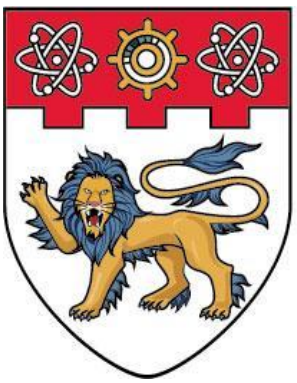


NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

SC2006 Final Report

Channel Five

Group members:

Cai Kai Hang (U2121031J)

Denzyl David Peh (U2122190F)

Hoo Jian Le (U2120301C)

Lau Yong Jie (U2121879D)

Tay Jia Yi (U2123090A)

Table of Contents

1. Product Description

1.1. Purpose of system

This document specifies a simplified WebUI functionality allowing Pre-University students to make better judgement when choosing their school, course, and subsequently a career path in the future with the information provided by our API.

1.2. Scope of system

The WebUI allow users to :

- Select different university to display all available courses from the university
- To display the salary and details of course.
- Register an account to utilise the chat and create function
- Join discussion conversation to allow users to ask more questions with existing university students

The API from the WebUI will be updated with accurate data.

1.3. Users and Stakeholders

This document is created for:

- Pre-university students
- Existing students from different university

1.4. Assumptions & Constraints

Assumptions :

- Users will always be able to retrieve the latest information on course and salary details
- Users will be able to use our system at all times except for scheduled maintenance.
- Users will be able to navigate easily in our system.

Constraints :

- The amount of space needed in the database is a constraint as the system will always receive new updates on new course and salary details
- The course and salary API must be up to date with the latest information. As such, we need to always look out for new and accurate information.
- Users might abuse the use of the discussion forum to post malicious content.

2. Functional Requirements

2.1. Atomization of functional requirements

Below shows the Atomization of our functional requirements for the system:

(USER LOGIN)

1. User shall be able to login into the system

1.1. Login requires user to input credentials

1.1.1 The system shall authenticate the input credentials

1.1.1.1 The system shall redirect to the homepage if the credentials are true.

1.1.1.2 The system shall redirect to the login page if the credentials are false.

1.2. User shall be able to reset their password of their account

1.2.1. The system will send a forget password form to input the email address.

1.2.1.1. The system shall authenticate the input credentials.

1.2.1.1.1. The system shall send an email containing temporary password link.

1.2.1.1.1.1. The system shall save the updated password in database.

1.2.1.1.1.1.1. The system will redirect back to login page once saved.

1.2.1.1.2. The system will redirect to the forgot password page.

(REGISTER AN ACCOUNT)

2. User shall be able to create an account

2.1. User shall be able to access the account creation page from the login page

2.1.1. Registering an account requires user to input email address and password.

2.1.1.1. The system shall be able to check for existing email and password.

2.1.1.1.1. The system shall send an verification email if the email and password provided is valid.

2.1.1.1.1.1. The system will redirect back to the loginpage once email has been verified.

2.1.1.1.2. The system will remain in the registerpage if the email and password provided is invalid.

(DISPLAY INFORMATION)

3. User shall be able to query the system for information.

3.1 User shall be able to select school directly.

3.1.1. The system shall be able to display all courses available for the school selected.

3.1.1.1. The user shall be able to select course from the list of courses.

3.1.1.1.1. The system must be able to retrieve salary and course details from the API for the selected course.

3.2. User shall be able to select view courses directly.

3.2.1. The system shall be able to display all courses available from all schools.

3.2.1.1. The user shall be able to select course from the list of courses.

3.2.1.1.1. The system must be able to retrieve salary and course details from the API for the selected course.

3.2.1.2. The user shall be able to search for course in the search bar.

3.2.1.2.1. The system shall be able to display the search result.

3.2.1.2.1.1. The system must be able to retrieve salary and course details from the API for the selected course.

(DISCUSSION FORUM)

4. User shall be able to view the discussion forum

4.1 User shall be able to create a discussion thread.

4.1.1. User must be able to login in order to create.

4.1.1.1. Once login successfully, User will be able to create a discussion thread.

4.2. User shall be able to select a discussion to view.

4.2.1. The system shall be able to display all the chat for the selected discussion.

4.2.1.1.User shall be able to post a message.

4.2.1.1.1. User must be able to login in order to post a message.

4.2.1.1.1.1 Once the login is successful, User will be able to post a message.

4.3 User shall be able to search for discussion threads in the search bar.

4.3.1. The system shall be able to display the search result.

4.3.1.1. User shall be able to post a message.

4.3.1.1.1. User must be able to login in order to post a message.

4.3.1.1.1.1. Once the login is successful, User will be able to post a message.

(ACCESS PROFILE)

5. User shall be able access the profile page.

5.1. User shall be able to change password.

5.1.1. The system must be able to save the updated password in the database.

2.2. Use Case Descriptions

Below show all the use case descriptions for the system:

Use Case Name: Register
Participating Actors: Guest, Email Server
Pre-Condition: ● Guest is not logged in.
Flow of Events: 1. Guest clicks 'Register' on the Login page 2. EasyUni responds by presenting a register form to the user. *The form includes email and password field for the user to complete.* 3. Guest complete the register form with their email address and password. 4. EasyUni verifies that the email address entered has not been used before. 5. Email Server sends out a verification email to the email address provided.
Post-Condition: ● Guest entered an invalid email address. ● Guest successfully registered.
Alternate flows: AF-S4: EasyUni detects that the email address has already been registered as a user 1. EasyUni displays the message "Email address already in use!" for 2 seconds 2. Return to Step 2
Exceptions: Ex 1: Email Server fails to send the verification email due to server/network issues 1. EasyUni displays the message "Error in sending verification email!" for 2 seconds 2. Return to Step 2 Ex 2: System failed to capture the details into database 1. Prompt user the error "something went wrong. Please try again later!" 2. Return to Step 2.

