

Software Requirements Specification

for

Softvengers

Version 1.3 approved

Prepared by Software Chasers

School of Computer Science and Engineering (SCSE),

Nanyang Technological University

21 April 2021

Table of Contents

Introduction	1
Purpose	1
Document Conventions	1
Intended Audience and Reading Suggestions	1
Product Scope	2
References	2
Overall Description	2
Product Perspective	2
Product Functions	3
User Classes and Characteristics	3
Operating Environment	4
Design and Implementation Constraints	4
User Documentation	5
Assumptions and Dependencies	5
External Interface Requirements	5
User Interfaces	5
Hardware Interfaces	24
Software Interfaces	24
Communications Interfaces	25
System Features	25
Web application for Teachers	25
Students side Functional Requirements	28
Use case Diagram:	36
Use Case Descriptions:	37
Other Nonfunctional Requirements	81
Flexibility Requirements	81
System Architecture	83
Primary candidate architecture	83
Entity Game Component Architecture - Game	85
Alternate Architectures	86
Layered System Architecture	86

Subsystem Interface	88
Leaderboard	88
Add Assignment	89
Game Play	89
Navigation	90
Progress	90
Settings	91
Testing	91
Unit Testing	92
Unit Test Scripts - Game Application	104
Unit Test Scripts - Backend Application (Student API)	105
Integration Testing	106
Testing Test Scripts	112
Test Scripts	114
System Testing	115
Load Testing – Student's game	115
Load Testing – Teacher's website	120
Security test – Backend application	125
Appendix A: Glossary	126
Data Dictionary	126
Appendix B: Analysis Models	128
Dialog Map	128
Data Flow Diagram	129
Entity Relationship Diagram	130
Decision Tables	131
Crudl Matrix	135
Component Diagram	137
Communication Diagram	138

Revision History

Name	Date	Reason For Changes	Version
SoftwareChasers	14th Feb 2021	Creation of SRS	1.0
SoftwareChasers	21st March 2021	Updation of SRS	1.1
Software Chasers	4th April 2021	Updation of SRS	1.2
Software Chasers	21st April 2021	Updation of SRS	1.3

1. Introduction

1.1 Purpose

The SRS specifies software requirements of a social game to gamify and socialize teaching and learning of software engineering courses. In the version 1.0 of the product, students can learn and compete with each other via playing the game, and teachers can assess the mastery of course via data analysis. The SRS looks at the functional and non-functional requirements and explores the intricacies of the system through use-case, data-flow, CRUDL and other such diagrams.

1.2 Document Conventions

The format adapted in the SRS is as follows:

- Main Section Heading:
 - Font: Times
 - Face: Bold
 - Size: 18
- Subsection Heading:
 - Font: Times
 - Face: Bold
 - Size: 14
- Body:
 - Font: Times
 - Face: Normal
 - Size: 11

1.3 Intended Audience and Reading Suggestions

The document is intended to be used for developers, project managers and testers. The document contains necessary information for the development of the entire system.

For Project Managers and Developers, this document is intended to assist in the maintenance and upgrade of features for the system. It is recommended for developers to begin reading the SRS from Section 1. For Quality Assurance Testers, it is recommended to focus on Section 4 and Section 5, which covers the Functional and Non-Functional Requirements.

1.4 Product Scope

The software will allow the students to augment their learning in the classroom of software engineering courses by gamifying the concepts taught. Thus, students can learn and compete with each other via playing the game, and teachers can assess the mastery of the course via data analysis.

The software will benefit the students by increasing their learning interests and interactions. They can learn while being engaged for getting higher ranks in the leaderboard. They will challenge each other by designing their own sections.

The software will benefit the teachers by releasing their burden from manual teaching activities. They will easily assess the mastery of course via data analysis (realtime and continuously) besides automatically performing assignment grading. It would also allow them to logically adjust the teaching contents and key points.

The software thus aims to achieve the goal of stimulating learning interests of the students by involving them in a continuous and interactive activity after class, while the teachers keep a track of the performance of the students.

1.5 References

These technologies and APIs were referred to and utilized in our projects.

- Unity Development: <https://docs.unity3d.com/Manual/index.htm>
- Express: https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/Introduction
- React Development: <https://reactjs.org/docs/getting-started.html>
- MongoDB Development: <https://docs.mongodb.com/>
- Auth0API: <https://auth0.com/docs/api>
- Reddit API: <https://www.reddit.com/dev/api/>
- Twitter API: <https://developer.twitter.com/en/docs>

2. Overall Description

2.1 Product Perspective

'Softvengers' is a new self-contained product which is created for the Teaching, Learning and Pedagogy Division (TLPD) and aims to provide a game to gamify and socialise the learning of software engineering concepts. It serves as a platform which makes learning fun and interactive.

It consists of 2 parts - Game for the student and Website for the teachers.

2.2 Product Functions

1. The game shall have a theme, and have different characters for players to choose.
2. The game shall consist of a series of worlds to be explored that represent different phases of the life cycle of software engineering, ranging from requirements engineering and architectural design to implementation and software testing. Each world shall have several sections - representing specific topics of each phase from basic ones to advanced ones, and each section has several levels-representing questions relevant to specific topics with increasing difficulties.
3. The game shall have a leaderboard to keep the players engaged for higher ranks.
4. The game shall support game data analysis in the following two aspects-
 - a. First, the game shall analyze each player's playing history to continuously obtain the student's mastery of the course in real-time. Based on this real-time analysis results, the questions of the following levels can be customized.
 - b. Second, the game shall analyze all the players' playing history to continuously obtain the overall mastery of the course in real time, and produce a summary report for the teachers (including which parts are well-mastered by the students, and which parts are not).
 - c. Based on this report, teachers can logically adjust the teaching contents and key points during the classroom teaching.
5. The game shall allow players to design their own levels so that students can challenge each other
6. The teachers can give assignments to students through social media.

2.3 User Classes and Characteristics

1. Teachers will be able to use this product to share assignments with students, monitor their progress and upload questions for testing the students.
2. Teachers can gain useful insights from the statistics of the student's performance.
3. Students will be able to test their understanding of software engineering by answering questions while they explore the universe. In this process they will collect badges and points which will boost their confidence and further motivate them.
4. Students will be able to create and challenge peers on social media.
5. Students can also attempt assignments.
6. Students can make posts on the discussion forum.
7. Both teachers and students can view the leaderboard to compare performance among all tutorial groups.

2.4 Operating Environment

Operating platform for Softvengers game:

- Development platform - Unity 2019.4.20f1
- Operating system platform - Windows and MacOS

Web browser for teacher's website

- Google Chrome
- Safari
- Mozilla Firefox
- Microsoft Edge

Database server

- MongoDB Atlas Server

Social Media API for challenge and assignment sharing

- Reddit API
- Twitter API

2.5 Design and Implementation Constraints

Some design considerations were made to ensure the project's success and completion within the deadline. The constraints include:

1. Concurrent Users: The system might not be able to accommodate a huge number of concurrent users as it needs to communicate online to our database. This is applicable for both the game and the website.
2. Lack of Assets: The current assets for the game might be inadequate and lacking (asteroid, spaceship, music, and effects) as we have used some of the default assets provided by Unity. Additional resources to improve the overall look and design of the game can be bought to further improve the graphics.
3. PC Only Application: The game and the website are running only on Windows and Mac OS. As such, users using devices that run on other operating systems (Linux) will not be able to use the game application. The website can be made so that it supports other devices like tablets, mobile phones.
4. Language Support: The system is only available in English and not in other languages.
5. Security Design: The Softvengers Game uses a simple authentication for login and might not be the most secure method for protecting students' information. Additional security measures need to be outsourced.

2.6 User Documentation

The students can view a tutorial video on youtube for playing the game and understanding different features.

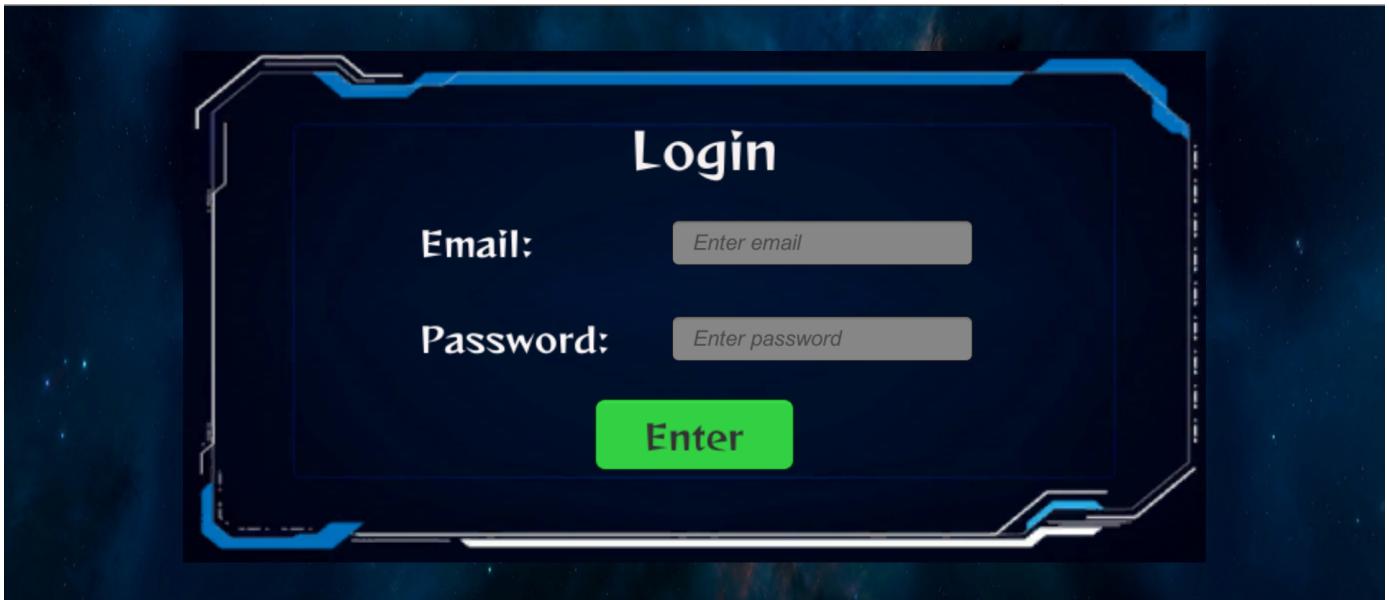
2.7 Assumptions and Dependencies

1. All students and teachers have a social media account
2. All students and teachers have a ntulearn account
3. The question bank has already been created by the teacher and divided into three levels of difficulty

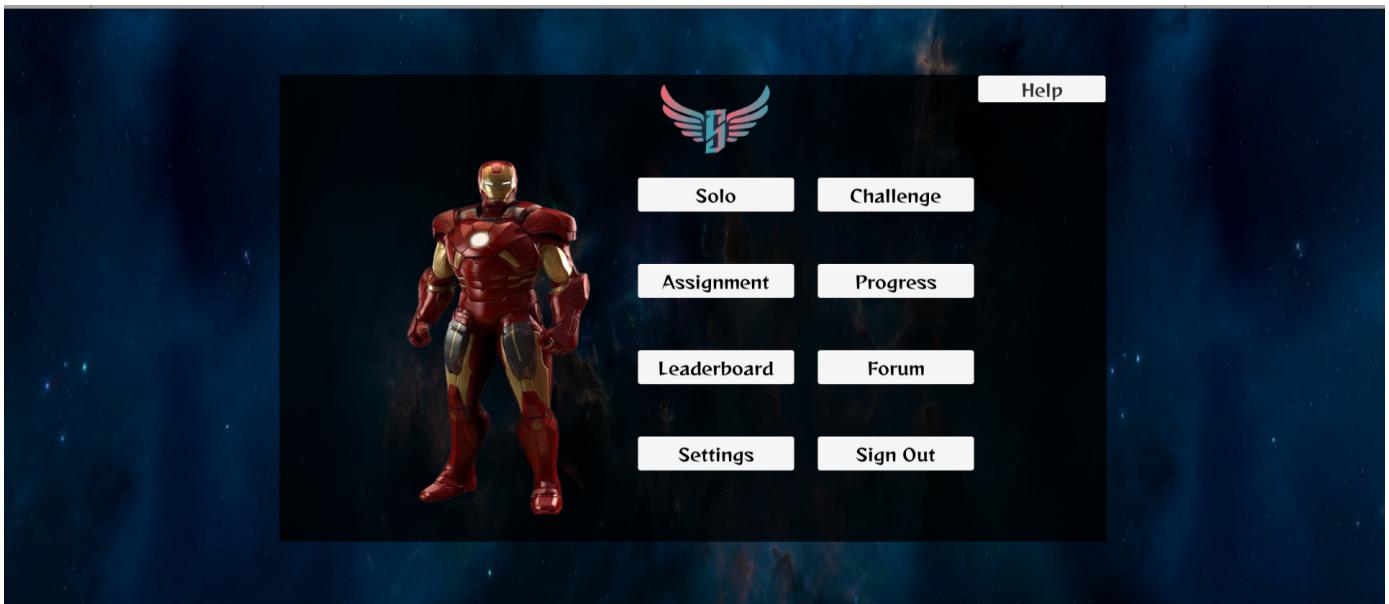
3. External Interface Requirements

3.1 User Interfaces

3.1.1 Softvengers Game Application Login Page



3.1.2 Softvengers Game Application Home Page



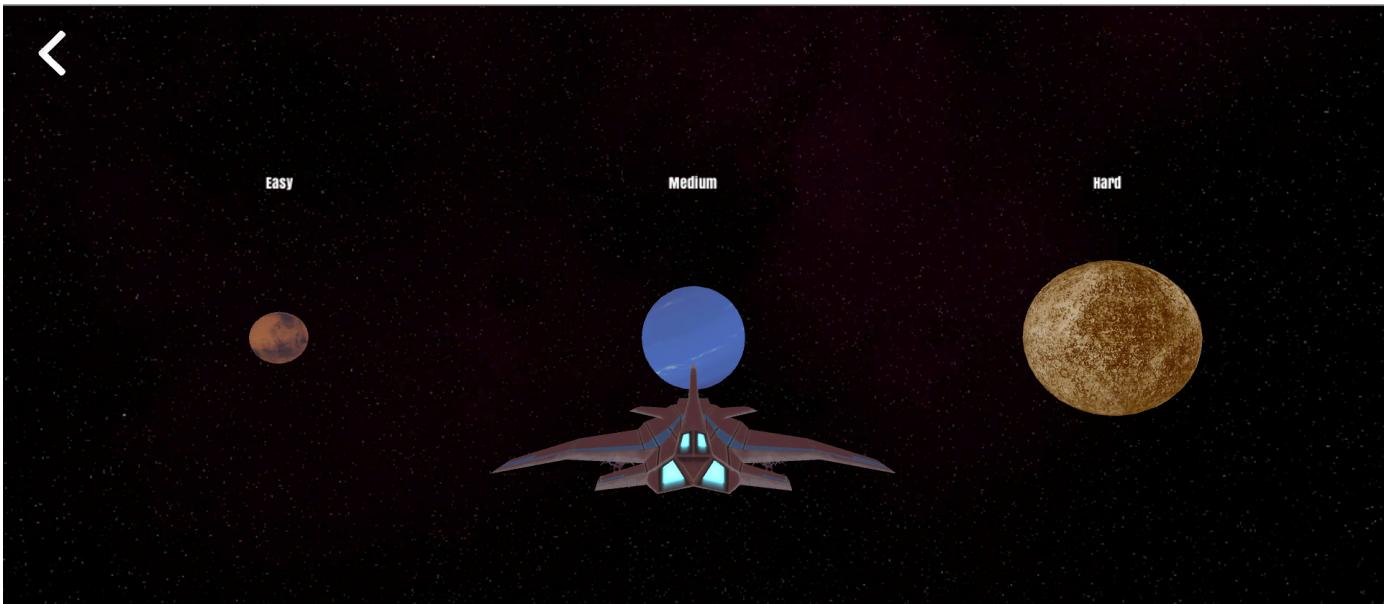
3.1.3 Softvengers Game Application Universe Selection Screen



3.1.4 Softvengers Game Application Solar System Selection Screen



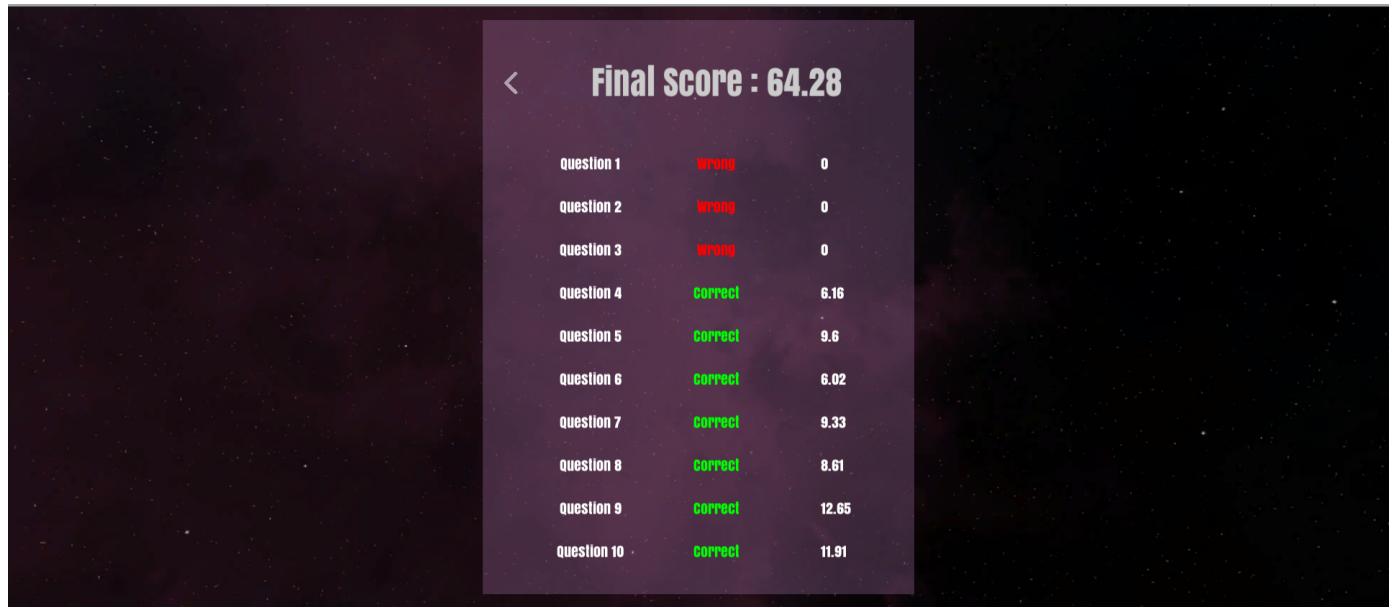
3.1.5 Softvengers Game Application Planet Selection Screen



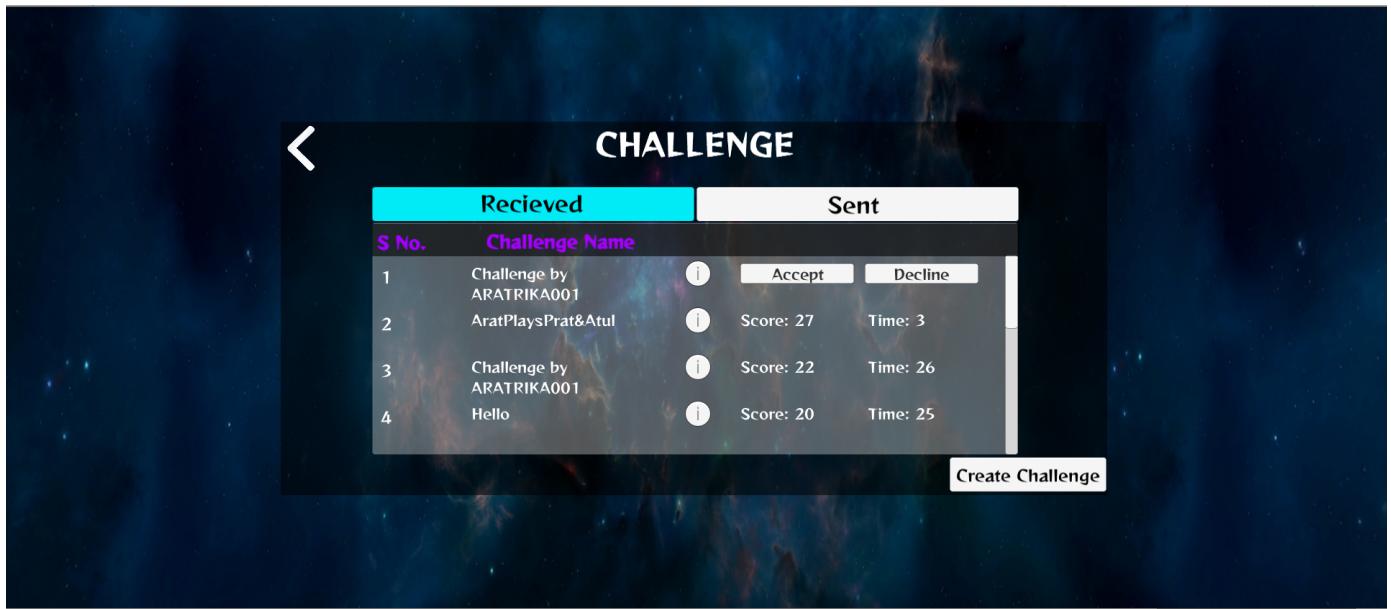
3.1.6 Softvengers Game Application Solo Mode Game Play



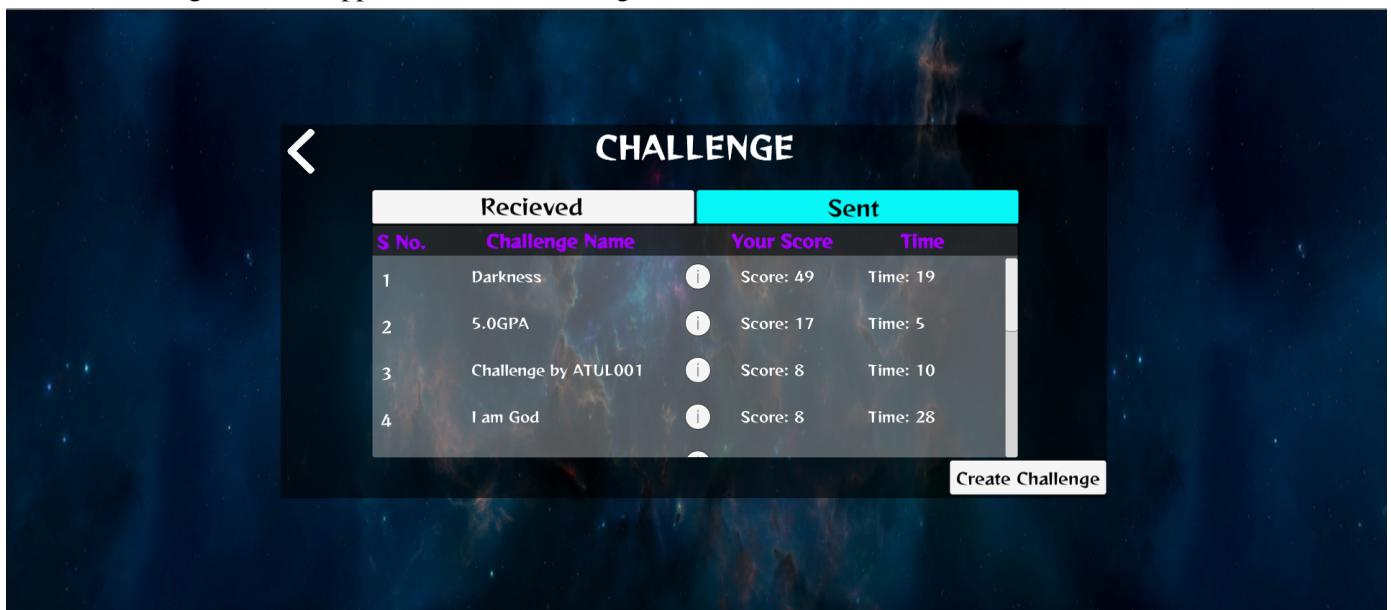
3.1.7 Softvengers Game Application Solo Mode Result Page



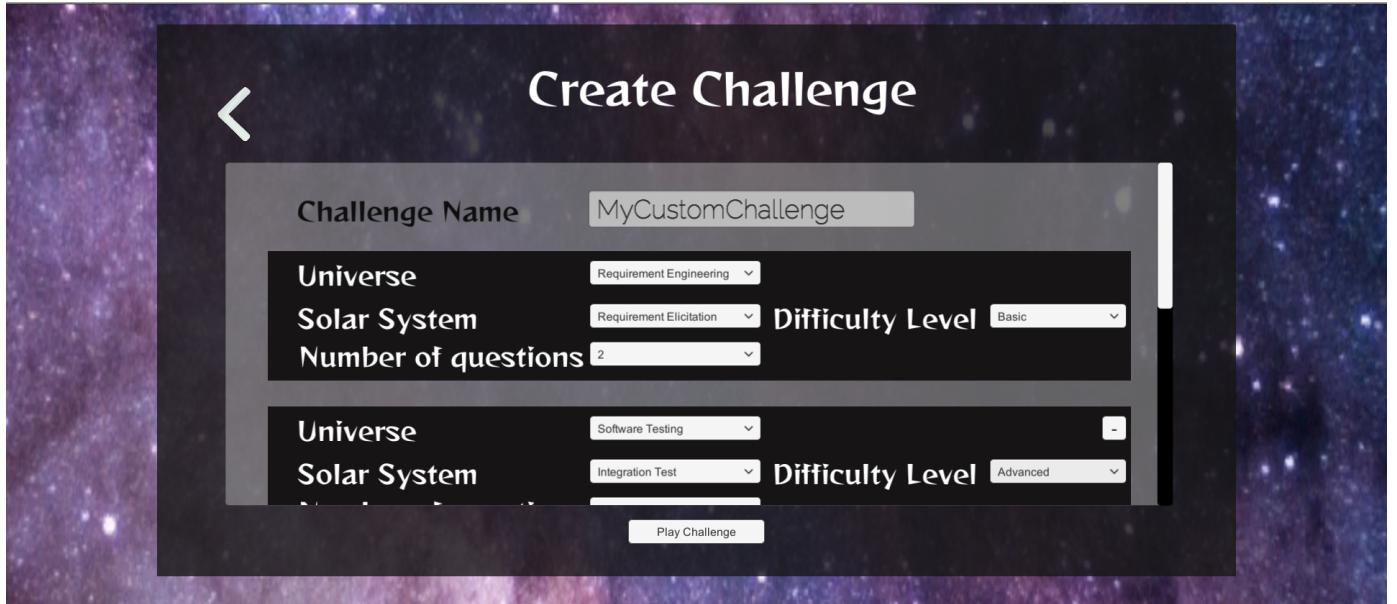
3.1.8 Softvengers Game Application Received Challenges Menu



3.1.9 Softvengers Game Application Sent Challenges Menu



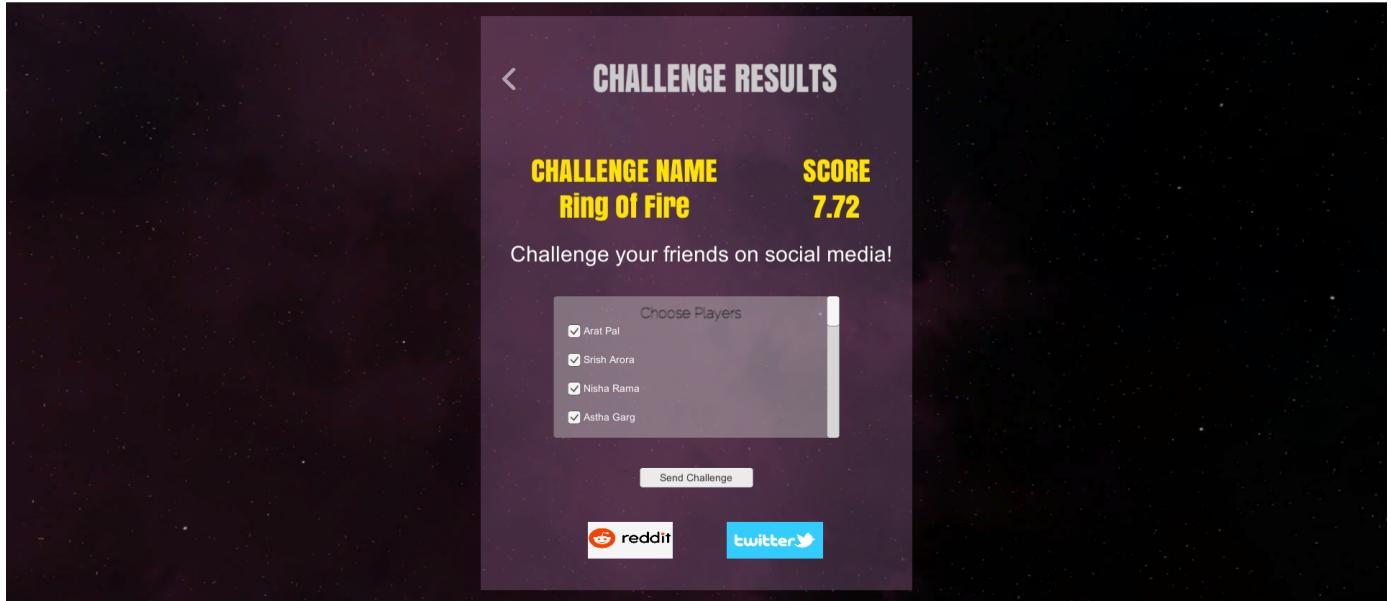
3.1.10 Softvengers Game Application Create Challenge Menu



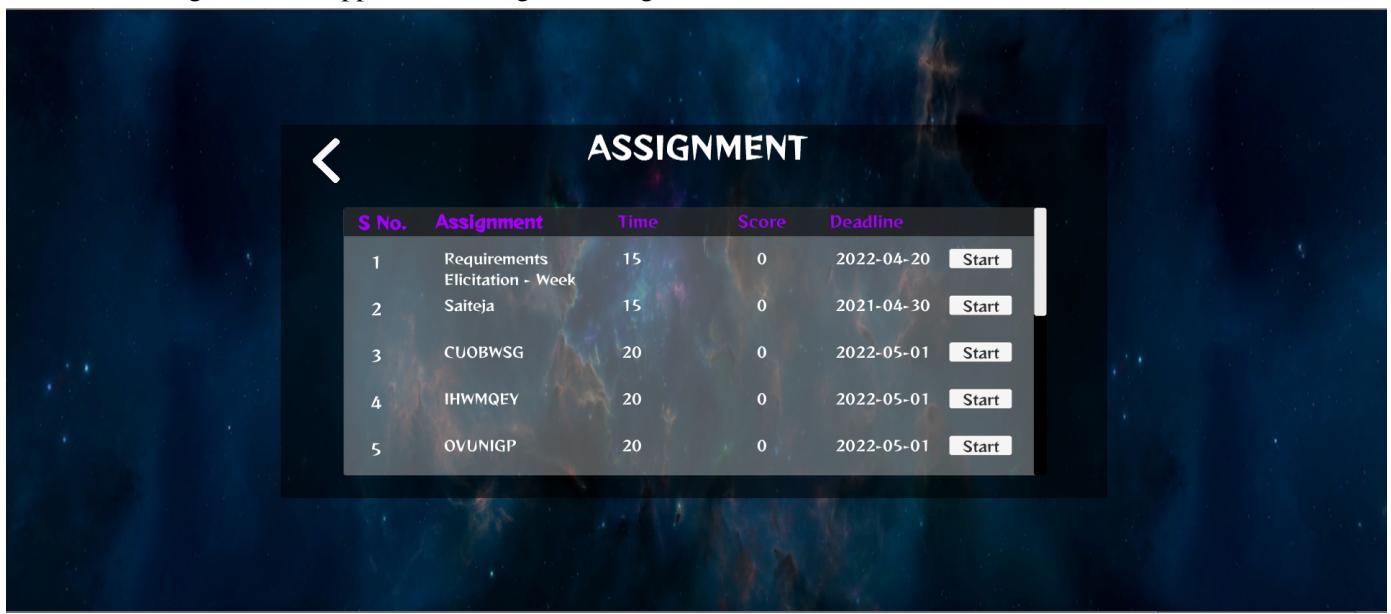
3.1.11 Softvengers Game Application Challenge Game Play



