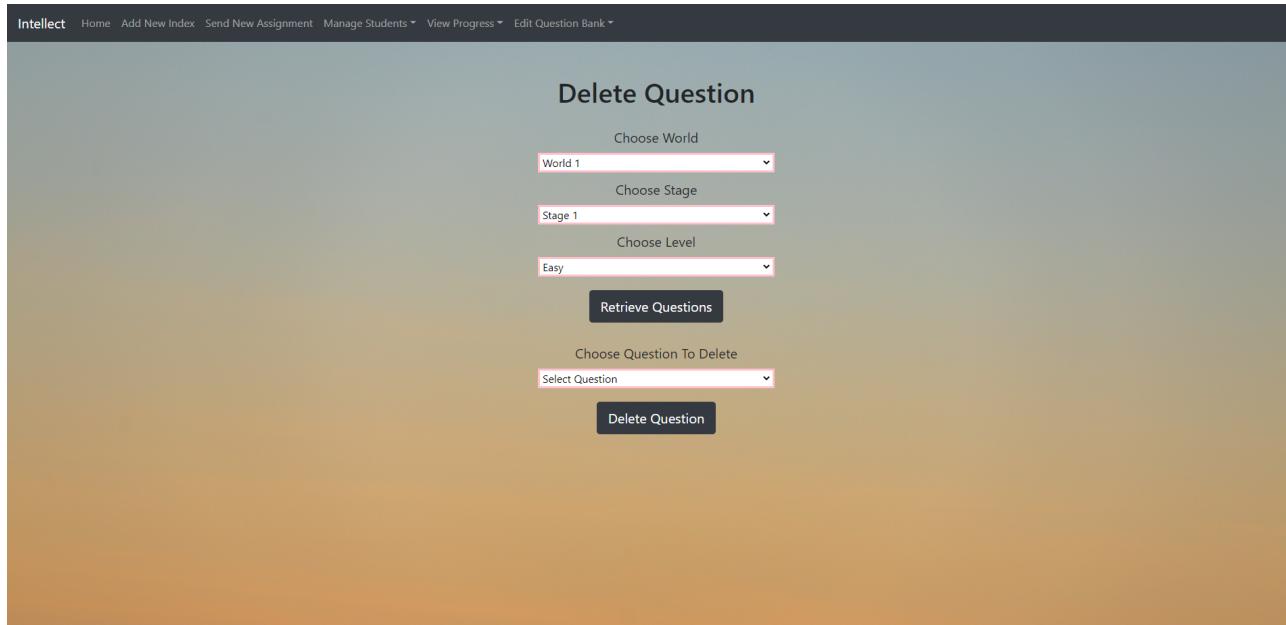
### 3.1.18 Delete Question

Teachers can delete questions to the database of the game. To delete a question, teachers will have to first select the world, the stage, followed by the difficulty level. Afterwhich, they can retrieve the questions and choose the one to delete.

Intellect Home Add New Index Send New Assignment Manage Students View Progress Edit Question Bank

### Delete Question

Choose World  
Select World ▾ Choose Stage  
Select Stage ▾ Choose Level  
Select Level ▾ Retrieve Questions



## 3.2 Hardware Interfaces

### 3.2.1 Game Application

The game application requires a mobile device that supports Android or iOS with Internet connection capability.

### 3.2.2 Teacher Web Application

The web application requires any hardware that can run major browsers such as Google Chrome, Mozilla Firefox and Microsoft Edge.

## 3.3 Software Interfaces

### 3.3.1 Intellect Game Application

- The game application runs on an Android or iOS mobile device.
- The platform will be collecting the results of the assignments and campaigns and uploading them to the database.

### **3.3.2 Teachers Web Application**

- The web interface is required to run on all frequently used browsers such as Google Chrome, Mozilla Firefox, Microsoft Edge.
- The platform will be handling the major features of the application including: account management, assignment management, editing of questions in database, viewing statistics, etc.

## **3.4 Communications Interfaces**

### **HTTPS Communication**

- HTTP protocol encrypted by TLS which provides secure communication over the network and authentication of the website as well as the associated web server the client is communicating with thus helping to guard against the man-in-the-middle attack.
- In addition, it also provides bidirectional encryption of communications between the client and server and helps guard against any eavesdropping attempt or tampering of data.

## **4. Functional Requirements**

### **4.1 System Functionality to be Performed**

#### **1. Account Management**

- 1.1. The game must have two types of accounts: Student and Teacher.
  - 1.1.1. The Student account must be used to login into the game application.
  - 1.1.2. The Teacher account must be used to login into the Teacher's webapp.
  - 1.1.3. The webapp must allow the Teachers to manage the accounts of the Students.
  - 1.1.4. The game application must not allow the Students to create accounts.

## **Game Application (Intellect)**

### **2. Student Login**

2.1. The system must allow the Student to login into the game.

2.1.1. The system must prompt the user for his/her username and password.

2.1.1.1. The system must validate that all the text fields are filled in.

2.1.1.2. The system must validate that the username entered exists.

2.1.1.3. The system must validate that the password entered is valid.

2.1.1.4. The system must display the corresponding error message when any of the above mentioned validation fails.

2.1.2. The system must check if the Student is logging in for the first time.

2.1.2.1. The system must prompt the Student to change the default password to that of his/her choice.

2.1.2.1.1. The system must prompt the user for a new password and to confirm the new password.

2.1.2.1.1.1. The system must validate that the new password entered matches the confirmed new password text field.

2.1.3. The system must provide the Student with a ‘Forget Password?’ option.

2.1.3.1. The system must prompt the user to enter the registered email address.

2.1.3.1.1. The system must validate that the email address is of appropriate format and exists.

2.1.3.2. The system must send an OTP to the entered email address.

2.1.3.3. The system must prompt the user to enter the OTP sent.

2.1.3.3.1. The system must validate that the OTP entered by the user matches the OTP sent to the email address.

2.1.3.3.1.1. The system must give the user three tries to enter the OTP.

2.1.3.3.1.2. The system must redirect to the initial page after the user has made three incorrect attempts.

2.1.3.3.2. The system must prompt the user for the new password and confirm the new password.

2.1.3.3.2.1. The system must validate that the entered password matches the confirmation password.

2.1.3.4. The system must save the new password into the database.

2.1.4. The system must redirect to the Home Screen upon successful login.

### **3. Home Screen**

3.1. The system must display the following options on the home screen:

3.1.1. Solo campaign

3.1.2. Leaderboard

3.1.3. Assignment

3.1.4. Player v/s Player

3.1.5. Change Character

3.1.6. Settings

3.2. The system must allow the user to navigate to the respective screens.

3.3. The system must have a signout button.

3.3.1. When pressed, the system must log the user out and redirect the user to the Login Screen.

### **4. Solo Campaign Mode**

4.1. The Student's main page must display the 'Solo Campaign' mode.

4.2. This mode consists of 3 worlds.

4.3. Each world represents questions of increasing difficulty in Mathematics.

4.4. The system must allow the Student to view the description for this mode.

4.5. The system must allow the player to view the locked and unlocked worlds.

4.6. The system must allow the player to choose a world from the list of unlocked worlds.

4.7. The system must not allow the player to choose a world which is locked.

4.8. The system must redirect the player to the stages upon choosing to play this mode.

4.8.1. Each world has 3 stages which have their own respective themes.

4.8.1.1. Each stage has three levels.

4.8.1.2. The first level is of difficulty ‘Easy.’

- 4.8.1.2.1. The system must allow the monster cards to attack the player’s character with questions from the ‘Easy’ dataset.
- 4.8.1.2.2. The player must defeat the monsters by answering the questions correctly.
- 4.8.1.2.3. Each player has 100 HP to clear the level.
- 4.8.1.2.4. The player must defeat 5 out of 10 monsters within the allotted time to proceed to the next level.
- 4.8.1.2.5. The system must award 10 points for each correct answer to the player.
- 4.8.1.2.6. The system must award a bonus of 30 points on getting all questions correct.
- 4.8.1.2.7. The system must increase the experience level of the player if he/she clears that particular level.
- 4.8.1.2.8. The player loses 20 HP for every incorrect question in this level.
- 4.8.1.2.9. The system must stop the game when the player has run out of HP.
- 4.8.1.2.10. The system must stop the game when the time allotted for the level is over.

4.8.1.3. The second level is of difficulty ‘Medium.’

- 4.8.1.3.1. The system must allow the monster cards to attack the player’s character with questions from the ‘Medium’ dataset.
- 4.8.1.3.2. The player must defeat the monsters by answering the questions correctly.
- 4.8.1.3.3. Each player has 100 HP to clear the level.
- 4.8.1.3.4. The player must defeat 5 out of 10 monsters in the allotted time to proceed to the next level.
- 4.8.1.3.5. The system must award 10 points for each correct answer to the player.

- 4.8.1.3.6. The system must award a bonus of 30 points getting all questions correct.
  - 4.8.1.3.7. The system must increase the experience level of the player if he/she clears that particular level.
  - 4.8.1.3.8. The player loses 20 HP for every incorrect question in this level.
  - 4.8.1.3.9. The system must stop the game when the player has run out of HP.
  - 4.8.1.3.10. The system must stop the game when the time allotted for the level is over.
- 4.8.1.4. The third level is of difficulty ‘Hard.’
- 4.8.1.4.1. The system must allow the monster cards to attack the player’s character with questions from the ‘Hard’ dataset.
  - 4.8.1.4.2. The player must defeat the monsters by answering the questions correctly.
  - 4.8.1.4.3. Each player has 100 HP to clear the level.
  - 4.8.1.4.4. The player must defeat 5 out of 10 monsters within allotted time to proceed to the next level.
  - 4.8.1.4.5. The system must award 10 points for each correct answer to the player.
  - 4.8.1.4.6. The system must award a bonus of 30 points on getting all questions correct.
  - 4.8.1.4.7. The system must increase the experience level of the player if he/she clears that particular level.
  - 4.8.1.4.8. The player loses 20 HP for every incorrect question in this level.
  - 4.8.1.4.9. The system must stop the game when the player has run out of HP.
  - 4.8.1.4.10. The system must stop the game when the time allotted for the level is over.

- 4.8.1.5. The player is allowed to retry each level.
- 4.8.1.6. The system must always store the highest score achieved by the player.
- 4.8.1.7. The system unlocks new stages after all three levels in the previous stage are cleared by the player.
- 4.8.1.8. The new worlds are unlocked once the previous world's stages have been entirely completed.
- 4.8.1.9. The system must keep track of every Student's progress in this mode.

## 5. Character Screen

- 5.1. The system must display the characters present in the game.
  - 5.1.1. The system must choose the first character by default for the Student logging in for the first time.
  - 5.1.2. The system must choose the first character by default for the Student who has not unlocked any other other character.
  - 5.1.3. The system must allow the Student to view the locked as well as unlocked characters with their special abilities.
    - 5.1.3.1. The system must allow the Student to choose a character of his/her choice from the unlocked characters.
    - 5.1.3.2. The system must display the locked characters to the Student.
      - 5.1.3.2.1. The system must display the experience level needed to unlock those characters.
      - 5.1.3.2.2. The system must not allow the Student to choose locked characters as their current character.
  - 5.1.4. The system must update the database when the user changes their chosen character.

## 6. Settings

- 6.1. The system must display the profile of the Student in the 'Settings' page.
- 6.2. The system must display the account details of the Student except his/her password.

- 6.3. The system must display the Student's experience level in the game.
- 6.4. The system must allow a Student to edit his profile.
  - 6.4.1. The system must allow the Student to turn on/off the game's background music.
  - 6.4.2. The system must allow the Student to reset their password.
    - 6.4.2.1. The system must prompt the Student for the old password.
      - 6.4.2.1.1. The system must validate that the old password does not match with the one entered.
    - 6.4.2.2. The system must prompt the Student for the new password.
    - 6.4.2.3. The system must prompt the Student to confirm the new password.
      - 6.4.2.3.1. The system must validate that both the text fields have the same value.
  - 6.4.3. The system must save the new password in the database.

## 7. Leaderboard

- 7.1. The system must allow the Student to choose the type of Leaderboard that he/she wants to view.
- 7.2. The system must prompt the Student to choose the options for world, stage and level.
  - 7.2.1. The system must prompt the Student to choose a world from the list of worlds that he/she has unlocked.
  - 7.2.2. The system must prompt the Student to choose 'All' or a single stage from the list of stages that he/she has unlocked.
  - 7.2.3. The system must display 'All' for levels if the Student has chosen 'All' for stages.
  - 7.2.4. The system must prompt the Student to choose 'All' or a single level from the list of levels that he/she has unlocked if the Student had chosen a single stage before.
- 7.3. The system must display the leaderboard according to the filters chosen by the Student.

## **8. Challenge Mode**

8.1. The system must allow the user to access one of the following option:

- 8.1.1. Create A New Challenge.
- 8.1.2. View Pending Challenges.
- 8.1.3. History of Challenges Sent.
- 8.1.4. History of Challenges Received.

8.2. Create A New Challenge.

- 8.2.1. The system must allow the Student to choose one of the Students in his/her class index to send the challenge.
- 8.2.2. The system must allow the Student to select one of the worlds, stages and levels which have been unlocked by him/her in the Solo Campaign Mode
- 8.2.3. The Student must first complete the challenge themselves.
- 8.2.4. Once their score is recorded the challenge is sent to the Student of their choice.

8.2.4.1. The system must allow the user to share their result on social media

- 8.2.5. The challenged Student must complete the challenge within one day or else it will expire.

8.3. View Pending Challenges.

8.3.1. The system must display a table containing their pending challenges.

8.3.1.1. This table must display the opponent, opponent score and the difficulty level of the challenge.

8.3.2. The system must allow the student to play any challenge which is pending.

8.3.3. The system must display to the user the result of the challenge.

8.3.3.1. The system must allow the user to share their result on social media.

8.3.4. The system must remove this challenge from the pending challenges screen as it has now been completed.

8.4. History of Challenges Sent.

8.4.1. The system must display a table of the challenges sent by this user.

8.4.1.1. The system must display the opponent, user's score, opponent score and the result of the challenge.

**8.5. History of Challenges Received.**

8.5.1. The system must display a table of the challenges received by this user.

8.5.1.1. The system must display the opponent, and scores of both the user and the opponent alongside the result.

8.5.1.2. The system must not give an option to attempt these challenges as they are already attempted or expired.

**9. Assignments**

9.1. The additional assignments added to the game by the Teachers can be seen in the ‘Assignment’ page.

9.2. The system allows the Student to view the pending assignments and the past assignments.

9.3. The system must display the table of pending assignments with assignment code, title, theme, due date and option to attempt.

9.3.1. Each assignment has a deadline of 3 days from the day it was added by the Teacher.

9.3.2. Each student can attempt the assignment only once.

9.3.3. The system must remove the assignment from the pending assignments table once attempted or expired.

9.4. The system must display the table of all past assignments attempted by the Student in the ‘Past Assignments’ page.

9.4.1. The system must display a table of the past assignments attempted with the assignment name, theme, title, code and score if the Student passed the assignment.

9.4.2. The system must now allow the Student to attempt these assignments again.

**Web Application For Teachers**

**10. Teacher’s Login**

10.1. The system must allow the Teacher to login into the home page.

10.2. The system must allow the Teacher to enter his/her email address and password.

- 10.2.1. The system must validate that the text fields are filled in.
  - 10.2.2. The system must validate that the email address entered is valid
  - 10.2.3. The system must validate that the email address entered is registered in the database.
  - 10.2.4. The system must validate that the password entered is valid.
  - 10.2.5. The system must display the corresponding error message
- 10.3. The system must display the ‘Forgot Password?’ option to the Teacher in the login page.
- 10.3.1. The system must prompt the Teacher to enter his/her registered email address.
    - 10.3.1.1. The system must validate that the email address entered is valid.
    - 10.3.1.2. The system must validate that the email address entered exists.
  - 10.3.2. The system must send an OTP to the entered email address.
  - 10.3.3. The system must prompt the user to enter the OTP sent.
    - 10.3.3.1. The system must validate that the OTP entered by the user matches the OTP sent to the email address.
  - 10.3.4. The system must prompt the user for the new password and confirm the new password.
    - 10.3.4.1. The system must validate that the entered password matches the confirmation password.
  - 10.3.5. The system must save the new password into the database.
- 10.4. The system must redirect the Teacher to the home page upon successful login.

## **11. Home Page**

- 11.1. The Teacher’s home page consists of the following components: ‘Manage Students’, ‘View Progress’, ‘Send Assignment’, ‘Edit Question Bank’ and ‘Add New Index.’

## **12. Manage Students**

- 12.1. The Teacher can choose to ‘Create New Student’ or ‘Delete Student’ in this section.
- 12.1.1. The Teacher can create a Student account for the game on the ‘Add New Student’ page.

- 12.1.1.1. The system must allow the Teacher to enter the Student's username.
  - 12.1.1.1.1. The system must validate that the username entered does not already exist.
  - 12.1.1.2. The system must allow the Teacher to enter the Student's full name.
  - 12.1.1.3. The system must allow the Teacher to enter the Student's email address.
    - 12.1.1.3.1. The system must validate that the email address entered is valid.
  - 12.1.1.4. The system must allow the Teacher to choose the Student's class index from the dropdown menu.
  - 12.1.1.5. The system must validate that all the text fields are filled in.
  - 12.1.1.6. The system must display the corresponding error message when any of the above validation fails.
  - 12.1.1.7. The system must assign the default password to each Student.
  - 12.1.1.8. The system must notify the Student through email upon successful account creation.
  - 12.1.1.9. The system must redirect the Teacher back to the same page.
- 12.1.2. The Teacher can delete a Student account in the 'Delete Student' page.
  - 12.1.2.1. The system must prompt the Teacher for the Student's username.
  - 12.1.2.2. The system must allow the Teacher to delete the account.
  - 12.1.2.3. The system must check if the username entered is valid.
    - 12.1.2.3.1. The system must display an appropriate message if the username entered is invalid.
  - 12.1.2.4. The system must notify the Student through email on successful termination of his/her account.
  - 12.1.2.5. The system must redirect the Teacher back to the same page.

### **13. View Progress**

13.1. The system must allow the Teacher to view ‘Class Progress’ or ‘Assignment Progress.’

13.1.1. The system must allow the Teacher to check on the progress of the class as well as Students in the Class Progress’ screen.

13.1.1.1. The system must prompt the Teacher to choose a world from the dropdown menu.

13.1.1.2. The system must prompt the Teacher to choose a single stage or choose ‘All’ from the dropdown menu if he/she wishes to see data for all stages.

13.1.1.3. The system must prompt the Teacher to choose a single level or choose ‘All’ from the dropdown menu if he/she wishes to see data for all stages.

13.1.1.4. The system must prompt the Teacher to choose a class index for which he/she wishes to view the progress.

13.1.1.5. The system must validate that all the fields are filled in.

13.1.1.5.1. The system must display the corresponding error message if any field is empty.

13.1.1.6. The system must display a list of all the Students in the index chosen by the Teacher.

13.1.1.7. The system must prompt the Teacher to choose a single Student or choose “All” from the dropdown menu.

13.1.1.8. The system must validate that the Teacher has chosen a value.

13.1.1.8.1. The system must display an appropriate error message if the field is empty.

13.1.1.9. The system must check if the data exists for a particular student for the world, stage, level chosen by the Teacher.

13.1.1.10. The system must display an appropriate message if there is no data for the filters chosen by the Teacher.

13.1.1.11. The system must display the table of score, attempts, correct questions and time taken according to the values chosen previously.

- 13.1.1.12. The system must display different visual analytics in the form of graphs and pie charts to the Teacher.
- 13.1.2. The system must allow the Teacher to view the assignment progress in the ‘Assignment Progress’ Page.
  - 13.1.2.1. The system must prompt the Teacher to choose the assignment for which he/she wants to see the progress.
  - 13.1.2.2. The system must validate that all the fields are filled in.
  - 13.1.2.3. The system must display an appropriate error message if the fields are not filled in.
  - 13.1.2.4. The system must redirect the Teacher to the progress screen on clicking the View Progress button.
  - 13.1.2.5. The system must display the details of the assignment chosen.
  - 13.1.2.6. The system must display the table of Students who cleared the assignment along with their scores.
  - 13.1.2.7. The system must display the Students who failed the assignment.
  - 13.1.2.8. The system must display visual analytics in the form of graphs to the Teacher.

## **14. Send Assignment**

- 14.1. The system must allow the Teacher to create and send assignments on social media in the ‘Send Assignment’ page.
- 14.2. The system must allow the Teacher to create assignments.
  - 14.2.1. The system must prompt the Teacher to choose the theme of the assignment from the dropdown menu.
  - 14.2.2. The system must prompt the Teacher to enter the title of the assignment.
  - 14.2.3. The system must prompt the Teacher to enter the code of the assignment.
  - 14.2.4. The system must prompt the Teacher to choose the class index for which the assignment is meant for from the dropdown menu.
  - 14.2.5. The system must validate that all the fields are correctly filled.

- 14.2.5.1. The system must display the corresponding error message if any of the fields are not filled.
- 14.3. The system must allow the Teacher to share the assignment code on social media.
  - 14.3.1. The system must allow the Teacher to share the code on facebook.
  - 14.3.2. The system must allow the Teacher to share the code on twitter.
- 14.4. The system must notify the Students in the chosen class index by sending an email to their respective email id.

## **15. Edit Question Bank**

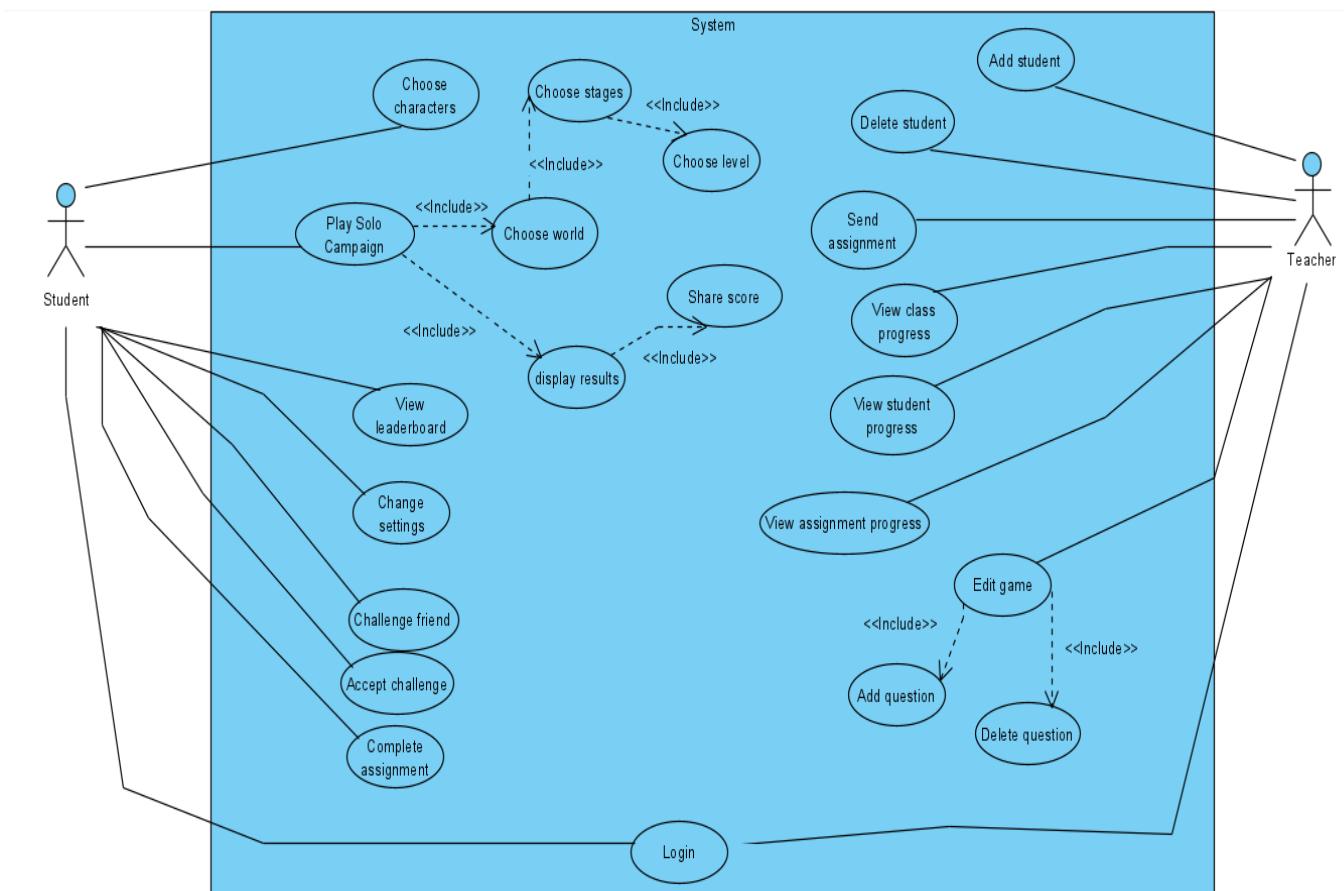
- 15.1. The Teacher can edit the questions in the database in the ‘Edit Question Bank’ page.
- 15.2. The system must allow the Teacher to add a question.
  - 15.2.1. The system must allow the Teacher to choose a world.
  - 15.2.2. The system must allow the Teacher to choose a stage.
  - 15.2.3. The system must allow the Teacher to choose a level.
  - 15.2.4. The system must allow the Teacher to enter the question.
  - 15.2.5. The system must allow the Teacher to enter the options for the question.
  - 15.2.6. The system must allow the Teacher to enter the correct answer for the question entered.
  - 15.2.7. The system must validate that all the fields are filled in.
    - 15.2.7.1. The system must display the corresponding error message if the validation fails.
  - 15.2.8. The system must display an appropriate message on successfully adding the question.
- 15.3. The system must allow the Teacher to delete a particular question.
  - 15.3.1. The system must allow the Teacher to choose a world.
  - 15.3.2. The system must allow the Teacher to choose a stage.
  - 15.3.3. The system must allow the Teacher to choose a level.
  - 15.3.4. The system must display all the questions on clicking the ‘Get Questions’ button.
  - 15.3.5. The system must allow the Teacher to choose a question to delete.

- 15.3.6. The system must validate that all the fields are filled in.
  - 15.3.6.1. The system must display the corresponding error message if the validation fails.
- 15.3.7. The system must delete the question selected and display the appropriate message on success

## **16. Add New Class Index**

- 16.1. The system must prompt the Teacher to enter the class index that she wished to add.
- 16.2. The system must validate that the field is filled in.
  - 16.2.1. The system must display the corresponding error message if the validation fails.
- 16.3. The system must check if the index entered by the Teacher already exists.
- 16.4. The system must display the corresponding message on successfully adding the new class index.