Use Case Name: Login
Participating Actors: Guest
Pre-Condition: ● Guest is not logged in.
Flow of events: 1. Guest clicks ‘Login’ on the home page. 2. EasyUni responds by presenting a login form to the user. *The form includes email and password field for the user to complete.* 3. Guest complete the login form with their email address and password. 4. EasyUni authenticate if the email address and password matches the account details 5. Guest successfully logged in to their account.
Post-Condition: ● Guest entered an invalid email address. ● Guest entered an invalid password. ● Guest successfully logged in.
Alternate flows: AF-S4: EasyUni detects that the email address used is not registered as a user 1. EasyUni displays the message “Invalid user!” for 2 seconds 2. Return to Step 2 AF-S4: EasyUni detects that the input password does not match the one in the database 1. EasyUni displays the message “Invalid password!” for 2 seconds 2. Return to Step 2
Exceptions: Ex 1: User has failed to login via the same email 3 times in a row within the span of 5 minutes 1. EasyUni times out the associated account under that email - access is locked even with the correct password 2. Email Server sends a Password Reset email to the email account

Use Case Name: Reset Password
Participating Actors: Guest, Email Server
Pre-Condition: ● Guest is not logged in.
Flow of events: 1. Guest clicks ‘Reset Password’ on the login page. 2. EasyUni responds by presenting a reset password form to the user. *The form includes email field for the user to complete.* 3. Guest complete the reset password form with their email address. 4. EasyUni verifies if the email address exists in the system. 5. Email Server sends out an email with detailed instructions on how to reset their password to the email address provided.
Post-Condition: ● Guest entered an invalid email address. ● Email Server successfully sent out the email.
Alternate flows: AF-S4: Invaild email address indicated in the reset password form 1. EasyUni Displays the message ” invalid email ! ” to the user for 2 seconds. 2. Return to step 3.
Exceptions: *Ex 1 :The user fails to complete the reset password form in one hour.* *1. Reset password form will become invalid, will not reset password.* *2. Users will have to request another reset password form from EasyUni.*

Use Case Name: Search Courses
Participating Actors: Guest
Pre-Condition:
Flow of events: 1. Guest type their search query onto the search bar on the website. 2. Guest click on the search button. 3. EasyUni responds by presenting a list of courses to the user matching to the search query. *Each of the list items includes course name and school name.*
Post-Condition: ● Search results have successfully displayed.
Alternate flows: AF-S4: The guest types into the search query a course that is not found in any university 1. EasyUni will display “No such courses found!” 2. Return to step 1
Exceptions:

Use Case Name: View Courses
Participating Actors: Guest, API
Pre-Condition:
Flow of events: 1. Guest clicks ‘View Courses’ on the website. 2. EasyUni displays all available courses from api. 3. Guest selected one course to view. 4. EasyUni requests course and salary details from api. 5. Api provides course and salary data to the user.
Post-Condition: ● Course and salary data successfully provided back to the user
Alternate flows:
Exceptions: Ex 1: API fails to respond 1. EasyUni will display “salary not found” on the salary data column instead.

Use Case Name: View Course
Participating Actors: Guest,API
Pre-Condition: ● Guest must have selected the course they wish to view
Flow of events: 1. Guest select the course they wish to view on the website. 2. EasyUni requests salary data from API. 3. GOV.SG API responds by providing salary data for the course. 4. EasyUni responds by redirecting guest to the selected course page. *The page contains all course details such as; course name, course IGP, school name, employment rate over the years (shown as a chart) and median salary over the years (shown as a chart)*
Post-Condition: ● All course details have been successfully displayed.
Alternate flows:
Exceptions: Ex 1: API fails to respond 1. EasyUni will display “Salary not found” on the salary data column instead.

Use Case Name: Edit Profile
Participating Actors: Registered User
Pre-Condition: ● Registered User is on their profile page
Flow of Events: 1. Registered User clicks ‘profile’ on the home page. 2. EasyUni responds by presenting a profile form to the user. *The form is pre-filled in with the user’s details that were set before.* 3. Registered User updates the details in the form 4. Registered User clicks on ‘save profile’ on the form
Post-Condition: ● Registered User has successfully updated their profile ● Registered User chooses to cancel and undo any edits made
Alternate flows: AF-S1: User attempts to leave the page without saving 1. EasyUni will prompt the user “Are you sure? Any changes made will not be saved.” to the user. 2. Return to step 2.
Exceptions: Ex 1: System failed to capture the details into database 1. EasyUni will prompt user the error “something went wrong. Please try again later!” 2. Return to Step 2.

Use Case Name: View Discussions
Participating Actors: Guest
Pre-Condition:
Flow of events: 1. Guest clicks ‘Discussions’ on the home page. 2. EasyUni responds by presenting the list of discussions to the user. *The list items include; discussion name, creator name, date, number of views and number of replies.*
Post-Condition: ● All discussions in the system have been successfully displayed.
Alternate flows:
Exceptions: Ex 1: There are no discussions in the list 1. EasyUni will display “There are no discussions, would you like to create one?” with a ‘new post’ button instead.

Use Case Name: New Discussion Topic
Participating Actors: Registered User
Pre-Condition: ● Registered User is in the Discussion page
Flow of Events: 1. Registered User clicks on ‘create new discussion’ on the discussion page. 2. EasyUni responds by presenting a textbox and a title prompt for the user to fill in. *Apart from those features, the Topic title, Registered User name and Current date is displayed as well.* 3. Registered User enters their discussion content into the textbox and fills in the title 4. Registered User clicks on ‘Submit’
Post-Condition: ● Registered User has successfully created a new Discussion post ● Registered User cancels the creation of the new Discussion post
Alternate flows: AF-S1: User attempts to leave the page without saving 1. EasyUni will prompt the user “Are you sure? Any changes made will not be saved.” to the user.
Exceptions: Ex 1: System failed to capture the details into database *1.* EasyUni will prompt user the error “Something went wrong. Please try again later!” 2. Return to step 2

Use Case Name: Reply to Discussion
Participating Actors: Registered User
Pre-Condition: ● Registered User is viewing a Discussion topic posted in the Discussion page
Flow of Events: 1. Registered User clicks on ‘Reply’ on the discussion page. 2. EasyUni responds by presenting a textbox for the user to fill in. *Apart from the textbox, the Topic title, Registered User name and Current date is displayed as well.* 3. Registered User fills in the textbox with their reply to the topic post 4. Registered User clicks on ‘Submit’
Post-Condition: ● Registered User has successfully created a new reply ● Registered User cancels the creation of the new reply
Alternate flows: AF-S1: User attempts to leave the page without saving 1. EasyUni will prompt the user “Are you sure? Any changes made will not be saved.” to the user.
Exceptions: Ex 1: System failed to save response from the Registered Users 1. EasyUni will display the message “Failed to submit,please try again” in 2 seconds.