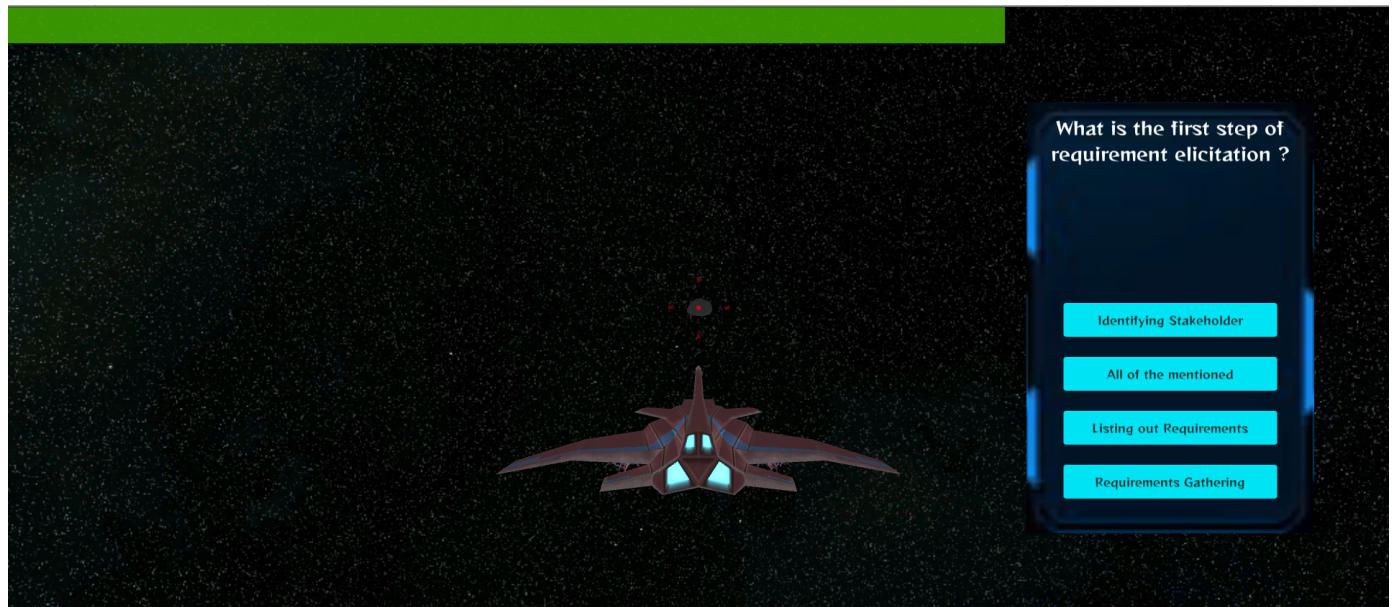
3.1.12 Softvengers Game Application Challenge Result Page



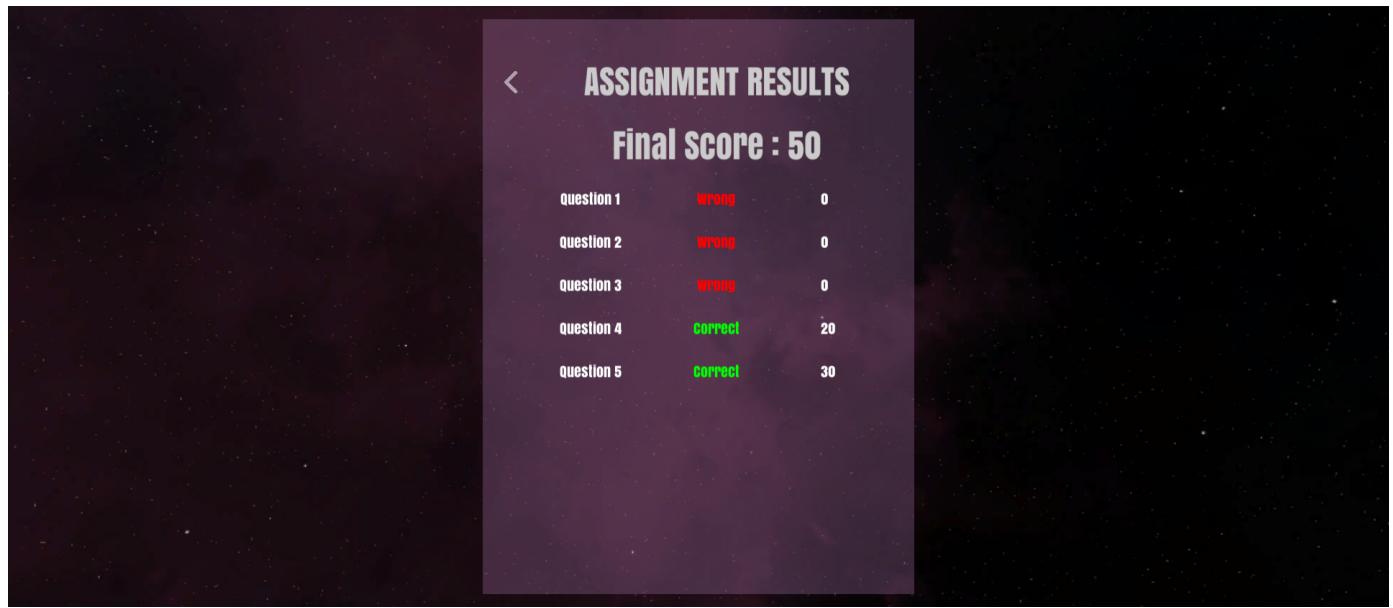
3.1.13 Softvengers Game Application Assignment Page



3.1.14 Softvengers Game Application Assignment Game Play



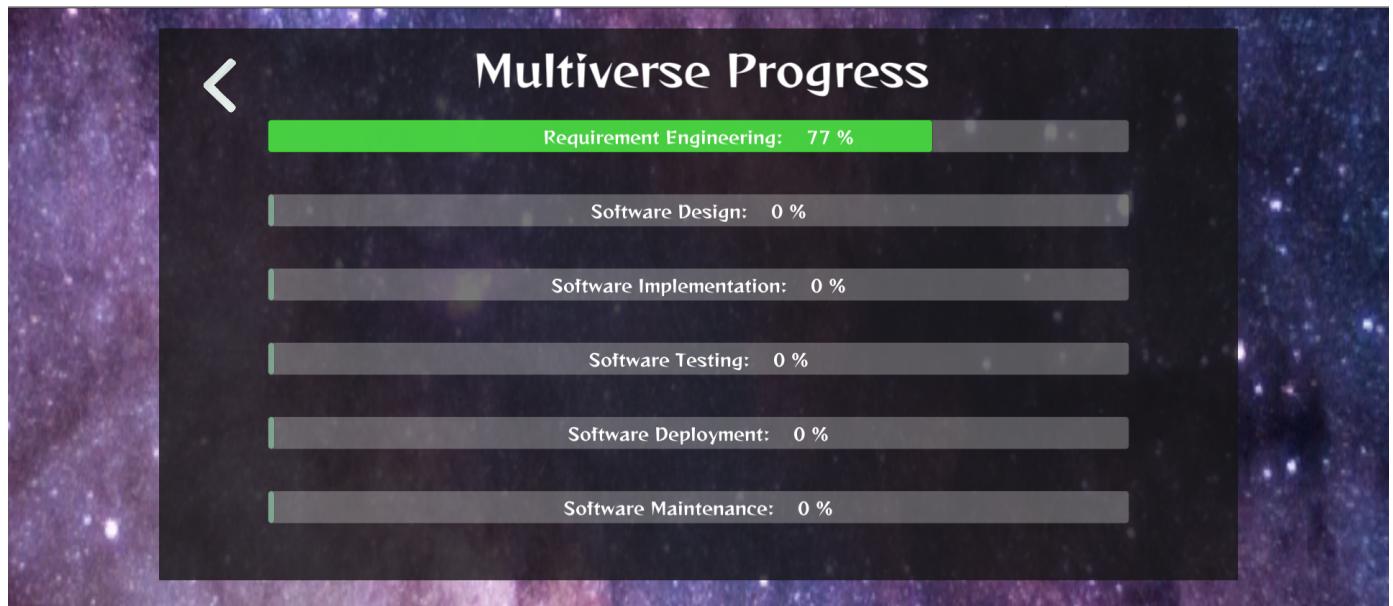
3.1.15 Softvengers Game Application Assignment Result Page



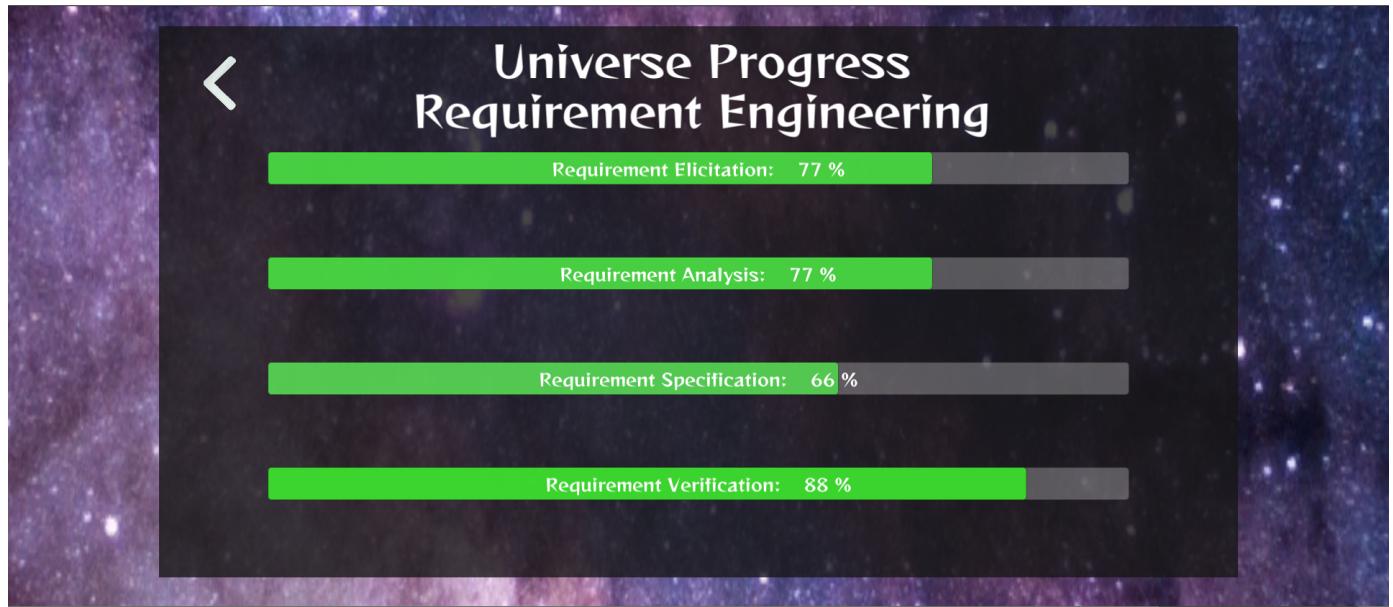
3.1.16 Softvengers Game Application Leaderboard Screen

Leaderboard		
Your current rank: 1		
RANKING	USERNAME	SCORE
1	ATUL ACHARYA	1191.866
2	ARAT PAL	291.284
3	ASTHA GARG	81.997
4	SRISH ARORA	0
5	NISHA RAMA	0

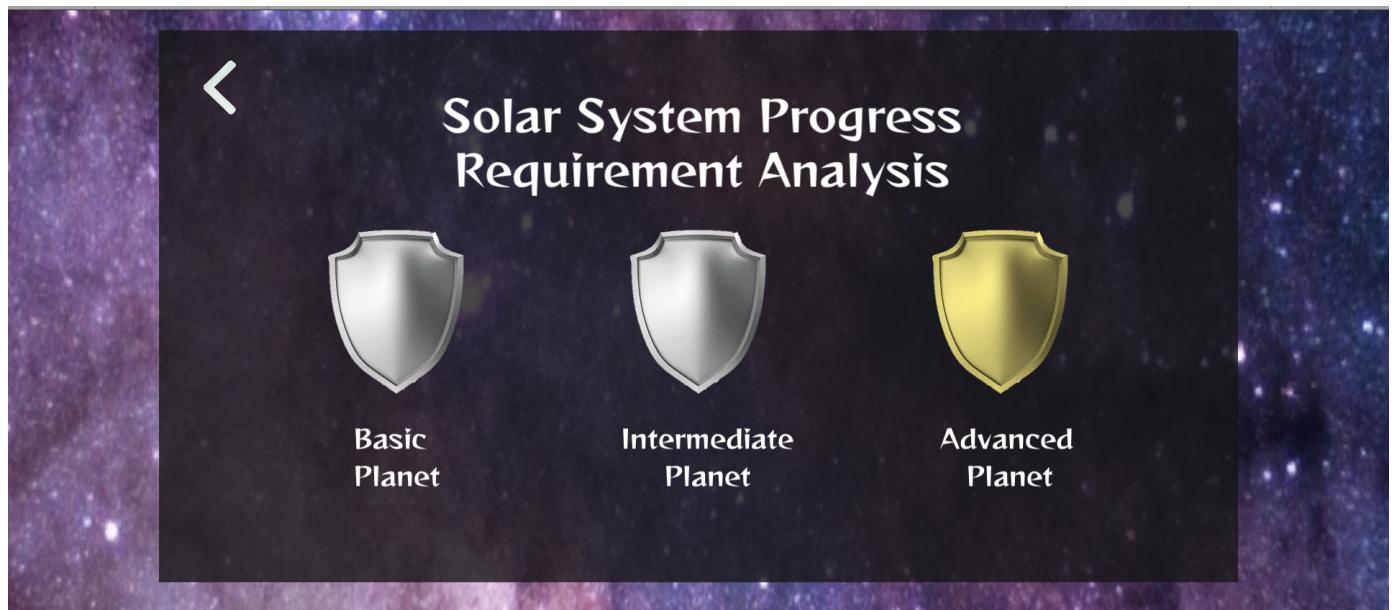
3.1.17 Softvengers Game Application Multiverse Progress Page



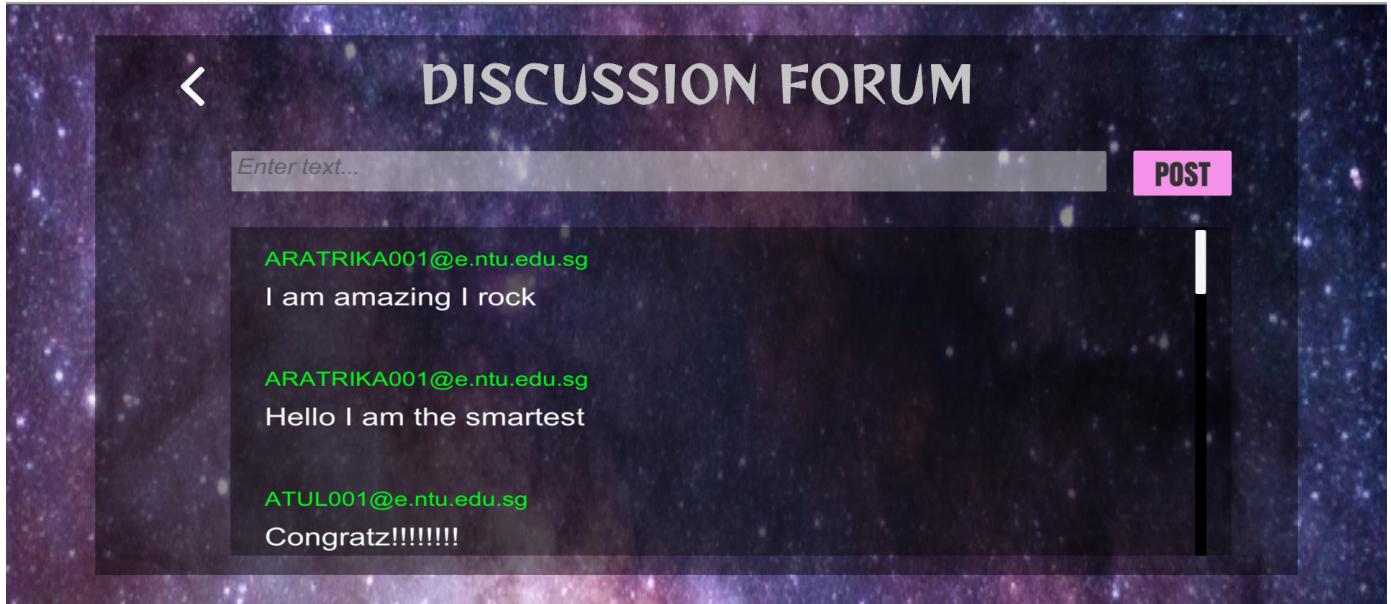
3.1.18 Softvengers Game Application Universe Progress Page



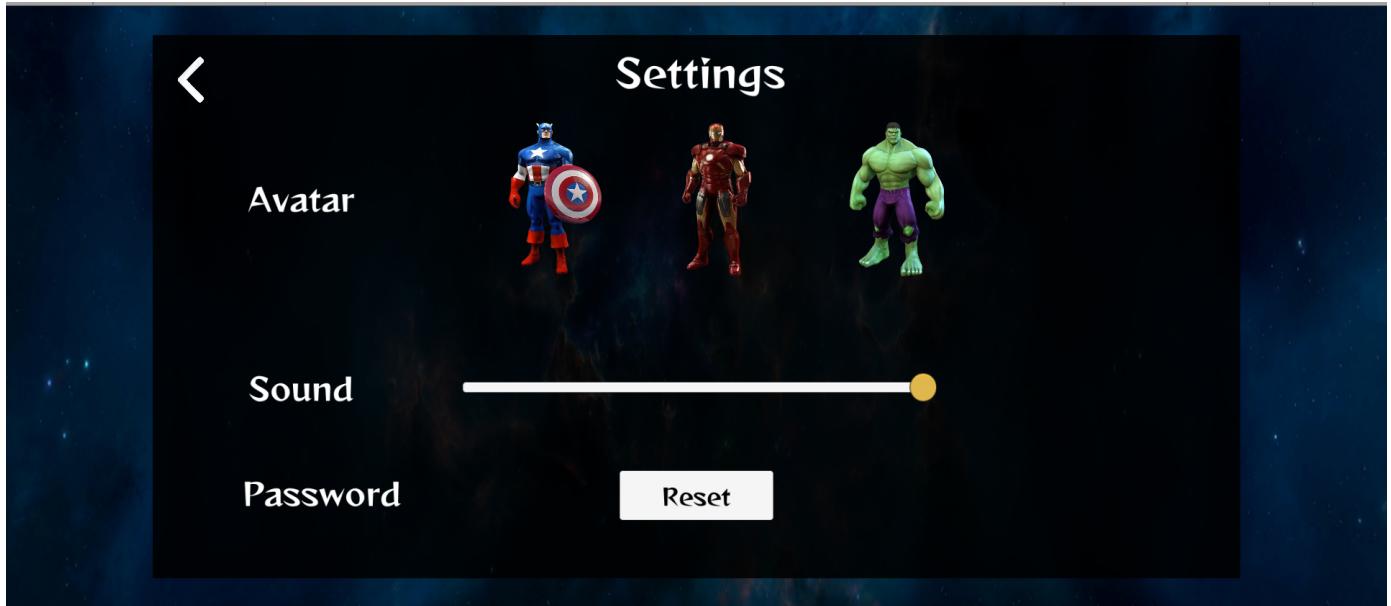
3.1.19 Softvengers Game Application Solar System Progress Page



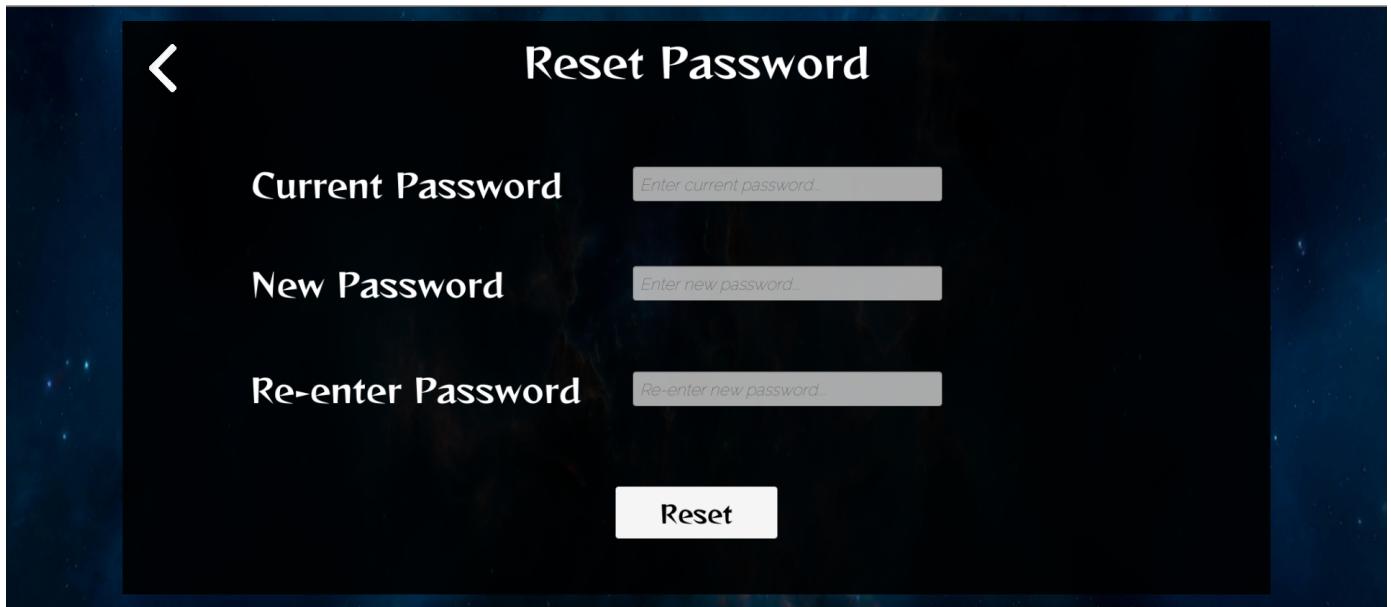
3.1.20 Softvengers Game Application Discussion Forum



3.1.21 Softvengers Game Application Settings



3.1.22 Softvengers Game Application Reset Password Page



3.1.23 Softvengers Teacher Webapp Login Page



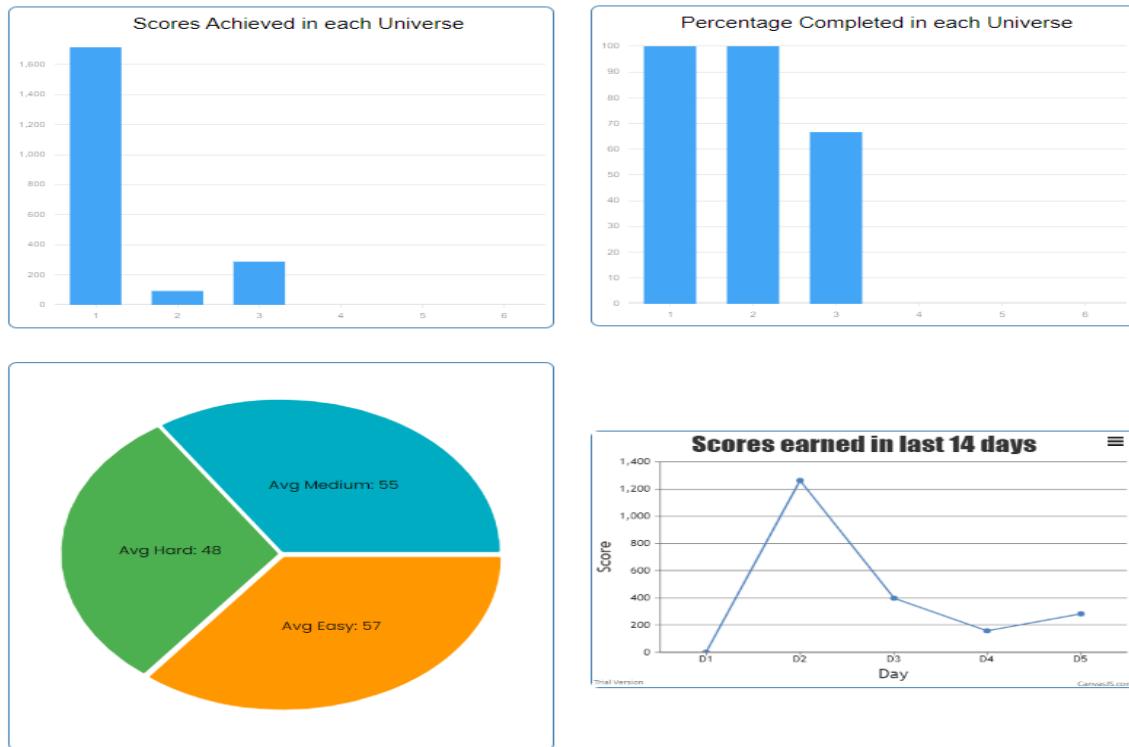
A screenshot of a web-based login form. At the top, it says "Please Log In". Below that, there is a label "Email:" followed by an input field containing "Enter Teacher Email". To the right of the input field is a small blue icon. Below the email field is a label "Password:" followed by an input field containing "Enter Password". To the right of the password input field is another small blue icon. At the bottom of the form is a large green button with the word "Login" in white.

3.1.24 Softvengers Teacher Webapp Statistics page

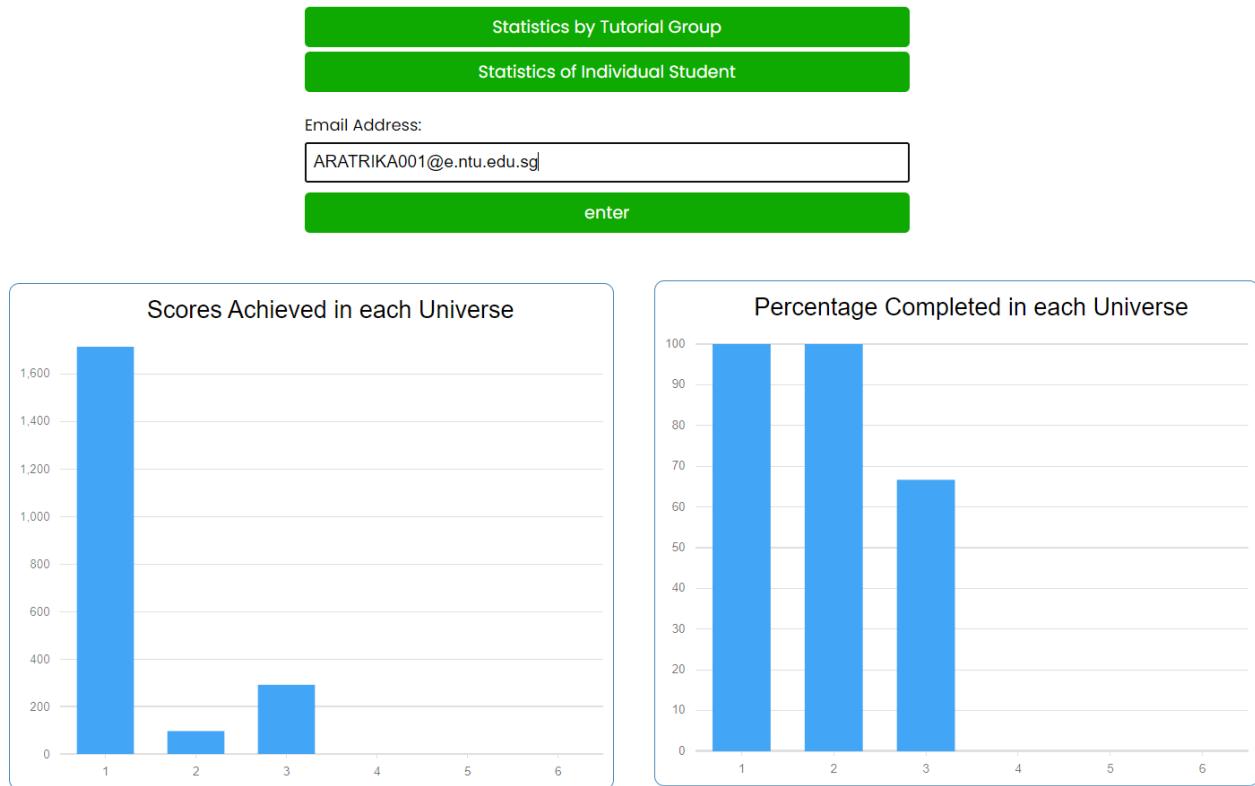
Scores Achieved in each Universe

Percentage Completed in each Universe

3.1.25 Softvengers Teacher Webapp group statistics



3.1.26 Softvengers Teacher webapp Individual statistics



3.1.27 Softvengers Teacher webapp Leaderboard page

Softvengers

Statistics Assignments QuestionBank Leaderboard Create Student Account Logout

LeaderBoard

Rank	Student Name	Total Score	Tutorial Group
Rank 1 🏆	Atul Acharya	1091.365	SCE5
Rank 2 🥈	Arat Pal	291.284	SCE5
Rank 3 🥉	Astha Garg	81.997	SCE5
Rank 4	Srish Arora	0.000	SCE5
Rank 5	Nisha Rama	0.000	SCE5

Rows per page: 5 ▾ 1-5 of 20 < >

3.1.28 Softvengers Teacher webapp Create Student Account page

Create Student Account

First Name:

Last Name:

Matric Number:

Password:

Email Address:

Tutorial Group :

Add Student

3.1.29 Softvengers Teacher webapp View Assignments page

Assignments

Assignment 129078
AKSHAT
Due : 2021-04-15T00:00:00.000Z
Timelimit: 10

Assignment 10259871
Requirements Elicitation - Week 2
Due : 2022-04-20T00:00:00.000Z
Timelimit: 15

SHARE ON TWITTER **SHARE ON REDDIT** **STATISTICS**

SHARE ON TWITTER **SHARE ON REDDIT** **STATISTICS**

3.1.30 Softvengers Teacher webapp Add Assignment

Add Assignment

Assignment ID *	Assignment Name *						
Time limit *	dd/mm/yyyy <input type="date"/>						
Tutorial Group *							
Question ID	Points	Body	Correct Option	Wrong1	ADD MORE QUES	CANCEL	SUBMIT ASSIGNMENT
wrong2	wrong3						

Requirements Elicitation - Week 2
Due : 2022-04-20T00:00:00.000Z
Timelimit: 15

SHARE ON TWITTER **SHARE ON REDDIT** **STATISTICS**

3.1.31 Softvengers Teacher webapp View Assignment statistics

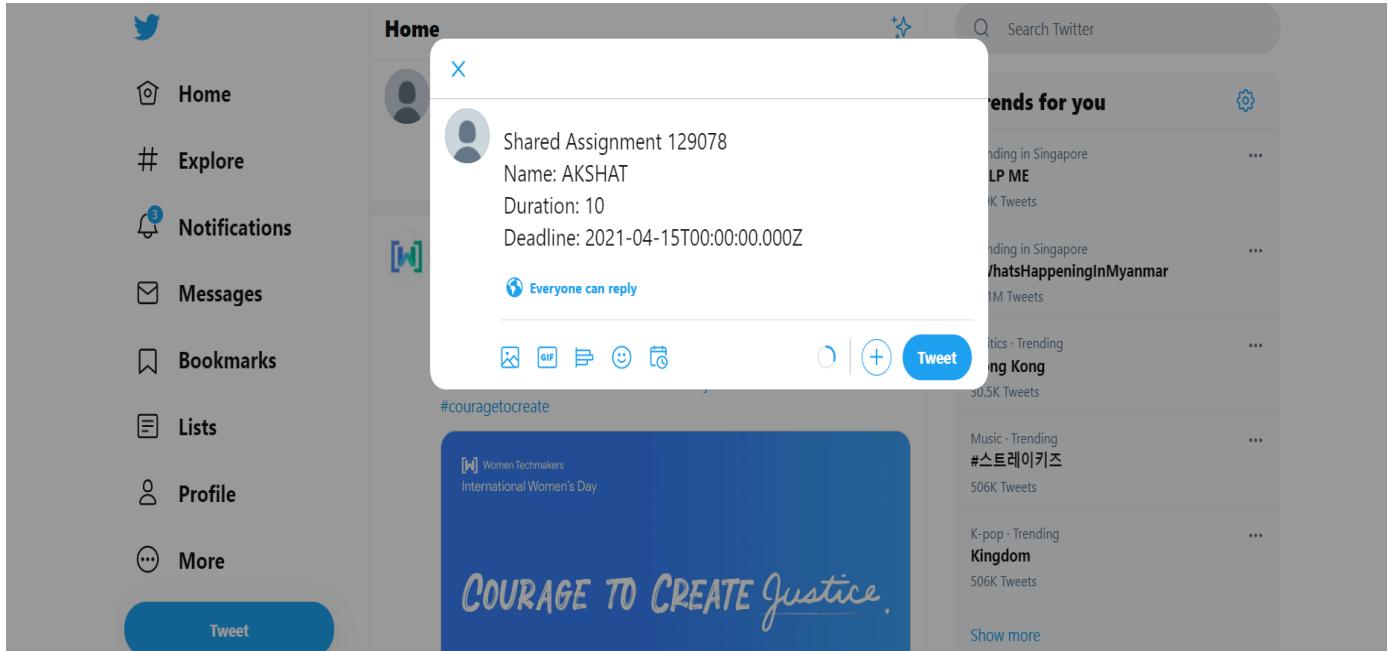


3.1.32 Softvengers teacher webapp View Assignment statistics

Max Score : 70 Min Score : 0
Mean : 23.33333333333332

Student Name	Total Score	Status
Atul Acharya	70	Attempted
Astha Garg	50	Attempted
Arat Pal	15	Attempted
SaitejaReddy Kondreddy	5	Attempted
Srish Arora	0	Not Attempted
Nisha Rama	0	Not Attempted

3.1.33 Softvengers Teacher webapp Share Assignment (Twitter)



3.1.34 Softvengers Teacher webapp Share Assignment (Reddit)

The screenshot shows the Reddit "Create a post" interface. At the top, there is a "Choose a community" dropdown, a "Post" button, and other options like "Images & Video", "Link", and "Poll". The main text area contains the following text:

Softvengers assignment

Shared Assignment 129078
Name: AKSHAT
Duration: 10
Deadline: 2021-04-15T00:00:00.000Z

Below the text area are several buttons: "+ OC", "+ SPOILER", "+ NSFW", and "FLAIR". There are also "Save Draft" and "Post" buttons. A checkbox "Send me post reply notifications" is checked, and a link "Connect accounts to share your post" is visible.