## 4.2 Use Case Diagram



## 4.3 Use Case Descriptions

### 4.3.1 Use Cases (Students)

Use Case ID:	1		
Use Case Name:	Login		
Created By:	Glenn	Last Updated By:	Glenn
Date Created:	6/2/21	Date Last Updated:	16/4/21

Actor:	Student (Initiating Actor)
Description:	The Student must be able to login with their valid username and password at the login page to be able to use the application.
Preconditions:	<ol style="list-style-type: none"><li>1. The system must be active.</li><li>2. The Student must have a valid account.</li></ol>
Postconditions:	System must grant access to the Student who has successful login or deny access to the Student who has an unsuccessful login. Successful logins bring Students to the homepage.
Priority:	High

Frequency of use:	Very frequent
Flow of events:	<ol style="list-style-type: none"> <li>1. The Student enters his username and password.</li> <li>2. System validates username and password.</li> <li>3. System retrieves Student Data from the database.</li> <li>4. System redirects the Student to the homepage.</li> </ol>
Alternative flows:	If user enters incorrect username or password: <ol style="list-style-type: none"> <li>1. Application informs Student that their username or password is invalid</li> <li>2. Application returns to step 1</li> </ol>
Exceptions:	N/A
Includes:	N/A
Special Requirements:	The checking of Student records in the database should be accurate.
Assumptions:	N/A
Notes and issues:	

Use Case ID:	2
Use Case Name:	Choose characters

Created By:	Glenn	Last Updated By:	Glenn
Date Created:	7/2/21	Date Last Updated:	16/4/21

Actor:	Student (Initiating Actor)
Description:	The Student is able to choose their preferred character in the 'Choose Character' page. The more 'Worlds' the Student has cleared, the more characters the Student is able to choose from. If the Student has not cleared any stages, he/she can only pick the default character.
Preconditions:	<ol style="list-style-type: none"> <li>1. The Student must have a valid Student account.</li> <li>2. The Student must be successfully logged in, i.e. he/she must enter his/her correct username/password.</li> </ol>
Postconditions:	<ol style="list-style-type: none"> <li>1. The system will redirect the Student to the Home page</li> </ol>
Priority:	High
Frequency of use:	Very frequent
Flow of events:	<ol style="list-style-type: none"> <li>1. The Student chooses his character from the list of all characters.</li> <li>2. The Student gets redirected to the main page.</li> </ol>
Alternative flows:	If the Student selects a character which he/she has not yet unlocked: <ol style="list-style-type: none"> <li>1. The 'Select Character' button is greyed and user is unable to click on it</li> </ol>

Exceptions:	N/A
Includes:	N/A
Special Requirements:	N/A
Assumptions:	N/A
Notes and issues:	

Use Case ID:	3		
Use Case Name:	Play Solo Campaign		
Created By:	Glenn	Last Updated By:	Glenn
Date Created:	7/2/21	Date Last Updated:	16/4/21

Actor:	Student (Initiating Actor)
Description:	The Student is given 50 seconds to fight monsters from a particular world and stage. The Student has to get a minimum number of questions correct(5) in order to defeat the monsters and progress to the next difficulty level of that

	<p>stage of that particular world.</p> <p>Each time the Student answers a question from that difficulty level incorrectly, the health of the Student's character will decrease. If the Student's character runs out of health or if time runs out and the student did not fulfil the minimum number of questions correct, he would be deemed to 'fail' that level. He cannot progress to the next difficulty level of that stage of that particular world.</p> <p>10 points will be given for every correct answer answered by the Student. A bonus of 30 points will be awarded to the Student if he answered all questions correctly for that level.</p>
Preconditions:	<ol style="list-style-type: none"> <li>1. The Student must have a valid Student account</li> <li>2. The Student must be successfully logged in, i.e. he/she must enter his/her correct username/password</li> </ol>
Postconditions:	The system will redirect the Student to 'Stages' page(refer to use case 4).
Priority:	High
Frequency of use:	Very frequent
Flow of events:	<ol style="list-style-type: none"> <li>1. The student clicks on 'Solo Campaign' in Home Page</li> </ol>
Alternative flows:	N/A
Exceptions:	N/A
Includes:	Choose World, Display Results
Special	N/A

Requirements:	
Assumptions:	N/A
Notes and issues:	

Use Case ID:	4		
Use Case Name:	Choose World		
Created By:	Glenn	Last Updated By:	Glenn
Date Created:	7/2/21	Date Last Updated:	16/4/21

Actor:	Student (Initiating Actor)
Description:	The Student is able to choose their world, i.e., what topic in Mathematics the Student would like to attempt questions from. Students can only progress to the next world only if they have cleared all the stages in the previous worlds. There are 3 worlds in total.
Preconditions:	<ul style="list-style-type: none"> <li>3. The Student must have a valid Student account.</li> <li>4. The Student must be successfully logged in, i.e. he/she must enter his/her correct username/password.</li> <li>5. The Student must have already chosen a world.</li> </ul>

	<p>6. The Student must have already chosen a stage.</p> <p>7. The Student must have already selected a difficulty level.</p>
Postconditions:	The system will redirect the Student to ‘Results’ page(refer to use case 4).
Priority:	High
Frequency of use:	Very frequent
Flow of events:	<p>2. The Student gets redirected to the ‘Battle’ page.</p> <p>3. The Student selects his answer for the question given through the monster.</p> <p>4. Each time the Student answers a question wrongly, the health of the character that the Student is using decreases.</p> <p>5. The battle ends either when:</p> <ul style="list-style-type: none"> <li>a. Time runs out</li> <li>b. Student has fulfilled minimum number of questions answered correctly</li> <li>c. Student’s character has no more health remaining</li> </ul> <p>6. The Student gets redirected to the ‘Display Results’ page(see use case 7).</p>
Alternative flows:	If Student selects a world he/she has not yet unlocked: <ul style="list-style-type: none"> <li>1. No change would happen in the UI and the Student is not directed to the ‘Choosing Stage’ page.</li> </ul>
Exceptions:	N/A
Includes:	N/A
Special Requirements:	N/A

Assumptions:	N/A
Notes and issues:	

Use Case ID:	5		
Use Case Name:	Choosing stage		
Created By:	Glenn	Last Updated By:	Glenn
Date Created:	7/2/21	Date Last Updated:	16/4/21

Actor:	Student (Initiating Actor)
Description:	The Student is able to choose their stage. Each stage is a specific topic in Mathematics.
Preconditions:	<ol style="list-style-type: none"> <li>1. The Student must have a valid Student account.</li> <li>2. The Student must be successfully logged in, i.e. he/she must enter his/her correct username/password.</li> <li>3. The Student must have already chosen a world.</li> </ol>

Postconditions:	The system will redirect the Student to ‘Choose Level’ page(refer to use case 5).
Priority:	High
Frequency of use:	Frequent
Flow of events:	<ol style="list-style-type: none"> <li>1. The Student selects which stage to enter.</li> <li>2. The Student gets directed to the ‘Choose Level’ page.</li> </ol>
Alternative flows:	If the Student selects a stage he/she has not yet unlocked: <ol style="list-style-type: none"> <li>1. No change would happen in the UI and the Student is not directed to the ‘Choosing level’ page.</li> </ol>
Exceptions:	N/A
Includes:	N/A
Special Requirements:	N/A
Assumptions:	N/A
Notes and issues:	

Use Case ID:	6
Use Case Name:	Choosing level

Created By:	Glenn	Last Updated By:	Glenn
Date Created:	7/2/21	Date Last Updated:	16/4/21

Actor:	Student (Initiating Actor)
Description:	The Student is able to choose from 3 levels: “easy”, “medium” and “hard”. This corresponds to the difficulty level of the questions of that stage.
Preconditions:	<ol style="list-style-type: none"> <li>1. The Student must have a valid Student account.</li> <li>2. The Student must be successfully logged in, i.e. he/she must enter his/her correct username/password.</li> <li>3. The Student must have already chosen a world.</li> <li>4. The Student must have already chosen a stage.</li> </ol>
Postconditions:	The system will redirect the Student to ‘Battle’ page(refer to use case 6).
Priority:	High
Frequency of use:	Frequent
Flow of events:	<ol style="list-style-type: none"> <li>1. The Student selects the level in that stage.</li> <li>2. The Student gets directed to the ‘Battle’ page.</li> </ol>
Alternative flows:	N/A
Exceptions:	N/A

Includes:	N/A
Special Requirements:	N/A
Assumptions:	N/A
Notes and issues:	

Use Case ID:	7		
Use Case Name:	Display results		
Created By:	Glenn	Last Updated By:	Glenn
Date Created:	7/2/21	Date Last Updated:	16/4/21

Actor:	Student (Initiating Actor)
Description:	The Student will be shown whether he/she has cleared the stage of that particular difficulty. There are also buttons where the Student can also share his/her score with his peers on social media, retry the stage, exit(go back to ‘main page’) or proceed to the next level(if he/she has cleared the current stage).

	If Student has failed that particular level, he cannot proceed to the next level.
Preconditions:	<ol style="list-style-type: none"> <li>1. The Student must have a valid Student account.</li> <li>2. The Student must be successfully logged in, i.e. he/she must enter his/her correct username/password.</li> <li>3. The Student must have already chosen a world.</li> <li>4. The Student must have already chosen a stage.</li> <li>5. The Student must have already selected a difficulty level.</li> <li>6. The Student must have finished the battle with the monster.</li> </ol>
Postconditions:	Depending on what the Student selects, he can share his score with his peers on social media, retry the stage, exit(go back to ‘main page’) or proceed to the next stage(if he has cleared the current stage).
Priority:	High
Frequency of use:	Frequent
Flow of events:	<ol style="list-style-type: none"> <li>1. Depending on the Student’s performance during his battle with the monster, if the Student has cleared the level, he will be shown his score. He can share his score with his peers on social media, retry the stage, exit(go back to ‘main page’) or proceed to the next stage(if he has cleared the current stage).</li> </ol>
Alternative flows:	<p>If Student did not clear the level:</p> <ol style="list-style-type: none"> <li>1. He can only share his score with his peers on social media, retry the stage, and exit(go back to ‘main page’). He cannot proceed to the next level of that stage.</li> </ol>
Exceptions:	N/A

Includes:	Share score
Special Requirements:	N/A
Assumptions:	N/A
Notes and issues:	

Use Case ID:	8		
Use Case Name:	View leaderboard		
Created By:	Glenn	Last Updated By:	Glenn
Date Created:	7/2/21	Date Last Updated:	16/4/21

Actor:	Student (Initiating Actor)
Description:	The Student can view his level, score for a particular level, and current ranking in his class index. The Student can only view leaderboard results for Worlds/Stages/Levels that he has cleared.
Preconditions:	1. The Student must have a valid Student account.

	2. The Student must be successfully logged in, i.e. he/she must enter his/her correct username/password.
Postconditions:	N/A
Priority:	Medium
Frequency of use:	Medium
Flow of events:	<ol style="list-style-type: none"> <li>1. Student selects a World</li> <li>2. Student selects a Stage</li> <li>3. Student selects a Level</li> <li>4. Student clicks on 'View' button to view leaderboard results</li> </ol>
Alternative flows:	N/A
Exceptions:	N/A
Includes:	N/A
Special Requirements:	N/A
Assumptions:	N/A
Notes and issues:	

Use Case ID:	9
--------------	---

Use Case Name:	Change settings		
Created By:	Glenn	Last Updated By:	Glenn
Date Created:	7/2/21	Date Last Updated:	16/4/21

Actor:	Student (Initiating Actor)
Description:	In the ‘Settings’ page, the Student can change his password and turn off the game volume
Preconditions:	<ol style="list-style-type: none"> <li>1. The Student must have a valid Student account.</li> <li>2. The Student must be successfully logged in, i.e. he/she must enter his/her correct username/password.</li> </ol>
Postconditions:	N/A
Priority:	Medium
Frequency of use:	Medium
Flow of events:	<ol style="list-style-type: none"> <li>1. Depending on what the Student selects, his password is changed or the game volume is turned off</li> <li>2. If Student chooses to change password, Student must enter his old password and his new password</li> <li>3. The system checks if user’s old password matches and if the new</li> </ol>

	<p>password is not used by another user</p> <ol style="list-style-type: none"> <li>4. System informs user that change is successful/unsuccessful</li> <li>5. If change is successful, Student's account password is updated in the database</li> </ol>
Alternative flows:	<p>If Student selects 'change password' and enters his old password incorrectly:</p> <ol style="list-style-type: none"> <li>1. The application will inform the Student that he has entered his old password incorrectly.</li> </ol> <p>If Student selects 'change password' and enters his old password or an already registered password:</p> <ol style="list-style-type: none"> <li>2. The application will inform the Student that he has entered his old password or the password that he entered has already been registered.</li> </ol>
Exceptions:	N/A
Includes:	N/A
Special Requirements:	N/A
Assumptions:	N/A
Notes and issues:	

Use Case ID:	10
Use Case Name:	Challenge friend

Created By:	Glenn	Last Updated By:	Glenn
Date Created:	7/2/21	Date Last Updated:	16/4/21

Actor:	Student (Initiating Actor)
Description:	<p>In the ‘Challenge Friends’ page, the Student can create challenges and challenge his peers. The Student is able to design the challenge and send it to a Student of his choice. The Student must first complete the challenge themselves before they can send the challenge to their peer. After the Student has completed the challenge, the challenge is sent to the peer automatically by the system. The student can check ‘Sent history’ to see all the past challenges he has sent.</p> <p>The challenged Student must complete the challenge within one day. The winner of the challenge will be reflected in ‘Results’ in ‘Challenge Friends mode upon completion of the challenge by both the Students. The Students can then share the results on social media.</p>
Preconditions:	<ol style="list-style-type: none"> <li>1. The Student must have a valid Student account.</li> <li>2. The Student must be successfully logged in, i.e. he/she must enter his/her correct username/password.</li> <li>3. The Student must click on ‘Challenge Friends’ on the main page.</li> </ol>
Postconditions:	The challenge is sent to his peer.
Priority:	Medium

Frequency of use:	Medium
Flow of events:	<ol style="list-style-type: none"> <li>1. The Student selects 'Challenge Friends' in home page</li> <li>2. The Student clicks on the 'Create Challenges' button.</li> <li>3. The Student selects the world .</li> <li>4. The Student selects the stage.</li> <li>5. The Student selects the level.</li> <li>6. The Student selects the student to send the challenge to.</li> <li>7. The Student completes the challenge.</li> <li>8. The system sends the challenge to his peer.</li> </ol>
Alternative flows:	N/A
Exceptions:	N/A
Includes:	N/A
Special Requirements:	N/A
Assumptions:	N/A
Notes and issues:	

Use Case ID:	11
--------------	----

Use Case Name:	Accept Challenge		
Created By:	Ze Ming	Last Updated By:	Ze Ming
Date Created:	7/2/21	Date Last Updated:	16/4/21

Actor:	Student (Initiating Actor)
Description:	<p>In the ‘Challenge Friends’ page under ‘Pending challenges’, the Student can view pending challenges that other Students have sent to him/her. He/she can view all the past challenges that he has received, under ‘Received History’</p> <p>The winner of the challenge will be reflected in ‘Received History’ in ‘Challenge Friends’ mode upon completion of the challenge by both the Students.</p>
Preconditions:	<ol style="list-style-type: none"> <li>1. The Student must have a valid Student account.</li> <li>2. The Student must be successfully logged in, i.e. he/she must enter his/her correct username/password.</li> <li>3. The Student must have already chosen a character.</li> <li>4. The Student must click on ‘Challenge Friends’ on the main page.</li> <li>5. Another Student must have issued a challenge to the Student currently playing the game.</li> </ol>
Postconditions:	The peer’s challenge is accepted and completed by the user.

Priority:	Medium
Frequency of use:	Medium
Flow of events:	<ol style="list-style-type: none"> <li>1. The Student clicks on the ‘pending challenges’ button.</li> <li>2. The Student chooses a challenge that has not expired and accepts it</li> <li>3. The Student completes the challenge.</li> <li>4. The system displays the challenge results screen, showing the opponent’s and the Student’s points for the game.</li> <li>5. The system informs who won the challenge.</li> </ol>
Alternative flows:	
Exceptions:	N/A
Includes:	N/A
Special Requirements:	N/A
Assumptions:	N/A
Notes and issues:	N/A

Case ID:	12
Use Case Name:	Complete Assignment

Created By:	Ze Ming	Last Updated By:	Ze Ming
Date Created:	7/2/21	Date Last Updated:	16/4/21

Actor:	Student (Initiating Actor)
Description:	<p>Assignments can be viewed under the ‘Assignment’ page and it will allow Students to track each assignment’s status, such as its completion and results. It will also display completed assignments and allow Students to share their results on social media.</p> <p>He can view past assignments that his teacher has sent under ‘Past Assignments’.</p>
Preconditions:	<ol style="list-style-type: none"> <li>1. The Student must have a valid Student account.</li> <li>2. The Student must be successfully logged in, i.e. he/she must enter his/her correct username/password.</li> <li>3. The Student must have already chosen a character.</li> <li>4. The Student’s Teacher must have already given an assignment through the game.</li> <li>5. The Student’s Teacher must have already given the code needed to access the assignment.</li> </ol>
Postconditions:	The assignment is completed by the Student.
Priority:	High

Frequency of use:	Medium-High
Flow of events:	<ol style="list-style-type: none"> <li>1. The Student clicks on the ‘Assignment’ button</li> <li>2. The Student clicks on the ‘Pending assignments’ button.</li> <li>3. Student chooses one of the assignments and clicks “Play” button</li> <li>4. The Student completes the assignment.</li> <li>5. The Student clicks on the ‘share’ button to share his/her results on social media.</li> </ol>
Alternative flows:	<ol style="list-style-type: none"> <li>1. The Student clicks on the ‘Assignment’ button.</li> <li>2. The Student clicks on the ‘Pending assignments’ button.</li> <li>3. The Student views his/her past assignment results.</li> </ol>
Exceptions:	N/A
Includes:	N/A
Special Requirements:	N/A
Assumptions:	N/A
Notes and issues:	N/A

Case ID:	13
Use Case Name:	Change Character

Created By:	Ze Ming	Last Updated By:	Ze Ming
Date Created:	7/2/21	Date Last Updated:	16/4/21

Actor:	Student (Initiating Actor)
Description:	The Student can change his/her character in the ‘Change Character’ screen.
Preconditions:	<ol style="list-style-type: none"> <li>1. The Student must have a valid Student account.</li> <li>2. The Student must be successfully logged in, i.e. he/she must enter his/her correct username/password.</li> <li>3. The Student must have already chosen a character.</li> </ol>
Postconditions:	The Student’s character has changed
Priority:	Medium
Frequency of use:	Medium
Flow of events:	<ol style="list-style-type: none"> <li>1. The Student clicks on the ‘Change Character’ screen.</li> <li>2. The Student is shown a list of characters that the user has unlocked.</li> <li>3. The Student chooses one of his/her characters from the list that is not his/her current character.</li> <li>4. The Student’s character has now been changed.</li> </ol>
Alternative flows:	N/A

Exceptions:	N/A
Includes:	N/A
Special Requirements:	N/A
Assumptions:	N/A
Notes and issues:	N/A

### 4.3.2 Use Cases (Teachers)

Case ID:	14		
Use Case Name:	Login		
Created By:	Ze Ming	Last Updated By:	Ze Ming
Date Created:	8/2/21	Date Last Updated:	16/4/21

Actor:	Teacher (Initiating Actor)
Description:	The user is able to login with their valid username and password at the login

	page to be able to use the application.
Preconditions:	<ol style="list-style-type: none"> <li>1. The application must be open.</li> <li>2. The user must have a valid Teacher's account with a registered email address.</li> <li>3. The user must be a Teacher.</li> <li>4. The system must already have a record in the database containing the Teacher's account details.</li> </ol>
Postconditions:	System must grant access to the user who has successful login or deny access to the user who has an unsuccessful login. Successful events bring users to the homepage.
Priority:	High
Frequency of use:	High
Flow of events:	<ol style="list-style-type: none"> <li>1. The user enters his/her password and email address.</li> <li>2. The system retrieves Teacher Data from the database.</li> <li>3. The system validates email address and password from the database.</li> <li>4. The system checks if the account is valid.</li> <li>5. The system redirects the user to the home page upon successful login.</li> </ol>
Alternative flows:	<ol style="list-style-type: none"> <li>1. The Teacher does not key in one of the above fields.             <ol style="list-style-type: none"> <li>1.1. The system displays an error message informing the Teacher that one of the fields is not filled in.</li> </ol> </li> <li>2. The Teacher keys in an incorrect email.             <ol style="list-style-type: none"> <li>2.1. The system displays an error message informing the Teacher that the email address is filled in wrongly.</li> </ol> </li> <li>3. The Teacher clicks on the forget password option.             <ol style="list-style-type: none"> <li>3.1. The Teacher keys in his/her registered email address.</li> </ol> </li> </ol>

	<p>3.2. The system validates the email address and sends a One-time Password(OTP) to the email address.</p> <p>3.3. The system prompts the Teacher to enter the OTP sent.</p> <p>3.4. The Teacher enters the OTP.</p> <p>3.5. The system validates the OTP entered with the one sent.</p> <p>3.6. The system prompts the Teacher for a new password and reconfirms it.</p> <p>3.7. The Teacher enters the new password twice.</p> <p>3.8. The system validates that the new password matches on both inputs.</p> <p>3.9. The system saves the new password into the database.</p>
Exceptions:	1. If Teacher fills in the OTP wrongly three times, the system redirects the Teacher back to the initial login page
Includes:	N/A
Special Requirements:	The checking of Teacher record in the Teacher Database should be accurate
Assumptions:	N/A
Notes and issues:	N/A

Case ID:	15
Use Case Name:	Add new Student

Created By:	Ze Ming	Last Updated By:	Ze Ming
Date Created:	8/2/21	Date Last Updated:	16/4/21

Actor:	Teacher (Initiating Actor)
Description:	The Teacher is able to create a Student account for the game on the 'Add New Student' Page.
Preconditions:	<ol style="list-style-type: none"> <li>1. The Teacher must have a valid Teacher's account</li> <li>2. The Teacher must have successfully logged in</li> <li>3. The Student has a registered email address</li> </ol>
Postconditions:	A new Student account is created by the Teacher
Priority:	High
Frequency of use:	Medium
Flow of events:	<ol style="list-style-type: none"> <li>1. The Teacher clicks on the 'Add New Student' option in the home page.</li> <li>2. System prompts the teacher to enter the Student's full name, username, email address and class index.</li> <li>3. The Teacher enters all the fields.</li> <li>4. The system validates all the entered fields to make sure that they are correct.</li> <li>5. The system assigns the default password to the new Student account.</li> </ol>

	<p>6. A new Student account is created.</p> <p>7. The system notifies both the Teacher and Student through an email upon successful account creation.</p>
Alternative flows:	<p>1. Teacher enters a Student username that already exists.</p> <p>1.1. The system displays a corresponding error message that the username is already taken.</p> <p>2. Teacher enters an email address that is invalid.</p> <p>2.1. The system displays a corresponding error message that the email address is invalid.</p> <p>3. Teacher does not fill in all of the fields.</p> <p>3.1. The system displays a corresponding error message that at least one of the fields are not filled in.</p>
Exceptions:	N/A
Includes:	N/A
Special Requirements:	<p>1. The default password must be given to the new Student by the system.</p>
Assumptions:	N/A
Notes and issues:	N/A

Case ID:	16
Use Case Name:	Delete Student Account

Created By:	Ze Ming	Last Updated By:	Ze Ming
Date Created:	8/2/21	Date Last Updated:	16/4/21

Actor:	Teacher (Initiating Actor)
Description:	The Teacher can delete a Student's account in the 'Delete Student' page.
Preconditions:	<ul style="list-style-type: none"> <li>1. The Teacher must have a valid Teacher's account.</li> <li>2. The Teacher must have successfully logged in.</li> <li>3. There is at least one Student account registered in the system.</li> </ul>
Postconditions:	The Student's account is deleted and removed from the system's database.
Priority:	High
Frequency of use:	Low-Medium
Flow of events:	<ul style="list-style-type: none"> <li>1. The Teacher clicks on the 'Delete Student Account' option in the home page.</li> <li>2. The System prompts Teacher for the Student-to-be-deleted username.</li> <li>3. The Teacher enters the Student's username.</li> <li>4. The system checks if the entered username is valid.</li> <li>5. The system deletes the entered username's Student account.</li> <li>6. The system sends the Student an email to notify successful termination of his/her account.</li> </ul>

Alternative flows:	<ol style="list-style-type: none"> <li>1. The Teacher enters an invalid username.           <ol style="list-style-type: none"> <li>1.1. The system checks and realises the username is invalid.</li> <li>1.2. The system displays a corresponding error message informing the Teacher that the username is invalid.</li> </ol> </li> </ol>
Exceptions:	N/A
Includes:	N/A
Special Requirements:	N/A
Assumptions:	N/A
Notes and issues:	N/A

Case ID:	17		
Use Case Name:	View Student's progress		
Created By:	Ze Ming	Last Updated By:	Ze Ming
Date Created:	8/2/21	Date Last Updated:	16/4/21

Actor:	Teacher (Initiating Actor)
Description:	<p>The Teacher is able to check on the progress of a particular student in a particular class index, in the ‘View Progress’ Screen. The Teacher can choose to view progress by World or Stage or Level, which shows data on a particular student in that world/Stage/Level, depending on what option the Teacher selects.</p> <p>To view the class progress of a particular stage, the field “Choose World” must be filled; similarly, to view the class progress of a particular level, the field “Choose World” and “Choose Stage” must be filled.</p> <p>After ‘Get Students’ button is clicked, the Teacher can select to view progress on a particular student</p>
Preconditions:	<ol style="list-style-type: none"> <li>1. The Teacher must have a valid Teacher’s account.</li> <li>2. The Teacher must have successfully logged in.</li> <li>3. The Teacher must have already clicked on a class index to view the list of Students.</li> <li>4. The Students in that class must have already registered their accounts in the system.</li> <li>5. The Student must have already cleared at least one world.</li> </ol>
Postconditions:	The Students progress in the game is clearly viewed by the Teacher.
Priority:	High
Frequency of use:	Medium

Flow of events:	<ol style="list-style-type: none"> <li>1. The Teacher clicks on the ‘View Progress’ screen.</li> <li>2. The Teacher selects the world</li> <li>3. The Teacher selects the stage(click on ‘all’ to view data on all stages of that particular World)</li> <li>4. The Teacher selects the Level(click on ‘all’ to view data on all levels of that particular Stage)</li> <li>5. The Teacher selects the Index</li> <li>6. The Teacher clicks on ‘Get Students’ button</li> <li>7. The System will pop-up a ‘Select Student’ option</li> <li>8. The Teacher selects a particular student under ‘Select Student’</li> <li>9. The Teacher clicks on ‘View Progress’ button</li> <li>10. The system displays the data of the student</li> </ol>
Alternative flows:	N/A
Exceptions:	N/A
Includes:	N/A
Special Requirements:	N/A
Assumptions:	N/A
Notes and issues:	N/A

Case ID:	18
Use Case Name:	Send assignment

Created By:	Ze Ming	Last Updated By:	Glenn
Date Created:	8/2/21	Date Last Updated:	6/4/21

Actor:	Teacher (Initiating Actor)
Description:	The teacher is able to create and send an assignment to all the Students in the same class index.
Preconditions:	<ol style="list-style-type: none"> <li>1. The Teacher must have a valid Teacher's account.</li> <li>2. The Teacher must have successfully logged in.</li> <li>3. The Teacher must have at least one class assigned to him/her.</li> </ol>
Postconditions:	The assignment and its corresponding code is sent to Students' accounts and social media respectively.
Priority:	High
Frequency of use:	High
Flow of events:	<ol style="list-style-type: none"> <li>1. The Teacher selects the Assignment title</li> <li>2. The Teacher selects the Assignment code</li> <li>3. The Teacher selects the class index</li> <li>4. The Teacher chooses the World for the topic of the assignment</li> <li>5. The system sends the assignment to all Students under the Teacher.</li> <li>6. The system provides a unique assignment code to the Teacher.</li> <li>7. The Teacher clicks on the 'Share' button to share the assignment code</li> </ol>

	<p>and chooses a social media to share it on.</p> <p>7.1. The Teacher chooses facebook.</p> <p>    7.1.1. The system shares the code on facebook.</p> <p>7.2. Teacher chooses twitter.</p> <p>    7.2.1. The system shares the code on twitter.</p>
Alternative flows:	N/A
Exceptions:	N/A
Includes:	N/A
Special Requirements:	<ul style="list-style-type: none"> <li>1. Requires a social media page on facebook and twitter where Students can go to view the assignment code.</li> </ul>
Assumptions:	<ul style="list-style-type: none"> <li>1. Students will actively go to the social media websites and look for the unique code.</li> </ul>
Notes and issues:	N/A