Use Case Name: Send Verification Email
Participating Actors: Email Server
Pre-Condition: ● This use case extends the Register use case, which is initiated by the Guest registering for a new account ● Email provided by the guest is verified by EasyUni to have no existing account tied to it
Flow of Events: 1. Email Server sends an email to the one specified by the guest via the register form
Post-Condition: ● Verification email has been successfully sent out
Alternate flows:
Exceptions: EX1: Guest email is unable to receive verification email (for several possible reasons - invalid email address, email is full etc.) 1. EmailServer sends an error log to EasyUni indicating that it was unable to send the email 2. EasyUni displays the message “Unable to send verification email to <insert_guest_email_here>, please check if it is a valid and working email account.” for 5 seconds

Use Case Name: Send Reset Password Email
Participating Actors: Email Server
Pre-Condition: ● This use case extends the Reset Password use case, which is initiated by the Guest upon clicking 'Reset Password' ● Guest has filled out the reset password form with their email and that has been verified by EasyUni
Flow of Events: 1. Email Server sends out an email with detailed instructions on how to reset their password to the email address provided.
Post-Condition: ● Reset password email has been successfully sent out
Alternate flows:
Exceptions: EX1: Guest email is unable to receive reset password email (for several possible reasons - invalid email address, email is full etc.) 1. EmailServer sends an error log to EasyUni indicating that it was unable to send the email 2. EasyUni displays the message "Unable to send reset password email to <insert_guest_email_here>, please check if it is a valid and working email account." for 5 seconds

Use Case Name: Provide Salary Data
Participating Actors: API
Pre-Condition: ● This use case extends the View Courses and View Course use cases. ● Initiated whenever the Guest wishes to access and view salary data
Flow of Events: 1. EasyUni request salary data from GOV.SG API. 2. GOV.SG API responds by providing salary data for the course.
Post-Condition: ● GOV.SG API successfully returns the salary data
Alternate flows:
Exceptions: EX1: GOV.SG API is unable to properly return the salary data 1. EasyUni detects that no data or corrupted data has been received. 2. EasyUni displays an error message that says "Unable to receive salary data" for 3 seconds. 3. EasyUni will display "Salary not found" on the salary data column instead.

