Software Requirements Specification

for

SSADungeon

Prepared by Group 1

Alvin, Bryan, Chirs Chong, Daniel, Hong Weng, Jonathan, Joshua, Keerthana, Sanath

Nanyang Technological University, Singapore

April 21, 2021

---

1. Introduction **3**
1.1 Purpose 3
1.2 Document Conventions 4
1.3 Intended Audience and Reading Suggestions 4
1.4 Product Scope 5

2. Overall Descriptions **5**
2.1 Product Perspective 5
2.2 Product Functions 6
2.3 User Classes and Characteristics 6
2.4 Operating Environment 7
2.5 Design and Implementation Constraints 7
2 .6 User Documentation 7
2 .7 Assumptions and Dependencies 8

3. External Interface Requirements **10**
3.1 User Interfaces 10
3.2 Hardware Interfaces 31
3.3 Software Interfaces 31

4. System Features **32**
4.1 Registering 32
4.2 Main Menu 32
4.3 Adventure Game Mode 32
4.4 Worlds inside Adventure Game Mode 33
4.5 Stages inside Adventure Game Mode 34
4.6 Levels Inside Adventure Game Mode 35
4.7 Leaderboards 37
4.8 Player Versus Player (PVP) Game Mode 38
4.9 Teacher’s Dashboard 39

5. Diagrams & Descriptions **42**
5.1 Use Case Diagram 42
5.2 Use Case Descriptions 43

6. Non-Functional Requirements **64**

7. Candidate Architecture Style **65**

8. Communication Diagram **89**
8.1 Student Diagram 89

8.2 Teacher Diagram 90
8.3 Leaderboard Diagram 90

9 Testing 91

Appendix A: Data Dictionary 105

Appendix B: Analysis Models 106

1. Introduction

1.1 Purpose

The purpose of this game is to gamify and socialize teaching and learning of software engineering. The game is also a way to encourage students to learn and compete with each other via a fun platform, where integrated learning can take place. Students can also learn at their own pace, allowing them to pick up skills relevant to software engineering.

The game will also allow teachers to access the mastery their students have over the course via data analytics, allowing them to more effectively craft their lesson plan.

1.2 Document Conventions

The headers in this document will be using the font Times New Roman of size 14. Furthermore, the content inside each heading will be written using the font Arial of size 11.
In numbering this document, we have employed an indented numbering system whereby a title numbered 1 has subheadings 1.1, 1.2 and so on.
Further, note that this document follows the IEEE template provided in its formatting as well.

1.3 Intended Audience and Reading Suggestions

The intended audience of this document includes the developer of the game, the project managers, and the users involved in the design of the game (including but not limited to teachers and students), personnel responsible for the development and testing of the application.

The suggested reading sequence are as follows.

1. Overall Description
2. External Interface Requirements
3. Initial UI Mockups
4. Functional Requirements

5. Non-functional Requirements
6. System Architecture
7. Appendix

Developers who might need to review important features, requirements and other information pertaining to the application are suggested to view sections 1, 2, 4, and 5. Further, they can view terminologies regarding the application under section 4 of Data Dictionary. Readers who may be interested to visualize the application can read and view the interface under section 3.

1.4 Product Scope

The main objective of this application is to gamify and socialize teaching and learning of software engineering courses in NTU, making learning engaging and competitive for students.
On a larger scope, it would be to incorporate this application to all schools teaching this subject to its students.

2. Overall Descriptions

2.1 Product Perspective

This application will be used to assist learning and teaching of software engineering concepts and topics through gaming in NTU. It can also be used to recap and revise software engineering concepts quickly and easily through quiz and challenges.

2.2 Product Functions

- The application shall consist of six worlds in the adventure game mode for players to navigate and play in, each one representing the different phases of the life cycle of software engineering, also known as Software Design Life Cycle (SDLC).
- Inside each of the world, there will be several stages, representing specific topics of each phase of SDLC from basic ones to advanced ones.
- Inside each of the stages, there will be five levels, each one representing a difficulty level from 1 to five.
- Inside each of the levels, there will be a dungeon with 10 NPC for players to challenge. Users can fight the NPC, and once the HP of the NPC reaches 0, a question will pop up and players have to answer before they can move on to challenge the next NPC, when all 10 are defeated, the user will have completed the dungeon.
- In order to proceed to the next level, the player shall defeat at least five NPC in that dungeon of that level. If he wins more than 7, he can unlock the next two levels.
- In order to proceed to the next stage, the player shall at least reach level three of the previous stage.
- In order to unlock the next world, the player shall pass a world quiz.
- Players can challenge each other in the PVP mode, either by designing their own levels, or challenging the level designed by other players.
- The system shall provide a leaderboard for both the PVE and PVP mode to engage students to aim for higher ranks.
- Teachers should be able to generate a summary report to monitor the progress of students in the course.
- Teachers shall be able to give out assignments through at least two social media platforms.

- Teachers shall be able to add questions into the question bank in any world, stage and level.

2.3 User Classes and Characteristics

- Teachers who will use this application to gamify and socialize learning of software engineering concepts. They can also monitor student's progress and assess their mastery through summary reports from data analysis.
- Students who are learning SDLC for the first time, can use this application to improve and solidify what they have learnt in class, as well as to keep track of their understanding. It is likely that these users would use the application very frequently.
- Students who are experienced can use the SDLC to recap, revise and test their concepts. It is likely that the usage of these users would be less frequent as they would use it once in a while to strengthen and recap their concepts.

2.4 Operating Environment

Our game application has been built on Android Studios, and utilizes the Flutter framework. Thus, this allows for cross platform compatibility and can thus be built on both IOS and Android devices. The mobile application will make use of the Internet access of the mobile device.
For building this game application, the following environment was used in Flutter
environment:
sdk: ">=2.7.0 <3.0.0"
The specification are as follows
Mobile operating systems:
1. Android- Android Jelly Bean, v16, 4.1.x or newer
2. iOS- iOS 8 or newer

2.5 Design and Implementation Constraints

Database: Firebase
CloudStorage: Firebase
FrontEnd: Dart/Flutter

Security Considerations: The user must login in the front end, and his subsequent access will be verified through Firebase Authentication.

2 .6 User Documentation

Our user documentation will consist of user manual, system reference, on-line help, tutorials and a video description.

To make the structure of the user documentation is clear and readable. Format requirements are:

1. Proper indentation
2. Insert headings, page number and section titles
3. Integrate text fonts and style (Times new roman,14)
4. Use columns to layout
5. Insert picture for key steps and buttons
6. Use bold to highlight dangerous or invalid operations

Since our users most not have high technical background so the standards are:

1. Simple words and short sentences only
2. More short Verb+Noun to show operations instead of explaining a long process
3. If use professional word or name, paste a link at the end

2 .7 Assumptions and Dependencies

The software dependencies we have in flutter are as follows and can be found in the pubspec.yaml file

dependencies:

```
  flutter:
    sdk: flutter


  flutter_share_me: ^0.9.1
  font_awesome_flutter: ^9.0.0
  share: ^2.0.1
  flame: ^0.26.0
  audioplayers: ^0.15.1
  bonfire: ^0.6.27
  cupertino_icons: ^1.0.2
  shared_preferences: ^0.5.7+3
  intl: ^0.17.0
  flutter_rating_bar: ^3.0.1+1
  provider: ^5.0.0
  sqflite: ^1.3.2+3
  path_provider: any
  firebase_auth: ^1.0.1
  firebase_core: ^1.0.2
  cloud_firestore: ^1.0.3
  flappy_search_bar: ^1.7.2
  flutter_spinkit: ^5.0.0
```

3. External Interface Requirements

3.1 User Interfaces

Attached are some screenshots of the application, that provide a basic insight on the user interface of the application, that is, the application in the user.

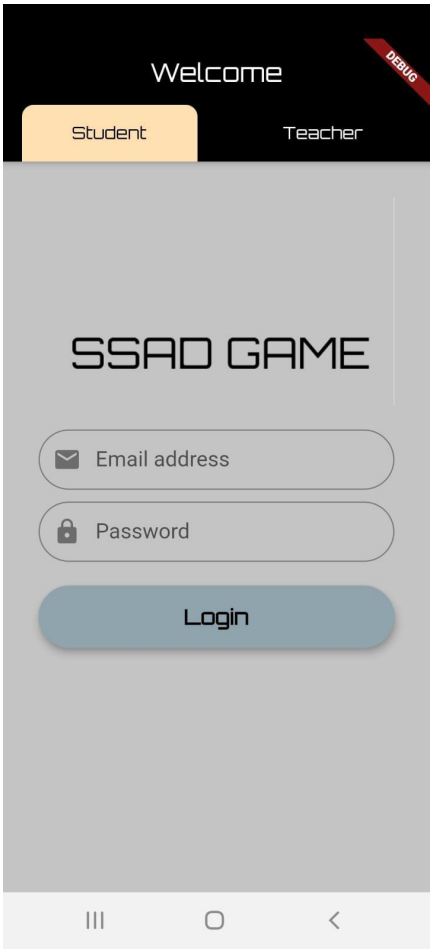


Figure 4.1.1 – Login Page for Student

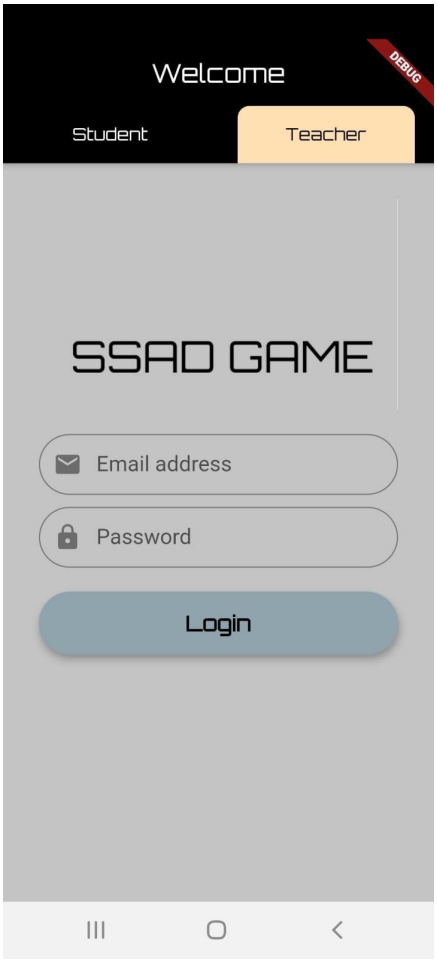


Figure 4.1.2 – Login Page for Teacher

Student Account Interfaces:

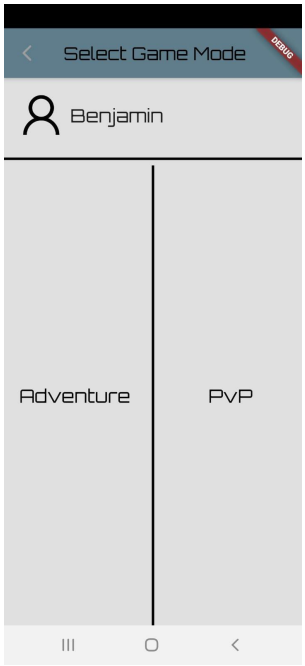


Figure 4.1.3 – Mode Selection

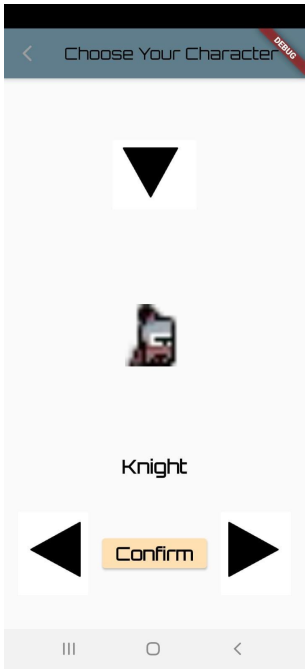


Figure 4.1.4.1 – Avatar Selection for PvE Mode

Figure 4.1.4.2 - Avatar Selection as Mage for PvE Mode

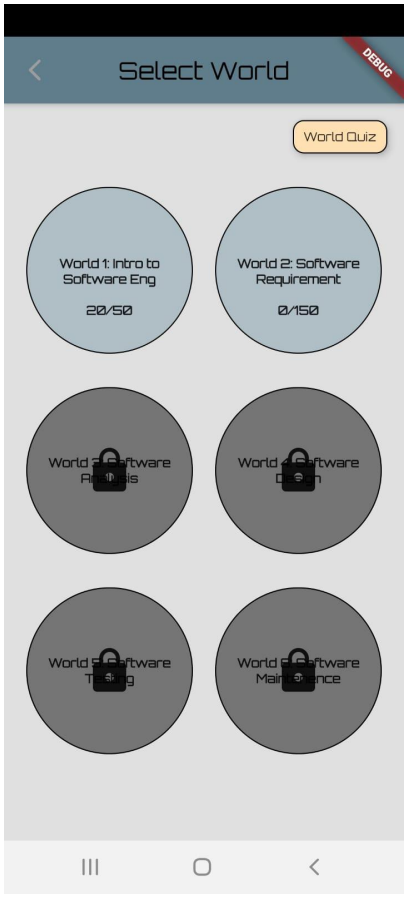


Figure 4.1.5.1 – World Selection

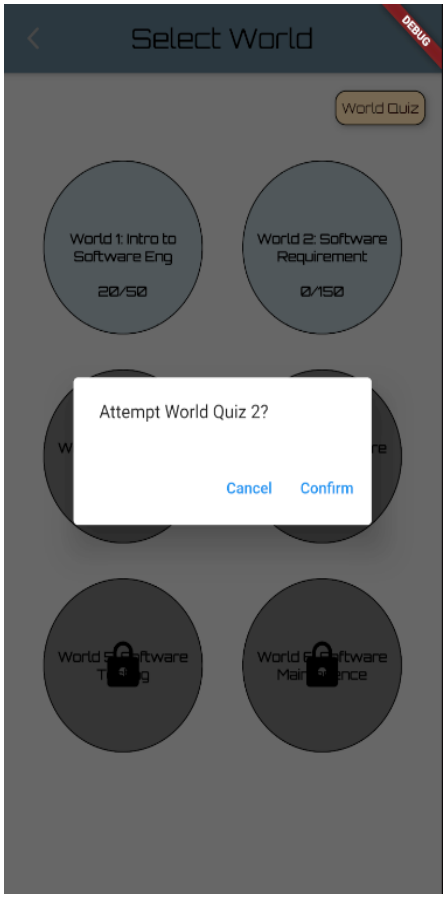


Figure 4.1.5.2 – World Quiz

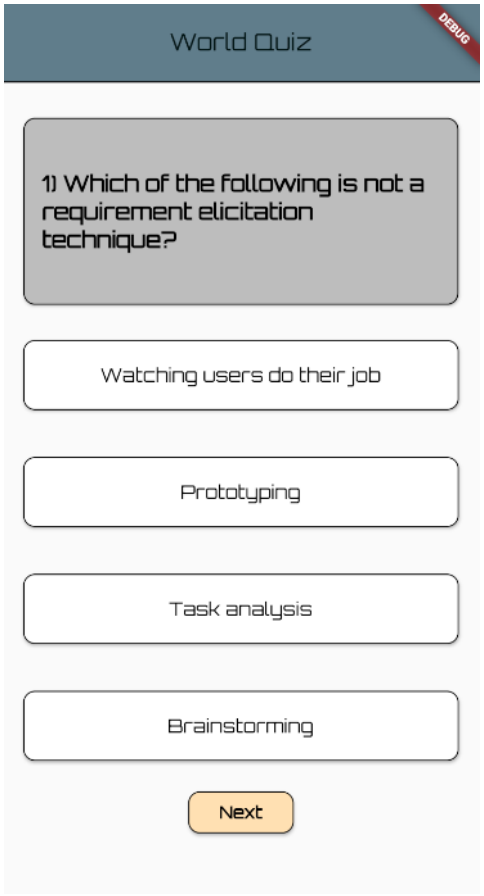


Figure 4.1.5.3 – World Quiz Questions

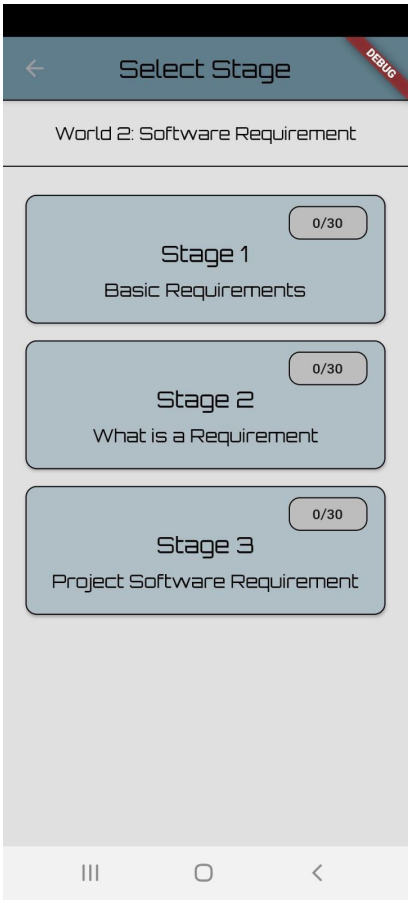


Figure 4.1.6 – Stage Selection

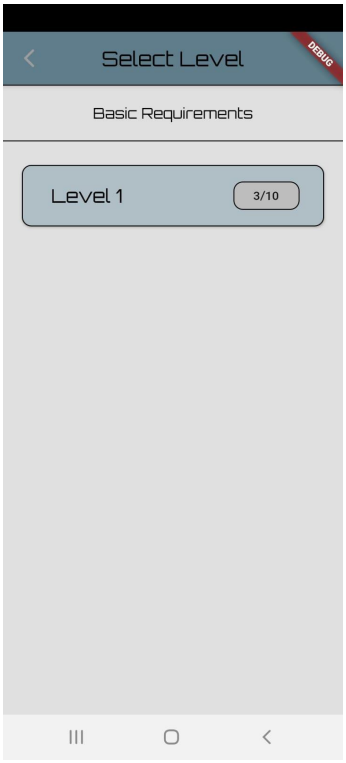


Figure 4.1.7 – Level Selection

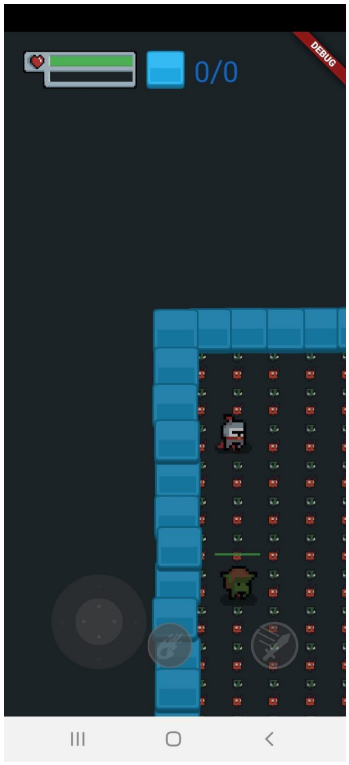


Figure 4.1.8.1 – Game Play as Knight

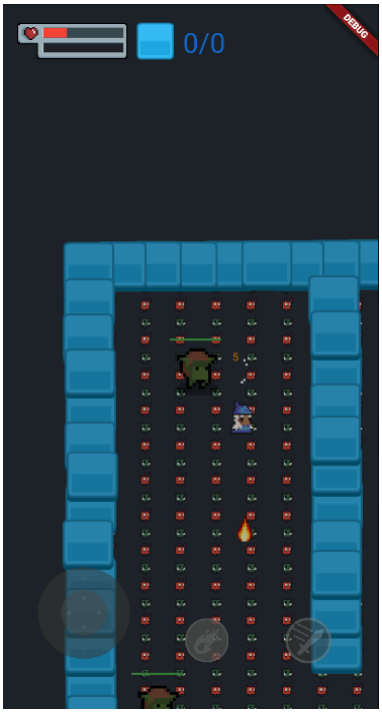


Figure 4.1.8.2 – Game Play as Mage shooting Fireballs to attack

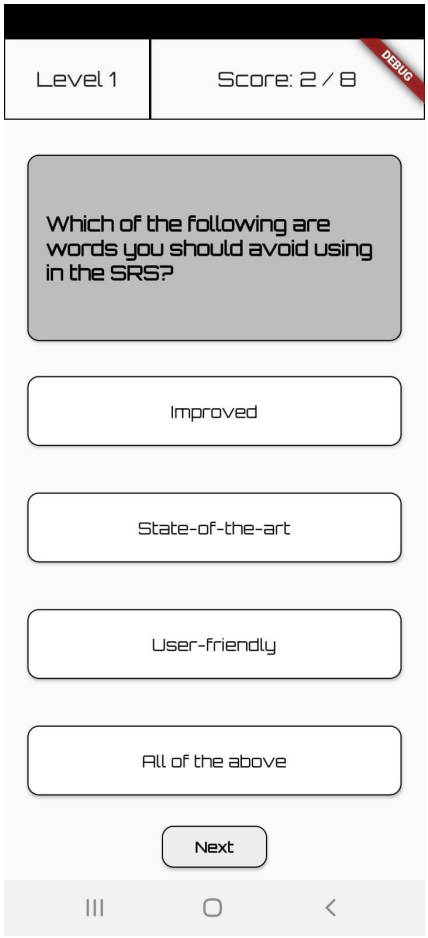


Figure 4.1.9.1 – Questions during Game Play

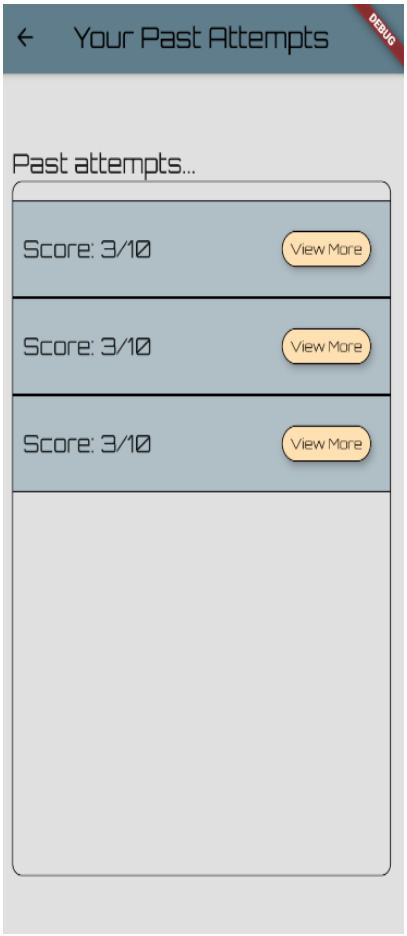


Figure 4.1.9.2 – Past Attempts after Game Play

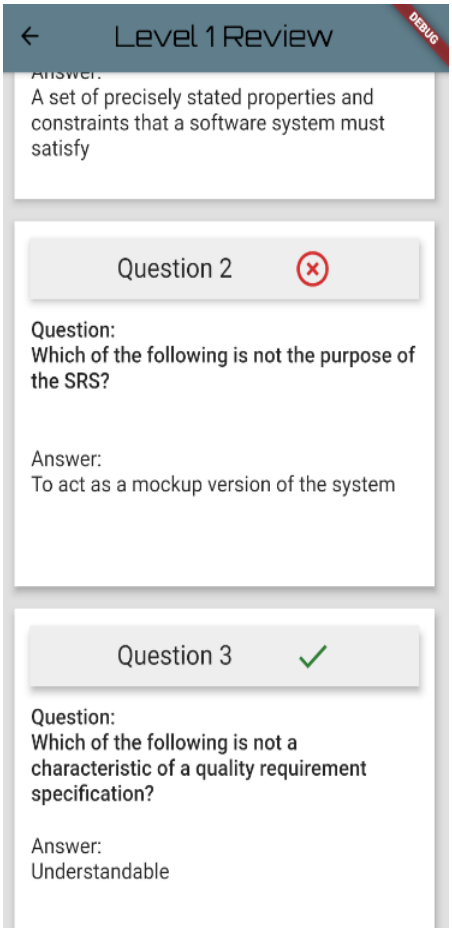


Figure 4.1.9.3 – Detailed Level Review

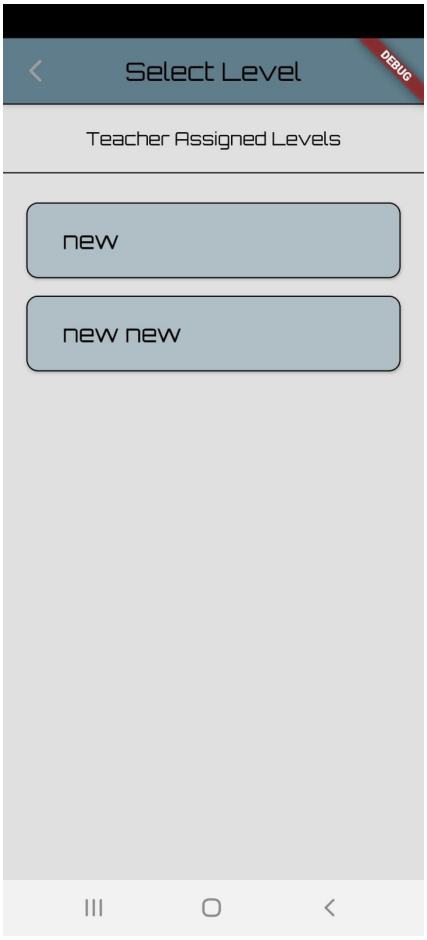


Figure 4.1.10 – Teacher Assignment Available

Figure 4.1.11 – Adventure (PvE) Leaderboard

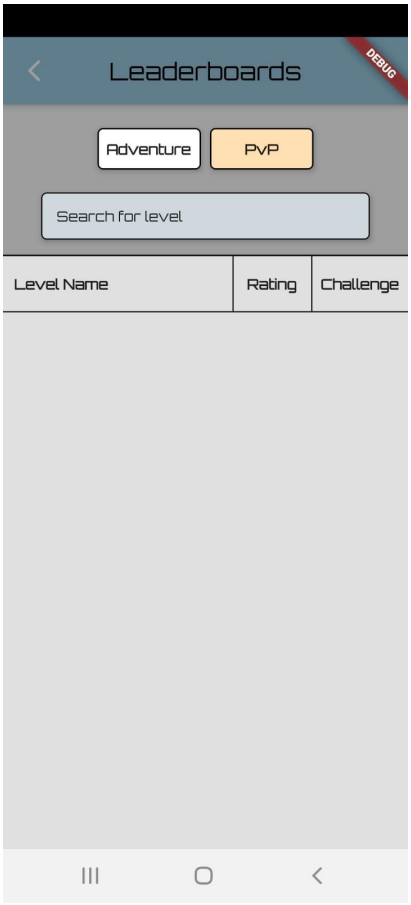


Figure 4.1.12 – PvP Leaderboard

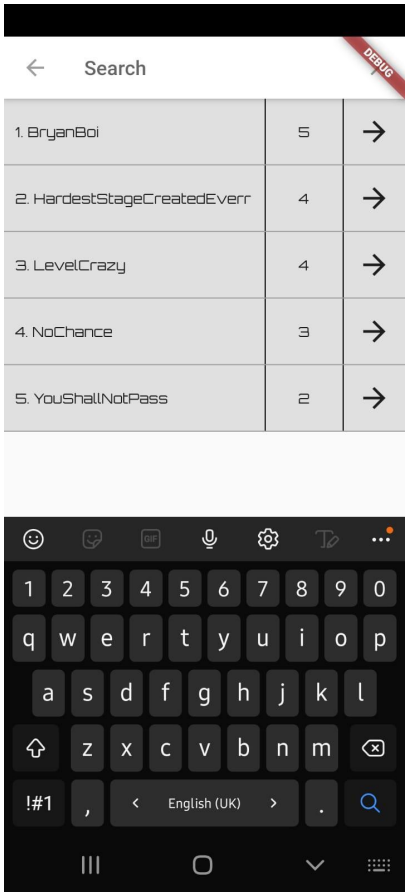


Figure 4.1.13 – Level Leaderboard

Figure 4.1.14 – Leaderboard in specific Level

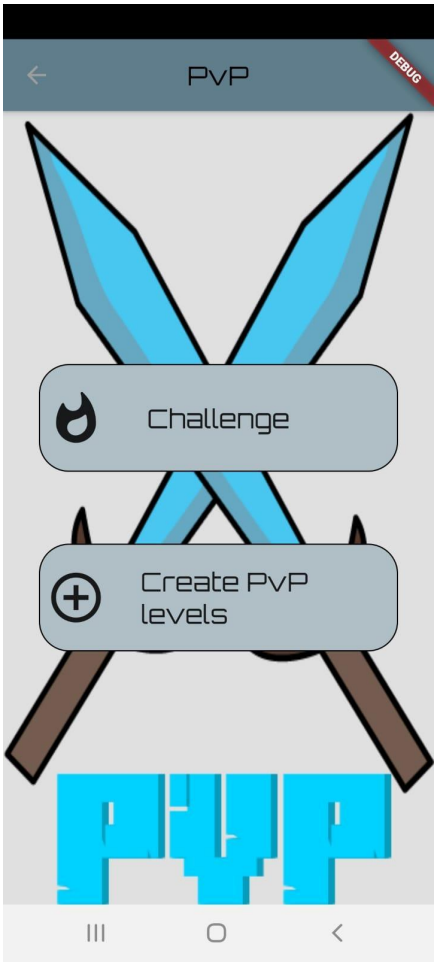


Figure 4.1.15 – Choices Available for PvP Mode

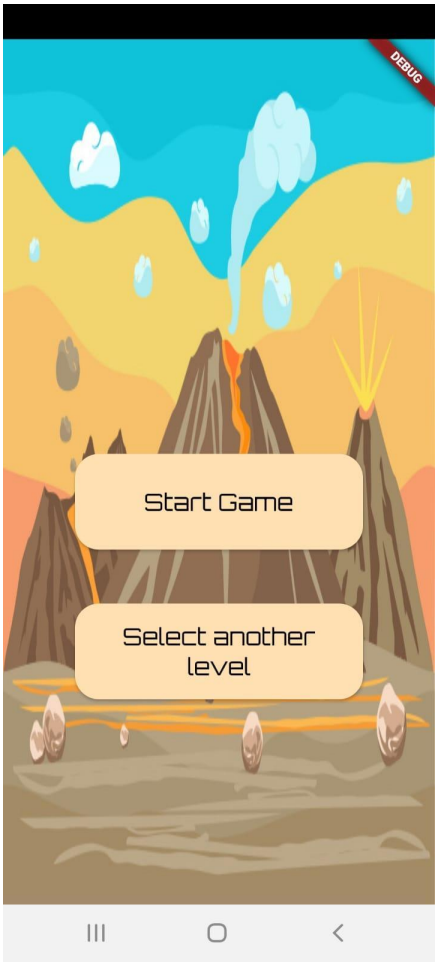


Figure 4.1.16 – PvP Challenge

Figure 4.1.17 – PvP Levels Available for Selection

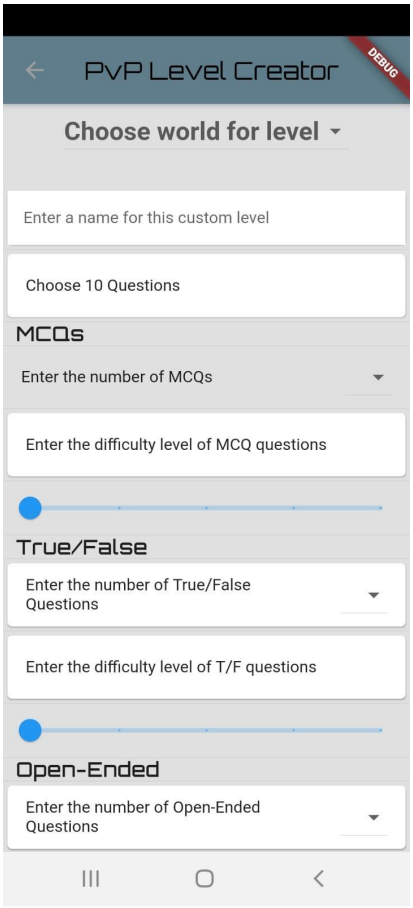


Figure 4.1.18 – PvP Level Creation

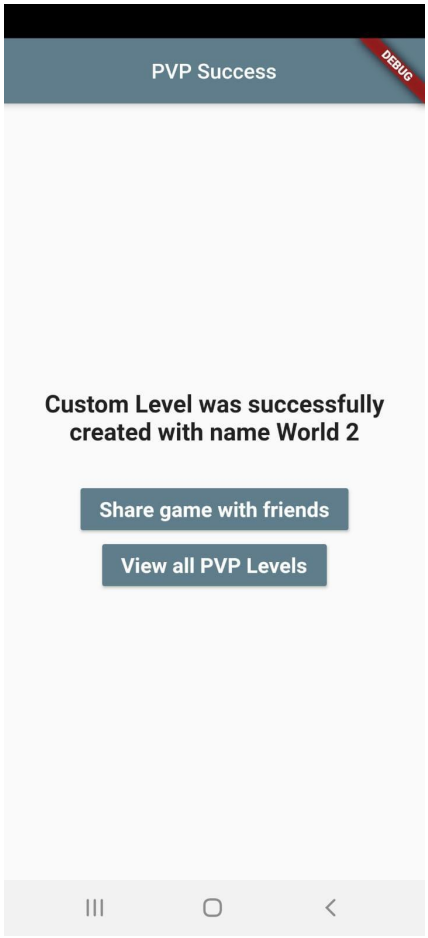


Figure 4.1.19 – PvP Level Successfully Created

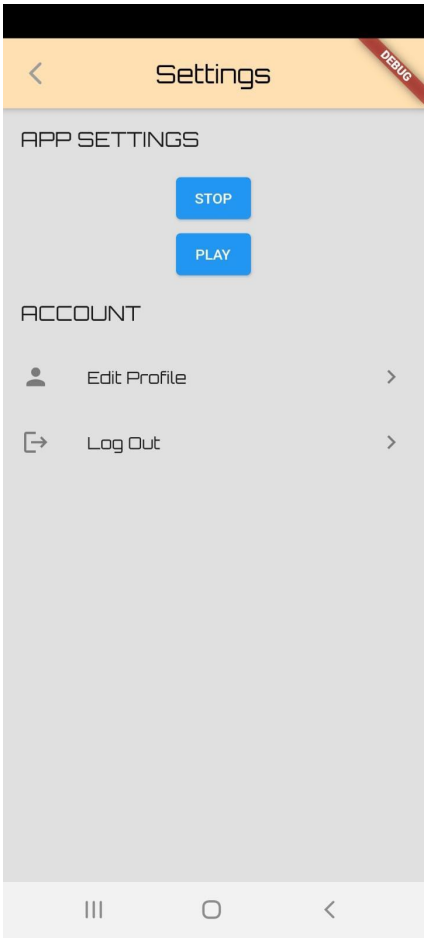


Figure 4.1.20 – Settings

Teacher Account Interfaces:

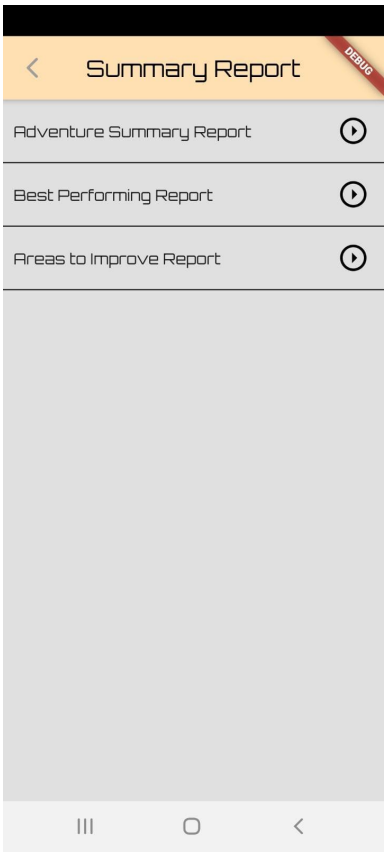


Figure 4.1.21 – Summary Report Selection

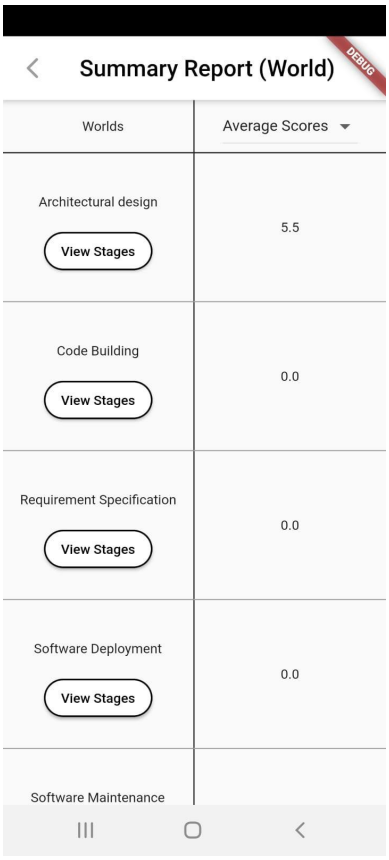


Figure 4.1.22 – World Summary

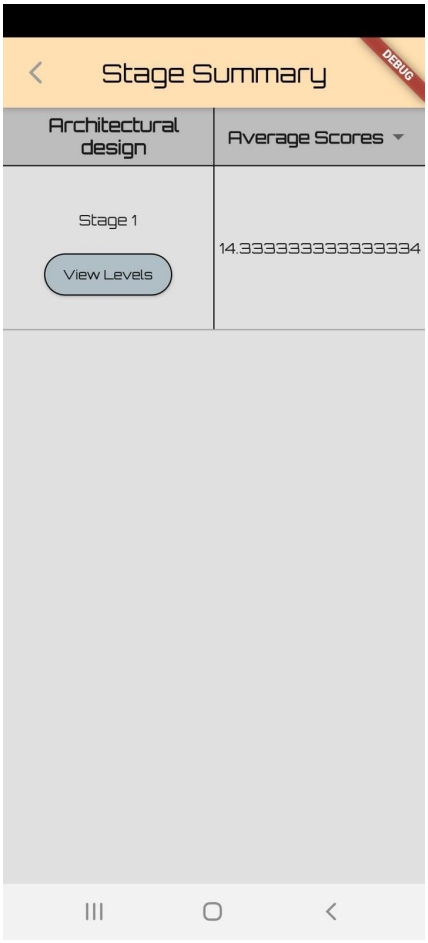


Figure 4.1.23 – Stage Summary

Figure 4.1.24 – Level Summary

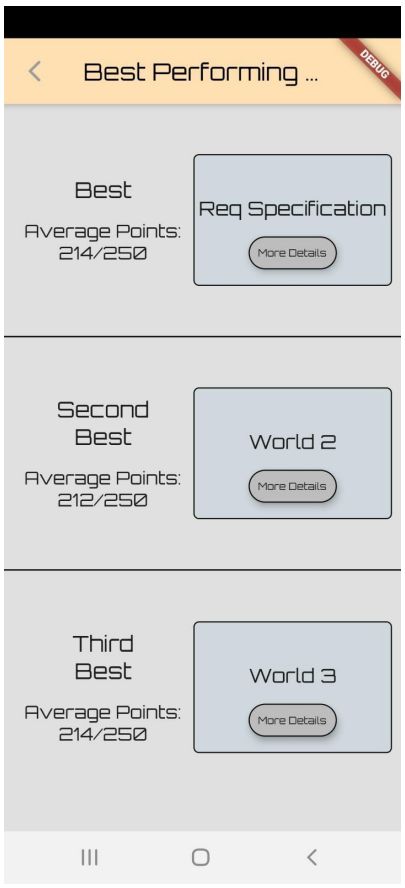


Figure 4.1.25 – Best Performing World

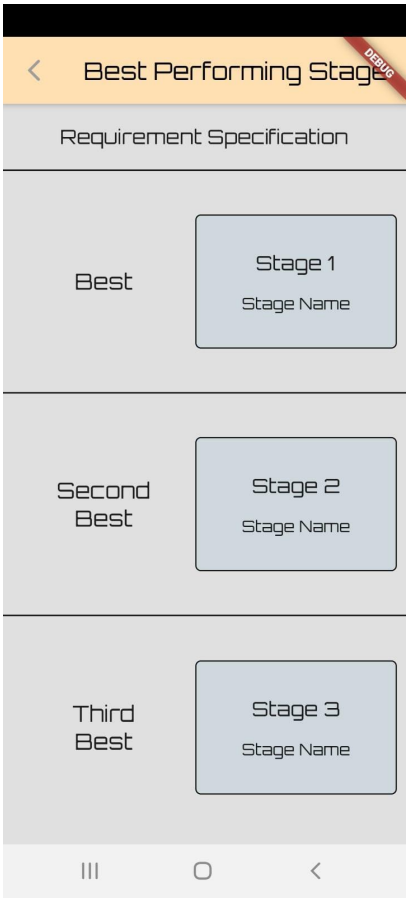


Figure 4.1.26 – Best Performing Stage

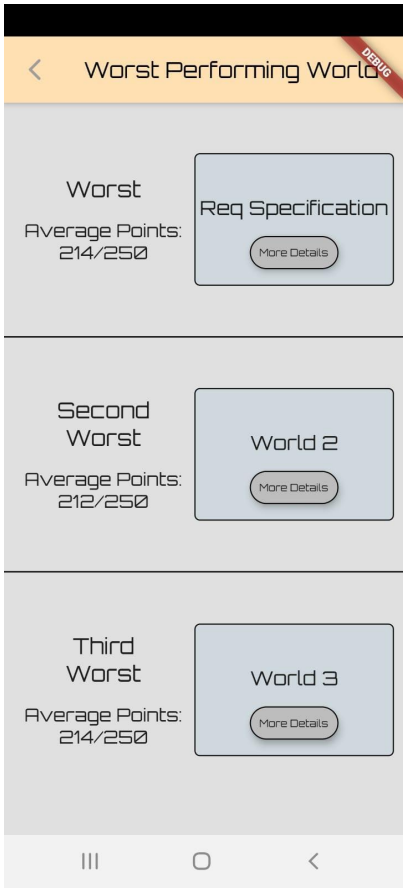


Figure 4.1.27 – Worst Performing World

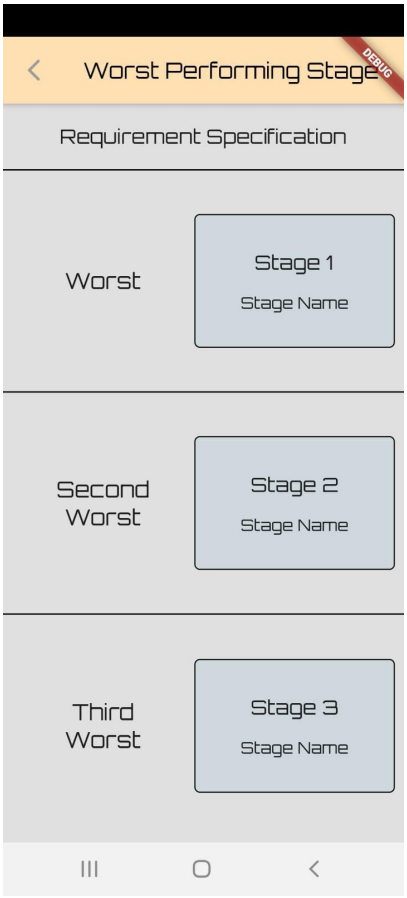


Figure 4.1.28 – Worst Performing Stage

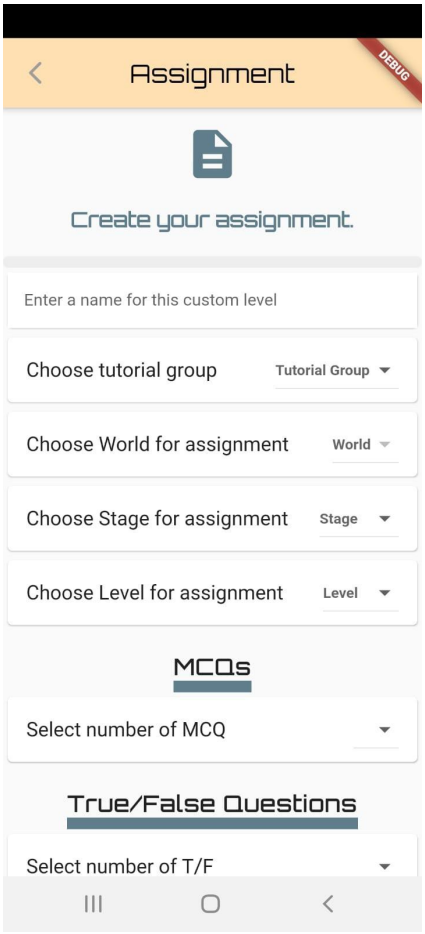


Figure 4.1.29 – Assignment Creation

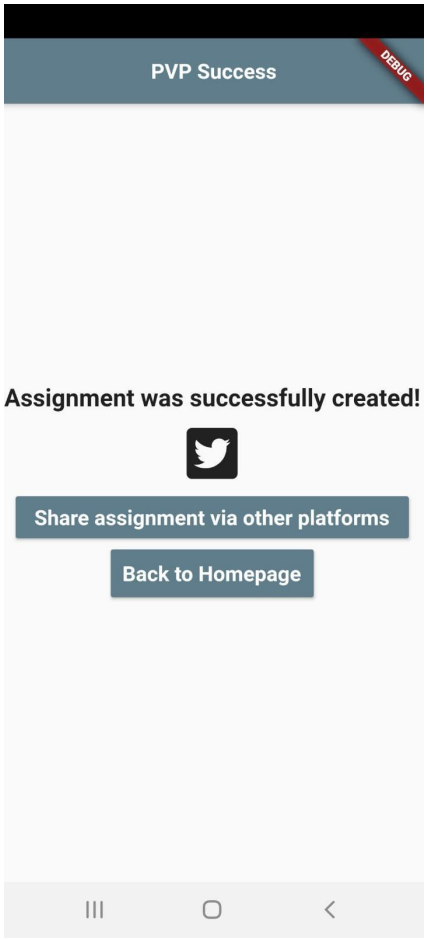


Figure 4.1.30 – Assignment Successfully Created

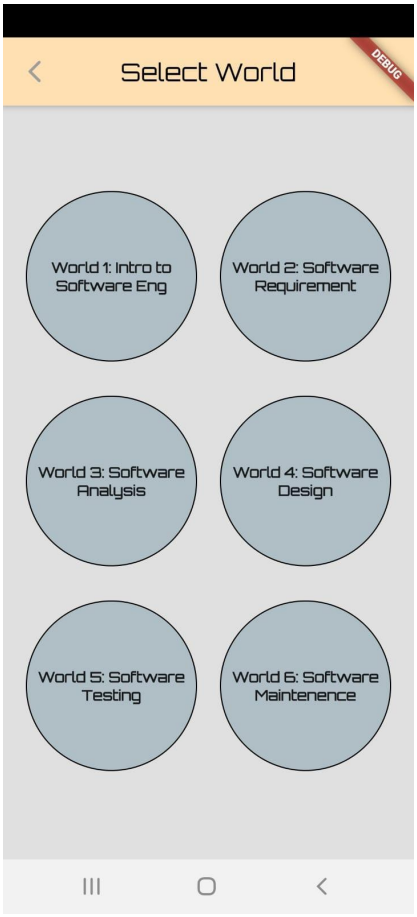


Figure 4.1.31 – World Selection to Add Question

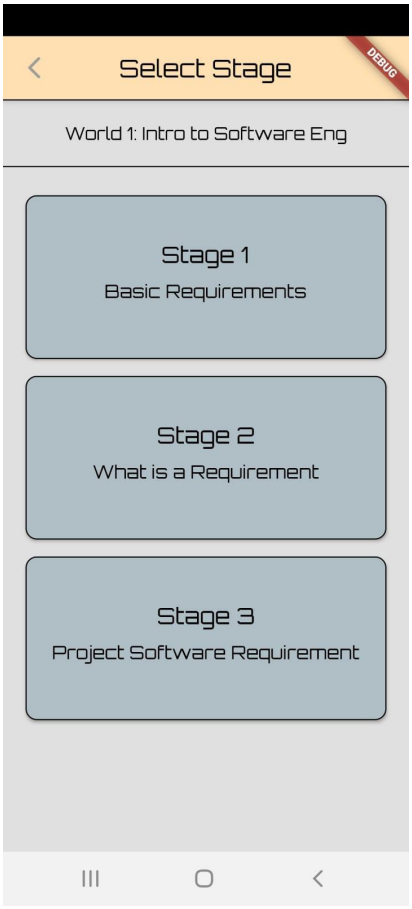


Figure 4.1.32 – Stage Selection

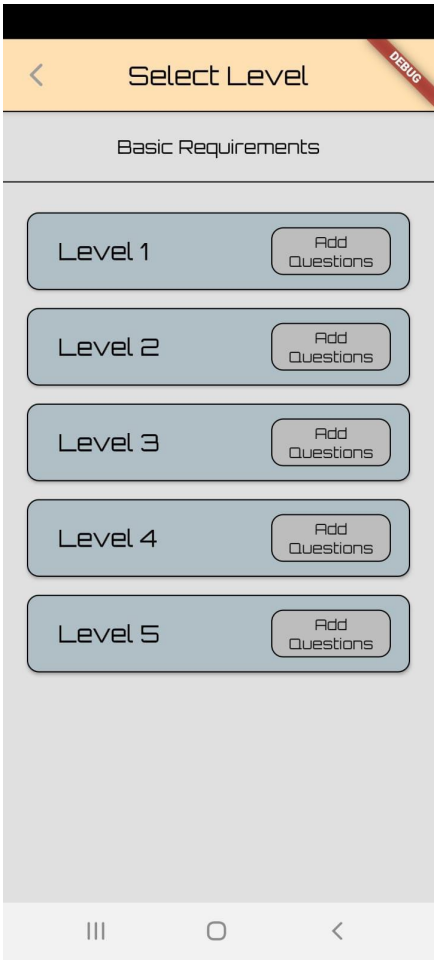


Figure 4.1.33 – Level Selection to Add Question

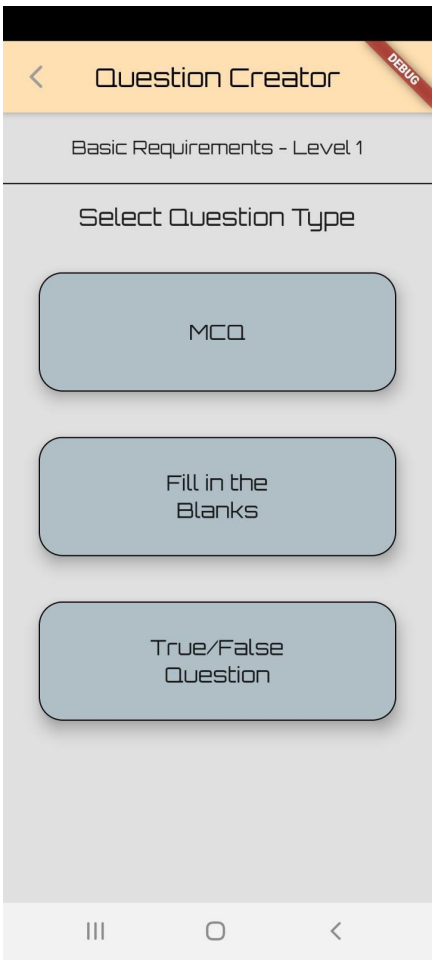


Figure 4.1.34 – Select Question Type to Add

Figure 4.1.35 – Add MCQ Question

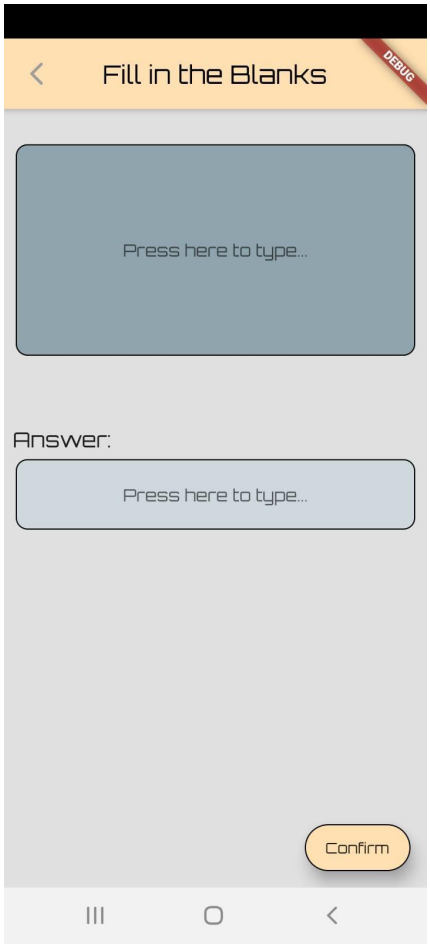


Figure 4.1.36 – Add Fill-in-the-blanks Question

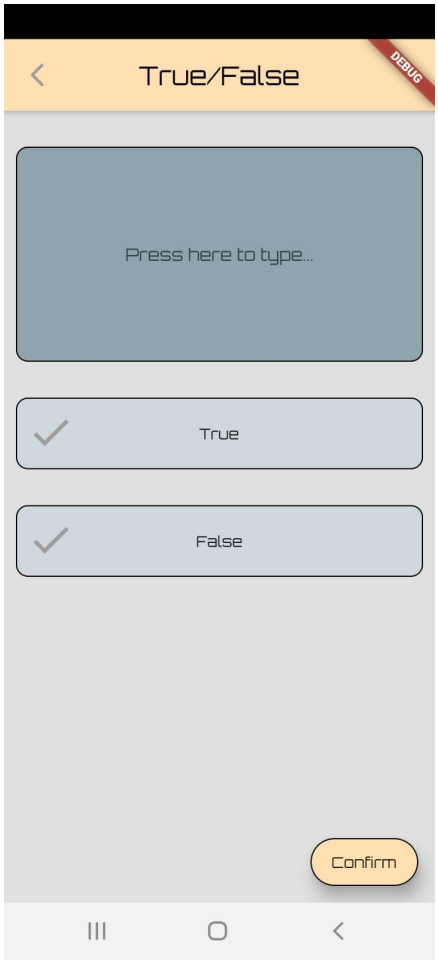


Figure 4.1.37 – Add True/False Question

3.2 Hardware Interfaces

As we have used the Flutter framework to build the game application, we are able to deploy on both android and IOS as mentioned earlier. Thus, the user will interact with the application mainly on the touchscreen of his phone.
For the game engine, the user shall interface with a directional joypad to move his character around. And to attack, the user can tap on a attack button to attack with his character

3.3 Software Interfaces

One of the software interfaces is towards the game engine, which was dependent on bonfire.
As for storage of databases, we have used Firebase to store data, and Firebase Auth to verify login for subsequent access to an account.

4. System Features

4.1 Registering

1. The system shall allow the user to register for an account.
    1.1. The system shall allow users to create an account through existing email or Facebook account.
        1.1.1. The password of the account shall have at least one capital letter.
        1.1.2. The password of the account shall have at least one lower case letter.
        1.1.3 The password of the account shall have at least one number.
    1.2. The system needs to send a notification email to users.
        1.2.1 The system shall notify users when they sign up for an account.
        1.2.2 The system shall request for verification in the email.
        1.2.3 The system shall create the account only when the account is verified.
    1.3. The system shall allow teachers to register for a teacher's account.
        1.3.1. The teacher shall enter a unique school-issued ID in order to create a teacher's account.

4.2 Main Menu

1. The system shall have an Adventure Game mode, PvP game mode, respective leaderboards, assignment tab, and settings in the main menu.

4.3 Adventure Game Mode

1. The system shall have three parts in hierarchical order in the adventure game mode.
    1.1 The system shall have six worlds, each representing the different phases of Software Development Life Cycle (SDLC)
    1.2 The system shall have different stages in each world, each stage representing a different concepts inside each cycle.

1.3 Inside each stage, the system shall have a dungeon with five levels, representing questions relevant to specific concepts with increasing difficulty.

2. The user shall be able to choose from 4 characters in the adventure game mode.

2.1 The user shall be able to choose a knight character.

2.3 The user shall be able to choose a mage character

4.4 Worlds inside Adventure Game Mode

1. The system shall only unlock World 1 while locking the remaining five worlds for first time users.
2. The system shall allow the user to explore unlocked worlds.

2.1 The user shall be allowed to enter unlocked worlds.

2.2 The system shall allow the user to explore stages inside each world.

3. The system shall not allow the user to explore locked worlds.

3.1 The system shall display a message “World is locked, please complete the world quiz prior to unlock!”

4. The player shall be able to unlock the next world by passing the world quiz for the current world. (i.e World 2 is only unlocked if the player has passed world 1 quiz).

4.1 The system shall set the passing score for the world quiz to be fifty percent.

4.1.1 There system shall set 20 questions inside each world quiz.

4.1.2 The system shall set the world quiz in MCQ format.

4.1.3 The system shall retrieve only the MCQ questions from the question bank.

4.1.3.1 The system shall have a question bank where teachers can add questions whenever they want.

4.1.3.1.1 The system shall allow the teacher to add any question type that they want in any world, stage and level.

4.2 The user shall be allowed to have multiple attempts to retry the world quiz.

4.3 The system shall have an exit button for users to return to the page on selection of worlds.

5. The system shall display a message “Congratulations, you have unlocked the next world!” if the user has achieved more than fifty percent for the world quiz.

5.1 The system shall have a button to redirect the user to the next world.

6. The system shall keep the next world locked if the user did not achieve more than fifty percent for the world quiz.

7. The user shall unlock the world progressively, from world 1 to world 6.
8. The system shall display the collective adventure score of students inside each of the world.

4.5 Stages inside Adventure Game Mode

1. After entering an unlocked world, the user shall be able to see the stages in a world.
    1.1 The system shall unlock all the stages in the world even if the user is entering the world for the first time.
    1.2 The user shall be able to choose the stage he/she wants to play.
    1.3 The system shall set the number of stages in a world to correspond to the number of concepts that could be taught for that topic (world).
2. The system shall allow the user to review the questions he has attempted for each level.
    2.1 The system shall show the user his highest score in each of the levels inside a stage.
    2.2 The system shall show the user the total combined scores of all his levels inside that stage.
    2.3 The system shall allow the user to view all his past attempts for each level
        2.3.1 The system shall show the user score out of ten for a level for all his past attempts.
            2.3.1.1 The system shall show the questions that he has answered correctly.
            2.3.1.2 The system shall show the questions that he has answered incorrectly.
                2.3.1.2.1 The system shall show the correct answer for all the questions for that attempt

3. If a user selects a stage, the system shall display the levels of the stage.
    3.1 The system shall display the users' progression in that stage by showing the level he Has unlocked.
    3.2 The system shall have five levels inside each stage.

4.6 Levels Inside Adventure Game Mode

1 The system shall use a question bank to set the questions in each level of a stage.

1.1 The system shall store the questions in a database.

1.1.1 The system shall sort the questions by the six phases of SDLC life cycle.

1.1.1.1 The system shall further sort the questions into topics of each phase.

1.1.2 The system shall sort the questions based on its difficulty level.

1.1.2.1 The system shall have difficulty levels ranging from one to five.

1.1.2.2 The teachers shall decide the level of difficulty for each question.

1.1.3 The system shall also have multiple types of questions.

1.1.3.1 The system shall have MCQ type of questions

1.1.3.2 The system shall have T/F type of questions

1.1.3.3 The system shall have fill in the blank type of questions

2. The system shall have ten questions in each level.

3. The system shall set the difficulty of the question corresponding to the level itself.

3.1 Inside each stage, the system shall have 5 levels and each level corresponds to the difficulty level. (Level 1 - Difficulty level 1, Level 2 - Difficulty level 2, … until Level 5 - Difficulty 5)

4. The system shall only unlock the first level if the user is entering for the first time.

5. The user shall be allowed to revisit the levels that he has already unlocked at any time.

5.1 The system shall also provide the information of the players playing history in that level.

5.1.1 The system shall show all past attempts for that level.

5.1.1.1 The system shall show the scores of all past attempts.

5.1.2 The system shall show the review for users' past attempts at that level.

5.1.2.1 The system shall show the questions that he had answered for each attempt.

5.1.2.1.1 The system shall provide a feedback/explanation for the questions that had been answered incorrectly.

6. The system shall create a dungeon for each level.

6.1 The system shall create 10 NPCs in the dungeon, each of whom will give out a question with difficulty corresponding to the level.

6.1.1 The system shall have 4 NPC giving out random MCQ questions from the question bank.

6.1.2 The system shall have 4 NPC giving out random T/F questions from the question bank.

6.1.3 The system shall have 2 NPC giving out random fill in the blanks questions from the question bank.

6.1.3.1 The system shall do some processing before matching the user answers to the actual answer stored in the database.

6.1.3.1.1 The system shall process the answer to ignore caps.

6.3.1.1.2 The system shall process the answer to ignore punctuation.

6.1.4 The system shall set the HP of the NPC to be 100.

6.1.5 The system shall set the attack power of NPC to be 25.

6.2 The user shall be able to attack the NPC (who has a fixed set of HP)

6.3 The system shall remove the NPC from the dungeon once the user has defeated it.

6.3.1 Upon defeating the NPC, the system shall ask the student a question.

6.4 The student shall complete the dungeon once he has defeated all 10 NPCs inside the dungeon.

6.4.1 The user shall be able to unlock the next level if he has won at least five of the challenges against the NPCs.

6.4.2 The system shall allow the user to unlock the next two levels if the user has won more than seven challenges.

6.4.3 If a user is in the fourth level, he shall only be allowed to unlock the final level (5th)

6.4.4 The user shall be allowed to retry the dungeon for that level.

7. The user shall be rewarded with points upon completing a dungeon.

7.1 The system shall reward the points based on the number of NPC that he has defeated.

7.2 The system shall update the total points a user has in the game if this is a new high score.

7.3 The system shall update the user total points in the leaderboard if a new high score is obtained

8. Depending on the character used, the system shall set a corresponding health point (hp) for that character as well as the skills for that character.

8.1 The system shall set the base hp of the warrior to be 100.

8.2 The system shall set the attack power of the warrior to be 25.

8.3 The system shall set the base hp of the mage to be 80.

8.4 The system shall set the attack power of the mage to be 40.

8.5 The system shall allow the warrior to attack with his swords.

8.6 The system shall allow the mage to cast fireball spells to attack the npc.

8.7 If a player dies, the system shall save all the questions that the user has attempted up to that point and store it into the database.

4.7 Leaderboards

1. The system shall come up with a leaderboard for adventure gamemode

1.1. The system shall come up with an overall leaderboard based on the total points that the user has accumulated throughout all the worlds in adventure gamemode.