Case ID:	19		
Use Case Name:	View Class progress		
Created By:	Ze Ming	Last Updated By:	Ze Ming

Date Created:	8/2/21	Date Last Updated:	16/4/21
---------------	--------	--------------------	---------

Actor:	Teacher (Initiating Actor)
Description:	<p>The Teacher is able to check on the progress of a particular class index as a whole, in the ‘View Progress’ Screen. The Teacher can choose to view progress by World or Stage or Level, which shows how many students cleared and how many did not clear a particular world/Stage/Level, depending on what option the Teacher selects.</p> <p>To view the class progress of a particular stage, the field “Choose World” must be filled; similarly, to view the class progress of a particular level, the field “Choose World” and “Choose Stage” must be filled.</p> <p>After the ‘Get Students’ button is clicked, the Teacher must select ‘All’ option to get progress data of the class index as a whole.</p>
Preconditions:	<ol style="list-style-type: none"> <li>1. The Teacher must have a valid Teacher’s account.</li> <li>2. The Teacher must have successfully logged in.</li> <li>3. The Teacher must have at least one class assigned to him/her.</li> </ol>
Postconditions:	The class’ progress is viewed by the Teacher.
Priority:	High
Frequency of use:	Medium-High

Flow of events:	<ol style="list-style-type: none"> <li>1. The Teacher clicks on the ‘View Progress’ screen.</li> <li>2. The Teacher selects the world</li> <li>3. The Teacher selects the stage(click on ‘all’ to view data on all stages of that particular World)</li> <li>4. The Teacher selects the Level(click on ‘all’ to view data on all levels of that particular Stage)</li> <li>5. The Teacher selects the Index</li> <li>6. The Teacher clicks on ‘Get Students’ button</li> <li>7. The System will pop-up a ‘Select Student’ option</li> <li>8. The Teacher selects ‘All’ option under ‘Select Student’</li> <li>9. The Teacher clicks on ‘View Progress’ button</li> <li>10. The system displays the list of Students in that class index in descending order of the total points scored.</li> </ol>
Alternative flows:	<ol style="list-style-type: none"> <li>1. Teacher did not select any option for any of the fields: ‘Choose World’, ‘Choose Stage’, ‘Choose Level’ and ‘Choose index’, but clicks on ‘Get Students’ button             <ol style="list-style-type: none"> <li>1.1. System will prompt user to Select a World</li> </ol> </li> <li>2. Teacher selects an option for one of the fields but did not select any option for the rest of the fields, and clicks on ‘Get Students’ button             <ol style="list-style-type: none"> <li>2.1. System will prompt user</li> </ol> </li> <li>3. Teacher forgot to select an option for one of the fields and clicks on ‘Get Students’ button             <ol style="list-style-type: none"> <li>3.1. System will prompt user</li> </ol> </li> </ol>
Exceptions:	N/A
Includes:	View Student’s progress

Special Requirements:	N/A
Assumptions:	N/A
Notes and issues:	N/A

Case ID:	_____		
Use Case Name:	View assignment progress		
Created By:	Ze Ming	Last Updated By:	Ze Ming
Date Created:	8/2/21	Date Last Updated:	16/4/21

Actor:	Teacher (Initiating Actor)
Description:	The Teacher can track the progress made by Students in the assignments in the ‘Assignment Progress’ page.
Preconditions:	<ol style="list-style-type: none"> <li>1. The Teacher must have a valid Teacher’s account.</li> <li>2. The Teacher must have successfully logged in.</li> <li>3. The Teacher must have already sent out at least one assignment.</li> </ol>

Postconditions:	The Teacher views the progress made by Students for his/her assignments.
Priority:	High
Frequency of use:	Medium-High
Flow of events:	<ol style="list-style-type: none"> <li>1. The Teacher clicks on the 'Assignment Progress' page.</li> <li>2. The system prompts the Teacher to enter the assignment's code which was previously sent during the assignment creation stage.</li> <li>3. The Teacher enters the assignment's unique code.</li> <li>4. The system displays the highest score obtained in that assignment.</li> <li>5. The system displays the number of Students that have failed the assignment.</li> <li>6. The system displays the number of Students that have not attempted the assignment.</li> </ol>
Alternative flows:	N/A
Exceptions:	N/A
Includes:	N/A
Special Requirements:	N/A
Assumptions:	N/A
Notes and issues:	N/A

Case ID:	21		
Use Case Name:	Edit game		
Created By:	Ze Ming	Last Updated By:	Ze Ming
Date Created:	8/2/21	Date Last Updated:	16/4/21

Actor:	Teacher (Initiating Actor)
Description:	The Teacher is able to add or delete questions in the database in the 'Edit Question Bank' page.
Preconditions:	<ol style="list-style-type: none"> <li>1. The Teacher must have a valid Teacher's account.</li> <li>2. The Teacher must have successfully logged in.</li> <li>3. There is already an existing question bank.</li> </ol>
Postconditions:	The questions in the question bank for the game is edited by the Teacher.
Priority:	Medium
Frequency of use:	Medium-High

Flow of events:	<ol style="list-style-type: none"> <li>1. The Teacher selects the World</li> <li>2. The Teacher selects the Stage</li> <li>3. The Teacher selects the Level.</li> <li>4. The Teacher chooses one of the two options, ‘add question’, ‘delete question’.</li> <li>5. The system displays an appropriate message upon completion of each of the two options.</li> </ol>
Alternative flows:	N/A
Exceptions:	N/A
Includes:	<ol style="list-style-type: none"> <li>1. Add question</li> <li>2. Delete question</li> </ol>
Special Requirements:	N/A
Assumptions:	<ol style="list-style-type: none"> <li>1. There is already an existing question bank with sufficient questions to be added or deleted.</li> </ol>
Notes and issues:	N/A

Case ID:	22
Use Case Name:	Add question

Created By:	Ze Ming	Last Updated By:	Ze Ming
Date Created:	8/2/21	Date Last Updated:	16/4/21

Actor:	Teacher (Initiating Actor)
Description:	The Teacher is able to add a question to the question bank in the 'Edit Question Bank' page.
Preconditions:	<ol style="list-style-type: none"> <li>1. The Teacher must have a valid Teacher's account.</li> <li>2. The Teacher must have successfully logged in.</li> <li>3. The Teacher must have clicked on the 'Edit Question Bank' page.</li> </ol>
Postconditions:	A new question is added to the question bank.
Priority:	Medium
Frequency of use:	Medium-High
Flow of events:	<ol style="list-style-type: none"> <li>1. The Teacher clicks on the 'Add question' button.</li> <li>2. The Teacher chooses a world.</li> <li>3. Teacher chooses a stage.</li> <li>4. The Teacher chooses a level.</li> <li>5. The Teacher enters a question and answer.</li> <li>6. The system validates that all the fields are filled in.</li> <li>7. The system adds a new question to the question bank.</li> <li>8. The system saves all the changes made into the database.</li> <li>9. The system displays an appropriate message upon completion of</li> </ol>

	addition.
Alternative flows:	<ol style="list-style-type: none"> <li>1. The system prompts the Teacher with the above options</li> <li>2. The Teacher fills one of the fields incorrectly.</li> <li>3. The system displays a corresponding error message informing the Teacher of the incorrect input.</li> <li>4. Go back to step 1.</li> </ol>
Exceptions:	N/A
Includes:	N/A
Special Requirements:	N/A
Assumptions:	N/A
Notes and issues:	N/A

Case ID:	23		
Use Case Name:	Delete question		
Created By:	Ze Ming	Last Updated By:	Ze Ming
Date Created:	8/2/21	Date Last Updated:	16/4/21

Actor:	Teacher (Initiating Actor)
Description:	The Teacher is able to delete a particular question in the ‘Edit Question Bank’ page
Preconditions:	<ol style="list-style-type: none"> <li>1. The Teacher must have a valid Teacher’s account.</li> <li>2. The Teacher must have logged in successfully.</li> <li>3. The Teacher must have clicked on the ‘Edit Question Bank’ button.</li> <li>4. There is at least one question in the question bank.</li> </ol>
Postconditions:	A question in the question bank is deleted
Priority:	Medium
Frequency of use:	Low-medium
Flow of events:	<ol style="list-style-type: none"> <li>1. The Teacher clicks on the ‘delete question’ button.</li> <li>2. The Teacher chooses a world.</li> <li>3. Teacher chooses a stage.</li> <li>4. The Teacher chooses a level.</li> <li>5. The Teacher chooses one of the questions to delete.</li> <li>6. The system validates that all the fields are filled in.</li> <li>7. The system deletes the chosen question.</li> <li>8. The system saves all the changes made into the database.</li> <li>9. The system displays an appropriate message upon completion of deletion.</li> </ol>
Alternative flows:	N/A
Exceptions:	N/A

Includes:	N/A
Special Requirements:	N/A
Assumptions:	N/A
Notes and issues:	N/A

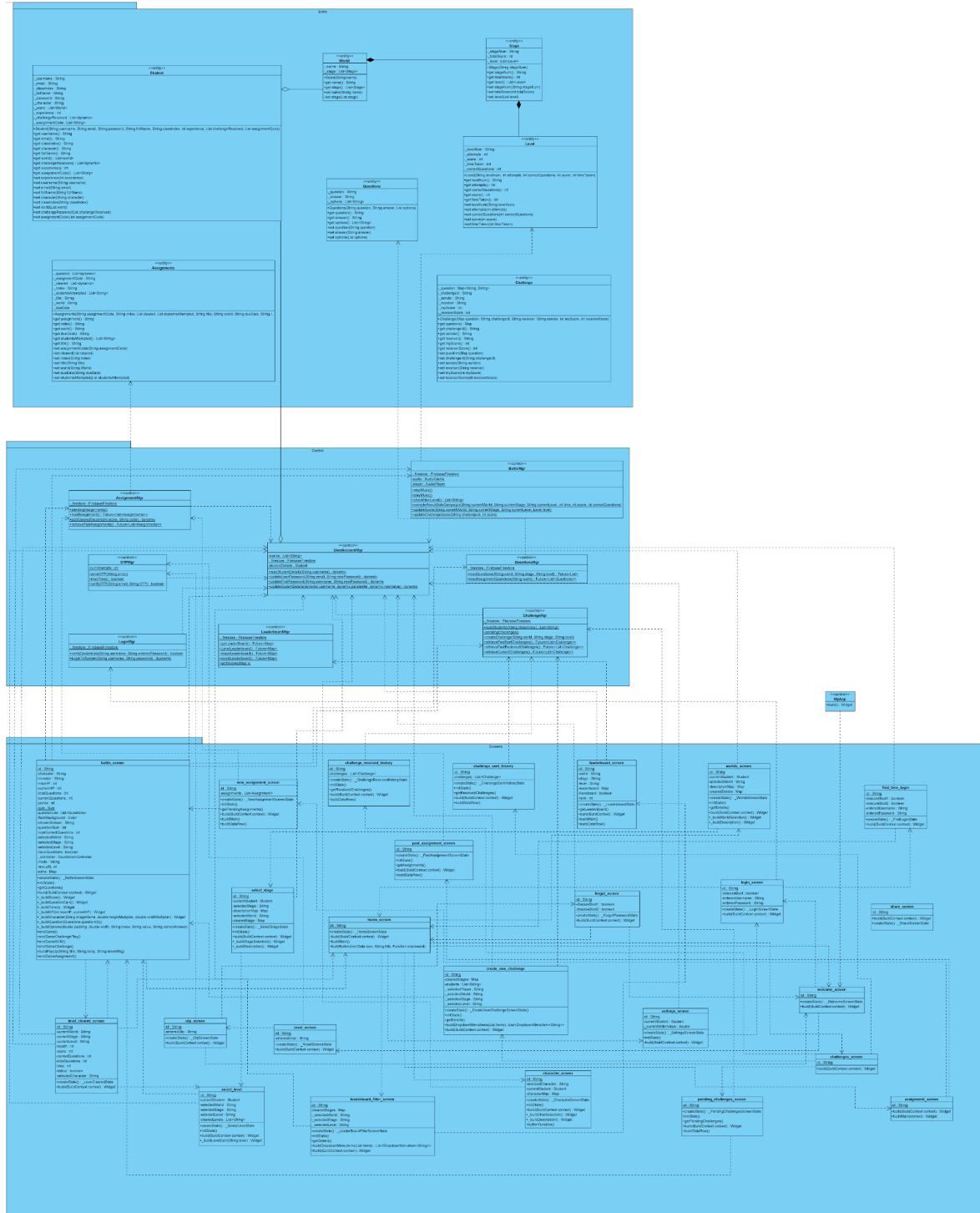
Case ID:	24		
Use Case Name:	Add new Class Index		
Created By:	Ze Ming	Last Updated By:	
Date Created:	17/4/21	Date Last Updated:	

Actor:	Teacher (Initiating Actor)
Description:	The Teacher is able to create a new class index
Preconditions:	<ol style="list-style-type: none"> <li>1. The Teacher must have a valid Teacher's account</li> <li>2. The Teacher must have successfully logged in</li> </ol>
Postconditions:	A new class index is created by the Teacher

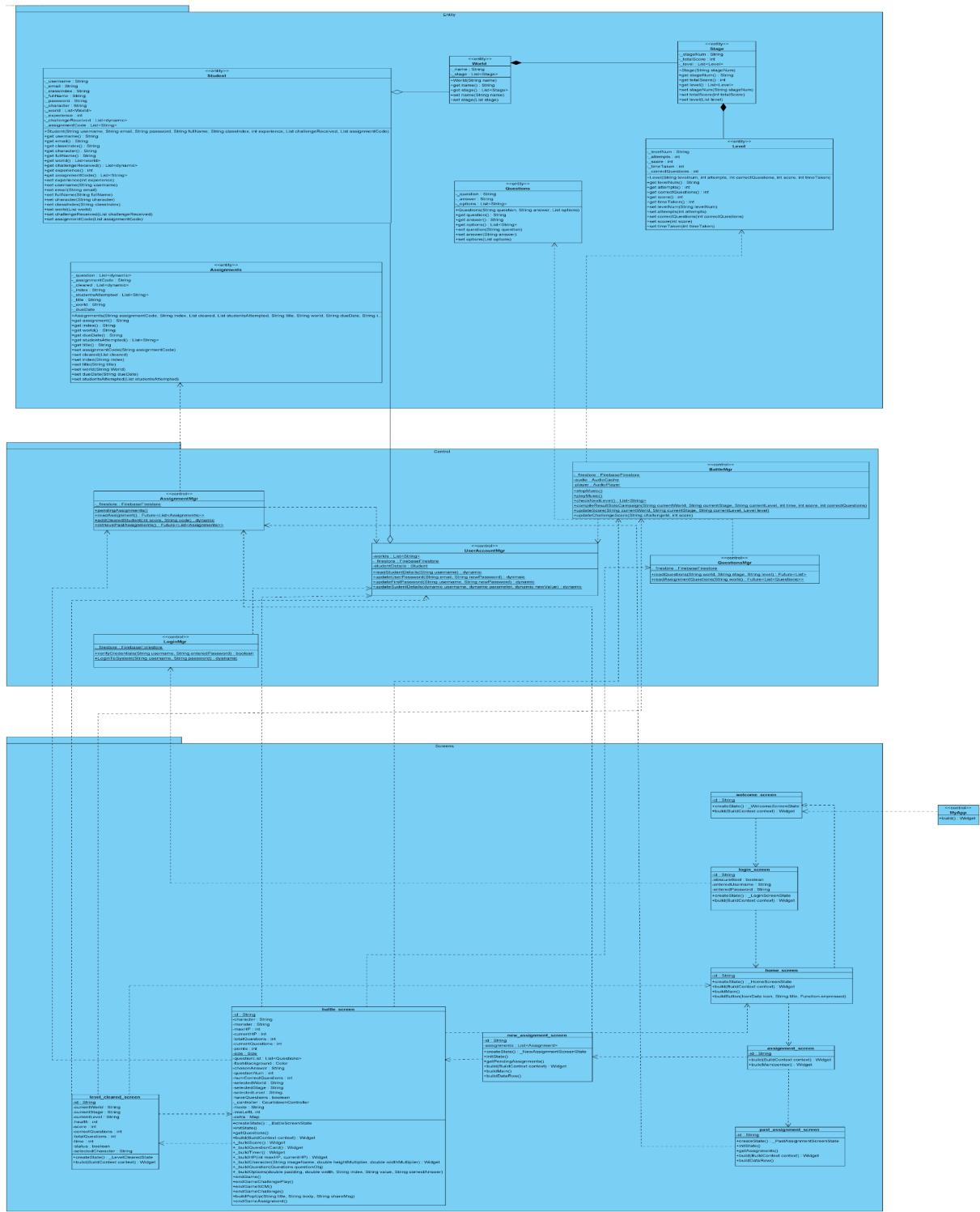
Priority:	Medium
Frequency of use:	Low
Flow of events:	<ol style="list-style-type: none"> <li>1. The Teacher clicks on the 'Add New Class Index' option in the home page.</li> <li>2. System prompts the teacher to enter the new class index.</li> <li>3. The Teacher enters all the fields.</li> <li>4. The system validates all the entered fields to make sure that they are correct.</li> <li>5. The system checks to make sure there is no existing class with the same index.</li> <li>6. A new class index is created and the system displays a corresponding message to signify it.</li> </ol>
Alternative flows:	<ol style="list-style-type: none"> <li>1. Teacher enters a class index that already exists.             <ol style="list-style-type: none"> <li>a. The system displays a corresponding error message that the index is already taken.</li> </ol> </li> <li>2. Teacher does not fill in all of the fields.             <ol style="list-style-type: none"> <li>a. The system displays a corresponding error message that at least one of the fields are not filled in.</li> </ol> </li> </ol>
Exceptions:	N/A
Includes:	N/A
Special Requirements:	N/A
Assumptions:	N/A
Notes and issues:	N/A

