Tests Documentation

for

Edumon

Version 1.0 approve

Prepared by Group1:

Beh Chee Kwang Nicholas (U1824125F)
Chen Xueyao (U1922640G)
Chong Jing Hong (U1922300B)
Goh Hong Xiang, Bryan (U1920609E)
Huang Shaohang (U1921245D)
Lim Jun Wei (U1922756C)
Muhammad Al-Muhazerin (U1921191L)
Quah Dian Wei (U1920106C)
Shauna Tan Li-Ting (U1840754E)
Swa Ju Xiang (U1822040G)

31/10/2021

Table of Contents

Revision History

Name	Date	Reason for change	Version
Everyone	23/10/2021	Initial draft	0.1
Everyone	31/10/2021	For submission	1.0

1. Introduction

The product tested in this document is Edumon, an educational game application intended and designed for students to learn Software Engineering concepts through the gameplay as the delivery method.

In order to amplify the effects of the game, automated tools like test runners were used throughout the process. Four types of tests were conducted, namely unit tests, integration tests, functional/system tests and performance/load tests. More specifically, our team decided on conducting the integration tests as API tests and system tests as UI tests. API tests were determined to be a good form of integration tests as most of the business logic of the application lives in the backend server. UI tests are appropriate system tests as the game interfaces determine how a user interacts with the entire system. Last but not least, load testing is used to determine how robust and performant the game is under heavy load from the users.

For each respective type of tests, a structured plan was followed to maximize test coverage and comprehensiveness. This includes important factors such as test objective, environment, approach and strategy.

2. Test Cases for System Level Functional Tests

Scripting test cases for system level functional tests is highly important as it will ensure that our application is working smoothly and the testers interacting with the application would understand what to expect. Another intended effect of the test cases is for the testers to report any bugs found and these bugs will be fixed immediately.

As the creators / developers of the application, only would we know how the flow and functions of the application would work. The test cases as depicted below are to give absolute clarity on what the intended audience would be anticipating as they dive deep into the application. Flows and expected results are to be adhered to and any bugs would not be tolerated and fixed immediately.

At the end of the day the goal of the test cases for system level functional tests is to provide users with the best user interface and experience as they use the application.

3. Integration Testing

Edumon has adopted integration testing, which is defined as a type of testing whereby software modules are integrated logically and tested as a group to ensure that the interaction between different modules and components have no defects prior to the completion of each module. This allows the team to identify and resolve defects or bugs earlier in the Software Development Life Cycle (SDLC). By doing so, less time, effort and budget will be wasted as mistakes and defects can be corrected earlier and there are less sunk costs if a module or component has to be changed substantially to ensure proper integration.

3.1 Approach and Methodology of Integration Tests

Our team has taken the **Hybrid Incremental Testing** approach, which is a combination of the Top-Down and Bottom-Up Incremental Testing approaches. The testing is done by integrating two or more modules that are logically related to each other and tested for whether there is proper, functional communication between the 2 modules. The other related modules are integrated incrementally and the process continues until all logically related modules are integrated and tested successfully.

The benefits of this approach is that fault localization is easier and early prototypes can be created. Major design flaws or defects can be found and fixed first, allowing for a more efficient development process.

As Edumon is a Unity-based game, most of the integration test cases involve testing the link between different screens. Multiple placeholder screens, panels, canvases and buttons were created during the testing process to serve as dummies.

3.2 Test Cases

The following are the integration test cases, grouped according to their functionality.

3.2.1 Authentication Functionalities

Test Case ID	Test Case Objective	Test Case Description	Expected Result	Pass /Fail
1	Check the interface link between the Login scene and the User Account scene.	Enter login credentials and click on the Login button	Students are logged in and redirected to the Student Home Page. Teacher is logged in and redirected to the Teacher Home Page.	Pass
2	Check the interface link between Register scene and Main Menu.	Enter the credentials and click on the Register button.	User account information is added to the database.	Pass
3	Check the interface link between Student Home Page and Main Menu scene.	Click on the logout button from the Student Home Page menu.	Students are redirected to the Edumon Main Menu.	Pass
4	Check the interface link between the Teacher HomePage scene and Main Menu scene.	Click on the logout button from the Teacher Homepage menu.	Teacher is redirected to the Edumon Main Menu scene.	Pass

3.2.2 Student Functionalities

Test Case ID	Test Case Objective	Test Case Description	Expected Result	Pass /Fail
1	Check the interface link between Student Home Page and Student Profile Page.	Click on the “Profile Page” button from the Student Home Page menu.	Students are redirected to their personal profile page. Placeholder text and buttons are displayed.	Pass
2	Check the interface link between Student Home Page and World module.	Click on the Enter World button from the Student Home Page menu.	Student is redirected to a world scene.	Pass
3	Check the interface link between Student Home Page and View Leaderboard module.	Click on the View Leaderboard button from the Student Home Page menu.	Students are redirected to the leaderboard scene containing the scoreboard.	Pass
4	Check the interface link between Student Home Page and Assignment module.	Click on the Attempt Assignment button from the Student Home Page menu.	Student is redirected to the Attempt Assignment scene.	Pass
5	Check the interface link between Attempt Assignment scene and Assignment Questions scene.	Click on the Attempt button from the Attempt Assignment scene.	Students are redirected to the Assignment Questions scene with placeholder text and buttons.	Pass
6	Check the interface link between the map scene and its corresponding gym scene.	Player enters the portal element.	Scene changes to the corresponding gym scene. If the gym scene is not yet implemented, use dummy (blank) scene or test stub Console.WriteLine("Event triggered."); To indicate the event that triggers the change in scene.	Pass

7	Check the interface link between Gym and Gym Battle scenes.	Player interaction with NPC.	Students are redirected to the MCQ Question scene. If the link between the two scenes has not yet been implemented, use a dummy scene or test stub Console.WriteLine("Event triggered."); To indicate the event that triggers the change in scene.	Pass

3.2.3 Teacher Functionalities

Test Case ID	Test Case Objective	Test Case Description	Expected Result	Pass /Fail
1	Check the interface link between the Teacher Homepage scene and Edit Gym Questions scene.	Click on the “Edit Gym Questions” button from the Teacher Homepage menu.	Teacher is redirected to the Edit Gym Questions scene containing placeholder buttons and text.	Pass
2	Check the interface link between the Teacher HomePage scene and Set Assignment scene.	Click on the “Set Assignment” button from the Teacher Homepage menu.	Teacher is redirected to the Set Assignment scene.	Pass
3	Check the interface link between the Teacher HomePage scene and View Summary scene.	Click on the “View Summary” button from the Teacher Homepage menu.	Teacher is redirected to the View Summary scene with placeholder buttons and text.	Pass
4	Check the interface link between Teacher Homepage scene and Teacher Profile scene.	Click on the “Profile Page” button from the Teacher Homepage menu.	Teacher is redirected to the Profile Page scene with placeholder buttons and text.	Pass

3.3 Blackbox Testing

Black box testing involves testing a system with no prior knowledge of its internal workings. A tester provides an input, and observes the output generated by the system under test.

3.3.1. Login

Test Case ID:	1	Test Designed by:	Goh Hong Xiang, Bryan
Module Name:	Login	Test Executed by:	Chong Jing Hong
Test Title:	Verify Login Credentials	Test Execution Date:	31/10/2021
Description	Verify Login Credentials from user input with Login Credentials in Database		

Pre-conditions:	User must already have an existing account
Dependencies:	
Post-conditions:	The system notifies the user when login is successful or login failure

Step	Test Step	Test Data	Expected Results	Actual Results	Status	Notes
1	Start the game	-	Login menu form appears	As expected	Pass	
2	Provide email	student@e.ntu.edu.sg		As expected	Pass	
3	Provide password	admin123		As expected	Pass	
4	Click login button		Direct user to main game	As expected	Pass	Login Credentials is correct and within the database
Alternate Flows:						
1	Start the game	-	Login menu form appears	As expected	Pass	
2	Provide email	student1@e.ntu.edu.sg	-	As expected	Pass	
3	Provide password	helloworld		As expected	Pass	
4	Click login button		"Invalid email/password"	As expected	Pass	Login Credentials is wrong since password does not match with database credentials
1	Start the game	-	Login menu form appears	As expected	Pass	
2	Provide email	student1	-	As expected	Pass	
3	Provide password	admin123		As expected	Pass	
4	Click login button		"Invalid email/password"	As expected	Pass	Login Credentials is wrong since invalid email format
1	Start the game	-	Login menu form appears	As expected	Pass	
2	Provide email	student123@e.ntu.edu.sg	-	As expected	Pass	
3	Provide password	admin123	-	As expected	Pass	
4	Click login button		"Invalid email/password"	As expected	Pass	Login Credentials is wrong since email not in database