1.1.1 The system shall sort the high score of the players from highest to lowest by default.

1.1.1.1 The system shall only display the scores of the top 50 players.

1.1.2 The system shall not allow the user to reset their points.

1.1.3 The system shall allow the user to search for a particular username to see his/her ranking on the leaderboard.

1.1.4 The system shall also display the user points and ranking at the bottom of the screen.

2. The system shall come up with a leaderboard for PVP gamemode

2.1. The system shall have a leaderboard with three different columns for the PVP gamemode.

2.1.1. The system shall have a column for the level name in the leaderboard.

2.1.2. The system shall have a column for the average rating out of 5 for each level in the leaderboard.

2.1.2.1 The system shall obtain the rating from players who has challenged that level.

2.1.3. The system shall have a column that stores the link to the level.

2.1.4. The system shall sort the levels by rating.

2.1.4.1 The system shall only display the 50 highest rated levels.

2.2 The user shall be allowed to search for a specific level.

2.2.1 The user shall be allowed to search for a specific level by its name.

2.3 The system shall display an individual level leaderboard when user selects the level name.

2.3.1 The system shall have three columns for this leaderboard.

2.3.1.1 The system shall have a column for the players name.

2.3.1.2 The system shall have a column for score

2.3.2. The system shall sort the leaderboard by the score.

2.3.2.1 The system will sort the leaderboard by the highest score.

2.3.2.2 The system will only display 50 players with the highest score.

2.3.3 The system shall also display the user points and ranking at the bottom of the screen.

4.8 Player Versus Player (PVP) Game Mode

1. The system shall allow users to challenge in the PVP mode regardless whether or not they have completed the adventure gamemode.

1.1 Users shall be able to challenge from a list of all created levels.

2. Users shall be able to design their own levels.

2.1 Users shall be able to specify the world that he/she wants in their designed level.

2.2 Users shall specify the number of questions of each type of question.

2.3 Users shall specify the difficulty of each type of question (from 1 to 5)

2.4 Users shall be allowed to share and publish the level they have designed on social media

2.4.1 Users shall be allowed to share and publish the level they designed on Twitter.

2.4.2 Users shall be allowed to share and publish the level they designed on Facebook.

2.4.3 Users shall be allowed to share and publish the level they designed on Telegram.

2.4.4 Users shall be allowed to share and publish the level they designed on Whatsapp.

2.5 The system shall create a dungeon with the configuration that has been designed by the student.

2.5.1 The system shall save this configuration (number of each type of question and difficulty) into the database.

2.6 The system shall allow the user to challenge the levels designed by other users.

3. After playing a level, The system shall allow users to rate the levels

3.1 The user shall be able to rate the level rating out of 5.

3.2 The user shall be allowed to rate the level after they play the level.

3.3 The system shall take the average of the total ratings given by all the users.

3.3.1 The system will rank the user-created levels by their rating in the leaderboard.

4.9 Teacher's Dashboard

1. The teacher shall be able to obtain a summary report of the student's progress in the game.

1.1 The system shall display the average scores of students in each part of the game of the world.

1.1.1 The system shall display the average scores of students in each of the worlds.

1.1.2 The system shall display the average scores of students in each of the stages.

1.1.3 The system shall display the average scores of students in each of the levels.

1.2 The system shall display the numbers of students at each part of the game.

1.3.1 The system shall display the number of students that have unlocked each world.

1.3.2 The system shall display the number of students that have unlocked each stage.

1.3.3 The system shall display the number of students that have unlocked each level.

1.4 The system shall display a summary on the parts of the game that are well mastered by students.

1.4.1 The system shall display the top three worlds that students have done well based on the total average points in that world.

1.4.1.1 The system shall display the top three stages inside each of the top three worlds.

1.5 The system shall display a summary on the parts of the game that the students have the worst mastery in.

1.5.1 The system shall display the worst three worlds that students have done the worst in based on the total average points in that world.

1.5.1.1 The system shall display the worst three stages inside each of the worst three worlds.

2. The teacher shall be able to assign assignments to the students.

2.1 The teacher shall be able to design the assignment by selecting the number of questions.

2.1.1 The teacher shall be able to choose the number of MCQ questions.

2.1.2 The teacher shall be able to choose the number of T/F type of questions

2.1.3 The teacher shall be able to choose the number of fill in the blank questions.

2.1.4 The teacher shall be able to choose the tutorial group that the assignment will be assigned to.

2.1.4.1 The students in the tutorial group shall receive the assignment in their assignment tab when they login.

2.1.4.1.1 The system shall then allow the student to attempt the assignment.

2.2 The teacher shall be able to choose which world the questions will be from

2.3 The teacher shall be able to further design the assignment by choosing the stages.

2.3.1 The teacher shall be able to select the stages that the questions will be from.

2.4 The teacher shall be able to choose the difficulty level of the questions assigned in the assignment.

2.5 The system shall select the questions from the question bank randomly according to the configuration selected by the teacher.

2.6 If the teacher does not choose any difficulty level, then the system shall take the default difficulty level as 1.

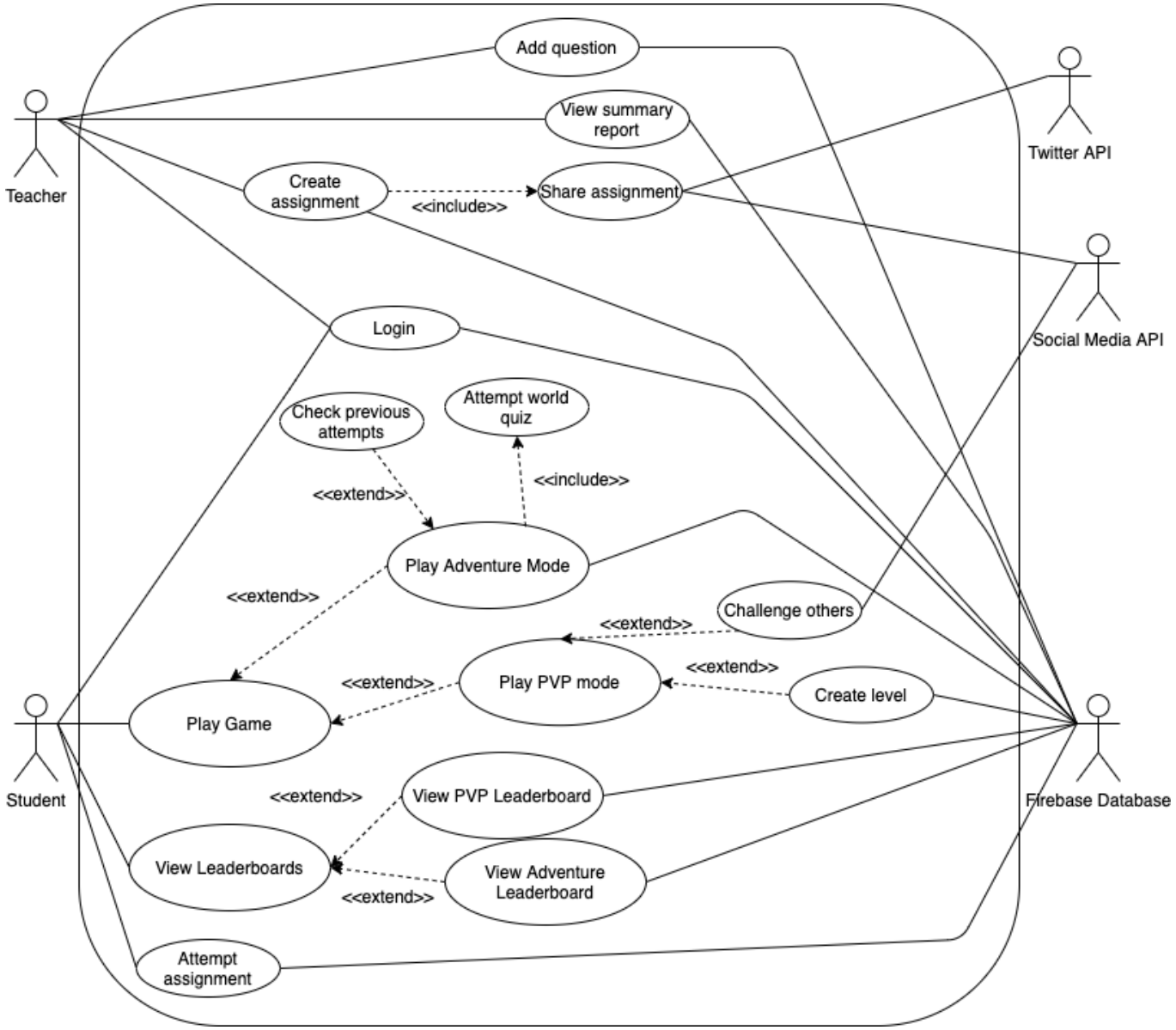
3. The teacher shall be able to send assignment links to students through social media.
    3.1 The teacher shall be able to give assignment links to students through Facebook.
    3.2 The teacher shall be able to give assignment links to students through Twitter.
    3.3 The teacher shall be able to give assignment links to students through Telegram.
    3.4 The teacher shall be able to give assignment links to students through Whatsapp

4. The teacher shall be able to add new questions to the game.
    4.1 The system shall display all the worlds available in the game
    4.2 The system shall display all the stages available in the game
    4.3 The system shall display all the levels available in the game that the teacher can add questions to
        4.3.1 The teacher shall be able to choose to add MCQ questions.
        4.3.2 The teacher shall be able to choose to add T/F type of questions
        4.3.3 The teacher shall be able to choose to add fill in the blank questions

.

5. Diagrams & Descriptions

5.1 Use Case Diagram



5.2 **Use Case Descriptions**

Use Case ID:	UC-01		
Use Case Name:	Login		
Created By:	Chris	Last Updated By:	Liew Hon Weng
Date Created:	1/2/2021	Date Last Updated:	20/4/2021

Actors	Student, Teacher
Description	User can log in to the game
Preconditions	1. User shall have a valid email address 2. User has selected the tab that the user will login to
Postconditions	1. User successfully signs in to an account
Priority	High
Frequency of use	Always
Flow of Events	1. User goes to the login page 2. System prompts the user to enter their login information 3. User fills in email particulars, selects if they are a teacher/student from the top bar and keys in their account password. 4. User clicks on the login button. 5. The system verifies the entered information in the database. 6. If the credentials match an existing account, the system will log the user in.

	7. The user will login successfully and see the home screen.
Alternate Flow	AF-6: Invalid login credentials 1. The user entered an invalid email, account or password. 2. The system will notify users with an error message. 3. Return to step 2. AF-7: Student tries to log in as teacher 1. The system will notify user with an error message. 2. Return to step 2. AF-8: Teacher tries to log in as a student 1. The system will notify user with an error message. 2. Return to step 2.
Exceptions	E1: Problems with Internet Connection. 1. The system shall notify the user of the problem with a message. 2. The system shall not proceed on with the login process. E2: The system is unable to validate the email address and password of the user. 1. The system shall notify the user of the error through the display. 2. The system shall display the main login screen. 3. The system shall prompt the user to key in their email address and password again.
Include	-

Use Case ID:	UC-02		
Use Case Name:	Student level design		
Created By:	Chris	Last Updated By:	Hon Weng
Date Created:	1/2/2021	Date Last Updated:	20/4/2021

Actors	Student
Description	Student can design a challenge for his peers
Preconditions	1. Student is logged in 2. Student shall have selected the PvP mode
Postconditions	1. Student successfully designs a challenge for his peers
Priority	Medium
Frequency of use	Occasionally
Flow of Events	1. Student selects PvP mode 2. Student clicks on ‘Design A Level’ 3. Student select the world, numbers of questions and the difficulty level of questions. 4. Student enters the name of their created world 5. Student confirms the creation of their level by pressing the ‘confirm’ button. 6. System verifies information and saves it into the database 7. System creates a dungeon of designed Level and updates it to the list of available

	dungeons. 8. Students are allowed to share the level they created with their friends through social media.
Alternate Flow	-
Exceptions	E1: Problems with Internet Connections 1. The system shall notify the user of the problem with a message. 2. The system shall not proceed on with the creation process. E2: Errors while retrieving data from the database 1. The system shall not proceed with the design of the challenge.
Includes	-

Use Case ID:	UC-03		
Use Case Name:	Adventure Mode Playthrough		
Created By:	Chris	Last Updated By:	Hon Weng
Date Created:	1/2/2021	Date Last Updated:	20/4/2021

Actors	Student
Description	Users can choose the worlds they would like to attempt.

Preconditions	1. Student is logged in 2. User shall have selected ‘adventure mode’
Postconditions	1. User is now able to interact with their selected world.
Priority	High
Frequency of use	Frequent
Flow of Events	1. System prompts the user to select an unlocked world. 2. Student selects the world they would like to attempt. 3. System brings the user to the selected world. 4. When the user has passed all stages in the world, the system prompts the user to take the world quiz. 5. Student attempt the world quiz. 6. System provides feedback on questions that students had answered incorrectly. 7. If student scored more than fifty percent for the world quiz, the system shall display a message “Congratulations, you have unlocked the next world!” 8. System unlocks the next world for the user.
Alternate Flow	AF-1: Student selects a world which they have not unlocked. 1. System displays “World is locked, please complete the world quiz in the previous world to unlock!” 2. Prompts the user to select a world which

	they have unlocked. 3. Return to step 1. AF-2: Student attempts the world quiz before they complete all the levels. 1. If they score more than fifty percent, system will unlock the next world. AF-7: Student scored lesser than fifty percent for the world quiz. 1. System keeps the next world locked. 2. Return to step 4.
Exceptions	E1: Problems with Internet Connections 1. The system shall notify the user of the problem with a message. 2. The system shall not proceed on with the game.
Includes	-

Use Case ID:	UC-04		
Use Case Name:	Stages and levels in adventure mode		
Created By:	Joshua	Last Updated By:	Hon Weng
Date Created:	1/2/2021	Date Last Updated:	20/4/2021

Actors	Student
Description	Users can select the stage they would like to attempt in the world they chose previously. Each stage will consist of exactly 5 levels, and users can

	start to challenge the levels starting from level 1.
Preconditions	1. Student is logged in 2. Student has already selected the world they would like to attempt.
Postconditions	1. Users playing in the dungeon based on the stage and level they have selected
Priority	High
Frequency of use	Very frequent
Flow of Events	1. System displays the stages in a world 2. User selects an unlocked stage they wish to attempt. 3. System displays the levels in the stage. 4. User selects the unlocked level they wish to attempt. 5. System shall redirect the user to a dungeon with 10 NPCs. 6. Student will attempt to beat the level by defeating the 10 NPCs. Every time they defeat a single NPC, user will be directed to a quiz consisting of a single question. 7. If user manages to obtain a score of more than 5, the next level will be unlocked.

Alternate Flow	AF-1: Student's character die in the challenge 1. A notice that 'your character has died' will appear on the screen. 2. Return to step 4 AF-2: Student answered less than 5 questions correctly. 1. System will not unlock the next level. 2. Return to step 4. AF-3: Student manages to answer 8 questions or more correctly in the level. 1. System will allow the user to unlock the next 2 levels. 2. Return to step 1. AF-4: Student has answered more than 8 questions correctly but they are already at level 4 1. System will only let the user unlock the 5th level. 2. Return to step 1.
Exceptions	E1: Problems with Internet Connections 1. The system shall notify the user of the problem with a message. 2. The system shall not proceed on with the game.
Includes	
Special requirements	
Assumptions	
Notes and issues	1. The user can reattempt an unlocked level anytime he/she wants and the system will spawn a new dungeon with different questions for the user.

	2. System will save the highest score that the student has gotten in the level.

Use Case ID:	UC-05		
Use Case Name:	Playing PvP mode		
Created By:	Joshua	Last Updated By:	Hon Weng
Date Created:	1/2/2021	Date Last Updated:	20/4/2021

Actors	Student
Description	Users can challenge a level designed by others inside PvP mode.
Preconditions	1. Student is logged in. 2. User have selected to play PvP mode.
Postconditions	1. User is playing PvP mode.
Priority	High
Frequency of use	Frequent
Flow of Events	1. System displays all the levels designed by others and sorts according to the level ratings given by challengers. 2. Users challenge a level designed by others. 3. System retrieves the questions from the Database saved to that respective created

	level. 4. System displays questions in MCQ, T/F and fills in the blank format for the challengers to answer. 5. The same format of 'adventure mode' dungeon will be used when users challenge the level. 6. Quiz ends when the challenger has submitted all his answers. 7. The challengers will be ranked on the number of questions they managed to obtain correctly. 8. System will store and update the score each challenger obtained in the database.
Alternate Flow	AF-1: Challenger dies while going through the dungeon. 1. The system will record whatever score the challenger obtained and update the database. 2. Return to step 1.
Exceptions	E1: Problems with Internet Connections 1. The system shall notify the user of the problem with a message. 2. The system shall not proceed on with the game.
Includes	-

Use Case ID:	UC-06		
Use Case Name:	Teacher's summary report		
Created By:	Joshua	Last Updated By:	Hon Weng
Date Created:	1/2/2021	Date Last Updated:	20/4/2021

Actors	Teacher, System
Description	Teacher chooses to generates a summary report of all the students playing the game.
Preconditions	1. Teacher shall be logged in to a teacher's account
Postconditions	1. Teacher can view the summary report of students
Priority	Medium
Frequency of use	Occasionally
Flow of Events	1. Teacher selects the adventure summary report. 2. System displays the average score of each world. 3. Teacher is allowed to select a specific world to view the summary report of said world. 4. System will then show the detailed summary report of each stage in the world. 5. System will show the best and worst

	performing stages and their respective average scores. 6. Teacher can enter a specific level and choose to view the summary report of said level.
Alternate Flow	AF-1: Teacher chooses to check the summary report for the best performing world. 1. System displays the best performing summary report for that world. 2. Return to step 2. AF-2: Teacher chooses to check the summary report for the worst performing world. 1. System displays the worst performing summary report for that world. 2. Return to step 2. AF-3: Teacher chooses to check the summary report for the best performing stage. 1. System displays the best performing summary report for that stage. 2. Return to step 4. AF-4: Teacher chooses to check the summary report for the worst performing stage. 1. System displays the worst performing summary report for that stage. 2. Return to step 4. AF-5: Teacher chooses to check the summary report for the best performing level. 3. System displays the best performing summary report of the level. 4. Return to step 6. AF-6: Teacher chooses to check the summary report for the worst performing level.

	5. System displays the worst performing summary report of the level. 6. Return to step 6.
Exceptions	E1: Problems with Internet Connections 1. The system shall notify the user of the problem with a message. 2. The system shall not proceed with the display.
Includes	-

Use Case ID:	UC-07		
Use Case Name:	Viewing PVP leaderboard.		
Created By:	Joshua	Last Updated By:	Hon Weng
Date Created:	1/2/2021	Date Last Updated:	20/4/2021

Actors	Student, System
Description	Users can view the PVP leaderboard
Preconditions	1. User shall be logged in.
Postconditions	1. User may view the leaderboard of the PVP gamemode.
Priority	Medium
Frequency of use	Occasionally

Flow of Events	1. User selects to view leaderboards. 2. User selects to view the PVP leaderboard. 3. System brings user to the PVP leaderboard where the levels created are ranked according to their ratings. 4. User may select the level they wish to view. 5. System brings user to a new page where challengers are ranked according to the score they obtained in the created level.
Alternate Flow	AF-4: Student chooses to challenge a created level. 1. Student clicks on the arrow to challenge the created level. 2. Student will be sent to the level.
Exceptions	E4: The system is unable to retrieve leaderboard data 1. The system will warn the student with a message of the problem 2. The system will not display the leaderboard
Includes	

Use Case ID:	UC-08		
Use Case Name:	Teacher assignments		
Created By:	Joshua	Last Updated By:	Hon Weng

Date Created:	1/2/2021	Date Last Updated:	20/4/2021

Actors	Teacher
Description	Teachers shall be able to create assignments for students.
Preconditions	1. Teacher must be logged into a privileged account.
Postconditions	1. Teacher creates a level as an assignment for their students.
Priority	High
Frequency of use	Frequently
Flow of Events	1. Teacher chooses to create an assignment. 2. System displays the level creator for teachers. 3. Teacher can choose the difficulty level and number of questions of the respective question type out of the three. T/F, mcq, short ans. 4. Teacher selects the world where the question will come from. 5. Teacher selects the create assignment button. 6. System creates the assignment level which will consist only of questions. level created will be in a quiz format and not a dungeon format. 7. Teacher can send the assignment via links to different social media platforms of their

	choice.
Alternate Flow	
Exceptions	E5: System is unable to create the level due to bad internet connectivity. 1. System will warn the teacher that they have no internet connection. 2. The stage will not be created
Includes	
Special requirements	
Assumptions	
Notes and issues	

.

