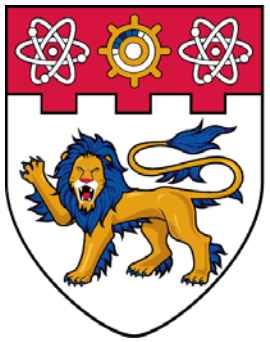

NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

CZ3003 Software System Analysis & Design

Lab 4 Deliverable: Test

Good Game Well Played (GGWP)

SS2

Leroy Tang Zi Wei

Alson Kong

Karen Wong

Lim Wei Cong

Sam Jian Shen

Cole Ting-Chun Kuo

Zane Ho

Norman Fung

Jing Herng Pow

Use Case: Register				
Test Data: Username: test_user Email: new_email@gmail.com Password: 12345678 Confirm Password: 12345678				
Step No	Steps	Expected	Result	Use Case Step
1	User loads application	Application to load successfully	Application loads successfully	-
2	User selects registration option in the application's login page	Application to redirect user to registration page	Application redirects the user to the registration page.	1,2
3	User enters required information: Username: test_user Email: new_email@gmail.com Password: 12345678 Confirm Password: 12345678	-	-	3
4	User submits the form	Application to inform user that registration is successful	"Account Created" message is displayed	4-7

Use Case: Register(AF-S4)

Test Data: Username: Empty Email: Empty Password: Empty Confirm Password: Empty				
Step No	Steps	Expected	Result	Use Case Step
1	User loads application	Application to load successfully	Application loads successfully	-
2	User selects registration option in the application's login page	Application to redirect user to registration page	Application redirects the user to the registration page.	1,2
3	Test Data: Username: Empty Email: Empty Password: Empty Confirm Password: Empty	-	-	3
4	User submits the form	Application to inform user that all fields are required for registration	"All fields are required" message is displayed	4-7

Use Case: Register(AF-S4)

Test Data:
Username: test_user
Email: new_emailAgmail.com
Password: 12345678
Confirm Password: 12345678

Step No	Steps	Expected	Result	Use Case Step
1	User loads application	Application to load successfully	Application loads successfully	-
2	User selects registration option in the application's login page	Application to redirect user to registration page	Application redirects the user to the registration page.	1,2
3	Test Data: Username: test_user Email: new_emailAgmail.com Password: 12345678 Confirm Password: 12345678	-	-	3
4	User submits the form	Application to inform user that email entered is not in valid email format	"Email entered is not in valid email format" message is displayed	4

Use Case: Register(AF-S4)

Test Data:
Username: test_user
Email: new_email@gmail.com
Password: 1234567
Confirm Password: 1234567

Step No	Steps	Expected	Result	Use Case Step
1	User loads application	Application to load successfully	Application loads successfully	-
2	User selects registration option in the application's login page	Application to redirect user to registration page	Application redirects the user to the registration page.	1,2
3	Test Data: Username: test_user Email: new_emailAgmail.com Password: 1234567 Confirm Password: 1234567	-	-	3
4	User submits the form	Application to inform user that the password entered is too short	"Password is required to be at least 8 characters" message is displayed	4

Use Case: Register(AF-S4)

Test Data:
Username: test_user
Email: new_email@gmail.com
Password: 123456781
Confirm Password: 12345678

Step No	Steps	Expected	Result	Use Case Step
1	User loads application	Application to load successfully	Application loads successfully	-
2	User selects registration option in the application's login page	Application to redirect user to registration page	Application redirects the user to the registration page.	1,2
3	Test Data: Username: test_user Email: new_emailAgmail.com Password: 123456781 Confirm Password: 12345678	-	-	3
4	User submits the form	Application to inform user that passwords entered have to match	"Password and confirm password does not match" message is displayed	4

Use Case: Register(AF-S5)

Existing Data:
Username: test_user
Email: new_email@gmail.com
Password: 12345678
Confirm Password 12345678

Test Data:
Username: test_user
Email: new_email@gmail.com
Password: 12345678
Confirm Password: 12345678

Step No	Steps	Expected	Result	Use Case Step
1	User loads application	Application to load successfully	Application loads successfully	-
2	User selects registration option in the application's login page	Application to redirect user to registration page	Application redirects the user to the registration page.	1,2
3	Test Data: Username: test_user Email: new_emailAgmail.com Password: 123456781 Confirm Password: 12345678	-	-	3
4	User submits the form	Application to inform user that username has been taken	"Username or email is taken already" message is displayed	4