3.3.2. Register

Test Case ID:	2	**Test Designed by:**	Goh Hong Xiang, Bryan
Module Name:	Register	**Test Executed by:**	Chong Jing Hong
Test Title:	Register a new user account	**Test Execution Date:**	31/10/2021
Description	Create new account		

Pre-conditions:	User must not have an existing account
Dependencies:	
Post-conditions:	The system notifies the user when register is successful or register failure

Step	Test Step	Test Data	Expected Results	Actual Results	Status	Notes
1	Start the game	-	Login menu form appears	As expected	Pass	
2	Click on register button	-	Register menu form appear	As expected	Pass	
3	Provide username	studentrino	-	As expected	Pass	
4	Provide email	student@e.ntu.edu.sg	-	As expected	Pass	
5	Provide password	student	-	As expected	Pass	
6	Provide password again	student	-	As expected	Pass	
7	Click create button	-	Direct user to login menu	As expected	Pass	Email and password are of the correct format and both passwords provided match
Alternate Flows:						
1	Start the game	-	Login menu form appears	As expected	Pass	
2	Click on register button	-	Register menu form appear	As expected	Pass	
3	Provide username	studentrino	-	As expected	Pass	
4	Provide email	student123	-	As expected	Pass	
5	Provide password	student	-	As expected	Pass	
6	Provide password again	student	-	As expected	Pass	
7	Click create button	-	"Invalid email/password"	As expected	Pass	Email is not of the correct format
1	Start the game	-	Login menu form appears	As expected	Pass	
2	Click on register button	-	Register menu form appear	As expected	Pass	
3	Provide username	studentrino	-	As expected	Pass	
4	Provide email	student@e.ntu.edu.sg	-	As expected	Pass	
5	Provide password	student	-	As expected	Pass	
6	Provide password again	student123	-	As expected	Pass	
7	Click create button	-	"Passwords do not match"	As expected	Pass	Password entries do not match each other
1	Start the game	-	Login menu form appears	As expected	Pass	
2	Click on register button	-	Register menu form appear	As expected	Pass	
3	Provide username	studentrino	-	As expected	Pass	
4	Provide email	student@e.ntu.edu.sg	-	As expected	Pass	
5	Provide password	student	-	As expected	Pass	
6	Provide password again	student	-	As expected	Pass	
7	Click create button	-	"Invalid email/password"	As expected	Pass	Email already in use

3.3.3. Control

Test Case ID:	3	**Test Designed by:**	Goh Hong Xiang, Bryan
Module Name:	Controls	**Test Executed by:**	Chong Jing Hong
Test Title:	Gameplay controls and flows	**Test Execution Date:**	31/10/2021
Description	Gameplay controls and flows		

Pre-conditions:	Users must have an existing account and are already logged in
Dependencies:	
Post-conditions:	Character moves

Step	Test Step	Test Data	Expected Results	Actual Results	Status	Notes
1	Press up arrow key / w key		Character moves up	As expected	Pass	
2	Press right arrow / d key		Character moves right	As expected	Pass	
3	Press down arrow / s key		Character moves down	As expected	Pass	
4	Press left arrow / a key		Character moves left	As expected	Pass	

3.3.4. World

Test Case ID:	4	**Test Designed by:**	Goh Hong Xiang, Bryan
Module Name:	World	**Test Executed by:**	Chong Jing Hong
Test Title:	Gameplay flows	**Test Execution Date:**	31/10/2021
Description	Gameplay flows		

Pre-conditions:	Users must have an existing account and are already logged in
Dependencies:	
Post-conditions:	Gameplay events

Step	Test Step	Test Data	Expected Results	Actual Results	Status	Notes
1	Move character to portal		Character is supposed to move to another scene	As expected	Pass	
1	Move character in the grass patches		Random knowledge will appear	As expected	Pass	

3.3.5. Gym

Test Case ID:	5	**Test Designed by:**	Goh Hong Xiang, Bryan
Module Name:	Gym	**Test Executed by:**	Chong Jing Hong
Test Title:	Gym battle	**Test Execution Date:**	31/10/2021
Description	Interacting with the gym leader NPCs		

Pre-conditions:	User must already entered through the gym portal
Dependencies:	
Post-conditions:	Users' scores are logged

Step	Test Step	Test Data	Expected Results	Actual Results	Status	Notes
1	Move the character to the NPC	-	Character is able to reach the NPC	As expected	Pass	
2	Interact with the NPC	Press the 'e' key	Dialog(s) between the player and NPC is shown	As expected	Pass	
3	Challenge the NPC	-	Player is brought to the question interface and starts solving the questions	As expected	Pass	

3.3.6. Questions

Test Case ID:	6	**Test Designed by:**	Goh Hong Xiang, Bryan
Module Name:	Questions	**Test Executed by:**	Chong Jing Hong
Test Title:	Questionaire	**Test Execution Date:**	31/10/2021
Description	Answering quiz questions		

Pre-conditions:	Users should already be logged in, users should have already interacted with the NPC
Dependencies:	
Post-conditions:	The system notifies the user on their score

Step	Test Step	Test Data	Expected Results	Actual Results	Status	Notes
1	User selects an answer for question 1	-	Answer is highlighted	As expected	Pass	
2	User confirms selection	-	Next question is shown	As expected	Pass	
3	User selects an answer for question 2	-	Answer is highlighted	As expected	Pass	
4	User confirms selection	-	Next question is shown	As expected	Pass	
5	User selects an answer for question 3	-	Answer is highlighted	As expected	Pass	
6	User confirms selection	-	Next question is shown	As expected	Pass	
7	User selects an answer for question 4	-	Answer is highlighted	As expected	Pass	
8	User confirms selection	-	Next question is shown	As expected	Pass	
9	User selects an answer for question 5	-	Answer is highlighted	As expected	Pass	
10	User confirms selection	-	Quiz ends, score is displayed	As expected	Pass	

3.3.7. Create Assignment / Challenges

Test Case ID:	7	**Test Designed by:**	Goh Hong Xiang, Bryan
Module Name:	Create Assignment / Challenges	**Test Executed by:**	Chong Jing Hong
Test Title:	Create Assignment / Challenges	**Test Execution Date:**	31/10/2021
Description	Create new questions for students to solve and challenge themselves		

Pre-conditions:	User must be a teacher and have already been logged in
Dependencies:	
Post-conditions:	The system notifies the teacher when questions are created already

Step	Test Step	Test Data	Expected Results	Actual Results	Status	Notes
1	Click on the + icon	-	Popup interface should appear for user input	Popup interface appears	Pass	
2	Input the question	-	User should be able to input text / string	User is able to input text / string	Pass	
3	Input the options	-	User should be able to input text / string	User is able to input text / string	Pass	
4	Input the correct answer	-	User should be able to input text / string	User is able to input text / string	Pass	
5	Input the marks	-	User should be able to input int	User is able to input int	Pass	
6	Click on the done button	-	System should notify the teacher when the questions are created successfully	System notifies the teacher when the questions are created successfully	Pass	

3.3.8. Report Generation

Test Case ID:	8	**Test Designed by:**	Goh Hong Xiang, Bryan
Module Name:	Report Generation	**Test Executed by:**	Chong Jing Hong
Test Title:	Generate Report	**Test Execution Date:**	31/10/2021
Description	Generate Report by teachers to view students' progress		

Pre-conditions:	User must be a teacher and has already logged in
Dependencies:	
Post-conditions:	The system generates a csv file

Step	Test Step	Test Data	Expected Results	Actual Results	Status	Notes
1	Click on the file icon	-	The csv file should be generated and console logged 'report generated'	The csv file is generated and console logged 'report generated'	Pass	
Alternate Flows: if student did not attempt any challenges / assignment						
1	Click on the file icon	-	The csv file should not be generated and console logged 'no report to be generated'	The csv file is not generated and console logged 'no report to be generated'	Pass	

3.3.9. Leaderboard

Test Case ID:	9	**Test Designed by:**	Goh Hong Xiang, Bryan
Module Name:	Leaderboard	**Test Executed by:**	Chong Jing Hong
Test Title:	Leaderboard	**Test Execution Date:**	31/10/2021
Description	Viewing student's ranking in the leaderboard based on gyms, assignments and challenges		