To the right of the main post area, there is a sidebar titled "Posting to Reddit" with the following steps:

1. Remember the human
2. Behave like you would in real life
3. Look for the original source of content
4. Search for duplicates before posting
5. Read the community's rules

At the bottom of the sidebar, it says "Please be mindful of reddit's content policy and practice good reddiquette."

The footer of the page includes links to "Help", "About", "Reddit App", "Careers", "Reddit Coins", "Press", "Reddit Premium", "Advertise", "Reddit Gifts", "Blog", "Terms", "Content Policy", "Privacy Policy", and "Mod Policy". It also states "Reddit Inc © 2021. All rights reserved".

3.1.35 Softvengers Teacher webapp View Question bank page

Question Bank

[ADD NEW QUESTION](#)

Q300
Universe 0, Solar System 0,Easy Question 1

A: Correct
B: Wrong
C: C
D: D

[EDIT](#) [DELETE](#)

Q301
Universe 0, Solar System 0,Easy Question 2

A: A
B: B
C: C
D: D

3.1.36 Softvengers Teacher webapp Add New Question page

Add Question

Universe * Solar *

Planet * QuestionID *

Planet Question *

Select the Checkbox with the correct option:

Option 1 *

Option 2 *

Option 3 *

Option 4 *

[CANCEL](#) [UPDATE QUIZ](#)

Question Bank

[ADD NEW QUESTION](#)

Q300
Universe 0, Solar System 0,Easy Question 1

A: Correct
B: Wrong
C: C
D: D

Q301
Universe 0, Solar System 0,Easy Question 2

A: A

3.1.37 Softvengers Teacher webapp Edit Question

Question Bank

Q300
Universe 0, Solar System 0, Easy Question 1
A: Correct
B: Wrong
C: C
D: D

Q301
Universe 0, Solar System 0, Easy Question 2
A: A

Edit Question

Universe *
0

Solar *
0

Planet *
0

Planet question *
Universe 0, Solar System 0,Easy Question 1

Select the Checkbox with the correct option:

Option 1 *
 Correct

Option 2 *
 Wrong

Option 3 *
 C

Option 4 *
 D

DELETE [CANCEL](#) [UPDATE QUESTION](#)

3.2 Hardware Interfaces

The Softvengers Game Application requires a compatible laptop or desktop that supports the Windows or MacOS operating platform with Unity. The device must have internet connection capability.

The Softvengers Web Application is designed to run on any frequency used browsers such as Google Chrome, Mozilla Firefox, Microsoft Edge, etc. The device must have internet connection capability.

3.3 Software Interfaces

Component	Software	Version	Purpose
Database	MongoDBAtlas	4.5.5	For storing Sofvengers application data

3.4 Communications Interfaces

3.4.1. Usage of HTTP Protocol for client-server communication:- The protocol followed for client server communication is the HTTP protocol. It is simple to learn and use, and also extensible. The client can send various requests such as GET, POST, PUT, PATCH or DELETE to the server through this protocol. After server-side processing, the server returns a response to the client. To access the private routes in the server, the client must include Bearer Token Authentication in the requests that it sends. This provides security to the routes.

3.4.2. Usage of SMTP Protocol for email sending:- When the teacher creates a new student account on our system, the student must be notified about his/her login credentials through an email. Hence the SMTP protocol is used for sending emails to the students upon account creation.

4. System Features

4.1 Web application for Teachers

1. The Teacher shall be able to Interact with the System

1.1 Teacher shall be able to login to their registered account

1.1.1 The Teacher must enter his/her NTU email address.

1.1.2 The Teacher must enter his/her password.

1.1.3 The system must be able to validate Teacher credentials.

1.1.4 The system must be able to query the database to authenticate and process Teacher login requests.

1.1.5 If the information of any field is incorrect, the system must display the error message “Email address/Password is filled in incorrectly, please try again.” and prompt the Teacher to try again.

1.1.6 If the Teacher is successfully logged in, the system must redirect the Teacher to the Home Page.

1.1.7 Upon logging in, the system must have a sign out option.

1.2 The System must allow the Teacher to Create an account for a Student.

1.2.1 Teacher must input the **Student name**.

1.2.1.1 The system must verify that the Student name must be alphabetical.

- 1.2.1.2 The system must verify that the Student name must be of at least 3 characters and less than 25 total characters.
 - 1.2.2 Teacher must enter the **NTU email address** of the Student as the username.
 - 1.2.3 Teacher must enter the matriculation number of the Student.
 - 1.2.3.1 The system must verify that the matriculation number must be 9 characters long.
 - 1.2.3.2 The system must set the matriculation number as the **password**.
 - 1.2.4 The system must be able to validate that all fields have been properly filled.
 - 1.2.5 If the information of any field is incorrect or in the wrong format, the system must display the error message “Name/Email address/Matriculation Number is filled in incorrectly, please try again.” and prompt the Teacher to try again.
 - 1.2.6 If the registration is successful, the system must display a ‘Successfully Registered’ message.
- 1.3 The Teacher must be able to view the question bank as a list of questions.
 - 1.3.1 The Teacher must be able to add a new question.
 - 1.3.1.1 The Teacher must add the text of the question.
 - 1.3.1.1.1 The text of the question must be alphanumeric.
 - 1.3.1.1.2 The text of the question must be between 10 characters and 200 characters long.
 - 1.3.1.2 The Teacher must add 4 options.
 - 1.3.1.2.1 The options must be between 10 characters and 200 characters long.
 - 1.3.1.3 The Teacher must select one option as the correct option.
 - 1.3.2 The Teacher must be able to modify the existing questions on the following basis.
 - 1.3.2.1 The Teacher must be able to modify the text of the question.
 - 1.3.2.2 The Teacher must be able to set a different option as the correct option.
 - 1.3.2.3 The Teacher must be able to change the text of the options.

1.3.2.4 The Teacher must be able to delete the question.

1.4 The Teacher must be able to view the statistics of the student's performance.

1.4.1 The Teacher shall be able to view statistics individually or by tutorial group.

1.4.2 The System must show statistics through the following graphs at the individual or group level.

1.4.2.1 The System must display a line chart showing "**Points achieved**" vs "**Time in days**"

1.4.2.2 The System must display a bar graph showing "**Phases of SDLC**" vs "**percentage completed**".

1.4.2.3 The System must display a bar graph showing "**Phases of SDLC**" vs "**Total points achieved**".

1.4.2.4 The System must display a Pie chart showing "**percentage of novice, intermediate, advanced questions completed**".

1.5 The System must allow the Teacher access to the leaderboard.

1.5.1 The System must display the following information on the leaderboard.

1.5.1.1 The System must display the student's position.

1.5.1.2 The System must display the student's name.

1.5.1.3 The System must display the student's total points accumulated.

1.6 The Teacher must be able to create an assignment on the System.

1.6.1 The Teacher must be able to add a new question to the assignment.

1.6.1.1 The Teacher must add the text of the question.

1.6.1.1.1 The text of the question must be alphanumeric.

1.6.1.1.2 The text of the question must be between 10 characters and 200 characters long.

1.6.1.1.3 The Teacher must add 4 options.

- 1.6.1.1.3.1 The options must be between 10 characters and 200 characters long.
- 1.6.1.1.4 The Teacher must select one option as the correct option.
- 1.6.1.1.5 The Teacher must assign points to the question.
 - 1.6.1.1.5.1 The points assigned to a question must be between 5 and 20 points.
- 1.6.2 The Teacher must set a time limit for the assignment.
 - 1.6.2.1 The time limit for the assignment must be between 15 minutes and 60 minutes.
- 1.6.3 The Teacher must set a deadline for the assignment.
 - 1.6.3.1 The deadline must contain the date in DD/MM/YYYY format.
 - 1.6.3.2 The deadline must contain the time in HH:MM format.
- 1.7 The Teacher must be able to view the performance of students in the assignment.
 - 1.7.1 The System must display the names of students who attempted the assignment in green.
 - 1.7.2 The System must display the following statistics about the assignment.
 - 1.7.2.1 The System must display the mean score.
 - 1.7.2.2 The System must display the maximum score.
 - 1.7.2.3 The System must display the minimum score.

4.2 Students side Functional Requirements

- 1. The student must be able to login to the system
 - 1.1 The user must be able to enter their NTU login details
 - 1.1.1 The student must be able to enter their email ID
 - 1.1.2 The student must be able to enter their password

- 1.1.3 If the login credentials entered are incorrect, system must display
an error message and ask them to re-enter credentials
2. The student must be able to change settings
 - 2.1 The student must be able to choose an avatar from the given set of avatars
 - 2.2 The student must be able to modify the volume of the background music using a slider.
 - 2.3 The student must be able to change password
 - 2.3.1 The student must enter the current password
 - 2.3.2 The student must enter the new password
 - 2.3.3 The student must confirm the new password
 - 2.3.4 The system must validate the new password and confirmation password.
3. The student shall be able to play the game in the solo mode.
 - 3.1 The student shall be able to enter a universe within the solo mode.
 - 3.1.1 The student must be able to enter a solar system when inside a universe
 - 3.1.1.1 The student must be able to choose a planet to save once inside a solar system.
 - 3.1.1.1.1 The system must display a total of 10 multiple-choice questions one at a time.
 - 3.1.1.1.1.1 Each question must have one correct option and 3 incorrect options.
 - 3.1.1.1.1.2 The student must be able to select an option and click the submit button to submit the answer for that question.

3.1.1.1.3 The system must be able to play an animation of destroying an asteroid if the correct answer is submitted

3.1.1.1.4 The system must be able to play an animation of the asteroid crashing into the planet if the incorrect is submitted

3.1.1.1.4.1 The system should reduce the HP of the planet by 20 points.

3.1.1.1.5 The system must analyse the student's performance in real-time and adaptively modify the difficulty of the questions asked.

3.1.1.1.6 The system should record the user's correct streak and incorrect streak

3.1.1.1.7 If the student chooses the correct answer and the question difficulty is the same as the planet difficulty, the value of the correct streak increases by one and the value of the incorrect streak becomes zero.

3.1.1.1.8 If the student chooses the correct answer and the question difficulty is lesser than the planet difficulty, the incorrect streak decreases by one

3.1.1.1.9 If the student chooses the incorrect answer and the question difficulty is the same as the planet difficulty, the value of the incorrect streak increases by one and the value of the correct streak becomes zero.

3.1.1.1.10 If the student chooses the incorrect answer and the question difficulty is greater than the planet difficulty, the correct streak decreases by one

3.1.1.1.11 If the correct streak reaches 6, the system should increase the difficulty level of the questions.

3.1.1.1.12 If the incorrect streak reaches 3, the system should decrease the difficulty level of the questions.

3.1.1.1.13 The system must ensure that the student cannot go back to the previous questions

3.1.1.1.2 The system calculates scores based on the student's performance on the planet which is reflected in the leaderboard rankings

3.1.1.1.2.1 The system must award each right answer with a

base score.

3.1.1.1.2.1.1 This base score is predefined and is characteristic of the level of difficulty of the question (more difficult question awards more score).

3.1.1.1.2.1.2 The base score must not be affected by the time the student takes to answer the question.

3.1.1.1.2.2 The system must award a time bonus which is inversely proportional to the time that student takes in answering the question.

3.1.1.1.2.2.1 A score of 0.1 is deducted from maximum possible time bonus for every second the student takes to answer the question.

3.1.1.1.2.2.2 The maximum possible time bonus must be characteristic of the difficulty level of the question

3.1.1.1.2.2.3 The minimum possible time bonus must be 0 i.e time bonus must not be negative

3.1.1.1.3 The system must display a timer

3.1.1.2.3.1 Once the timer reaches zero, the system must award 0 for the time bonus for that question

3.1.1.1.4 The total score for each question is the sum of the base score and the time bonus

3.1.1.1.5 The student must answer at least half of the questions correctly to save the planet.

3.1.1.1.6 The student must be awarded badges to recognize their success in rescuing the planet.

3.1.1.1.6.1 The student must be awarded a bronze badge if 50-70 percent of questions in an attempt have been answered correctly.

3.1.1.6.2 The student must be awarded a silver badge if 70-90

percent of questions in an attempt have been answered

correctly (70 not inclusive).

3.1.1.6.3 The student must be awarded a gold badge if 90-100 percent of questions in an attempt have been answered correctly (90 not inclusive)

3.1.1.7 The student must be awarded a mastery stone for a universe if they receive gold badges for every planet in that particular universe

3.1.1.2 Each planet can be unlocked if the Student has rescued the previous planet.

3.1.1.3 If all planets are rescued, then the current solar system is rescued

3.1.2 Each solar system is unlocked only if the Student has rescued the previous system

3.1.3 If all solar systems are rescued, then the current universe is rescued

3.2 Each universe is unlocked only if the previous universe has been rescued.

4. The student shall be able to play the game in the challenge mode.

4.1 The student shall be able to create a new challenge to send to other players.

4.1.1 The student shall be able to enter the challenge details.

4.1.1.1 The student shall be able to enter information in the card for challenge topics.

4.1.1.1.1 The student shall be able to choose a universe from a dropdown.

4.1.1.1.2 The student shall be able to choose a solar system for that universe from a dropdown.

4.1.1.1.3 The student shall be able to choose a level of difficulty for the questions from a dropdown.

4.1.1.1.4 The student shall be able to choose the number of questions from a dropdown.

4.1.1.2 The student shall be able to add a new card to the challenge topics list.

4.1.2 The student must attempt the challenge before it is sent to other students

4.1.2.1 The system shall display the questions one-by-one to the student.

4.1.2.2 The system must calculate the student's score for the challenge as a cumulative of their score for each question.

4.1.3 The student must be able to send a challenge to multiple students by selecting their names from a drop down

4.1.4 The student must be able to share the challenge details on social media

4.2 The student shall be able to view their sent challenges

4.2.1 Each challenge shall display the name of the challenge details, the student's own score, the attempt status of the students they have sent it to, and their scores.

4.3 The student shall be able to view their received challenges

4.3.1 Each challenge shall display the name of the challenge details and the sender's details.

4.3.2 The student shall be able to attempt or decline the challenge via the attempt button for each challenge.

4.3.2.1 If the challenge is denied, it disappears from the received challenges list.

4.3.2.2 If the challenge is accepted, the student must play the challenge immediately.

5. The student shall be able to view posts about challenges on social media.

5.1 The post shall contain the details about the challenge.

6. The student shall be able to view the leaderboard

6.1 The system shall display the student's leaderboard rank at the top of the page.

6.2 The system should display the list of the top 10 students in the tutorial group along with their scores in a table format in decreasing order of their scores.

7. The student shall be able to play the game in the assignment mode.

7.1 The student shall be able to view assignments sent by the teacher.

7.2 The system shall sort the assignments by attempt status.

7.2.1 The completed assignment shall display the assignment details and the student's attempt score

7.2.2 The pending assignments shall display the assignment details and the due date.

7.2.2.1 The student shall be able to attempt the pending assignments by clicking on the attempt now button.

7.2.2.1.1 The system shall display the questions shall be displayed one-by-one.

7.2.2.1.1.1 Each correct answer must be allocated the score set by the teacher.

7.2.2.2 The assignment shall be allocated a score based on the accumulated score of each question.

7.2.3 If the student does not attempt an assignment before its due date, the system shall assign a score of 0 to the student for that assignment.

8. The student shall be able to view posts about newly posted assignments on social media.

8.1 The posts shall contain information about the assignment details and its due date.

9. The student shall be able to view their progress

9.1 The system shall display a progress bar corresponding to each universe

9.1.1 On clicking any of progress bars, the system shall display progress bars corresponding to each solar system in the universe

9.1.1.1 On clicking on any of the solar system progress bars, the system shall display each badge earned by the student for each of the

Planets

9.1.1.1.1 A gold badge for a planet shall contribute to 33% progress for that solar system.

9.1.1.1.2 A silver badge for a planet shall contribute to 22% progress for that solar system

9.1.1.1.3 A bronze badge for a planet shall contribute to 11% progress for that solar system.

9.1.1.2 The progress of the solar system contributes to a progress of (100/

total number of solar systems)) to each universe progress bar

9.2 The system shall display an infinity stone badge next to the corresponding universe progress bar once a progress of 100 percent has been achieved.

10. The student shall be able to access the tutorial via the Tutorial button in the Home Page

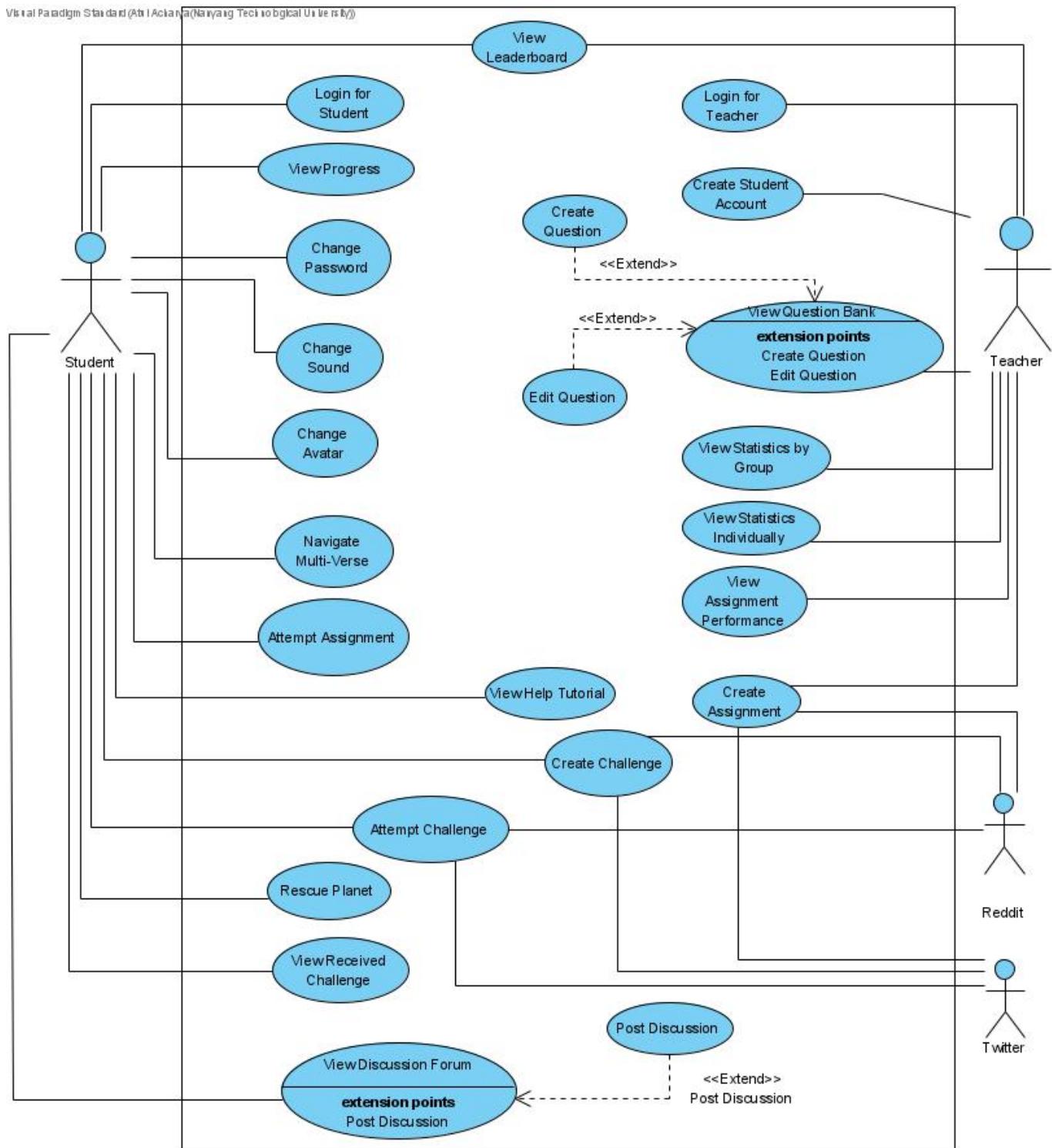
10.1 The system should open a Youtube Video in the browser when the button is pressed

11. The student shall be able to access the discussion forum

11.1 The student shall be able to send messages to other students in the same tutorial group.

11.2 The student shall be able to view received messages from other students in the same tutorial group in real-time.

4.3 Use case Diagram:



4.4 Use Case Descriptions:

Use Case ID:	4.2.1		
Use Case Name:	Login for student		
Created By:	Aks	Last Updated By:	Nisha
Date Created:	06/02/2021	Date Last Updated:	10/04/2021

Actor:	Student
Description:	The student must be able to sign in to their account in the game
Preconditions:	The student must have a valid account in the game
Postconditions:	The student is successfully logged in to their account
Priority:	
Frequency of Use:	Once per session
Flow of Events:	<ol style="list-style-type: none"> 1. The student enters their NTU email ID 2. The student enters their password 3. The system validates the credentials 4. The student is successfully logged in to their account

Alternative Flows:	<p>AF-S3: If the user enters incorrect login credentials</p> <ol style="list-style-type: none"> 1. The system displays an error message “Incorrect email ID and password! Please enter again!” and prompts the user to enter the login details again 2. The system returns to step 1 of main flow
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	4.2.2.1		
Use Case Name:	Change avatar		
Created By:	Aks	Last Updated By:	Nisha
Date Created:	08/02/2021	Date Last Updated:	12/04/2021

Actor:	Student
Description:	The student must be able to change their avatar in the game
Preconditions:	The student has logged in to his account in the game
Postconditions:	The avatar is set to the one desired by the student
Priority:	
Frequency of Use:	Once per month
Flow of Events:	<ol style="list-style-type: none"> 1. The student opens the settings option in the homepage. 2. The student selects an avatar from the available options. 3. The system sets the student's avatar to the desired one and displays the selected one on the homepage
Alternative Flows:	-
Exceptions:	-
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	4.2.2.2		
Use Case Name:	Change sound		
Created By:	Aks	Last Updated By:	Nisha
Date Created:	08/02/2021	Date Last Updated:	12/04/2021

Actor:	Student
Description:	The student must be able to adjust the level of the background sounds in the game
Preconditions:	The student has logged in to his account in the game
Postconditions:	The background volume is at the level desired by the student
Priority:	
Frequency of Use:	Once per session
Flow of Events:	<ol style="list-style-type: none"> 1. The student opens the settings menu in the game. 2. The student selects the Change volume option from the menu 3. The student moves the slider left or right to decrease or increase the volume respectively.

	4. The system sets the volume at the desired level
Alternative Flows:	-
Exceptions:	-
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	4.2.2.3		
Use Case Name:	Change password		
Created By:	Aks	Last Updated By:	Nisha
Date Created:	08/02/2021	Date Last Updated:	12/04/2021

Actor:	Student
Description:	The student must be able to change their password in the game
Preconditions:	The student has logged in to his account in the game
Postconditions:	The password is successfully updated in the database and the student can use the new credentials to login to their account
Priority:	
Frequency of Use:	Once every six months
Flow of Events:	<ol style="list-style-type: none"> 1. The student opens the settings menu in the game. 2. The student selects the Change password option from the menu 3. The system redirects the student to the Change Password page 4. The student enters their current password, which is then authenticated by the system. 5. The student enters their new password. 6. The student re-enters their new password. 7. The system updates the student's password in the database.
Alternative Flows:	<p>AF-S4: The entered password is not valid</p> <ol style="list-style-type: none"> 1. The system displays an error message “Current password is incorrect, please try again!” to the student. 2. The system returns to step 4 of main flow. <p>AF-S5: The new password entered is an empty string</p>

	<ol style="list-style-type: none"> 1. The system displays an error message “New password field cannot be left empty, please try again!” to the student. 2. The system returns to step 5 of main flow. <p>AF-S6: The re-entered password does not match the first entry</p> <ol style="list-style-type: none"> 1. The system displays an error message “The passwords don’t match, please try again!” to the student. 2. The system returns to step 6 of main flow.
Exceptions:	-
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	4.2.3.1		
Use Case Name:	Navigate Multiverse		
Created By:	Atul	Last Updated By:	Nisha
Date Created:	08/02/2021	Date Last Updated:	12/04/2021

Actor:	Student
Description:	The student navigates the multiverse and can select a planet of a solar system of a universe to rescue.
Preconditions:	<ol style="list-style-type: none"> 1. The student must be logged in to the system
Postconditions:	<ol style="list-style-type: none"> 1. The student enters the planet
Priority:	1
Frequency of Use:	
Flow of Events:	<ol style="list-style-type: none"> 1. The student enters the multiverse 2. The system displays all the universes 3. The student navigates the multiverse and chooses a universe to explore. 4. The student enters the universe 5. The system displays all the solar systems of the universe. 6. The student navigates the universe and chooses a solar system to explore. 7. The student enters the solar system. 8. The system displays all the planets of the solar system 9. The student chooses a planet to rescue.
Alternative Flows:	<p>AF-S3 The student has not unlocked the universe</p> <ol style="list-style-type: none"> 1. The system displays the name of the universe in red <p>AF-S6 The student has not unlocked the solar system</p> <ol style="list-style-type: none"> 1. The system displays the the name of the solar system in red

	AF-S9 The student has not unlocked the planet 1. The system displays the name of the planet in red
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	4.2.3.1.1.1.1		
Use Case Name:	Rescue Planet		
Created By:	Atul	Last Updated By:	Nisha
Date Created:	08/02/2021	Date Last Updated:	12/04/2021

Actor:	Student
Description:	The student rescues the planet by answering questions
Preconditions:	<ul style="list-style-type: none"> 1. The student must be logged in to the system 2. The student must have performed the 'Navigate Multiverse' use case.
Postconditions:	<ul style="list-style-type: none"> 1. The student either saves the planet or fails and the planet gets destroyed.
Priority:	1
Frequency of Use:	
Flow of Events:	<ul style="list-style-type: none"> 1. The system displays a question, 3 incorrect options and 1 correct option. 2. The student selects an option. 3. The system records a score based on the accuracy of the answer and the time taken to answer the question. The system also displays an animation depending on whether the attempt is correct or not 4. The system selects a question of the same difficulty level 5. The flow from 1-4 is repeated for a total of 10 times 6. The system displays the score of the student for each question along with the attempt status (correct/incorrect/not attempted) 7. The student rescues the planet if he answers 50% of the questions correctly
Alternative Flows:	<p>AF-S5 The student answers 50% of the questions incorrectly.</p> <ul style="list-style-type: none"> 1. The health bar has a value of 0 and the student is directed to the challenge results screen <p>AF-S4 The student answers 3 questions incorrectly.</p>

	<p>1. The system selects a question of a lower difficulty level</p> <p>AF-S4 The student answers 6 questions correctly</p> <p>1. The system selects a question of a higher difficulty level</p>
Exceptions:	<p>EX-1 The database does not have question for the topic</p> <p>1. The system must display the message “The teacher is yet to upload the question bank. Please try again later”.</p> <p>2. The system goes back to the “Select Planet” UI</p>
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	4.2.4.1
Use Case Name:	Create challenge

Created By:	Srishti	Last Updated By:	Nisha
Date Created:	08/02/2021	Date Last Updated:	12/04/2021

Actor:	Student (Initiating Actor)
Description:	<p>The student must be able to customise the number of questions he gets from every topic and select their difficulty. He should be given questions according to his customisation. After he has completed the quiz his final score and the time taken to complete the quiz must be displayed.</p> <p>He should then be able to challenge other students on social media by sharing his score, level and number of questions of every topic.</p>
Preconditions:	The student must be logged in and must have selected the challenge mode from the main menu.
Postconditions:	<ol style="list-style-type: none"> 1. Other students receive the challenge in their received challenges screen and have an option to accept or deny the challenge. 2. The challenger can view the sent challenge under the sent tab in view challenges screen after successfully creating the challenge.
Priority:	
Frequency of Use:	
Flow of Events:	<ol style="list-style-type: none"> 1. The student selects the universe, the solar system, and the difficulty level. 2. The student attempts the challenge. 3. System displays the score and time taken by the student. 4. The student selects students from a checkbox list to challenge.

