
Software Requirements Specification

for
<SchoolPicker>

Version 1.0 approved

**Prepared by <Wang Shi Ying, Yong Kiong Lu, Tang Guan Yu, Tay Yuan
Geng, Nicholas Tan, Woo Weng Tai>**

<SC2006 SCS7 Team30>

<01/10/2024>

Table of Contents

1. Introduction	1
1.1 Purpose	1
1.2 Document Conventions	1
1.3 Intended Audience and Reading Suggestions	1
1.4 Product Scope	1
1.5 References	2
2. Overall Description	3
2.1 Product Perspective	3
2.2 Product Functions	4
2.2.1 Authentication	4
2.2.2 Profile Management	4
2.2.3 School Recommendations	4
2.2.4 School Comparison	4
2.2.5 School Search	4
2.2.6 Forum and Community Engagement	4
2.3 User Classes and Characteristics	6
2.3.1 Student	6
2.3.2 Parents/Guardians	6
2.3.3 Admin	7
2.4 Operating Environment	7
2.4.1 Hardware Platform	7
2.4.2 Operating System Compatibility	7
2.4.3 Frontend	7
2.4.4 Backend	7
2.4.5 Database and Data Storage	7
2.4.6 Integration with External Applications	8
2.5 Design and Implementation Constraints	8
2.5.1 Regulatory and Compliance Requirements	8
2.5.2 Security Considerations	8
2.5.3 Hardware and Performance Constraints	8
2.6 User Documentation	8
2.7 Assumptions and Dependencies	8
2.7.1 Stable Access to Third-Party APIs	9
2.7.2 Data Availability and Accuracy	9
3. External Interface Requirements	10
3.1 User Interfaces	10
3.1.1 Navigation bar	10
3.1.2 Dropdown Menu	10
3.1.3 Sign Up Page	11

3.1.4 Login Page	12
3.1.5 School Search Page	13
3.1.6 School Compare Page	14
3.1.7 School Recommendation Page	15
3.1.8 Forum Page	16
3.1.8.1 View Post Page	17
3.1.8.2 My Post Page	17
3.1.8.3 Edit Post	18
3.1.8.4 Delete Post	18
3.1.8.5 Manage Activity Page	19
3.1.8.6 View Flagged Post Page	19
3.1.9 User Profile Page	20
3.2 Hardware Interfaces	21
3.3 Software Interfaces	21
3.3.1 Database	21
3.3.2 API	21
3.3.3 Tech Stack	21
3.4 Communications Interfaces	22
4. System Features	22
4.1 Authentication Features	22
4.1.1 Signup	22
4.1.1.1 Description and Priority	23
4.1.1.2 Stimulus/Response Sequences	23
4.1.1.3 Functional Requirements	23
4.1.2 Sign in	24
4.1.2.1 Description and Priority	24
4.1.2.2 Stimulus/Response Sequences	24
4.1.2.3 Functional Requirements	25
4.2 User Features	25
4.2.1 Search Schools	25
4.2.1.1 Description and Priority	25
4.2.1.2 Stimulus/Response Sequences	25
4.2.1.3 Functional Requirements	26
4.2.2 Filter Schools	26
4.2.2.1 Description and Priority	26
4.2.2.2 Stimulus/Response Sequences	27
4.2.2.3 Functional Requirements	27
4.2.3 Compare Schools	27
4.2.3.1 Description and Priority	27
4.2.3.2 Stimulus/Response Sequences	28
4.2.3.3 Functional Requirements	28

4.2.4 Forum	28
4.2.4.1 Browse Forum Posts	28
4.2.4.2 Create Posts	29
4.2.4.3 Edit Own Posts	30
4.2.4.4 Delete Own Posts	31
4.2.4.5 View Own Posts	33
4.2.4.6 Comment on Posts	33
4.2.4.7 Report Posts	34
4.2.5 Profile Builder	35
4.2.5.1 Preferred Criterias	36
4.2.5.2 Recommendation	36
4.3 Admin Features	37
4.3.1 View Flagged Post	37
4.3.1.1 Description and Priority	37
4.3.1.2 Stimulus/Response Sequences	38
4.3.1.3 Functional Requirements	38
4.3.2 Unflag Posts	38
4.3.2.1 Description and Priority	38
4.3.2.2 Stimulus/Response Sequences	39
4.3.2.3 Functional Requirements	39
4.3.3 Delete comments	39
4.3.3.1 Description and Priority	39
4.3.3.2 Stimulus/Response Sequences	39
4.3.3.3 Functional Requirements	40
4.3.4 Delete Flagged Post	40
4.3.4.1 Description and Priority	40
4.3.4.2 Stimulus/Response Sequences	40
4.3.4.3 Functional Requirements	41
5. Other Nonfunctional Requirements	42
5.1 Legal Requirements	42
5.2 Reuse Requirements	42
5.3 Usability Requirements	42
5.4 Internationalisation Requirements	42
5.5 Business Rules	42
6. Other Requirements	43
Functional Requirement 1	48
a. RegisterAccount	48
b. Login	48
Functional Requirement 2	50
a. ProfileBuilderActivity	50
Functional Requirement 3	51

a. FindSchool	51
Functional Requirement 4	52
a. CompareSchool	52
Functional Requirement 5	53
a. RecommendSchool	53
Functional Requirement 6	54
a. ForumActivity	54
b. CreatePost	55
c. ViewPost	56
d. ManagePostActivity	57
e. EditPost	58
f. DeletePost	58
Functional Requirement 7	59
a. RemovePost	59
API Function	60
a. RetrieveSchoolDetails	60
1. Black Box Testing	60
2. White Box Testing	65
3. Other Test Cases	69

Revision History

Name	Date	Reason For Changes	Version
	09/11/2024	Initial Document	1.0

1. Introduction

1.1 Purpose

This Software Requirements Specification (SRS) document is intended for the **SchoolPicker** web application, version 1.0. The purpose of this document is to specify the features, functionalities, and requirements of the **SchoolPicker** system, to ensure a clear understanding of the application's objectives, scope and technical specifications. The **SchoolPicker** web application is designed mainly to aid pre-tertiary students and their parents in the process of school selection and decision making, based on non-academic criterias such as zones, interested subjects, co-curricular activities, as well as related discussions in forums. This document aims to document the entire process, and outline important aspects such as overall description, external interface requirements, system features and non-functional requirements. This documentation serves as a guidance in development, testing, and maintenance stages, to ensure that project objectives are achieved.

1.2 Document Conventions

This Software Requirements Specification (SRS) document follows specific conventions to ensure clarity, consistency, and ease of use across all sections.

1. Font styles:
 - Header 1: Font - Times, size - 18
 - Header 2: Font - Times, size - 14
 - Header 3: Font - Times, size - 12
 - Header 4: Font - Times, size - 11
 - Content: Font - Arial, size - 11

2. Names and key terms will be bolded.

1.3 Intended Audience and Reading Suggestions

This document is intended for the developers, testers and users of the application. Each section contains a detailed explanation of the requirements for the **SchoolPicker** web application. It is recommended to read this documentation in sequence from the start to the end to gain a full understanding of the requirements and features.

1.4 Product Scope

SchoolPicker is a web application solution that aims to be a one stop solution that allows users such as parents or students to explore and find out more about the schools in Singapore ranging from the primary and secondary levels. Users would be able to find a school through the school name or through a set of filters, after learning more about a specific school, users would be able to compare between two schools as well. The platform also provides a forum that allows different users to discuss various topics regarding school selection.