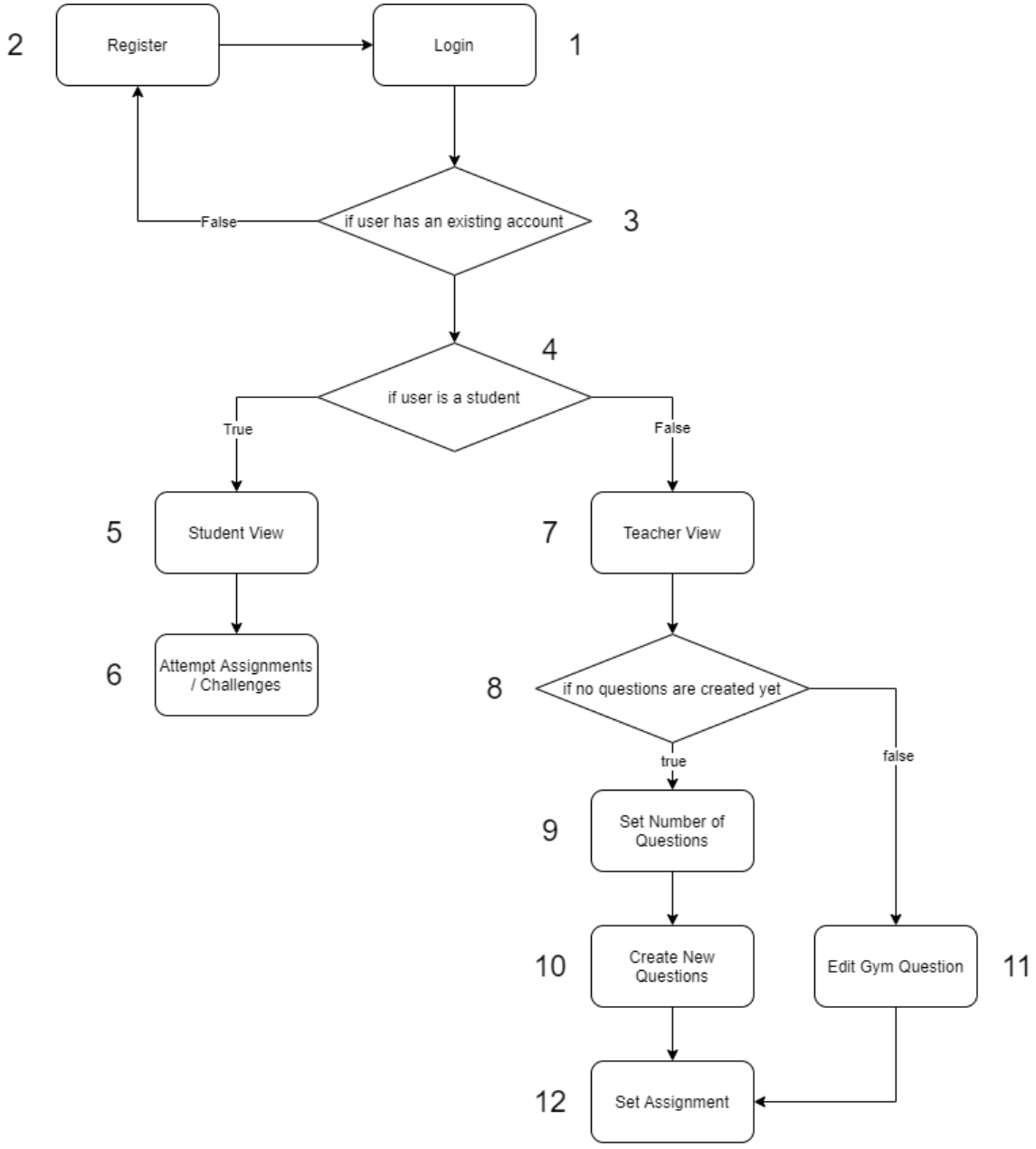
Pre-conditions:	Users should already be logged in, users should have already finished at least 1 gym battle, assignment or challenge
Dependencies:	Gym battle score, Assignment score, Challenge score
Post-conditions:	Students' ranking are correctly displayed

Step	Test Step	Test Data	Expected Results	Actual Results	Status	Notes
1	Student clicks on 'View Leaderboard' button	-	Leaderboard page is displayed, student's rank is highlighted and score is correctly displayed	Leaderboard page is displayed, student's rank is highlighted and score is correctly displayed	Pass	

3.4 Whitebox Testing

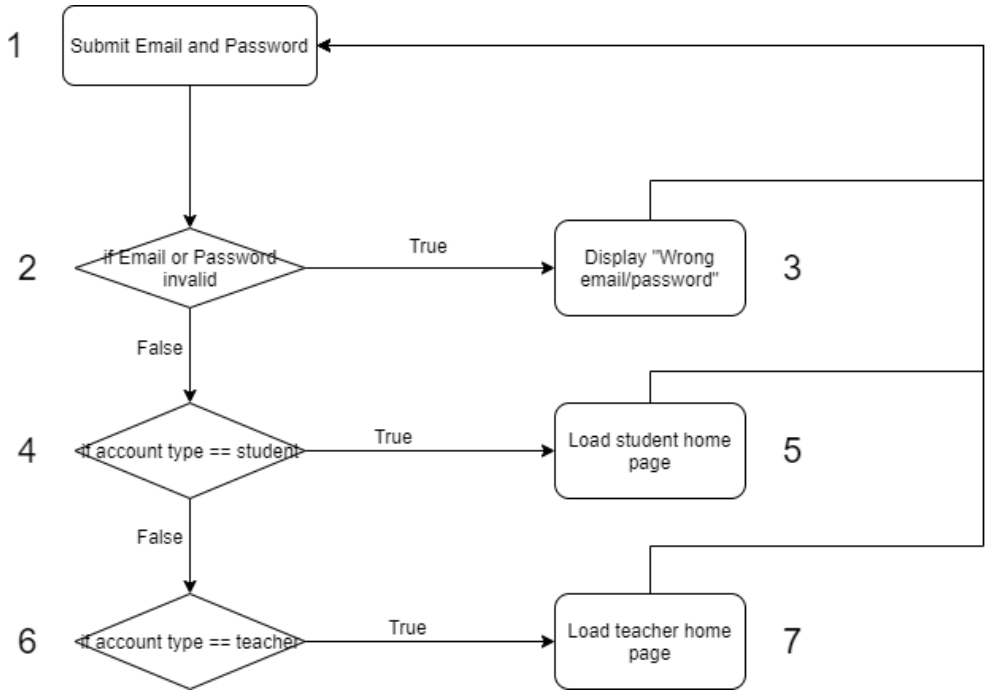
White box testing involves the testing of the software code for the following: Internal security holes, broken or poorly structured paths in the coding processes, the flow of specific inputs through the code.