	5. System sends the challenge to the respective students. 6. The student shares the challenge on social media. 7. System displays the challenge on social media.
Alternative Flows:	
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	4.2.4.3
Use Case Name:	View received challenge

Created By:	Pratyush	Last Updated By:	Nisha
Date Created:	09/02/2021	Date Last Updated:	15/04/2021

Actor:	Student (Initiating Actor)
Description:	The student must be able to view the received challenges and must be able to accept or reject the same.
Preconditions:	<ul style="list-style-type: none"> 1. The student must be logged in and must have been on the received tab of the view challenges screen.
Postconditions:	<ul style="list-style-type: none"> 1. The student is able to view all information regarding pending and completed challenges. and 2. The student plays the challenge after confirming to start it Or 3. The student no longer sees challenges that have been declined
Priority:	-
Frequency of Use:	-
Flow of Events:	<ul style="list-style-type: none"> 1. System displays the challenge details (refer to sec. 5 of functional requirements) for pending and completed challenges

	<ol style="list-style-type: none"> 2. Student can accept a pending challenge 3. System displays a confirmation dialog to start the challenge 4. Student confirms to start the challenge 5. System starts the challenge
Alternative Flows:	<p>AF-S2 The student declines a pending challenge</p> <ol style="list-style-type: none"> 1. System displays a confirmation dialog to delete the challenge 2. Student confirms that the challenge is to be declined 3. System sets the challenge status as 'declined' for the student once the student confirms. The challenge is never displayed to the student again
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	4.2.4.1.2
--------------	-----------

Use Case Name:	Attempt challenge		
Created By:	Aks	Last Updated By:	Nisha
Date Created:	09/02/2021	Date Last Updated:	15/04/2021

Actor:	Student
Description:	The student answers questions from the challenge.
Preconditions:	<ol style="list-style-type: none"> 1. The student must be logged in to the system 2. The student must have created a challenge or they must be attempting an accepted challenge.
Postconditions:	The challenge is completed and the student's score is displayed on their screen.
Priority:	1
Frequency of Use:	Once every week
Flow of Events:	<ol style="list-style-type: none"> 1. The system displays a question, 3 incorrect options and 1 correct option. 2. The student selects an option. 3. The system records a score based on the accuracy of the answer and the time taken to answer the question. The system also displays an animation depending on whether the attempt is correct or not 4. The flow from 1-3 for total of 10 times 5. The system displays the score of the student for each question along with the attempt status (correct/incorrect/not attempted)

Alternative Flows:	
Exceptions:	<p>EX-1 The database does not have question for the topic</p> <ol style="list-style-type: none"> 1. The system must display the message “The teacher is yet to upload the question bank. Please try again later”. 2. The system goes back to the “View Challenge” UI
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	4.2.7		
Use Case Name:	Attempt assignment		
Created By:	Nisha	Last Updated By:	Nisha
Date Created:	09/02/2021	Date Last Updated:	15/04/2021

Actor:	Student (Initiating Actor)
--------	----------------------------

Description:	A student must be able to attempt an assignment given by a teacher via social media.
Preconditions:	<ol style="list-style-type: none"> 1. The student must be logged in to the System. 2. The assignment must be displayed under the pending tab.
Postconditions:	<ol style="list-style-type: none"> 1. The student has attempted assignment 2. The assignment must be displayed under the completed tab with the corresponding score for it
Priority:	Medium
Frequency of Use:	Monthly
Flow of Events:	<ol style="list-style-type: none"> 1. The student clicks on the “Attempt Assignment” button 2. The system displays a time bar corresponding to the time set by the teacher per question 3. The system displays a question with 4 options 4. The student chooses an option 5. The system assigns a score for the question 6. The flow from steps 2-5 is repeated for all the questions of the assignment 7. Once the assignment has been completed, the system displays the assignment with a completed status along with the time taken and score
Alternative Flows:	
Exceptions:	<p>EX-1: If student attempts the assignment past the deadline</p> <ol style="list-style-type: none"> 1. The assignment is displayed under the completed tab with a score of 0 as the deadline has been missed

Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	4.2.9		
Use Case Name:	View Progress		
Created By:	Nisha	Last Updated By:	Nisha
Date Created:	11/02/2021	Date Last Updated:	15/04/2021

Actor:	Student
Description:	A student can view his/her progress in terms of the number of gold, silver and bronze badges earned for each universe and solar system. The student can also see the badge earned for each level as well as the infinity stone badges earned for each level

Preconditions:	<ol style="list-style-type: none"> 1. The student must be logged in
Postconditions:	<ol style="list-style-type: none"> 1. The student has viewed progress made
Priority:	
Frequency of Use:	
Flow of Events:	<ol style="list-style-type: none"> 1. The student clicks the 'View Progress' button 2. The system displays progress bars corresponding to each universe along with the infinity stone badges earned 3. The student clicks on any of the universe progress bars 4. The system displays progress bars corresponding to each solar system in each universe 5. The student clicks on any of the solar system progress bars 6. The system displays the badge earned by the student for each of the planets in the solar system
Alternative Flows:	<p>AF-S9: If student has not earned a badge for a particular level yet</p> <ol style="list-style-type: none"> 1. The system displays a translucent badge
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	

Notes and Issues:	
-------------------	--

Use Case ID:	4.2.10		
Use Case Name:	View Help Tutorial		
Created By:	Atul	Last Updated By:	Nisha
Date Created:	11/02/2021	Date Last Updated:	18/04/2021

Actor:	Student
Description:	A student can view a tutorial video which explains how to use the application
Preconditions:	1. The student must be logged in to the system
Postconditions:	1. The student is directed to a YouTube video on his/her default browser
Priority:	Low

Frequency of Use:	Once a month
Flow of Events:	<ol style="list-style-type: none"> 1. The student clicks the “Help” button 2. The system opens the Softvengers Tutorial Video on Youtube in the student’s default browser
Alternative Flows:	
Exceptions:	EX1 - URL not found <ol style="list-style-type: none"> 1. The web page displays a 404 Status
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	4.2.11.2		
Use Case Name:	View Discussion Forum		
Created By:	Nisha	Last Updated By:	Nisha
Date Created:	11/02/2021	Date Last Updated:	18/04/2021

Actor:	Student
Description:	The student must be able to view messages posted by other students
Preconditions:	The student has logged in to his account in the game
Postconditions:	The student has viewed messages posted by all students in the same tutorial group as the student
Priority:	
Frequency of Use:	Weekly
Flow of Events:	<ol style="list-style-type: none"> 1. The student clicks the discussion forum option on the homepage 2. System displays the messages posted by all students in the same tutorial group with a timestamp
Alternative Flows:	-
Exceptions:	-
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	4.2.11.1		
Use Case Name:	Post Discussion		
Created By:	Nisha	Last Updated By:	Nisha
Date Created:	11/02/2021	Date Last Updated:	18/04/2021

Actor:	Student
Description:	The student must be able to post a message on the discussion forum
Preconditions:	View Discussion Forum use case has been invoked
Postconditions:	The student has posted a message on the discussion forum
Priority:	
Frequency of Use:	Weekly
Flow of Events:	1. The student enters a message and clicks the submit button

	2. The system displays the new message at the top of the discussion forum with a timestamp
Alternative Flows:	-
Exceptions:	-
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	4.1.1.1		
Use Case Name:	Login		
Created By:	Aratrika	Last Updated By:	Aratrika
Date Created:	8/2/2021	Date Last Updated:	12/04/2021

Actor:	Teacher (Initiating Actor)
Description:	The Teacher logs into the System using their NTU Email address and password.
Preconditions:	<ul style="list-style-type: none"> 1. The Teacher must have an NTU Email address. 2. The Teacher's NTU Email address and password must be stored in the System's database.
Postconditions:	<ul style="list-style-type: none"> 1. The Teacher is logged into the Home Page of the System. 2. The System shows a sign out option on the Home Page.
Priority:	High priority. This use case needs to be performed before any other use cases can be performed by the Teacher.
Frequency of Use:	Frequently used. This use case is performed every time the Teacher needs to use the app.
Flow of Events:	<ul style="list-style-type: none"> 1. The System must display a login form with 2 parameters: email and password. 2. The Teacher must enter his/her NTU email address. 3. The Teacher must enter his/her password. 4. The Teacher must press the “Login” button. 5. If all input fields are filled, the System must query the database to authenticate the credentials and process the login request. 6. If all input fields are authenticated, the System must log the Teacher into the application.
Alternative Flows:	<p>AF-S4. If the NTU Email address or password or both fields are empty</p> <ul style="list-style-type: none"> 1. The system displays the message “Not all required fields have been filled, Please try again.” 2. The system will return to step 1. <p>AF-S6. If the NTU Email address or password or both fields are incorrect and failed authentication</p>

	<ol style="list-style-type: none"> 1. The System displays the message “Email address/Password is filled in incorrectly, please try again.” 2. The system will return to step 1.
Exceptions:	<p>EX1: If the connection to the database is lost midway of processing</p> <ol style="list-style-type: none"> 1. The system displays “Connection to database failed, account was not created. This could be due to database maintenance or connectivity issues. Please ensure you are connected to the internet or try again later. Thank you”. 2. The system will return to step 1.
Includes:	
Special Requirements:	<ol style="list-style-type: none"> 1. Usability: Error messages must be written in English, in a clear, concise manner.
Assumptions:	<ol style="list-style-type: none"> 1. The Teacher's information is already present in the System database.
Notes and Issues:	

Use Case ID:	4.1.1.2
Use Case Name:	Creation of Student Account

Created By:	Saiteja	Last Updated By:	Akshat
Date Created:	8/2/2021	Date Last Updated:	12/04/2021

Actor:	1. Teacher (Initiating Actor)
Description:	The Teacher creates accounts for students of his/her class.
Preconditions:	1.The Student doesn't have an account registered with the System from before.
Postconditions:	1.The Student account has been Successfully registered with the system by the Teacher.
Priority:	High
Frequency of Use:	Frequently used in the first two weeks of every semester.
Flow of Events:	<ol style="list-style-type: none"> 1. The Teacher selects the “Create Student Account” button 2. The system displays the registration form. 3. The Teacher inputs the Student name. 4. The Teacher inputs the NTU Email Address. 5. The Teacher inputs the password. 6. The Teacher clicks on the “Submit” button. 7. If the system validates that the input fields are correct, the system successfully creates the account in its database 8. The system displays a message informing the Teacher that registration is successful.
Alternative Flows:	<p>AF-S6: If input fields are left empty</p> <ol style="list-style-type: none"> 1. The system displays the message “Not all required fields have been filled, Please try again.”

	<p>2. The system will return to step 5</p> <p>AF-S6: If input fields are filled incorrectly</p> <ol style="list-style-type: none"> 1. The system displays the message “Not all fields have been filled correctly, Please try again.” 2. The system will return to step 5 <p>AF-S6: Account with the same username already exists</p> <ol style="list-style-type: none"> 1. The system displays the message “An account with the same username already exists. Please choose another username.” 2. The system will return to step 5
Exceptions:	<p>EX1: If the connection to the database is lost midway of processing</p> <ol style="list-style-type: none"> 3. The system displays “Connection to database failed, account was not created. This could be due to database maintenance or connectivity issues. Please ensure you are connected to the internet or try again later. Thank you”. 4. The system will return to step 2.
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	4.1.1.3		
Use Case Name:	View Question Bank		
Created By:	Astha	Last Updated By:	Akshat
Date Created:	8/2/2021	Date Last Updated:	12/04/2021

Actor:	Teacher (initiating actor)
Description:	The teacher views the question bank.
Preconditions:	The teacher has successfully logged in to his/her account.
Postconditions:	The system displayed the question bank and filtered it as desired by the teacher.
Priority:	Medium
Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> 1. The Teacher selects the “Question Bank” option from the displayed menu. 2. The system displays a drop down menu of the different phases in SDLC. 3. The Teacher selects one of the displayed options. 4. The system displays a drop down list of the subtopics in the chosen phase of SDLC.

	<ol style="list-style-type: none"> 5. The Teacher selects one subtopic from those being displayed. 6. The Teacher clicks the proceed button. 7. The system displays three horizontal tabs for easy, medium and hard questions. 8. The Teacher clicks on one of the tabs to view the full list of questions under that selection. 9. The system displays the text of the questions that fall into the selected category in a list format.
Alternative Flows:	<p>AF-S7: If there are no questions present in the selected subtopic.</p> <ol style="list-style-type: none"> 1. The system will display a message informing the Teacher that no question has been added. 2. The system will display an option to the Teacher to add a question to the current category. <p>AF-S8: If there are no questions present in either easy, medium and/or hard tabs.</p> <ol style="list-style-type: none"> 1. The system will display a message informing the Teacher that no question has been added. 2. The system will display an option to the Teacher to add a question to the tab of the chosen difficulty level.
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	4.1.1.3.1		
Use Case Name:	Add Question		
Created By:	Aratrika	Last Updated By:	Akshat
Date Created:	8/2/2021	Date Last Updated:	12/04/2021

Actor:	Teacher(Initiating Actor)
Description:	The Teacher adds a new question into the question bank.
Preconditions:	<ol style="list-style-type: none"> 1. The Teacher has successfully logged in to his/her account. 2. The Teacher has performed the “View Question Bank” use case.
Postconditions:	<ol style="list-style-type: none"> 1. The new question is added to the System database. 2. The new question is shown in the list of questions in the question bank.
Priority:	High
Frequency of Use:	Frequency high towards the beginning of the semester but expected to decrease as the semester proceeds.
Flow of Events:	<ol style="list-style-type: none"> 1. The Teacher must click the ‘+’ button to add a new question. 2. The Teacher must add the text of the question. 3. The Teacher must add 4 options. 4. The Teacher must mark one option as the correct option.

	<ol style="list-style-type: none"> 5. The Teacher must click on the “Add Question” button. 6. If all the fields are filled properly, the System must add the question to the database. 7. The System must display the message “New Question added successfully.”
Alternative Flows:	<p>AF-S6. If any of the fields are missing</p> <ol style="list-style-type: none"> 1. The System must display the message “Not all required fields have been filled, Please try again.” 2. The system will return to step 2. <p>AF-S6. If the text of the question is incorrectly filled</p> <ol style="list-style-type: none"> 1. The System must display the message “The text of the question must be alphanumeric and between 10 characters and 200 characters long.” 2. The System must return to step 2. <p>AF-S6. If the text of any option is incorrectly filled</p> <ol style="list-style-type: none"> 1. The System must display the message “The text of the question must be between 10 characters and 200 characters long.” 2. The System must return to step 3.
Exceptions:	
Includes:	
Special Requirements:	<ol style="list-style-type: none"> 1. Usability: Error messages must be written in English, in a clear, concise manner.
Assumptions:	
Notes and Issues:	

Use Case ID:	4.1.1.3.2		
Use Case Name:	Edit Question		
Created By:	Astha	Last Updated By:	Akshat
Date Created:	10/02/2021	Date Last Updated:	15/04/2021

Actor:	Teacher
Description:	The teacher edits the contents, correct answer or other options of the chosen question.
Preconditions:	<ol style="list-style-type: none"> 1. The teacher successfully logged in to his/her account. 2. The system displayed the correct question bank according to the options chosen by the teacher. 3. The question to be edited already exists in the database.
Postconditions:	<ol style="list-style-type: none"> 1. The teacher successfully edited the selected question. 2. The change is reflected in the database and the display.
Priority:	Low
Frequency of Use:	Rarely

Flow of Events:	<ol style="list-style-type: none"> 1. The teacher clicks the button to edit the question. 2. The teacher can select radio buttons for changing the phase, subtopic and difficulty. 3. The teacher then modifies the content of the question. 4. The teacher modifies the correct answer. 5. The teacher modifies the wrong options. 6. The teacher clicks the save button to save the changes done. 7. The system goes back to the view of the question bank with the changes being reflected.
Alternative Flows:	<p>AF-S6. If any of the fields are missing</p> <ol style="list-style-type: none"> 1. The System must display the message “Not all required fields have been filled, Please try again.” 2. The system will return to step 2. <p>AF-S6. If the text of the question is incorrectly filled</p> <ol style="list-style-type: none"> 1. The System must display the message “The text of the question must be alphanumeric and between 10 characters and 200 characters long.” 2. The System must return to step 2. <p>AF-S6. If the text of any option is incorrectly filled</p> <ol style="list-style-type: none"> 1. The System must display the message “The text of the question must be between 10 characters and 200 characters long.” 2. The System must return to step 3.
Exceptions:	
Includes:	
Special Requirements:	<ol style="list-style-type: none"> 1. Usability: Error messages must be written in English, in a clear, concise manner.
Assumptions:	

Notes and Issues:	
-------------------	--

Use Case ID:	4.1.1.4		
Use Case Name:	View Statistics Individually		
Created By:	Saiteja	Last Updated By:	Akshat
Date Created:	10/02/2021	Date Last Updated:	15/04/2021

Actor:	1. Teacher(Initiating Actor)
Description:	The Teacher shall be able to view statistics individually.
Preconditions:	1. The Teacher has successfully logged in to his/her account.
Postconditions:	1. System Displays Individual Statistics Successful.
Priority:	High
Frequency of Use:	Frequently

Flow of Events:	<ol style="list-style-type: none"> 1. The Teacher Selects the “View Statistics” button. 2. The System will display the following options for the teacher to select <ol style="list-style-type: none"> a. View Statistics Individually b. View Statistics by Group 3. The Teacher shall be able to Select View Statistics “View Statistics Individually” 4. System will ask Teacher to enter the “Student Name”. 5. The Teacher shall be able to enter the “Student Name”. 6. The System must display a line chart showing “Points achieved” vs “Time in days” 7. The System must display a bar graph showing “Phases of SDLC” vs “percentage completed”. 8. The System must display a bar graph showing “Phases of SDLC” vs “Total points achieved”. 9. The System must display a Pie chart showing “percentage of novice, intermediate, advanced questions completed”.
Alternative Flows:	<p>AF-S6. If Wrong Student Name is entered</p> <ol style="list-style-type: none"> 1. The System must display the message “You have entered the wrong Student Name, Please try again.” 2. The system will return to step 5. <p>AF-S6. If Student with entered Student Name is not found</p> <ol style="list-style-type: none"> 3. The System must display the message “Student Not Found, Please try again.” 4. The system will return to step 5.
Exceptions:	-
Includes:	-
Special Requirements:	-

Assumptions:	-
Notes and Issues:	-

Use Case ID:	4.1.1.4.1		
Use Case Name:	View Statistics by Group		
Created By:	Kushal	Last Updated By:	Akshat
Date Created:	10/02/2021	Date Last Updated:	15/04/2021

Actor:	1. Teacher(Initiating Actor)
Description:	The Teacher shall be able to view statistics individually or by group.
Preconditions:	1. The Teacher has successfully logged in to his/her account.
Postconditions:	
Priority:	High
Frequency of Use:	Frequently
Flow of Events:	<ol style="list-style-type: none"> 1. The Teacher Selects the “View Statistics” button. 2. The System will display the following options for the teacher to select

	<ul style="list-style-type: none"> a. View Statistics Individually b. View Statistics by Group <p>3. The Teacher shall be able to Select View Statistics “View Statistics by Group”</p> <p>4. The System will provide a dropdown menu to the teacher to filter statistics for his/her tutorial group by:</p> <ul style="list-style-type: none"> a. Group Statistics for Universe b. Group Statistics for Solar System c. Group Statistics for Planet <p>5. The Teacher will be shown 4 graphs containing the statistics:</p> <ul style="list-style-type: none"> a. line chart showing “Points achieved” vs “Time in days” b. bar graph showing “Phases of SDLC” vs “percentage completed”. c. bar graph showing “Phases of SDLC” vs “Total points achieved”. d. Pie chart showing “percentage of novice, intermediate, advanced questions completed”. <p>6. The System displays an enlarged graph when the teacher clicks on any of the four smaller graphs.</p>
Alternative Flows:	
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	<p>1. If none of the students have played the game then the statistics graphs will be empty.</p>
Notes and Issues:	

Use Case ID:	4.1.1.5		
Use Case Name:	View leaderboard		
Created By:	Akshat	Last Updated By:	Akshat
Date Created:	09/02/21	Date Last Updated:	15/04/2021

Actor:	<ol style="list-style-type: none"> 1. Teacher (Initiating Actor) 2. Student (Initiating Actor)
Description:	Students and teachers view the leaderboard to check the rankings of students
Preconditions:	<ol style="list-style-type: none"> 1. Students play the game and gain points
Postconditions:	
Priority:	Medium
Frequency of Use:	More than once in a Week

Flow of Events:	<ol style="list-style-type: none"> 1. Teacher clicks on the ‘View Leaderboard’ tab in the sidebar. 2. System displays the leaderboard consisting of: <ol style="list-style-type: none"> a. Students Position/Ranking b. Students Username c. Students total points 3. The teacher can add filters for the leaderboard to only display the rankings of students in his/her tutorial group.
Alternative Flows:	<p>AF-S1 Student clicks on the ‘View Leaderboard’ button in the game application</p> <ol style="list-style-type: none"> 1. System displays the leaderboard in game application consisting of: <ol style="list-style-type: none"> a. Students Position/Ranking b. Students Username c. Students total points
Exceptions:	<ol style="list-style-type: none"> 1. If none of the students have played the game then the leaderboard will be empty.
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	4.1.1.6		
Use Case Name:	Create Assignment		
Created By:	Aratrika	Last Updated By:	Akshat
Date Created:	11/02/21	Date Last Updated:	15/04/2021

Actor:	1. Teacher (Initiating Actor)
Description:	Teacher can create an assignment by setting questions, time-limit and a deadline and send it to his/her students.
Preconditions:	1. The Teacher must be logged into the System.
Postconditions:	1. The Assignment created is stored into the System database. 2. The Assignment is sent to the students whose names are selected by the Teacher.
Priority:	Medium
Frequency of Use:	Used once in a month when the Teacher sets Assignment

Flow of Events:	<ol style="list-style-type: none"> 1. The System must display an Assignment Creation form. 2. The Teacher must add the question text for a new question. 3. The Teacher must add 4 options for the question. 4. The Teacher must select 1 option as the correct option. 5. The Teacher must assign points to the question. 6. The Teacher must set a time-limit for the question. 7. The Teacher must set a deadline for the Assignment. 8. The Teacher repeats step 2 to step 7 for each new question to be added. 9. The Teacher presses the “Create Assignment” button. 10. The System stores the assignment questions in the database. 11. The System sends the Assignment to all the students in the tutorial group.
Alternative Flows:	<p>AF-S9. If any of the fields are missing</p> <ol style="list-style-type: none"> 1. The System must display the message “Not all required fields have been filled, Please try again.” 2. The System will return to step 2. <p>AF-S9. If any of the fields are incorrectly filled</p> <ol style="list-style-type: none"> 1. The System must display the message “Not all required fields have been filled correctly, please try again.” 2. The System will return to step 2.
Exceptions:	
Includes:	
Special Requirements:	

Assumptions:	1. The Teacher creates an assignment once a month to check the student's learning at a monthly frequency.
Notes and Issues:	

Use Case ID:	4.1.1.7		
Use Case Name:	View Assignment Performance		
Created By:	Astha	Last Updated By:	Akshat
Date Created:	11/2/2021	Date Last Updated:	15/04/2021

Actor:	Teacher(Initiating Actor)
Description:	The Teacher views the assignment performance of the students.
Preconditions:	<ol style="list-style-type: none"> 1. The Teacher has successfully logged in to his/her account. 2. The Teacher has performed the “Create Assignment” use case. 3. The Students have attempted the assignment.
Postconditions:	<ol style="list-style-type: none"> 1. The Teacher views the list of students who attempted the assignment.

	<ol style="list-style-type: none"> 2. The Teacher views the performance of each student. 3. The Teacher views the statistics of the overall performance of the students.
Priority:	Medium
Frequency of Use:	Frequency high towards the end of the semester.
Flow of Events:	<ol style="list-style-type: none"> 1. The Teacher selects the assignment for which the performance has to be checked. 2. The System displays the list of students who were supposed to take the assignment. 3. The System displays the names of students who attempted the assignment in green, with the score displayed next to it. 4. The System displays mean score, median score, maximum score, minimum score and average time for completion about the assignment.
Alternative Flows:	
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

5. Other Nonfunctional Requirements

5.1. Flexibility Requirements

Flexibility

The non-functional requirement that our system has tried to achieve is flexibility. It has done so, in the following ways:-

Independent modules:

The backend application for the Softvengers game and the Softvengers teacher website is a standalone, independent API, which is not dependent upon the frontend to work. The backend application acts as an API service for providing data access object layer abstraction and app logic operations like calculation and parsing of statistics on the teacher side. In the figure given below, we have shown how requests can be sent to the application via Postman.

Please note that the URLs for main app logic or data access can only be used when login has been successful via the login URL. This greatly enhances the security of the backend application. The backend application can be connected to any front end service following the client server architecture, hence, imparting modularity and extensibility to our software application.

http://localhost:5000/student/challenge/getSentChallenges?emailID=ARATRIKA001@e.ntu.edu.sg

GET http://localhost:5000/student/challenge/getSentChallenges?emailID=ARATRIKA001@e.ntu.edu.sg

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/> emailID	ARATRIKA001@e.ntu.edu.sg			
Key	Value	Description		

Body Cookies Headers (7) Test Results

Status: 200 OK Time: 788 ms Size: 5.03 KB Save Response

Pretty Raw Preview Visualize JSON ↻

```

11 "receivers": [
12   {
13     "attempted": 1,
14     "_id": "6069af82beed1821f588094e",
15     "emailID": "ATUL001@e.ntu.edu.sg",
16     "score": 22.133178770542145,
17     "timeTaken": 26.99176025390625
18   },
19   {
20     "attempted": 0,
21     "_id": "6069af82beed1821f588094f",
22     "emailID": "SRISHTI003@e.ntu.edu.sg",
23     "score": 0,

```

← Response

Figure: Response-request communication between client and backend application

Loose coupling between modules:

Additionally, backend code has been modularized into Routes, Controllers and Models. Here, Routes are responsible for routing of requests, Controllers contain the main function logic to process and provide a server response, and Models define the database schema. Thus, adding new Controllers and Routes is extremely easy since the code is so modular. The server side code is organized in a layered architecture format hence it processes requests in a layered top-down manner. Firstly the requests reach the index.js file and then passes through the appropriate Routes to find the required Controller which then implements function logic to send back a response. Additionally, each Controller

is only associated with a few relevant entities, instead of all entities. Hence adding new capabilities to our system is very easy, making the software very flexible.

6. System Architecture

6.1. Primary candidate architecture

Client-Server Architecture

Our selected candidate architecture is the 3 tier Client Server Architecture. At the client level, we have 2 clients: The Softvengers Game and the Teacher's WebApp. These interact with the 1 application server, which in turn interacts with the MongoDB database as shown in the figure below.

The communication between the clients and the server takes place through the usage of requests and responses that follow the HTTP protocol.

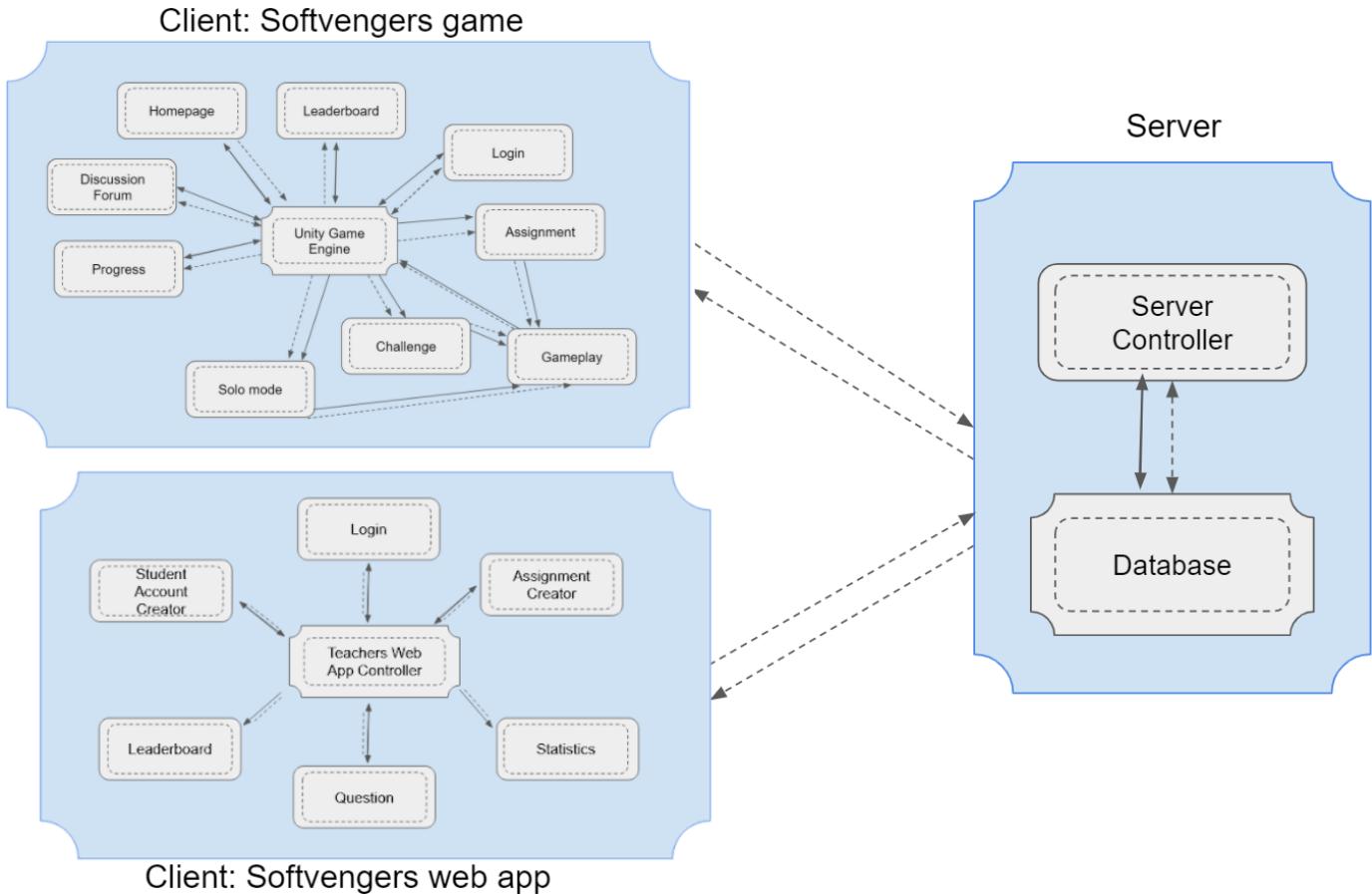


Figure: Client-server architecture diagram for the system

For the Softvengers game, the Unity Game Engine communicates with the database server by sending and receiving information and controls all the other components in that client.