1.5 References

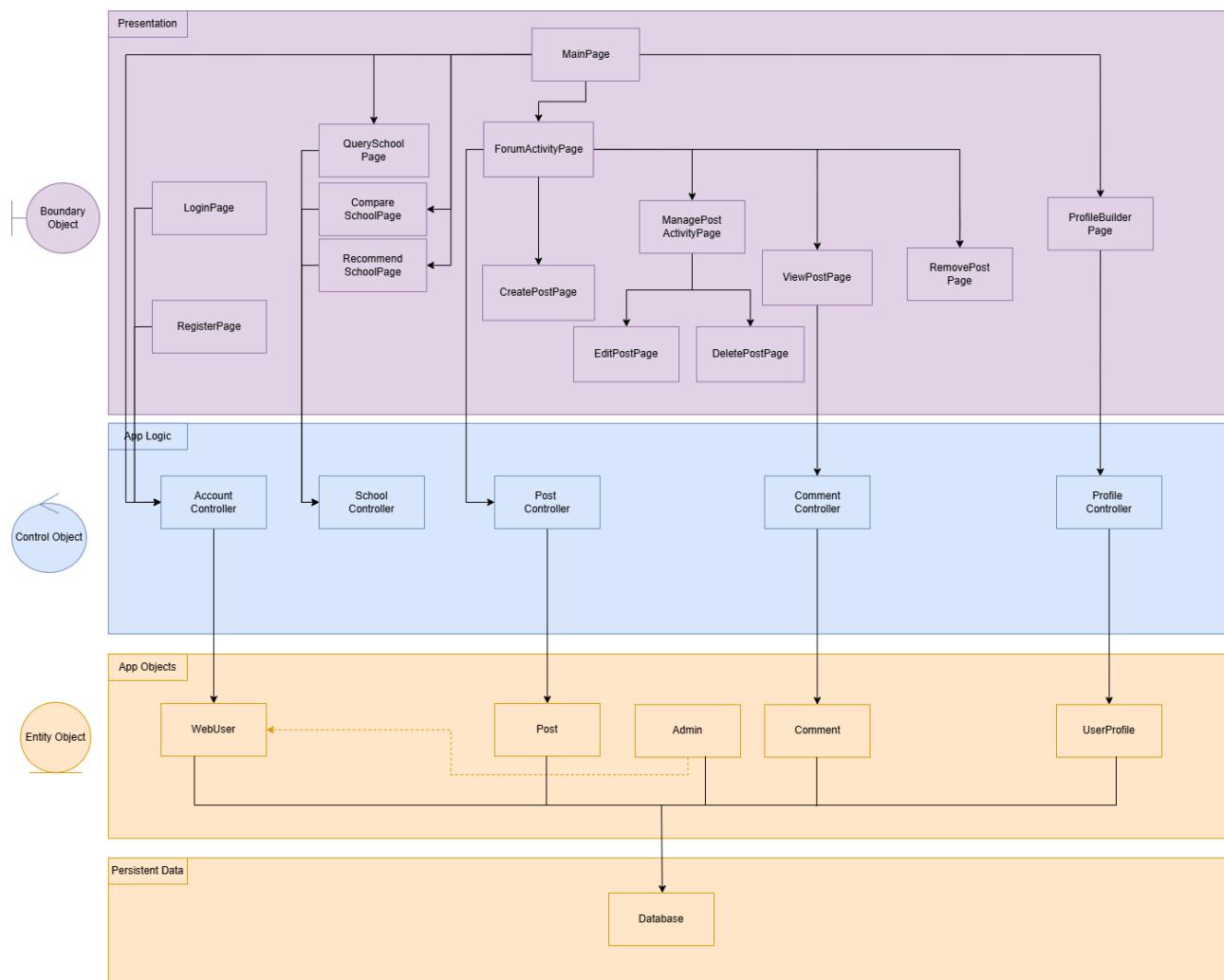
- I. ASP.NET: <https://learn.microsoft.com/en-us/aspnet/core/web-api/?view=aspnetcore-8.0>
- II. Data.gov (API Documentation): <https://guide.data.gov/developer-guide/collection-apis>
- III. PostgreSQL: <https://www.postgresql.org/docs/current/index.html>
- IV. React: <https://react.dev/learn/typescript>

2. Overall Description

2.1 Product Perspective

The **SchoolPicker** website is a new, self-contained product designed to assist students and parents in Singapore in selecting suitable schools based on non-academic criterias such as zone locations, and academic or extracurricular interests. The platform is an interactive guide for school exploration, aiming to streamline the decision-making process for pre-tertiary students, from primary school to junior college. This product is built to address the needs of students who are transitioning between key educational stages and wish to find schools that align with their interests and location preferences. It allows users to explore schools, compare schools and receive tailored recommendations. All in all, the **SchoolPicker** website serves as a digital tool, integrating school data and user preferences to offer a seamless and personalized browsing experience.

The system architecture diagram shown below outlines the interconnections between components of the system, showing how they communicate and interact with each other to achieve the desired functionalities.



System Architecture Diagram

2.2 Product Functions

The **SchoolPicker** website must enable users to perform the following major functions:

2.2.1 Authentication

- Sign up and log in to create and access personalized profiles.
- Validate user inputs, such as ensuring email format and Terms & Conditions agreement.

2.2.2 Profile Management

- Build and save personalized profiles for school recommendations based on location, subject interests, and extracurricular activities.

2.2.3 School Recommendations

- Generate customized school recommendations based on the user's profile preferences.
- Update recommendations dynamically when profile settings are adjusted.

2.2.4 School Comparison

- Allow users to compare two selected schools side-by-side.
- Display key differences in areas such as location, subjects offered, and co-curricular activities.

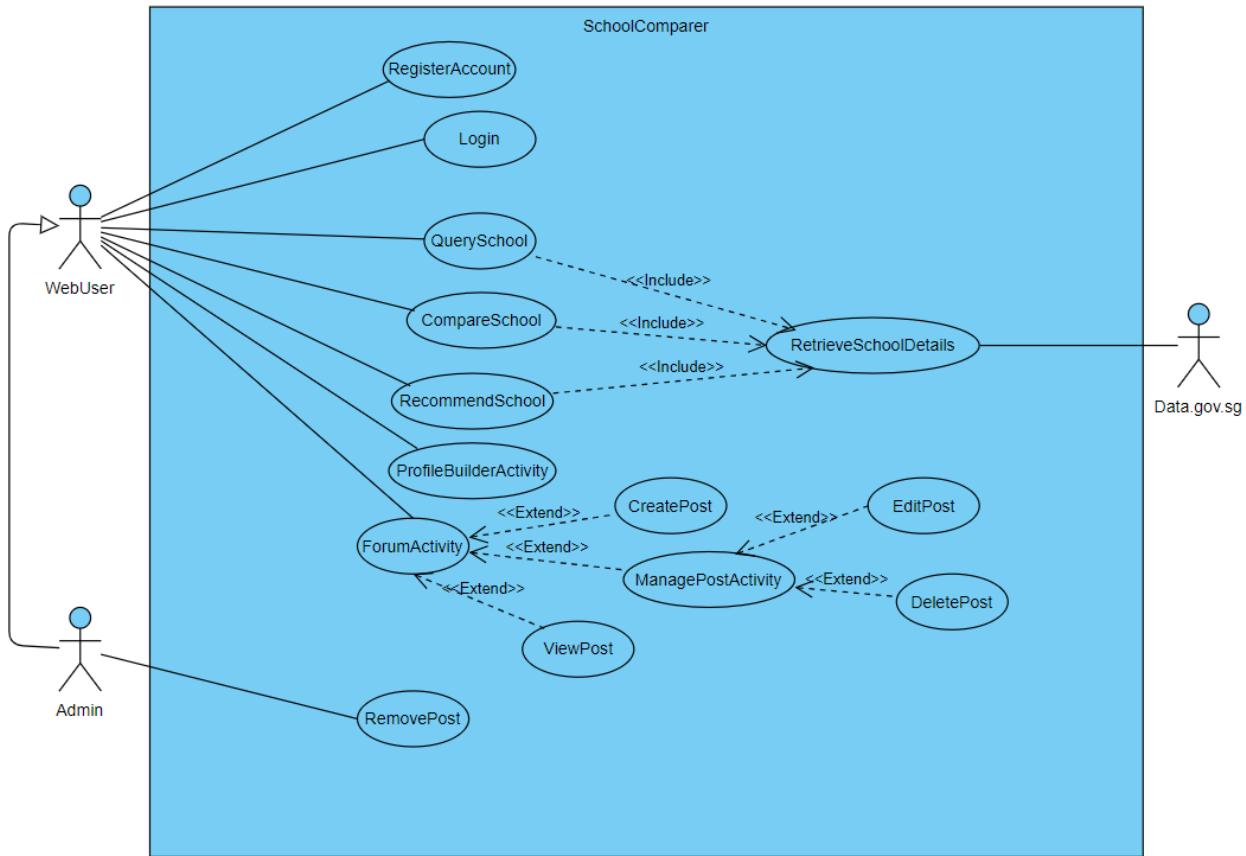
2.2.5 School Search

- Enable users to search for specific schools without altering their saved profiles.
- Access school websites directly for further exploration by pressing on the redirection provided in the search.