3.4.1. Overall



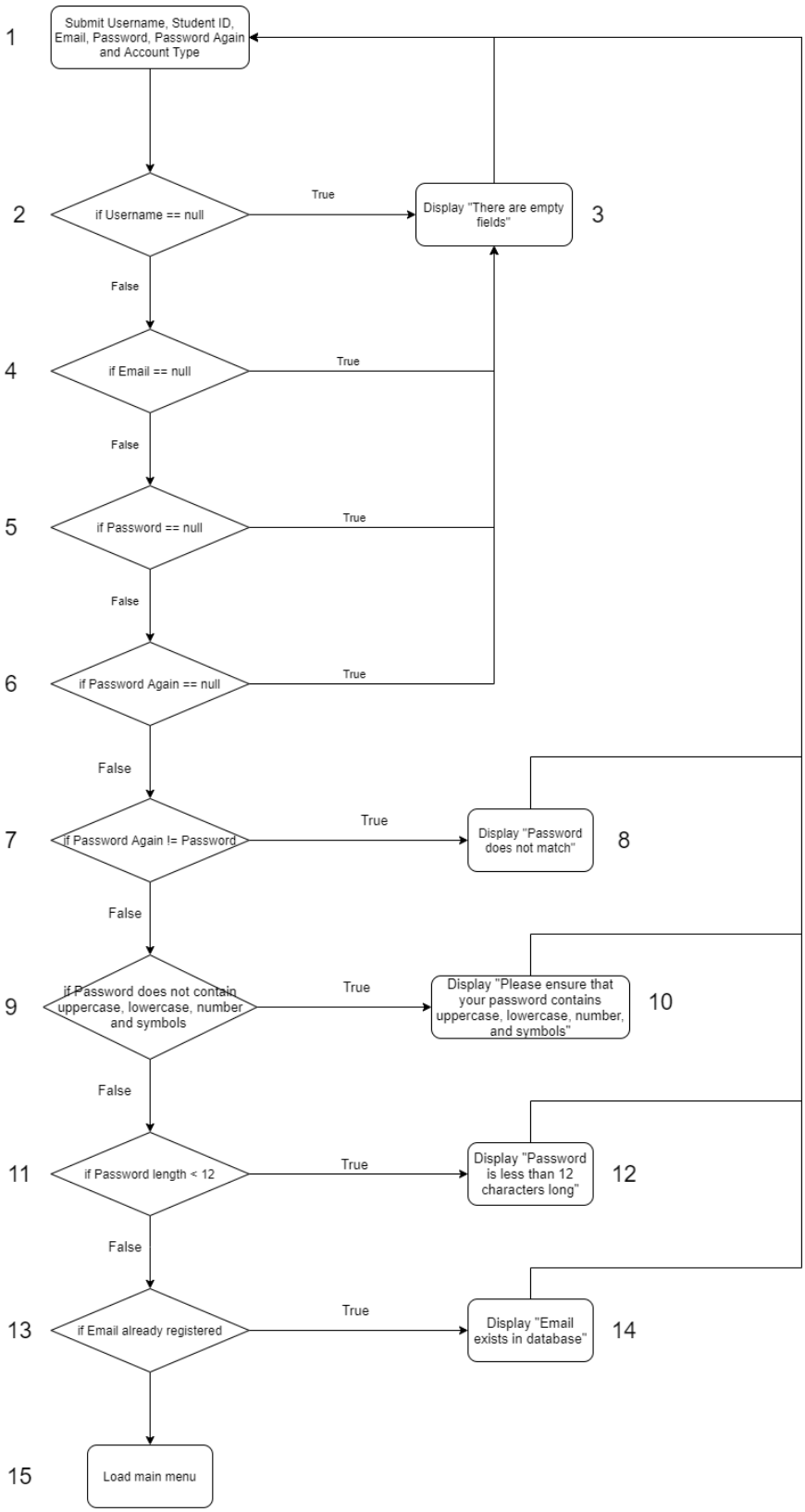
Test Case ID #01	Test Case Name: Overall
Test Priority: High	
Test Designed By: Chong Jing Hong	Design Date: 31/10/21
Test Executed By: Goh Hong Xiang, Bryan	All Basis Paths Passed: Yes
Basis Paths: 1. 1, 3, 2, 1 2. 1, 3, 4, 5, 6 3. 1, 3, 4, 7, 8, 11, 12 4. 1, 3, 4, 7, 8, 9, 10, 12	

3.4.2. Login



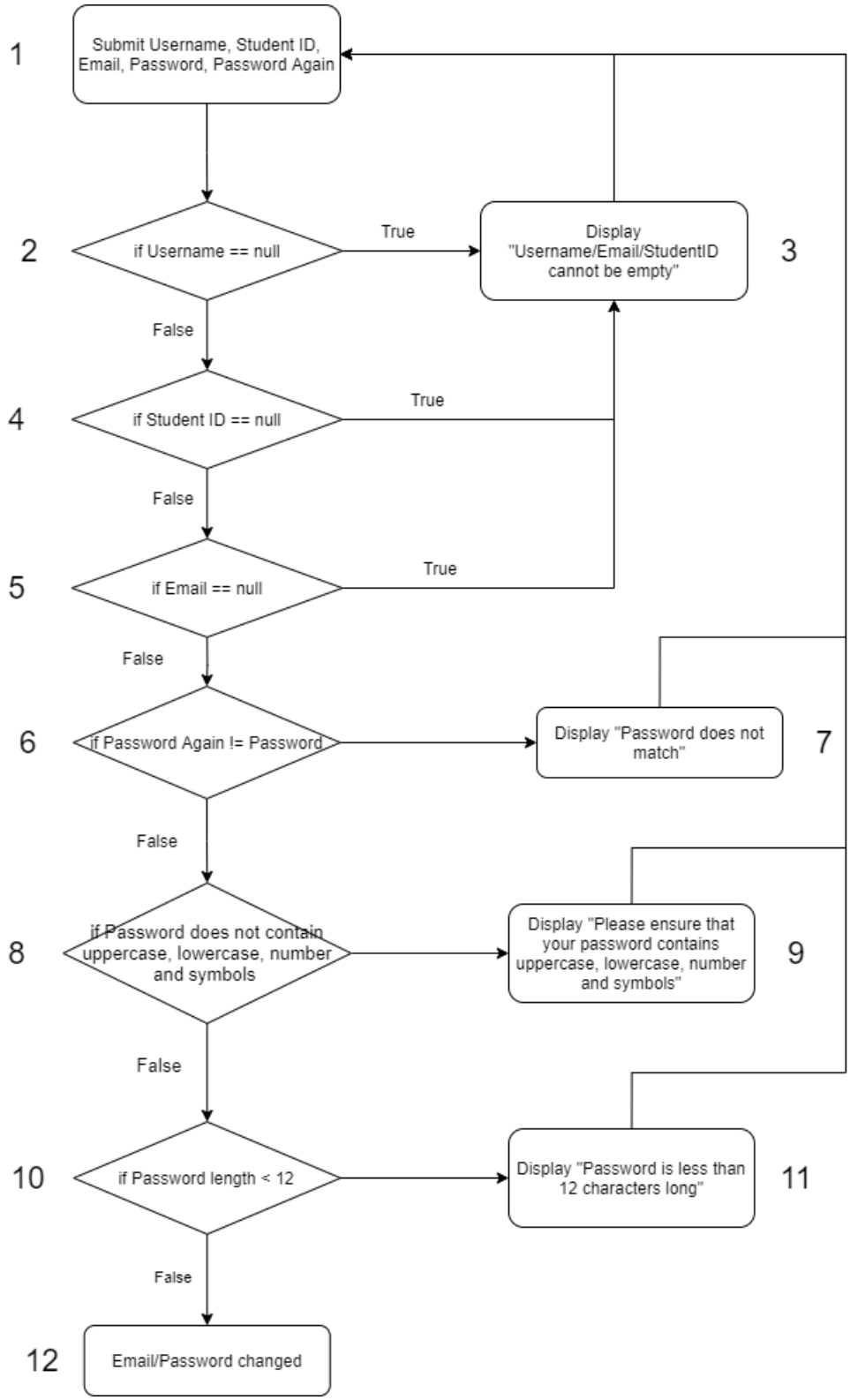
Test Case ID #02	Test Case Name: Login
Test Priority: High	
Test Designed By: Chong Jing Hong	Design Date: 31/10/21
Test Executed By: Goh Hong Xiang, Bryan	All Basis Paths Passed: Yes
Basis Paths: 1. 1, 2, 3, 1 2. 1, 2, 4, 5, 1 3. 1, 2, 4, 6, 7, 1	