This architecture allows modularity and supports modifiability since it decouples the logic of the clients and the servers. It supports future scalability since more clients can be added into this architecture in the future. It allows the 2 clients: Softvengers Game and Teacher's WebApp to carry out concurrent execution.

Lastly, this architecture makes it easy to divide work in the software development process. We divided our team into Teacher Developers, Game Developers and Server Developers to support independent development of these independent components and then their combination once they're ready. Such a work division, also facilitates focussed debugging, error fixing and maintenance.

6.1.1. Entity Game Component Architecture - Game

For the gameplay, entity game component architecture is suitable. It is an architecture that is primarily used in game design. Every object in a game scene would be an entity typically consisting of an id with a list of components attached to it. Each component would have data or a state.

Thus, the behaviour of the entities can be changed during runtime by a system that can add, modify or remove these components. The system designed for the gameplay would be loosely coupled and modifications/enhancements can be easily added to it.

This architecture uses composition rather than inheritance. It handles dependencies in a safe and simple manner. Each system registers the components an entity would require for it to perform logic on an entity. Every component type is stored as a separate list. Thus each system iterates only over objects that are relevant to it. By leveraging CPU instruction and data caches, this becomes a major performance advantage.

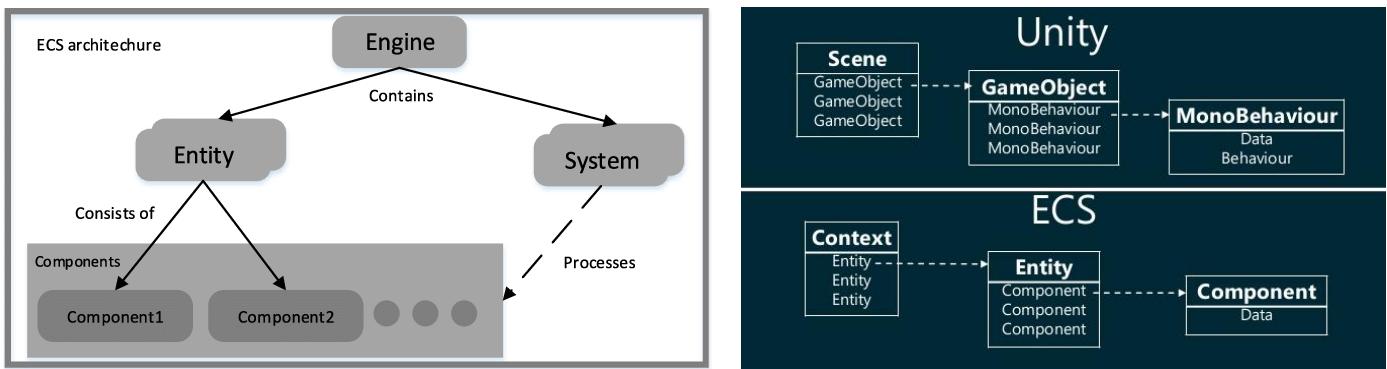


Figure: ECS Architecture

6.1.2 Model View Controller Architecture - Website

Model View Controller is the most popular architecture for web apps. The main goal of this architecture is to separate functionality, logic, and the interface of an application. Moreover, Isolation of domain logic from user interface permits independent development, testing and maintenance thus allowing separation of concerns. It allows multiple developers in our team to work on different parts of the same project. 2 developers worked on the models and controllers, while 2 worked on the views which fastened up the development process.

In the backend, there are models which define the database schema and deal with the data. The models interact with our MongoDB database. There are also controllers which have the business logic. These contain logic to handle HTTP requests. Lastly, in the frontend, we create views using React. Views are the user interface which displays the data to the user and the user can interact with it.

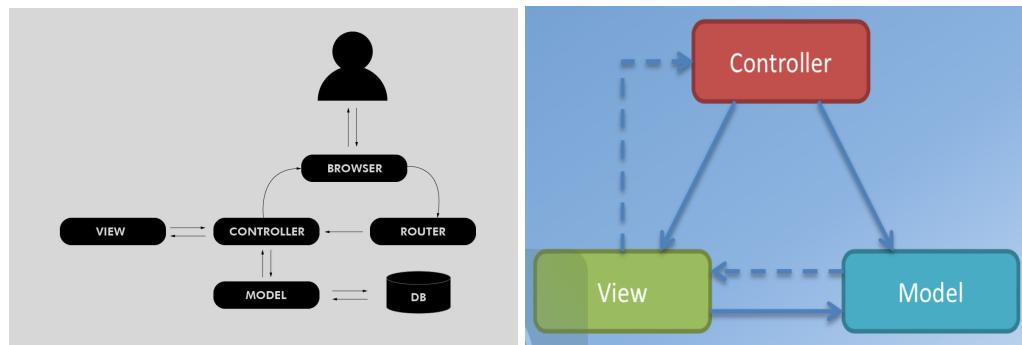


Figure: MVC Architecture

6.2. Alternate Architectures

6.2.1. Layered System Architecture

An alternative architecture that can be considered is the layered architecture whose components consists of several layers where each layer consists of several subroutines. Each component is connected by procedure calls. Reasons for considering this architecture style include:

- It supports designs based on increasing levels of abstraction
- It supports modifiability. Functionality can be enhanced by introducing new components. Lower layer components can be reused even when changes are made to the upper layers

However this architecture has been ultimately rejected due to the following reasons:

- Our application requires control to flow from one component to another and not necessarily from one layer to another. It would be difficult to find the right layers of abstraction and to design to our application such that control flows strictly from one layer to another
- Our primary non-functional requirement is performance. Layered system architecture can negatively impact performance. Each call would require some overhead. Each request would have to go through multiple layers thus reducing performance.

6.2.2. Pipe-and-filter Architecture

An alternative architecture that can be considered is the Pipe-and-filter architecture where computational components are connected through data flow links, following a directed graph topology. While this architecture would provide multiple benefits such as:

- Reusability of the components
- Easy modification by changing of component levels or addition of new components. This does not create problems since each component is unaware of the components that are upstream and downstream.
- Better performance due to a concurrent execution of processing elements, similar to an assembly line.

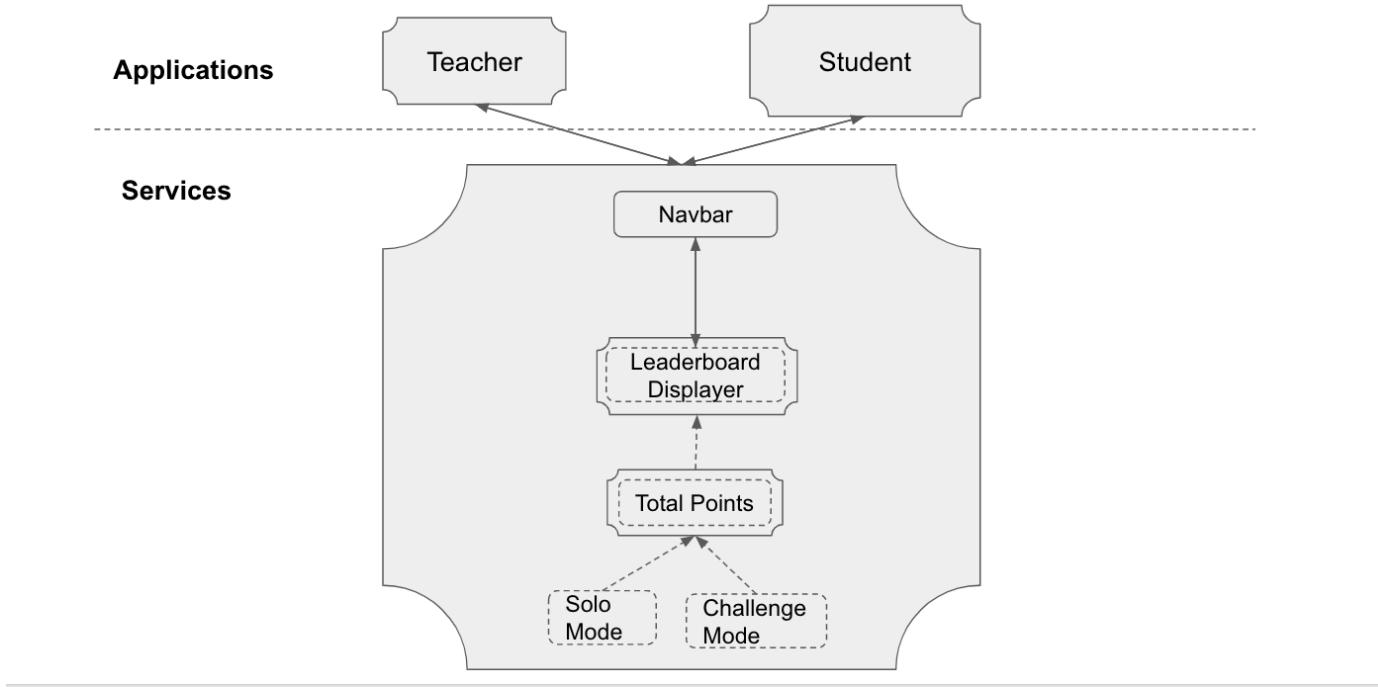
However, it had some overpowering drawbacks because of which we chose to go with the Client Server architecture instead of this:

- Pipe-and-filter architecture is not good for interactive applications since all the components must be executed before the input is transformed into output. This is a substantial disadvantage since a gaming platform must be interactive in nature.
- It is difficult for novice programmers to perform multithreaded programming to support the concurrent execution of processing elements.
- Synchronization between the components is extremely difficult to achieve, this may create bottlenecks and hence affect performance negatively.

7. Subsystem Interface

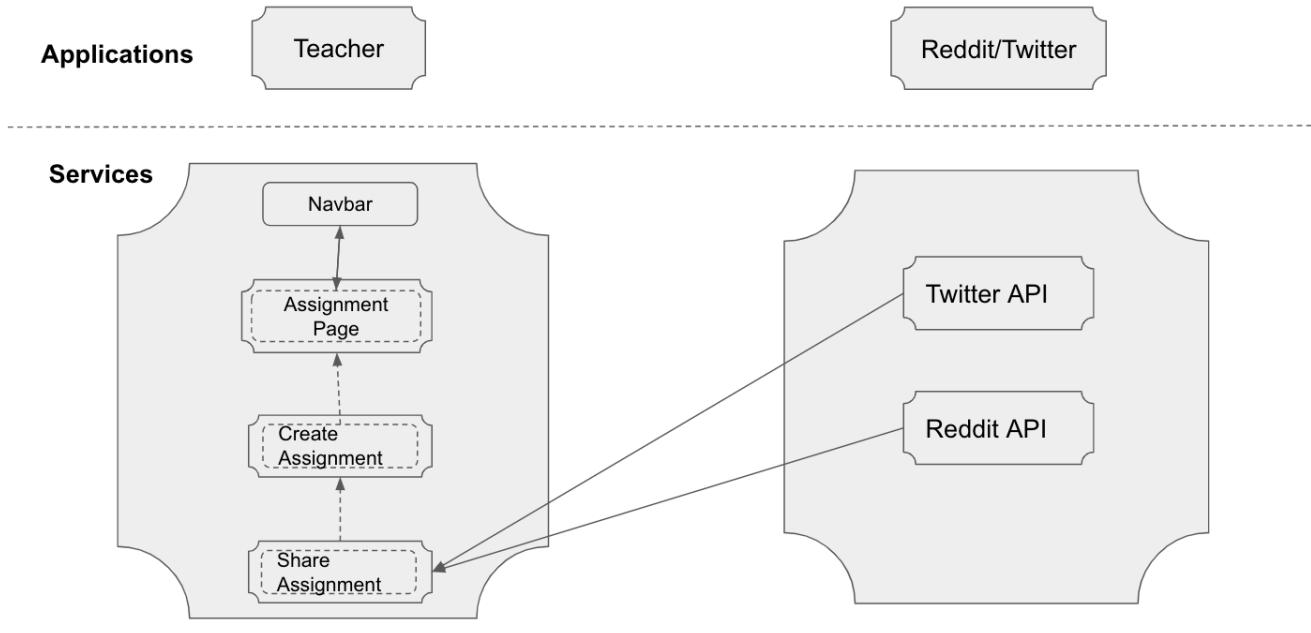
7.1. Leaderboard

Leaderboard Diagram

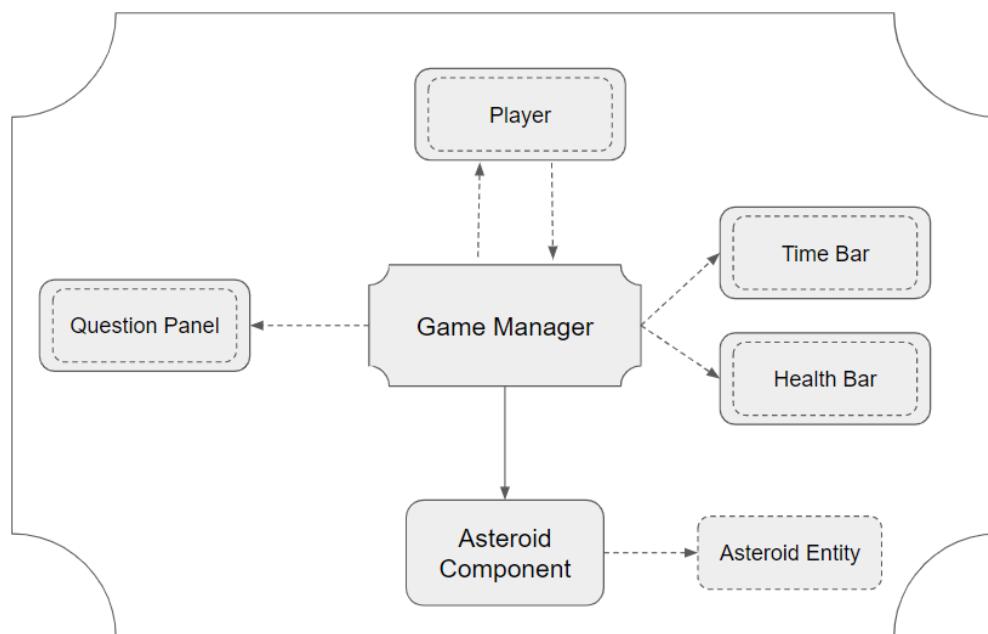


7.2. Add Assignment

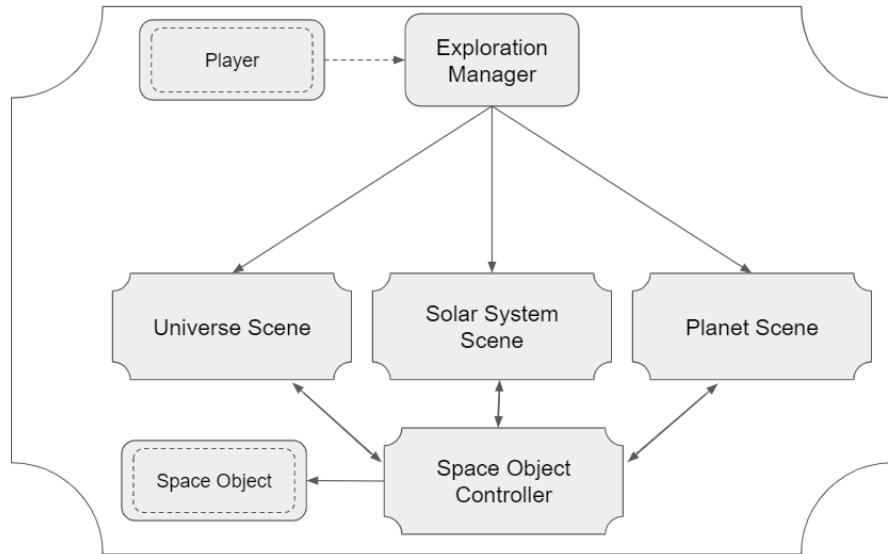
Adding Assignment



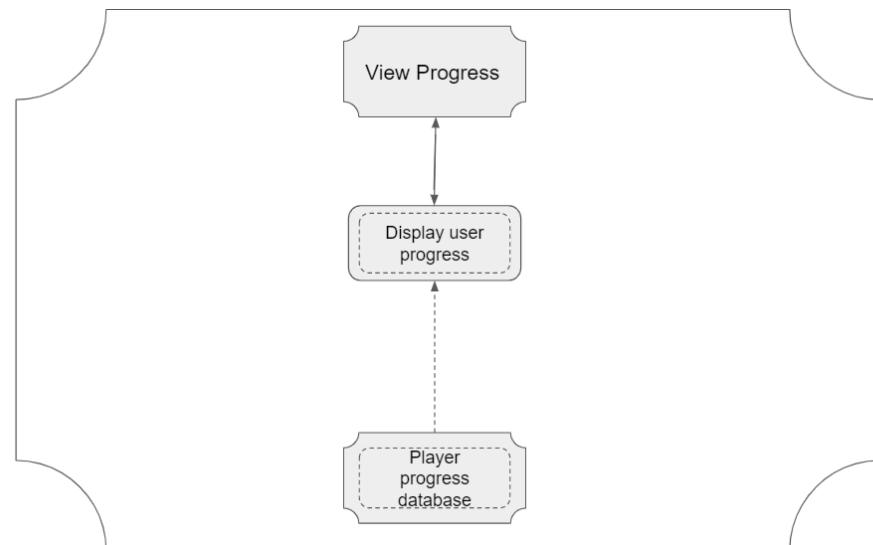
7.3. Game Play



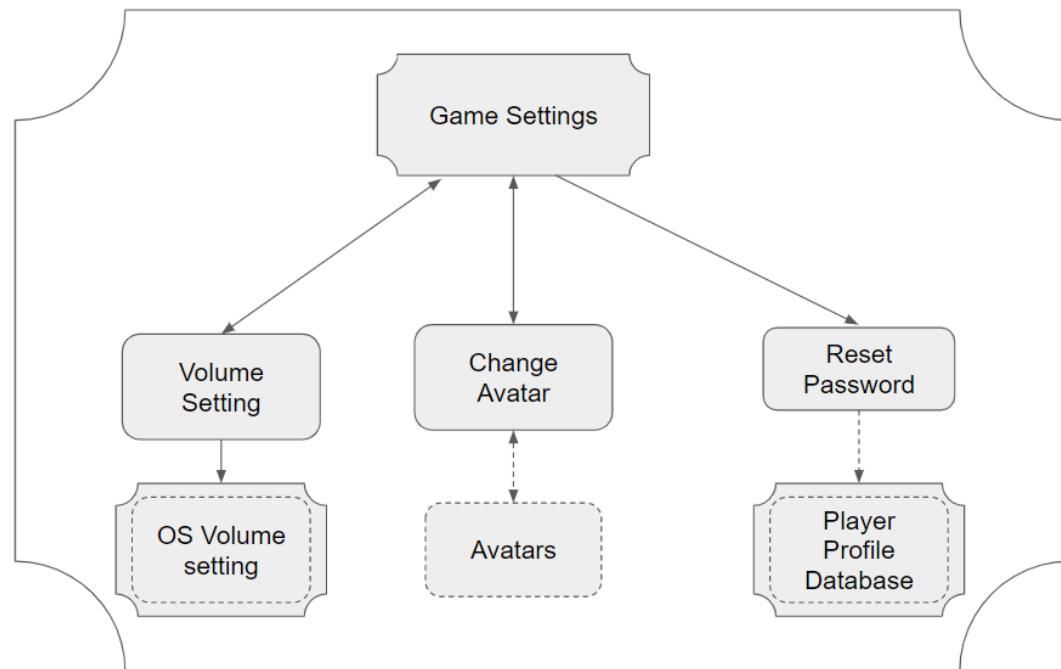
7.4. Navigation



7.5. Progress



7.6. Settings



8. Testing

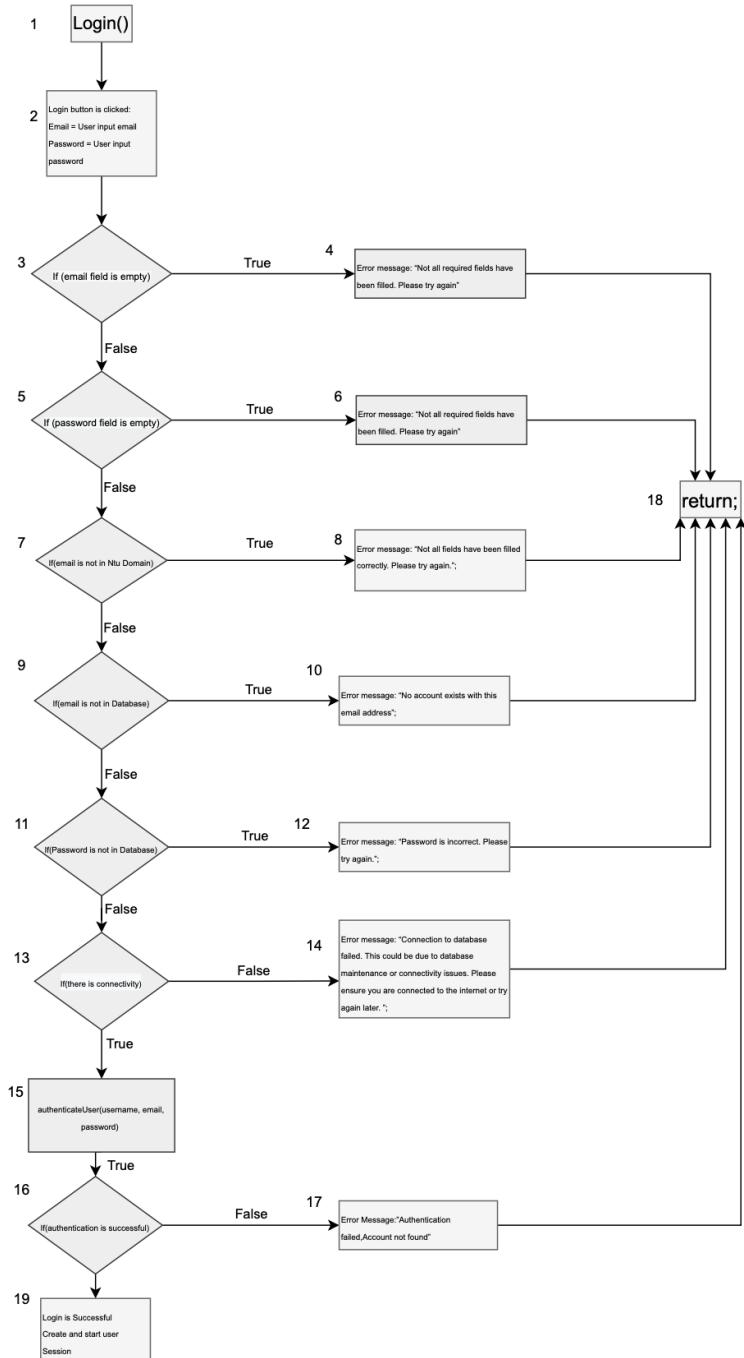
The following tools were used to test different aspects of the system:

S. No.	Type of Testing	Tool/Technique used
1	White Box testing	Control Flow Graphs and basis paths
2	Black Box testing	Equivalence classes
3	Unit Testing	Test Script in C# (Unity Test Runner) for the game and 'Mocha and Chai' in Node js for the website backend.
4	Integration testing	Scripts to mock the server(backend) for the website and in C# (Unity Test Runner) for the game.
5	Load Testing	Apache JMeter

8.1. Unit Testing

8.1.1 White Box Testing

1. Teacher Login on Website



Cyclomatic Complexity:

$\lceil \text{decisionpoint} \rceil + 1 = 7 + 1 = \mathbf{8}$

Basic Paths:

1. Baseline path: 1, 2, 3, 5, 7, 9, 11, 13, 15, 16, 19
2. Basic path 2: 1, 2, 3, 4, 18
3. Basic path 3: 1, 2, 3, 5, 6, 18
4. Basic path 4: 1, 2, 3, 5, 7, 8, 18
5. Basic path 5: 1, 2, 3, 5, 7, 9, 10, 18
6. Basic path 6: 1, 2, 3, 5, 7, 9, 11, 12, 18
7. Basic path 7: 1, 2, 3, 5, 7, 9, 11, 13, 14, 18
8. Basic path 8: 1, 2, 3, 5, 7, 9, 11, 13, 15, 16, 17, 18

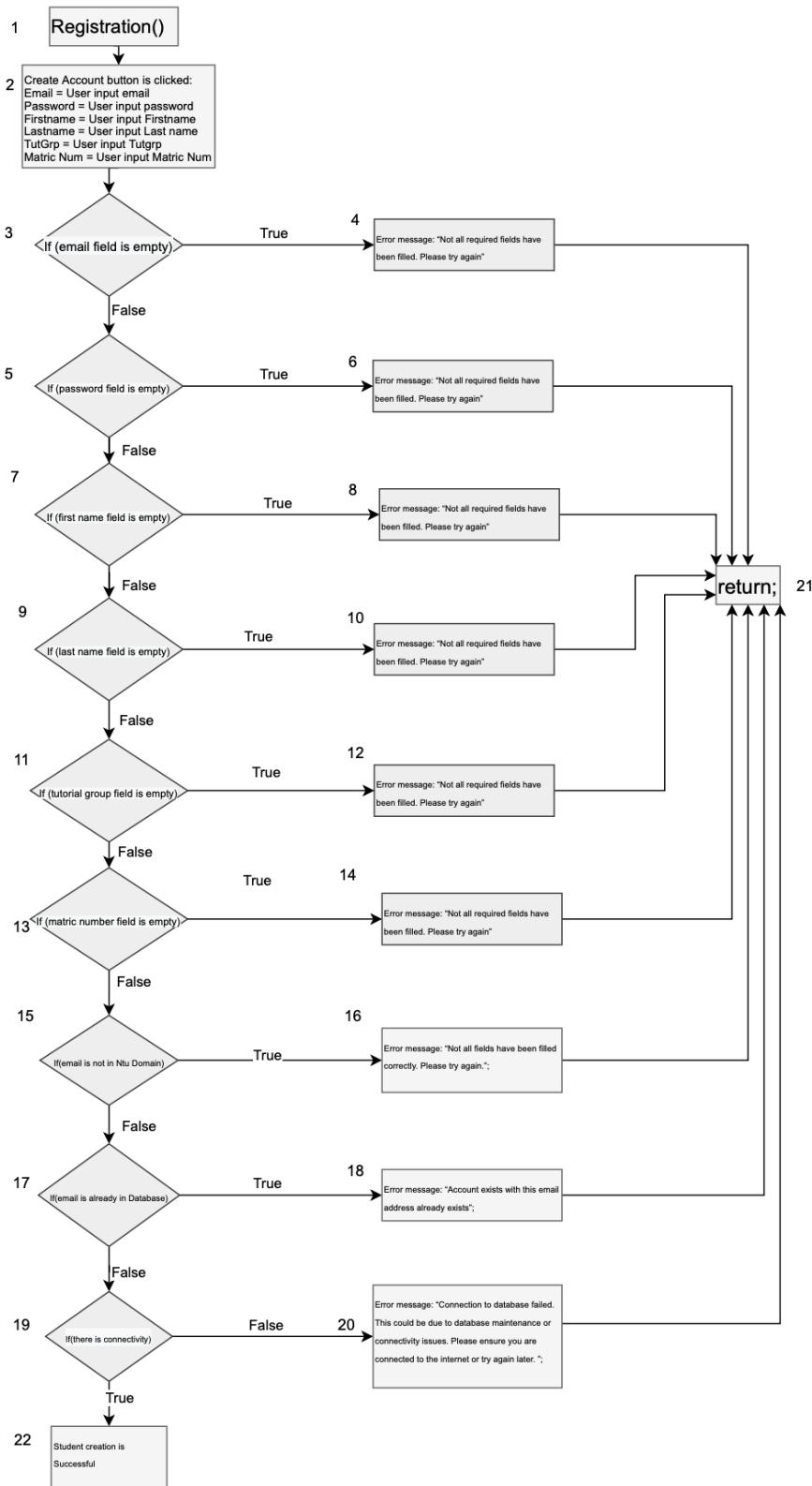
Execution Paths:

1. 1, 2, 3, 5, 7, 9, 11, 13, 15, 16, 19
2. 1, 2, 3, 4, 18
3. 1, 2, 3, 5, 6, 18
4. 1, 2, 3, 5, 7, 8, 18
5. 1, 2, 3, 5, 7, 9, 10, 18
6. 1, 2, 3, 5, 7, 9, 11, 12, 18
7. 1, 2, 3, 5, 7, 9, 11, 13, 14, 18
8. 1, 2, 3, 5, 7, 9, 11, 13, 15, 16, 17, 18

Test Cases (I assume email= SU001@e.ntu.edu.sg.com,password= 123456)

1. email = “ ” ; password: “123456” ; internet switched on
2. email = “123@e.ntu.edu.sg.com” ; password: “ ” ; internet switched on
3. email = “123@e.ntu.edu.sg”; password: “123456” ; internet switched on
4. email = “123@e.ntu.edu.sg.com” ; password: “!” ; internet switched on
5. email = “hello@hotmail.com” (non-existent email) ; password: “123456” ; internet switched on
6. email = ““123@e.ntu.edu.sg.com” ; password: “1234567890” (incorrect password) ; internet switched on
7. email = “123@e.ntu.edu.sg.com” ; password: “123456” ; internet switched off
8. email = “123@e.ntu.edu.sg.com” ; password: “123456” ; internet switched on

2. Creating Student Account



Cyclomatic Complexity:

$|decisionpoint| + 1 = 9 + 1 = \underline{\text{10}}$

Basic Paths:

1. Baseline path: 1, 2, 3, 5, 7, 9, 11, 13, 15, 17, 19, 22
2. Basic path 2: 1, 2, 3, 4, 21
3. Basic path 3: 1, 2, 3, 5, 6, 21
4. Basic path 4: 1, 2, 3, 5, 7, 8, 21
5. Basic path 5: 1, 2, 3, 5, 7, 9, 10, 21
6. Basic path 6: 1, 2, 3, 5, 7, 9, 11, 12, 21
7. Basic path 7: 1, 2, 3, 5, 7, 9, 11, 13, 14, 21
8. Basic path 8: 1, 2, 3, 5, 7, 9, 11, 13, 15, 16, 21
9. Basic path 8: 1, 2, 3, 5, 7, 9, 11, 13, 15, 17, 18, 21
10. Basic path 8: 1, 2, 3, 5, 7, 9, 11, 13, 15, 17, 19, 20, 21