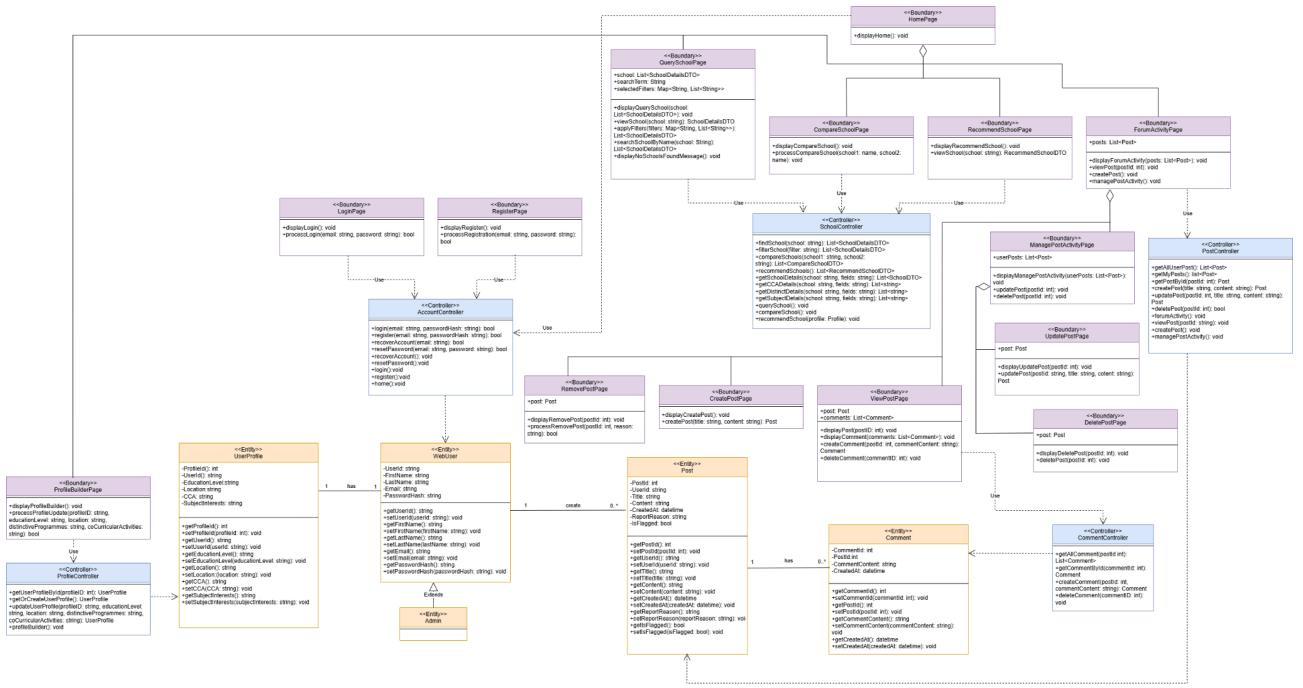
2.2.6 Forum and Community Engagement

- Create, view, and comment on posts related to school experiences and questions.
- Flag posts or comments for inappropriate content
- Provide administrators with tools to delete or unflag reported posts and comments.

Use case diagram and class diagram are shown below to provide a visualization for the functionalities.



Use case Diagram



Class Diagram

2.3 User Classes and Characteristics

The **SchoolPicker** website serves multiple user classes with varying levels of access, frequency of use, and functionalities. The following are the primary user classes:

2.3.1 Student

- Description: Students exploring pre-tertiary school options based on their interests, location, and co-curricular preferences.
- Frequency of Use: Moderate to frequent; usage may increase during key enrollment periods.
- Functions Used: Profile Builder, School Recommendations, School Comparison, School Search, and Forum.
- Characteristics: Students who want to make more informed decisions about their pre-tertiary school selections; may be less technically inclined and require user-friendly navigation.

2.3.2 Parents/Guardians

- Description: Parents or guardians assisting their children in selecting suitable schools.

- Frequency of Use: Moderate; typically during school enrollment cycles.
- Functions Used: Profile Builder, School Recommendations, School Comparison, School Search, and Forum.
- Characteristics: Likely to use the platform to make informed decisions about their child's education; may be less technically inclined and require user-friendly navigation.

2.3.3 Admin

- Description: Website admin responsible for overseeing content, managing user-reported issues, and maintaining platform functionality.
- Frequency of Use: Regular; ongoing maintenance and monitoring.
- Functions Used: Full access, including Manage Forum Activities which includes forum moderation and flagged content review.
- Characteristics: Technically skilled, familiar with website management, and responsible for ensuring a safe and well-maintained community space.

2.4 Operating Environment

2.4.1 Hardware Platform

- The web application will be accessible on common consumer devices, including desktop computers and laptops.

2.4.2 Operating System Compatibility

- Windows (Version 10 and above)
- MacOS (Version 10.15 Catalina and above)

2.4.3 Frontend

- React for frontend framework and Typescript as the programming language.

2.4.4 Backend

- ASP.NET for backend framework, PostgreSQL for database.
- The server environment will be hosted on a cloud-based server such as AWS.
- The backend services used a RESTful API architecture, allowing the frontend application to retrieve and post data asynchronously.

2.4.5 Database and Data Storage

- PostgreSQL will store user profiles, school data, and forum posts and comments.

2.4.6 Integration with External Applications

- The **SchoolPicker** website redirects users to external school websites, allowing users to access more detailed school-specific information.

2.5 Design and Implementation Constraints

2.5.1 Regulatory and Compliance Requirements

- Data Protection and Privacy: The system must comply with data protection regulations such as the General Data Protection Regulation (GDPR) and Singapore's Personal Data Protection Act (PDPA). This includes secure handling, storage, and transmission of user data, as well as providing users with rights over their data.
- Educational Content Standards: Any content related to school information should comply with relevant educational standards and present accurate and unbiased information.

2.5.2 Security Considerations

- Data Encryption: Sensitive data, particularly user profiles and personal information, must be encrypted both in transit and at rest to prevent unauthorized access.

2.5.3 Hardware and Performance Constraints

- Response Time: Recommendations and search features must deliver results within 2-3 seconds for optimal user experience. Efficient database queries and caching mechanisms may be necessary to meet this requirement.

2.6 User Documentation

A guide to install and run the web application is provided in the README file in our Github Repository.

https://github.com/cjkejw/SC2006_T30_Code_New

2.7 Assumptions and Dependencies

The following assumptions and dependencies are considered for the development and operation of the **SchoolPicker** website. These factors are not guaranteed and could impact the project if they are incorrect, not widely shared, or change during development.

2.7.1 Stable Access to Third-Party APIs

- The project assumes consistent and reliable access to data.gov.sg APIs.

2.7.2 Data Availability and Accuracy

- The platform assumes regular and accurate updates of school data, including information on locations, academic subjects, and co-curricular activities (CCAs) in the APIs. If school data becomes outdated or unavailable, recommendations may not accurately reflect current school offerings, affecting user trust.

3. External Interface Requirements



Welcome to School Picker!

Unsure about Schools? We are here to help!

3.1 User Interfaces

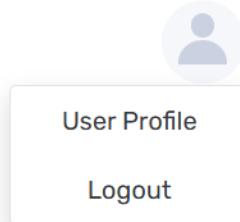
The user interface is designed to provide users a clear direction on using the web application functionality. Additionally, the choice of using purple with white brings about a calm environment with an easily accessible interface. The homepage shown above displays a welcome message to show hospitality and make a good impression on the user.

3.1.1 Navigation bar



All webpages in the web application use a standardized navigation bar at the top header. The navigation bar provides the links to key webpages of the application for the user to access to any time they wish. These links include the “School Search”, “Compare School”, “Recommend School” and “Forum” sections. The logo can also be clicked for the user to return to the homepage.

3.1.2 Dropdown Menu



The user icon can be clicked to reveal a dropdown-type menu for the profile builder function or logging out of that account. It can be clicked again to hide the menu, making the webpage header cleaner by hiding these sub-functions as they are not the main focus of the web application.

3.1.3 Sign Up Page

Sign Up

Email

First name Last name

Password

Confirm Password

• All-in-one information
• Find the **MOST SUITABLE** school
• Not sure which school?
COMPARE!

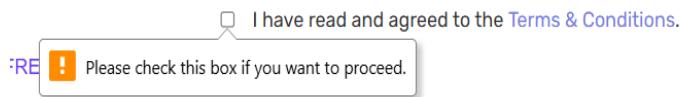
I have read and agreed to the [Terms & Conditions](#).

Sign up **NOW** for [FREE!](#)

CONFIRM

Already have an account? [Sign in](#)

The sign up page requires the user to input the following fields: Email, first name, last name, and password, and re-type the password again for confirmation. Along with the input fields, they must also check the terms and conditions box before the sign-up process can be carried out. The 'Terms & Conditions' can be clicked to view the content should the user would like any clarity about it.



The email field has a validation check to see if the inputted info is an actual email, else it will prompt an error message to the user to enter a valid format.

try@gmail

Email is invalid

The password field has a format which requires at least an 8-character input with the usage of lowercase, uppercase and numbers. The 'confirm password' field will validate with the above password to ensure they are matching before proceeding. The password characters are also hidden for security purposes.