3.4.3 Register



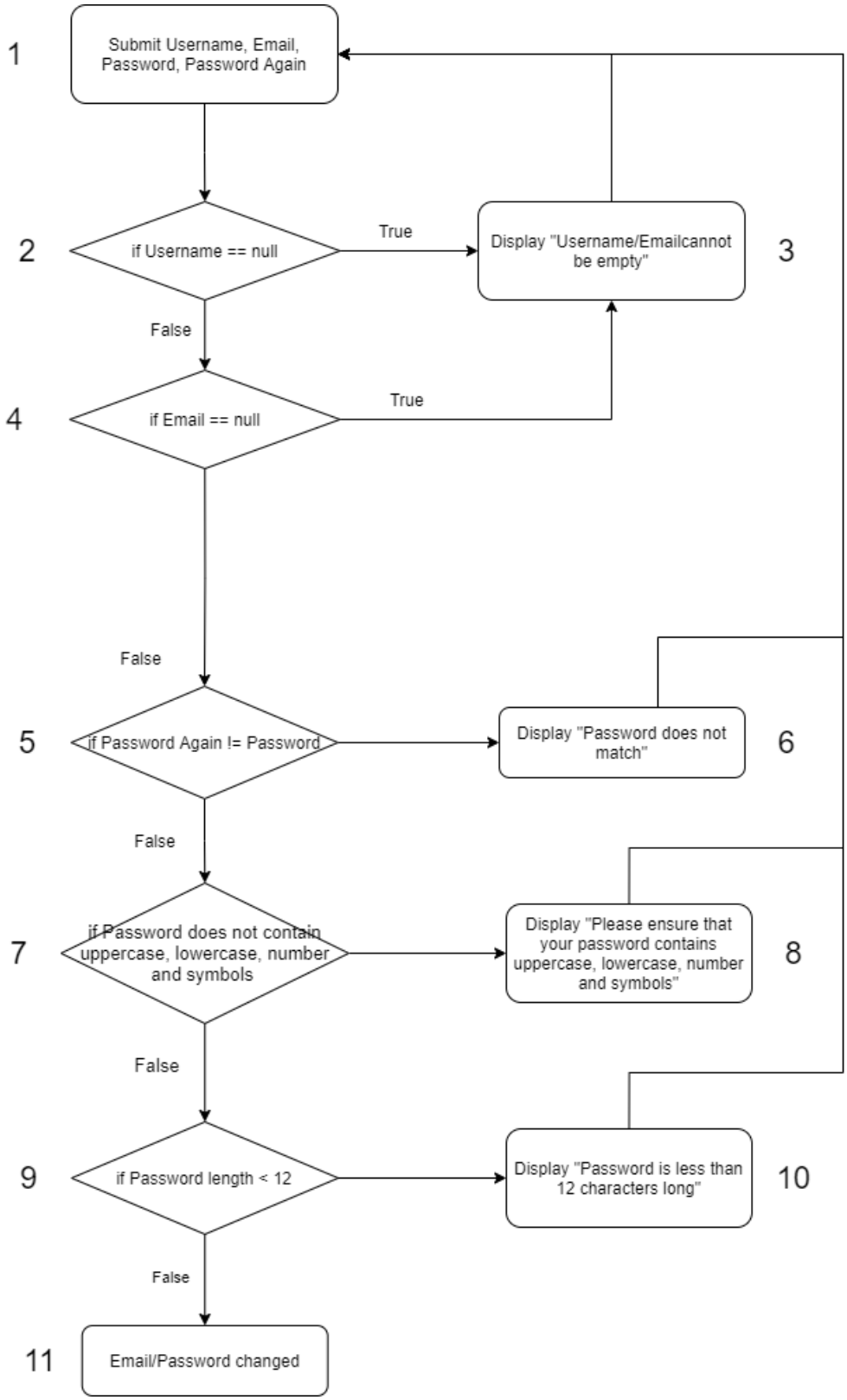
Test Case ID #03	Test Case Name: Register
Test Priority: High	
Test Designed By: Chong Jing Hong	Design Date: 31/10/21
Test Executed By: Goh Hong Xiang, Bryan	All Basis Paths Passed: Yes
Basis Paths: 1. 1, 2, 3, 1 2. 1, 2, 4, 3, 1 3. 1, 2, 4, 5, 3, 1 4. 1, 2, 4, 5, 6, 3, 1 5. 1, 2, 4, 5, 6, 7, 8, 1 6. 1, 2, 4, 5, 6, 7, 9, 10, 1 7. 1, 2, 4, 5, 6, 7, 9, 11, 12, 1 8. 1, 2, 4, 5, 6, 7, 9, 11, 13, 14, 1 9. 1, 2, 4, 5, 6, 7, 9, 11, 13, 15	

3.4.4 Edit Student Profile



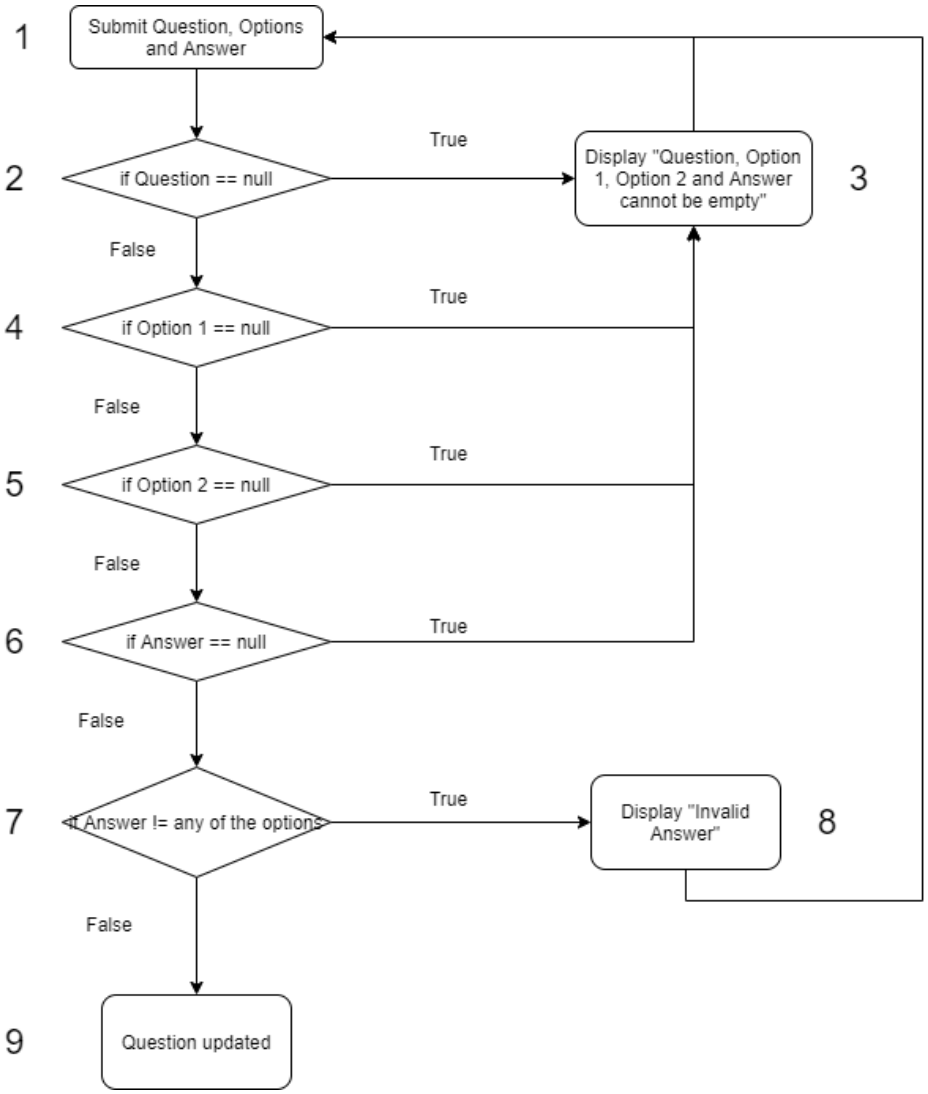
Test Case ID #04	Test Case Name: Edit Student Profile
Test Priority: High	
Test Designed By: Chong Jing Hong	Design Date: 31/10/21
Test Executed By: Goh Hong Xiang, Bryan	All Basis Paths Passed: Yes
Basis Paths: 1. 1, 2, 3, 1 2. 1, 2, 4, 3, 1 3. 1, 2, 4, 5, 3, 1 4. 1, 2, 4, 5, 6, 7, 1 5. 1, 2, 4, 5, 6, 8, 9, 1 6. 1, 2, 4, 5, 6, 8, 10, 11, 1 7. 1, 2, 4, 5, 6, 8, 10, 12	