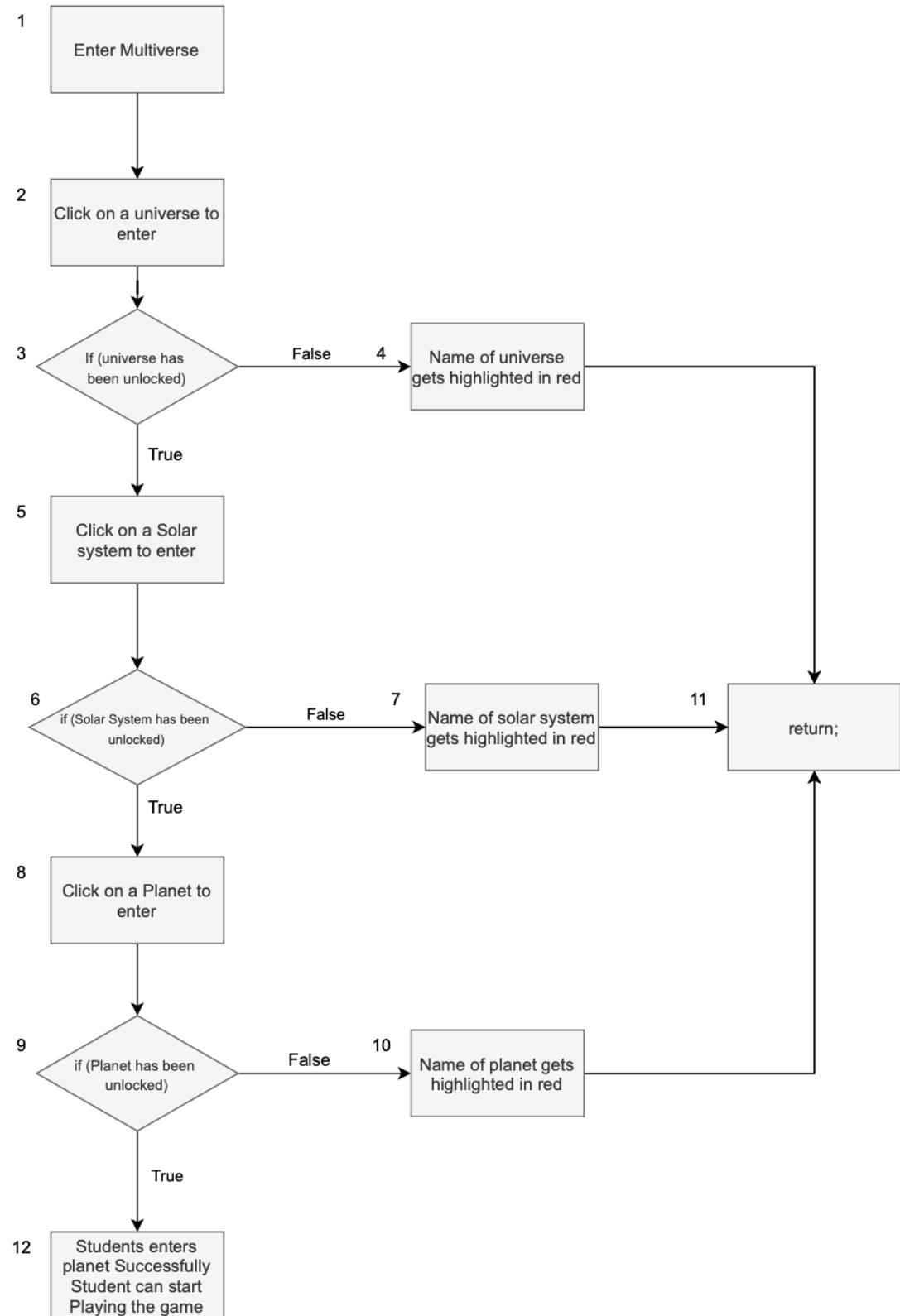
Execution Paths:

1. 1, 2, 3, 5, 7, 9, 11, 13, 15, 17, 19, 22
2. 1, 2, 3, 4, 21
3. 1, 2, 3, 5, 6, 21
4. 1, 2, 3, 5, 7, 8, 21
5. 1, 2, 3, 5, 7, 9, 10, 21
6. 1, 2, 3, 5, 7, 9, 11, 12, 21
7. 1, 2, 3, 5, 7, 9, 11, 13, 14, 21
8. 1, 2, 3, 5, 7, 9, 11, 13, 15, 16, 21
9. 1, 2, 3, 5, 7, 9, 11, 13, 15, 17, 18, 21
10. 1, 2, 3, 5, 7, 9, 11, 13, 15, 17, 19, 20, 21

Test Cases (I assume email= Aks001@e.ntu.edu.sg.com,password= U1927645F, first name: Aks; last name: Tayal; matric number:U1927645F; tutorial group: SCE5, account having email SU001@e.ntu.edu.sg already exists in the database)

1. email = “”; password: “U1927645F”; first name: “”; last name: “”; matric number: ””; tutorial group: “”; internet switched on
2. email = “Aks001@e.ntu.edu.sg.com” ; password: “”; first name: “”; last name: “”; matric number: ””; tutorial group: “”; internet switched on
3. email = “”; password: “”; first name: “Aks”; last name: “”; matric number: ””; tutorial group: “”; internet switched on
4. email = “”; password: “”; first name: “”; last name: “Tayal”; matric number: ””; tutorial group: “”; internet switched on
5. email = “”; password: “”; first name: “”; last name: “”; matric number: ”U1922309J”; tutorial group: “”; internet switched on
6. email = “”; password: “”; first name: “”; last name: “”; matric number: ””; tutorial group: “SCE5”; internet switched on
7. email = “Aks001@e.ntu.edu.sg” ; password: “U1927645F”; first name: “Aks”; last name: “Tayal”; matric number: ”U1927645F”; tutorial group: “SCE5”; internet switched off
8. email = “Aks001@gmail.com” ; password: “U1927645F”; first name: “Aks”; last name: “Tayal”; matric number: ”U1927645F”; tutorial group: “SCE5”; internet switched on
9. email = “SU001@e.ntu.edu.sg” ; password: “U1927645F”; first name: “Aks”; last name: “Tayal”; matric number: ”U1927645F”; tutorial group: “SCE5”; internet switched on
10. email = “Aks001@e.ntu.edu.sg” ; password: “U1927645F”; first name: “Aks”; last name: “Tayal”; matric number: ”U1927645F”; tutorial group: “SCE5”; internet switched on

3. Student Navigating Game



Cyclomatic Complexity:

$|decisionpoint| + 1 = 3 + 1 = \underline{4}$

Basic Paths:

1. Baseline path: 1, 2, 3, 5, 6,8,9,12
2. Basic path 2: 1, 2, 3, 4, 11
3. Basic path 3: 1, 2, 3, 5, 6,7,11
4. Basic path 4: 1, 2, 3, 5, 6,8,9,10,11

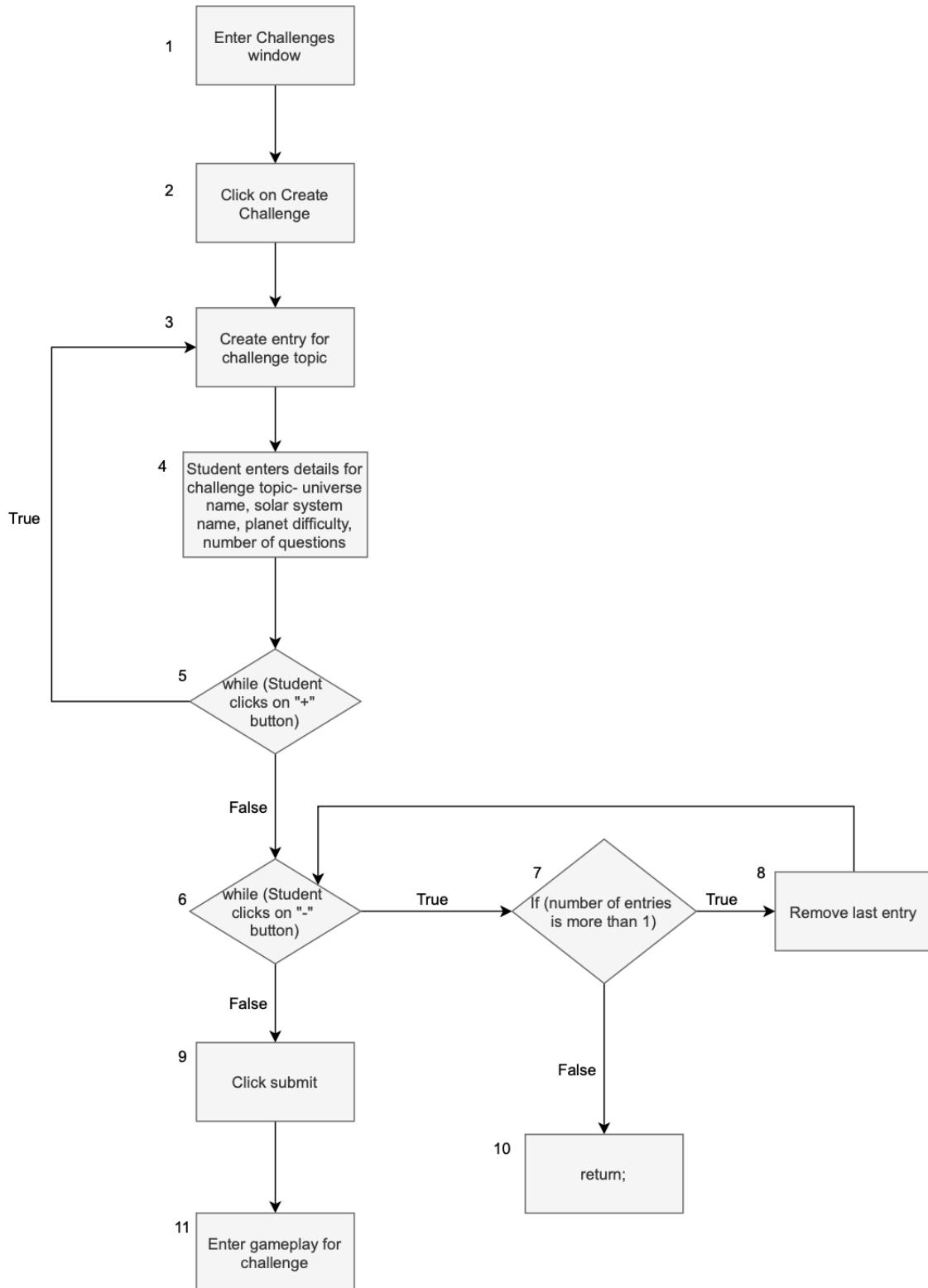
Execution Paths:

1. 1, 2, 3, 5, 6,8,9,12
2. 1, 2, 3, 4,11
3. 1, 2, 3, 5, 6,7,11
4. 1, 2, 3, 5, 6,8,9,10,11

Test Cases (I assume Universe 1 Unlocked ,Solar System 1 Unlocked,Planet 1 Unlocked)

1. Universe = “2” ; Solar System: “”; Planet: “ ”;
2. Universe = “1” ; Solar System: “2”; Planet: “ ”;
3. Universe = “1” ; Solar System: “1”; Planet: “2 ”;
4. Universe = “1” ; Solar System: “1”; Planet: “1 ”;

4. Student Creating Challenge



Cyclomatic Complexity:

$|decisionpoint| + 1 = 3 + 1 = 4$

Basic Paths:

1. Baseline path: 1, 2, 3, 4, 5, 6, 9, 11
2. Basic path 2: 1, 2, 3, 4, 5, 3, 4, 5, 6, 9, 11
3. Basic path 3: 1, 2, 3, 4, 5, 3, 4, 5, 6, 7, 8, 6, 9, 11
4. Basic path 4: 1, 2, 3, 4, 5, 6, 7, 10

Execution Paths:

1. 1, 2, 3, 4, 5, 6, 9, 11
2. 1, 2, 3, 4, 5, 3, 4, 5, 6, 9, 11
3. 1, 2, 3, 4, 5, 3, 4, 5, 6, 7, 8, 6, 9, 11
4. 1, 2, 3, 4, 5, 6, 7, 10

Test cases:

1. Universe = “2” ; Solar System: “3”; Difficulty: “Easy”; Number of questions: “5“;
2. Universe = “1” ; Solar System: “2”; Difficulty: “Medium”; Number of questions: “2“; Add topic;
Universe = “1” ; Solar System: “3”; Planet: “Medium”; Number of questions: “6“;
3. Universe = “1” ; Solar System: “2”; Difficulty: “Medium”; Number of questions: “2“; Add topic;
Universe = “1” ; Solar System: “3”; Planet: “Medium”; Number of questions: “6“; Remove topic;
4. Universe = “1” ; Solar System: “2”; Difficulty: “Medium”; Number of questions: “2“; Add topic;
Universe = “1” ; Solar System: “3”; Planet: “Medium”; Number of questions: “6“; Remove topic;
Remove topic;

8.1.2 Black Box Testing

1. Teacher Login/Student Login

a. Generic Cases

Test ID	Scenario	Expected Result	Actual Result
1	Login with a valid account username and password	The system displays the Home Page for the user to continue operation	The system displays the Home Page for the user to continue operation
2	Login without valid credentials	The system prompt the user to enter the credentials again	The system prompt the user to enter the credentials again
3	Login without filling up the valid fields	The System prompts the user to fill up the required fields for logging in	The System prompts the user to fill up the required fields for logging in

b. Specific Cases (Combination)

EmailID	Password	Expected Result	Actual Result
SU001@e.ntu.edu.sg	ntu123	Successful login	Successful login
ABC@e.ntu.edu.sg	ntu123	Invalid email	Invalid email
SU001@e.ntu.edu.sg	123	Invalid password	Invalid password
Empty("")	ntu123	Please fill in all required fields	Please fill in all required fields
SU001@e.ntu.edu.sg	Empty("")	Please fill in all required fields	Please fill in all required fields

2. Student Account Creation

Test ID	Scenario	Expected Result	Actual Result
1	Register with valid emailID and password	The System stores the student in the database	The System stores the student in the database
2	Register with incomplete fields	The System prompts the teacher to fill up all the fields for creating the Student Account	The System prompts the teacher to fill up all the fields for creating the Student Account
3.	Registering a student whose emailID already exists in the database	The System shows an error that the student already exists in the database	The System shows an error that the student already exists in the database

3. Teacher adding questions in Question Bank

1. Generic Cases

TestID	Scenario	Expected Result	Actual Result
1	The question body, correct option and incorrect options fields are filled	The system saves the question to the server and displays a success message	The system saves the question to the server and displays a success message
2	One or more fields of the question body, correct option and incorrect options are not filled	The system prompts the user to fill the required fields	The system prompts the user to fill the required fields
3	The added questionID already exists in the database	The system does not save it again and notifies the user that the added question	The system does not save it again and notifies the user that the added question

		already exists	already exists
--	--	----------------	----------------

4. Teacher adding Assignments

TestID	Scenario	Expected Result	Actual Result	P/F
1	All fields for adding Assignment have been added in correctly - such as time limit, deadline, points, assignment questions, assignment name and assignment ID	The System saves the assignment to the database and displays a success message	The System saves the assignment to the database and displays a success message	P
2	Teacher tries to create assignment with incomplete fields	The system prompts the user to enter all the fields	The System prompts the user to enter all the fields	P
3	The assignmentID provided already exist in the database	The system does not save it again and notifies the user that the added assignment already exists	The system does not save it again and notifies the user that the added assignment already exists	P

Unit Test Scripts - Game Application

Unit Test Scripts were written in C# and executed using the unity test runner. The scripts are divided such that each script is a test suite which tests a particular functionality that sits at the core of the application. There are multiple unit tests in each test suite script which cover different scenarios which tests if our component works with a variety of inputs.

Shown below are the results of the automated tests carried out for various units.



Unit Test Scripts - Backend Application (Student API)

Unit Test Scripts were written using the Mocha and chai framework in Node.js. The scripts are divided such that each script is a test suite which tests a particular URL service that sits at the core of the backend application. There are multiple unit tests in each test suite script which cover different scenarios which tests if our component works with a variety of request cases.

Shown below are the results of the automated tests carried out for various units.

```
> server@1.0.0 test C:\Users\praty\Desktop\SSAD\Softvengers--Teacher-Version\server
> mocha

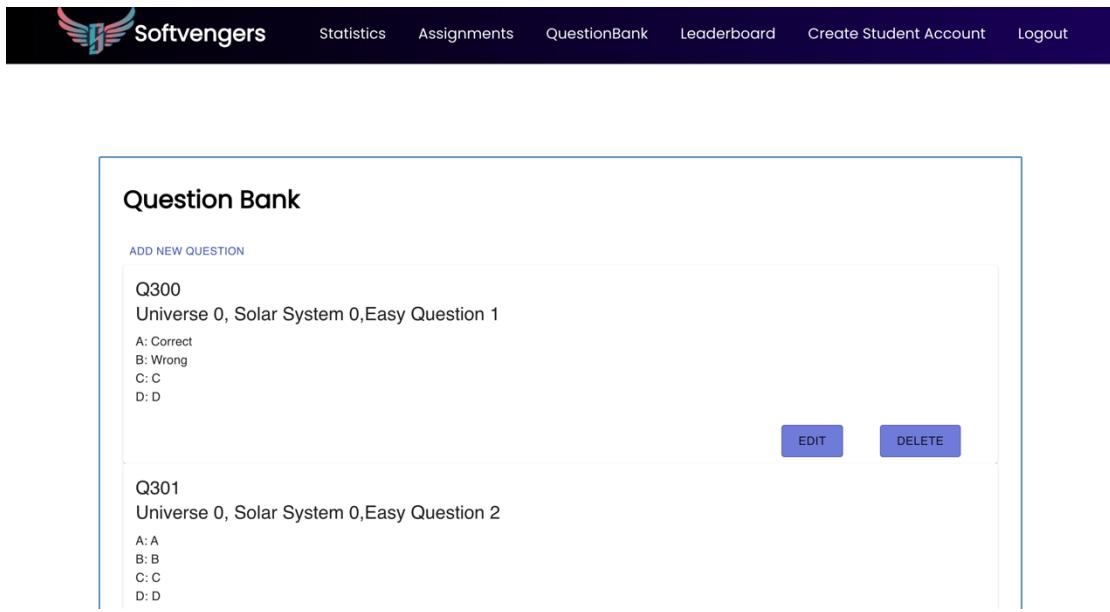
Student API
POST /student/login
  ✓ Successful Login
  ✓ Unsuccessful Login - Password Incorrect
  ✓ Unsuccessful Login - Username Incorrect
GET /student/details/getLeaderboard
  ✓ Sorted Leaderboard
  ✓ Bad request
GET /student/details/getAllStudents/:tut_grp
  ✓ Tutorial Group exists
  ✓ Tutorial Group does not exist
  ✓ Bad request
GET /student/challenge/getSentChallenges
  ✓ Student exists
  ✓ Student does not exist
  ✓ Bad request
GET /student/challenge/getReceivedChallenges
  ✓ Student exists
  ✓ Student does not exist
  ✓ Bad request
GET /student/details/getAllStudents
  ✓ Get all students
  ✓ Bad request
GET /student/posts
  ✓ Get all posts
  ✓ Bad request
GET /student/details/getStudent
  ✓ Student exists
  ✓ Student does not exist
GET /student/questions
  ✓ Topic exists
  ✓ Topic does not exist
JWT Authentication
  ✓ No header - Unauthorized
  ✓ Incorrect token

24 passing (145ms)
```

8.2. Integration Testing

8.2.1. Teacher's Web App with Backend

Black Box Testing

S.No	Scenario	Expected Result	Actual Result
1	Question Bank Page is integrated with backend - GET	Questions are dynamically fetched and displayed on screen.	Questions are dynamically fetched and displayed on screen.
			
<p>Figure 1 Question Bank Page getting data from backend</p>			
2	Question Bank Page is integrated with backend – POST + GET	On clicking add question, new questions are added to database and displayed on screen within 1 second	On clicking add question, new questions are added to database and displayed on screen within 1 second

Add Question

Universe*
1

Solar*
1

Planet*
1

QuestionID*
10

Planet Question*
testing

Select the Checkbox with the correct option:

- Option 1* a
- Option 2* b
- Option 3* c
- Option 4* d

CANCEL UPDATE QUIZ

Figure 2 Add New Question

Question Bank

ADD NEW QUESTION

Q10
Testing

A: 0
B: 1
C: 2
D: 3

Q300
Universe 0, Solar System 0, Easy Question 1

A: Correct
B: Wrong
C: C
D: D

EDIT DELETE

Figure 3 New Question integrated with backend and View Question

3	Assignment Page is integrated with backend - GET	Assignments are dynamically fetched and displayed on screen.	Assignments are dynamically fetched and displayed on screen.
---	--	--	--

The screenshot shows the 'Assignments' section of the Softvengers platform. At the top, there's a navigation bar with links for Statistics, Assignments, QuestionBank, Leaderboard, Create Student Account, and Logout. Below the navigation bar, there's a large blue header with the text 'Assignments'. Underneath this, three assignment entries are listed:

- Assignment 129078** by AKSHAT. Due date: 2021-04-15T00:00:00.000Z. Timelimit: 10. Buttons for Share on Twitter, Share on Reddit, and Statistics.
- Assignment 10259871** by Requirements Elicitation - Week 2. Due date: 2022-04-20T00:00:00.000Z. Timelimit: 15. Buttons for Share on Twitter, Share on Reddit, and Statistics.
- Assignment 10521188** by Saiteja. Due date: 2021-04-30T00:00:00.000Z. Buttons for Share on Twitter, Share on Reddit, and Statistics.

Figure 4 ViewAssignment component integrated with backend

4	Assignment Page is integrated with backend – POST + GET	On clicking add assignment, new assignments are added to database and displayed on screen within 1 second	On clicking add assignment, new assignments are added to database and displayed on screen within 1 second
---	---	---	---

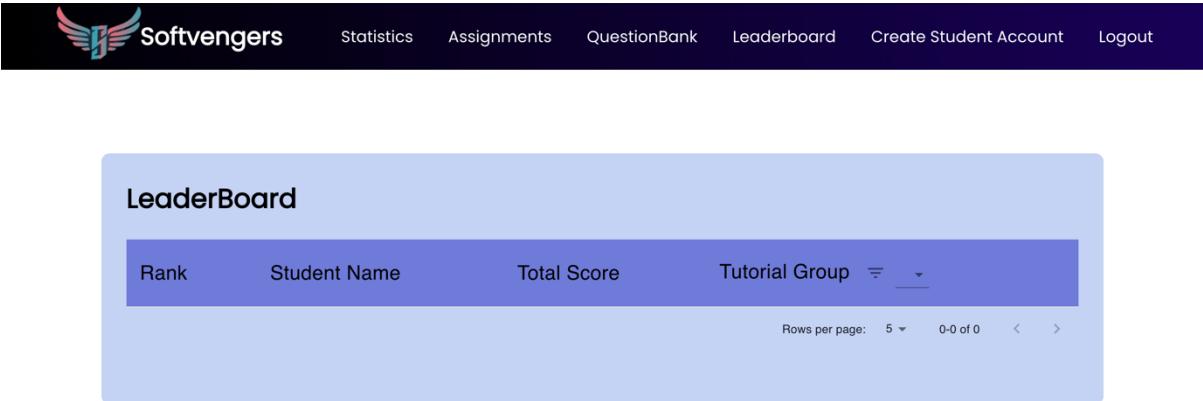
Add Assignment

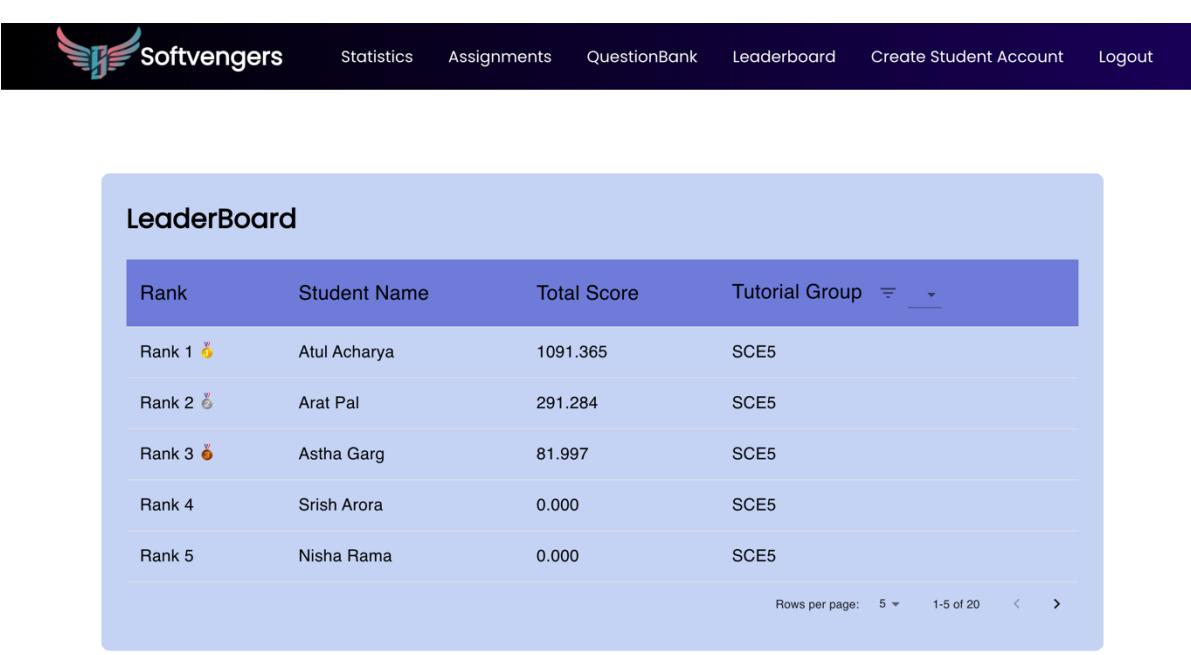
Assignment ID *	Assignment Name *							
Time limit*	<input type="text" value="dd/mm/yyyy"/> <input type="button" value="CALENDAR"/>							
Tutorial Group *								
Question ID	Points	Body	Correct Option	Wrong1	Wrong2	<input type="button" value="ADD MORE QUES"/>	<input type="button" value="CANCEL"/>	<input type="button" value="SUBMIT ASSIGNMENT"/>
wrong3								
Due : 2022-04-20T00:00:00.000Z Timelimit: 15								
 SHARE ON TWITTER SHARE ON REDDIT STATISTICS								
Assignment 10521188 Saiteja Due : 2021-04-30T00:00:00.000Z								

Figure 5 Add New Assignment

Assignments

Assignment 1	
Test Assign	
Due : 2021-04-20T00:00:00.000Z	
Timelimit: 10	
 SHARE ON TWITTER SHARE ON REDDIT STATISTICS	
Assignment 129078 AKSHAT Due : 2021-04-15T00:00:00.000Z Timelimit: 10	
 SHARE ON TWITTER SHARE ON REDDIT STATISTICS	
Assignment 10259871 Requirements Elicitation - Week 2 Due : 2022-04-20T00:00:00.000Z	

	<p><i>Figure 6 Newly added assignment integrated with backend and reflected in ViewAssignment</i></p>										
5	Leaderboard is integrated with backend - GET	The leaderboard displays ranking and names of students	The leaderboard displays ranking and names of students								
 <p>LeaderBoard</p> <table><thead><tr><th>Rank</th><th>Student Name</th><th>Total Score</th><th>Tutorial Group</th></tr></thead><tbody><tr><td></td><td></td><td></td><td>▼</td></tr></tbody></table> <p>Rows per page: 5 0-0 of 0 < ></p>				Rank	Student Name	Total Score	Tutorial Group				▼
Rank	Student Name	Total Score	Tutorial Group								
			▼								



The screenshot shows the Softvengers application interface. At the top, there is a dark header bar with the "Softvengers" logo on the left and navigation links for Statistics, Assignments, QuestionBank, Leaderboard, Create Student Account, and Logout on the right. Below the header is a light blue sidebar containing the text "LeaderBoard". The main content area features a table titled "LeaderBoard" with the following data:

Rank	Student Name	Total Score	Tutorial Group
Rank 1 🏆	Atul Acharya	1091.365	SCE5
Rank 2 🥈	Arat Pal	291.284	SCE5
Rank 3 🥉	Astha Garg	81.997	SCE5
Rank 4	Srish Arora	0.000	SCE5
Rank 5	Nisha Rama	0.000	SCE5

At the bottom of the table, there are pagination controls: "Rows per page: 5", "1-5 of 20", and navigation arrows.

Figure 8 ViewLeaderboard component integrated with backend

Testing Test Scripts

Scripts were written to mock the server (backend) and test its integration with the client (frontend). This prevents the server from processing unnecessary computations while allowing us to test our frontend at the same time.

Three scripts were written for the following purposes:

1. AddQuestion.test.js – To test the integration of Question Bank Page with backend (server)
2. AddAssignment.test.js – To test the integration of Assignment Page with backend (server)
3. getLeaderboard.test.js – To test the integration of Leaderboard with backend (server)

```
PASS  src/__test__/addQuestion.test.js
PASS  src/__test__/addAssignment.test.js
PASS  src/__test__/getLeaderboard.test.js (5.89 s)

Test Suites: 3 passed, 3 total
Tests:       3 passed, 3 total
Snapshots:   0 total
Time:        10.989 s
Ran all test suites related to changed files.

Watch Usage: Press w to show more.
```

Figure 9: Test Scripts' Results

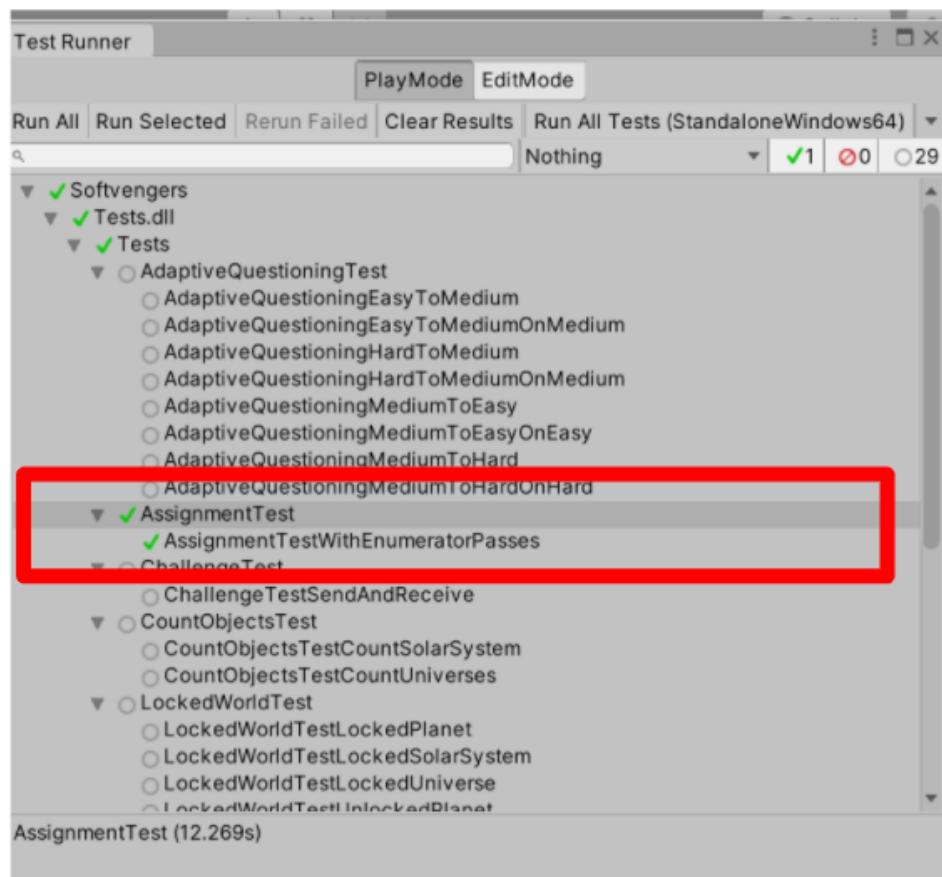
8.2.2 Teacher Create Assignment Shows up in Student Game

Black Box Testing

S.No	Scenario	Expected Result	Actual Result
1	POST Assignment in WebApp is integrated with GET Assignment in Game Application	The new assignment appears in the student's assignment list in the game	The new assignment appears in the student's assignment list in the game

Test Scripts

The above integration test was automated and tested using C# and Unity Test Runner. Below is the result of the integration test.