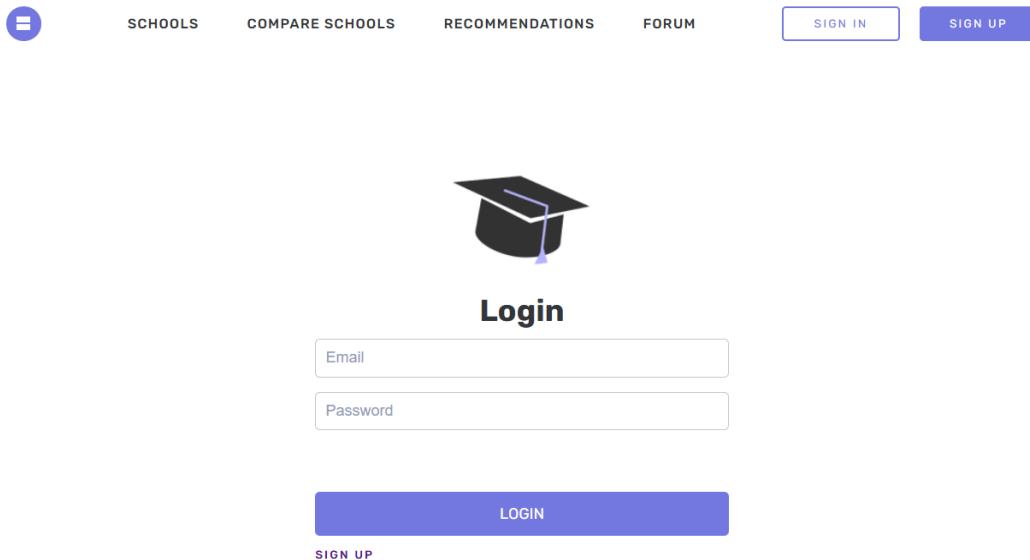
.....

Password must contain at least 1 number.

....

Passwords do not match

3.1.4 Login Page



The login page requires the user to key in the email and password input of their account created from the sign up page. A mini ‘Sign Up’ link is below the login button and serves as a small indicator of where to go in the event the user currently doesn’t have an account yet.

Attempting to login into an account using email and password which is not recorded in the web application’s database will result in an invalid username error message prompt to the user.

hi@gmail.com

.....

Invalid Username!

The email field will check if the inputted data is an actual email, else it will prompt the user to key in a valid email input.

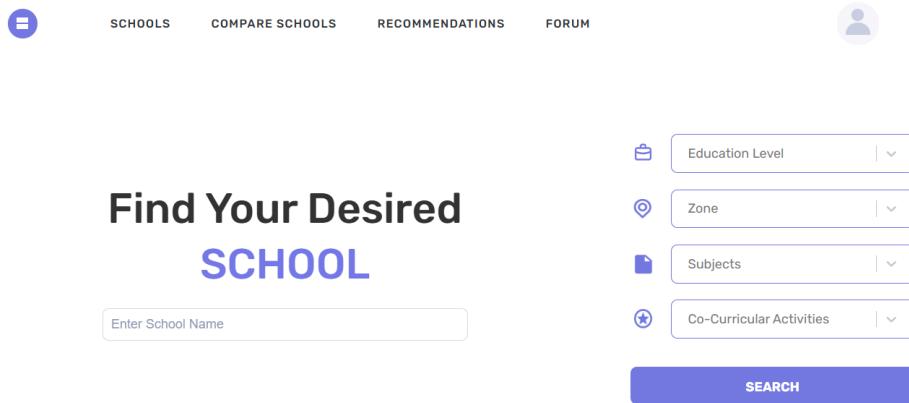
try@

Email ! Please enter a part following '@'. 'try@' is incomplete.

Password

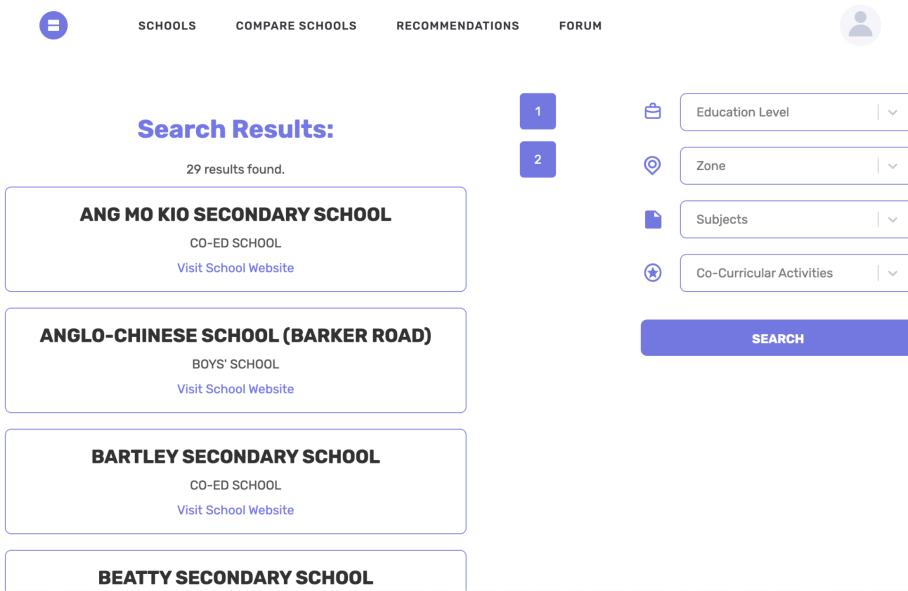
Password is required

3.1.5 School Search Page



The screenshot shows the School Search Page. At the top, there is a navigation bar with icons for Schools, Compare Schools, Recommendations, and Forum, along with a user profile icon. Below the navigation bar, the main heading "Find Your Desired SCHOOL" is displayed in large, bold letters. To the left of the heading is a search input field labeled "Enter School Name". To the right of the heading are four filter dropdowns: "Education Level", "Zone", "Subjects", and "Co-Curricular Activities". A large blue "SEARCH" button is located at the bottom right of the filter area.

The school search page allows the user to search for a school through the school's name by inputting into the 'Enter School Name' fields there. Users can also search for schools through specific filter options located at the right side of the webpage. Currently the options 'Education Level', 'Zone' 'Subjects' and 'Co-Curricular Activities' are options available for the user to filter their search results.



The screenshot shows the Search Results Page. At the top, it features the same navigation bar and user profile icon as the search page. Below the navigation bar, the heading "Search Results:" is displayed in blue. It indicates "29 results found." Below this, there are four search results cards, each containing the school name, type, and a link to visit the school's website. To the right of the results, there are two small numbered boxes (1 and 2) above the filter dropdowns and the "SEARCH" button, likely indicating the current step or result count in a multi-step process.

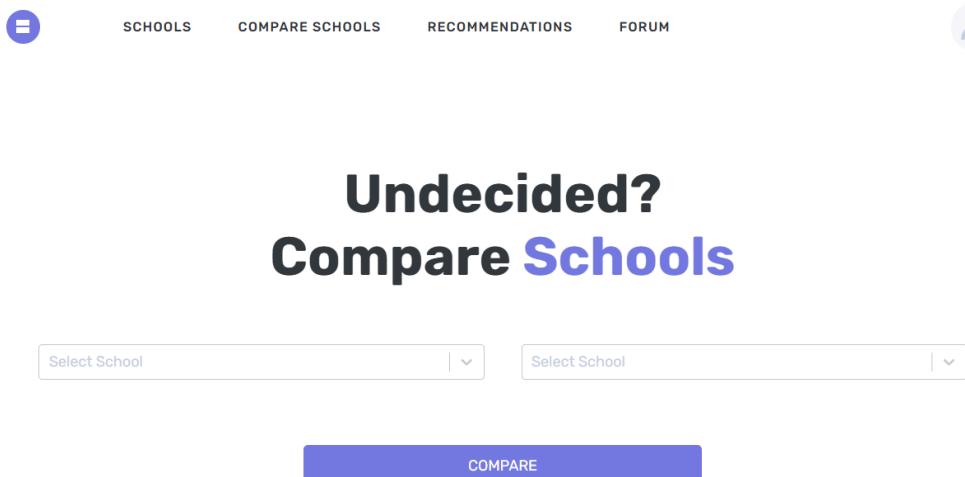
- 1**
- 2**

ANG MO KIO SECONDARY SCHOOL CO-ED SCHOOL Visit School Website	Education Level
ANGLO-CHINESE SCHOOL (BARKER ROAD) BOYS' SCHOOL Visit School Website	Zone
BARTLEY SECONDARY SCHOOL CO-ED SCHOOL Visit School Website	Subjects
BEATTY SECONDARY SCHOOL	Co-Curricular Activities

SEARCH

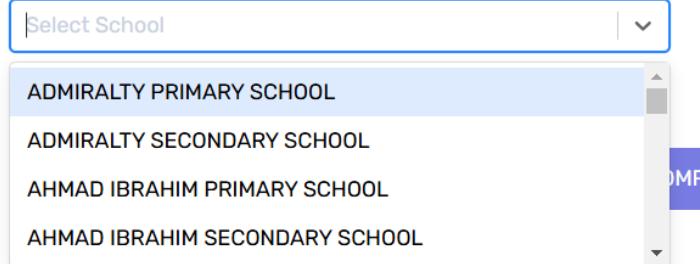
When the search button is clicked, the application will return the user a list of schools that matched their search criteria. The options for the filter are also in a drop-down list so that the user can see what existing available choices they can pick from already under a specific filter type. From the return options, there will also be a link to that specific school's official website for the user to further explore a specific school.