## **4.4 Class Diagrams**

### **4.4.1 Main**



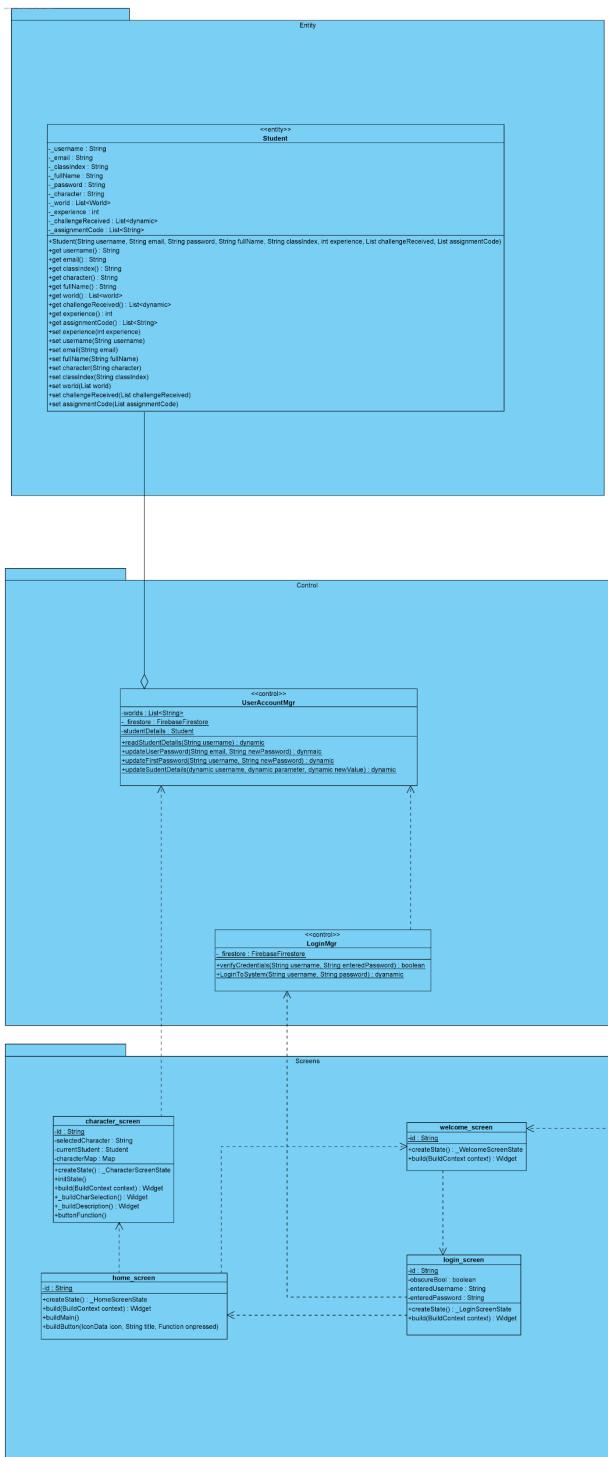
## 4.4.2 Assignment



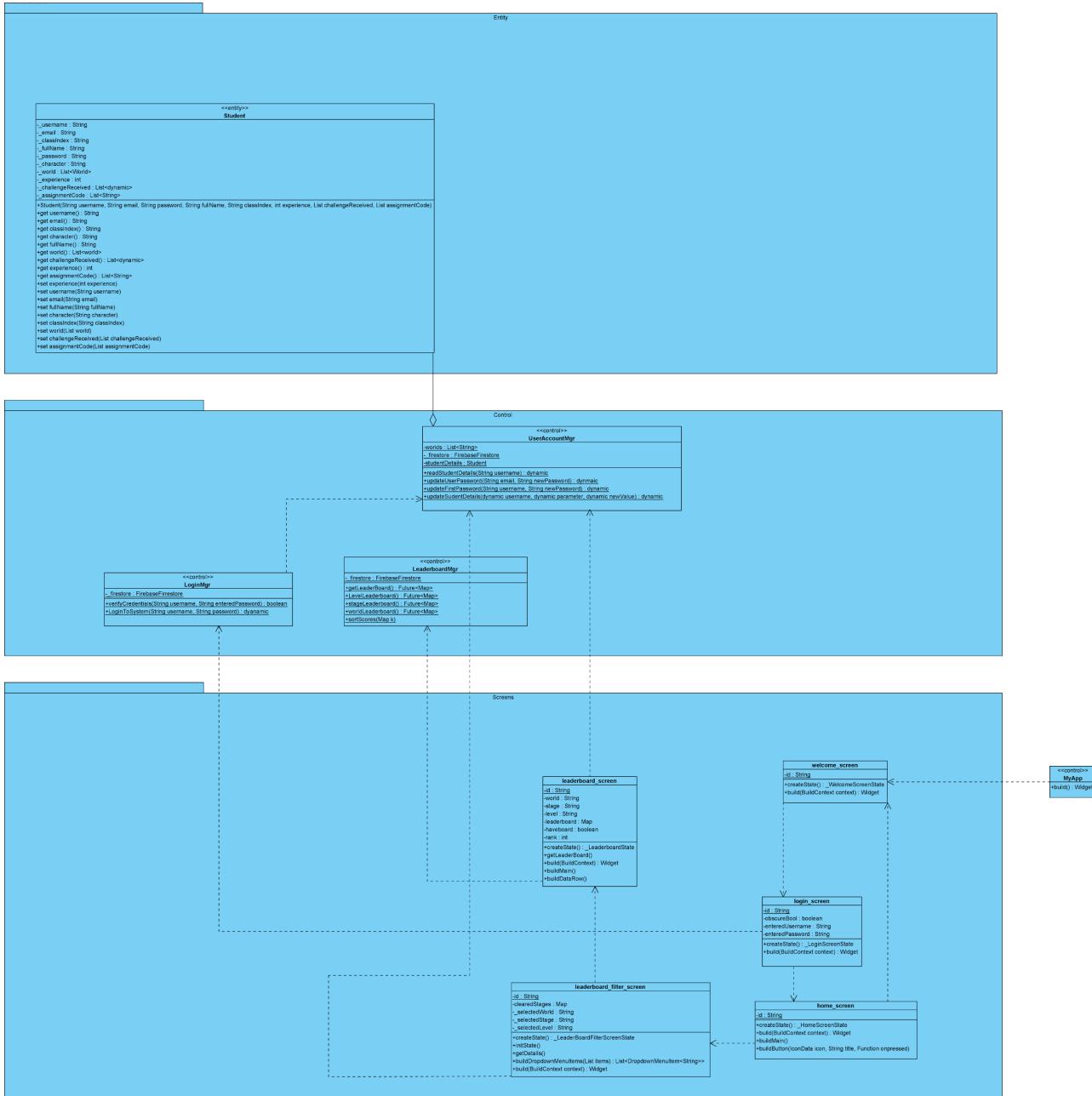
#### 4.4.3 Challenge



#### 4.4.4 Character



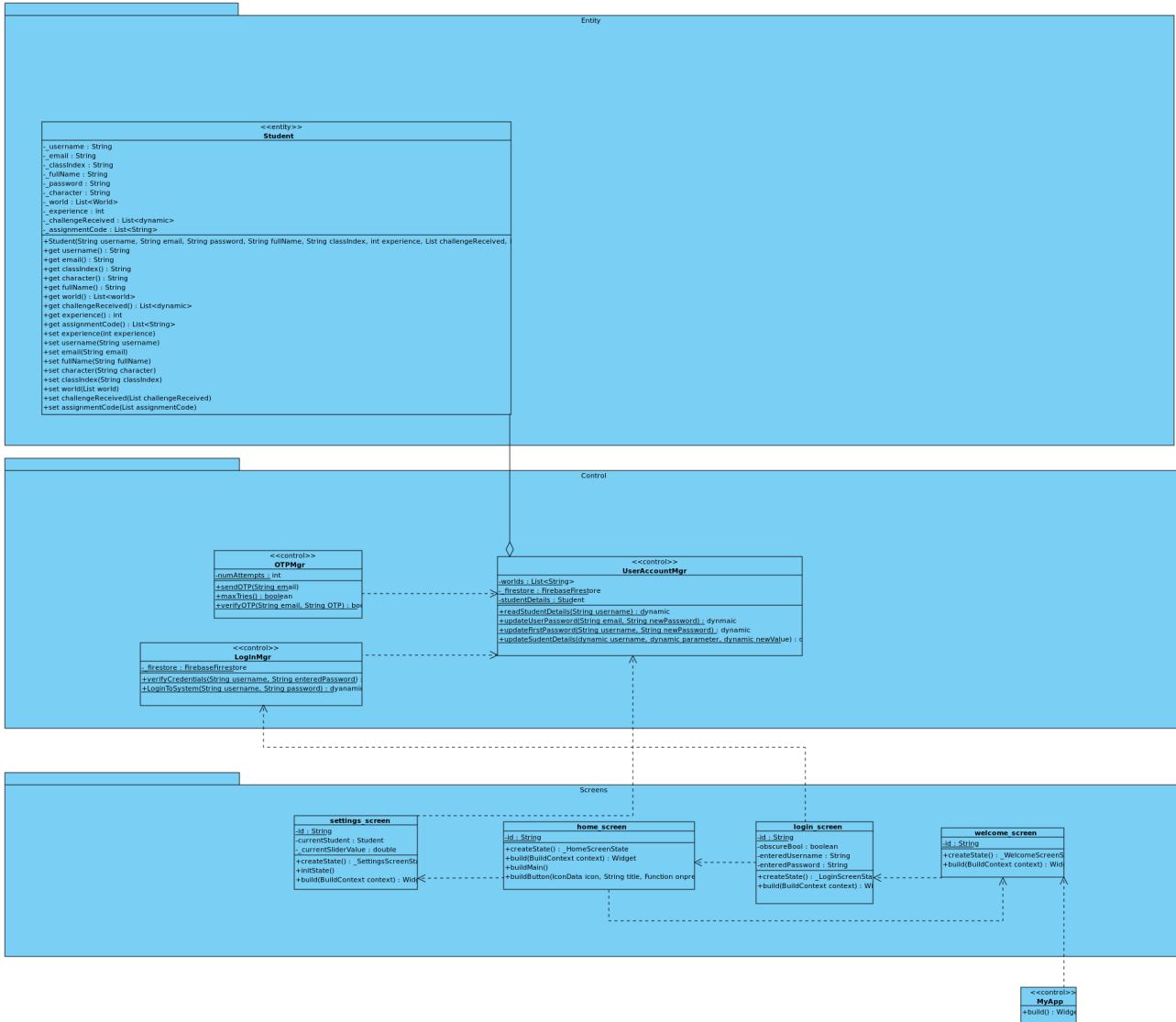
#### 4.4.5 LeaderBoard



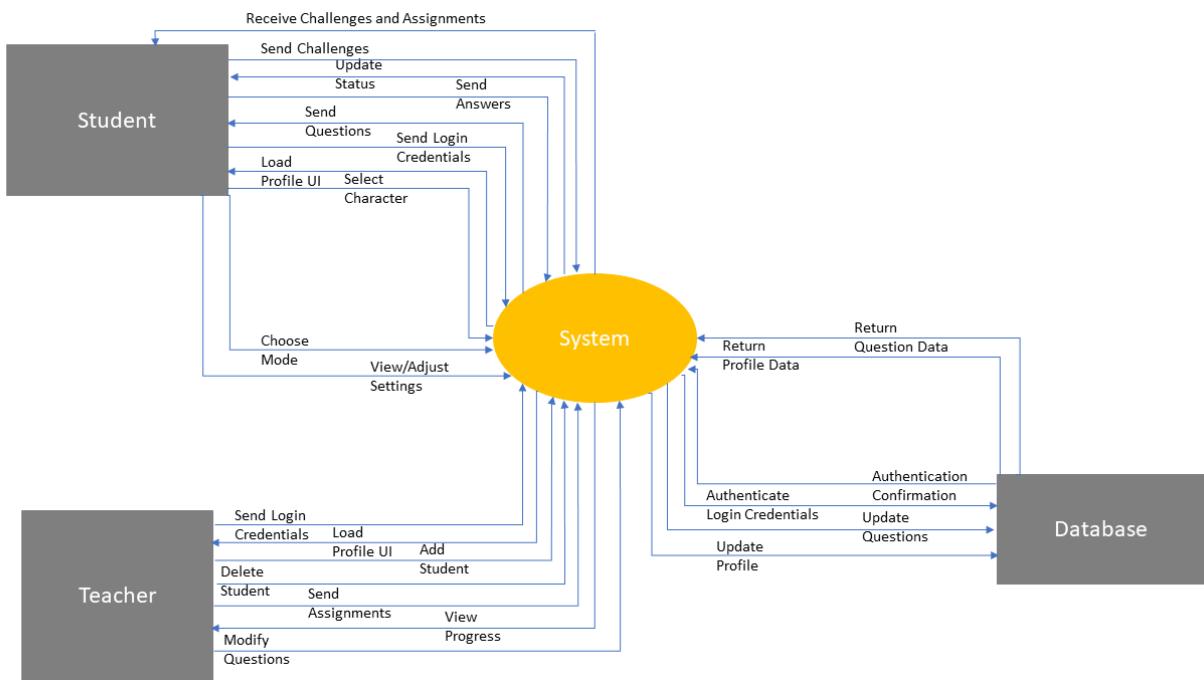
#### 4.4.6 Solo Campaign



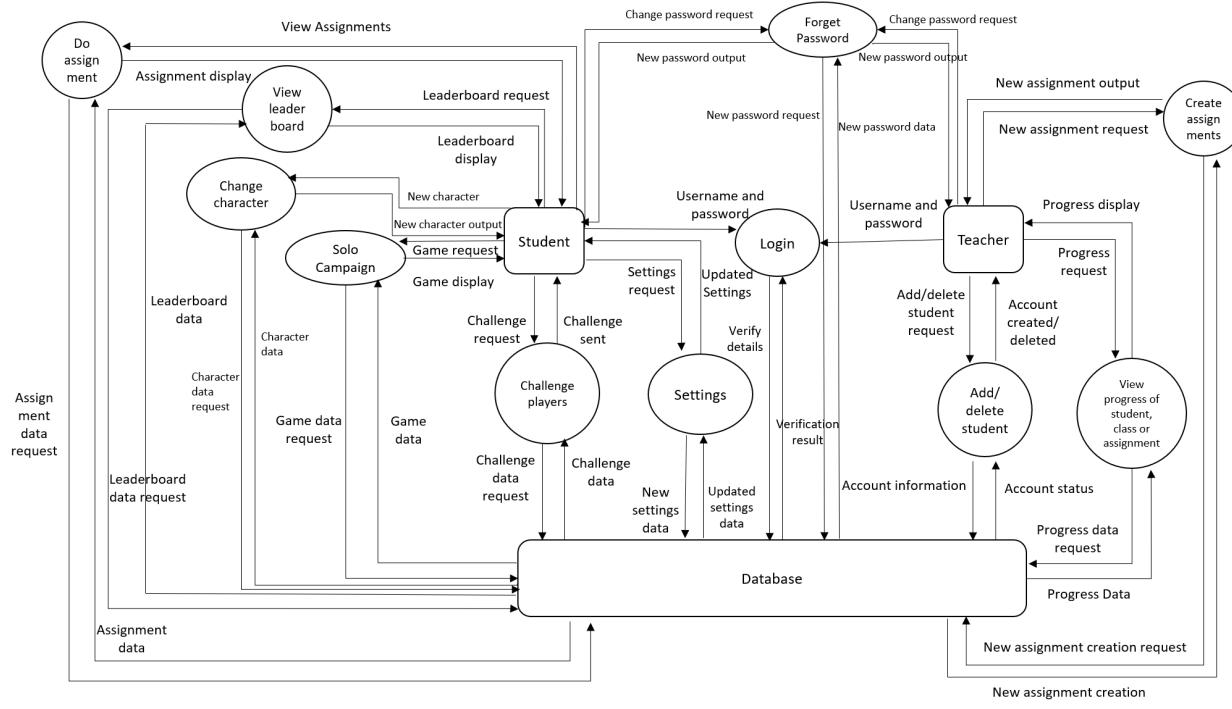
#### 4.4.7 Settings



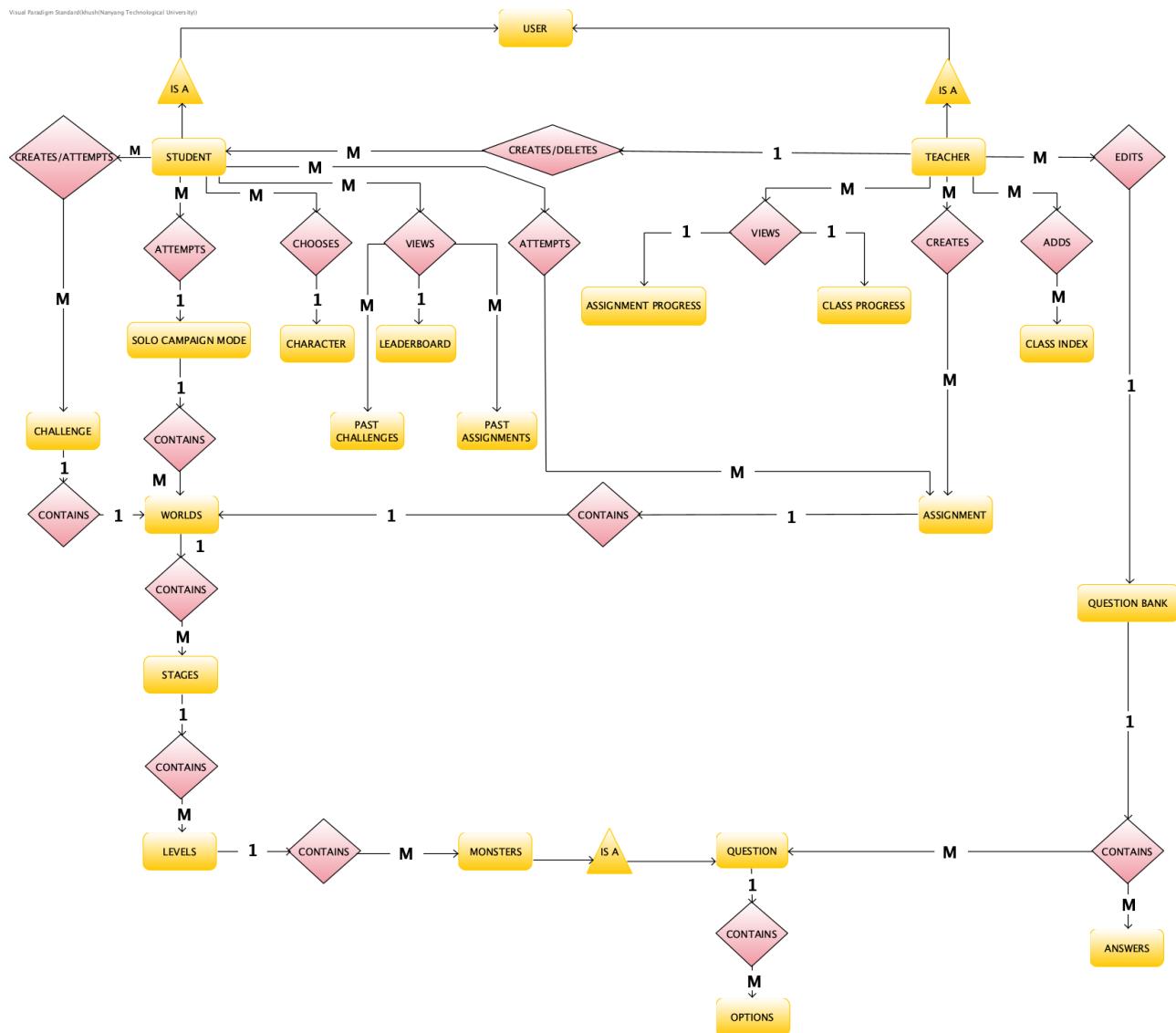
## 4.5 Context Diagram



## 4.6 Data Flow Diagram



## 4.7 Entity Relationship Diagram



## 4.8 Decisions Table

### 4.8.1 Challenge

Challenge mode			
Pending challenge	Y	N	N
Sent challenges	N	Y	N
Received challenges	N	N	Y
Actions			
See pending challenges	Y	N	N
See past challenges	N	Y	N
See past received challenges	N	N	Y

### 4.8.2 Sending Challenge

Challenge mode					
World	Y	N	N	Y	Y
Stage	N	Y	N	Y	Y
Level	N	N	Y	Y	Y
User	N	N	N	N	Y
Actions					
Challenge sent	N	N	N	N	Y

### 4.8.3 Assignment

Assignments			
Pending	Y	Y	N
Past	N	Y	Y
Actions			
Access to assignment	Y	Y	N

#### 4.8.4 Leaderboard

Leaderboard				
Difficulty	N	N	Y	Y
World	Y	N	N	Y
Stage	N	Y	N	Y
Actions				
Arrange by points	N	N	N	Y

#### 4.8.5 Completion of Solo Campaign

Table 1					
Solo campaign mode (Completion)					
Special ability	N	N	N	Y	Y
Clear 50% of the monster	N	N	Y	N	Y
Time limit	N	Y	Y	Y	Y
Action					
Cleared	N	N	Y	N	Y
Point	N	Y	Y	Y	Y
Round over	N	Y	Y	Y	Y

#### 4.8.6 Playing Solo Campaign

Solo Campaign mode (Creation)			
<b>Unlocked</b>	Y	Y	Y
<b>Hard</b>	Y	N	N
<b>Medium</b>	N	Y	N
<b>Easy</b>	N	N	Y
<b>Action</b>			
<b>Duration 50 sec</b>	Y	Y	Y
<b>Max enemy 10</b>	Y	Y	Y

#### 4.8.7 Selecting Character

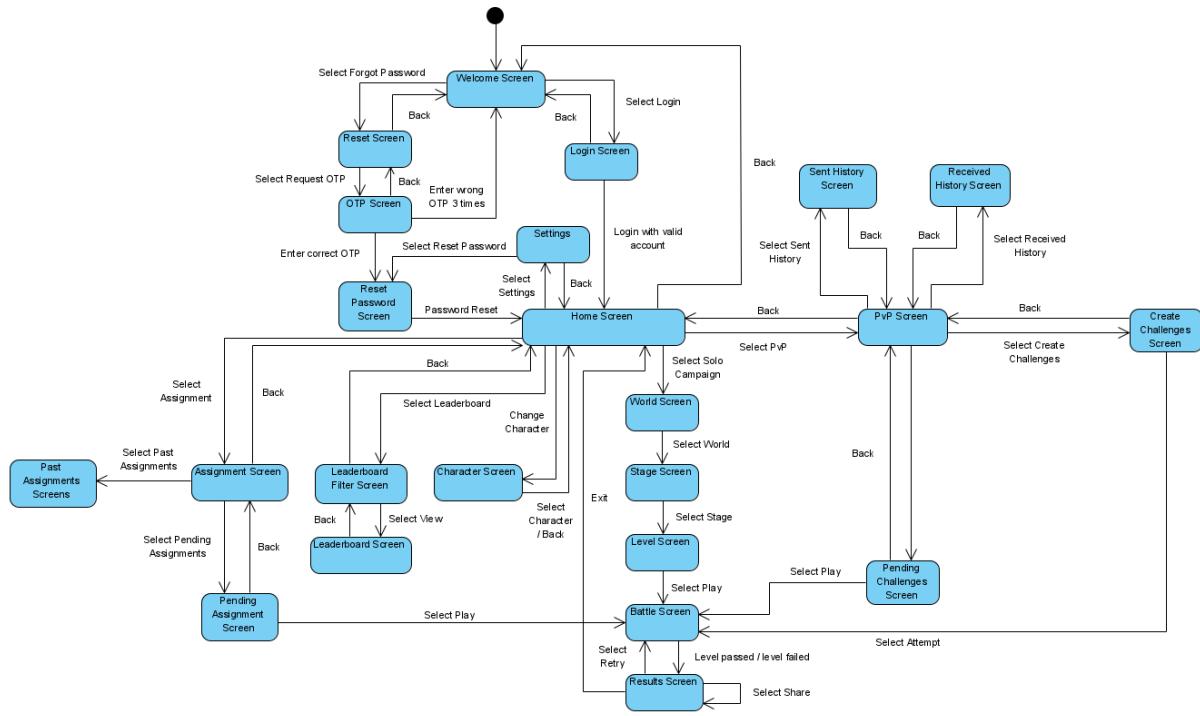
Character screen				
<b>Character</b>	N	N	Y	Y
<b>Ability</b>	Y	Y	Y	Y
<b>Unlocked</b>	N	Y	N	Y
<b>Actions</b>				
<b>Character select</b>	N	N	N	Y

#### 4.8.8 Student Login

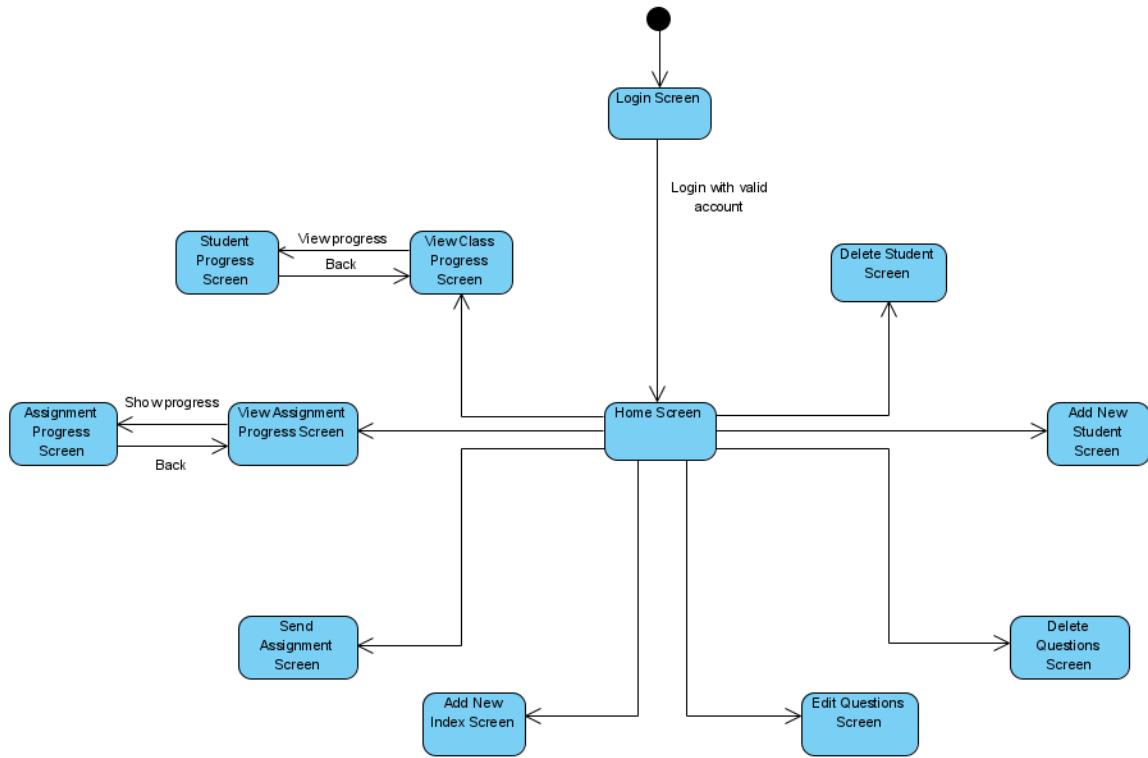
Account management					
<b>Student account</b>	N	N	N	Y	Y
<b>Teacher account</b>	N	Y	Y	Y	Y
<b>Teacher manage student account</b>	N	N	Y	N	Y
Actions					
<b>Teacher edit student account</b>	N	N	N	Y	Y
<b>Teacher creates account</b>	N	Y	Y	N	N

## 4.9 Dialog Map

### 4.9.1 Student:

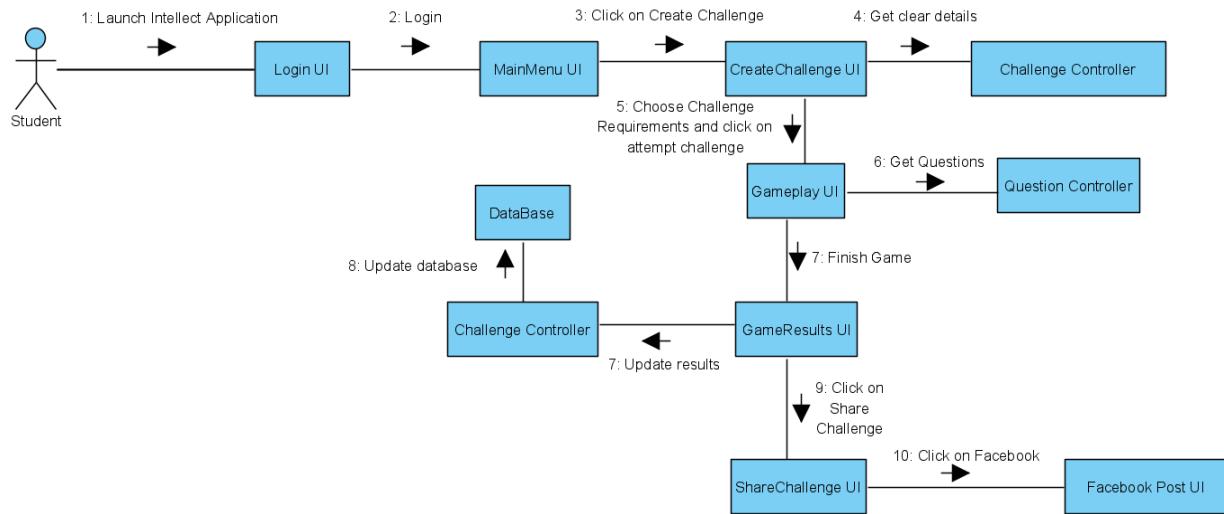


#### 4.9.2 Teacher:

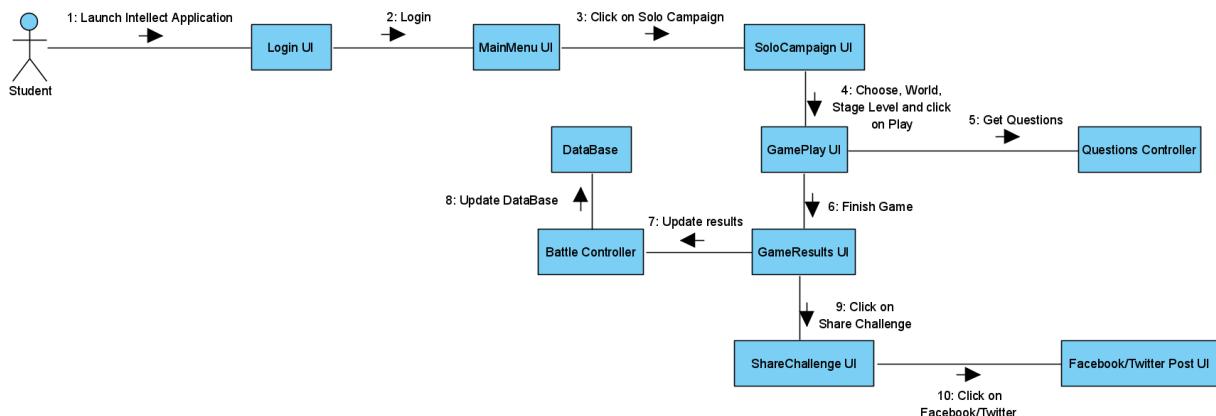


## 4.10 Communication Diagram

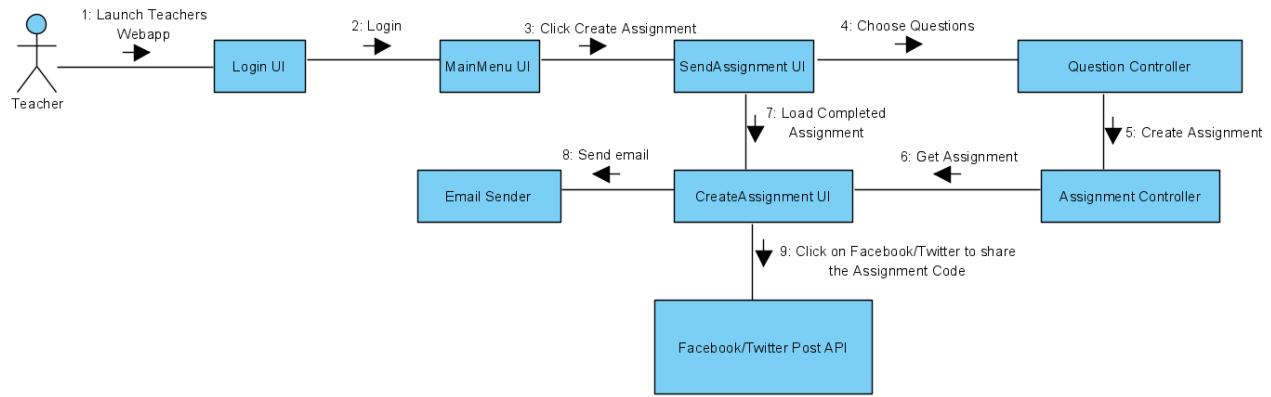
### 4.10.1 Student Creating Challenge



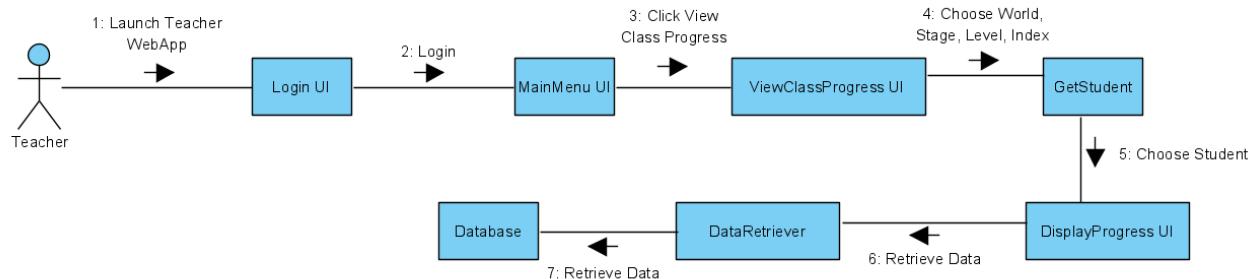
### 4.10.2 Student Attempting Solo Campaign