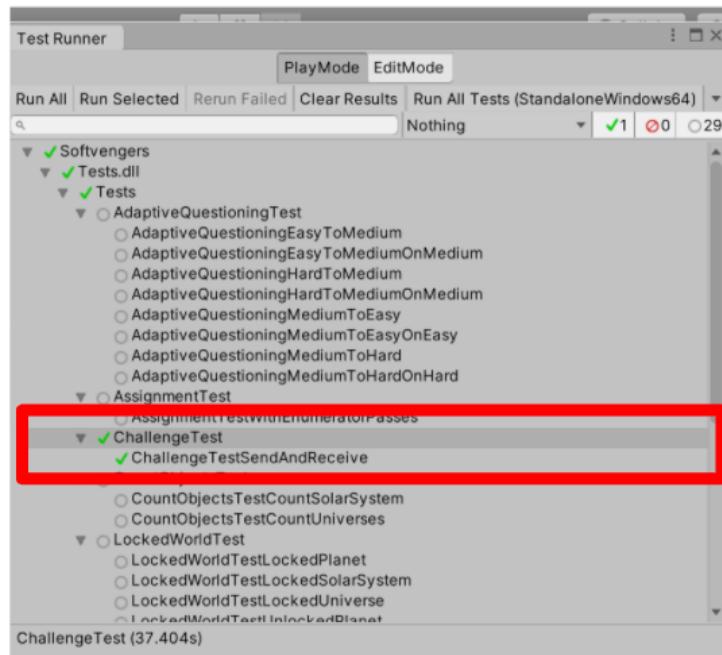
8.2.2 Student Creates and Sends Challenge

Black Box Testing

S.No	Scenario	Expected Result	Actual Result
1	A student creates (POST) a challenge, then plays the challenge. After completion of the challenge the student sends the challenge to another student. The challenge should appear (GET) for the other student.	The new challenge appears on the second student's challenge list	The new challenge appears on the second student's challenge list

Test Scripts

The above integration test was automated and tested using C# and Unity Test Runner. Below is the result of the integration test.



System Testing

System testing involves testing the entire software system based on functionality, usability, load and performance. Hence we carried out Load Testing on the Student's Game and the Teacher website, using Apache JMeter which is a load testing tool for analyzing and measuring the performance of services.

Load Testing – Student's game

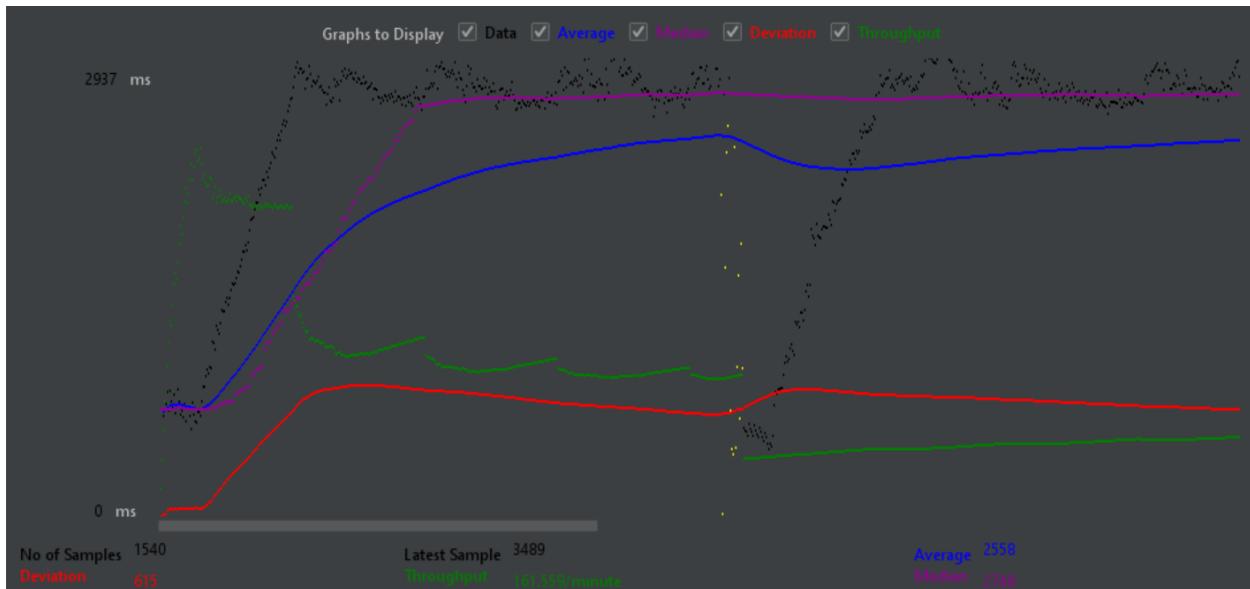
Thread Properties

Number of Threads (users):	100
Ramp-up period (seconds):	1
Loop Count:	<input type="checkbox"/> Infinite 1

Student login

Average time for 100 users: 2558 ms

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
Student Login	1540	2558	22	3499	615.59	0.97%	2.7/sec	1.92	0.67	731.0
TOTAL	1540	2558	22	3499	615.59	0.97%	2.7/sec	1.92	0.67	731.0



Get Questions

Average time for 100 users: 2461 ms

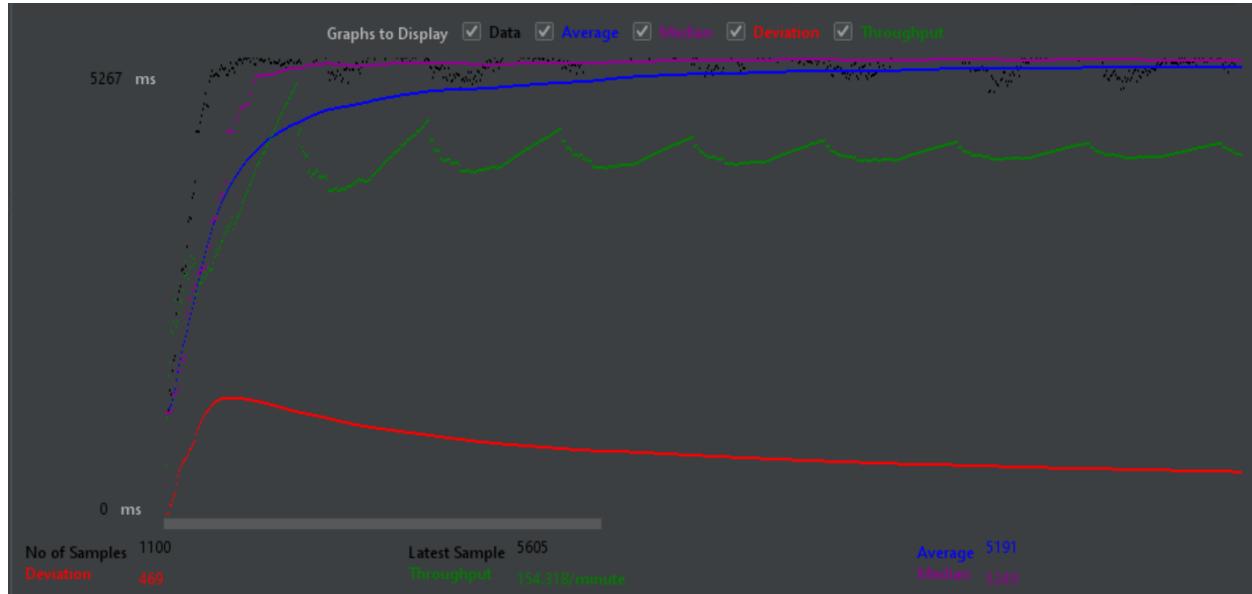
Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB...	Sent KB/sec	Avg. Bytes
Get Questions	1525	2461	68	3080	452.57	0.98%	2.7/sec	10.88	0.50	4193.7
TOTAL	1525	2461	68	3080	452.57	0.98%	2.7/sec	10.88	0.50	4193.7



getReceivedChallenges

Average time for 100 users: 5131 ms

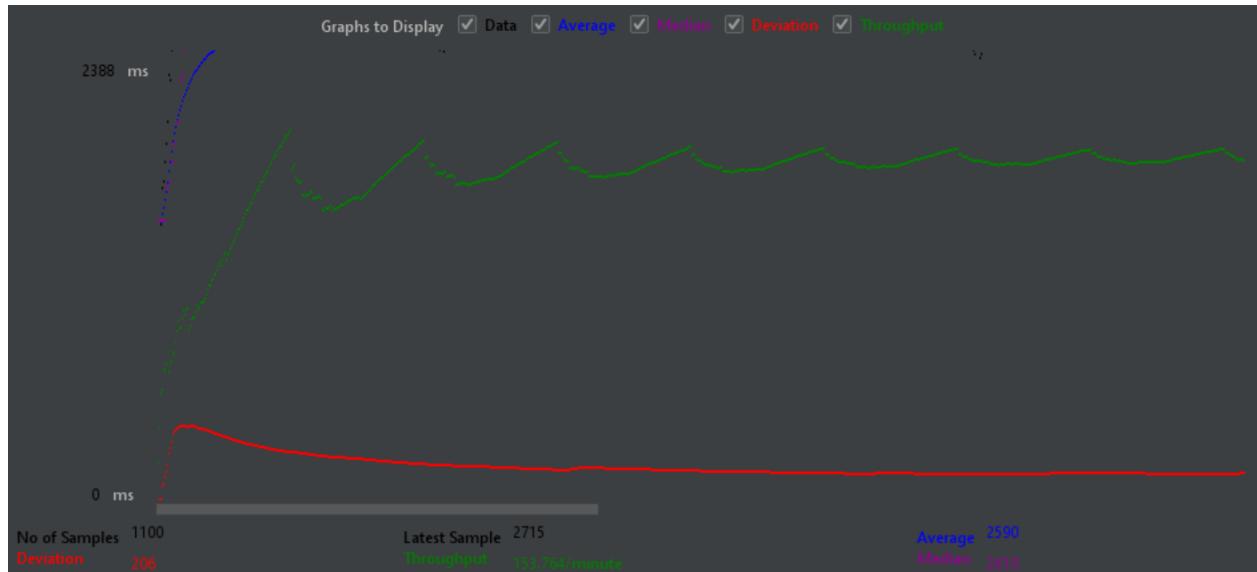
Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB...	Sent KB/sec	Avg. Bytes
get Recieve...	1510	5131	218	6130	600.12	0.66%	2.6/sec	9.29	0.57	3648.7
TOTAL	1510	5131	218	6130	600.12	0.66%	2.6/sec	9.29	0.57	3648.7



getSentChallenges

Average time for 100 users: 2558ms

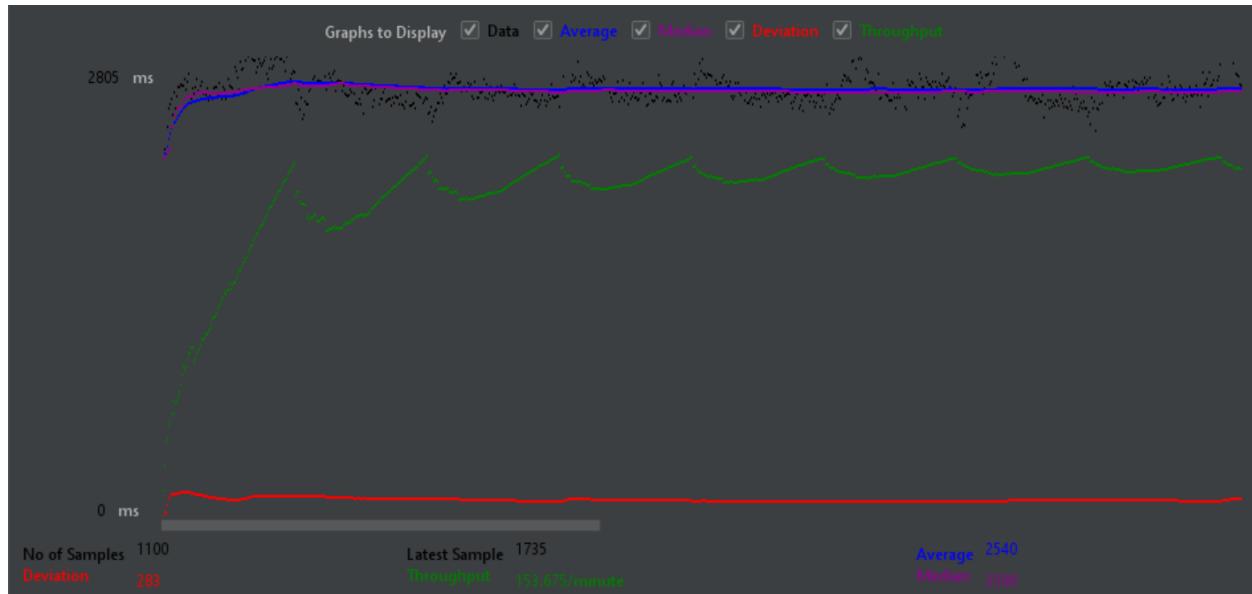
Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB...	Sent KB/sec	Avg. Bytes
get Sent Ch...	1500	2558	147	3104	286.01	2.40%	2.6/sec	12.83	0.54	5086.8
TOTAL	1500	2558	147	3104	286.01	2.40%	2.6/sec	12.83	0.54	5086.8



getLeaderboard

Average time for 100 users: 2535ms

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
get Leaderb...	1464	2535	9	3094	317.74	1.64%	2.5/sec	2.84	0.52	1155.3
TOTAL	1464	2535	9	3094	317.74	1.64%	2.5/sec	2.84	0.52	1155.3



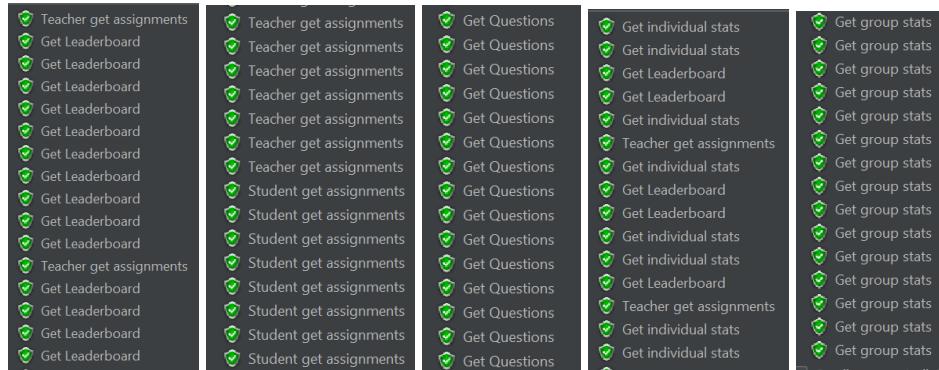
Load Testing – Teacher’s website

Case 1

50 users, 1 request per user, to the following routes:

- Get questions
 - Get assignments- students
 - Get assignments- teachers
 - Get leaderboard
 - Get individual statistics
 - Get group statistics

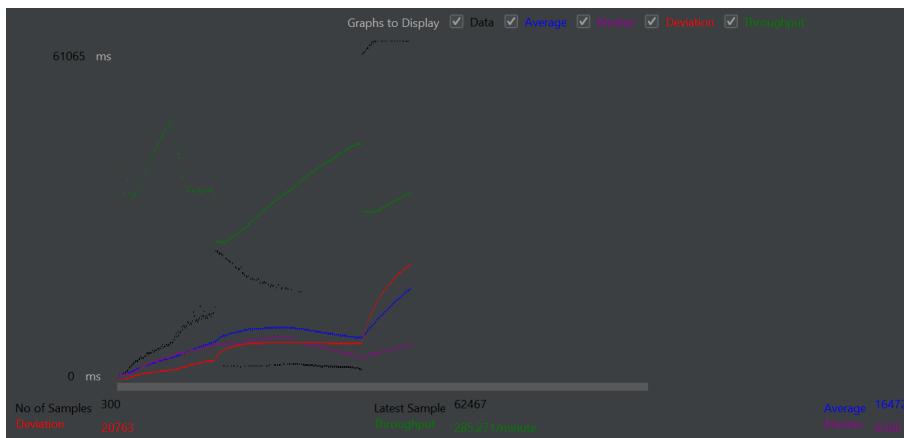
Below are some screenshots of the [Results tree](#):-



Here all the routes pass the test.

Note: Get group statistics is a route with a small error rate, due to the fact that it involves heavy processing, it sometimes takes too long to send a response.

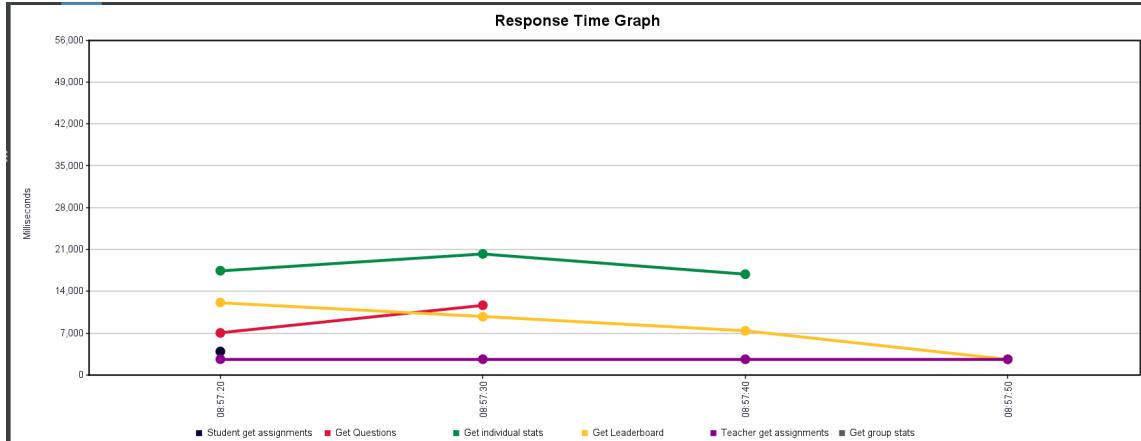
Here are the Graph Results:-



Here is the **summary report**:-

Label ↑	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
Get group stats	750	64865	58755	120038	14756.48	6.67%	1.8/sec	3.22	0.23	1881.0
Get individual st...	900	21314	15880	35773	4870.12	0.00%	2.3/sec	4.61	0.35	2053.0
Get Leaderboard	900	3002	2399	4958	689.16	0.00%	2.5/sec	2.25	0.33	919.0
Get Questions	900	10885	4783	19455	3062.99	0.00%	2.3/sec	199.25	0.29	87197.0
Student get assi...	900	4807	807	11008	2545.51	0.00%	2.4/sec	5.45	0.40	2318.0
Teacher get assi...	900	2767	1932	4640	660.04	0.00%	2.5/sec	1.44	0.35	584.0
TOTAL	5250	16599	807	120038	21646.74	0.95%	12.3/sec	194.52	1.72	16223.7

Here are the **response time graphs** of the various routes:-



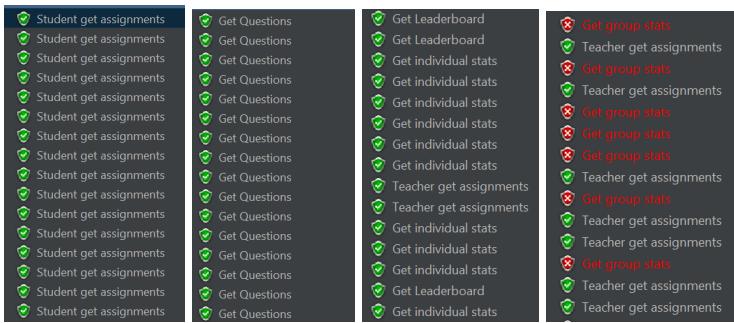
Case 2

100 users, 2 requests per user, to the following routes:

- Get questions
- Get assignments- students
- Get assignments- teachers

- Get leaderboard
- Get individual statistics
- Get group statistics

Below are some screenshots of the **Results tree**:-



All routes except Get group statistics passed the test. For the route Get group statistics, the server was unable to provide a response in time, because of the high load.

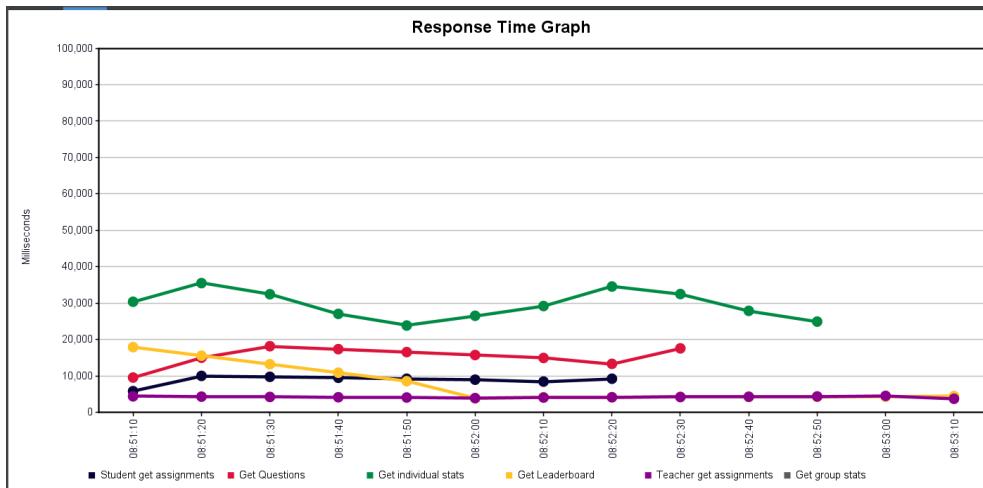
Here are the **Graph Results**:-



Here is the **summary report**:-

Label ↑	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB...	Sent KB/sec
Get group st...	50	120008	120001	120038	7.23	100.00%	24.6/min	0.84	0.00
Get individu...	200	29130	23781	35773	3106.23	0.00%	1.9/sec	3.72	0.28
Get Leaderb...	200	4233	3580	4958	299.34	0.00%	2.6/sec	2.38	0.35
Get Questions	200	14718	8827	19455	2582.00	0.00%	2.2/sec	185.68	0.27
Student get ...	200	7842	1266	11008	2221.35	0.00%	2.7/sec	6.12	0.44
Teacher get ...	200	3889	2317	4640	410.52	0.00%	2.7/sec	1.54	0.37
TOTAL	1050	17107	1266	120038	24862.45	4.76%	8.5/sec	147.78	1.15

Here are the **response time graphs** of the various routes:-



Case 3

The following routes were tested:-

- Get questions
 - Get assignments- students
 - Get assignments- teachers
 - Get leaderboard
 - Get individual statistics
 - Get group statistics

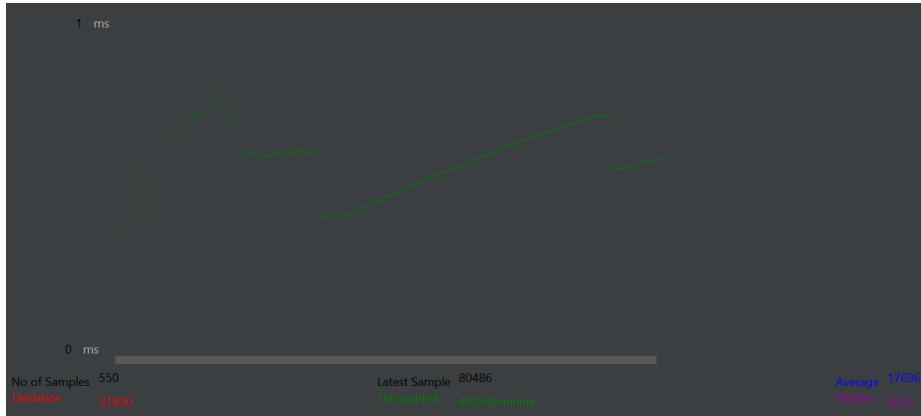
All routes except Get group statistics had 100 users, with 1 request per user.

Get group statistics had 50 users with 1 request per user.

Below are some screenshots of the **Results tree**:

All the routes pass all the tests.

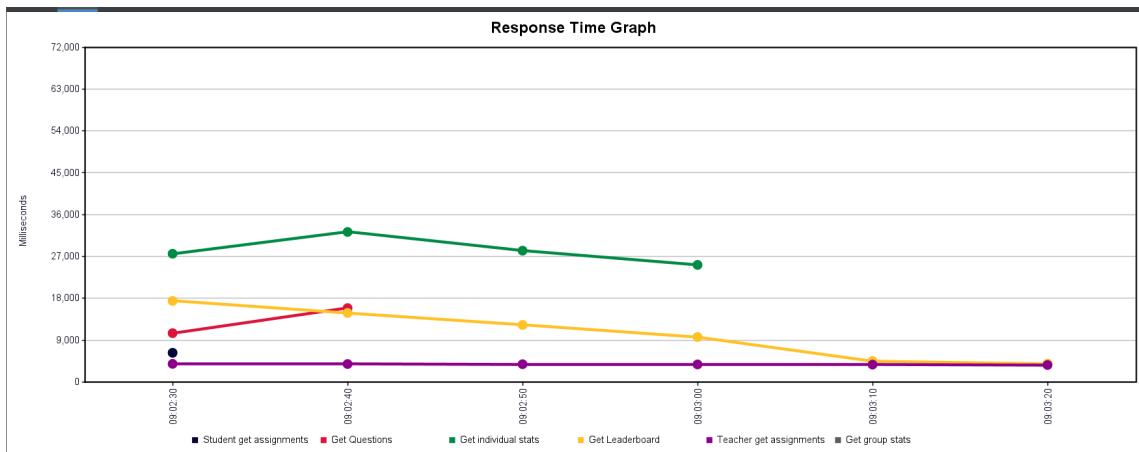
Here are the **Graph Results**:-



Here is the **summary report**:-

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
Student get assi...	100	6293	1364	9966	2519.87	0.00%	8.5/sec	19.13	1.39	2318.0
Get Questions	100	14204	8581	17808	2255.59	0.00%	3.5/sec	301.66	0.45	87197.0
Get individual st...	100	28772	24806	33517	2497.88	0.00%	2.2/sec	4.51	0.34	2053.0
Get Leaderboard	100	4079	3483	4651	305.41	0.00%	6.9/sec	6.15	0.91	919.0
Teacher get assi...	100	3668	3322	4323	264.37	0.00%	7.4/sec	4.20	1.01	584.0
Get group stats	50	80626	79529	81730	413.14	0.00%	36.4/min	1.11	0.08	1866.0
TOTAL	550	17696	1364	81730	21930.32	0.00%	6.7/sec	111.41	0.95	17091.6

Here are the **response time graphs** of the various routes:-



Security test – Backend application

1. The database records the password of the user which needs to be encrypted and stored so that it cannot be made known to the database administrator.

The following image shows the **encryption of password** in the cloud database

```
_id:ObjectId("6966f1205868442768808c31")
volume:5
avatar:0
totalScore:201.2841812875122
tutGrp:"SCE5"
scoreHistory:Array
__v:0

> _id:ObjectId("6966f14d5868442768808c82")
volume:5
avatar:1
totalScore:1191.8658125489543
password:"$20505051$ed029c7aek2h0Yo.ThA4H#716115adktbr5DqyB9Nm0X6"

> _id:ObjectId("6966f1965868442768808cd")
volume:5
avatar:0
totalScore:1191.8658125489543
password:"$20505051$ed029c7aek2h0Yo.ThA4H#716115adktbr5DqyB9Nm0X6"
scoreHistory:Array
__v:0
```

2. The backend application being a standalone application should not allow access to the main app logic URLs unless the user is logged in. A student account should be allowed access to the student specific URLs and a teacher account can only access the teacher specific URLs on the backend application server. This is ensured by allowing JWT authentication in the app logic and data access URLs after student or teacher login. This ensures that private data of a student or teacher is not leaked to any unauthorized account.

The following image shows how **unauthorized access to student specific URL** is handled:

http://localhost:5000/student/challenge/getSentChallenges?emailID=ARATRIKA001@e.ntu.edu.sg

GET http://localhost:5000/student/challenge/getSentChallenges?emailID=ARATRIKA001@e.ntu.edu.sg

Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Type Inherit auth fr...

The authorization header will be automatically generated when you send the request.
Learn more about authorization ↗

This request is not inheriting any authorization helper at the moment. Save it in a collection to use the parent's authorization helper.