3.1.6 School Compare Page



The school compare page allows the user to compare information of 2 specific schools side by side to give them a better visualization of what each selected school can provide or offer under specific areas.

The user can select 2 schools through the drop-down list which displays the list of all currently existing schools available from the data.gov.sg API. After the user selects 2 schools to compare with, they will click the compare button and an expanded detail of the schools chosen will be put in a side by side format to allow easy comparison by the user.



Compare Results

<p>Subjects Offered: Basic Chinese Language, Computer Applications, Design & Technology, Elements Of Business Skills, English Language, Mathematics, Basic Malay Language, Science, Social Studies, Additional Mathematics, Art, Chinese Language, Food & Consumer Education, Geography, History, Literature In English, Malay Language, Nutrition And Food Science, Principles Of Accounts, Science (Chem, Bio), Science (Phy, Chem), Humanities (SS, Geography), Humanities (SS, History), Chemistry, Computing, Higher Chinese Language, Higher Malay Language, Physics, Humanities (SS, Literature In Malay), Music, Physical Education</p> <p>Distinctive Programmes: NIL</p> <p>CCAs Offered: Badminton, Netball, Rugby, Track And Field, Wushu, National Police Cadet Corps, National Cadet Corps (land), National Civil Defence Cadet Corps, Modern Dance, English Drama, Concert Band, Art And Crafts, Robotics,</p>	<p>Subjects Offered: Food & Consumer Education, Basic Chinese Language, Computer Applications, English Language, Mathematics, Basic Malay Language, Mobile Robotics, Science, Social Studies, Basic Tamil Language, Additional Mathematics, Art, Chinese Language, Design & Technology, Malay Language, Principles Of Accounts, Science (Phy, Chem), Humanities (SS, Geography), Humanities (SS, History), Tamil Language, Biology, Chemistry, Geography, Higher Chinese Language, Higher Malay Language, Higher Tamil Language, Literature In English, Physics, Humanities (SS, Literature In English), Humanities (SS, Literature In Malay), Music, Physical Education</p> <p>Distinctive Programmes: NIL</p> <p>CCAs Offered: Student Leadership (council), Basketball, Netball, Football, Table Tennis, Wushu, Volleyball, Girl Guides, National Police Cadet Corps, National Cadet Corps (land), Choir, Chinese Dance, Malay Dance, English Drama,</p>
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

3.1.7 School Recommendation Page

The screenshot shows a user interface for a school recommendation service. At the top, there is a navigation bar with icons for user profile, schools, compare schools, recommendations, and forum. Below the navigation bar, the title "Recommended Schools" is displayed. Three school entries are listed in a grid:

- CATHOLIC JUNIOR COLLEGE** Type: CO-ED SCHOOL Zone: SOUTH [Visit School Website](#)
- NANYANG JUNIOR COLLEGE** Type: CO-ED SCHOOL Zone: SOUTH [Visit School Website](#)
- ST. ANDREW'S JUNIOR COLLEGE** Type: CO-ED SCHOOL Zone: SOUTH [Visit School Website](#)

The school recommendation page will automatically return a list of recommended schools to the user upon entering the webpage. The webpage will generate results based on the user's profile current preferences to tailor suggestions to the user.

If the webpage is unable to tailor to the user preferences, it will display a message saying there are no results matching the user's preferences and for them to re-adjust the criteria of their user profile.

Recommended Schools

No schools match your profile. Try adjusting your criteria.

3.1.8 Forum Page

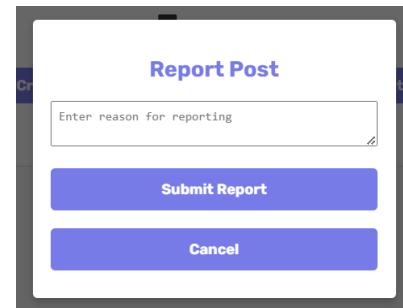
The screenshot shows the 'Forum' section of the SchoolPicker website. At the top, there is a navigation bar with links for 'SCHOOLS', 'COMPARE SCHOOLS', 'RECOMMENDATIONS', 'FORUM', and a user profile icon. Below the navigation, the word 'Forum' is centered in large bold letters. Underneath it are two buttons: 'Create Post' and 'View My Posts'. A section titled 'Posts by: Joseph Tan' contains a single post. The post title is 'Entering JC soon!', followed by the text 'Would like tips to prepare for!', and the creation timestamp 'Created At: 11/9/2024, 4:42:48 PM'. A blue 'Report' button is located at the bottom of this post's container.

Posts by: Paimon Brando

Is Ai Tong Secondary a good school?

Still deciding my new secondary school, would like to know if Ai Tong is worth it

The forum page allows communication and engagement between different users of the web application through the usage of posts. Entering the webpage will bring up a list of existing posts in the web application and provide some of the following forum-related links here such as 'Create Post' and 'View My Post', which allow the creation of posts and management of personal posts respectively. The individual posts can be viewed by clicking the post title and it will direct the user to view the post. Additionally, the user can also report posts here they deem inappropriate by clicking the report button and it will be flagged for the admins to deal with it. They have to key in a reason first and click the 'Submit Report' to flag it. 'Cancel' can be clicked if they wish to not report the post.



A third option 'Manage Activity' is only visible to users with the 'Admin' role which allows them to fulfill their moderating duties by removing posts and comments that are 'flagged' by other users.

The screenshot shows the 'Forum' section of the SchoolPicker website. At the top, there is a navigation bar with links for 'SCHOOLS', 'COMPARE SCHOOLS', 'RECOMMENDATIONS', 'FORUM', and a user profile icon. Below the navigation, the word 'Forum' is centered in large bold letters. Underneath it are three buttons: 'Create Post', 'View My Posts', and 'Manage Activity'. A section titled 'Posts by: Paimon Brando' contains a single post. The post title is 'Is Ai Tong Secondary a good school?', followed by the text 'Still deciding my new secondary school, would like to know if Ai Tong is worth it'.

3.1.8.1 View Post Page

The screenshot shows a user interface for a forum. At the top, there is a navigation bar with icons for Schools, Compare Schools, Recommendations, and Forum, along with a user profile icon. Below the navigation bar is a post card. The post title is "Entering JC soon!". The post content says "Would like tips to prepare for!". It was created at "11/9/2024, 4:42:48 PM". Below the post is a "Comments:" section containing a single comment: "All the best! - 11/9/2024, 5:25:22 PM". There is a text input field for adding a new comment and a "Submit" button.

The view post page will show the details of a specific selected post by the user. This is accessed by clicking the title of the post on the forum webpage. Here users can leave a comment on a post for interaction by keying a comment content on the textbox and clicking the submit button. Users can hit 'Back to Forum' at the top and return to where they were previously.

3.1.8.2 My Post Page

The screenshot shows a user interface for viewing a user's own posts. At the top, there is a navigation bar with icons for Schools, Compare Schools, Recommendations, and Forum, along with a user profile icon. Below the navigation bar is a section titled "My Posts". It displays a single post with the title "How do you cope with O levels?". The post content says "Too stressful rightnow, need help :=". It was created at "11/9/2024, 5:30:35 PM" and posted by "Admin Guy (admin@gmail.com)". Below the post are two buttons: "Edit" and "Delete".

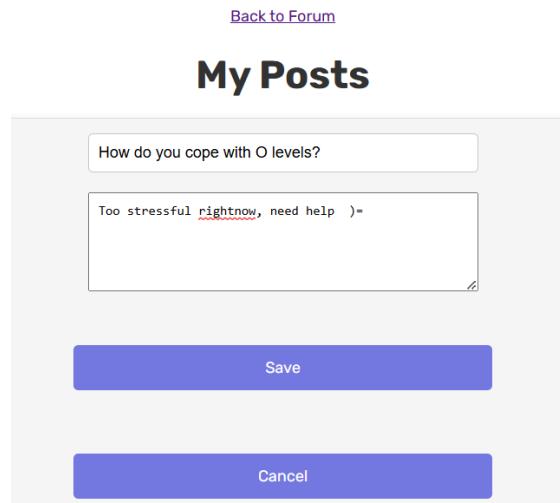
Here users can view all of the posts that they have created specifically. This allows for better management if they need it. Here they can 'Edit' and 'Delete' posts if they wish.



No Posts.