2.3. Full Class Diagram

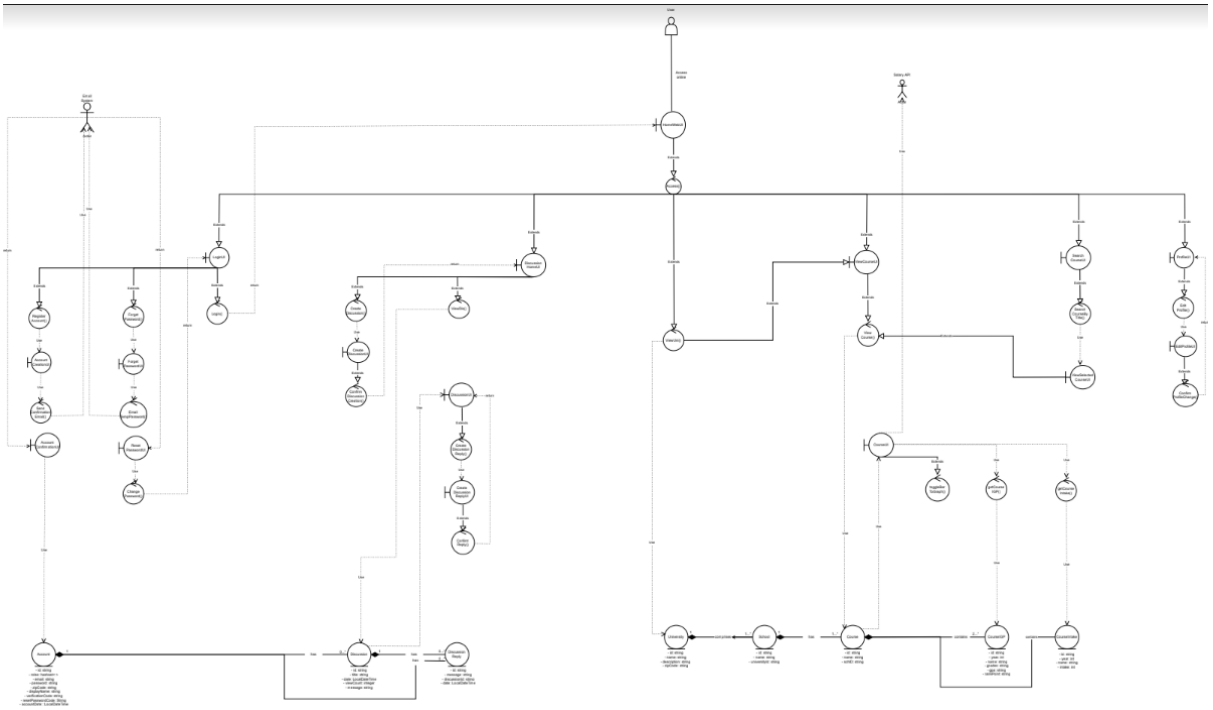


Fig 1. Full Class Diagram

2.4. Sequence Diagram

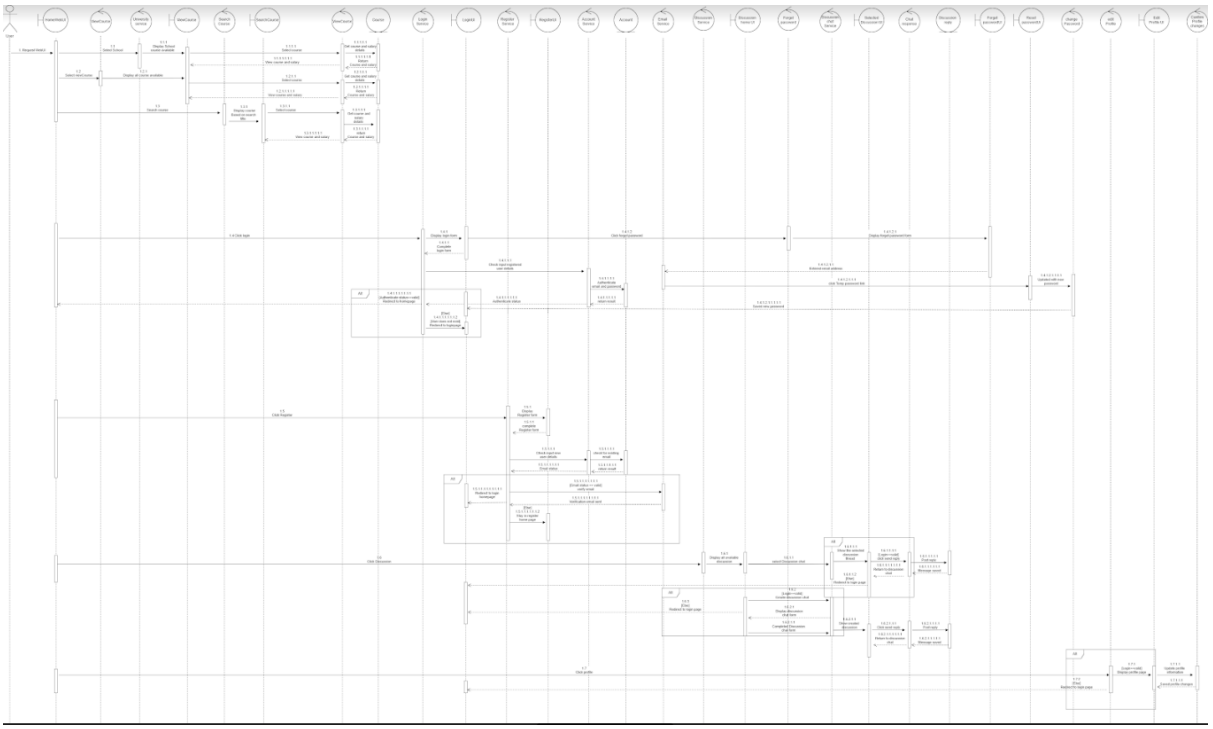


Fig 2. Full Sequence Diagram

2.5. State Machine Diagram

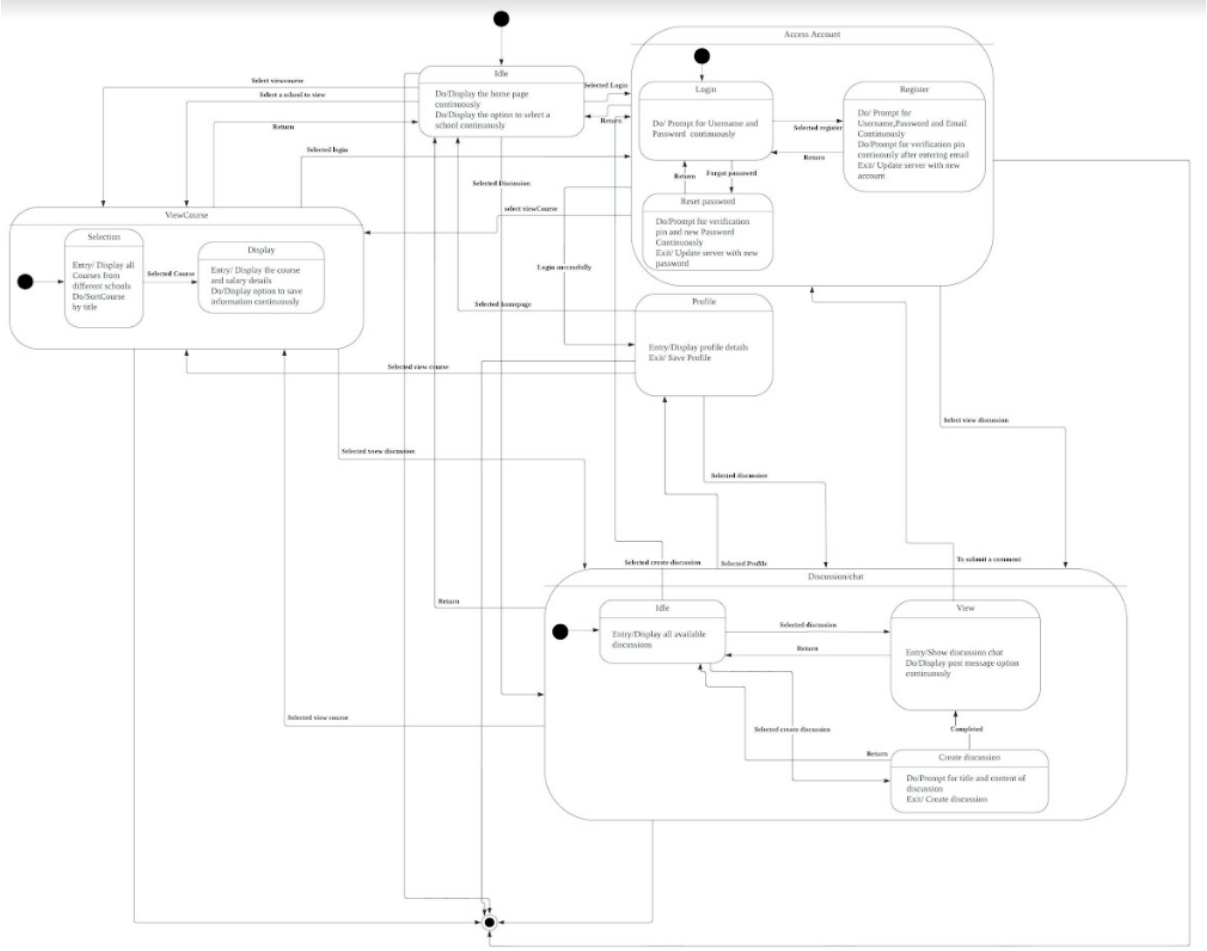


Fig 3. Full State Machine Diagram

2.6. Test cases

Black box testing

1. Register

Test id	Input description	Expected output	Actual output
1	Attempt registration with valid email, password, and display name	System alerts user to check email for verification, and sends verification email.	System alerts user to check email for verification, and sends verification email.
2 (a-g)	Attempt registration with invalid email, password or display name	System displays error alert, with guiding messages for invalid fields	System displays error alert, with guiding messages for invalid fields

Test id	Inputs			Expected output	Actual output
	Email	Password	Display name		
1	“test@gmail.com”	“aP2)0n%W”	“John”	Verification email sent	Verification email sent
2a	“invalid”	“aP2)0n%W”	“John”	Invalid email	Invalid email
2b	“test@gmail.com”	“aP2)0n%”	“John”	Invalid password (must have at least 8 characters)	Invalid password
2c	“test@gmail.com”	“p@5sword”	“John”	Invalid password (must contain an uppercase)	Invalid password
2d	“test@gmail.com”	“P@5SWORD”	“John”	Invalid password (must contain a lowercase)	Invalid password
2e	“test@gmail.com”	“P@ssword”	“John”	Invalid password (must contain a number)	Invalid password
2f	“test@gmail.com”	“P4ssword”	“John”	Invalid password (must contain a special character)	Invalid password
2g	“test@gmail.com”	“aP2)0n%W”	“”	Invalid display name	Invalid display name

2. Login

Test id	Input description	Expected output	Actual output
1	Attempt login with valid email and password	Login successful, redirect user to home page	Login successful, redirect user to home page
2 (a-b)	Attempt login with invalid email or password	System displays error alert, User is not logged in	System displays error alert, User is not logged in