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize Text

Status: 401 Unauthorized Time: 8 ms Size: 259 B Save Response

1 Unauthorized

Appendix A: Glossary

Data Dictionary

Term	Definition
Multiverse	A collection of 6 universes representing the field of software engineering.
Universe	A one-to-one mapping to each stage in the Software Development Life Cycle: Requirements, Architecture, Design, Implementation, Testing and Maintenance. It is a collection of solar systems.
Solar System	A one-to-one mapping to various sub-topics under a particular stage of the Software Development Life Cycle. It is a collection of planets.
Planet	A one-to-one mapping to various difficulty levels for a particular subtopic. It can be Novice, Intermediate or Advanced. Each planet has a proportionate mix of questions of varying difficulty level based on the planet's overall difficulty level.
Bronze Badge	The badge awarded for getting 50-70% questions correct in an attempt for a particular planet.
Silver Badge	The badge awarded for getting 70-99% questions correct in an attempt for a particular planet.
Gold Badge	The badge awarded for passing a topic by scoring 100% of questions correct in an attempt for a particular planet.
Infinity Stone Badge	Accomplishment that can be earned if a student has achieved 100% score in a universe.
Infinity Gauntlet Badge	Highest accomplishment in the game where a student has earned an Infinity Stone Badge for each and every universe.
Avatar	An icon or figure representing a student in the game.
Streak	The number of questions that the student has answered correctly or incorrectly in a row while rescuing a planet.
Question difficulty	The difficulty level of a question (Novice, Intermediate, Advanced) assigned by the teacher while creating the question and adding it to the database.
Planet difficulty	The difficulty level of the entire attempt when a student chooses a particular planet. Adaptive questioning decides the difficulty level of the questions displayed in the attempt. The question difficulty of majority of the questions will match the planet difficulty. Depending on the real time attempt of the student, questions belonging to

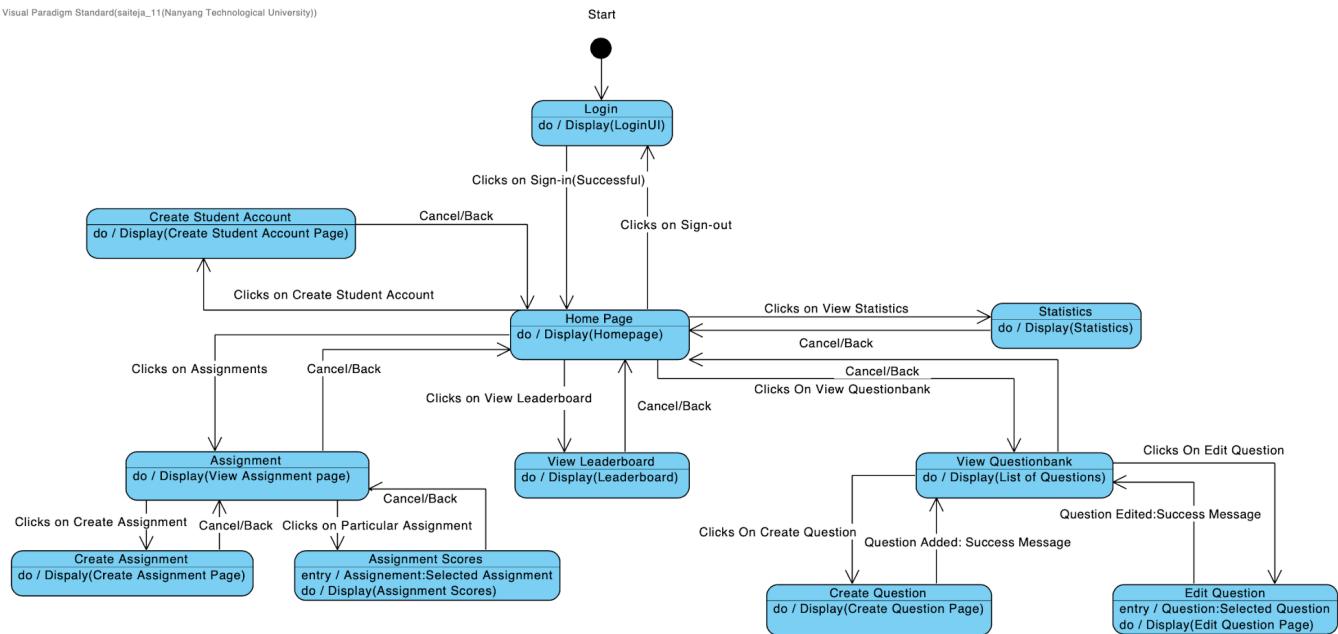
	other question difficulty levels can be displayed for a planet with a particular planet difficulty level.
NTU email address	Email address assigned to all staff and students of Nanyang Technological University.
Matriculation Number	Unique identification number assigned to all students of Nanyang Technological University. It is alphanumeric in nature with 9 characters.
Tutorial Group	Student groups of size 20-30 taught by a teacher at NTU.
Assignment	A multiple choice questions test set by the teacher for the students in his/her tutorial group with a time limit and a deadline.
Challenge	A game mode where the student challenges his friends to outperform his/her score..
Leaderboard	A scoreboard showing the names and points of students.

Appendix B: Analysis Models

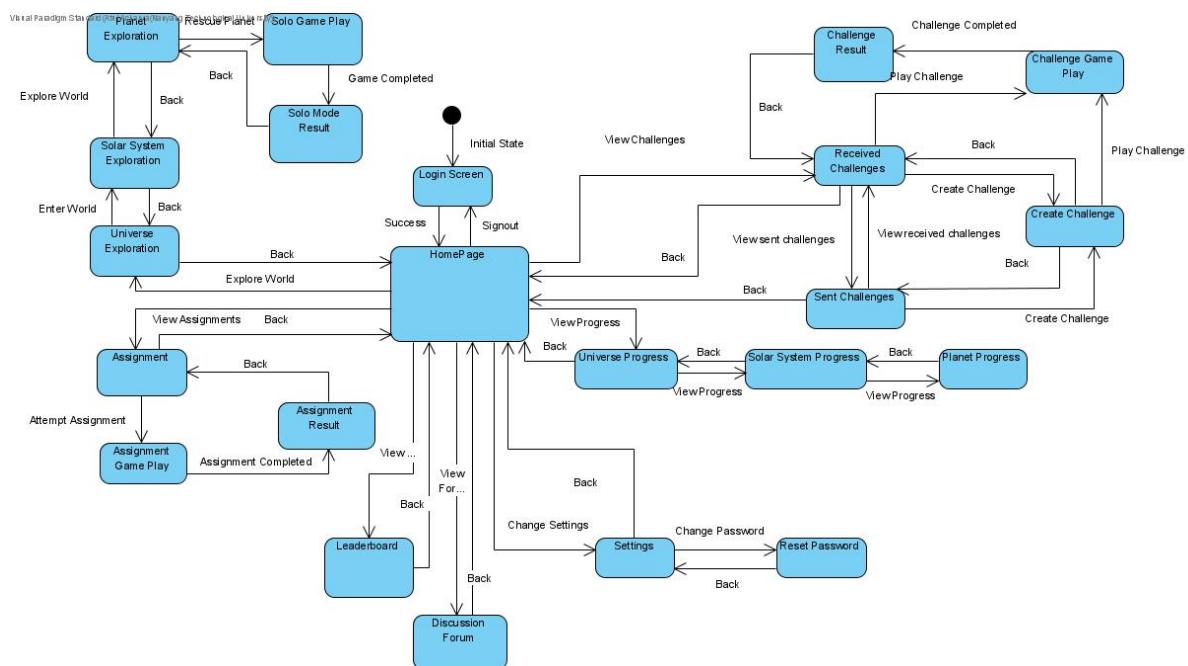
1. Dialog Map

1.1. Teacher Website

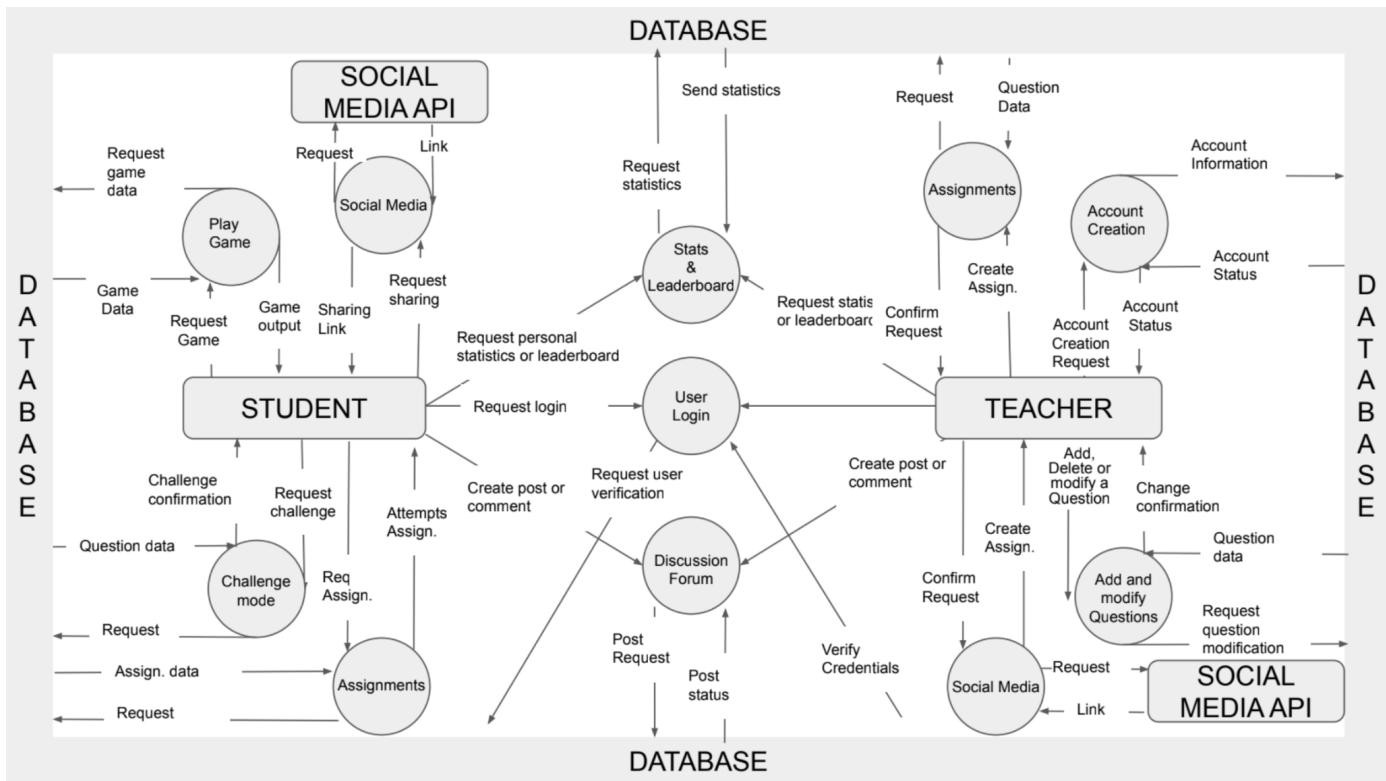
Visual Paradigm Standard(saiteja_11(Nanyang Technological University))



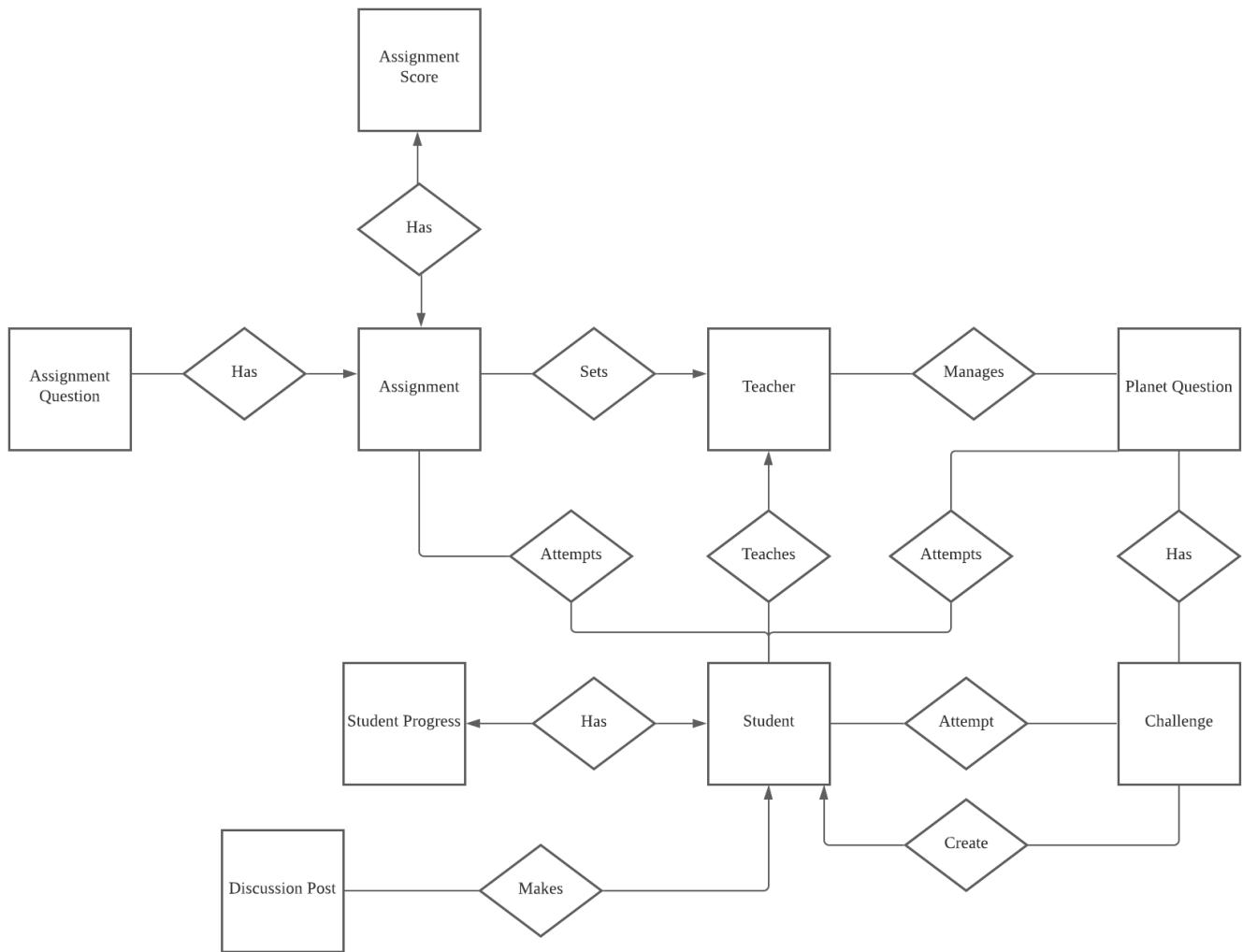
1.2. Dialog Map Student Game



2. Data Flow Diagram



3. Entity Relationship Diagram



4. Decision Tables

Navigate Universe

Condition	Requirement Number				
	1	2	3	4	5
Logged In	F	T	T	T	T
Universe is Unlocked	-	F	T	T	T
Solar system is unlocked	-	-	F	T	T
Planet is unlocked	-	-	-	F	T
Action					
Start Rescue Planet					X
Don't start Rescue Planet	X	X	X	X	

Create Assignments

Condition	Requirement Number					
	1	2	3	4	5	
						6

Logged In	F	T	T	T	T	T
Number of questions created >0	-	F	T	T	T	T
Number of students selected>0	-	-	F	T	T	T
Time-limit is set	-	-	-	F	T	T
Deadline is set	-	-	-	-	F	T
Action						
Send Assignment						X
Don't send assignment	X	X	X	X	X	

Creating a challenge

Condition	Requirement Number					
	1	2	3	4	5	6
Logged In	F	T	T	T	T	T
At Least one	-	F	T	T	T	T

universe selected						
At Least one section selected	-	-	F	T	T	T
Number of questions selected	-	-	-	F	T	T
Difficulty level selected for all questions	-	-	-	-	F	T
Action						
Create challenge						X
Don't create challenge	X	X	X	X	X	

Sending a challenge

Condition	Requirement Number		
	1	2	3
Created and played the challenge	F	T	T
Selected at	-	F	T

least one challengers			
Action			
Send challenge			X
Don't send challenge	X	X	

Scoring

Condition	Requirement Number			
	1	2	3	4
Logged In	F	T	T	T
Chosen answer is correct	-	F	T	T
Answer submitted within time limit	-	-	F	T
Action				
Bonus awarded				X
No bonus awarded	X	X	X	

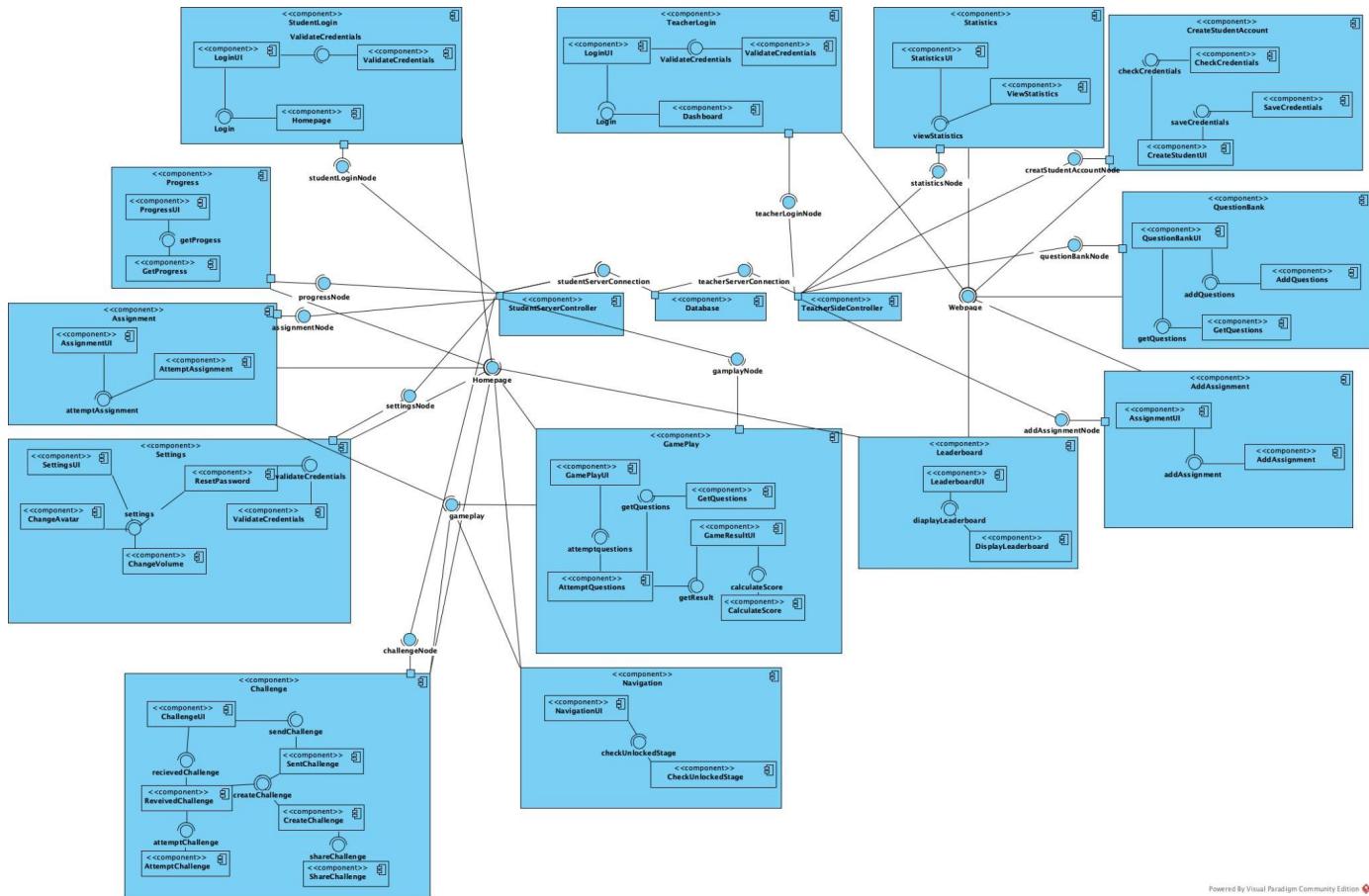
5. Crudl Matrix

Use Case\ Entity	Teach er	Studen t	Questi on	Planet Question	Assignment Question	Assignment	StudentPro gress	Challenge	Posts
Login	R	R							
Create Student Account		C							
View Question Bank			R, L	R, L					
Create Question			C	C					
Edit Question			U, D	U, D					
View Group Statistics	R	R, L					R, L		
View Individual Statistics	R	R, L					R, L		
Create Assignment		R,L	C, U, D		C, U, D	C			
View Assignments					R, L	R, L			
View		L	R, L		R, L	R, L			

Assignment Performance								
View Leaderboard		R						
Change Password		U						
Change avatar		U						
Navigate multiverse		R						
Attempt Assignment			R	R	R, L, U			
Challenge Peers on Social			R	R			C, U	
Attempt Challenge			R	R			R, L, U	
Rescue Planet			R, L	R, L		C, U		
View Leaderboard		R, L				R		
View Discussion Forum								R, L

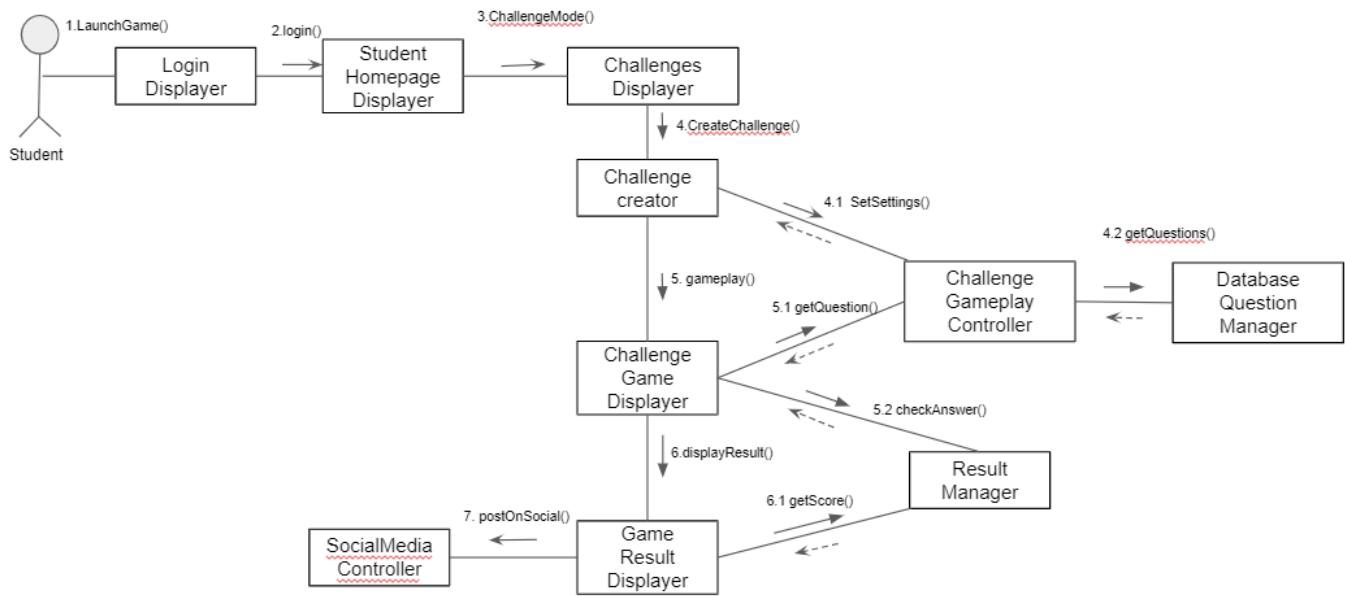
Write posts									C
-------------	--	--	--	--	--	--	--	--	---

6. Component Diagram

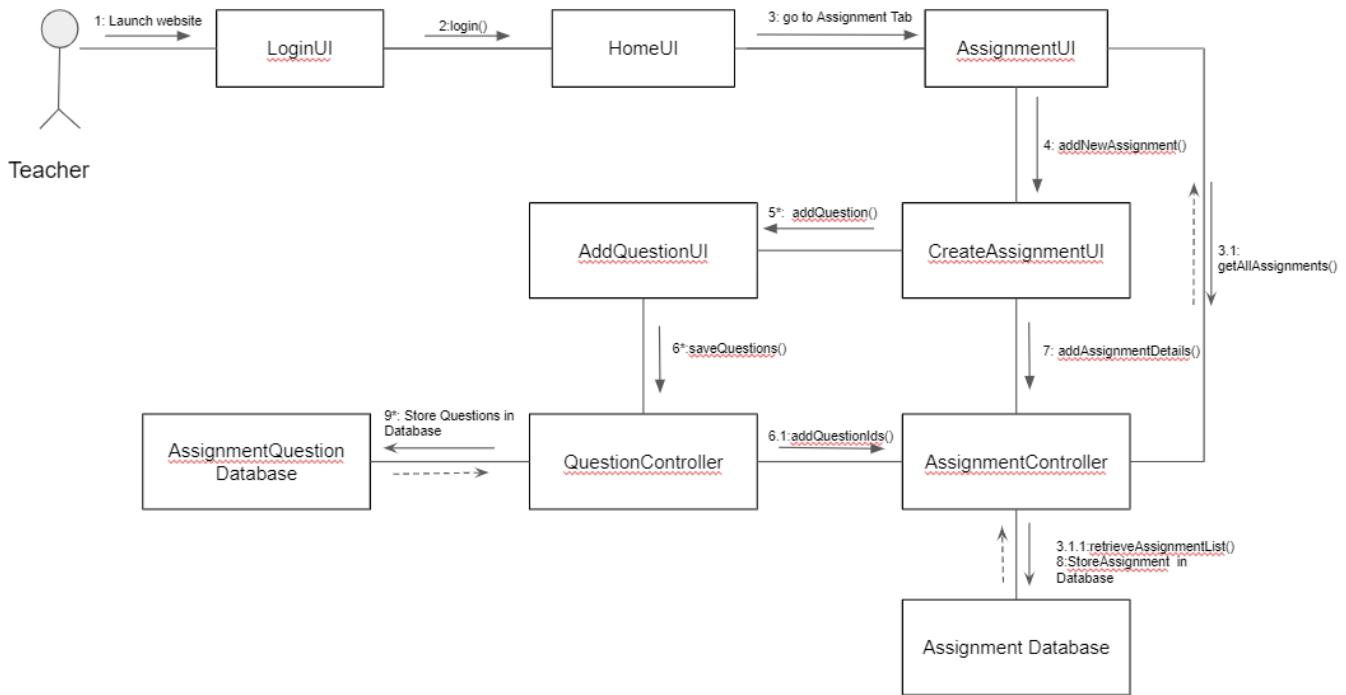


7. Communication Diagram

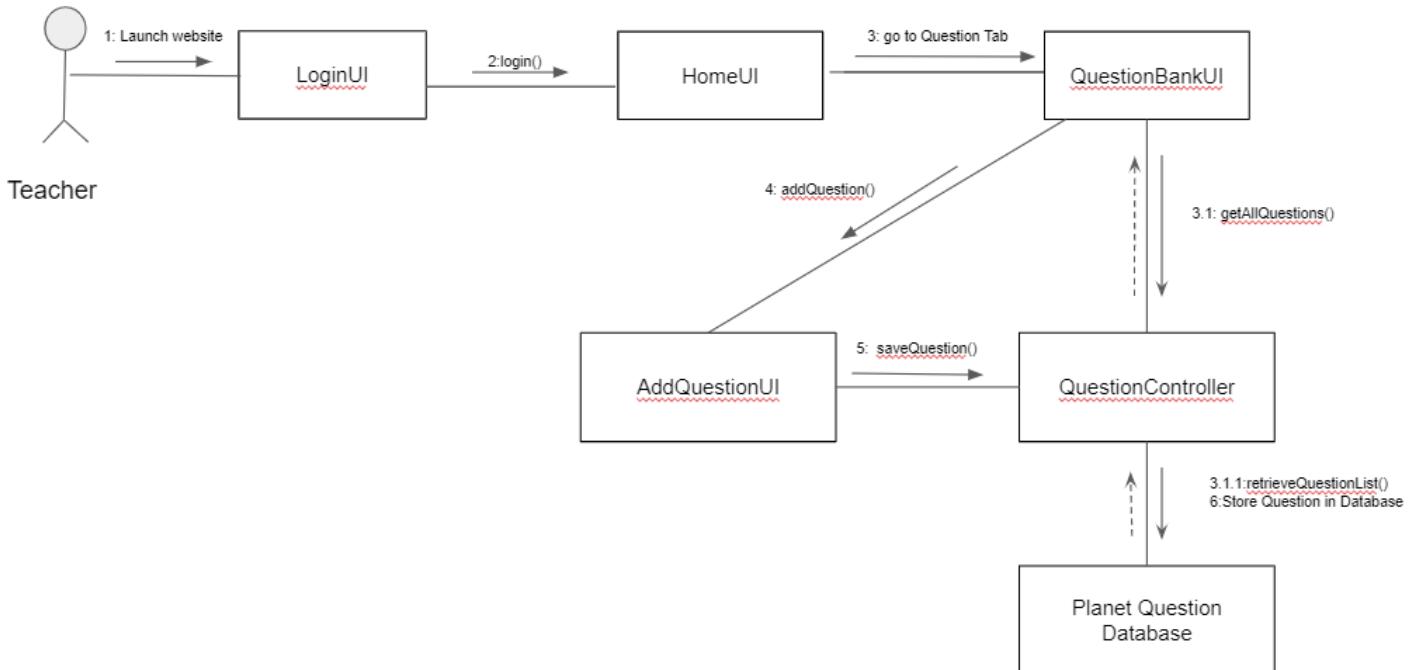
7.1. Create Challenge



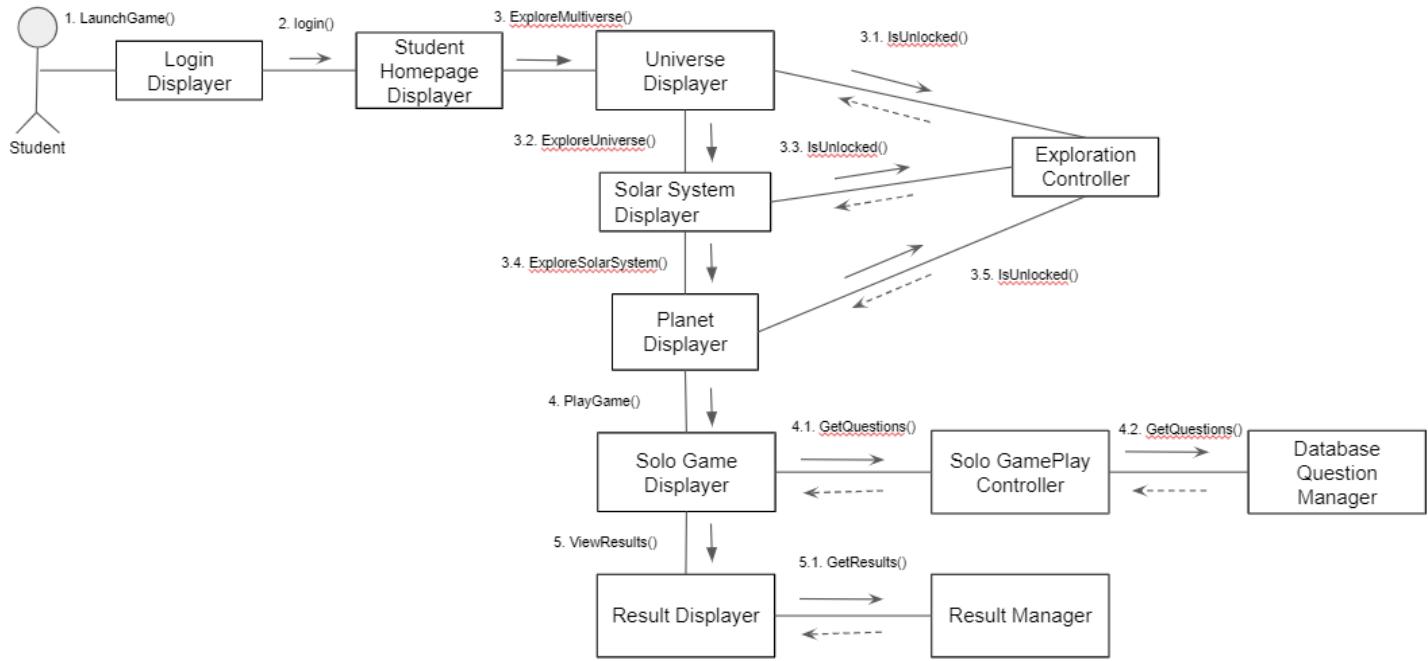
7.2. Add Assignment Questions



7.3 Add Question in Question Bank



7.4 Play Solo Mode



Source: http://www.frontiernet.net/~kwiegers/process_assets/srs_template.doc