Use Case ID:	UC-09		
Use Case Name:	Viewing adventure mode leaderboard		
Created By:	Hon Weng	Last Updated By:	Hon Weng
Date Created:	20/4/2021	Date Last Updated:	20/4/2021

Actors	Student, System
Description	Users can view the adventure mode leaderboard.
Preconditions	1. The user must be logged in.
Postconditions	1. The user can view the adventure mode

	leaderboard.
Priority	Medium
Frequency of use	Occasionally
Flow of Events	1. User selects to view leaderboards. 2. User selects to view the adventure mode leaderboard. 3. System shows the leaderboard of the adventure mode where users are ranked by points gained in adventure mode.
Alternate Flow	AF-1: The user searches for another user on the leaderboard 1. The rank of the searched user will be shown on the leaderboard. 2. Return to step 3.
Exceptions	E5: System is unable to load the leaderboard due to bad internet connectivity. 1. System will show a warning that there is no internet connection. 2. Leaderboard will not be displayed.
Includes	
Special requirements	
Assumptions	
Notes and issues	

Use Case ID:	UC-10

Use Case Name:	Student attempts assignment created by teacher.		
Created By:	Hon Weng	Last Updated By:	Hon Weng
Date Created:	20/4/2021	Date Last Updated:	20/4/2021

Actors	Student, System
Description	Student attempts an assignment assigned to them by a teacher.
Preconditions	1. Student must be logged in.
Postconditions	1. Student attempts the assignment created by teacher.
Priority	High
Frequency of use	Frequently.
Flow of Events	1. Student chooses to attempt teacher assignment on the home screen. 2. System will display all the assignments the student's teacher has assigned to them. 3. Student can choose to do any assignment assigned to them. 4. When student chooses the assignment, system will load the stored assignment from the database. 5. Student will attempt the level, and will obtain a score when level is complete and all questions are answered. 6. System will record their score and store it in the database for teacher to view.

	7. System will exit the assignments page.
Alternate Flow	AF-1: Student quits the assignment half way 1. The system will record their score and save their progress into the database. 2. When user re-attempts the assignment, they will start from the question they left off. 3. When all questions are done, return to step 6.
Exceptions	EX-1: Student tries to resubmit the assignment. 4. The system will not allow student to redo the assignment.
Includes	
Special requirements	
Assumptions	
Notes and issues	

Use Case ID:	UC-11		
Use Case Name:	Teacher adds a question into the question database		
Created By:	Hon Weng	Last Updated By:	Hon Weng
Date Created:	20/4/2021	Date Last Updated:	20/4/2021

Actors	Teacher, System
Description	Teacher uses the add question function to add a question into the question database.
Preconditions	1. Teacher must be logged in to a privileged account.
Postconditions	1. A question is added into the question database.
Priority	High
Frequency of use	Frequently
Flow of Events	1. Teacher selects the add question button in the home page. 2. The system will prompt teacher to select the world they wish to add the question to. 3. The system will prompt teacher to select the question difficulty level and the question type. 4. Teacher will fill in the question they wish to add. 5. System will save the question and add the question into the question database.
Alternate Flow	AF-1: The teacher selects question type ‘mcq’ in step 3. 1. System will provide 4 additional text boxes for teacher to key in options. 2. Teacher will be prompted to input 4 options to their mcq question. 3. Return to step 4.
Exceptions	E5: System is unable to store the question into the

	database due to poor internet connectivity. 1. System will display an error message
Includes	
Special requirements	
Assumptions	
Notes and issues	

6. Non-Functional Requirements

Non-Functional Requirements

Reliability

1.1. The system shall have at least 99% uptime over its entire lifespan

1.1.1. The system shall always be available unless it is due to an acceptable downtime (e.g. System update)

1.1.2. Acceptable system downtime shall not occur more than once a week, unless it is a critical patch

1.1.3. An acceptable system downtime shall not take more than 4 hours

1.1.4. An unpredicted system downtime shall not affect the system for more than 4 hours

1.2. The system shall not deviate from any use case scenarios

1.2.1. The functionalities of the system shall be intact with the proposed functional requirements

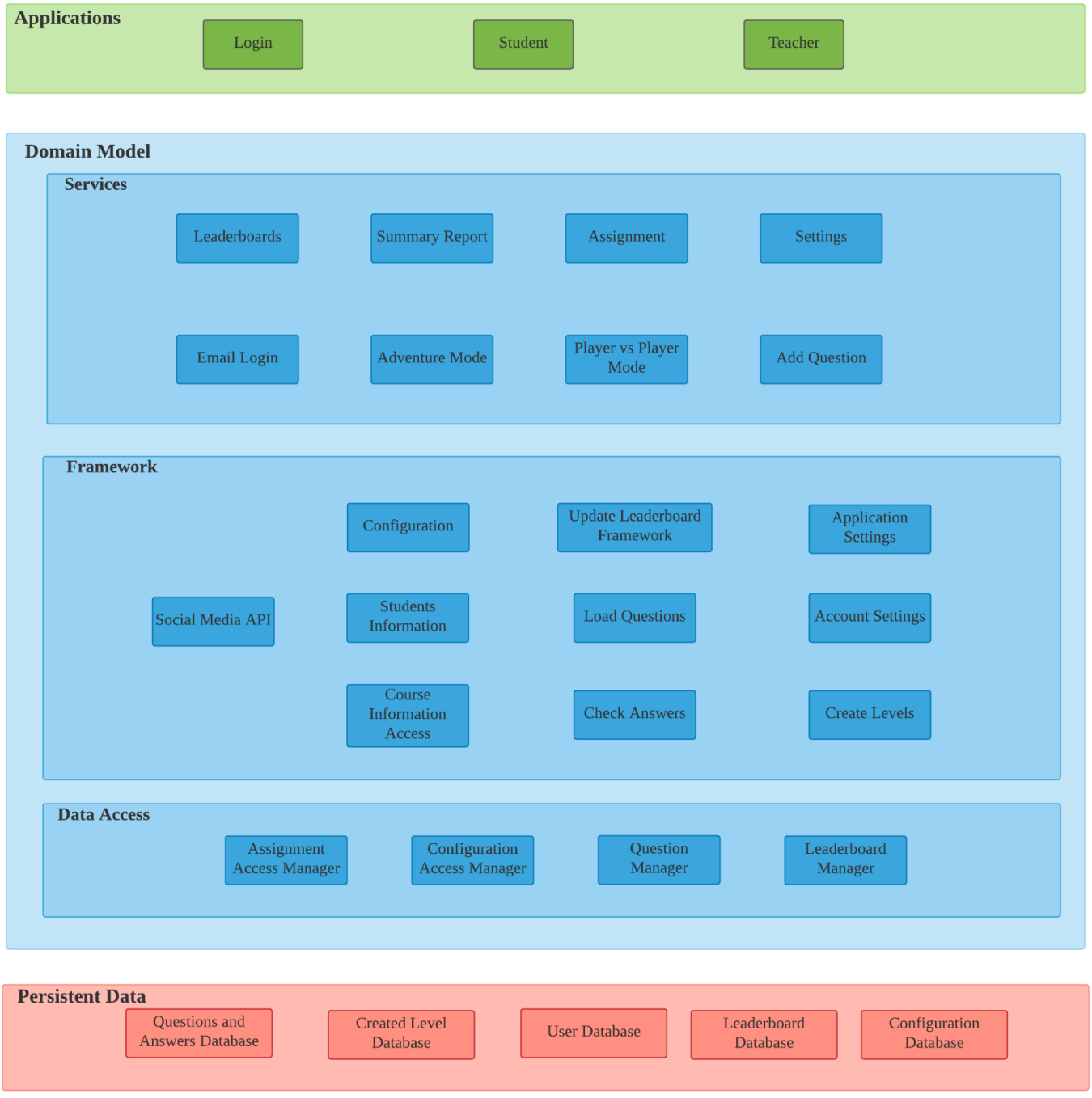
1.2.2. The system shall experience a 99% probability of success when executing a feature (i.e. the feature shall perform as expected 99% of the time)

Performance

2.1 The system shall always maintain a minimum of 60 frames per second.

2.1.1 The CPU of the system shall be newer than Quad-core 2.7 GHz Krait 450

7. Candidate Architecture Style



7.1 Rationale for Candidate Architecture

7.1.1 Our Approach

Software Architecture is considered as a description of the high level structure of a software system that includes architectural elements such as Components and the interactions between them which are called Connectors. We first begin by analysing the advantages and disadvantages of the various architectural styles, then followed by overall analysis and conclusion of the chosen architecture style.

7.1.2 Compared with other Architecture Styles

Architecture Style	**Advantages**	**Disadvantages**
Independent Components	1. Scalability - Many users can play the game and communicate with the server 2. Concurrency - Processes in the system can run independent of each other at the same time. 3. Accessibility - Server may not be physically close to clients but is still able to be accessed	1. Overloading - If clients request data from server simultaneously, it might get overloaded 2. Fault Intolerant - If server goes down, all clients will be unable to access data
Data Flow (pipe-and-filter)	1. Modularity - All components are independent of one another. 2. Reusability - We can reuse different components in different processes because they are independent of one another. 3. Maintenance - Easy to maintain as components can be added and modified without	1. Synchronization - Difficult to synchronize the different filters as they need to follow a multi-programming paradigm. - There is a possibility the components will read and write the same data simultaneously. - For synchronization, there is a need to avoid race condition.

	affecting the other components in the process. - This is because the components are not aware of their upstream and downstream components. - A component can be inserted as long as we know the interface. 4. Concurrency - All components are active processes by default. - All components can run independently of each other at the same time. 5. Performance - The advantage where components can run concurrently allows us to achieve parallelism. - This results in increased performance.	2. Bottleneck - May force a lowest common denominator on data transmission - This leads to the performance being dominated by the most time consuming component
Call & Return (Layer System)	1. Abstraction - Supports designs with increasing levels of abstraction. - This allows a problem to be divided into a sequence of incremental steps. 2. Enhancements - Supports enhancements by adding new components. - Layers of Isolation: Any changes to a layer's function will only affect that particular layer and its associated layers. 3. Reusability - Lower layer's components can be reused while the upper layer's components remain intact.	1. Applicability - Layered System Architecture cannot be applied to all the systems. - This is because not all systems can be easily structured in a layered format. - Even if we are able to structure the logic in layers, the need for a better performance might result in closer coupling of logically high- level functions with their low-level implementations. 2. Level of Abstraction

	- Allows different implementations of the same layer to be used interchangeably with the condition that they support the same interfaces to their adjacent layers. 4. Testability - Easy to test systems that are structured in layers due to well-defined components and limited component scope.	- Difficult to find the correct level of abstraction for the system. 3. Performance - Lower Performance - This is because a request for Layer X would have to go through all the other layers that are above it. - This adversely affects the performance as the request has to go through irrelevant layers.

7.1.3 Rationale for choosing Layered System Architecture

As mentioned previously, in order to select the most appropriate architecture style for our App, we would need a set of criteria or Quality Concerns to aid us in our decision making process. As such the 3 main Quality Concerns we focused on are:

1. Modifiability - ability to make changes to the system
2. Reusability - ability for one component to be used by other components
3. Performance - speed of the system

Keeping these 3 quality concerns in mind, we had decided to not use the pipe-and-filter architecture style despite it allowing components to be reused and high performance. This is because, the key features of the pipe-and-filter architecture style are as follows:

1. The input and output flow continuously through the system.
2. The components include “Filters” which are active components
3. The connectors include “Pipes” which are mere channels through which data flows through.

However this is not the case with our app, the inputs and outputs do not flow continuously through our system and the components of our systems are activated through function calls hence, our components are

passive components that might store internal data state and our connectors include function calls through which control and maybe data is passed on. In addition, the pipe-and-filter style might not be the best style when it comes to interactive applications. Due to the pipelining structure, we will have to restart the process every time a user faces an error.Hence, we have chosen not to implement the pipe-and-filter style.

The architecture style Batch Processing was not chosen for the same reasons as not choosing the pipe-and-filter architecture. Another additional reason for not choosing is that the execution of the components happens in a sequential manner thus compromising performance.

Next, the reason why we did not choose independent components is that due to the components being independent, when a component sends a message to another component, it cannot assume that the component that is receiving the message will respond to it. To overcome this problem, interaction protocols must be designed and implemented for all components to follow. This would result in overhead costs in the system. In addition, components of the app can make requests to the server simultaneously so it might get overloaded. Also there is heavy reliance on the server, as if the server goes down all other components will not be able to communicate with the server. Hence, we did not choose independent components.

Hence, the most promising architecture style happens to be the Layered System Architecture which is a substyle of the Call and Return Architecture. We have chosen Layered System architecture style over the Main Program with Subroutine style due to the fact that the Layered style supports design based on increasing levels of abstraction. This allows us to partition a complex problem into incremental steps. For example, having different worlds, stages and levels. To be specific, we have selected the Closed Layered Architecture Style which means that the subroutines in the higher layer can call only the subroutines in the lower layer.

In addition, the Layered Architecture Style fulfills the criteria of Quality Concerns as follows:

1. Modifiability - The subroutines are grouped into layers, hence any changes to a subroutine would only affect that subroutine.
2. Reusability - The subroutines in the higher layer can call any subroutines in the lower layer as such, a subroutine in the lower layer can be called by 2 different subroutines in the higher layer.
3. Performance - There might be overhead costs as a subroutine would have to go through unnecessary layers to get to a subroutine in the lower layer. However, our ap only has 3 main layers and hence performance is not heavily impacted adversely.

Hence we have decided to select the Layered Systems Architecture.

As seen in Figure 1.0, we have 3 main Layers, Applications, Game Engine and Persistent Data. The Layer Game Engine in turn has 3 layers inside of it. These 3 nested layers are Services, Frameworks and Data Access Layer. The components are mostly Passive such as Computational Components, ie: Passive Components that store Data internally while the connectors are Function or Procedural Calls. We have segregated each layer such that each layer manages its own collection of tasks, procedures and controls. This makes the software useful for maintenance and it is easy to delegate different tasks to them.

We have opted to break the Game Engine Layer into three major layers, Framework, Services and Data Access. The lower layer provides the underpinning infrastructure that drives them and consists of structures for question and level access, checking of answers and student information. The Data Access Layer uses the data from the relevant Databases and does the querying for the necessary functions. The subroutines from the above layers such as Framework and Services calls in the subroutines from this layer. A list of the components in the respective nested layers of the Game Engine Layer can be seen under the Subsystem Interface.

A third, auxiliary layer called Persistent Data is located below the layer of the Framework. It includes the databases that the program needs, such as the databases for student and created levels. This allows for a degree of abstraction, as the user faces the services layer, while the layers of the Framework and Permanent Data are not seen by the users.

Attribute Architecture	***Independent Component***	***Pipe and Filter***	***Layered***
Handling Interactive Application	+	-	-
Change in data representation	+	-	+
Reuse	+	+	+
Handling of overloading of server	-	+	+
Performance	-	+	-

7.2 Subsystem Interface

Game Engine

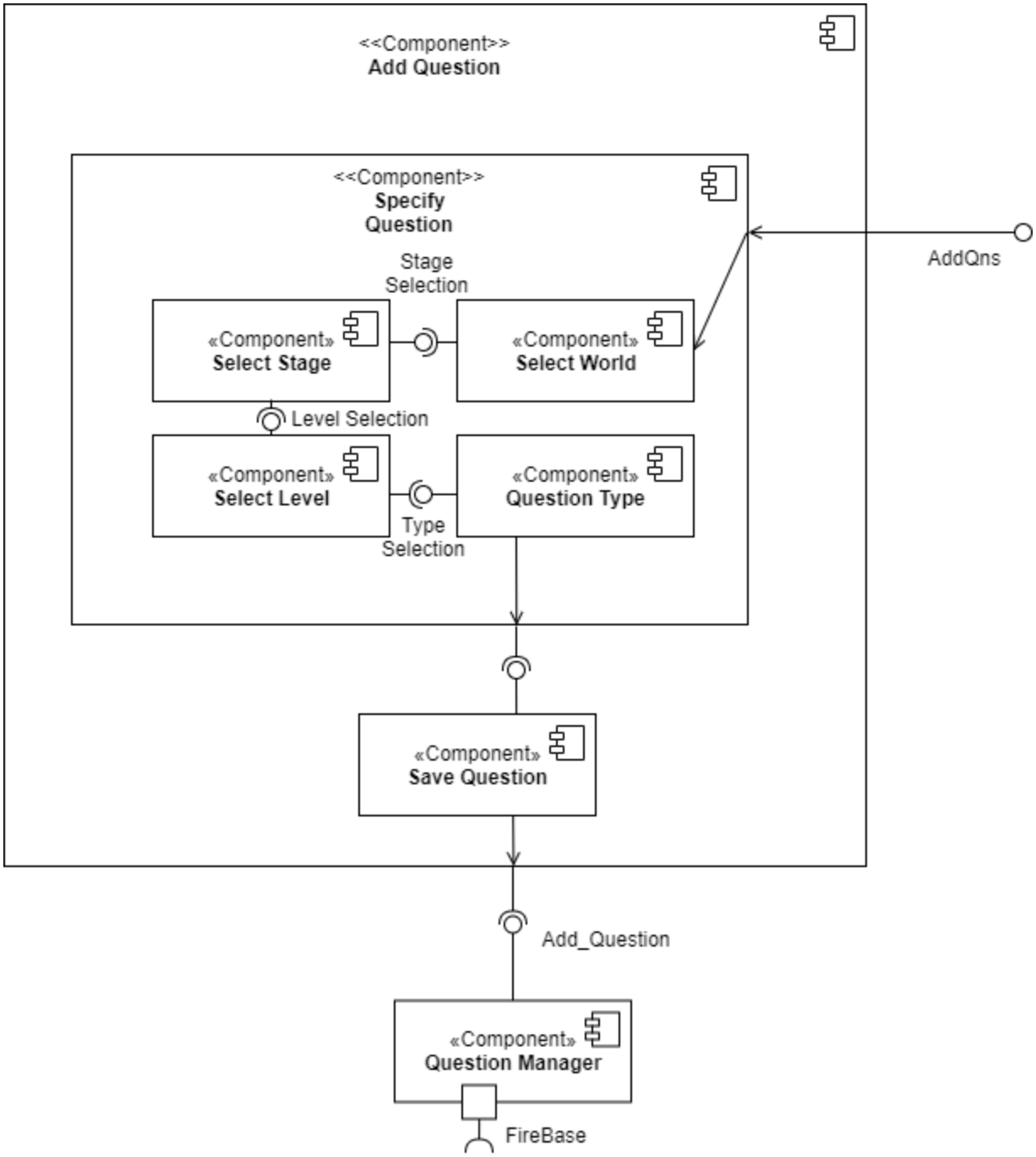
The Game Engine Layer contains all the components that are responsible for implementing the main functional requirements of the app. As observed from our Layered Structure Architecture Diagram, we have 3 main Sublayers in the Game Engine Layer, mainly:

1. Services
2. Framework
3. Data Access

what functions each component uses, and other components, parameters that are passed,
The below table shows the names of the Sublayers and their respective subroutines.