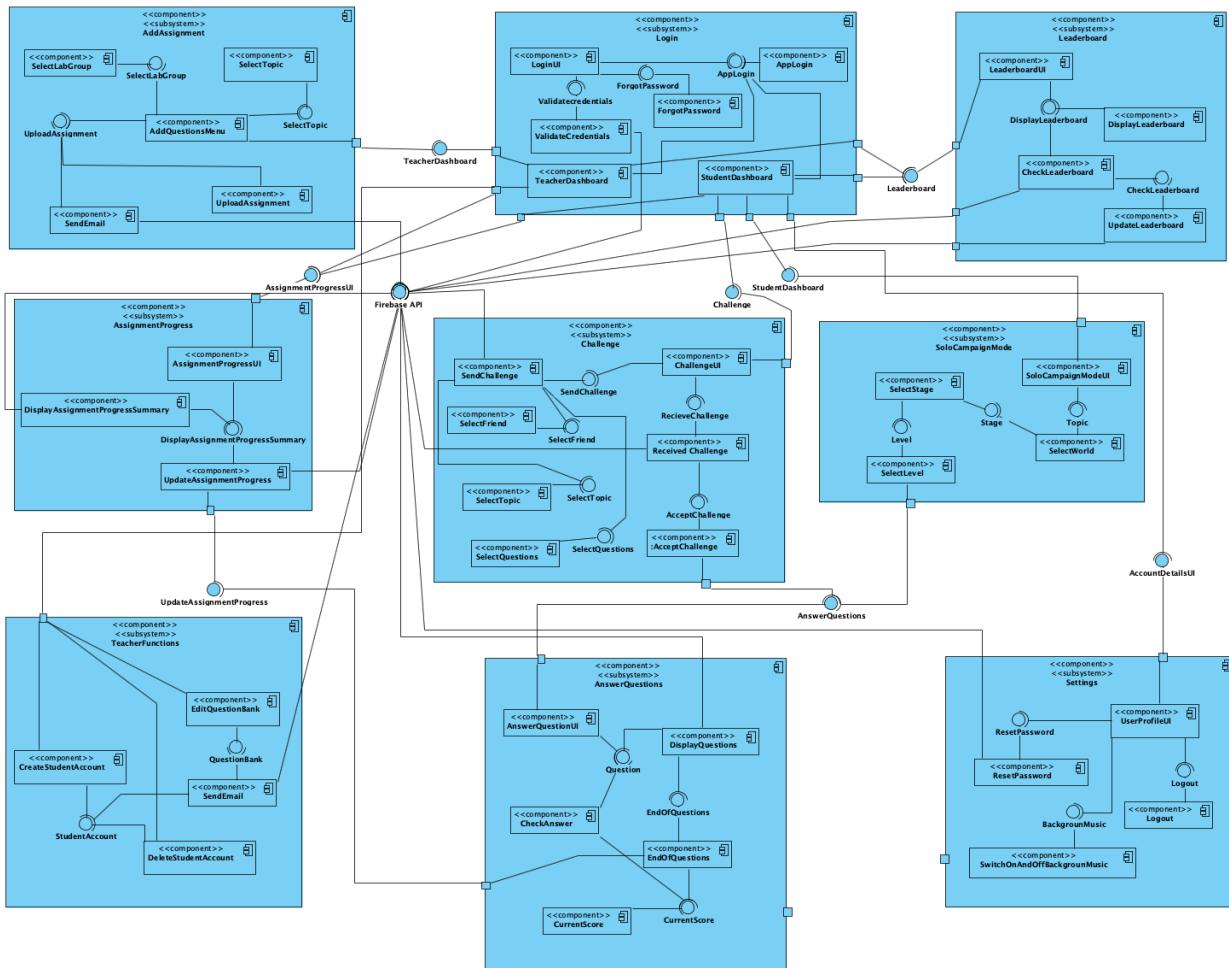
#### 4.10.3 Teacher Creating Assignment



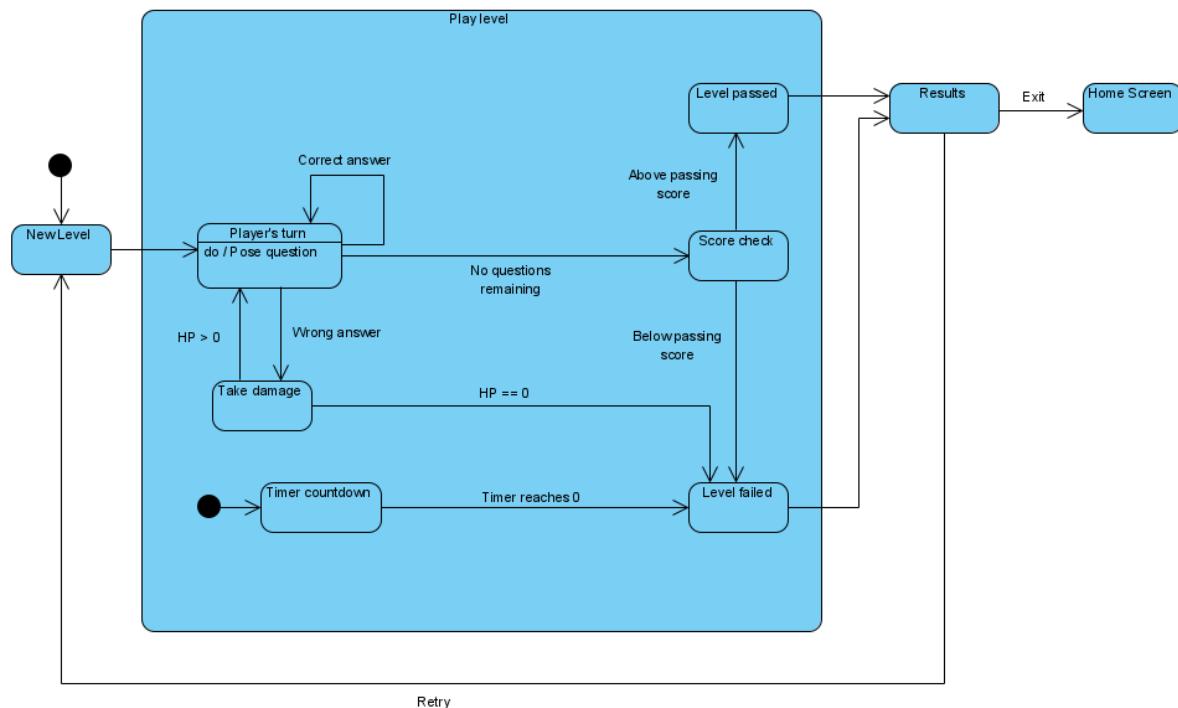
#### 4.10.4 Teacher Checking Progress



## 4.11 Component Diagram



## 4.12 State Transition Diagram



## 4.13 Crudl Matrix

Use Case/Entity	Student	Teacher	Challenge	Solo Campaign Mode	Arcade	Character	Assignment	Question Bank
Login	R	R						
Choose Characters	R,U					R,U		
Choose World	R,U	C,U,D		R	R			
Playing the Game	R		R	R,U	R,U	R,U	R,U	
View Leaderboard	R,L							
Challenge Friend	R,U,D		R					
Complete Assignment	R,U						U	
Arcade Mode	R				R,U			
Add a new student		C,U						
Delete a student		D						
View Student Progress		R		R				
Add Assignment		C,R					R,U	
View Assignment Progress		R					R	
Edit Game		C,U,D						U,D

## **5. Other Nonfunctional Requirements**

### **5.1 Maintainability**

- Editing the game
  - We must be able to change game content easily by changing the question bank or the database.
- Separate systems
  - The intellect system is split into 2 separate components. We must be able to develop both systems independently and faults in each system must not affect the functional requirements of other individual components.

## **6. Sub-System Interface Design**

### **6.1 Intellect Web Application**

```

1 import firebase from "firebase";
2 export async function readWLDData(set_function, setChoice, student, world, stage, level) {
3     const db = firebase.firestore()
4     setChoice(1)
5     var studentData = {}
6     const data = await db.collection("Students").doc(student).collection(world).get()
7     if (data.empty == false) {
8         data.docs.forEach((doc) => {
9             if (doc.id == stage) {
10                 var temp = doc.data()['Levels']
11                 for (var i in temp) {
12                     if (i == level) {
13                         if (temp[i]) {
14                             console.log(temp)
15                             studentData[level] = temp[i]
16                             break;
17                         }
18                     }
19                 })
20             }
21             console.log(studentData)
22         set_function(studentData)
23     }
}

```

#### Function to Retrieve Student Data

The above function retrieves the data of a single student for the given world, stage and level. The parameters for this function are passed from the DisplayProgressStudent.js file which calls this function when the user chooses to view data for an individual student in a particular world, stage and level. The function acts as intermediate between the boundary and entity. It retrieves data from the online Cloud Firestore Database in Firebase by looping through the ‘Students’ collection. It also checks if that level exists or not for that particular student and sends back the data to the boundary respectively. This also follows our layered architecture that we had described earlier.

```

1 import firebase from "firebase";
2 async function readIndexList(set_function){
3     const db=firebase.firestore();
4     const indexList=[]
5     var user=firebase.auth().currentUser;
6     var email=user.email;
7     const data= await db.collection("Teacher").get()
8     data.docs.forEach((doc)=>{
9         if(doc.id==email){
10             var temp=doc.data()
11             set_function(temp["index"])
12         }
13     })
14 }
15
16
17 export default readIndexList;

```

#### Function to Retrieve Class Indexes of a Particular Teacher

The above function is used multiple times in the Web Application as each student is in a particular class index and the teacher can view students that are only in his/her index. This function is called by multiple boundary classes like ‘ClassProgress.js’, ‘ReadStudents.js’ etc. The data is retrieved by looping through the ‘Teachers’ collection in the online Cloud Firestore Database in Firebase. This function ensures reusability and reduces redundancy in the code. Thus, it makes the application much more efficient and plays a crucial role in data retrieval between the boundary and entity.

## 6.2 Intellect Student Game

```

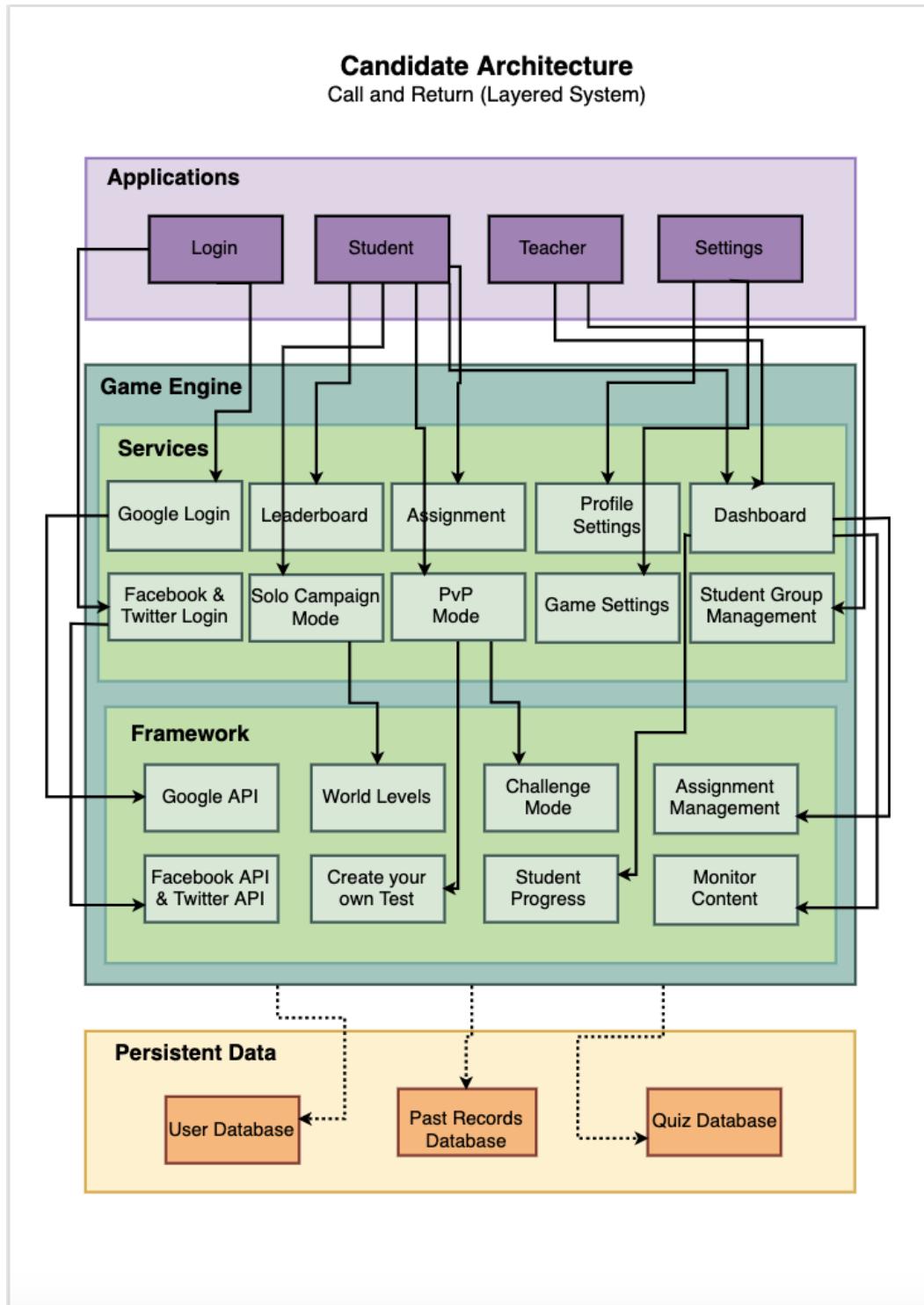
static Future<List<Challenge>> retrievePastReceivedChallenges() async {
  List<Challenge> receivedChallenges = [];
  await for (var snapshot in _firestore
    .collection("Challenges")
    .where("receiver", isEqualTo: UserAccountMgr.studentDetails.username)
    .snapshots()) {
    var documents = snapshot.docs;
    if (documents.isNotEmpty) {
      for (var document in documents) {
        //print(document.data());
        Challenge challenge = Challenge();
        challenge.sender = document["sender"];
        challenge.receiver = document["receiver"];
        print(challenge.receiver);
        challenge.challengeId = document["id"];
        challenge.myScore = document["my_score"];
        challenge.receiverScore = document["opponent_score"];
        challenge.question = document["Question"];
        receivedChallenges.add(challenge);
      }
    }
    break;
  }
  // print(receivedChallenges);
  return receivedChallenges;
}

```

#### Function to Retrieve Past Challenges

In the function `retrievePastReceivedChallenges()` inside the `ChallengeMgr` class, we first initialise an empty array of `Challenge` entity. We then establish a connection to our firestore database and get a list of all those docs who have their attribute "receiver" as the logged in user's username inside the "Challenges" collection. We parse through this retrieved list and initialize all of `Challenge` object attributes as ones found in the document one by one and keep adding them to the list initialised earlier. After finishing parsing, we return this list of challenges objects.

## 7. System Architecture



## **Rationale behind the chosen architecture:**

‘Intellect’ has adapted to the layered subtype of the call-and-return architectural style. This is because the aforementioned architecture segregates the program into departments where each layer looks after its set of functions, processes and controls.

The Game Engine layer of the architectural model represents all the classes which manage the main functionalities of our application including API calls and database management.

The third layer of the architecture - Persistent Data sits below the Game Engine. It comprises three databases - User Database which contains all user information and their respective account settings, Past Records Database which contains all valid attempts of students taking the quiz and lastly the Quiz Database which contains all the questions of the Solo Mode and assignment questions created by Teachers.

The architecture provides for a level of abstraction, since the Services (part of the Game Engine) layer faces the user, and the Framework (also part of the Game Engine) and Persistent Data layers act as the backend of the application.

Low coupling and high cohesion is made possible by MVC (Model / View / Controller). Changes made to a model are propagated to all the related views and controllers using that Model. It also provides great flexibility in terms of adding, removing and updating classes while also providing clarity in terms of dividing the code into smaller logical sections. Future upgrades can be made more easily as relevant changes to the model can be made and required controllers can also be added to the project.

## **8. Testing**

To ensure that our software fulfils the functional requirements defined in the System Requirement Specifications and to assure the customer that the software system will be stable for consumer usage upon releasing it to the market, Software Testing will be conducted as part of our Software Development Process.

All testing scripts for the Intellect Game were conducted using Flutter's testing framework while those for the teacher's web application were conducted using ReactJs test packets.

Various levels of testing were conducted as follows:

- Black Box Testing
- White Box Testing
- Unit Testing
- Integration Testing
- System Testing

### **8.1 Black Box Testing for Teachers**

1. Login
  - a. Generic cases

Test ID	Test Name	Scenario	Expected Result	Actual Result
1	Correct Login Credentials	Login with valid username and password	The system redirects the user to the home screen	The system redirects the user to the home screen

2	Incorrect Login Credentials	Login with invalid username and password	The system displays an error message and prompts the user to re-enter details	The system displays an error message and prompts the user to re-enter details
3	Empty Fields	Login without filling up email address or password	The system prompts the user to fill up the required fields	The system prompts the user to fill up the required fields

b. Specific cases

Email Address	Password	Expected Result	Actual Result
k@gmail.com	123456	Successful Login	Successful Login
Empty("")	123456	Please fill in this field	Please fill in this field

k@gmail.com	Empty("")	Please fill in this field	Please fill in this field
Empty("")	Empty("")	Please fill in this field	Please fill in this field
WrongUser	123456	Failed to log in	Failed to log in

2. Add new student

a. Generic cases

Test ID	Test Name	Scenario	Expected Result	Actual Result
1	Valid details	Providing valid student details to add a new student	The system informs the user that new student has been successfully added	The system informs the user that new student has been successfully added

2	Invalid details	Providing invalid student details to add a new student	The system displays an error message and prompts the user to re-enter details	The system displays an error message and prompts the user to re-enter details
3	Empty Fields	Providing no student details to add a new student	The system prompts the user to fill up the required fields	The system prompts the user to fill up the required fields
4	Re-adding an existing student account	Providing student details that already exist in the database	The system prompts the user to fill up different details	The system prompts the user to fill up different details

b. Specific cases

Assumption : rhea@gmail.com is a valid existing email address.

Assumption: k@gmail.com is an existing student account

Full Name	Email Address	Username	Select Class Index	Expected Result	Actual Result

Rhea Jain	rhea@gmail.com	rhea08	SS1	Successfully created student account	Successfully created student account
Empty("")	rhea@gmail.com	rhea08	SS1	Please fill in this field	Please fill in this field
Rhea Jain	Empty("")	rhea08	SS1	Please fill in this field	Please fill in this field
Rhea Jain	rhea@gmail.com	Empty("")	SS1	Please fill in this field	Please fill in this field
Rhea Jain	rhea@gmail.com	rhea08	Empty("")	Please fill in this field	Please fill in this field
Empty("")	Empty("")	Empty("")	Empty("")	Please fill in this field	Please fill in this field
Rhea Jain	InvalidEmail (eg. rhe08)	rhea08	SS1	Please enter a valid email ID	Please enter a valid email ID

k	k@gmail.com	k	SS1	User already exists, please try again	User already exists, please try again
---	-------------	---	-----	---------------------------------------	---------------------------------------

3. Delete student account

a. Generic cases

Test ID	Test Name	Scenario	Expected Result	Actual Result
1	Valid details	Providing valid student details to delete a student account	The system informs the user that new student account has been successfully deleted	The system informs the user that new student account has been successfully deleted
2	Invalid details	Providing invalid student details to delete a student account	The system displays an error message and prompts the user to re-enter student details	The system displays an error message and prompts the user to re-enter student details

3	Empty Fields	Providing no student details to delete a student account	The system prompts the user to fill up the required fields	The system prompts the user to fill up the required fields
---	--------------	--	--	--

b. Specific cases

Username	Expected Result	Actual Result
rhea08	Student account successfully deleted	Student account successfully deleted
Empty("")	Please fill in this field	Please fill in this field
WrongUser (eg. rheaj11)	Username entered does not exist. Try again.	Username entered does not exist, try again

--	--	--

4. Delete question

a. Generic cases

Test ID	Test Name	Scenario	Expected Result	Actual Result
1	All fields selected	Selecting an option in all fields (World, Stage, Level and Question)	The system informs the user that the question has been successfully deleted	The system informs the user that the question has been successfully deleted
2	All fields unselected	Not selecting an option in all fields (World, Stage, Level)	The system prompts the user to fill up the required fields	The system prompts the user to fill up the required fields
3	'World' field unselected	Not selecting an option in the field	The system prompts the user to fill up the required fields	The system prompts the user to fill up the required fields

4	'Stage' field unselected	Not selecting an option in the field	The system prompts the user to fill up the required fields	The system prompts the user to fill up the required fields
5	'Level' field unselected	Not selecting an option in the field	The system prompts the user to fill up the required fields	The system prompts the user to fill up the required fields
6	'Question' field unselected	Not selecting an option in the field	The system prompts the user to fill up the required fields	The system prompts the user to fill up the required fields

b. Specific Cases

Note: the 'Question' field only becomes available when the other fields are filled in

World	Stage	Level	Question	Expected Result	Actual Result
World 1	Stage 1	Easy	2+2	Question deleted successfully	Question deleted successfully

Empty("")	Empty("")	Empty("")	NULL	Please fill in this field	Please fill in this field
Empty("")	Stage 1	Easy	NULL	Please fill in this field	Please fill in this field
World 1	Empty("")	Easy	NULL	Please fill in this field	Please fill in this field
World 1	Stage 1	Empty("")	NULL	Please fill in this field	Please fill in this field
World 1	Stage 1	Easy	Empty("")	Please fill in this field	Please fill in this field

## 5 Add question

### a. Generic cases

Test ID	Test Name	Scenario	Expected Result	Actual Result
1	All Fields Selected	Selecting an option in all fields	The system informs the user that the question	The system informs the user that the question

			has been successfully added	has been successfully added
2	All fields unselected or empty	Not selecting an option in all fields	The system prompts the user to fill up the required fields	The system prompts the user to fill up the required fields
3	‘World’ field unselected	Not selecting an option in the field	The system prompts the user to fill up the required fields	The system prompts the user to fill up the required fields
4	‘Stage’ field unselected	Not selecting an option in the field	The system prompts the user to fill up the required fields	The system prompts the user to fill up the required fields
5	‘Level’ field unselected	Not selecting an option in the field	The system prompts the user to fill up the required fields	The system prompts the user to fill up the required fields
6	‘Question and options’ field is empty	Not inputting any information	The system prompts the user to fill up the required fields	The system prompts the user to fill up the required fields

b. Specific cases

<b>World</b>	<b>Stage</b>	<b>Level</b>	<b>Question</b>	<b>Expected Result</b>	<b>Actual Result</b>
World 1	Stage 1	Easy	3+8, 11, 12, 10, 13	Question added successfully	Question added successfully
Empty("")	Empty("")	Empty("")	Empty("")	Please fill in this field	Please fill in this field
Empty("")	Stage 1	Easy	3+8, 11, 12, 10, 13	Please fill in this field	Please fill in this field
World 1	Empty("")	Easy	3+8, 11, 12, 10, 13	Please fill in this field	Please fill in this field
World 1	Stage 1	Empty("")	3+8, 11, 12, 10, 13	Please fill in this field	Please fill in this field
World 1	Stage 1	Easy	Empty("")	Please fill in this field	Please fill in this field

6. Send assignment

a. Generic cases

<b>Test ID</b>	<b>Test Name</b>	<b>Scenario</b>	<b>Expected Result</b>	<b>Actual Result</b>
1	All Fields Selected	Selecting an option or inputting information in all fields	The system prompts the user to share the assignment on social media and click on 'Finish'	The system prompts the user to share the assignment on social media and click on 'Finish'
2	All fields unselected or empty	Not selecting an option in all fields or leaving them empty	The system prompts the user to fill up the required fields	The system prompts the user to fill up the required fields
3	'Assignment title' field is empty	Not inputting any information	The system prompts the user to fill up the required fields	The system prompts the user to fill up the required fields
4	'Assignment code' field is empty	Not inputting any information	The system prompts the user to fill up the required fields	The system prompts the user to fill up the required fields

5	'World' field unselected	Not selecting an option in the field	The system prompts the user to fill up the required fields	The system prompts the user to fill up the required fields
6	'Class index' field unselected	Not selecting an option in the field	The system prompts the user to fill up the required fields	The system prompts the user to fill up the required fields

b. Specific cases

Test ID	Test Name	Scenario	Expected Result	Actual Result
1	All Fields Selected	Selecting an option or inputting information in all fields	The system prompts the user to share the assignment on social media and click on 'Finish'	The system prompts the user to share the assignment on social media and click on 'Finish'

2	All fields unselected or empty	Not selecting an option in all fields or leaving them empty	The system prompts the user to fill up the required fields	The system prompts the user to fill up the required fields
3	'Assignment title' field is empty	Not inputting any information	The system prompts the user to fill up the required fields	The system prompts the user to fill up the required fields
4	'Assignment code' field is empty	Not inputting any information	The system prompts the user to fill up the required fields	The system prompts the user to fill up the required fields
5	'World' field unselected	Not selecting an option in the field	The system prompts the user to fill up the required fields	The system prompts the user to fill up the required fields
6	'Class index' field unselected	Not selecting an option in the field	The system prompts the user to fill up the required fields	The system prompts the user to fill up the required fields

<b>Assignment Title</b>	<b>Assignment Code</b>	<b>World</b>	<b>Class Index</b>	<b>Expected Result</b>	<b>Actual Result</b>
Discrete Maths	MH1812	World 1	SS1	Assignment sent successfully	Assignment sent successfully
Empty("")	Empty("")	Empty("")	Empty("")	Please fill in this field	Please fill in this field
Empty("")	MH1812	World 1	SS1	Please fill in this field	Please fill in this field
Discrete Maths	Empty("")	World 1	SS1	Please fill in this field	Please fill in this field
Discrete Maths	MH1812	Empty("")	SS1	Please fill in this field	Please fill in this field
Discrete Maths	MH1812	World 1	Empty("")	Please fill in this field	Please fill in this field

## 7. View Class Progress

### a. Generic cases

<b>Test ID</b>	<b>Test Name</b>	<b>Scenario</b>	<b>Expected Result</b>	<b>Actual Result</b>
1	All Fields Selected	Selecting an option in all fields	The system displays the progress of one or all students.	The system displays the progress of one or all students.
2	All fields unselected	Not selecting an option in all fields	The system prompts the user to fill up the required fields	The system prompts the user to fill up the required fields
3	‘World’ field unselected	Not selecting an option in the field	The system prompts the user to fill up the required fields	The system prompts the user to fill up the required fields
4	‘Stage’ field unselected	Not selecting an option in the field	The system prompts the user to fill up the required fields	The system prompts the user to fill up the required fields
5	‘Level’ field unselected	Not selecting an option in the field	The system prompts the user to fill up the required fields	The system prompts the user to fill up the required fields

6	'Index' field is empty	Not selecting an option in the field	The system prompts the user to fill up the required fields	The system prompts the user to fill up the required fields
7	'Students' field is empty	Not selecting an option in the field	The system prompts the user to fill up the required fields	The system prompts the user to fill up the required fields
8	'Index' selected has no students enrolled in it	No student option displayed to select from	NULL results	NULL results

a. Specific cases

Assumption : Index SS4 has no students enrolled in it

World	Stage	Level	Class Index	Students	Expected Result	Actual Result
World 1	Stage 1	World 1	SS1	rhea08	Student progress displayed	Student progress displayed

Empty("")	Empty("")	Empty("")	Empty("")	NULL	Please fill in this field	Please fill in this field
Empty("")	Stage 1	World 1	SS1	NULL	Please fill in this field	Please fill in this field
World 1	Empty("")	World 1	SS1	NULL	Please fill in this field	Please fill in this field
World 1	Stage 1	Empty("")	SS1	NULL	Please fill in this field	Please fill in this field
World 1	Stage 1	World 1	Empty("")	NULL	Please fill in this field	Please fill in this field
World 1	Stage 1	World 1	SS4	NULL	NULL	NULL

8. View assignment progress

a. Generic cases

Test ID	Test Name	Scenario	Expected Result	Actual Result

1	Option Selected	Selecting an option in the Assignment Field	The system displays the progress students have made on the selected assignment	The system displays the progress students have made on the selected assignment
2	Empty field	Not selecting an option	The system prompts the user to fill up the required field	The system prompts the user to fill up the required field
3	‘Assignment’ selected has not been attempted	No details found in database as assignment has not been attempted	NULL results	NULL results

b. Specific cases

Assumption: Assignment titled, ‘Discrete Maths’ has not been attempted by any student

Assignment	Expected Result	Actual Result

Software Engineering	Assignment progress displayed	Assignment progress displayed
Empty("")	Please fill in this field	Please fill in this field
Discrete Maths	NULL	NULL

9. Add new index

a. Generic cases

Test ID	Test Name	Scenario	Expected Result	Actual Result
1	Valid details	Providing valid index details	The system informs the user that new index has	The system informs the user that new index has

			been successfully added	been successfully added
2	Invalid details	Providing invalid index details	The system displays an error message and prompts the user to re-enter details	The system displays an error message and prompts the user to re-enter details
3	Empty Fields	Providing no index details	The system prompts the user to fill up the required fields	The system prompts the user to fill up the required fields
4	Re-adding an existing student account	Providing index details that already exist in the database	The system prompts the user to fill up different details	The system prompts the user to fill up different details

b. Specific cases

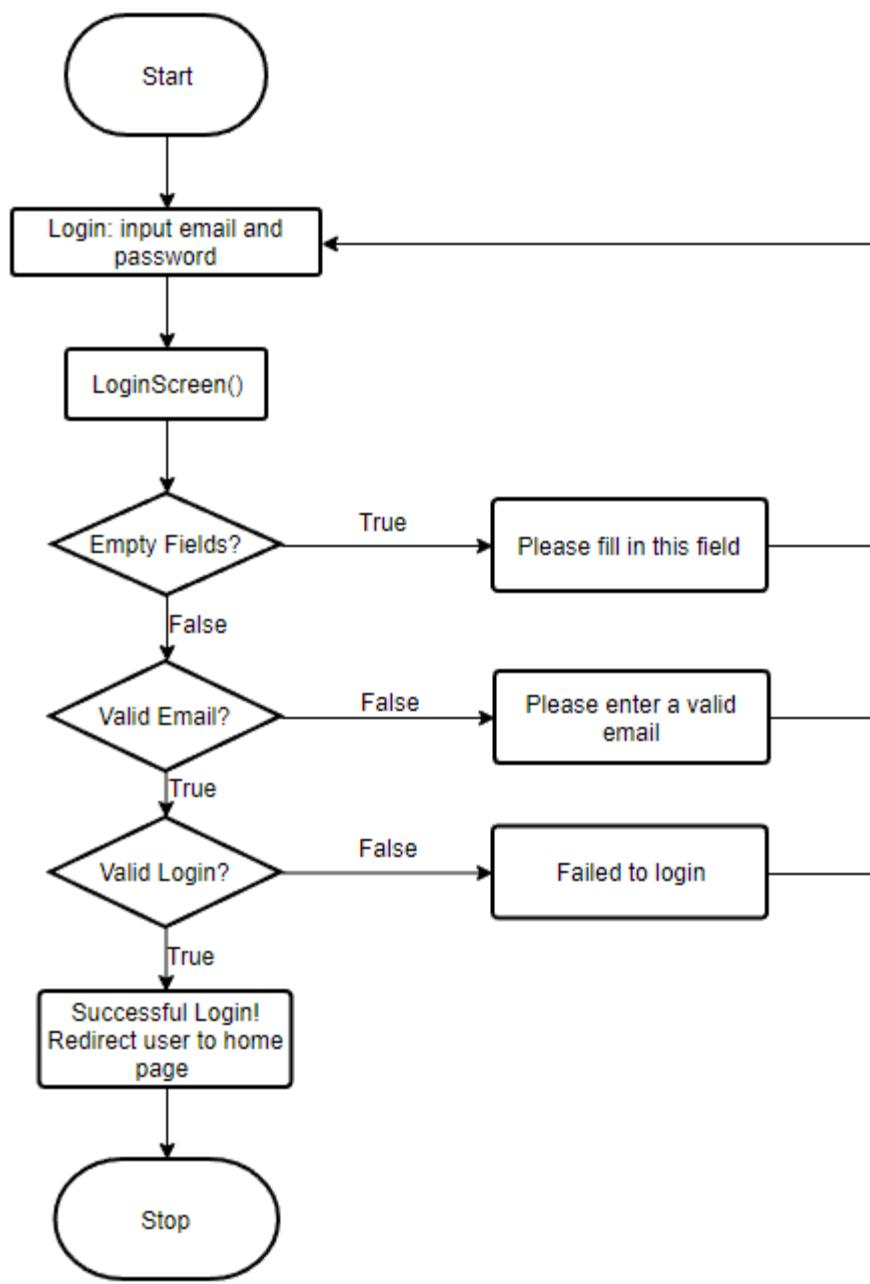
Assumption: Index SS1 already exists in the database