Test id	Inputs		Expected output	Actual output
	Email	Password		
1	“test@gmail.com”	“aP2)0n%W”	Login successful	Login successful
2a	“wrong@gmail.com”	“aP2)0n%W”	Wrong email/password	Wrong email/password
2b	“test@gmail.com”	“wrongpass”	Wrong email/password	Wrong email/password

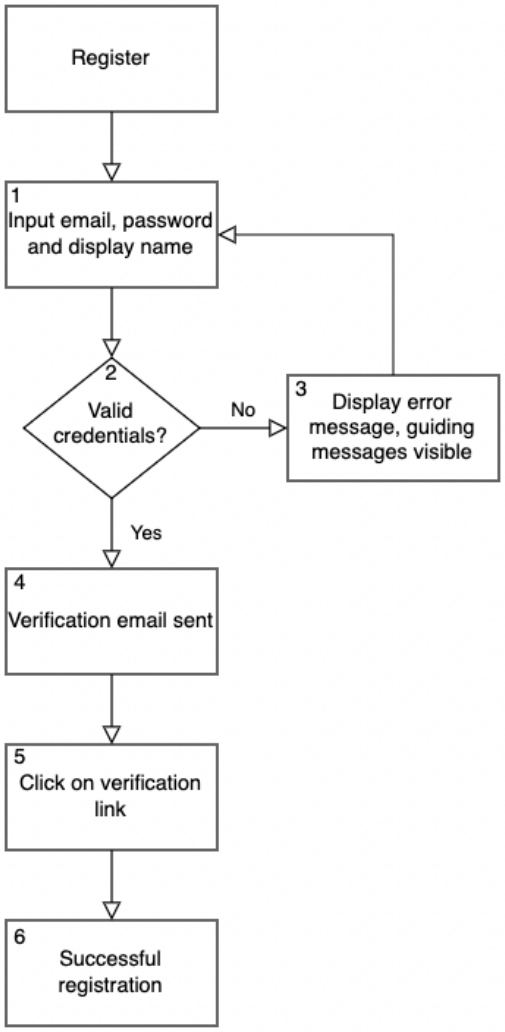
3. Profile / Forget Password

Test id	Input description	Expected output	Actual output
1a	Change password to valid password by clicking “forgot password?” button	Password has been changed, redirect user to home page	Password has been changed, redirect user to home page
1 (b-f)	Change password to invalid password by clicking “forgot password?” button	System displays error alert	System displays error alert
2a	Change password to valid password in profile	Password has been changed, redirect user to home page	Password has been changed, redirect user to home page
2 (b-f)	Change password to invalid password in profile	System displays error alert	System displays error alert
3a	Change display name to valid display name in profile	Display name has been changed, redirect user to home page	Display name has been changed, redirect user to home page
3b	Change display name to invalid display name in profile	System displays error alert	System displays error alert

Test id	Inputs		Expected output	Actual output
	New Password	Display name		
1a	“aP2)0n%W”	-	Password successfully changed	Password successfully changed
1b	“aP2)0n%”	-	Invalid password (8 characters)	Invalid password
1c	“p@5sword”	-	Invalid password (no uppercase)	Invalid password
1d	“P@5SWORD”	-	Invalid password (no lowercase)	Invalid password
1e	“P@ssword”	-	Invalid password (no number)	Invalid password
1f	“P4ssword”	-	Invalid password (no special character)	Invalid password
2a	“aP2)0n%W”	-	Password successfully changed	Password successfully changed
2b	“aP2)0n%”	-	Invalid password (8 characters)	Invalid password
2c	“p@5sword”	-	Invalid password (no uppercase)	Invalid password
2d	“P@5SWORD”	-	Invalid password (no lowercase)	Invalid password
2e	“P@ssword”	-	Invalid password (no number)	Invalid password
2f	“P4ssword”	-	Invalid password (no special character)	Invalid password
3a	-	“Bob”	Display name successfully changed	Display name successfully changed
3b	-	“”	Invalid display name	Invalid display name