3.4.5 Edit Teacher Profile



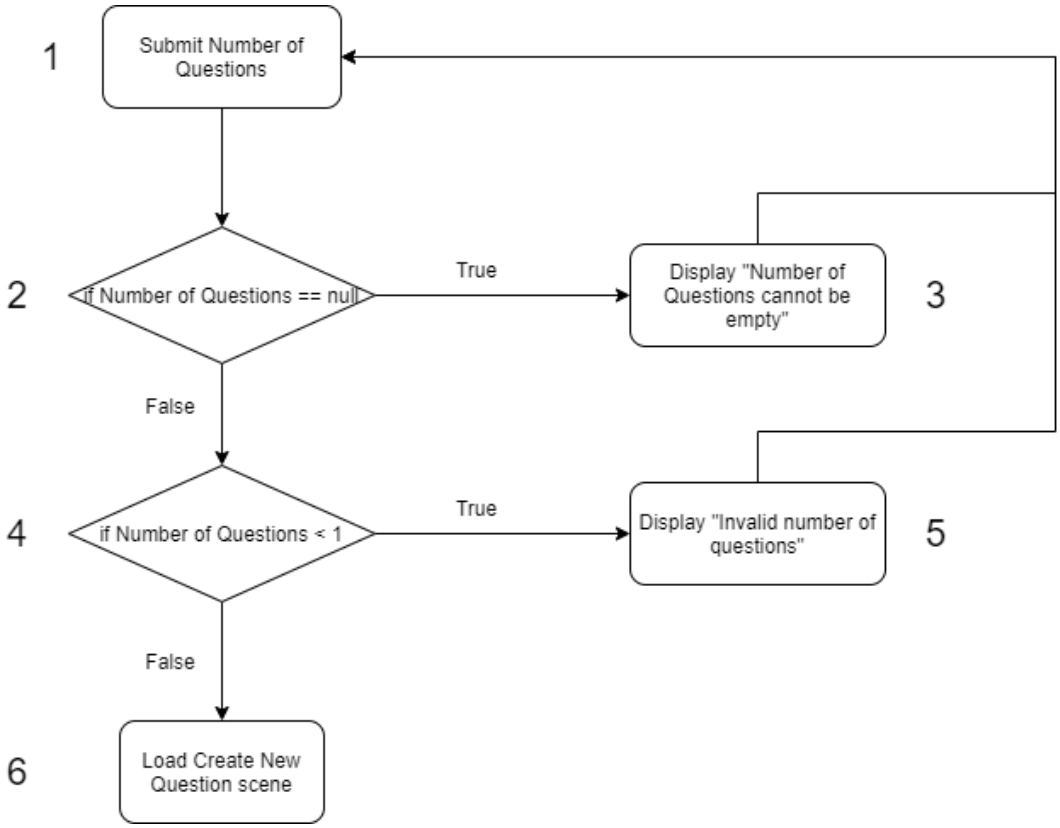
Test Case ID #05	Test Case Name: Edit Teacher Profile
Test Priority: High	
Test Designed By: Chong Jing Hong	Design Date: 31/10/21
Test Executed By: Goh Hong Xiang, Bryan	All Basis Paths Passed: Yes
Basis Paths: 1. 1, 2, 3, 1 2. 1, 2, 4, 3, 1 3. 1, 2, 4, 5, 6, 1 4. 1, 2, 4, 5, 7, 8, 1 5. 1, 2, 4, 5, 7, 9, 10, 1 6. 1, 2, 4, 5, 7, 9, 11	

3.4.6 Edit Gym Question



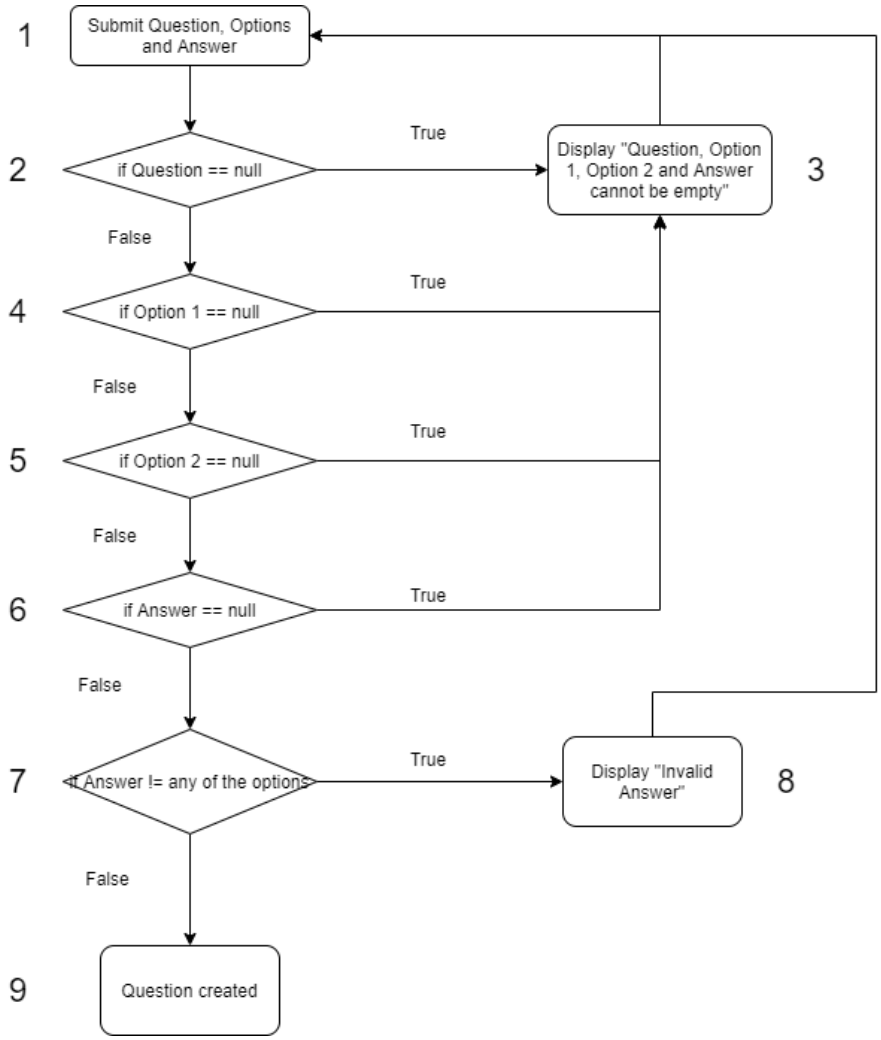
Test Case ID #06	Test Case Name: Edit Gym Questions
Test Priority: High	
Test Designed By: Chong Jing Hong	Design Date: 31/10/21
Test Executed By: Goh Hong Xiang, Bryan	All Basis Paths Passed: Yes
Basis Paths: 1. 1, 2, 3, 1 2. 1, 2, 4, 3, 1 3. 1, 2, 4, 5, 3, 1 4. 1, 2, 4, 5, 6, 3, 1 5. 1, 2, 4, 5, 6, 7, 8, 1 6. 1, 2, 4, 5, 6, 7, 9	

3.4.7 Set Number of Questions



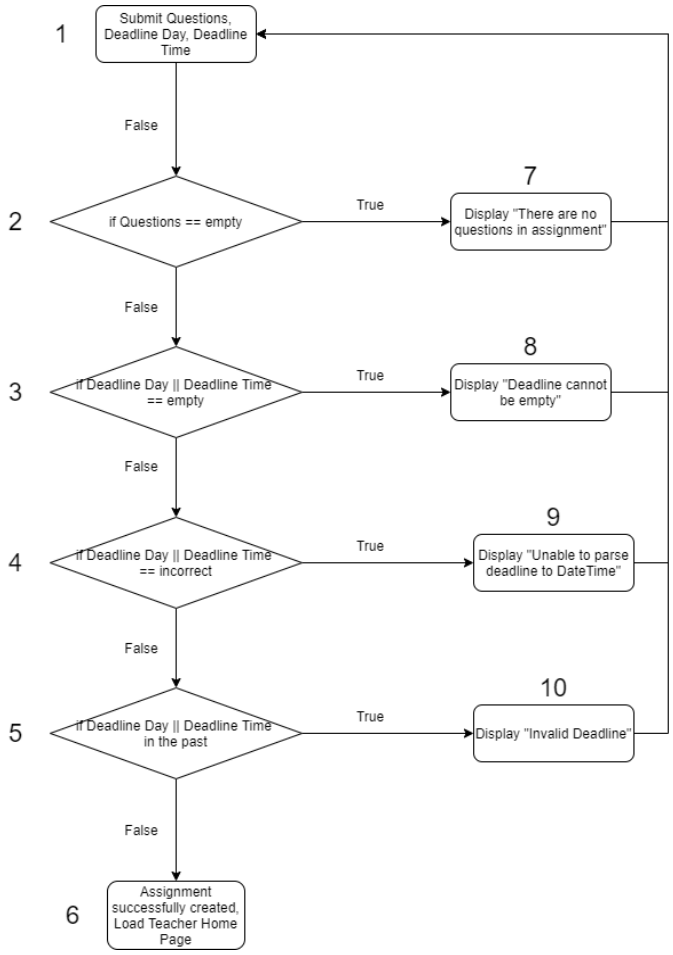
Test Case ID #07	Test Case Name: Set Number of Questions
Test Priority: High	
Test Designed By: Chong Jing Hong	Design Date: 31/10/21
Test Executed By: Goh Hong Xiang, Bryan	All Basis Paths Passed: Yes
Basis Paths: 1. 1, 2, 3, 1 2. 1, 2, 4, 5, 1 3. 1, 2, 4, 6	