Services	**Frameworks**	**Data Access**
Email Login	Create Levels	Assignment Access Manager
Settings	Social Media API	Leaderboard Manager
Leaderboards	Account Settings	Configuration Access Manager
Adventure Mode	Students Information	Question Manager
Player Vs Player Mode	Configuration	
Assignment	Update Leaderboard Framework	
Add Question	Load Questions	
Summary Report	Check Answers	
	Course Information Access	
	Application Settings	

Add Questions



This component enables the teachers to create and add questions to the database for the students to answer. It contains subcomponents ‘Select Properties’ which in turn contains 4 subcomponents:

1. Select World
2. Question Type
3. Select Level
4. Select Stage

Component it interacts with: Teacher Component in Applications layer, Configuration and Load Questions, Main Menu
Function Calls and parameters:

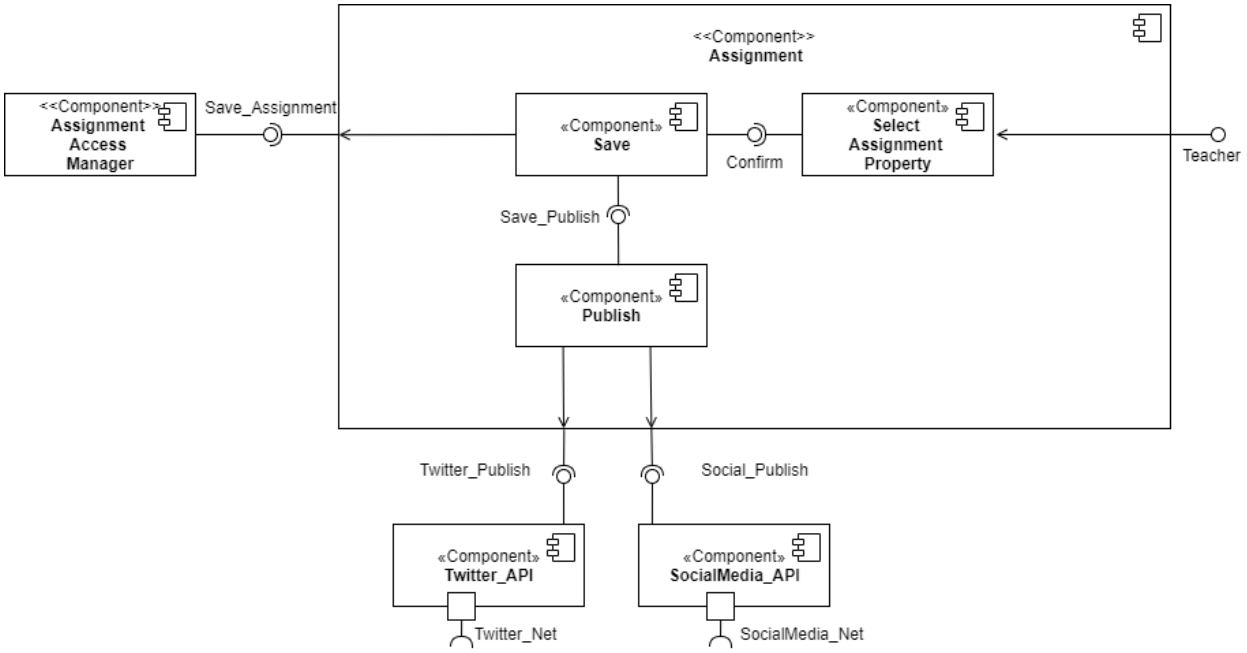
	Name	**Parameters**	**Functionality**	**Usage Scenarios**	**Design Rationales**
1	createNewTrueFalseQn()	int wNumber, int sNumber, int lNumber, String question, bool answerQ	Create True/False questions and save it in Firebase	When user wants to add new question into question bank	So that questions will not get outdated and students can have an accurate learning experience
2	createNewFITBQn()	int wNumber, int sNumber, int lNumber, String question, String answerQ	Create Fill-In-The-Blanks questions and save it in Firebase	When user wants to add new question into question bank	So that questions will not get outdated and students can have an accurate learning experience
3	createNewMCQQn()	int wNumber, int sNumber, int lNumber, String question, List<String> answers, String answerQ	Create MCQ questions and save it in Firebase	When user wants to add new question into question bank	So that questions will not get outdated and students can have an accurate learning experience
4	getIdOfAllMcqWorldLevel(), getIdOfAllTFWorldLevel(), getIdOfAllOEWorldLevel	int worldName, int levelName	Retrieve all questions in a level for a specific world	When questions needs to be retrieved	Questions are required for PvP and PvE
5	getIdOfAllMcqQuestions(), getIdOfAllOEQuestions(), getIdOfAllTFQ	int worldName, int stageName, int levelName	Retrieve all questions	When questions needs to be retrieved	Questions are required for PvP and PvE

	uestions(),				

Teacher's Assignments

Teacher's Assignments allows the Teachers to create an assignment by selecting the questions from the database and assigning them to students.It contains 3 components composed inside of it:

1. Save,
2. Select_Property
3. Publish



Component it interacts with: 'Teacher' component from the Applications layer, Configuration, Facebook API, Twitter API, Load Questions

Function Calls and parameters:

	Name	**Parameters**	**Functionality**	**Usage Scenarios**	**Design Rationales**
1	shareAssignment()	NIL	Share assignment created by Teacher via various social media APIs including Twitter and WhatsApp	When Teacher wants to share the created assignment via social media	So that Teachers can let Students know about new assignments

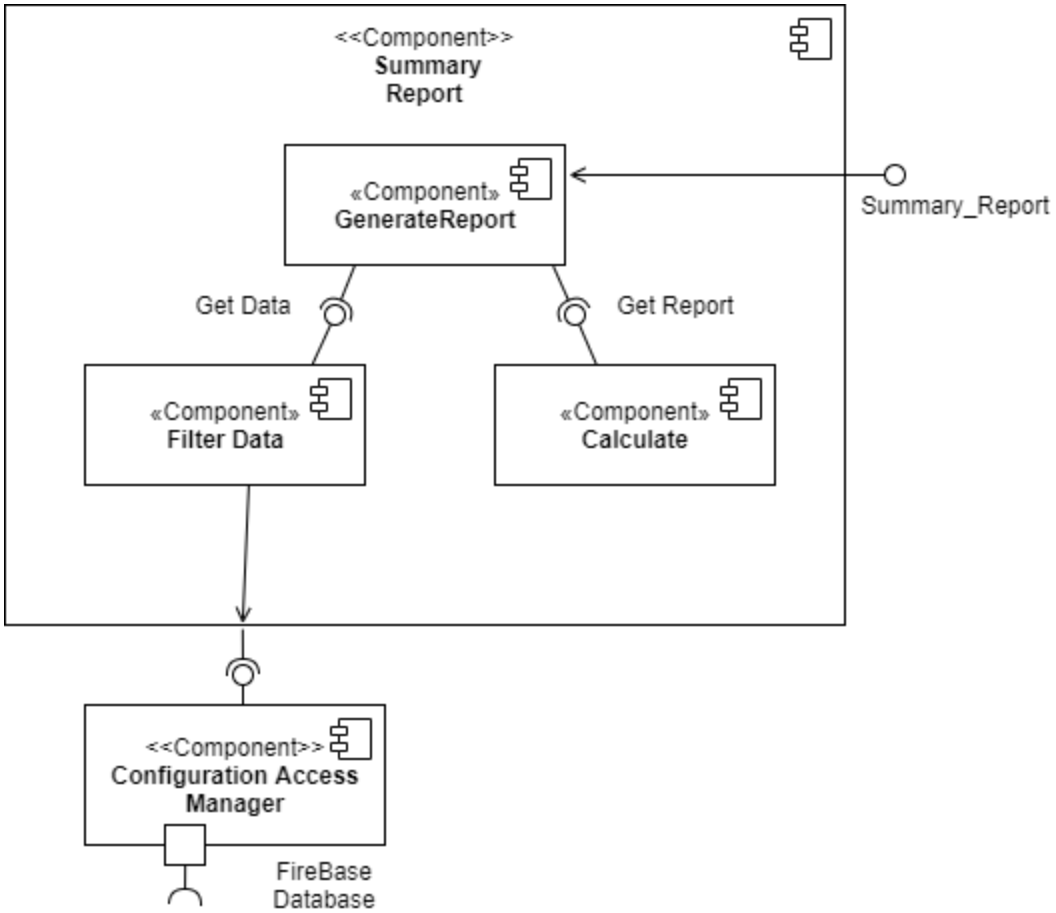
2	savePVPConfigurationForStudent()	int mcqDifficulty, int OEDifficulty, int TFDifficulty, int worldSelected, int noOfMCQ, int noOfOE, int noOfTF, String createdLevelName	Create new PvP level	When teacher wants to create new PvP level as an assignment for Students	So that Teachers can create a PVP level assignments easily
3	giveAccessToTutorialGroup()	String levelName, String tutorialGroupNum	Update tutorial group after world selection	When teacher wants to create new PvP level as an assignment for Students	So that Teachers can create a PVP level assignments easily

Teacher's Summary Reports

Teacher's Summary Report includes 3 subcomponents:

1. GenerateReport
2. FilterData
3. Calculate

Which aids in preparing a Summary Report for the Teachers to have an overall idea of their students' performance.



Component it interacts with:'Teacher' in the Applications Layer, Main Menu, Course Information Access

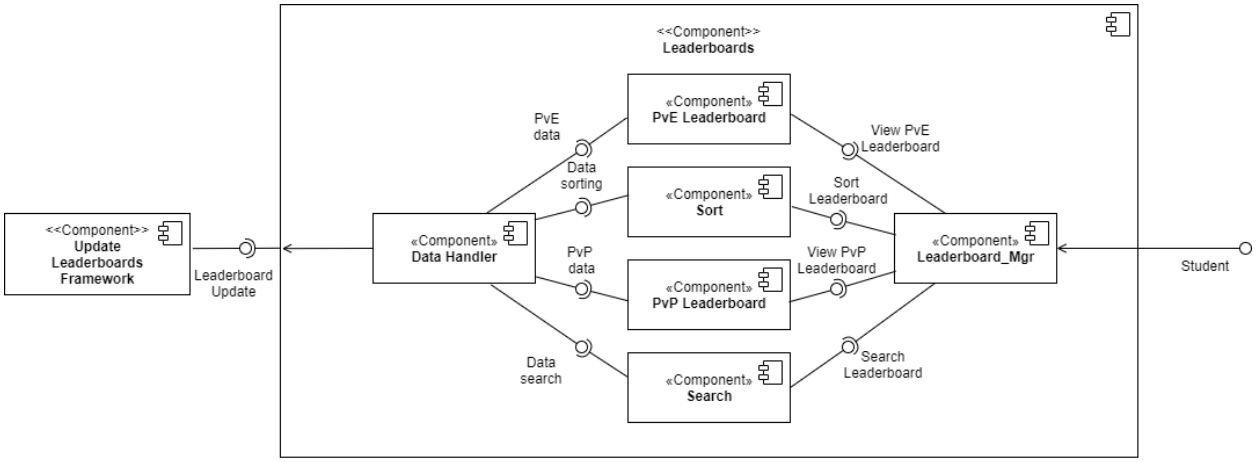
Description of Function Calls:

	Name	**Parameters**	**Functionality**	**Usage Scenarios**	**Design Rationales**
1	SummaryReport Controller()	-	Calls the relevant summary report based on the world, stage and level.	When the teacher wants to analyse the students' performance according to the world, stage or/and level.	This function was designed to allow the easy display of the different types of summary reports.
2	setSelectedWorld()	String world	To set a value for the variable stage.	When the teacher chooses to view the students' performance according to the	To allow flexibility in designing the View Summary

				world.	Report functionality.
3	getSelectedWorld()	-	Fetches the selected value of world.	When the teacher chooses to view the students' performance according to the world.	To allow flexibility in designing the View Summary Report functionality.
4	getWorldAverage()	-	Gets the average scores for a world from Firebase	When the teacher chooses to view the students' performance according to the world.	To allow flexibility in designing the View Summary Report functionality.
5	setSelectedStage()	String stage	To set a value for the variable stage.	When the teacher chooses to view the students' performance according to the stage.	To allow flexibility in designing the View Summary Report functionality.
6	getSelectedStage()	-	Fetches the selected value of stage.	When the teacher chooses to view the students' performance according to the stage.	To allow flexibility in designing the View Summary Report functionality.
7	getStageAverage()	-	Gets the average scores for a stage from Firebase	When the teacher chooses to view the students' performance according to the stage.	To allow flexibility in designing the View Summary Report functionality.
8	setSelectedLevel()	String level	To set a value for the variable level.	When the teacher chooses to view the students' performance according to the level.	To allow flexibility in designing the View Summary Report functionality.

9	getSelectedLevel()	-	Fetches the selected value of level.	When the teacher chooses to view the students' performance according to the level.	To allow flexibility in designing the View Summary Report functionality.
10	getLevelAverage()	-	Gets the average scores for a level from Firebase	When the teacher chooses to view the students' performance according to the level.	This function was designed to allow Teachers to create assignments.

Leaderboards



This component shows the top 50 students PvE and PvP scores, respectively, against one another.
Sub components include:

1. LB_Mgr
2. PvE LB
3. Filter

4. PvP LB
5. Search
6. Data Handler.

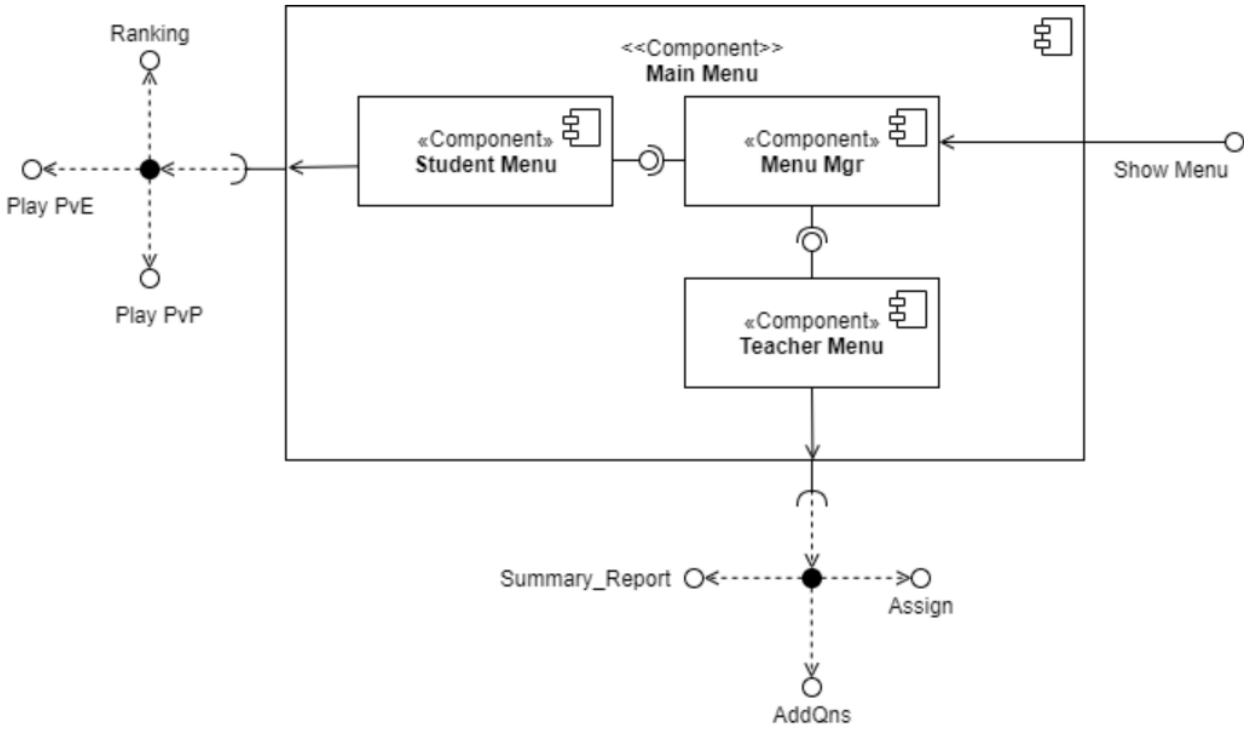
Component it interacts with: Student in the Application layer, Students Information, Main Menu, FireBase_API

Description of Function Calls:

No.	Name	Parameters	Functionality	Usage Scenarios	Design Rationales
1	AdventureLeaderBoardController()	-	Returns users' PvE rankings based on student data	When user wants to know the PvE rankings in the game	To show the level of achievement amongst different players, based on PvE scores earned
2	loadStudentInfo()	String name, int score	Returns students' name and score for PvE	To retrieve student data for display in PvE Leaderboard	If there exists student data, it can be displayed in PvE Leaderboard
3	PvPLeaderBoardController()	-	Returns users' PvP rankings based on ratings of the levels created	When user wants to know the PvP rankings in the game based on ratings of levels	To show the level of achievement amongst different players, based on PvP level ratings earned
4	loadLevelInfo()	String title, String nameOfCreater, double ratings	Returns title of level, name of creator and ratings for PvP	To retrieve student and level data for display in PvP Leaderboard	If there exists student data, it can be displayed in PvP Leaderboard
5	LevelLeaderBoardController()	-	Returns users' PvP level rankings based on time taken and score earned	When user wants to know the PvP level rankings in the game based on time taken and/or score earned	To show the level of achievement amongst different players, based on PvP individual level's time and score
6	setSelectedLevel()	String level	To set a value for the variable level	When level is required to render corresponding data	User needs to retrieve rankings based on levels in PvP
7	getSelectedLevel()	-	Returns selected level	When level is required to render corresponding data	User needs to retrieve rankings based on levels in PvP

8	getName()	String leaderBoardType, int index	Returns name of users based on time or score	When user wants to know the identity of players in PvP level leaderboard based on time or score	User wants to know the top players in that particular PvP level
9	getScores()	String listName, int index	Returns scores of users based on scores	When user wants to know the identity of players in PvP level leaderboard based on time or score	User wants to know the top players in that particular PvP level
10	getTimes()	String listName, int index	Return time used by users based on time	When user wants to know the identity of players in PvP level leaderboard based on time or score	User wants to know the top players in that particular PvP level
11	sortMap()	-	Map the name, time and score together	When user wants to know the identity of players in PvP level leaderboard based on time or score	User wants to know the top players in that particular PvP level
12	itemCount()	-	Return top 50 players of that level	When user wants to know the identity of players in PvP level leaderboard based on time or score	User wants to know the top 50 players in that particular PvP level

Main Menu



This component displays the main page (the page after logging in) for both Students and Teachers.
Sub components include:

1. Menu Mgr
2. Student Menu
3. Teacher Menu

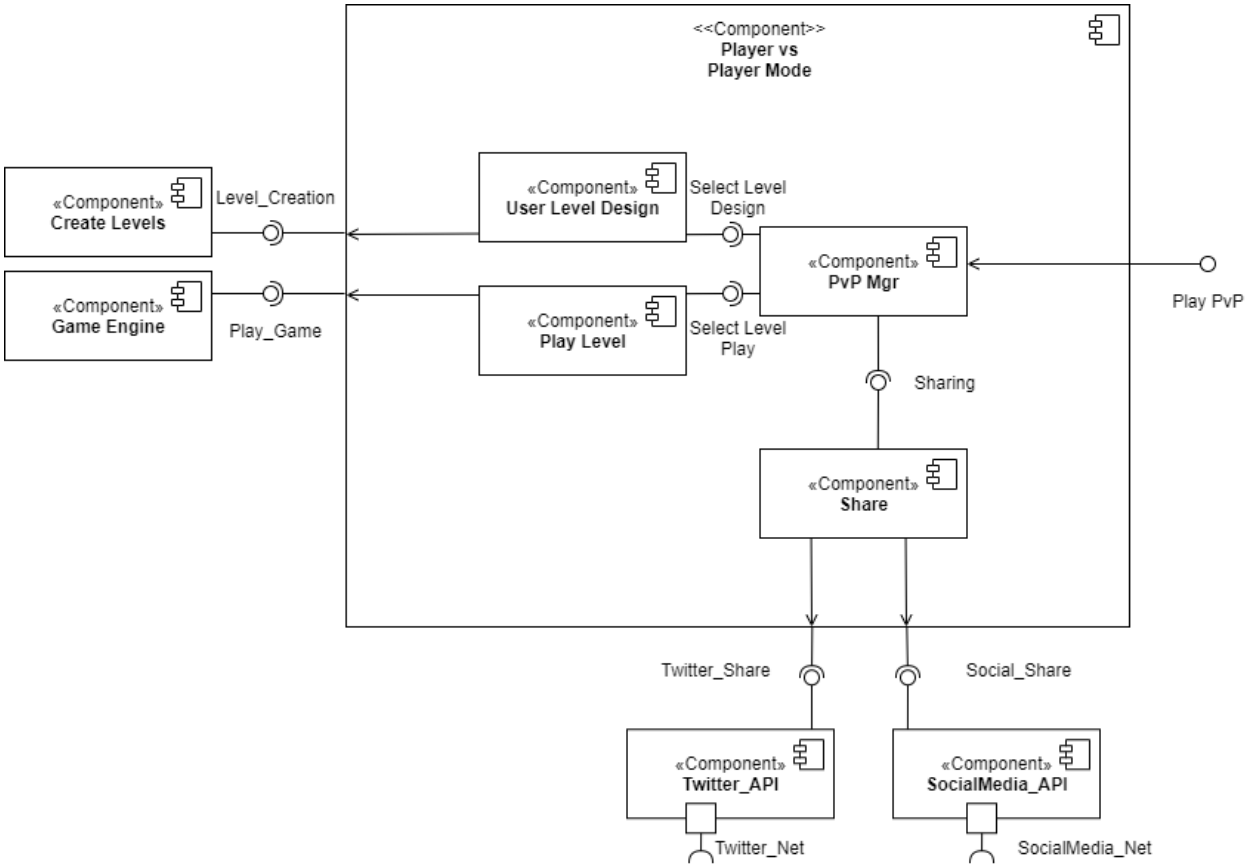
Component it interacts with: Login, Adventure Mode, PvP Mode, Summary Report, Assignment, Add Question

Description of Function Calls:

No.	Name	Parameters	Functionality	Usage Scenarios	Design Rationales
1.	main()	NA	Entry point of the entire Flutter application	Everytime the Flutter application is run, this main.dart will be the entry point	To ensure all the backend services are initialized and the starting point of the app is properly.