Use Case: Local Account Login				
Existing Data: Username: test_user Email: new_email@gmail.com Password: 12345678 Confirm Password 12345678				
Step No	Steps	Expected	Result	Use Case Step
1	User loads application(first page of application is login)	Application to load successfully	Application loads successfully	1,2
3	User enters the login information Test Data: Username: test_user Password: 12345678	-	-	3
4	User submits the form	Application to inform user that he has been successfully authenticated as a player and provides redirect link to start the game	“Logged in as player. Click here to start the game” message is displayed.	4-6

Use Case: Local Account Login(AF-S4)
Existing Data: Username: test_user

Email: new_email@gmail.com Password: 12345678 Confirm Password 12345678				
Step No	Steps	Expected	Result	Use Case Step
1	User loads application(first page of application is login)	Application to load successfully	Application loads successfully	1,2
3	User enters the login information Test Data: Username: test_user Password: 1234	-	-	3
4	User submits the form	Application to inform user that password provided is too short	“Password has to be at least 8 characters” message is displayed.	4-6

Use Case: Local Account Login(AF-S5)
Existing Data: Username: test_user Email: new_email@gmail.com

Password: 12345678 Confirm Password 12345678				
Step No	Steps	Expected	Result	Use Case Step
1	User loads application(first page of application is login)	Application to load successfully	Application loads successfully	1,2
3	User enters the login information Test Data: Username: test_user Password: 1234	-	-	3
4	User submits the form	Application to inform user that login details is invalid	"Invalid account details entered" message is displayed.	4

Use Case: Local Account Login(AF-S7)
Existing Data: Username: test_management Email: new_maangement@gmail.com Password: 12345678

Confirm Password 12345678				
Step No	Steps	Expected	Result	Use Case Step
1	User loads application(first page of application is login)	Application to load successfully	Application loads successfully	1,2
3	User enters the login information Test Data: Username: test_management Password: 12345678	-	-	3
4	User submits the form	Application to inform user that he has been successfully authenticated as a staff and provides redirect link to access the management interface	“Logged in as a player. Click here to start to access the management interface” message is displayed.	4-6

Use Case: Logout
Existing Data: - Currently logged in as any user

Step No	Steps	Expected	Result	Use Case Step
1	User starts this test case by logging in a player's account	Login is successful	User is logged in successfully	-
2	User selects the option to logout on the main page	Application to logout user and redirect to login page	Application redirects user to login page	1-4

Use Case: Reset Password				
Existing Data: - Has an existing account				
Step No	Steps	Expected	Result	Use Case Step

1	User loads the application	Application loads successfully	Application load successfully	-
2	User selects the forget password option on the login page	Application redirects user to forget password page	Application redirected user to forget password page	1,2
3	User enters the username of an existing account	-	-	3
4	User submits the form	Application informs user that a reset password email will be sent to the user's inbox if it exists	"A reset password email will be sent to your inbox" message is displayed	4-6
5	User checks his email inbox and opens the link for reset password	Application to redirect user to reset password form	Application redirects the user to reset password form.	7
6	User enters the new password and the confirm password. Password: TestPassword Confirm Password: TestPassword	-	-	8
7	User submits the form	Application to inform user that the password has been reset	"Password has been updated with the new password" message is displayed	8-10

Use Case: Reset Password(AF-S9)

Test Data: - Username that does not already exists				
Step No	Steps	Expected	Result	Use Case Step
1	User loads the application	Application loads successfully	Application load successfully	-
2	User selects the forget password option on the login page	Application redirects user to forget password page	Application redirected user to forget password page	1,2
3	User enters the username of an existing account	-	-	3
4	User submits the form	Application informs user that a reset password email will be sent to the user's inbox if it exists	"A reset password email will be sent to your inbox" message is displayed	4-6
5	User checks his email inbox and opens the link for reset password	Application to redirect user to reset password form	Application redirects the user to reset password form.	7
6	User enters the new password and the confirm password. Password: TestPassword Confirm Password: TestPassword1	-	-	8
7	User submits the form	Application to inform user that password and confirm password is required to match.	"Password and confirm does not match" message is displayed	8-9

Use Case: Social Login				
Test Data: - Facebook account				
Step No	Steps	Expected	Result	Use Case Step
1	User loads the application	Application loads successfully	Application load successfully	-
2	User selects the social login option on the login page	Application redirects user to the facebook login page	Application redirects user to the facebook login page	1,2
3	User enters login details on the Facebook Login Page	Login is successful, Facebook API redirects user back to application login page, player profile is retrieved and application notifies the user that login is successful	Facebook API redirects the user back to the application login page. “Logged in as a player. Click here to start the game” message is displayed.	3-6