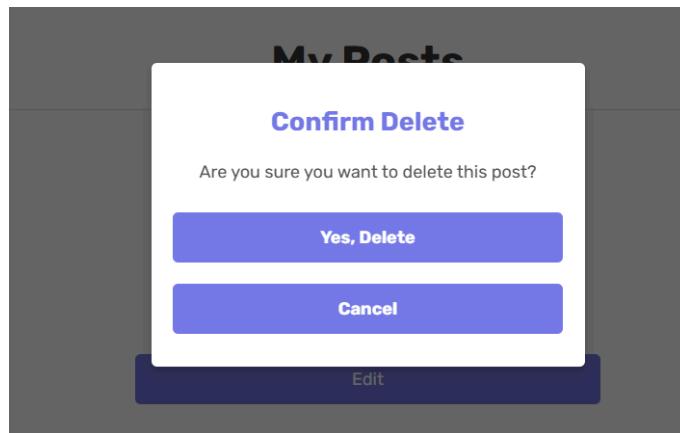
If the user does not have any posts, a prompt 'No Posts' will be displayed instead.

3.1.8.3 Edit Post



Clicking the 'Edit' button of a specific post will allow the user to modify the contents of the selected post. They can modify the title and content field and hit the 'Save' to reflect the changes. They can also hit 'Cancel' to return if they wish.

3.1.8.4 Delete Post



Hitting the 'Delete' button of a specific post will bring a pop up alert asking the user to confirm the deletion process. If the user hits 'Yes, Delete' the post will be removed from the web application. 'Cancel' can be clicked to return in the event the user does not want to proceed with the deletion.

3.1.8.5 Manage Activity Page

The manage activity page allows admins to handle moderating tasks on the web application's forum. It will display what task they can do. For now they can only handle 'Flagged' posts through the 'View Flagged Posts' button.

3.1.8.6 View Flagged Post Page

The view flagged post page allows admins to view a list of all reported posts by users. Here they can see the content of the post and the reason for the report. Admin will be presented with the 'Delete Post' button to delete the post from the forum should they deem the reason justified. Else they can click the 'Unflag Post' button to revert it back to 'unflagged'.

If there is no flagged post in the web application, a message ‘No flagged posts found.’ will be displayed to indicate there are no results.

Flagged Posts

[Back to Forum](#)

No flagged posts found.

3.1.9 User Profile Page

The screenshot shows a user profile page with a navigation bar at the top featuring 'SCHOOLS', 'COMPARE SCHOOLS', 'RECOMMENDATIONS', and 'FORUM' tabs. A user icon is also present. The main section is titled 'User Profile' and contains four dropdown menus:

- Education Level:** Secondary
- Zone:** North
- Subjects:** Chemistry
- Co-Curricular Activities (CCA):** Football

A large blue 'SAVE' button is located at the bottom of the profile section.

The user profile page access through the ‘User Profile’ option at the user hamburger menu allows users to pick specific preferences that help them with their recommendation function. The page will display all available customization for the user to edit and their current preferences will be displayed in each of the available sections. The edit options are picked through a drop-down list. Once done, the user can hit the ‘Save’ button and an alert will inform the user that the information has been updated. Recommendations will change accordingly to match the newly selected options.

localhost:5173 says

Profile updated successfully!

OK

3.2 Hardware Interfaces

For the user side, our web application currently can only be accessed through computer devices, mainly through Windows or MacOs. The web application is compatible on most internet browsers such as Google Chrome, Safari, Firefox, Microsoft Edge etc. The device interacts by sending HTTP requests from the user to the server and receives HTML, CSS, JavaScript and JSON format data. The user's device controls the navigation of the web application through the browser and interactions such as selecting drop-down list options and form submission are handled through the webpage javascript.

For the server side, the web application can be run through hosting a cloud-based or dedicated server. This hosting server will run the web application's backend functions, interacting with the database server and occasionally sending API requests to data.gov.sg for real-time school data for school-related functions and requests. The database server will be along with the main server to support the usage of PostgreSQL for the web application usage. Communication between the user and the server will be handled using HTTP protocols and internal communication within the server can include SQL queries for database-related tasks.

3.3 Software Interfaces

3.3.1 Database

The database used for the web application is PostgreSQL. This will be the choice for our database management system to store the user information and the models they are associated with like individual forum posts and user profile information. Our application communicates with the PostgreSQL database by using an Entity Framework Core, allowing the application to interact with the database using the models developed. The connection to the database is defined as a string configuration defined in the appsettings.json file. Here we specified the database host, name, username, and password to ensure access is granted to the web application for database usage.

3.3.2 API

APIs used for the web application are all from data.gov.sg. Through calling these APIs, the web application will send requests to retrieve the real-time data of Singapore's schools which our web application will use in its functions. The API request will send queries to the data.gov.sg server and the web application will receive data in the form of JSON objects format and the web application will handle the display of these data for the users. Many of our drop-down list options also utilize the real-time data options from data.gov.sg API to ensure that the availability and update of choices the user can make on the web application is always up to date.

3.3.3 Tech Stack

Our web application is built in TypeScript, JavaScript, and CSS styling for the front end, and C# for the back end. The frameworks used are React and ASP.NET Core. React framework

allows easy integration with backend APIs, which allows smooth integration of our front and back end together to complete our application. Additionally also brings about more choices for a responsive user interface and allows for faster performance updates of the application UI. The usage of ASP.NET Core allows support for RESTful API usage and comes built-in with security and authentication features to utilize for web development. ASP.NET comes along with the Entity Framework Core which easily allows us to establish an efficient database communication with our application without much complication.

The usage of ASP.NET Core also allows us to use their built in Swagger API to provide clear documentation of our backend functions and processes by giving us a human-readable description of our backend endpoints. Such descriptions include expected data types required by endpoints and expected response. We are also able to test our functionality and logic without the need of our front end to be completely ready, allowing us to make progress on this section of our web application without delays.

3.4 Communications Interfaces

Our web application uses HTTP protocols for communication between the user and the backend server. RESTful API endpoints are implemented in specific backend functions in the controllers that allow communication and running of these functions with the front end. These endpoints also include PUT, GET, POST, and DELETE HTTP requests to indicate the type of function they are. Specific endpoints can be assigned with the 'Authorized' header to ensure that proper user authentication is made before that endpoint will carry out the function, especially in sections where user's information is involved.

Our web application also uses ASP.NET Core's built in asynchronous API to improve the responsiveness and scalability of our functions. It helps the application process requests simultaneously without waiting on blocking calls. For example, on certain sections of the function where the application is sending multiple API requests to data.gov.sg's server to get the school data from different table sources, the process does not wait for one to be completed before sending the next request. This speeds up our data retrieval process to return the end result without long delays.

To link our front end and back end components together, we utilize Axios HTTP client for our frontend to allow communication with our backend API functions built. Axios uses the same HTTP protocols to send requests and receive responses in JSON format which is compatible with RESTful API endpoints.

4. System Features

4.1 Authentication Features

Authentication feature is a security mechanism that allows new users to create an account and verifies the identity of users trying to access the system.

4.1.1 Signup

4.1.1.1 Description and Priority

Description:

- The Signup feature allows new users to register for an account on our **SchoolPicker** website, providing them with personalized access to features such as profile management and forum participation.

Priority: High

- This feature is crucial to enabling user access and engagement with the website's key functions.

Priority Component Ratings:

- Benefit: 8 - High user acquisition and engagement potential.
- Penalty: 7 - Without this feature, user growth and personalized experiences would be limited.
- Cost: 4 - Development cost is moderate, with ongoing maintenance to ensure security and usability.
- Risk: 7 - High risk if poorly implemented, primarily around user data security and registration process reliability.

4.1.1.2 Stimulus/Response Sequences

1. User Action: The new user clicks on the "SIGN UP" button on the **SchoolPicker** website
 - 1.1. System Response: The system displays a registration form requiring input fields such as first name, last name, email, password, confirm password.
2. User Action: The user fills in the form, tick the Terms & Condition and submits it by clicking the "Confirm" button.
 - 2.1. System Response: The system validates the input fields for no empty fields, format correctness and password strength. If validation passes the system registers the user and the user is brought back to the landing page to sign in.
3. User Action: The user enters any invalid or missing information and clicks "Confirm"
 - 3.1. The system prompts the user with an error message detailing the specific input issue, allowing corrections before resubmission.

4.1.1.3 Functional Requirements

REQ-1: The website must allow users to select the "SIGN UP" button.

REQ-2: The website must provide a registration form with required fields such as first name, last name, email, password, confirm password.

REQ-3: The website must validate all input fields to ensure that the fields are filled and follow the correct format such as valid email structure and strong password requirement.

REQ-4: The website must display error messages for missing or invalid inputs, specifying which field needs to be corrected.

REQ-5: Upon successful registration, the website must be able to store the user's information securely and create a new account.

4.1.2 Sign in

4.1.2.1 Description and Priority

Description:

- The Sign in feature allows users to access their accounts on the **SchoolPicker** website.

Priority: High

- This feature is crucial to enabling user access and engagement with the website's key functions.

Priority Component Ratings:

- Benefit: 9 - Essential for user retention and personalisation engagement.
- Penalty: 8 - Without this feature, users cannot access their account, limiting platform functionality.
- Cost: 4 - Development cost is moderate, with periodic security updates.
- Risk: 7 - High risk if poorly implemented, primarily related to secure handling of login credentials and prevention of unauthorized access.