Assumption: Index SS5 does not already exist in the database

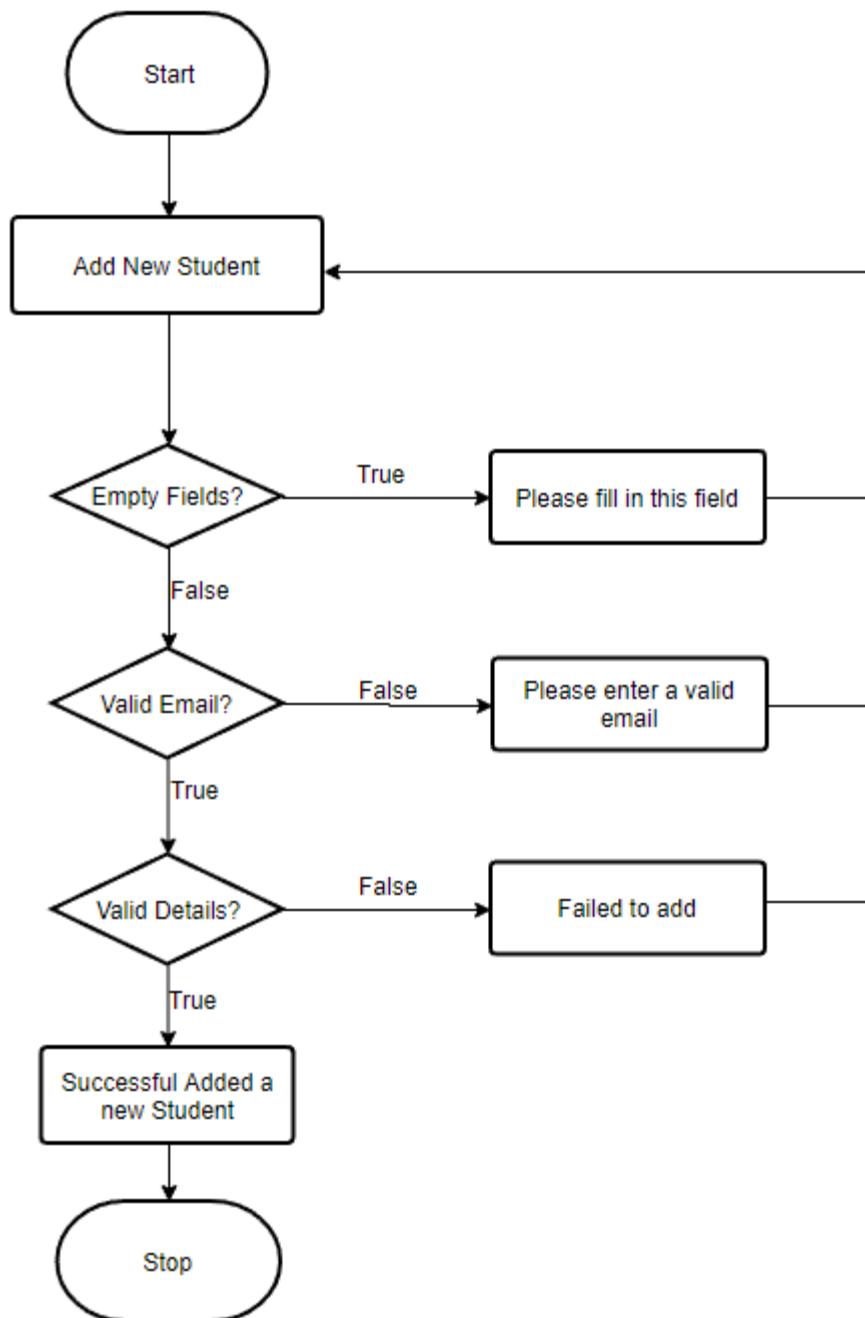
<b>Index</b>	<b>Expected Result</b>	<b>Actual Result</b>
SS5	Student account successfully deleted	Student account successfully deleted
Empty("")	Please fill in this field	Please fill in this field
WrongUser (eg. @!?#)	Please enter valid details and try again	Please enter valid details and try again
SS1	Index already exists please try again	Index already exists please try again

## 8.2 White Box Testing for Teachers

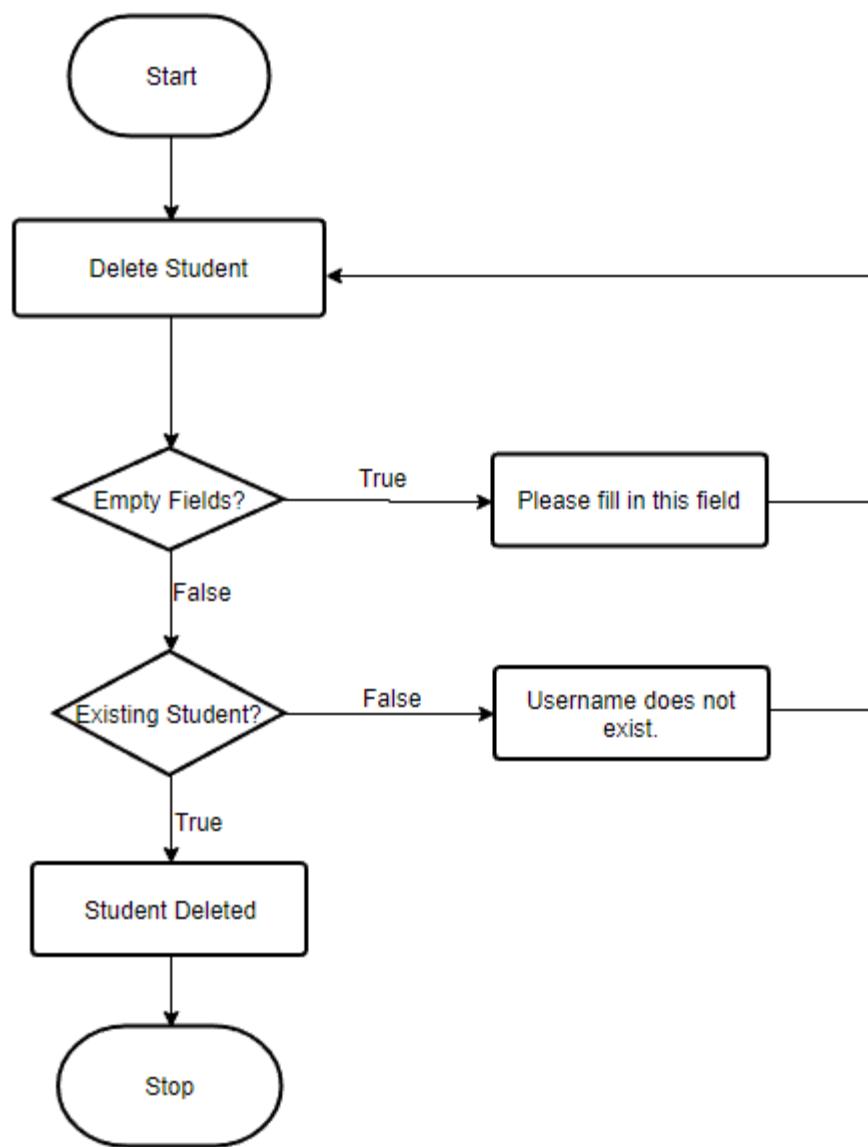
### 7.2.1 Teacher's Login



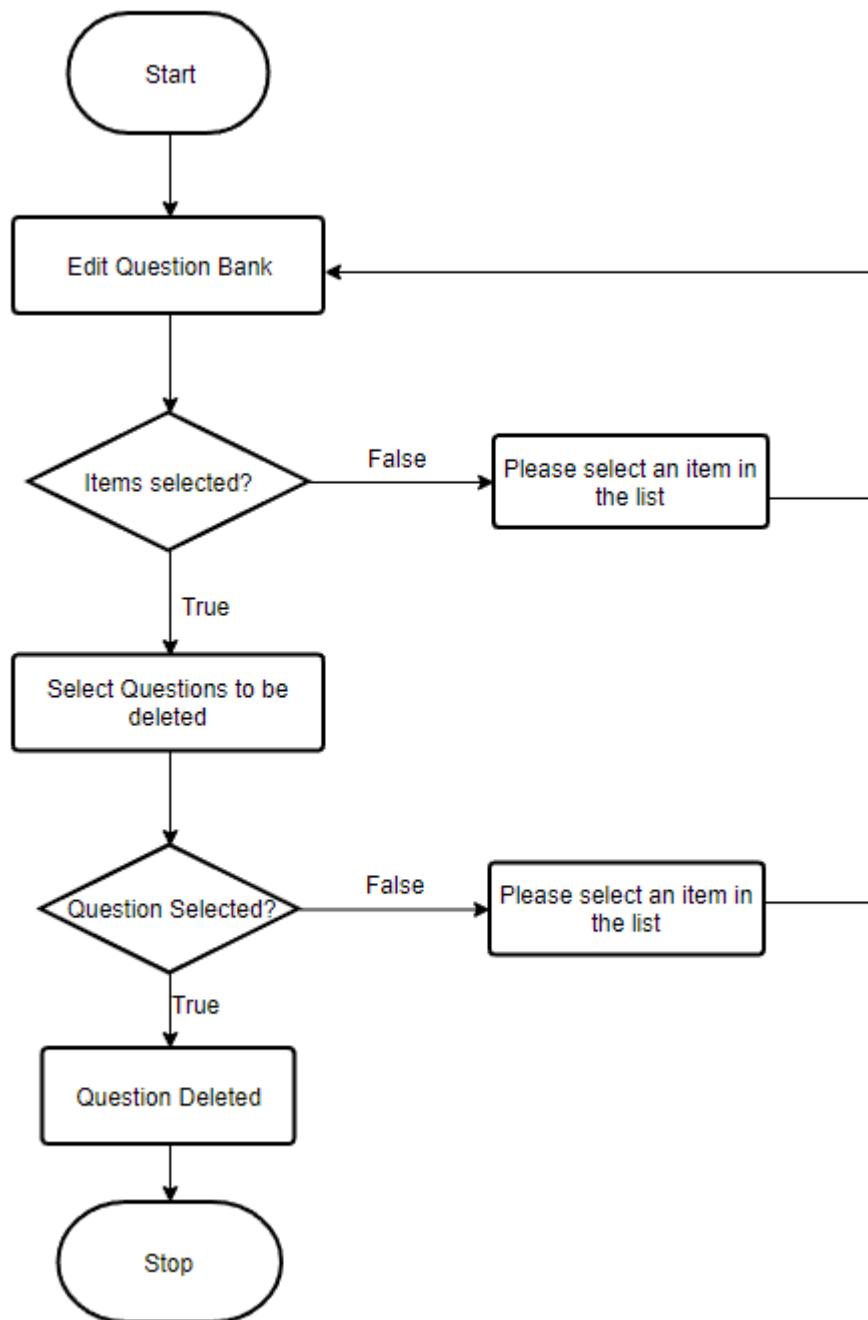
### 7.2.2. Teacher Add New Student



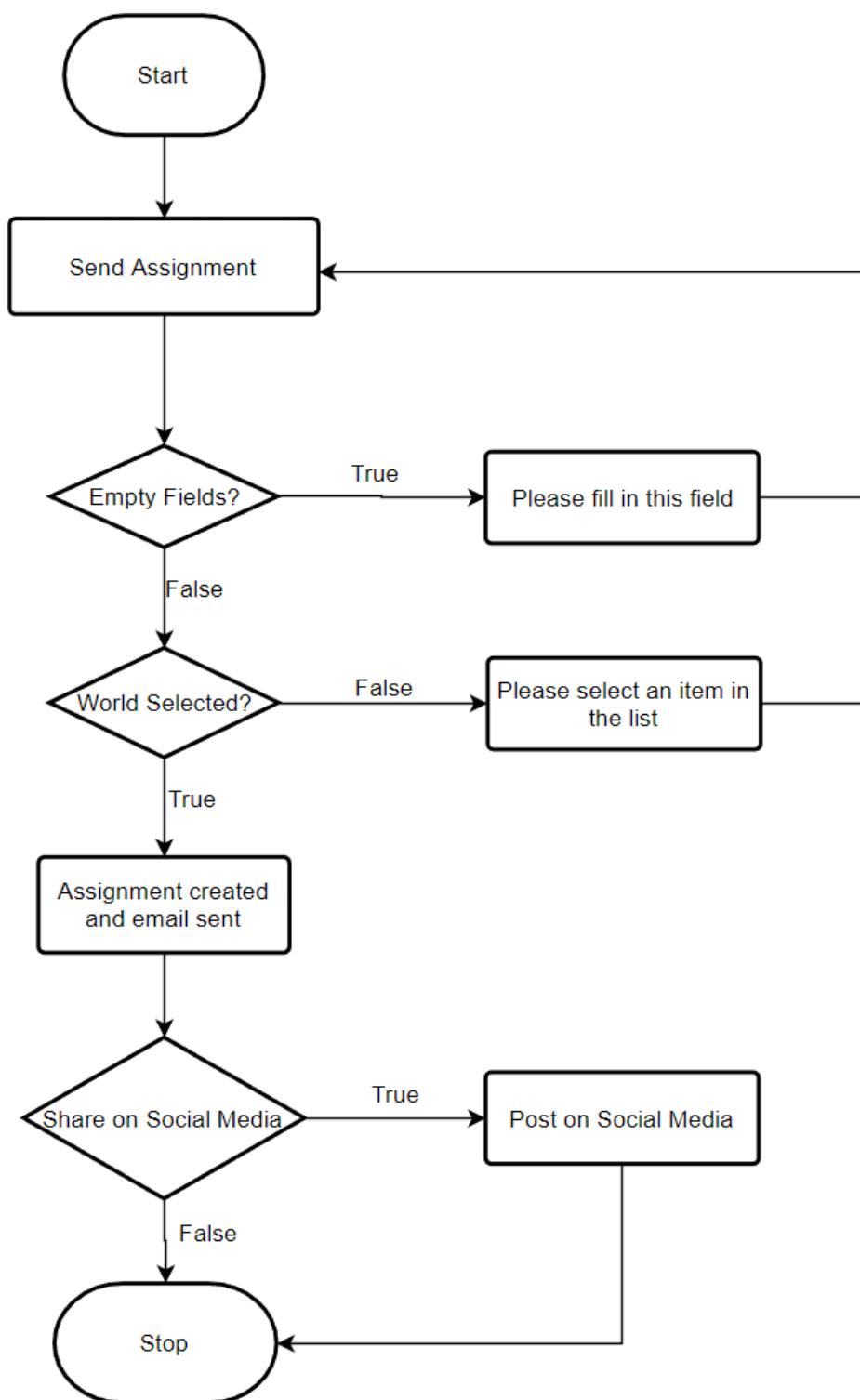
### 7.2.3. Teacher Delete Student



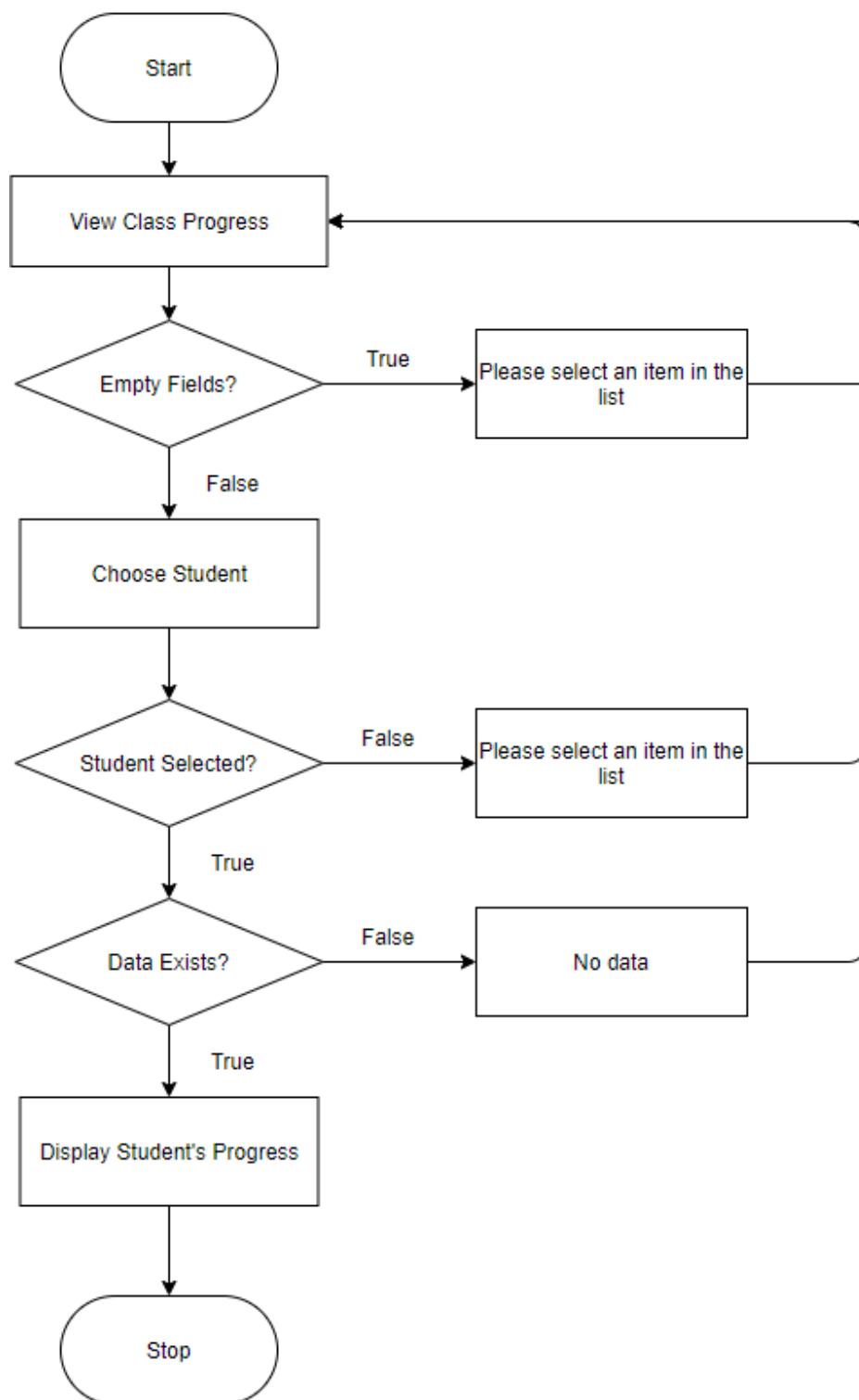
#### 7.2.4. Teacher Delete Question



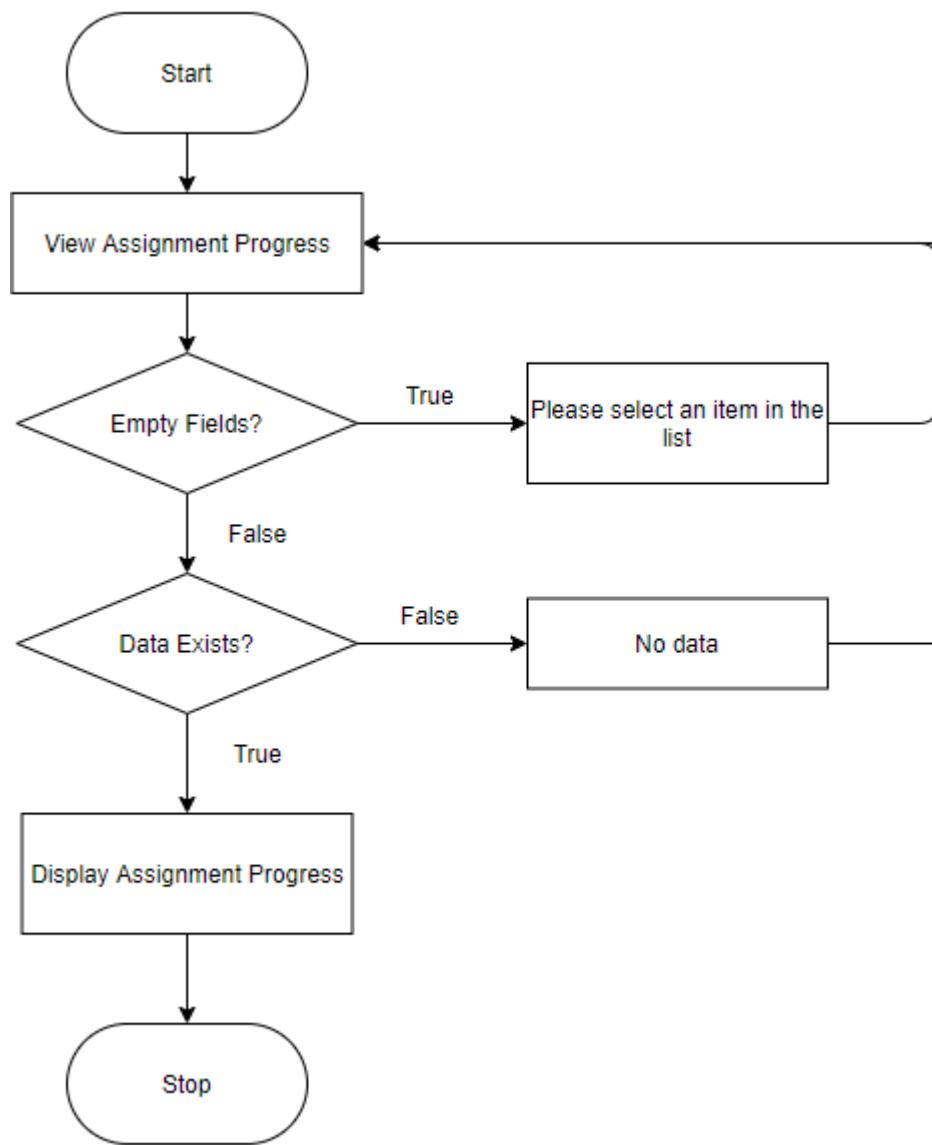
#### 7.2.5. Teacher Send Assignment



#### 7.2.6. Teacher view Class Progress



#### 7.2.7. Teacher view Assignment Progress



### 8.3 Unit Testing for Teachers

Unit testing, also known as component testing, is meant to ensure that each component or function will operate accurately as it is intended to. Individual classes will be tested to ensure reliability and functionality within a unit-level. White box testing is conducted at this phase. Since the team has developed the codes for the software, we have full knowledge of the fields necessary fields for each system. By studying the implemented code, we will determine all legal (valid and invalid) and illegal inputs and verify the outputs against the expected outcomes.