T

Playing PvP Mode



This component shows all the available options when a Student selects the PvP Mode, from creation of a level, playing a level to sharing of the results.
Sub component includes:

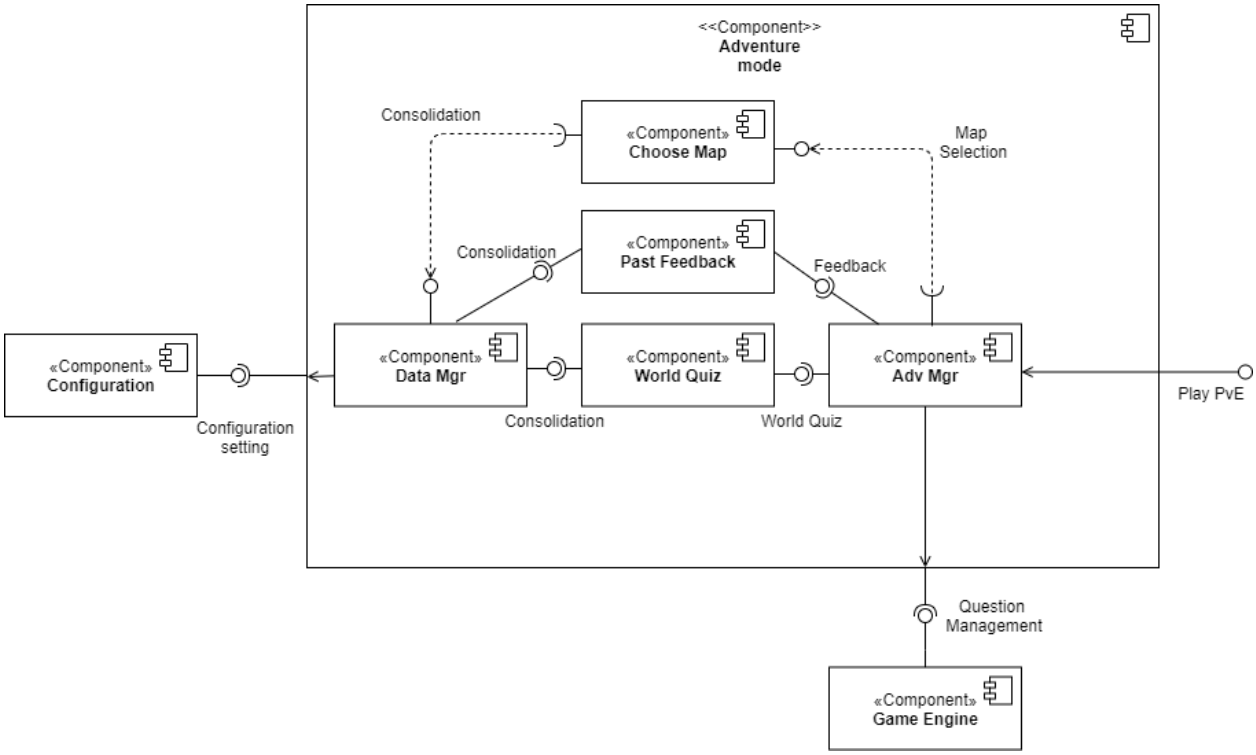
1. PvP Mgr
2. Share
3. Create Level
4. Play Level
5. Data handler

Component it interacts with: Student in the Application layer, Game Engine, Create Level, FireBase_API, Twitter_API, Facebook_API
Description of Function Calls:

No.	Name	Parameters	Functionality	Usage Scenarios	Design Rationales
1	saveScore()	String userID, String levelName, int newScore	User's PvP score after playing will be saved	After user played PvP Mode and the played scores need to be saved	To ensure users' scores are recorded and can be tracked
2	savePVPConfigurationForS	int mcqDifficulty, int OEDifficulty,	Create new PvP level	When user wants to create new PvP	A free-for-all level creation that

	tudent()	int TFDifficulty, int worldSelected, int noOfMCQ, int noOfOE, int noOfTF, String createdLevelName		level to challenge other players	promotes friendly competition
3	giveAccessToTutorialGroup()	String levelName, String tutorialGroupNum	Update tutorial group after world selection	When user wants to create new PvP level to challenge other players	A free-for-all level creation that promotes friendly competition
4	checkTeacherAssignment()	String tutorialGroupUser	Student can receive assignment created by teacher	When user needs to do assignments created by teacher	An extra homework that targets specific group of students, designed to focus on a specific portion

Adventure Mode



This component shows all the available options when a Student selects the PvE Mode, from selection of a level, playing of a level to playing the world quiz.
Sub components include:

1. Adv Mgr
2. Game Engine
3. Load Qns
4. Check Answers
5. World quiz
6. Past Feedback
7. Choose Map
8. Data Mgr

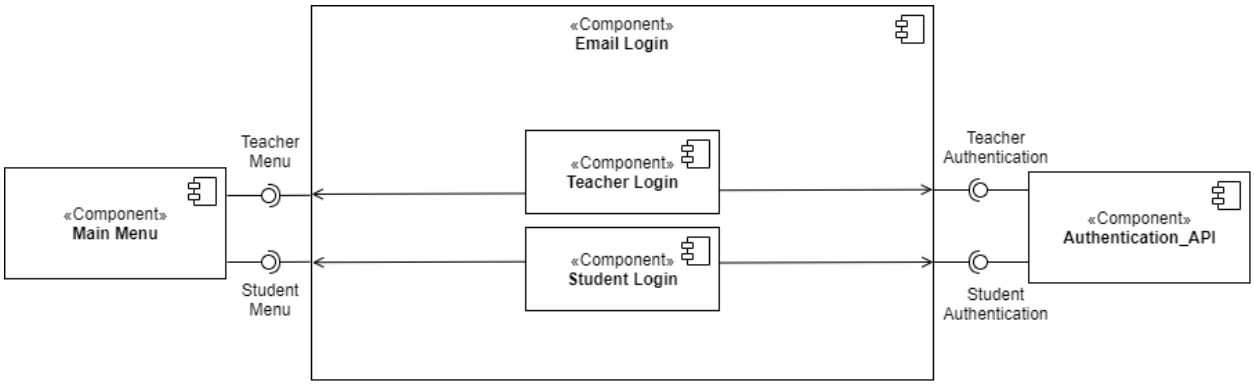
Component it interacts with: Student in the Application layer, Worlds, Stages, Level, Game Engine, FireBase_API
Description of Function Calls:

No.	Name	Parameters	Functionality	Usage Scenarios	Design Rationales
1	GameEngine	-	The actual playing of the levels for both PvP and PvE, where the user walks around a map and meets NPCs to answer questions.	When user plays a level in PvP or PvE	User can still enjoy the game element even in an educational application, hence more fun and attractive to use the

			Every level renders a new map.		application
2	getWorldName(), getWorldFName()	Firebase Collection ‘worlds’, currently logged in user	User can view the available worlds in the game and worlds will be unlocked or locked based on user progress	When a user selected PvE Mode and wants to play on a certain world	By showing all worlds in the game with a availability (locking) function, users will be forced to complete the prerequisite to continue progression, controlling users’ progress and learning
3	getWorldsScores(), getWorldsScoresInt()	List<String> wNames	User can view the total scores of a certain world by adding up all the stages scores under that world	When a user selected PvE Mode and wants to play on a certain world	By showing all worlds in the game with a availability (locking) function, users will be forced to complete the prerequisite to continue progression, controlling users’ progress and learning
4	getStageName(), getStageNumber()	Firebase Subcollection ‘Stage’ under ‘worlds’, currently logged in user	User can view the available stages in the game and stages will be unlocked or locked based on user progress	When a user selected PvE Mode and wants to play on a certain stage under a certain world	By showing all worlds in the game with a availability (locking) function, users will be forced to complete the prerequisite to continue progression, controlling users’ progress and learning
5	getStageScores(), getStagesScoresInt()	String wName, List<String> sNumbers	User can view the total scores of a certain stage by adding up all the level scores under that stage	When a user selected PvE Mode and wants to play on a certain stage under a certain world	By showing all worlds in the game with a availability (locking) function, users will be forced to complete the prerequisite to continue

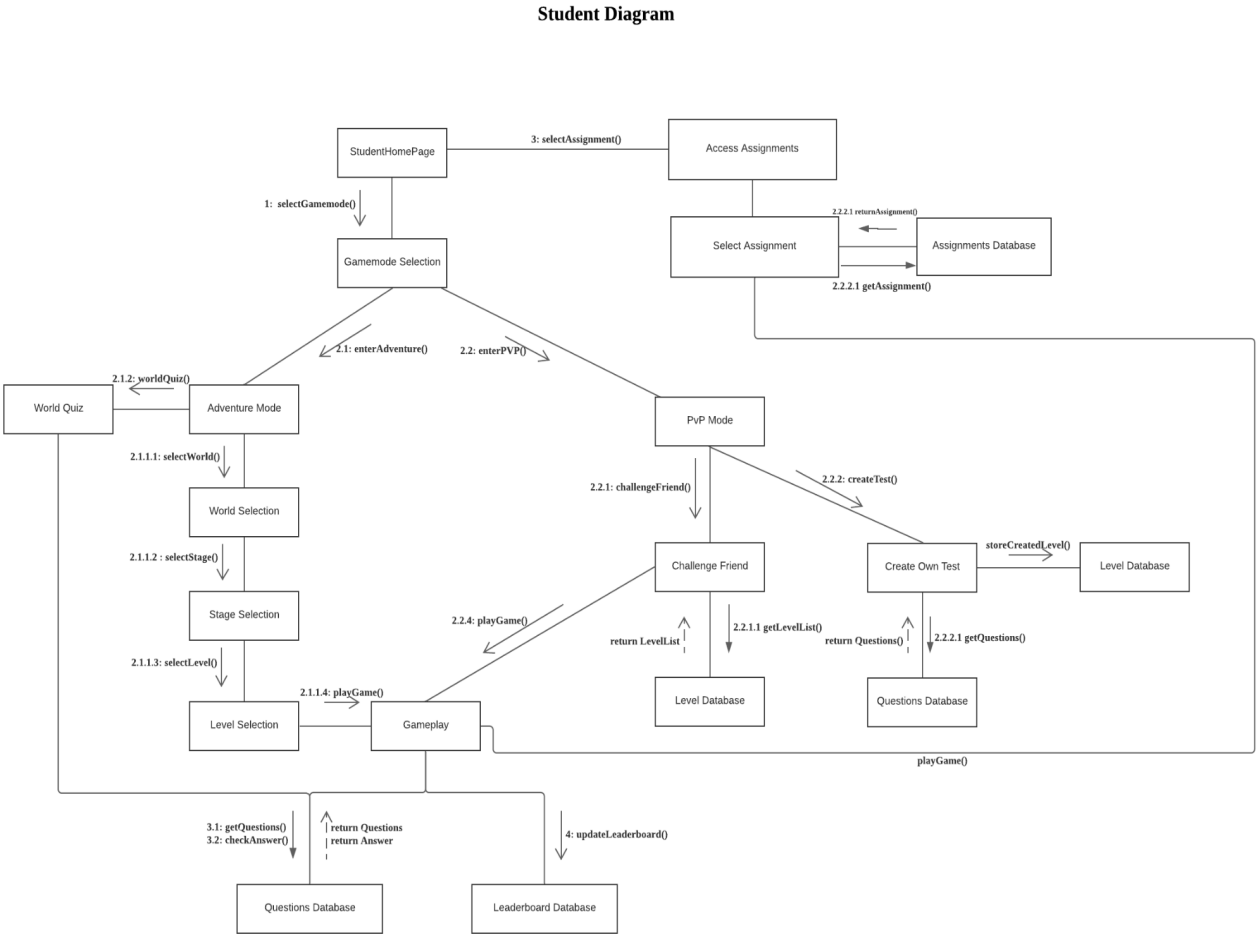
					progression, controlling users' progress and learning
6	getLevelNumber()	String worldName, String stageNumber	User can view the available levels in the game and levels will be unlocked or locked based on user progress	When a user selected PvE Mode and wants to play on a certain level under a certain stage, which is in turn under a certain world	By showing all worlds in the game with a availability (locking) function, users will be forced to complete the prerequisite to continue progression, controlling users' progress and learning
7	saveStudentAttempt()	HashMap<String, bool> studentResult, String world, int stage, int level	User's results after playing will be saved	After user played PvE Mode and the played results need to be saved	To ensure users' scores are recorded and can be tracked
8	saveAdventureScore()	String userID, int newHighScore, int oldHighScore	User's adventure score after playing will be saved	After user played PvE Mode and the played scores need to be saved	To ensure users' scores are recorded and can be tracked
9	saveAdventureScoreinStages()	String userID, Int newHighScore, int oldHighScore(), String world, int stage	User's adventure score after playing will be saved in stages	After user played PvE Mode and the played scores need to be saved	To ensure users' scores are recorded and can be tracked
10	checkMCQ(), checkTrueFalse(), checkFillInTheBlanks()	int userAns, int qnid	User's answers for MCQ questions will be checked to identify its correctness	After user played PvP/PvE Mode and the scores have to be checked against	To ensure users' scores are checked to view if it is correct

				its correctness	
11	getTotalAccessForLevels()	String worldName, String stageNumber	Retrieve the total number of level access	For user to know the PvE game progress	Control user's learning progress
12	checkAccessForWorlds()	String worldName	Retrieve the total number of world access	For user to know the PvE game progress	Control user's learning progress
13	getTotalAccessForStages()	String worldName	Retrieve the total number of stage access	For user to know the PvE game progress	Control user's learning progress
14	getCharacterSelection()	-	Return the character selected	When user wants to experience different playstyle	To increase the gaming elements in the application and attract users to have more interest in the game
15	WorldQuizController()	-	Return world quiz questions and the corresponding results	When user wants to progress to the next world	Control user's learning pace

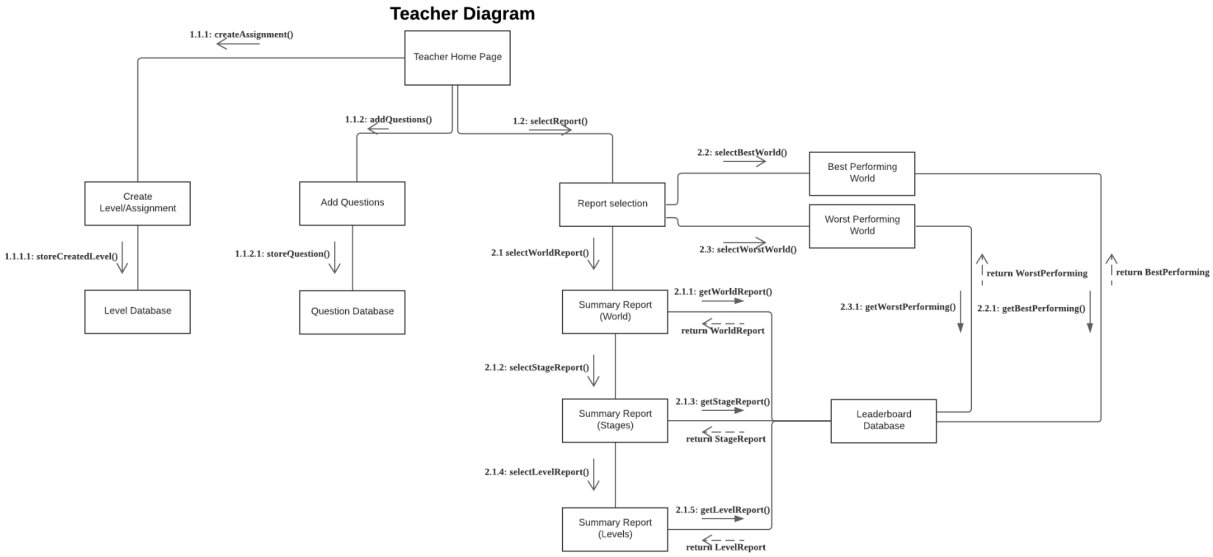


8. Communication Diagram

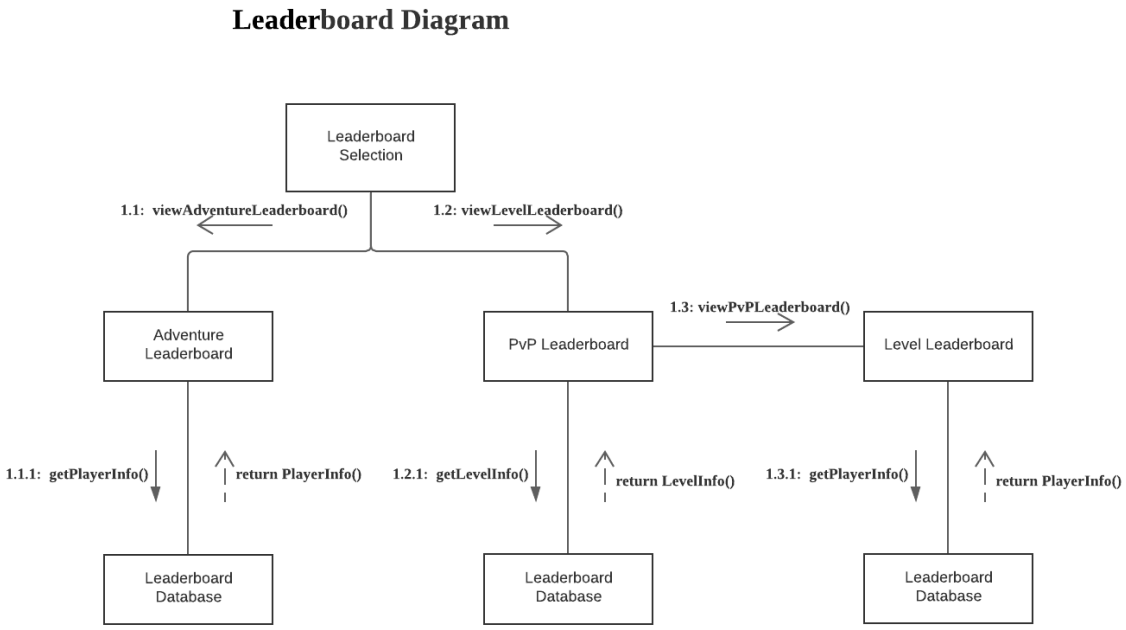
8.1 Student Diagram



8.2 Teacher Diagram



8.3 Leaderboard Diagram



9 Testing

9.1 Test Objectives

The test objectives are to verify the functionality and performance of the application in the real environment.

9.2 Test performed

9.2.1 Unit Testing

Test Method	White-box testing.
Test Level	The first level of software testing.
Test Coverage	It is carried out on every subsystem.
Test Purpose	The purpose is to ensure that every subsystem works as intended.

9.2.2 Integration Testing

Test Method	White-box testing.
Test Level	The second level of software testing.
Test Coverage	The overall testing of functions affected by the integration of subsystems.
Test Purpose	The purpose is to ensure that the complete system works when integrating the subsystems together.

9.2.1 System Testing

Test Method	Black-box testing.
Test Level	The third level of software testing.
Test Coverage	It is carried out on every feature of the entire system based on all possible usage scenarios.
Test Purpose	The purpose is to verify that the system meets the specified requirements.

9.3 Test Cases

Functional Testing

Blackbox testing was used to test the entire system based on the usage scenarios after the integration of subsystems. We have divided the usage scenarios based on the Student's usage and Teacher's usage. The purpose of the test is to evaluate the system's compliance with the specified requirements.