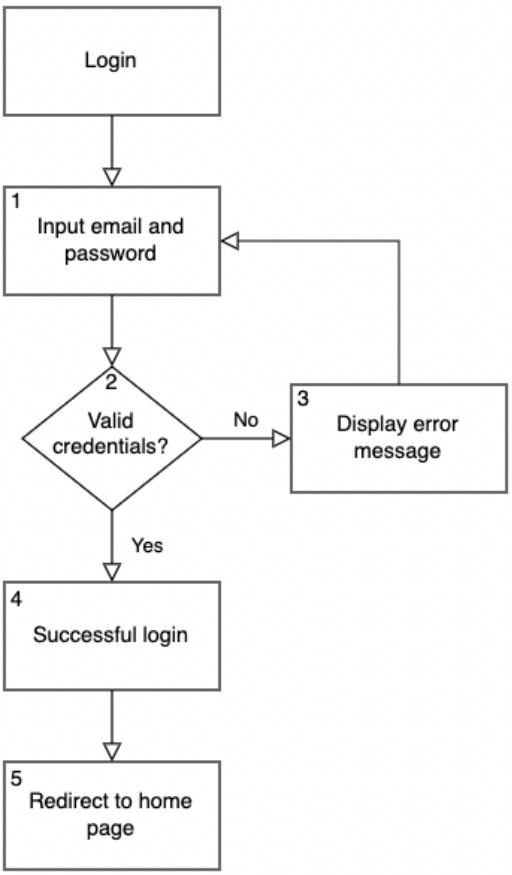
White Box Testing

1. Register



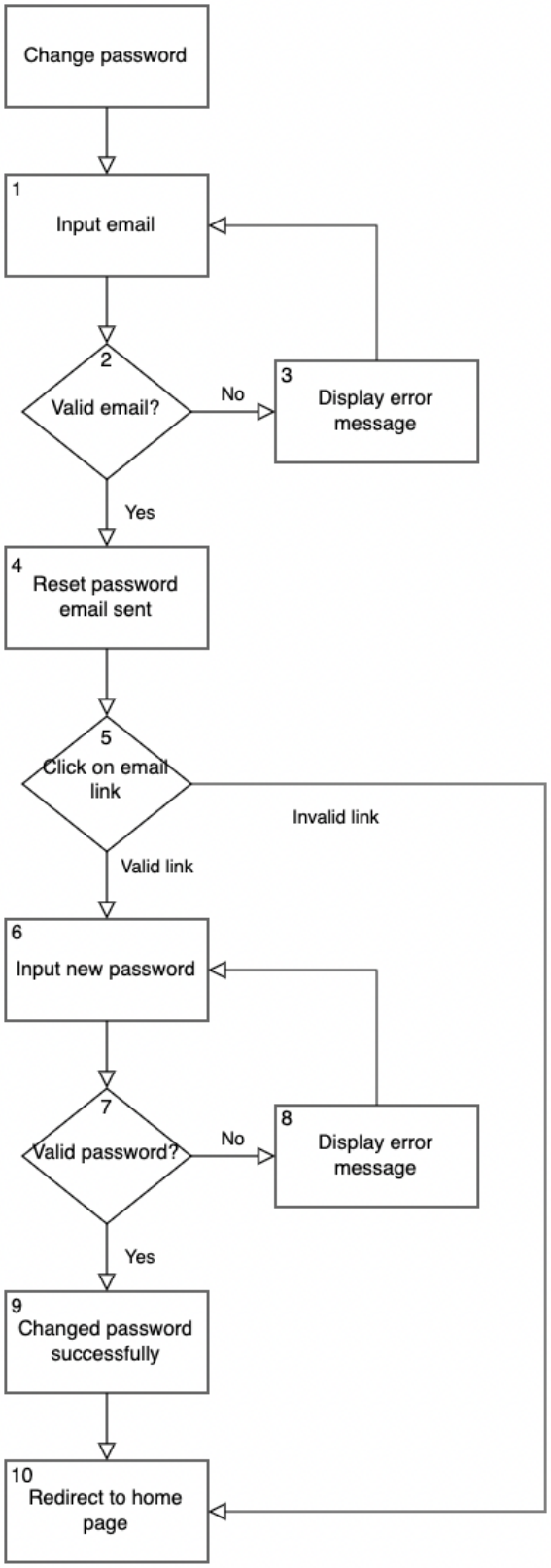
Test id	Basis path	Inputs	Expected output	Actual output
1	1, 2, 3, 1, 2, 4, 5, 6	1: “invalid”, “aP2)0n%W”, “John” 1: “test@gmail.com”, “aP2)0n%W”, “John”	System displays error message and guiding messages with invalid email, then registration successful on second try	System displays error message and guiding messages with invalid email, then registration successful on second try

2. Login



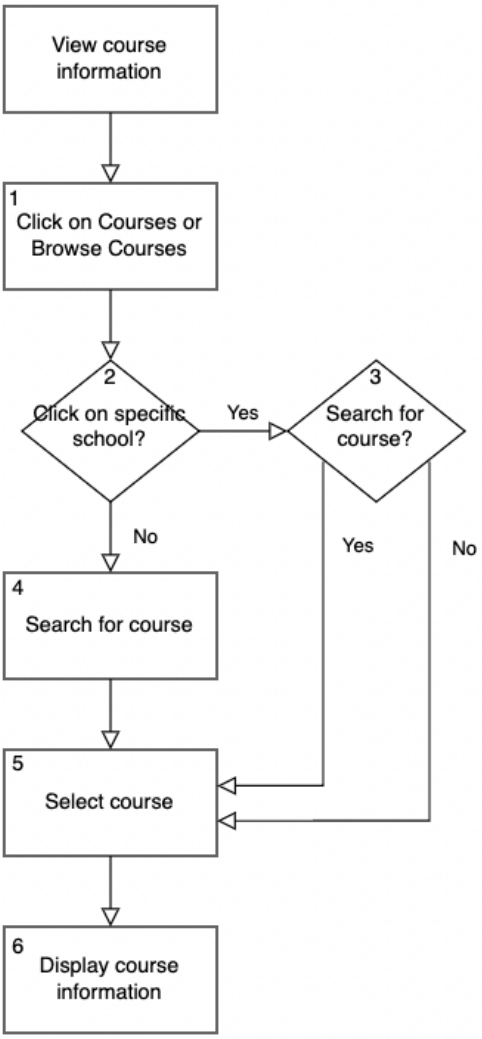
Test id	Basis path	Inputs	Expected output	Actual output
1	1, 2, 3, 1, 2, 4, 5	1: “test@gmail.com”, “wrongpass” 1: “test@gmail.com”, “aP2)0n%W”	System displays error message with wrong password, then successful login on second try. User is redirected to home page	System displays error message with wrong password, then successful login on second try. User is redirected to home page

3. Change Password



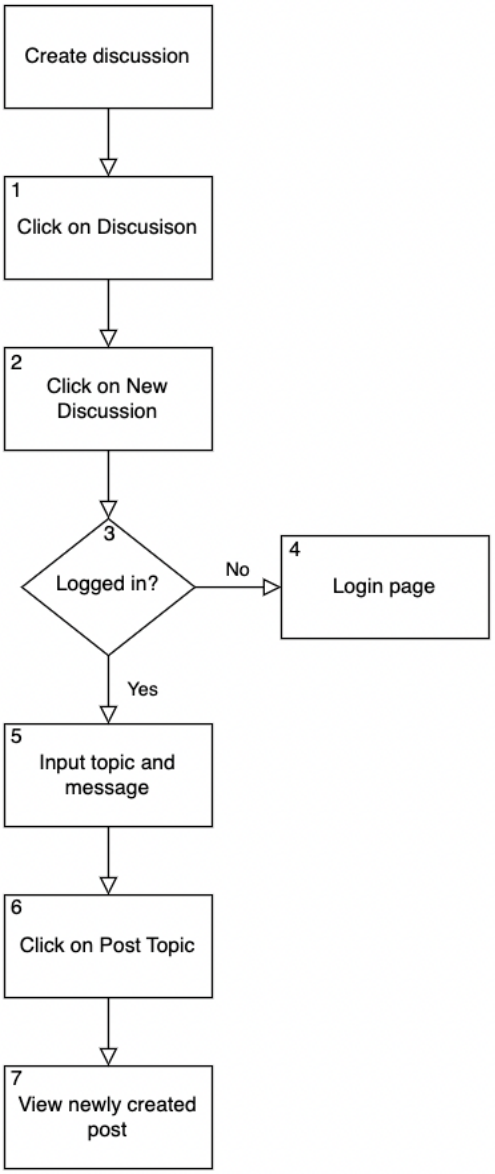
Test id	Basis path	Inputs	Expected output	Actual output
1	1, 2, 3, 1, 2, 4, 5, 10	1: “unregistered@gmail.com” 1: “test@gmail.com” 5: click on link after expired	System displays error message with unregistered email, then sends email on second try. Clicking expired link redirects user to home page. User is not logged in	System displays error message with unregistered email, then sends email on second try. Clicking expired link redirects user to home page. User is not logged in
2	1, 2, 4, 5, 6, 7, 8, 6, 7, 9, 10	1: “test@gmail.com” 5: click on valid link 6: “invalid” 6: “aP2)0n%W”	System sends email, link directs user to reset password form. System displays error message on invalid password, then password is changed successfully on second try. User is redirected to home page	System sends email, link directs user to reset password form. System displays error message on invalid password, then password is changed successfully on second try. User is redirected to home page