4.1.2.2 Stimulus/Response Sequences

1. User Action: The user clicks on the "SIGN IN" button on the **SchoolPicker** website
 - 1.1. System Response: The system displays a sign in form requiring the user's registered email and password.
2. User action: The user fills in the form and submits it by clicking the "LOGIN" button.
 - 2.1. The system verifies the entered credentials against stored data. If valid, the user is granted access to their account and redirected to the home page.
3. User Action: The user enters an incorrect email or password and clicks "LOGIN" button
 - 3.1. The system displays an error message indicating incorrect credentials, prompting the user to re-enter their information.

4.1.2.3 Functional Requirements

REQ-1: The website must provide a “SIGN IN” button accessible on the landing page.

REQ-2: The website must display a sign in form requiring email and password fields.

REQ-3: The website must verify that both fields are filled.

REQ-4: The website must authenticate the entered email and password against stored credentials in the database.

REQ-5: The website must display an error message for incorrect email or password, prompting the user to re enter credentials.

REQ-6: Upon successful sign-in, the website must redirect users to the home page.

4.2 User Features

User feature section provides an overview of all functionalities available to regular users within the system.

4.2.1 Search Schools

4.2.1.1 Description and Priority

Description:

- The Search School feature allows users to search for school they want to find by entering the name of the school

Priority: Middle

Priority Component Ratings:

- Benefit: 6 - More convenient for users to find school they are interested in.
- Penalty: 4 - Without this feature, users would have a harder time exploring different school options.
- Cost: 5 - Moderate development cost with potential updates to improve the searching algorithm
- Risk: 2 - Low risk, mainly around ensuring search accuracy and performance.

4.2.1.2 Stimulus/Response Sequences

1. User Action: User clicks “SCHOOLS” on the Navbar.

- 1.1. System Response: System redirects users to the school page and displays a search bar.
2. User Action: The user enters the search term and clicks “SEARCH” button
 - 2.1. System Response: The system processes the search term and displays a list of matching schools, along with brief details and school website link for each result.
3. User Action: User can choose to click on the school website link
 - 3.1. System Response: The user will be redirected to the school website.

4.2.1.3 Functional Requirements

REQ-1: The navbar of the website must have a “SCHOOL” option.

REQ-2: The website must display a search bar after the clicks on “SCHOOL”

REQ-3: The website must allow users to enter keywords for the search.

REQ-4: The website must retrieve and display a list of schools matching the search criteria.

REQ-5: The website must ensure that the user is redirected to the school website after the user clicks on the school website link button.

4.2.2 Filter Schools

4.2.2.1 Description and Priority

Description:

- The filter school feature allows users to view a list of schools matching their preferences by applying filters such as Education Level, Zone, Subjects and Co-Curricular Activities.

Priority: medium

Priority Component Ratings:

- Benefit: 6 - Improves user experience by enabling users to filter school based on applied filters.
- Penalty: 5 - Without this feature, users may become overwhelmed by broad search results and may be less likely to find relevant schools.
- Cost: 4 - Moderate cost for initial implementation with updates as filtering options evolve.
- Risk: 2 - Low risk, mainly related to ensuring the accuracy and responsiveness of filtering.

4.2.2.2 Stimulus/Response Sequences

1. User Action: User clicks “SCHOOLS” on the Navbar.
 - 1.1. System Response: System redirects users to the school page and displays a filter options of Education Level, Zone, Subjects and Co-Curricular Activities.
2. User Action: User selects a filter option for Education Level, Zone, Subjects and Co-Curricular Activities. Afterwards, the user clicks the search button.
 - 2.1. System Response: The system processes the filters that were selected and displays a list of schools in accordance to user choice of filters.
 - 2.2. System Response: Each school is displayed with a brief introduction of the school and a “Visit School Website” button.
3. User Action: User can choose to click on the school website link
 - 3.1. System Response: User will be redirected to the school website.

4.2.2.3 Functional Requirements

REQ-1: The navbar of the website must have a “SCHOOL” option.

REQ-2: The website must display all the filter options after the clicks on “SCHOOL”

REQ-3: Each filter option must have a dropdown list showing all the possible values that the user can choose.

REQ-4: The website must retrieve and display a list of schools matching the filter.

REQ-5: The website must ensure that the user is redirected to the school website after the user clicks on the school website link button.

4.2.3 Compare Schools

4.2.3.1 Description and Priority

Description:

- The compare school feature allows users to select two schools and view them side-by-side, comparing key aspects like Zone, Location, Subject Offered, Distinctive Programmes and CCAs Offered.

Priority: High

Priority Component Ratings:

- Benefit: 8 - Supports decision-making by allowing users to efficiently compare school features in a single view.

- Penalty: 7 - Without this feature, users would need to manually gather and analyze information, reducing the platform's convenience.
- Cost: 4 - Development cost is moderate, with potential for expanded criteria
- Risk: 2 - Low risk, minor challenges in ensuring accurate and responsive comparisons.

4.2.3.2 Stimulus/Response Sequences

1. User Action: User clicks “COMPARE SCHOOLS” on the Navbar.
 - 1.1. System Response: System redirects users to the compare schools page
2. User Action: The user chooses two schools by either typing or selecting the name of the school from a dropdownlist. User then clicks the “COMPARE” button.
 - 2.1. System Response: System processes the two schools that were selected
 - 2.2. System Response: System displays a side by side comparison view, presenting key information like Zone, Location, Subject Offered, Distinctive Programmes and CCAs Offered.

4.2.3.3 Functional Requirements

REQ-1: The navbar of the website must have a “COMPARE SCHOOLS” option.

REQ-2: The website must display the dropdown list so that user can select two school

REQ-3: The website must have a “COMPARE” button

REQ-4: Upon clicking “Compare,” the website must generate a comparison view that shows relevant attributes like Zone, Location, Subject Offered, Distinctive Programmes and CCAs Offered.

4.2.4 Forum

Forum allows users to share their thoughts on the different schools, this feature would only be available when users create an account and sign in.

4.2.4.1 Browse Forum Posts

4.2.4.1.1 Description and Priority

Description:

- The Browse Forum Posts feature allows users to view posts shared by others on the **SchoolPicker** website’s forum. This feature allows users to explore community insights, participate in discussions, and learn more about schools, experiences, and educational topics.

Priority: Medium

Priority Component Ratings:

- Benefit: 7 - Encourages user engagement by allowing users to learn from others' experiences and ask questions.
- Penalty: 7 - without this feature, the website loses a valuable community-driven resource, potentially reducing user engagement.
- Cost: 4 - Moderate development cost with low maintenance costs.
- Risk: 7 - High risk, mainly involving content moderation and user-generated content quality.

4.2.4.1.2 Stimulus/Response Sequences

1. User Action: User clicks "Forum" on the Navbar.
 - 1.1. System Response: System redirects users to the Forum Page and displays all the posts created by users.

4.2.4.1.3 Functional Requirements

REQ-1: The navbar of the website must have a "Forum" option.

REQ-2: The website must display the forum page after the user clicks on the "Forum" option on the Navbar.

4.2.4.2 Create Posts

4.2.4.2.1 Description and Priority

Description:

- The create post feature allows users to create a new post.

Priority: medium

Priority Component Ratings:

- Benefit: 7 - Encourages community engagement and provides users with valuable peer insights.
- Penalty: 6 - Without this feature, user participation and content generation on the forum would be limited.
- Cost: 4 - Moderate cost, including maintenance to ensure a smooth posting experience.
- Risk: 7 - High risk, mainly involving content moderation and user-generated content quality.

4.2.4.2.2 Stimulus/Response Sequences

1. User Action: The user clicks on the “Create Post” button in the forum page.
 - 1.1. System Response: The system displays a post creation form, with fields for title and content.
2. User Action: The user fills out the title and content and clicks on the “Submit Post” button.
 - 2.1. System Response: The system validates the input fields, checking for missing content. Upon successful validation, the system publishes the post and redirects the user back to the forum page.
3. User Action: The user attempts to submit a post with empty fields.
 - 3.1. System Response: The system prompts the user with error messages, specifying which fields need attention before the post can be submitted.

4.2.4.2.3 Functional Requirements

REQ-1: The website must provide a “Create Post” button in the forum section to enable new post creation.

REQ-2: The website must display a post creation form with required fields: title and content.

REQ-3: The website must validate all input fields to ensure that title and content are not empty.

REQ-4: The website must display error messages for any missing fields, clearing indicating which fields are empty.

REQ-5: Upon successful submission, the website must save the new post and immediately display it in the forum page.

4.2.4.3 Edit Own Posts

4.2.4.3.1 Description and Priority

Description:

- The edit post feature allows users to edit the content of their previously created forum posts.