For the Intellect Web Application, we have used the ReactJs testing framework to implement unit testing. We have written scripts within ReactJs to test the inner workings of the application such as reading questions, adding questions, reading students from an index, reading all indexes and checking student progress. This type of testing has been carried out to ensure that each unit of our software is working as expected.

```
▽ _test_
  JS addQuestion.test.js
  JS checkProgress.test.js
  JS index.test.js
  JS readQuestions.test.js
  JS readStudents.test.js
```

```
PASS  src/_test_/readQuestions.test.js
PASS  src/_test_/checkProgress.test.js
PASS  src/_test_/addQuestion.test.js
PASS  src/_test_/index.test.js
PASS  src/_test_/readStudents.test.js

Test Suites: 5 passed, 5 total
Tests:      5 passed, 5 total
Snapshots:  0 total
Time:       4.072 s
Ran all test suites.
```

Following is an example of one of the unit tests written to check if a question is added to the firestore database.

```
1 import axios from "axios";
2 test("Check Add Question", async()=>[
3     var k= await axios.get("http://localhost:8000/api/addQuestion")
4     var difference=k.data.data;
5     expect (difference).toBe(1);
6 ]);
7
7 router.get("/addQuestion", async (req, res) => {
8     const db = app1.firebaseio();
9     const doc = await db.collection("World 3").doc("Stage 3").get();
10    var k = doc.data();
11    var len_initial = k["Hard"].length;
12    k["Hard"].push({
13        Answer: "0",
14        Question: "0 + 0 - 0 x 0",
15        Options: ["0", "1", "2", "3"],
16    });
17    var len_final = k["Hard"].length;
18    const userRef = db
19        .collection("World 3")
20        .doc("Stage 3")
21        .update({ Easy: k["Easy"], Medium: k["Medium"], Hard: k["Hard"] });
22    res.json({ data: len_final - len_initial });
23});
```

## 8.4 Black Box Testing for Students

1. Login
  - a. Generic Cases

Test ID	Test Name	Scenario	Expected Result	Actual Result
1.	Correct Username and Correct Password	Login with valid username and password	The system redirects user to homescreen	The system redirects user to homescreen
2.	Correct Username and Incorrect Password	Login with valid username and	An error message pops up, prompting	An error message pops up, prompting the user to

		invalid password	the user to re-enter their info	re-enter their info
3.	Incorrect Username and existing Password	Login with invalid username and valid password	An error message pops up, prompting the user to re-enter their info	An error message pops up, prompting the user to re-enter their info
4.	Incorrect Username and Incorrect Password	Login with invalid username and password	An error message pops up, prompting the user to re-enter their info	An error message pops up, prompting the user to re-enter their info
5.	Empty Fields	Login without filling up any fields	The System will prompt the user to enter their details	The System will prompt the user to enter their details

b. Specific Cases

Username	Password	Expected Result	Actual Result
agarwal	1234	Successful Login	Successful Login
agarwal	12345	Invalid Username/Password	Invalid Username/Password
agarwal0	1234	Invalid Username/Password	Invalid Username/Password

agarwal0	12345	Invalid Username/ Password	Invalid Username/ Password
agarwal	(Leave Empty)	Please fill in all fields	Please fill in all fields
(Leave Empty)	12345	Please fill in all fields	Please fill in all fields
(Leave Empty)	(Leave Empty)	Please fill in all fields	Please fill in all fields

## 2. OTP

### a. General Case

Test ID	Test Name	Scenario	Expected Result	Actual Result
1.	Valid email OTP	Enter an email that exists in the system	OTP sent to email listed	OTP sent to email listed
2.	Invalid email OTP	Enter an email that does not exist in the system	No existing email message	No existing email message
3.	Empty Fields	Leave the email field empty	The System will prompt the user to enter their details	The System will prompt the user to enter their details
4.	Valid OTP	Enter the OTP sent to the email address	The System will allow the user to reset the password	The System will allow the user to reset the password
5.	Invalid OTP with less than 3 consecutive invalid entries	Enter something that is not the OTP sent to the email address with less than 3 consecutive	An error message pops up, prompting the user to re-enter the OTP	An error message pops up, prompting the user to re-enter the OTP

		invalid entries		
6.	Empty Fields	Select Enter OTP without filling in the OTP field	The System will prompt the user to enter the OTP	The System will prompt the user to enter the OTP
7.	Invalid OTP with less than 3 consecutive invalid entries	Enter something that is not the OTP sent to the email address with 3 consecutive invalid entries	An error message pops up and boots the user to the homescreen	An error message pops up and boots the user to the homescreen

b. Specific Case

Email	Expected Result	Actual Result
agarwal@mail.com	OTP Successfully sent	OTP Successfully sent
wrong@mail.com	No email found	No email found
654321	Invalid OTP	Invalid OTP

3. Battle System

a. General Case

Test ID	Test Name	Scenario	Expected Result	Actual Result
1.	Select Solo campaign	Select the Solo campaign button on the main page	Reach the Select World Page	Reach the Select World Page
2.	Select an Unlocked	Choosing a world,	Reach the Select	Reach the Select Stage,

	world, stage or difficulty	stage or difficulty that is unlocked	Stage, Select Difficulty or Battle Page respectively	Select Difficulty or Battle Page respectively
3.	Select a Locked world, stage or difficulty	Choosing a world, stage or difficulty that is locked	The respective buttons are greyed out	The respective buttons are greyed out
4.	Selecting a correct answer	Select a correct answer	Number of points increases, question counter increases, next question appears and option gets highlighted in green	Number of points increases, question counter increases, next question appears and option gets highlighted in green
5.	Selecting a wrong answer	Select a wrong answer	Health decreases, question counter increases, next question appears and option gets highlighted in red	Health decreases, question counter increases, next question appears and option gets highlighted in red
6.	Finishing a level with more than 5 questions correct	Finish a level with more than 5 questions correct	Result Screen appears with Level Passed on it and next level is unlocked	Result Screen appears with Level Passed on it and next level is unlocked
7.	Finishing a level with less than 5 questions correct	Finish a level with less than 5 questions correct	Result Screen appears with Level Failed on it. If level is previously passed, it will remain passed	Result Screen appears with Level Failed on it. If level is previously locked, it will remain locked

			locked, it will remain locked	
8.	The user does not answer all questions before the timer runs out	Let the timer run out	Result Screen should appear with either Level Passed or Failed on it	Result Screen should appear with either Level Passed or Failed on it
9.	Select the Share button on the results screen	Select the Share button on the results screen	Score is posted onto the User's Social Media	Score is posted onto the User's Social Media
10.	Select the Retry button on the results screen	Select the retry Button on the results screen	Restarts the level the results page is on	Restarts the level the results page is on
11.	Select the Exit button on the results screen	Select the exit Button on the results screen	Returns the user to the main page	Returns the user to the main page
12.	Select the next level button on the results screen	Select the next level button on the results screen	Starts the next level for the user	Starts the next level for the user

b. Specific Case

Question	Answer selected	Result
3+5	8	Number of points increases, question counter increases, next question appears and option gets highlighted in green
3+5	13	Health decreases, question counter increases, next question appears and option gets highlighted in red

#### 4. Leaderboard

##### a. General Case

Test ID	Test Name	Scenario	Expected Result	Actual Result
1.	Pressing Leaderboard button	Select the Leaderboard button on the main page	Reach the Leaderboard Page	Reach the Leaderboard Page
2.	Pressing View button	Select the corresponding world, stage, level and difficulty and get the Leaderboard for that combination	Reach the corresponding Leaderboard result page	Reach the corresponding Leaderboard result page
3.	Checking viewable options	Users should only be able to view leaderboards for worlds, levels and stages they already cleared	Only be able to view leaderboards for worlds, levels and stages they already cleared	Only be able to view leaderboards for worlds, levels and stages they already cleared

##### b. Specific case

World	Stage	Level	Result
World 1	Stage 1	Easy	Leaderboard sorted according to the

			mentioned parameters
--	--	--	----------------------

## 5. Assignment

### a. General Case

Test ID	Test Name	Scenario	Expected Result	Actual Result
1.	Pressing Assignment button	Select the Assignments button on the main page	Reach the Assignments Page	Reach the Assignments Page
2.	Pressing Pending Assignment button	Select the Pending Assignments button on the assignments page	Reach the Pending Assignments Page with a table containing 2 options	Reach the Pending Assignments Page with a table containing 2 options
3.	Pressing the Attempt Assignment button	Select the Attempt Assignments button on the pending assignments page	Reach the Battle Screen for the Assignment	Reach the Battle Screen for the Assignment
4.	Pressing the Past Assignment button	Select the Past Assignments button on the pending assignments page	Displays all past assignments done by the user	Displays all past assignments done by the user
5.	Checking button Functionality	Select the Past Assignments button on the assignments page	Reach the Past Assignments Page	Reach the Past Assignments Page

## 6. Challenges

### a. General case

Test ID	Test Name	Scenario	Expected Result	Actual Result
1.	Select Challenge mode	Select the Challenge button on the main page	Reach the Select World Page	Reach the Select World Page
2.	Select Create Challenge	Select the Create Challenge button on the Challenge Friends page	Reach the attribute page for the challenge	Reach the attribute page for the challenge
3.	Select an Unlocked world, stage or difficulty	Choosing a world, stage or difficulty that is unlocked	Reach the Select Stage, Select Difficulty or Battle Page respectively	Reach the Select Stage, Select Difficulty or Battle Page respectively
4.	Ensure that locked worlds, stages or difficulties do not appear	Locked worlds, stages or difficulties do not appear	Locked worlds, stages or difficulties do not appear	Locked worlds, stages or difficulties do not appear
5.	Select an Unlocked world, stage or difficulty	Choosing a world, stage or difficulty that is locked	The respective buttons are greyed out	The respective buttons are greyed out
6.	Select a user	Choosing a user from the options	User chosen is selected as the target of the challenge to be issued to.	User chosen is selected as the target of the challenge to be issued to.

7.	Ensure that only users in your class index appears	Only users in your class index appears	Only users in your class index appears	Only users in your class index appears
8.	Selecting a correct answer	Select a correct answer	Number of points increases, question counter increases and next question appears	Number of points increases, question counter increases and next question appears
9.	Selecting a wrong answer	Select a wrong answer	Health decreases, question counter increases and next question appears	Health decreases, question counter increases and next question appears
10.	Time running out	Let the timer run out	Score is saved and sent to the user being challenged	Score is saved and sent to the user being challenged
11.	Select Send Challenge on results screen	Select Send Challenge on results screen	Challenge is sent to selected user	Challenge is sent to selected user
12.	Select Pending Challenges	Select Pending Challenges	Reach the Pending Challenges page	Reach the Pending Challenges page
13.	Select Past Sent Challenges	Select Past Sent Challenges	Views all challenges that the user sent	Views all challenges that the user sent
14.	Select Past Received Challenges	Select Past Received Challenges	Views all challenges that the user received	Views all challenges that the user received

b. Specific case

World	Stage	Level	Student	result
World 1	Stage 1	Easy	khush_156	Challenge issue user proceeds to attempt questions.

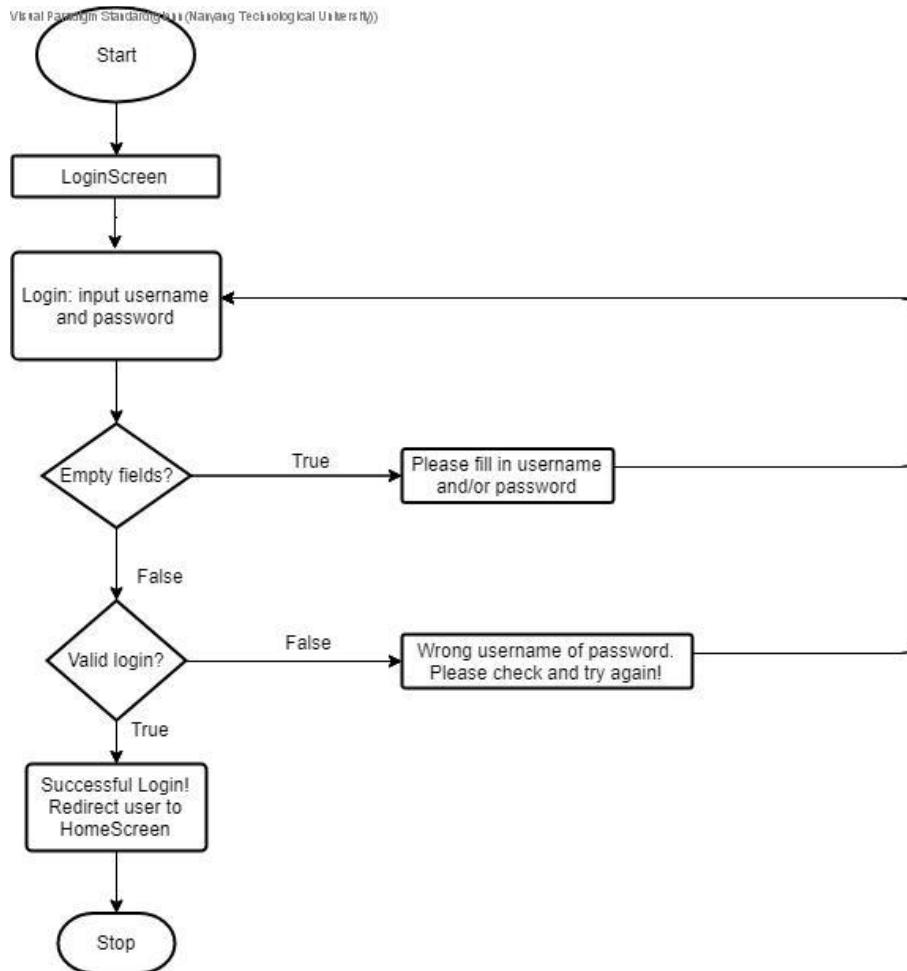
## 7. Characters

### a. General Case

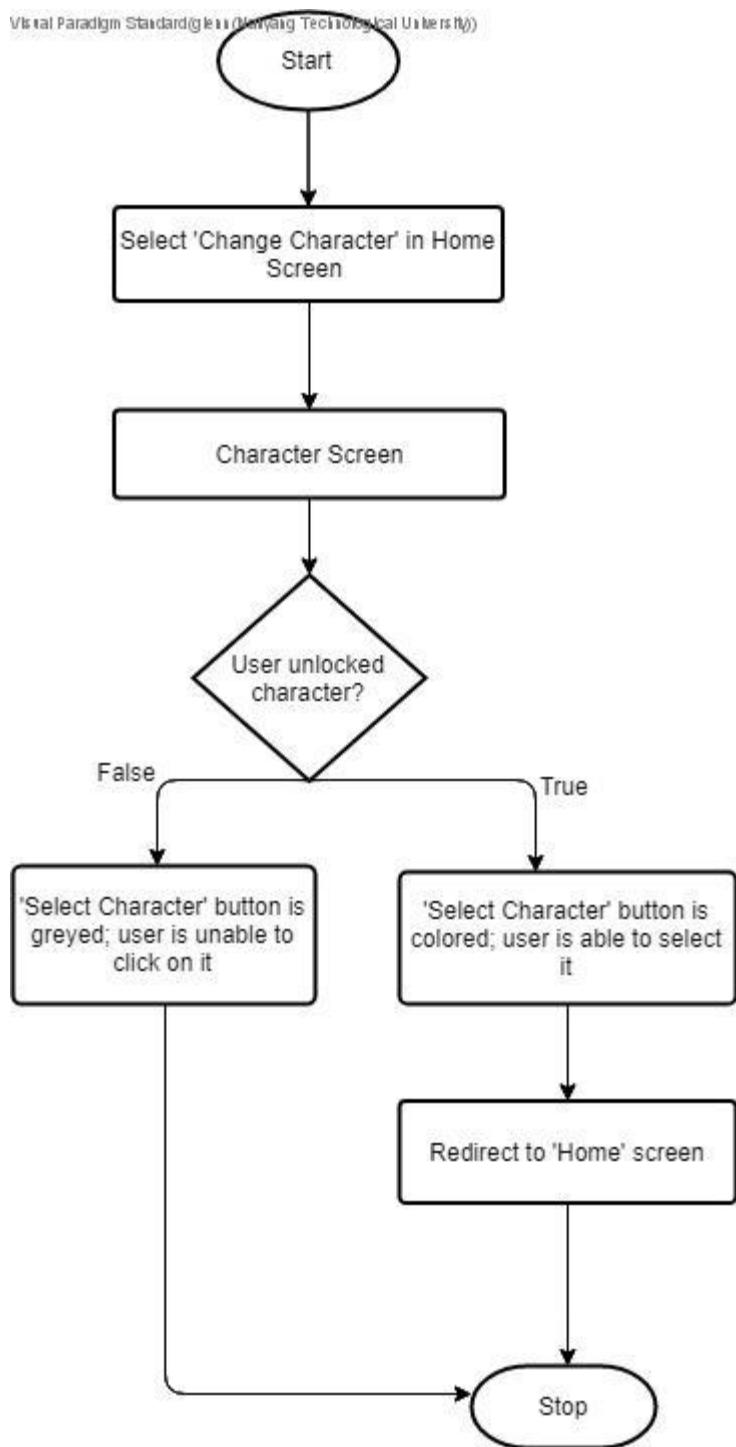
Test ID	Test Name	Scenario	Expected Result	Actual Result
1.	Clicking the Select Character Button	Clicking the Select Character Button	Reach the Select Character Screen	Reach the Select Character Screen
2.	Select an already selected character	Select the same character as the one that is currently selected	Return to the main page with current character still selected	Return to the main page with current character still selected
3.	Select an Unlocked character	Select a character that is unlocked	Return to the main page with selected character selected	Return to the main page with selected character selected
	Select a Locked character	Select a character that is locked	Select Character button should be greyed out	Select Character button should be greyed out