4. View Course Information



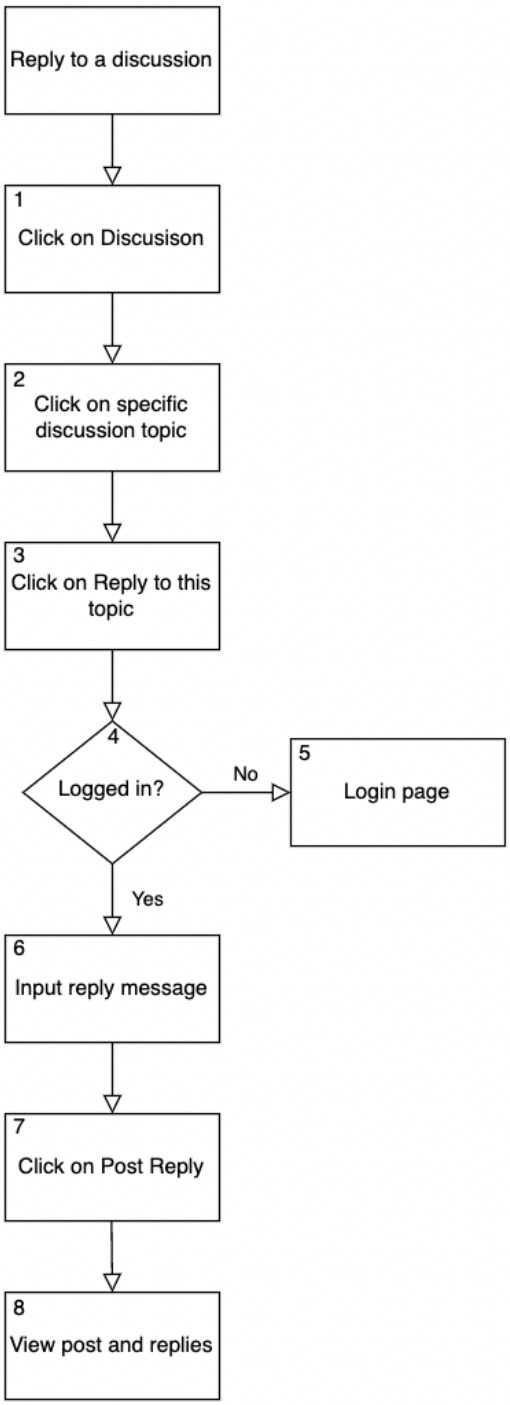
Test id	Basis path	Inputs	Expected output	Actual output
1	1, 2, 4, 5, 6	4: “computer science” 5: Click on Computer Science (NTU)	User has navigated to correct page, which displays relevant information to the course	User has navigated to correct page, which displays relevant information to the course
2	1, 2, 3 (yes), 5, 6	2: Click on NTU 3: “computer engineering” 5: Click on Computer Engineering (NTU)	System directed user to browse NTU courses. User has navigated to correct page, which displays relevant information to the course	System directed user to browse NTU courses. User has navigated to correct page, which displays relevant information to the course
3	1, 2, 3 (no), 5, 6	2: Click on NUS 5: Click on Bachelor of Computing (Computer Science)	System directed user to browse NUS courses. User has navigated to correct page, which displays relevant information to the course	System directed user to browse NUS courses. User has navigated to correct page, which displays relevant information to the course

5. Create a New Discussion



Test id	Basis path	Inputs	Expected output	Actual output
1	1, 2, 3, 5, 6, 7	0: User is logged in 5: “NTU vs NUS”, “testing”	Discussion with topic “NTU vs NUS” and message “testing” has been created. There are no replies.	Discussion with topic “NTU vs NUS” and message “testing” has been created. There are no replies.
2	1, 2, 3, 4	0: User is not logged in	User is redirected to the login page	User is redirected to the login page

6. Reply to a Discussion



Test id	Basis path	Inputs	Expected output	Actual output
1	1, 2, 3, 4, 6, 7, 8	2: Click on “NTU vs NUS” discussion 4: “replying”	User is able to view discussion topic and message, as well as other replies. Once posted, reply is visible on the discussion page	User is able to view discussion topic and message, as well as other replies. Once posted, reply is visible on the discussion page
2	1, 2, 3, 4, 5	0: User is not logged in	User is redirected to the login page	User is redirected to the login page

3. Non-Functional Requirements

Security:

1. Authentication for login, user password must be encrypted.

Reliability:

1. System must be online and ready for use at all times of the day, except for scheduled maintenance.
2. Information provided must be updated to at most one year's recency.
3. System must be able to restart and update without losing data.

Performance:

1. System should respond to any request given by the user within 0.5 seconds.
2. System should be able to display information in several languages accurately.

Maintainability:

1. Write clean code and follow Object-Oriented principles so that any changes to the code will not break the whole system.
2. The system must be able to debug
3. If system is down, the system should be under maintenance for no more than 3 hours

Scalability:

1. Able to add functions to the website easily with minimal changes to code.

Usability: Friendliness

1. Any user who knows how to read and knows what they are searching for should be able to use it without the user guide.

4. Interface Requirements

4.1. User Interfaces

EasyUni interface should be intuitive, such that 99.9% of all new EasyUni users can use the app without any assistance.

The UI shown in this project is divided into sub parts which are:

- Upper horizontal bar which includes login, view courses, discussion and profile.
- Login allows guests to login into their account.
- The Register UI is located in the login page for user that don't have an account.
- The forget password is also located in the login page.
- View University UI will be shown on the homepage for user selection.
- View Courses UI allows user to look for courses.
- A search bar is also available in the View Course page for users to look for the courses they want.
- Discussion form is available for users to create a thread to ask questions.
- Any user that has logged in can post a message in the thread.

4.2. Hardware Interfaces

No hardware interfaces are needed for this system.

4.3. Software Interfaces

The following are the software used for the EasyUni website:

Software used:	Description:
Back-end	MongoDB for database, Springboot and Maven to build the RESTful API server.
Front-end	Vuejs, bootstrap and javascript as HTML templates.
Courses	To provide the salary information, we have decided to use a government API https://data.gov.sg/
HTTP Server	For hosting of our web application, we have decided to use Tomcat Apache.