3.4.8 Create New Question



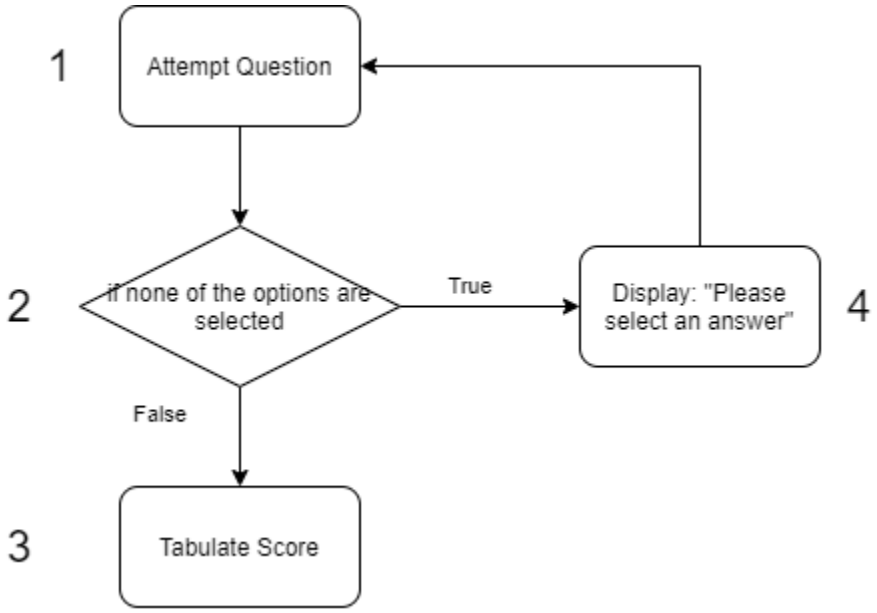
Test Case ID #08	Test Case Name: Create New Question
Test Priority: High	
Test Designed By: Chong Jing Hong	Design Date: 31/10/21
Test Executed By: Goh Hong Xiang, Bryan	All Basis Paths Passed: Yes
Basis Paths: 1. 1, 2, 3, 1 2. 1, 2, 4, 3, 1 3. 1, 2, 4, 5, 3, 1 4. 1, 2, 4, 5, 6, 3, 1 5. 1, 2, 4, 5, 6, 7, 8, 1 6. 1, 2, 4, 5, 6, 7, 9	

3.4.9 Set Assignment



Test Case ID #09	Test Case Name: Set Assignment
Test Priority: High	
Test Designed By: Chong Jing Hong	Design Date: 31/10/21
Test Executed By: Goh Hong Xiang, Bryan	All Basis Paths Passed: Yes
Basis Paths: 1. 1, 2, 7, 1 2. 1, 2, 3, 8, 1 3. 1, 2, 3, 4, 9, 1 4. 1, 2, 3, 4, 5, 10, 1 5. 1, 2, 3, 4, 5, 6	

3.4.10 Attempt Assignments / Challenges



Test Case ID #10	Test Case Name: Attempt Question
Test Priority: High	
Test Designed By: Chong Jing Hong	Design Date: 31/10/21
Test Executed By: Goh Hong Xiang, Bryan	All Basis Paths Passed: Yes
Basis Paths: 1. 1, 2, 4, 1 2. 1, 2, 3	

4. Load/Performance Testing

4.1 Introduction

For the game to be tested in real life scenarios, Load tests must be carried out. This allows us to evaluate how the game would perform upon release and if further changes are needed.

4.2 Setup

The Locust load testing tool was used to perform the load test. A test script was created using the locus library written in python.

The script is located at “Test Script/locust.py”

```python
import time
from locust import HttpUser, task, between

class ApiLoadTest(HttpUser):
    wait_time = between(1, 10)

    @task
    def get_account(self):
        self.client.get(url='account/6SjXMF6QFphad1QaNQbPzhc98I22')

    @task
    def get_attempt(self):
        self.client.get(url='attempt/D2vrZQSiHUQuNG2u2qMT')

    @task
    def get_question(self):
        self.client.get(url='question/5XpEOTq7yUD36MhiVwq8')

    @task
    def get_gym(self):
        self.client.get(url='gym/2AcLYXxjcwUH8M6pS4kZ')

    @task
    def get_assignment(self):
        self.client.get(url='assignment/LUpzwPCNSUvWLw9NFeNM')
```

Figure 1: Screenshot of locust test script

As the game mainly uses API calls, the script focuses on testing the API calls. For each API called, the same weight would be applied as we need to stress the server as much as possible

Server – Firebase

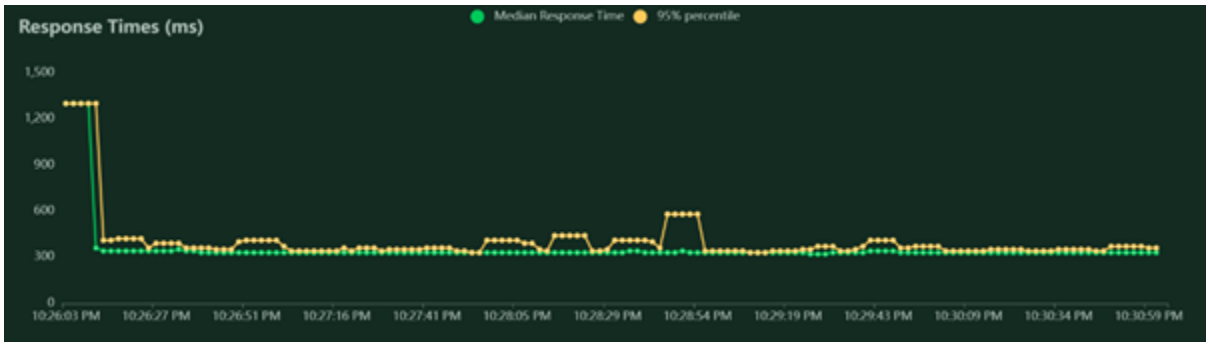
Client – The load test was conducted on a laptop running on 100Mbps network

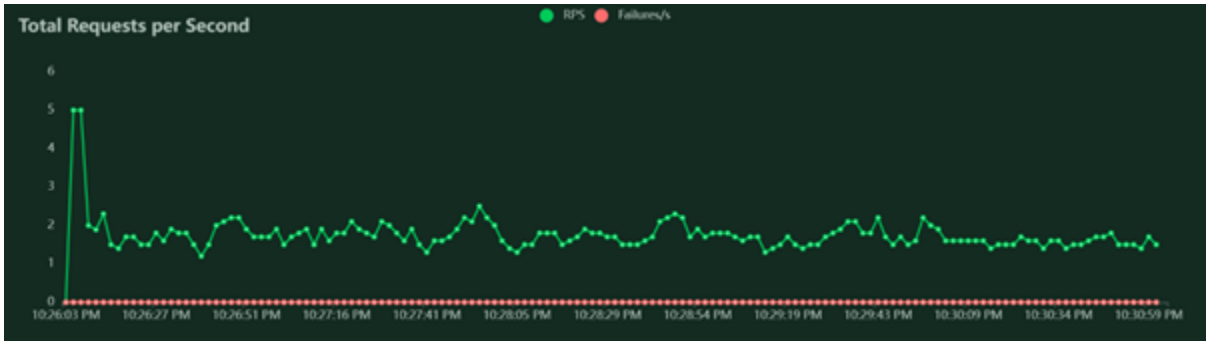
4.3 Result

10 Concurrent Users

The load test was firstly conducted with 10 concurrent users for 5 minutes. The results are as of follows:

Type	Name	# Requests	# Fails	Median (ms)	90%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s
GET	/account/6SjXMF6QFphad1QaNQbPzhc98I22	106	0	330	350	363	314	1275	183	0.2	0
GET	/assignment/LUpzwPCNSUvWLw9NFeNM	98	0	330	360	345	314	1286	189	0.2	0
GET	/attempt/D2vrZQSiHUQuNG2u2qMT	101	0	330	350	343	314	1282	271	0.2	0
GET	/gym/2AcLYXxjcwUH8M6pS4kZ	109	0	330	350	359	315	1305	162	0.4	0
GET	/question/5XpEOTq7yUD36MhiVwq8	105	0	330	360	352	313	1335	333	0.5	0
	Aggregated	**519**	**0**	**330**	**350**	**352**	**313**	**1335**	**227**	**1.5**	**0**