9.4 Student

UI Module	**Scenarios**	**Action**	**Expected output**	**Actual output**	**Status (Pass/ Fail)**
HomePage	Select between game modes	Select 'Play'	Display the ChoicesPage	Display the ChoicesPage	Pass
HomePage	Display the given assignment	Select 'Teacher's Assignment'	Display the AssignmentPage which shows all assignments created by the teacher.	Display the AssignmentPage which shows all assignments created by the teacher.	Pass
HomePage	Display the adventure leaderboard	Select 'Leaderboards'	Show the adventure leaderboard with player name and scores of the 50 players with the highest score. Show the user's points and ranking.	Show the adventure leaderboard with player name and scores of the 50 players with the highest score. Show the user's points and ranking.	Pass
HomePage	Enter settings page	Select 'Settings'	Display the SettingsPage	Display the SettingsPage	Pass
ChoicesPage	Enter Adventure gamemode	Select 'Adventure'	Display the CharacterSelectionPage with the user's name displayed	Display the CharacterSelectionPage	Pass
ChoicesPage	Enter PvP gamemode	Select 'PvP'	Display the PvPPage	Display the PvPPage	Pass
CharacterSelectionPage	Select characters	Use the left and right arrows to choose between characters	Display different characters (mage, knight) when arrows are pressed.	Display different characters (mage, knight) when arrows are pressed.	Pass

CharacterSelectionPage	Confirm character selection	Select 'Confirm'	Display WorldPage with user's: Locked worlds with a lock icon. Unlocked worlds with user's score of the world.	Display WorldPage with user's: Locked worlds with a lock icon. Unlocked worlds with user's score of the world.	Pass
WorldPage	Enter unlocked world	Select any unlocked worlds	Display the StagePage with: The world it belongs to. Every stage of the world with the user's score for each stage.	Display the StagePage with: The world it belongs to. Every stage of the world with the user's score for each stage.	Pass
WorldPage	Enter locked world	Select any locked worlds	Display a dialog with 'World is locked, please complete the world quiz prior to unlock!"	Display a dialog with 'World is locked, please complete the world quiz prior to unlock!"	Pass
WorldPage	Select world quiz	Select 'World Quiz'	Display a dialog to verify if the user wants to attempt the world quiz of the user's most recently unlocked world.	Display a dialog to verify if the user wants to attempt the world quiz of the user's most recently unlocked world.	Pass
WorldPage	Attempt world quiz	Select 'Confirm'	Attempt the world quiz with 20 questions set in MCQ format.	Attempt the world quiz with 20 questions set in MCQ format.	Pass
StagePage	Enter stage	Select any stages	Display the LevelPage with: The stage it belongs to. The user's unlocked levels with the user's best score for each level upon 10.	Display the LevelPage with: The stage it belongs to. The user's unlocked levels with the user's best score for each level upon 10.	Pass
LevelPage	Challenge level	Select any levels	Enter the game where the user will play with their selected	Enter the game where the user will play with their selected	Pass

			character.	character.	
LevelPage	View all past attempts of a level	Select the scores of the level	Display the PastAttemptPage with all past attempts of the level by the user and its score.	Display the PastAttemptPage with all past attempts of the level by the user and its score.	Pass
PastAttemptsPage	View review of a past attempt	Select 'View More'	Display a review of the selected past attempt with: The questions the user has answered. Feedback of the question if it was answered wrongly.	Display a review of the selected past attempt with: The questions the user has answered. Feedback of the question if it was answered wrongly.	Pass
PvPPage	Challenge a create level	Select 'Challenge'	Display a list of all created levels.	Display a list of all created levels.	Pass
PvPPage	Create a level	Select 'Create PvP levels'	Display LevelCreatorPage where the user can create a level by specifying: The world the questions will come from. The number of questions for each type of question. The difficulty of each type of question.	Display LevelCreatorPage where the user can create a level by specifying: The world the questions will come from. The number of questions for each type of question. The difficulty of each type of question.	Pass
LevelCreatorPage	Share created level to social media	Select 'Create Custom Level'	A slide up menu appears allowing the user to share the created level to Facebook, Telegram and Whatsapp.	A slide up menu appears allowing the user to share the created level to Facebook, Telegram and Whatsapp.	Pass
Game	Attack an NPC	Attack an NPC	The HP bar of the NPC will be reduced.	The HP bar of the NPC will be reduced.	Pass
Game	Kill an NPC	Kill an NPC	A question will pop up. The question will	A question will pop up. The question will	Pass

			be dependent on the world, stage and level selected by the user.	be dependent on the world, stage and level selected by the user.	
Game	Lose health points	NPC attacks user	The HP bar of the user will be reduced.	The HP bar of the user will be reduced.	Pass
Game	Lose all health points	NPC kills user	Display a dialog “You have ran out of HP, Game Over!” and return the user back to the LevelPage. The attempt will be recorded in past attempts. Scores of the level and leaderboard will be updated if a high score is obtained.	Display a dialog “You have ran out of HP, Game Over!” and return the user back to the LevelPage. The attempt will be recorded in past attempts. Scores of the level and leaderboard will be updated if a high score is obtained.	Pass
AdventureLeaderBoard Page	Show PvP leaderboard	Select ‘PvP’	Display the PvP leaderboard with the level name, its rating and a button to challenge the level. The levels are sorted based on ratings and only the 50 highest rated levels will be shown.	Display the PvP leaderboard with the level name, its rating and a button to challenge the level. The levels are sorted based on ratings and only the 50 highest rated levels will be shown.	Pass
PvPLeader BoardPage	Show Adventure leaderboard	Select ‘Adventure’	Show the adventure leaderboard with player name and scores of the 50 players with the highest score. Show the user’s points and ranking.	Show the adventure leaderboard with player name and scores of the 50 players with the highest score. Show the user’s points and ranking.	Pass
PvPLeader BoardPage	Show Level leaderboard	Select the name of a level	Show the individual level leaderboard with player name and scores of the 50 players with the highest score for that	Show the individual level leaderboard with player name and scores of the 50 players with the highest score for that	Pass

			level. Show the user's points and ranking for that level.	level. Show the user's points and ranking for that level.	

9.5 Teacher

UI Module	Scenarios	Action	Expected output	Actual output	Status (Pass/ Fail)
HomePage	Adding new questions	Select “Add Question”	Display the world, stage and level for teacher’s to choose where to add the new question to.	Display the world, stage and level for teacher’s to choose where to add the new question to.	Pass
HomePage	Give an assignment	Select “Give Assignment”	Display AssignmentPage where the teacher can create an assignment by specifying: The tutorial group the assignment will be assigned to. The world, stage and level the questions will come from. The number of questions for each type of question.	Display AssignmentPage where the teacher can create an assignment by specifying: The tutorial group the assignment will be assigned to. The world, stage and level the questions will come from. The number of questions for each type of question.	Pass
HomePage	View summary report	Select “Summary Report”	Displays 3 types of summary report options for the teacher to select: Adventure Summary Report. (SummaryReportPage) Best Performing Report. (BestPerformingPage) Areas to Improve Report (WorstPerformingPage)	Displays 3 types of summary report options for the teacher to select: Adventure Summary Report. (SummaryReportPage) Best Performing Report. (BestPerformingPage) Areas to Improve Report (WorstPerformingPage)	Pass
QuestionCreatorPage	Create a new question with type MCQ	Select ‘MCQ’	Display CreateMCQPage with the following:	Display CreateMCQPage with the following:	Pass

			A question box where teachers have to fill in the question. Four answer boxes where teachers have to fill in four possible answers.	A question box where teachers have to fill in the question. Four answer boxes where teachers have to fill in four possible answers.	
CreateMCQ Page	Fill in the questions and possible answers	Select the question box and fill in the question. Select the answers box and fill in three wrong answers and one right answer. Select the check icon to confirm the correct answer. Select 'Confirm'	Display a dialog 'Question has been added' and return to QuestionCreatorPage.	Display a dialog 'Question has been added' and return to QuestionCreatorPage.	Pass
CreateMCQ Page	Question or answer boxes not filled in	Select 'Confirm'	Display a dialog 'Some of the fields are left empty, please try again'.	Display a dialog 'Some of the fields are left empty, please try again'.	Pass
QuestionCr eatorPage	Create a new question with type Fill in the Blanks.	Select 'Fill in the Blanks'	Display CreateFITBPage with the following: A question box where teachers have to fill in the question. An answer box where teachers have to fill in the answer.	Display CreateFITBPage with the following: A question box where teachers have to fill in the question. An answer box where teachers have to fill in the answer.	Pass
CreateFITB	Fill in the	Select the	Display a dialog	Display a dialog	Pass

Page	questions and answer	question box and fill in the question. Select the answers box and fill in the answer Select 'Confirm'	'Question has been added' and return to QuestionCreatorPage.	'Question has been added' and return to QuestionCreatorPage.	
CreateFITB Page	Question or answer boxes not filled in	Select 'Confirm'	Display a dialog 'Some of the fields are left empty, please try again'.	Display a dialog 'Some of the fields are left empty, please try again'.	Pass
QuestionCreatorPage	Create a new question with type True/False	Select 'True/False'	Display CreateTFPage with the following: A question box where teachers have to fill in the question. Two answer boxes where teachers have to fill in the correct answer and a wrong answer.	Display CreateTFPage with the following: A question box where teachers have to fill in the question. Two answer boxes where teachers have to fill in the correct answer and a wrong answer.	Pass
CreateTFPage	Fill in the questions and answers	Select the question box and fill in the question. Select the answers box and fill a correct answer and a wrong answer. Select the check icon to confirm the correct answer. Select	Display a dialog 'Question has been added' and return to QuestionCreatorPage.	Display a dialog 'Question has been added' and return to QuestionCreatorPage.	

		‘Confirm’			
CreateTFPage	Question or answer boxes not filled in	Select ‘Confirm’	Display a dialog ‘Some of the fields are left empty, please try again’.	Display a dialog ‘Some of the fields are left empty, please try again’.	Pass
Assignment Page	Share created level to social media	Select ‘Create Assignment’	A slide up menu appears allowing the teacher to share the created assignment to Twitter, Facebook, Telegram and Whatsapp.	A slide up menu appears allowing the teacher to share the created assignment to Twitter, Facebook, Telegram and Whatsapp.	Pass
SummaryReportPage (World)	The teacher wants to see the average scores of students in each world	Select ‘Average Scores’ in the dropdown menu	Display average score of students in each world.	Display average score of students in each world.	Pass
SummaryReportPage (World)	The teacher wants to see the number of students that unlocked the world	Select ‘No. of Unlocked’ in the dropdown menu	Display the number of students that have unlocked each world.	Display the number of students that have unlocked each world.	Pass
SummaryReportPage (World)	The teacher wants to see the adventure summary report of stages.	Select ‘View Stages’	Display SummaryReportPage (Stage)	Display SummaryReportPage (Stage)	Pass
SummaryReportPage (World)	The teacher wants to see the average scores of students in each stage.	Select ‘View More’ in the dropdown menu	Display average score of students in each stage.	Display average score of students in each stage.	Pass
SummaryReportPage (Stage)	The teacher wants to see the adventure summary	Select ‘View Levels’	Display SummaryReportPage (Level)	Display SummaryReportPage (Level)	Pass

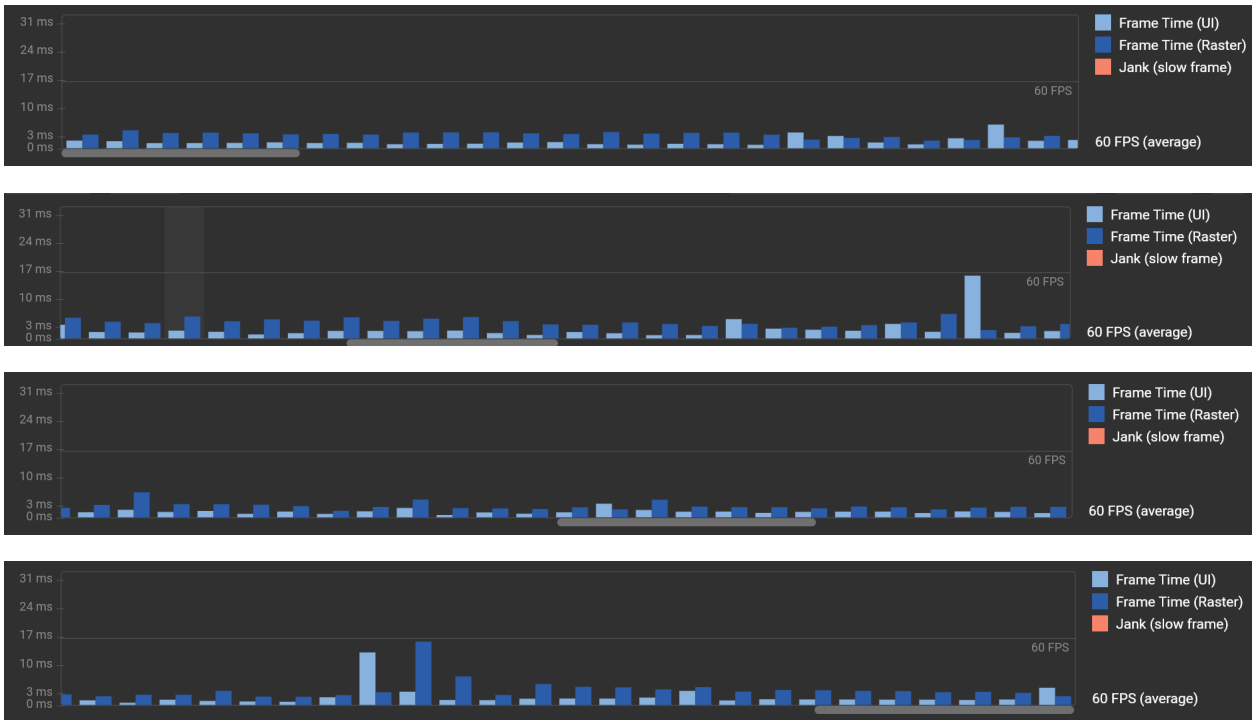
	report of levels.				
SummaryReportPage (Level)	The teacher wants to see the average scores of students in each level.	Select 'Average Scores' in the dropdown menu	Display average score of students in each level.	Display average score of students in each level.	Pass
SummaryReportPage (Level)	The teacher wants to see the number of students that unlocked the level	Select 'No. of Unlocked' in the dropdown menu	Display the number of students that have unlocked each level.	Display the number of students that have unlocked each level.	Pass
SummaryReportPage (Level)	The teacher wants to see the average number of re-attempts by students in each level	Select 'No. of Re-attempts' in the dropdown menu	Display the average number of re-attempts by students in each level.	Display the average number of re-attempts by students in each level.	Pass
BestPerformingPage (World)	The teacher wants to see the top three best performing worlds	-	Display the highest scoring three worlds based on the average points of the students in that world.	Display the highest scoring three worlds based on the average points of the students in that world.	Pass
BestPerformingPage (World)	The teacher wants to see the top three best stages in the world.	Select 'More Details'	Display the top three best performing stages of the selected world.	Display the top three best performing stages of the selected world.	Pass
WorstPerformingPage (World)	The teacher wants to see the top three worst performing worlds	-	Display the lowest scoring three worlds based on the average points of the students in that world.	Display the lowest scoring three worlds based on the average points of the students in that world.	Pass
WorstPerformingPage (World)	The teacher wants to see the three worst performing stages in the world.	Select 'More Details'	Display the three worst performing stages of the selected world.	Display the three worst performing stages of the selected world.	Pass

Non-Functional Testing

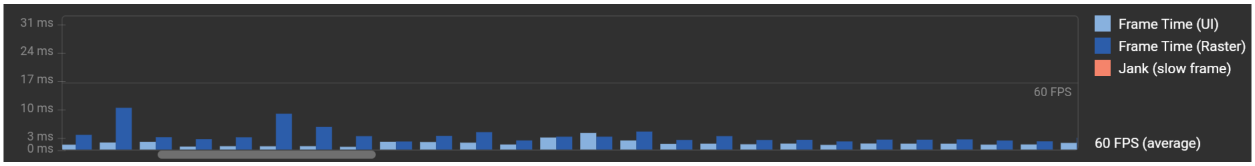
We tested the performance of the application using a built in automated function, DevTools, provided by Flutter. The charts below display timeline metrics 300 frames at a time. Any frame going over the 60 FPS horizontal line will result in a jittery user experience for the user. To maintain 60 FPS, the system must not take longer than 16ms to render.

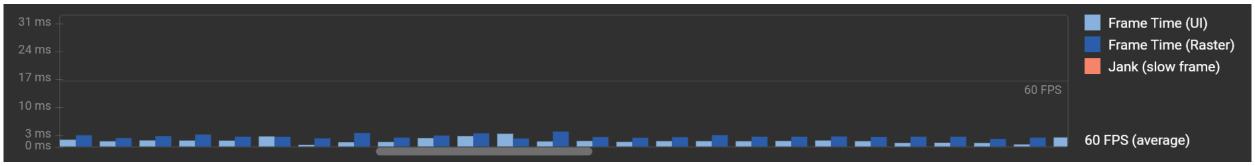
Running of the application consists of log in, attempting the world quiz, playing the game and viewing the leaderboards as we deem the following 4 to be the most performance consuming.

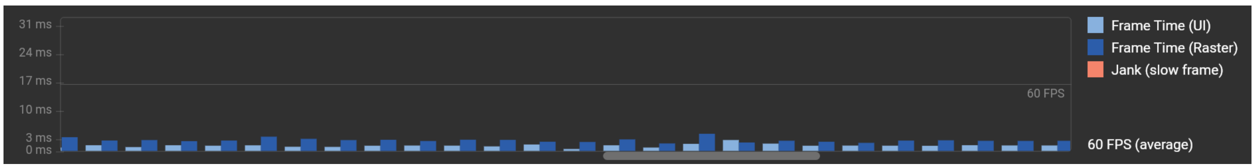
The following bar charts shows the performance of a Samsung Note 4 while running the application.

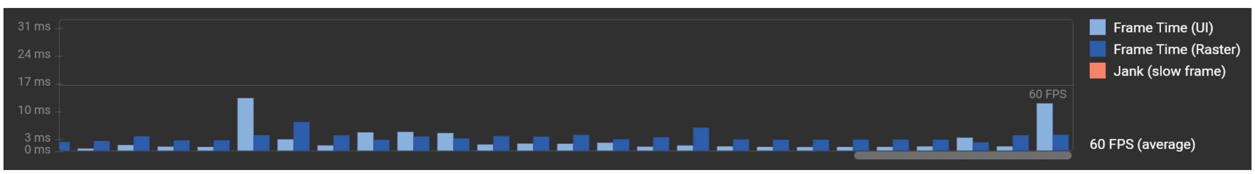


The bar chart above shows the performance of a Samsung Galaxy FE20 while running the application.









Appendix A: Data Dictionary

Terms	Definition
Topic (Worlds)	The topic (worlds) refers to the six different phases in the Software Design Life Cycle (SDLC)
Concepts (Stage)	The concepts (stage) refers to the concepts that can be found in a phase of the SDLC.
Difficulty for a concept (Level)	The levels refer to the difficulty of questions that will be asked in that level. The difficulty of a question is determined by the teachers who have set the questions.
Software Design Life Cycle (SDLC)	It is a process used to design, develop and test a software, such that the software can be completed within time and budget, and more importantly meeting client's requirements.
Non-Player Character (NPC)	A character in a game that is not being controlled by humans, rather it can be controlled by a program code that we have written.
Player versus Player (PVP)	A term used to describe a multiplayer interactive conflict within a game, where a human player has other human player(s) as the enemy.
Player versus Environment (PVE)	A term used to describe a single player having non-human players (usually AI) as the enemy, and has to challenge against various computer-controlled obstacles.

Appendix B: Analysis Models

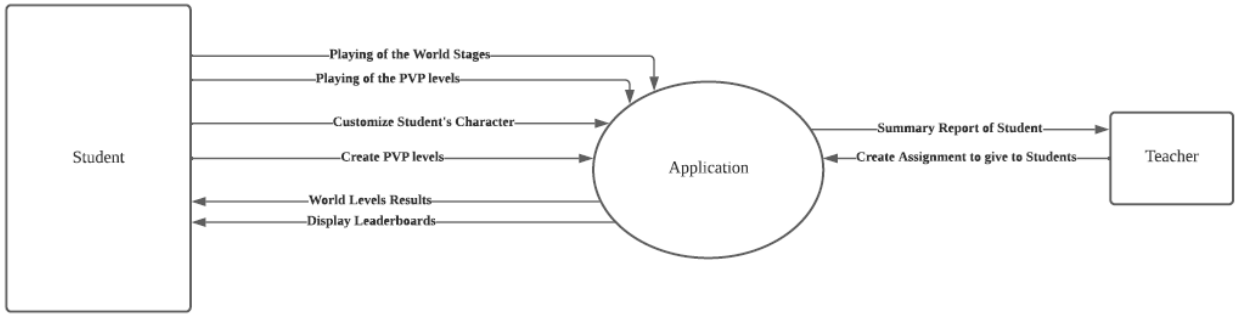


Figure 1: Conceptual Model

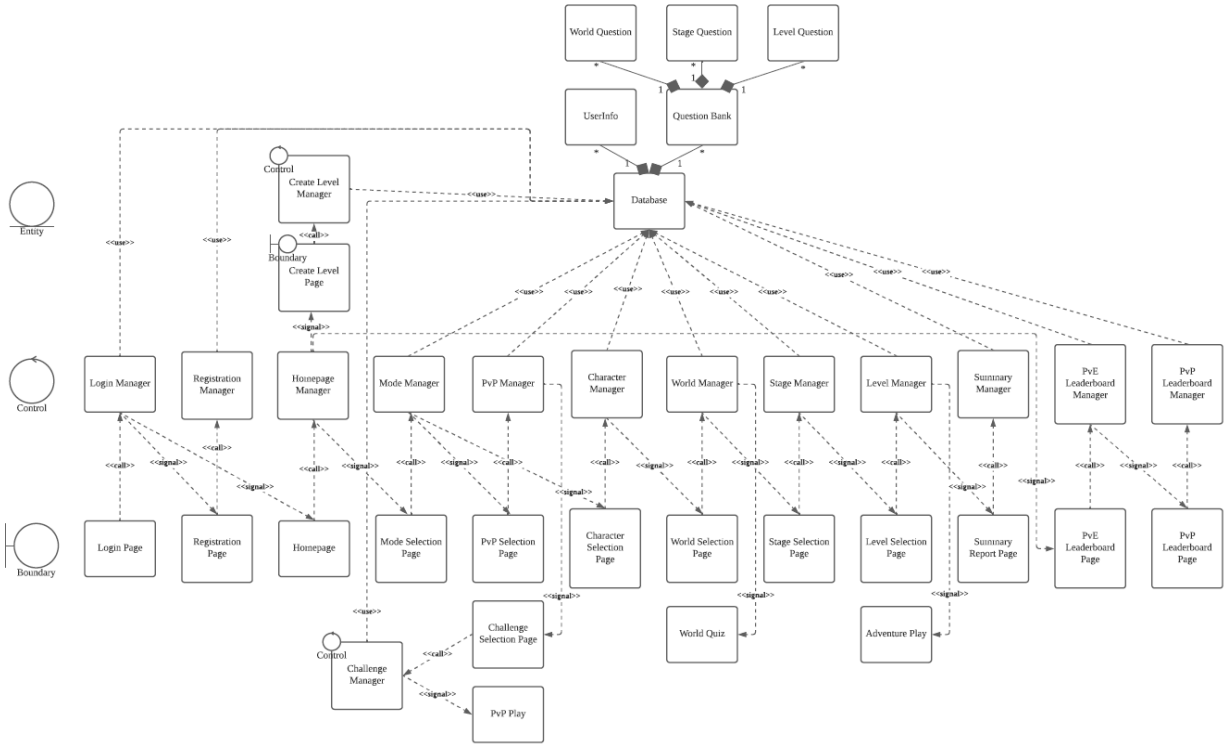


Figure 2: Simple Class Diagram