Use Case: Social Login(AF-S4)				
Test Data: - Facebook account				
Step No	Steps	Expected	Result	Use Case Step

1	User loads the application	Application loads successfully	Application load successfully	-
2	User selects the social login option on the login page	Application redirects user to the facebook login page	Application redirects user to the facebook login page	1,2
3	User enters login details on the Facebook Login Page but does not use a valid Facebook account	Login is unsuccessful, Facebook API redirects user to login page	User remains at login page	3,4

Use Case: Social Login(AF-S5)				
Test Data: - Facebook account				
Step No	Steps	Expected	Result	Use Case Step
1	User loads the application	Application loads successfully	Application load	-

			successfully	
2	User selects the social login option on the login page	Application redirects user to the facebook login page	Application redirects user to the facebook login page	1,2
3	User enters login details on the Facebook Login Page	Login is successful, Facebook API redirects user back to application login page, player profile is created and application notifies the user that login is successful	Facebook API redirects the user back to the application login page. “Logged in as a player. Click here to start the game” message is displayed.	3-6

Use Case: View Profile				
Test Data: User account that is logged into game				
Step No	Steps	Expected	Result	Use Case Step
1	User loads the application	Application loads successfully	Application load	-

			successfully	
2	User selects the option to view profile	Application retrieves account details and display user information on the view profile UI. Username, Subject name, level and progress for each subject, total score will be displayed.	Application displays profile with the given information.	1,2

Use Case: Move and Interact with Map and NPCs				
Test Data: User account that is logged into game				
Step No	Steps	Expected	Result	Use Case Step
1	User loads the application	Application loads successfully	Application load	-

	and logs in as a player		successfully	
2	User selects start game option on the main screen	User is redirected to main map of the game	User is redirected to main map of the game	1,2
3	User navigates around the school compound in the game and enters a classroom	User is redirected to the classroom map	User is redirected to the classroom map	3-6
4	User gets near to a question NPC and interacts with the question NPC by pressing Spacebar	User is shown the Question Dialog which is handled by the included Answer Question NPC use case	Answer Question NPC use case is executed	7-8
5	After user completed the question NPC usecase, user talks to a leadership NPC.	User is shown the leadership dialog which is handled by the included Talk to Leadership NPC Use Case	Talk to leadership NPC use case is executed.	9-11

Use Case: Move and Interact with Map and NPCs(AF-S3)				
Test Data: User account that is logged into game				
Step No	Steps	Expected	Result	Use Case Step
1	User loads the application	Application loads successfully	Application load	-

	and logs in as a player		successfully	
2	User selects start game option on the main screen	User is redirected to main map of the game	User is redirected to main map of the game	1,2
4	User gets near to a question NPC and interacts with the question NPC by pressing Spacebar	User is shown the Question Dialog which is handled by the included Answer Question NPC use case	Answer Question NPC use case is executed	5-7
5	After the user completes the question NPC use case, the user talks to a leadership NPC.	User is shown the leadership dialog which is handled by the included Talk to Leadership NPC Use Case	Talk to the leadership NPC use case is executed.	9-11

Use Case: Share question				
Test Data: User account that is logged into game				
Step No	Steps	Expected	Result	Use Case Step
1	User loads the application and logs in as a player	Application loads successfully	Application load successfully	-

2	User selects the sharing option	User is redirected to Facebook API	User is redirected to Facebook API	1-2
3	User will be prompted to share the content and user clicks it to submit the form	Facebook API Actor will prompt user to share the content and form will be ready for user to click	Facebook API Actor will prompt user to share the content and form is submitted	3-4

Use Case: Talk to Leadership NPC				
Test Data: User account that is logged into game				
Step No	Steps	Expected	Result	Use Case Step
1	User loads the application and logs in as a player	Application loads successfully	Application load successfully	-

2	User gets near to a Leadership NPC and interacts with the Leadership NPC by pressing Spacebar	User is shown the Leadership information for the subject or globally	User is shown the Leadership information for the subject or globally	1-2
3	User is done viewing and closes the dialog option	The dialog option closes	The dialog option closes	3

Use Case: Answer Question NPC				
Test Data: User account that is logged into game				
Step No	Steps	Expected	Result	Use Case Step
1	User loads the application and logs in as a player	Application loads successfully	Application load successfully	-