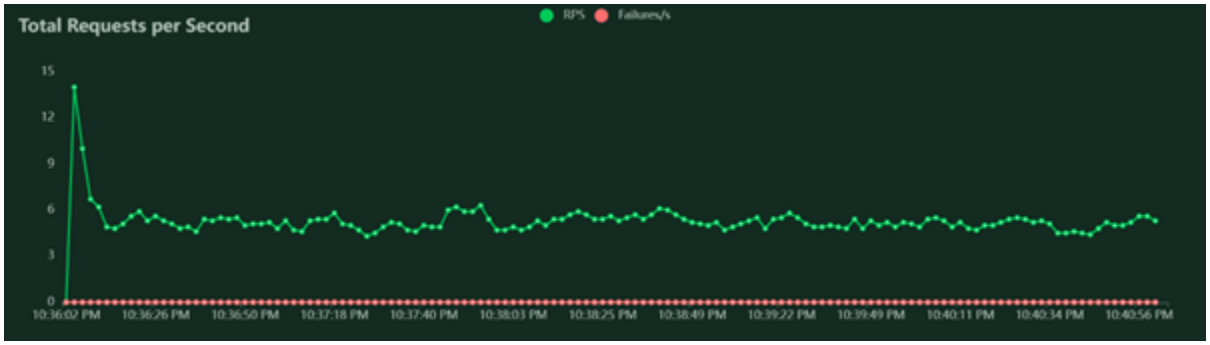


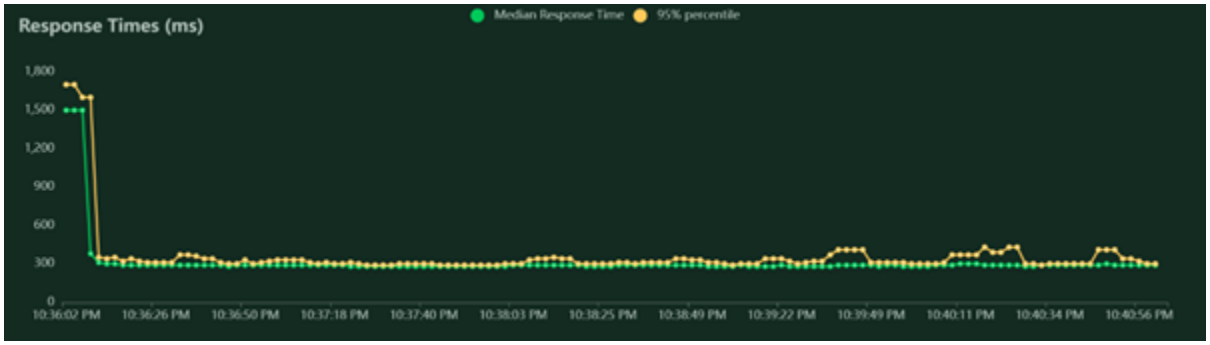
From the report shown above, on average the response time is around 350ms which shows that there was very little stress done on the server. Across all API calls, the average response time was the same with 0 fail requests. With each API a maximum response time of 1.2 – 1.3 seconds, the game was still playable.

30 Concurrent Users

The load test was next conducted with 30 concurrent users for 5 minutes to stimulate a scenario where a lab session is being carried out. The results are as follows:

Type	Name	# Requests	# Fails	Median (ms)	90%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s
GET	/account/6SjXMF6QFphad1QaNQbPzhc98I22	332	0	290	320	310	278	1555	183	0.7	0
GET	/assignment/LUpzwPCNSUvWLw9NFeNM	320	0	290	320	332	277	1695	189	1.3	0
GET	/attempt/D2vrZQSiHUQuNG2u2qMT	300	0	290	310	319	277	1564	271	1	0
GET	/gym/2AcLYXxjcwUH8M6pS4kZ	316	0	290	310	312	277	1484	162	1.1	0
GET	/question/5XpEOTq7yUD36MhiVwq8	311	0	290	310	305	278	1491	333	1.2	0
	Aggregated	**1579**	**0**	**290**	**310**	**316**	**277**	**1695**	**226**	**5.3**	**0**



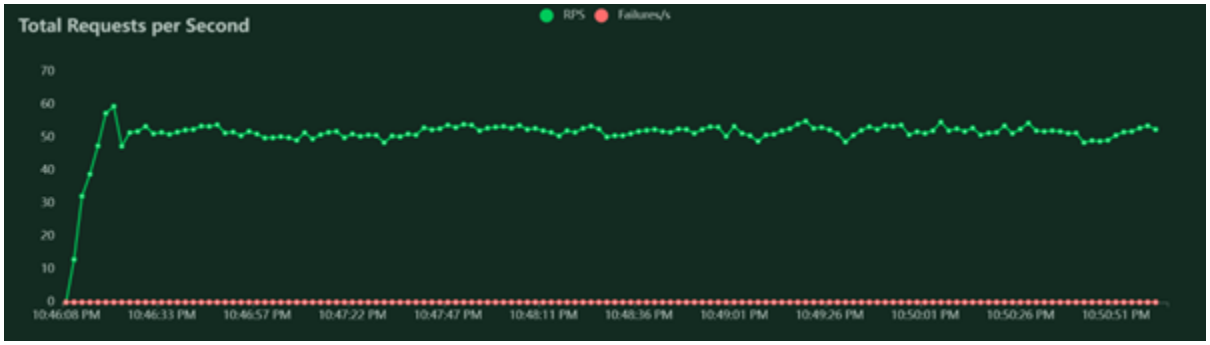


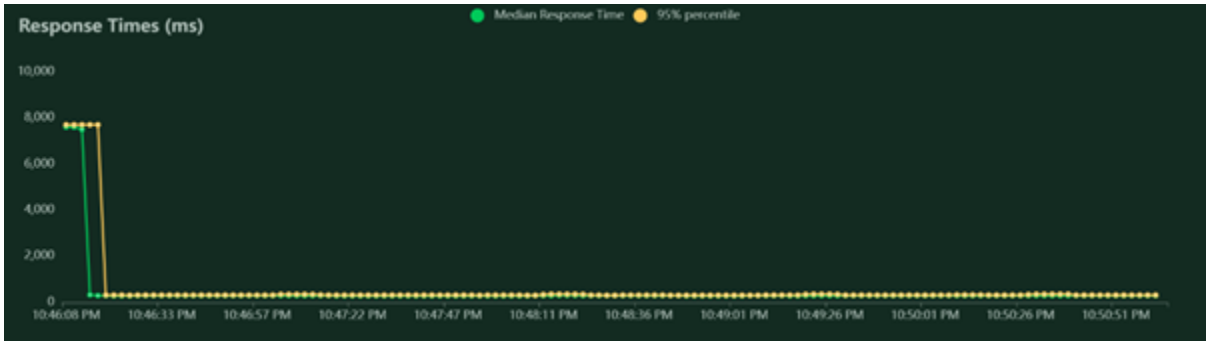
From the report above, the average response time is slightly better as compared to the test with 10 users. Even though the number of attempts increased significantly, the response time was similar to the previous test with 0 fail attempts. With the maximum response time of 1.4-1.5 seconds for each API, the game was still playable.

300 Concurrent Users

To simulate a scenario where the whole course is playing the game at the same time, we next conducted a test with 300 concurrent users. The results are as of follows:

Type	Name	# Requests	# Fails	Median (ms)	90%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s
GET	/account/6SjXMF6QFphad1QaNQbPzhc98I22	3073	0	290	310	435	276	11259	183	9.6	0
GET	/assignment/LUpzwPCNSUvWLw9NFeNM	3025	0	290	310	416	274	7741	189	10.5	0
GET	/attempt/D2vrZQSiHUQuNG2u2qMT	3114	0	290	310	457	276	12427	271	11.3	0
GET	/gym/2AcLYXxjcwUH8M6pS4kZ	3176	0	290	310	420	275	7775	162	10.9	0
GET	/question/5XpEOTq7yUD36MhiVwq8	3023	0	290	310	452	277	7741	333	10.2	0
	Aggregated	**15411**	**0**	**290**	**310**	**436**	**274**	**12427**	**227**	**52.5**	**0**





From the report above, the average time is slightly higher from the first and second test. Even though the initial response time is around 7 – 10 seconds across all API, it quickly drops as time passes. With an average response time of 436ms, the game will still be playable. Even though there was a slight change in average response, there are still 0 failed attempts.

4.4 Conclusion

In conclusion, the game runs most optimally with 10 concurrent users but realistically, the game can run with 30 users or even 300 users with certain requests having a slight delay which is acceptable. With 300 concurrent users, Firebase is able to handle the request with 0 failed attempts. Performance might further improve with different hosting services.