5. Data Dictionary

Term	Definition
Guest	Guests are users that have not logged in. All users are guests by default, until they log in. These users have limited access to features.
Student	Students are users that have logged in. These users have access to more features, in addition to the guest-level features.
Discussion Forum	The discussion forum is a platform for students to ask and answer questions and discuss topics. Guests may only view the discussion forums.
Discussion Reply	A comment that extends from the original discussion
Profile	The profile is a personalized page for students to save information that they deem useful or relevant.
Email	Email refers to either school email or personal email.
IGP (Indicative Grade Profile)	A grade measure of a course for pre-university students to check whether they are eligible to enter
Course intake	Number of students enrolled into a certain course for that year

Appendix A: Meeting Minutes

MINUTES OF MEETING I

Date	:	6 September 2022	
Time	:	1400 - 1430	
Venue	:	Zoom	
Reference No.	:	1	
Present	:	Team Leader	Lau Yong Jie
		Assistant Leader	Tay Jia Yi
		Members	Cai Kai Hang
			Denzyl David
			Hoo Jian Le
Absent with apologies	:		
Minutes by	:	Hoo Jian Le	
Vetted by	:	Lau Yong Jie	
Signed by	:	Lau Yong Jie	

Yong Jie

Led the team in the discussion for the submission of Lab 1 deliverables

Worked on the creation of UI mockups (wireframes)

Created an example class diagram to be improved upon

Jia Yi

Worked on the creation of use cases

Worked on the data dictionary

Kai Hang

Worked on the creation of use cases

Worked on the data dictionary

Denzyl

Worked on the atomization of functional requirements

Consolidated the functional & non-functional requirements

Jian Le

Worked on the atomization of functional requirements

Consolidated the functional & non-functional requirements

Action

MINUTES OF MEETING II

Date	:	20 September 2022	
Time	:	1400 - 1430	
Venue	:	Zoom	
Reference No.	:	2	
Present	:	Team Leader	Lau Yong Jie
		Assistant Leader	Tay Jia Yi
		Members	Cai Kai Hang
			Denzyl David
			Hoo Jian Le
Absent with apologies	:		
Minutes by	:	Hoo Jian Le	
Vetted by	:	Lau Yong Jie	
Signed by	:	Lau Yong Jie	

Yong Jie Led the team in the discussion for submission of Lab 2 deliverables Created the stereotype class diagram to be further built upon Worked on sequence diagram for some use cases **Jia Yi** Worked on sequence diagram for some use cases Worked on state machine diagram for the system UI **Kai Hang** Worked on completion of all use cases **Denzyl** Worked on completion of all use cases **Jian Le** Worked on the full class diagram with controller, boundary and entity classes	**Action**

MINUTES OF MEETING III

Date	:	30 September 2022	
Time	:	1100 - 1200	
Venue	:	Zoom	
Reference No.	:	3	
Present	:	Team Leader	Lau Yong Jie
		Assistant Leader	Tay Jia Yi
		Members	Cai Kai Hang
			Denzyl David
			Hoo Jian Le
Absent with apologies	:		
Minutes by	:	Hoo Jian Le	
Vetted by	:	Lau Yong Jie	
Signed by	:	Lau Yong Jie	

Yong Jie

Led the discussion for submission of Lab 3 deliverables as well as what to learn for creation of website

Created system architecture model

Tasked to create the login functionallity

Action

Jia Yi

Tasked to learn SpringBoot, Maven & MongoDB for building website

Tasked to create the discussion functionality

Kai Hang

Tasked to learn SpringBoot, Maven & MongoDB for building website

Tasked to create the admin functionalities

Denzyl

Tasked to learn SpringBoot, Maven & MongoDB for building website

Tasked to create the course functionalities

Jian Le

Tasked to learn SpringBoot, Maven & MongoDB for building website

Tasked to create the course functionalities

Refined the existing class diagram to make it more presentable

MINUTES OF MEETING IV

Date	:	11 October 2022	
Time	:	1400 - 1430	
Venue	:	Zoom	
Reference No.	:	4	
Present	:	Team Leader	Lau Yong Jie
		Assistant Leader	Tay Jia Yi
		Members	Cai Kai Hang
			Denzyl David
			Hoo Jian Le
Absent with apologies	:		
Minutes by	:	Hoo Jian Le	
Vetted by	:	Lau Yong Jie	
Signed by	:	Lau Yong Jie	

Yong Jie

Led the discussion for submission of Lab 3 deliverables, as well as checking on the progress of the website

Created the application skeleton

Worked on the login functionalities

Action

Jia Yi

Worked on the discussion functionalities

Kai Hang

Changed to work on the website front-end instead

Tasked to learn Vue.js to build the front-end

Denzyl

Changed to work on the website front-end instead

Tasked to learn Vue.js to build the front-end

Jian Le

Worked on the course functionalities

MINUTES OF MEETING V

Date	:	18 October 2022	
Time	:	1400 - 1500	
Venue	:	Zoom	
Reference No.	:	5	
Present	:	Team Leader	Lau Yong Jie
		Assistant Leader	Tay Jia Yi
		Members	Cai Kai Hang
			Denzyl David
			Hoo Jian Le
Absent with apologies	:		
Minutes by	:	Hoo Jian Le	
Vetted by	:	Lau Yong Jie	
Signed by	:	Lau Yong Jie	

Yong Jie

Led the discussion for checking the progress of the website

Worked on the login functionalities

Action

Jia Yi

Worked on the discussion functionalities

Kai Hang

Worked on the front-end development

Denzyl

Worked on the front-end development

Jian Le

Worked on the course functionalities

MINUTES OF MEETING VI

Date	:	23 October 2022
Time	:	1300 - 1400
Venue	:	Zoom

Reference No. : 6

Present	:	Team Leader	Lau Yong Jie
		Assistant Leader	Tay Jia Yi
		Members	Cai Kai Hang
			Hoo Jian Le

Absent with apologies : Denzyl David (Attended a funeral at Johor Bahru)

Minutes by	:	Hoo Jian Le
Vetted by	:	Lau Yong Jie
Signed by	:	Lau Yong Jie

Yong Jie

Led the discussion for submission of Lab 4 deliverables

Touched up and completed the login, discussion and search course functionalities

Finished up the front-end of website

Tasked Denzyl to finish up on the test cases

Action

Jia Yi

Changed the existing sequence diagram to suit the changes done during the coding period

Changed the existing state machine diagram to suit the changes done during the coding period

Kai Hang

Finished up the front-end of website (Course information and Vuechart)

Jian Le

Changed the existing class diagram to suit the changes done during the coding period

Changed the use cases to suit the changes done during the coding period