## **8.5 White Box Testing for Students**

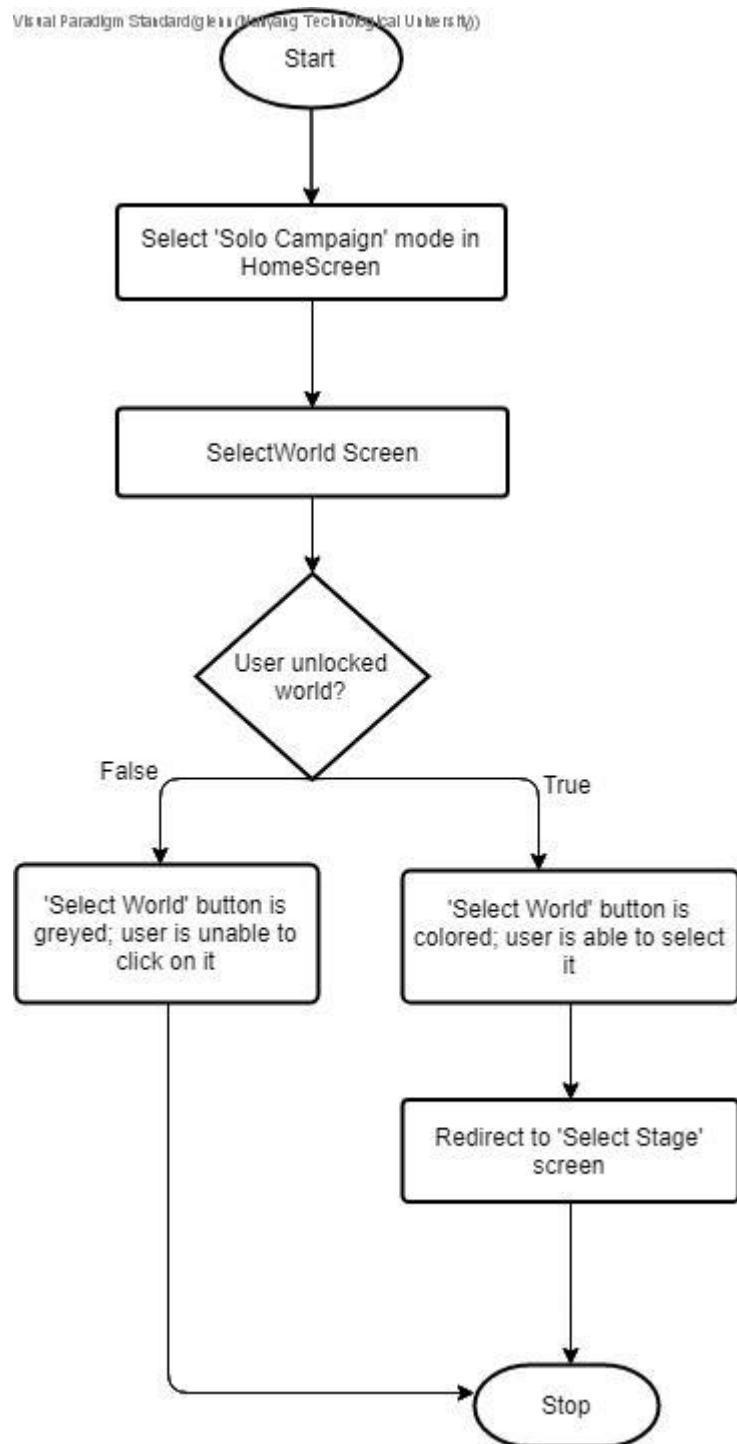
### **7.4.1 Student Login**



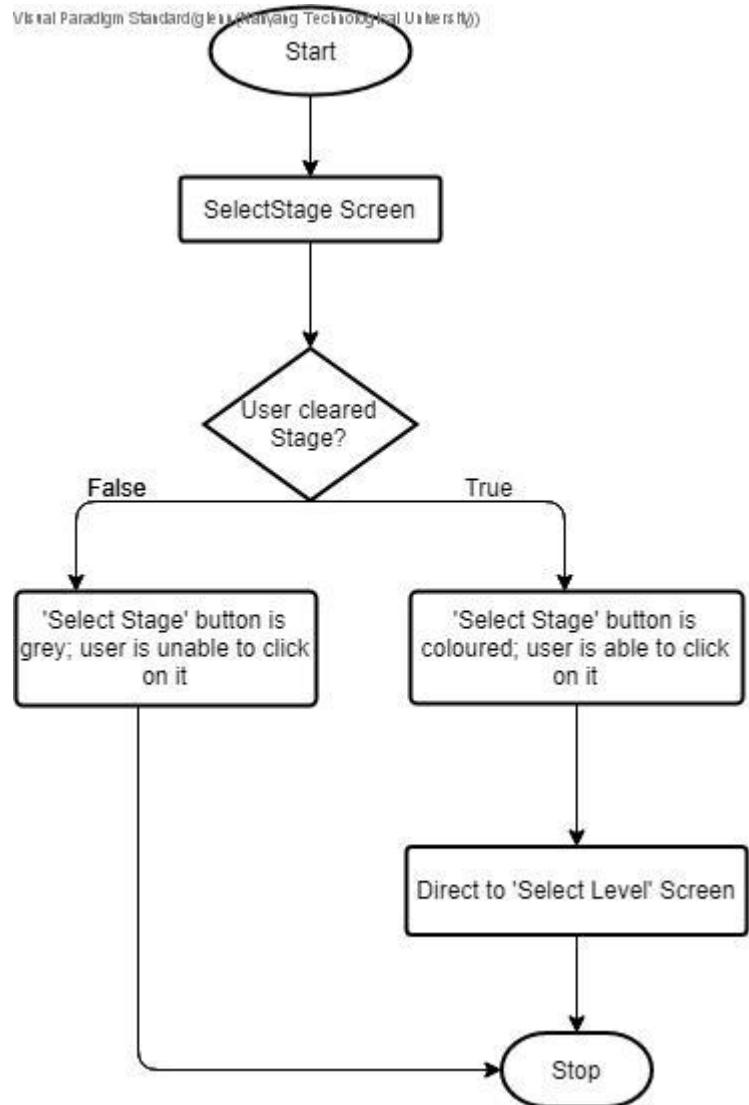
#### 7.4.2. Select Character



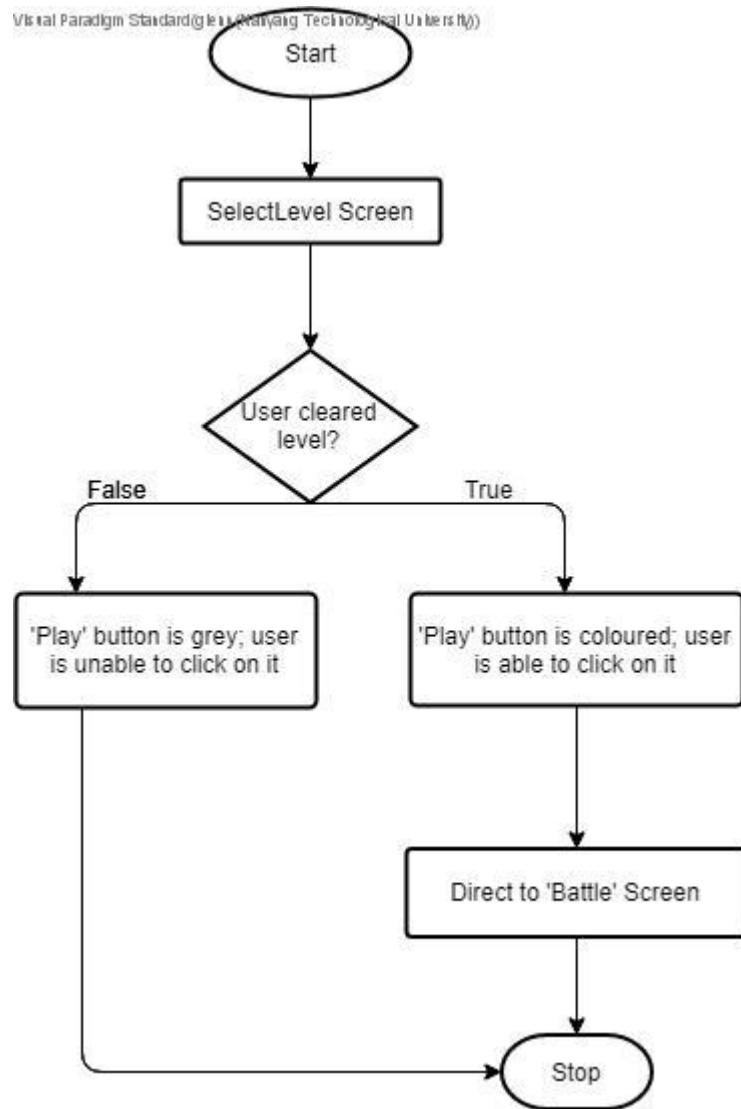
### 7.4.3. Select a World



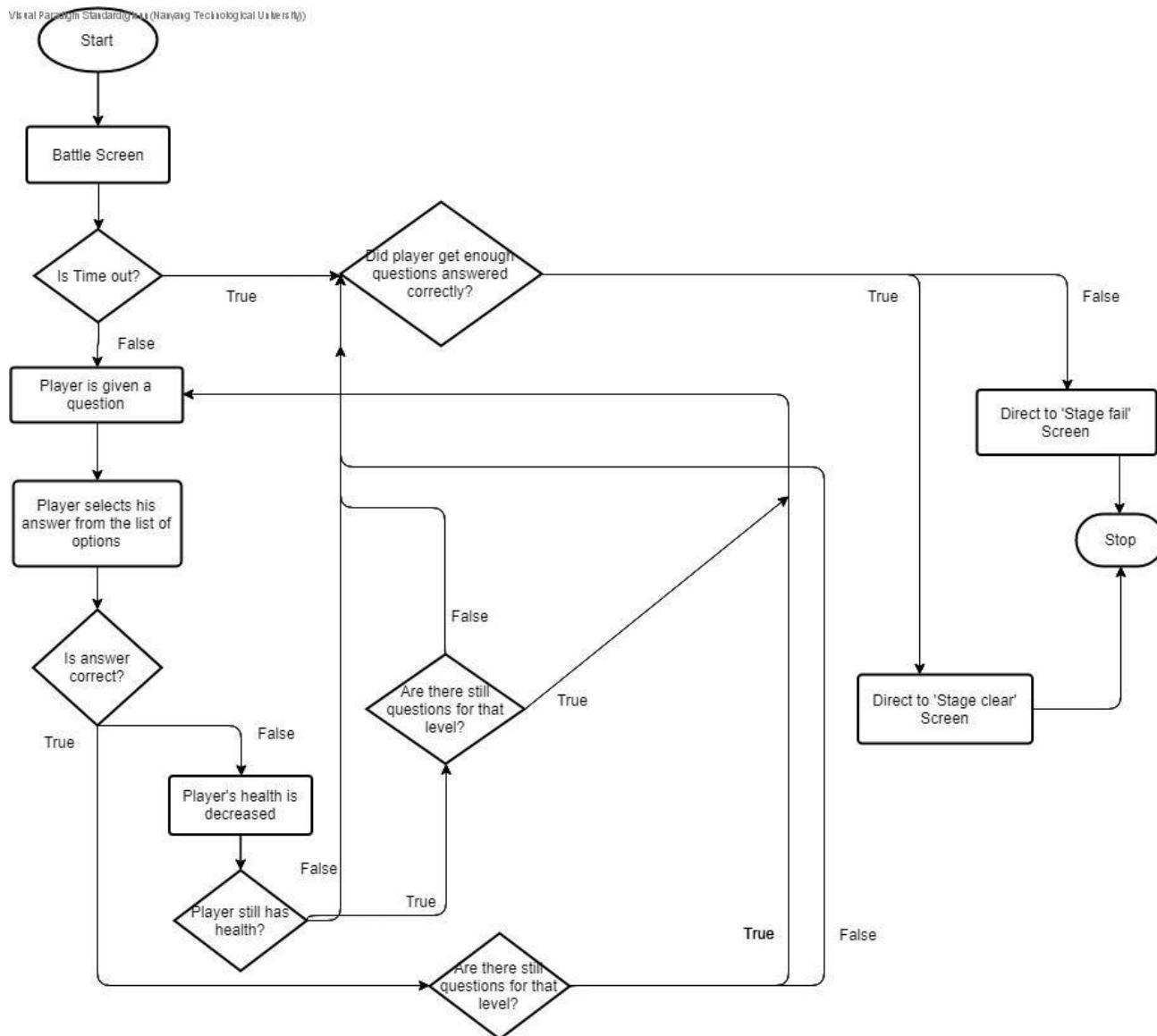
#### 7.4.4 Select a Stage



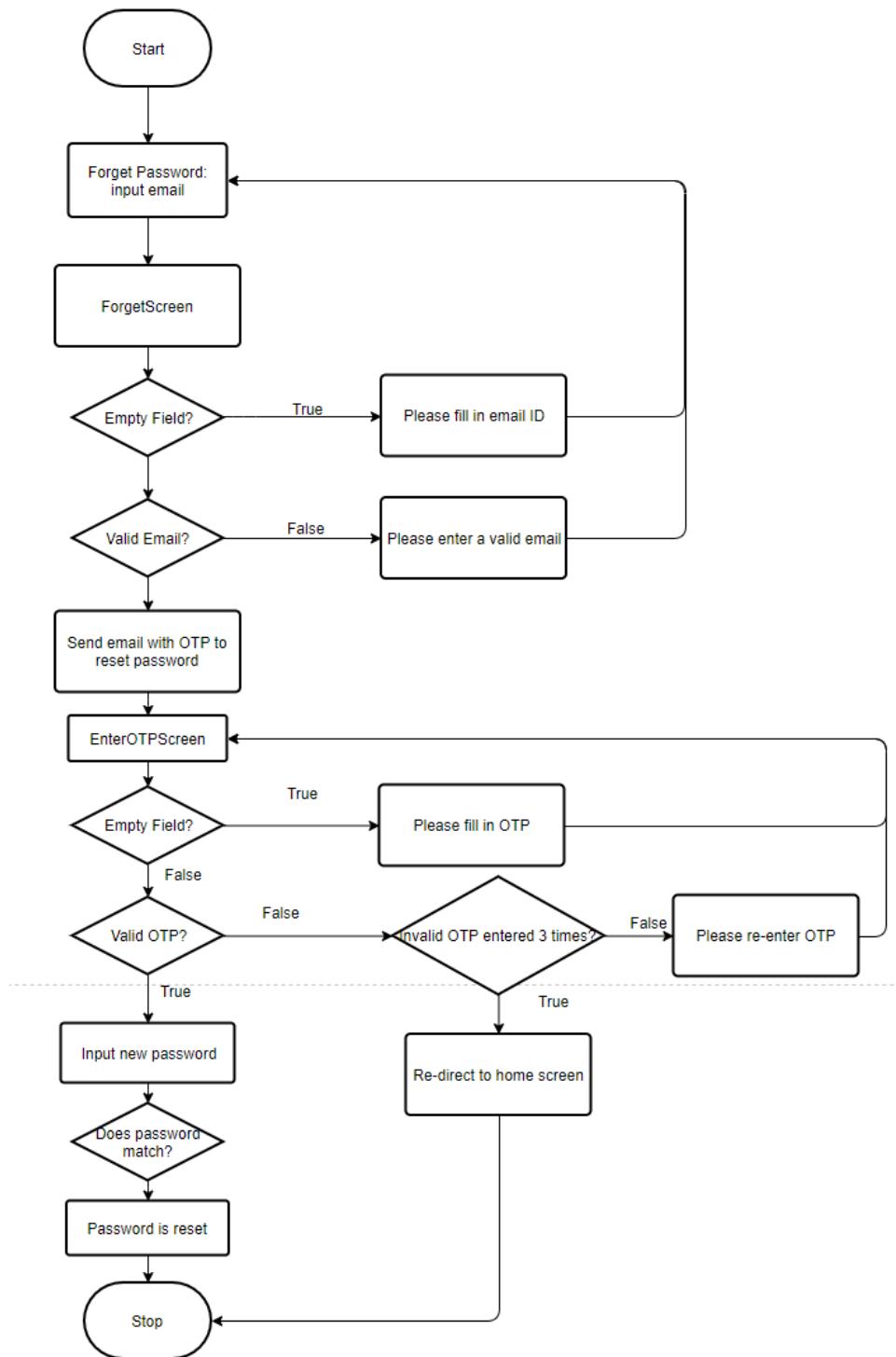
#### 7.4.5 Select a Level



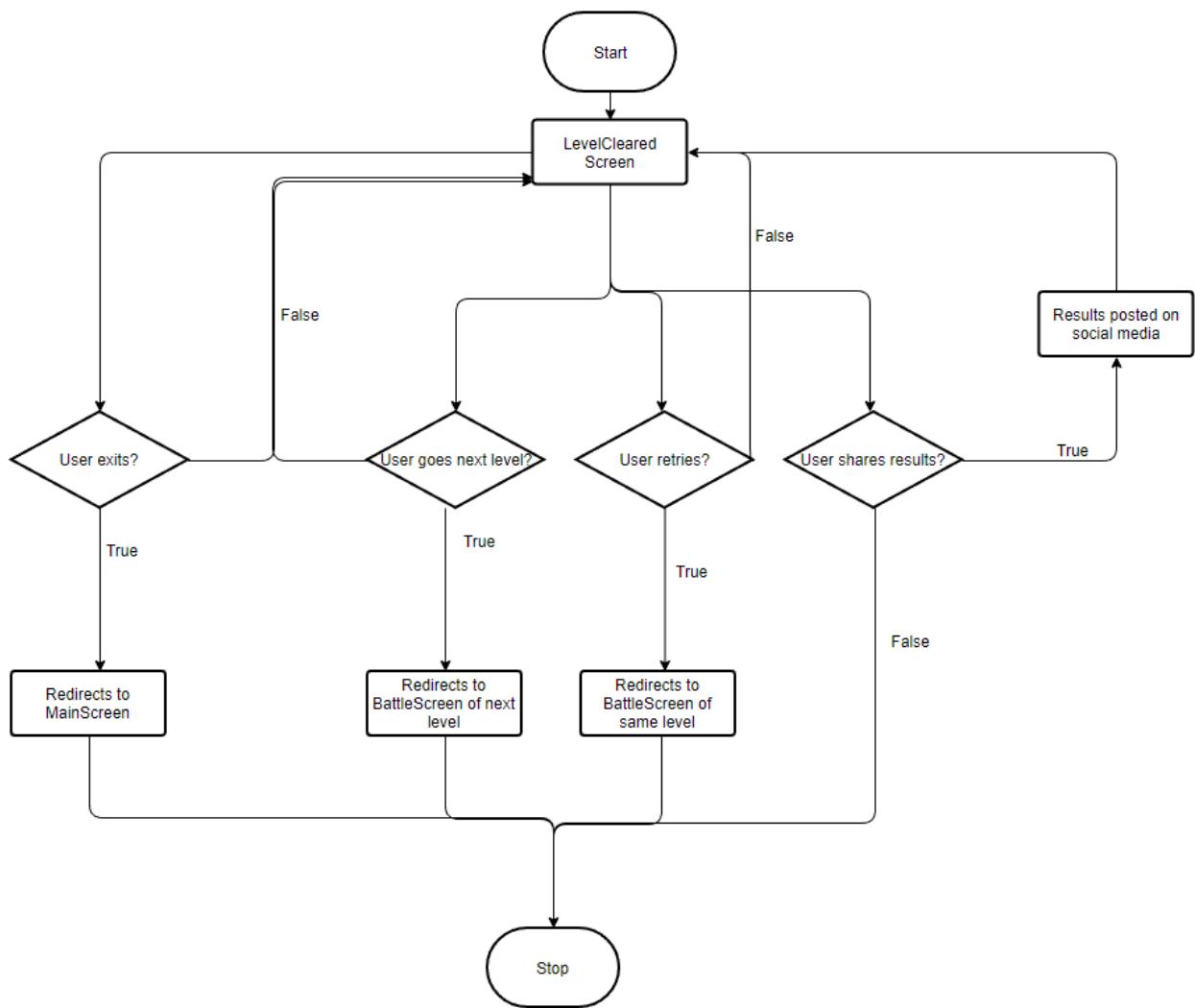
#### 7.4.6 Playing the game(battle screen)



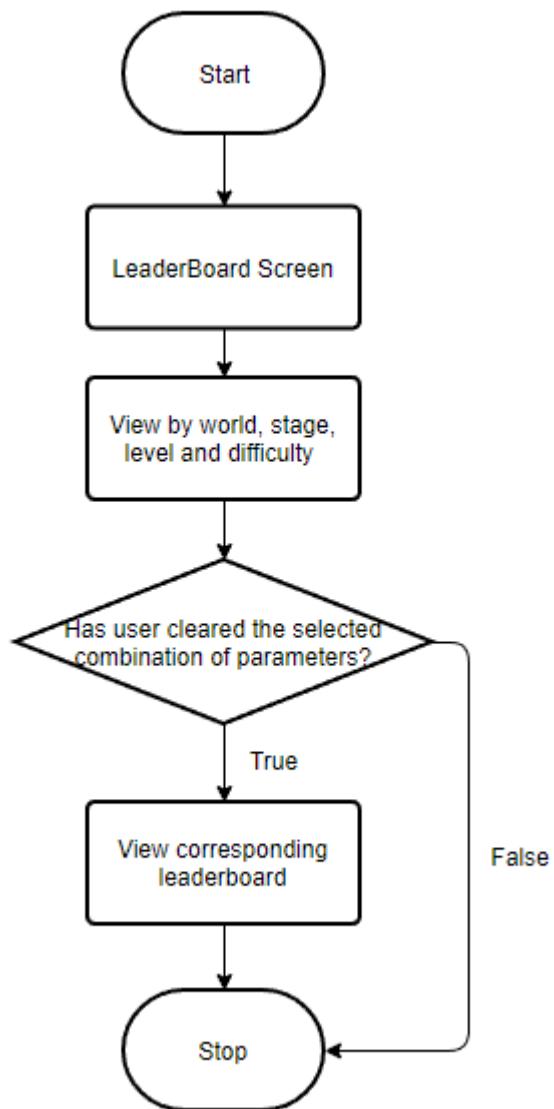
#### **7.4.7. Forget Password**



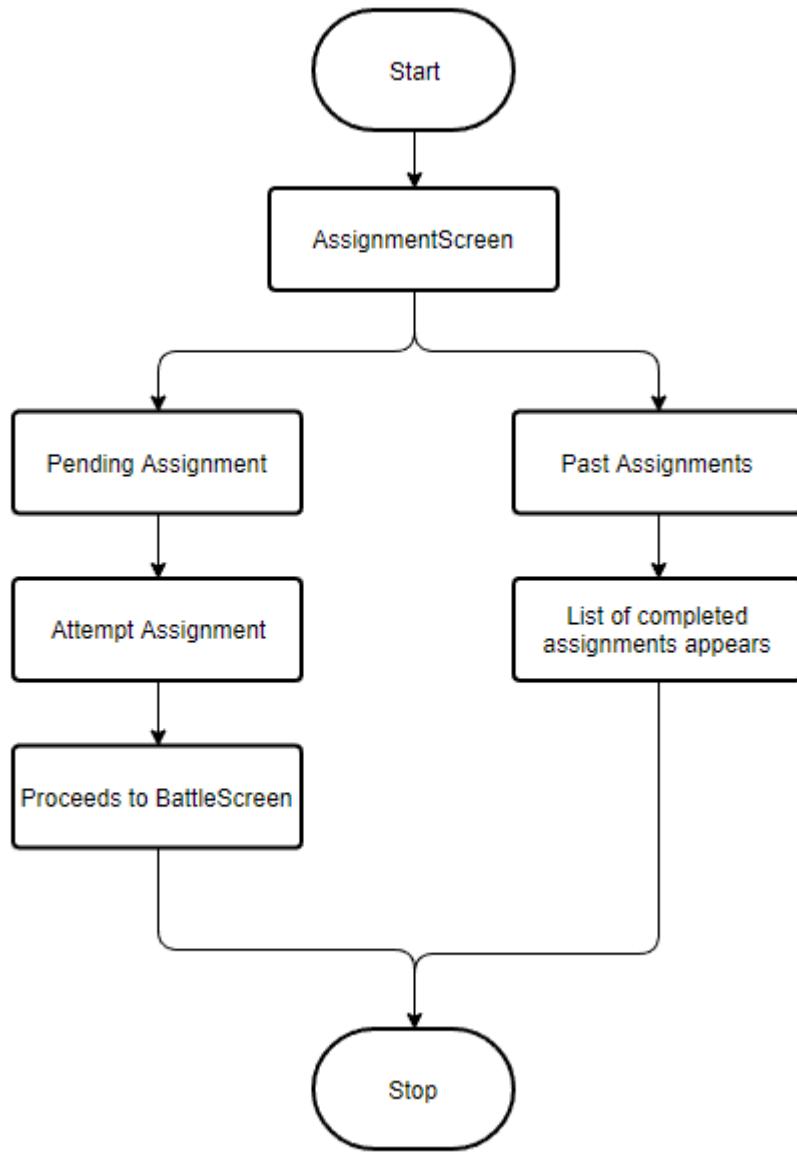
#### 7.4.8. Clearing a level



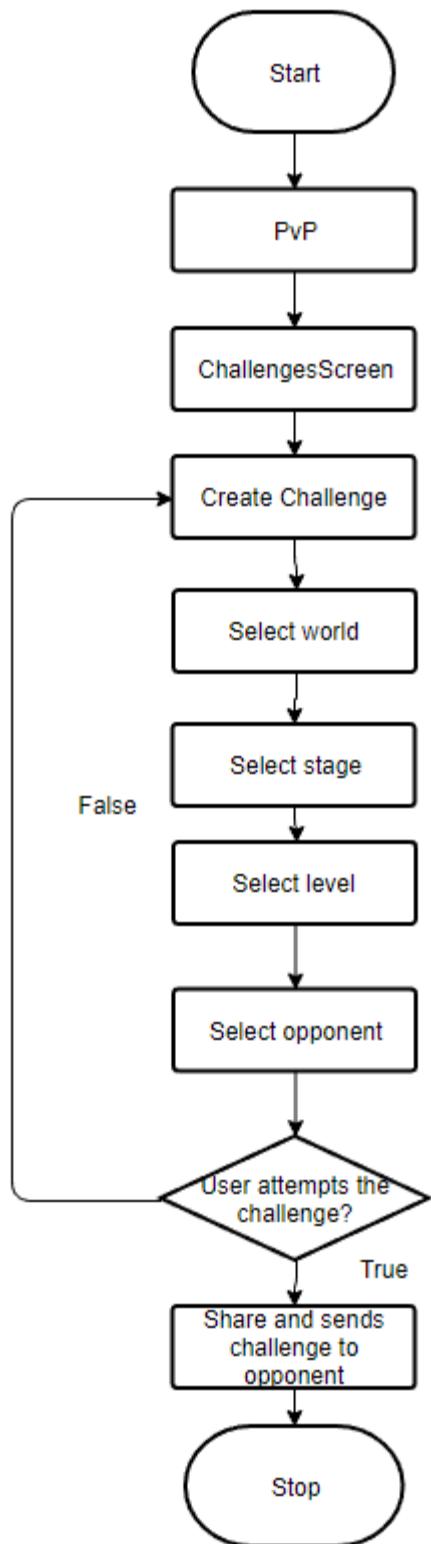
#### 7.4.9. Leaderboard



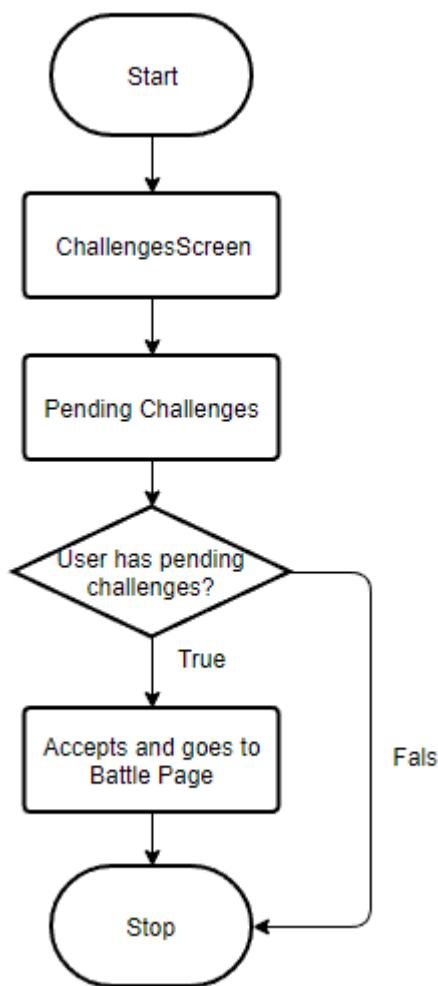
#### 7.4.10. Assignment Screen



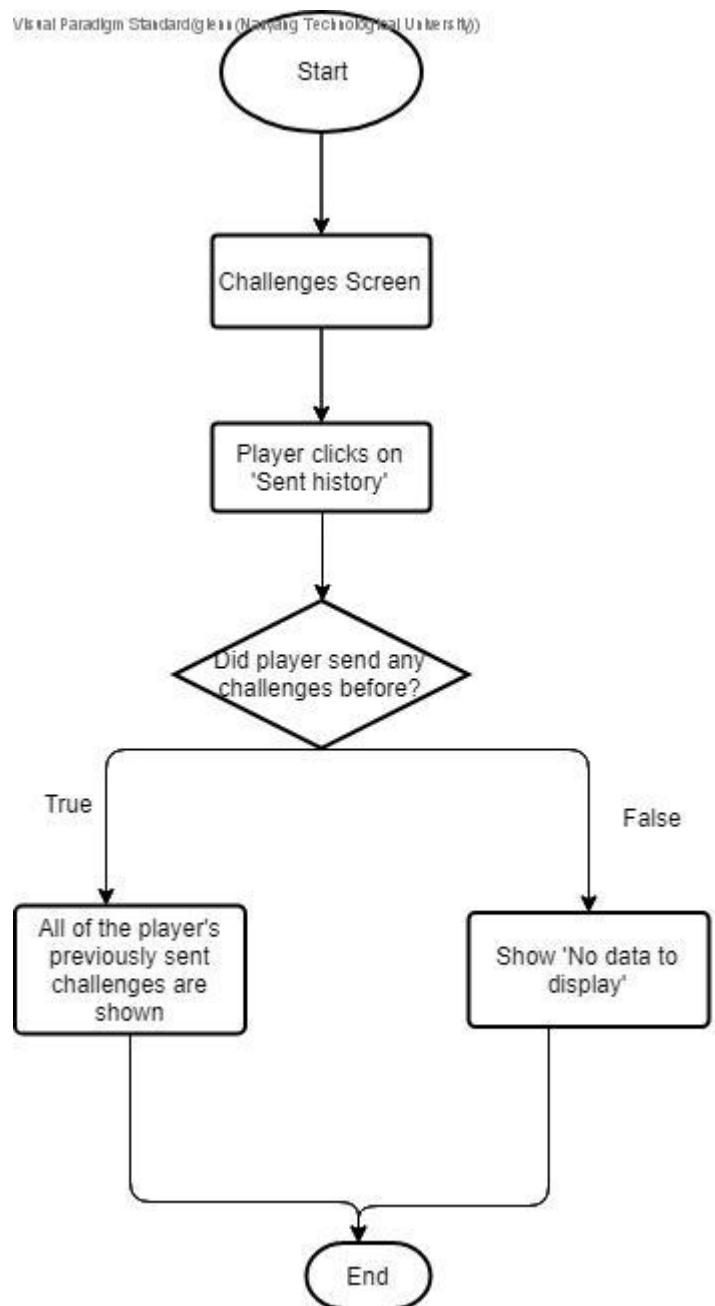
#### **7.4.11. Challenge Friends, creating and sending challenge**



#### 7.4.12. Viewing Challenge History

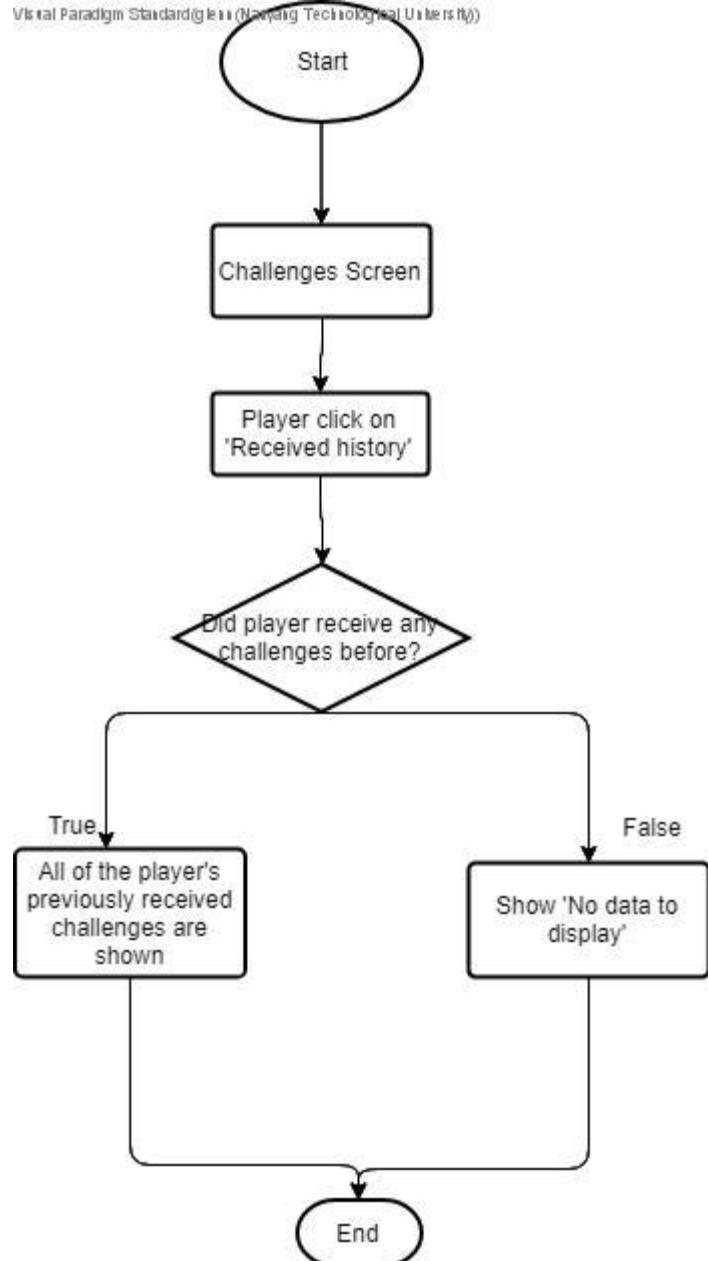


#### 7.4.13 View Past Sent History



#### 7.4.14 View Past Received Challenge

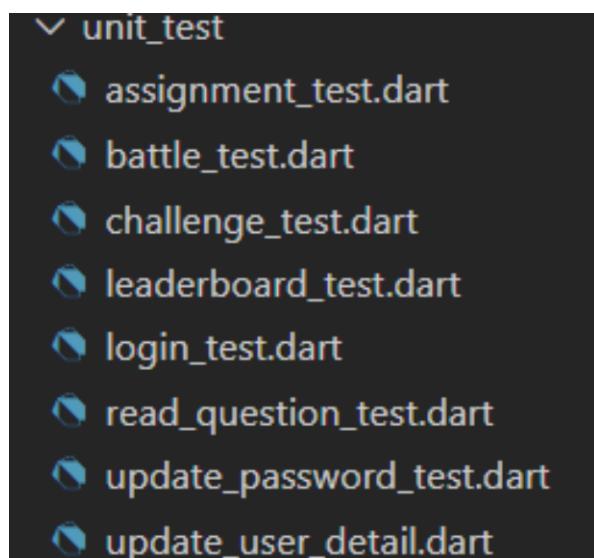
Visual Paradigm Standard (Java (Nanyang Technological University))



## 8.6 Unit Testing for Students

Unit testing, also known as component testing, is meant to ensure that each component or function will operate accurately as it is intended to. Individual classes will be tested to ensure reliability and functionality within a unit-level. White box testing is conducted at this phase. Since the team has developed the codes for the software, we have full knowledge of the fields necessary fields for each system. By studying the implemented code, we will determine all legal (valid and invalid) and illegal inputs and verify the outputs against the expected outcomes.

For the Intellect Game, we have used the Flutter testing framework to implement unit testing. We have written scripts within Flutter to test the inner workings of the application such as testing assignments, battle score update, challenges, leaderboard, login, reading questions, update password, update user detail. This type of testing has been carried out to ensure that each unit of our software is working as expected.



```
C:\Users\user\Desktop\Intellect-Student\Intellect-Student-Application>flutter test test\  
00:32 +8: All tests passed!
```

Following is an example of one of the unit tests written to check if the password of the user is successfully updated.

```
import 'package:test/test.dart';
import 'package:http/http.dart' as http;
import "dart:convert";

void main() {
  test("Update Password Test", () async {
    var k;
    var url = Uri.parse('http://localhost:8000/api/updatePassword');
    var response = await http.get(url);
    k = jsonDecode(response.body);
    expect(k["data"], true);
  });
}
```

```
59  router.get("/updatePassword", async (req, res) => {
60    const db = app1.firestore();
61    var verify = false;
62    const data = await db
63      .collection("Students")
64      .where("email", "==", "gopalagarwal1119@gmail.com")
65      .get();
66    data.docs.forEach((doc) => {
67      var temp = doc.data();
68      if (temp["email"] == "gopalagarwal1119@gmail.com") {
69        db.collection("Students").doc(doc.id).update({ password: "test" });
70      }
71    });
72    const data1 = await db.collection("Students").doc("gopal19").get();
73    var t = data1.data();
74    console.log(t);
75    if (t["password"] == "test") {
76      verify = true;
77    }
78    res.json({ data: verify });
79  });
}
```

## **8.7 Integration Testing**

Integration testing is performed to expose defects in the interfaces and in the interactions between integrated components or systems. We will be checking whether each unit is interacting correctly and message passing between two units are functioning properly. The integration test will be done after unit testing is being done on each sub system.

## **8.8 System Testing**

At this stage, all integrations between components will be done and the system would be completely functional. A validation test will be done to ensure that requirements of the system meets the specifications of the SRS in terms of its functional and non-functional requirements. The system testing will also surface limitations that the software will face.

We will be doing so manually using test cases. It would be more focused on UI and the User based. We would be focusing on transitions between scenes and ensuring that the application displays the correct information and scene when certain actions are being executed.

## 8.9 Performance Testing

Here we have tested if there are multiple continuous connections to the database and see how it performs. We observe for a 100 connection, time taken is around 28 seconds.

```
import 'package:test/test.dart';
import 'package:http/http.dart' as http;
import "dart:convert";

void main() {
  test("Load Test", () async {
    var k;
    var url = Uri.parse('http://localhost:8000/api/loadTest');
    var response = await http.get(url);
    k = jsonDecode(response.body);
    expect(k["data"], lessThan(300));
  }, timeout: Timeout.none);
}
```

## **Appendix A: Glossary**

<b>Term</b>	<b>Definition</b>
Account	The account of the User who is currently using the application.
Web Application	A browser based system for Teachers to login to monitor the Students' progress in the game.
Application	The main mobile based game that Students log in to play.
Students	The target audience for the mobile game application. Used interchangeably with players.
Teacher	The person monitoring the progress of the Students who are playing the game.
OTP	A randomly generated string of numbers sent to a user's email to confirm change in password.
Character	A profile with its own special abilities. Newer characters are initially locked and can be unlocked as the user's player level increases.
Locked/Unlocked	Some characters and worlds are locked and inaccessible until the prerequisite player level, at which point they become unlocked and accessible.
Login	By inputting the correct Username and Password, it would allow the user to access the profile associated with the Username.
Special Abilities	The unique abilities that a character can activate once per level.
Points	Reward for each level is given based on how many questions are answered correctly in the level.

Player Level	Every certain total amount of points will cause the player level to increase.
World	A world is made up of 3 stages based on a certain software engineering topic.
Stage	A Stage is made up of 3 levels with similar sub-topics.
Level	A Level is made up of varying amount of monsters depending on difficulty
Difficulty	There are 3 difficulties for each stage: easy, medium and hard.
Monster	“Attacks” the user by giving them questions. Defeat them by answering the question correctly. Represents the questions given by the Teacher.
Hit Points (HP)	When a question is answered incorrectly, the player’s HP will be deducted. When HP reaches 0, the user loses the level and has to either restart or return to the main page.
Score	Score for each level is equivalent to the amount of points scored in that level.
Profile	Each user’s personal data and settings collectively forms his/her profile.
Leaderboard	Shows the ranks of different Students playing the game.
Rank	Rank describes how well a User is doing as compared to other users, with a lower numerical rank being indicative that a player is better.
Assignment	Task given by a Teacher for the Students to complete.