2	User gets near to a question NPC and interacts with the question NPC by pressing Spacebar	User is shown the Question Dialog which was initialized during map initialization.	Question Dialog comes out	1,2
3	User will make his selection and the application will inform the user if the option he has chosen is right	User will make his selection and the application will inform the user if his selection is right	User will make his selection and the application will inform the user if his selection is right	3-4
4	The application will display the explanation why the selection is right or wrong and an option to end the question will appear	The application will display the information about the right/ wrong answer with an option for user to closer the dialog box	The application will display the information about the right/ wrong answer with an option for user to closer the dialog box	5-6
5	User will click the option to close the dialog box	The application will close the dialog box, update the user progress and update the records	The application will close the dialog box, update the user progress and update the records	7-9

Objectives

The objectives for the development and testing strategy for our subsystems is laid out below:

- Achieve functional and nonfunctional requirements laid out in the Software Requirement Specification.
- Develop the subsystems while following SOLID principles
    - Single Responsibility Principle
    - Open Closed Principle
    - Liskov Substitution Principle
    - Interface Segregation Principle
    - Dependency Inversion Principle

- Achieve the following Software Quality Attributes:
    - Testability
    - Reliability
    - Correctness

The following sections will explain the design of our software components, consideration of SOLID principles that we have considered.

Design of software components

We will be making use of a Test Driven Development approach where unit tests are written before the actual development code is written, changes are later then made to ensure that the code is well designed. This cycle is also known as the “Red-Green-Refactor” cycle in the Test Driven Development community.

The practice of writing tests before writing code indirectly helps to guide the **Single Responsibility Principle**, as the length of the test often reflects the number of responsibilities of the code. The number of dependencies that have to be stubbed in a test can also be bad code design that do not follow the **Dependency Inversion Principle**.

As such, throughout this project we will be using tests to help ensure functionality and good design. The following subsections are different explanations for the respective subsystem and their reason for the design.

Subsystem Background

Management Interface

This is a web interface that is built using React and Redux Single Page Application Framework.
<Insert flux architecture here>
A particularly unique point about React and Redux is that the data flows in a single direction. Because of this, by writing unit tests for the different checkpoints of the data flow, one can easily ensure functionality and isolate problems. Another point of the framework

architecture that is worth mentioning is the link between each component is dependency injected(**Dependency Inversion Principle**). As a result, tests for this can be focused on the key functionality and not be bothered by retrieving or finding the required dependencies. Without the use of dependency injection, mocking would have to be done at a lower level instead of in code. (ie mock datastore in code by providing a memory based datastore instead of setting up a whole replica and test instance). This interface is also written in Typescript(See below)

API Server

This is a network server that will be built using Node.JS and Typescript. The purpose of this setup is for 2 reasons: Node.JS is a suitable and simple enough http server that is easy to understand and gets the things done.Node.JS is usually written in Javascript but we have chosen to write in Typescript due to its static typing capability. This helps to reduce most issues with incompatible data types and eliminate possible development mistaekes such as syntax errors. The use of Typescript ensures that the code, once able to compile, has a certain level of correctness. The use of Javascript however does not guarantee that until run time.

Game Client

This is a Game Client that will be built using C# and Unity. Unity has been a popular choice among amateur game developers and has significant documentation available. This can help in maintainability of the application as new developers can pick it up easily. It also offers some levels of testing capabilities within the game runtime.

Approach to Testing

Throughout our development lifecycle, we will testing the following these 2 principles:

1. Test only what others can see
2. Test expected interactions among components

The reasons for testing using the above 2 principles can be thought of as complementary:

- Test only what others can see
    - Internal methods is blackbox
- Test expected interactions among components

- White box: when there is multiple conditional statements in a block of code, tests are written to ensure that the different branches of code is executed as expected
- Blackbox: Test if the functionality is correct

The following sections will explain the design of our tests for the different subsystems.

Management Interface

Due to the functional nature of its framework architecture, we will be ensuring functions return values as expected.This is because even if mutations are called in code, they will not be respected unless returned. A variation of principle #2 is applied here, based on given an initial set of parameters, we check if the values returned are as expected.

Game Client

One constraint with the game client implementation is that it is difficult to test game interactions without writing something close to an integration test. At the point of this writing, these are called Playmode Tests which are heavily dependent on the game runtime.We will be using these integration-like tests as an Integration test and we will test the control logic using written unit tests.

API Server

The API Server contains a layer of request delegation which sends the request to a core business logic set of methods. Unit tests will be written for this set of methods. The data persistence layer will be replaced by an in-memory database server to ensure that tests run as quickly as possible.