Priority: Medium

Priority Component Ratings:

- Benefit: 7 - Allows users to update their post if they make any mistake writing the post.
- Penalty: 5 - Without this feature, outdated or incorrect information may persist on the forum.
- Cost: 4 - Moderate cost, mainly developing user-specific editing permissions and content tracking.

- Risk: 5 - Moderate risk, associated with ensuring that only the creator of the post can edit their own posts.

4.2.4.3.2 Stimulus/Response Sequences

1. User Action: The user navigates to one of their posts and clicks the “Edit” button.
 - 1.1. System Response: The system displays the post content in an editable form, with fields pre-filled with the original title and content.
2. User Action: The user makes changes to the post content and clicks the “Save” button.
 - 2.1. System Response: The system saves the changes and redirects users to My post page.
3. User Action: The user attempts to save the post with empty fields.
 - 3.1. The system displays error messages, indicating which fields need correction before the changes can be saved.

4.2.4.3.3 Functional Requirements

REQ-1: The website must provide an “Edit” button for each post, visible to the original creator when viewing their post.

REQ-2: The website must display an editable form with fields for title and content, pre populated with the original post title and content.

REQ-3: The website must validate all updated fields to ensure that both the title and content fields are not empty.

REQ-4: The website must display error messages for empty fields, prompting users that the field cannot be empty.

REQ-5: Upon successful validation, the website must save the edited content, immediately displaying the updated post in the forum page.

REQ-6: The post must ensure that only the original post's creator can access and submit edits to that post

REQ-7: The website must redirect users to My post page after they click the “Save” button.

4.2.4.4 Delete Own Posts

4.2.4.4.1 Description and Priority

Description:

- The delete post feature allows users to permanently remove their own forum posts from the **SchoolPicker** website.

Priority: Medium

Priority Component Ratings:

- Benefit: 6 - Allows users to maintain control over their contributions and ensure they can remove outdated or unwanted content.
- Penalty: 4 - Without this feature, users may be unable to remove posts, leading to potential irrelevant or redundant information.
- Cost: 3 - Low cost to implement a delete action with confirmation and proper access.
- Risk: 5 - Moderate risk, associated with ensuring that only the creator of the post can delete their own posts.

4.2.4.4.2 Stimulus/Response Sequences

1. User Action: The user navigates to one of their posts and clicks the “Delete” button.
 - 1.1. System Response: The system displays a confirmation prompt, asking if the user is sure they want to delete the post.
2. User Action: The user confirms deletion by clicking “Yes, Delete” on the confirmation prompt.
 - 2.1. System Response: The system removes the post from the forum, updates the forum to reflect the deletion
3. User Action: The user cancels deletion by clicking “Cancel” on the confirmation prompt.
 - 3.1. System Response: The system closes the confirmation prompt without deleting the post.

4.2.4.4.3 Functional Requirements

REQ-1: The website must provide an “Delete” button for each post, visible to the original creator when viewing their post.

REQ-2: The website must display a confirmation prompt when the user clicks the delete button.

REQ-3: The website must require the user to confirm or cancel deletion before proceeding.

REQ-4 Upon confirmation, the website must permanently delete the post from the forum database and remove it from all views.

REQ-5: The website must immediately update the forum page to reflect the deletion.

REQ-6: The website must ensure that only the original creator of the post can delete it.

4.2.4.5 View Own Posts

4.2.4.5.1 Description and Priority

Description:

- The View Own Posts feature enables users to access and review all posts they have created on the **SchoolPicker** forum. This feature is useful for quickly locating and managing their own posts.

Priority: Medium

Priority Component Ratings:

- Benefit: 6 - Improves user experience by allowing users to efficiently manage their posts.
- Penalty: 3 - Without this feature, users would need to manually locate their posts within the forum, which could be time-consuming.
- Cost: 4 - Low cost, involving filtering forum posts by webuser ID.
- Risk: 2 - Low risk, as this feature does not affect other users or the integrity of forum content.

4.2.4.5.2 Stimulus/Response Sequences

1. User Action: The user Clicks on the “View My Posts” button in the forum page.
 - 1.1. System Response: The system retrieves a list of posts created by the user and displays them.

4.2.4.5.3 Functional Requirements

REQ-1: The website must provide a “View My Posts” option in the user’s profile, allowing them to access posts they created.

REQ-2: The website must retrieve and display a list of all posts created by the user.

REQ-3: Each post in the list should include key details such as title, content, date created, edit button and delete button.

REQ-4: The website should display a message if the user has not created any post and there are no posts to display.

4.2.4.6 Comment on Posts

4.2.4.6.1 Description and Priority

Description:

- The comment on post feature allows users to add comments to forum posts, encouraging discussion, feedback, and engagement within the **SchoolPicker** Community.

Priority: Medium

Priority Component Ratings:

- Benefit: 6 - Encourages more interaction within the **SchoolPicker** Community.
- Penalty: 6 - Without this feature, user engagement will be limited.
- Cost: 4 - Moderate development cost, mainly comment management.
- Risk: 5 - Medium risk, particularly around ensuring content moderation and preventing spam or inappropriate content.

4.2.4.6.2 Stimulus/Response Sequences

1. User Action: The user views a post and decides to add a comment and clicks submit.
 - 1.1. System Response: The system validates the comment input, stores the comment in the database, and refreshes the post view page to include the new comment.
2. User Action: The user submits an empty field.
 - 2.1. System Response: The system displays an error message indicating that the comment field cannot be empty and prompts the user to enter valid content.

4.2.4.6.3 Functional Requirements

REQ-1: The website must display a comment input field in the post view page, allowing users to type their comment.

REQ-2: The website must validate the comment to ensure that it is not empty before submission.

REQ-3: Upon successful validation, the website must store the comment in the database.

REQ-4: The website must refresh the post view page to display the newly added comment immediately after submission.

REQ-5: The website should support timestamping each comment, showing the date and time it was posted.

REQ-6: The website must display error messages for empty comment submissions, guiding users to fill up the empty fields.

4.2.4.7 Report Posts

4.2.4.7.1 Description and Priority

Description:

- The report post feature allows users to flag inappropriate, offensive, or spam content on the **SchoolPicker** Forum, notifying the admin to review and take necessary action.

Priority: High

Priority Component Ratings:

- Benefit: 9 - Helps keep the forum appropriate for younger audiences by allowing prompt removal of harmful or unwanted content.
- Penalty: 7 - Without this feature, inappropriate content may be visible in the forum page for a long duration, reducing trust in the platform.
- Cost: 5 - Moderate development cost, including reporting interfaces, notification workflows, and tracking mechanisms.
- Risk: 4 - Low to moderate risk, with potential concerns around reporting abuse and the need for efficient moderation.

4.2.4.7.2 Stimulus/Response Sequences

1. User Action: The user selects the “Report” button at the bottom of each post they believe is inappropriate.
 - 1.1. System Response: The system displays a reporting form asking the user to input a reason for reporting the post.
2. User Action: The user input the reason and submit the report.
 - 2.1. System Response: The system flag the post and the admin will be able to see that the post is flagged.

4.2.4.7.3 Functional Requirements

REQ-1: The website must provide a “Report” button at the bottom of each post, allowing users to flag content.

REQ-2: When a user selects the “Report” option, the website must display a form where the user can input a reason for reporting, such as spam, offensive content, or other.

REQ-3: The website must include a Cancel button that allows users to cancel reporting if the user changes his/her mind.

4.2.5 Profile Builder

The Profile Builder feature allows users to create and customize their profiles on the **SchoolPicker** website.

4.2.5.1 Preferred Criterias

4.2.5.1.1 Description and Priority

Description:

- The Profile Builder feature enables users to create and personalize their profiles by updating information such as Education level, Zone, Subjects, and Co-Curricular Activities.

Priority: High

Priority Component Ratings:

- Benefit: 8 - Enhances user experience by enabling personalized experience.
- Penalty: 6 - Without this feature, users may find it harder to receive relevant school recommendations.
- Cost: 4 - Development cost is moderate, mainly due to development and data security measures
- Risk: 4 - Low risk if data privacy and security standards are maintained.

4.2.5.1.2 Stimulus/Response Sequences

1. User Action: User navigates to user profile page by click on the user icon and User Profile button
 - 1.1. System Response: The system displays the Profile Builder form, including dropdown lists for Education level, Zone, Subjects, and Co-Curricular Activities.
2. User Action: The user updates their information in the provided fields and clicks "Save".
 - 2.1. System Response: The system saves the data in the database and confirms the update with a success message.

4.2.5.1.3 Functional Requirements

REQ-1: The Profile Builder must include editable fields for Education level, Zone, Subjects, and Co-Curricular Activities.

REQ-2: The website must securely store updated profile data and associate it with the webuser ID.

REQ-3: Upon successful update, the website must confirm with a success message.

REQ-4: The website should allow users to revisit and update their profile anytime.

4.2.5.2 Recommendation

4.2.5.2.1 Description and Priority

Description:

- The Recommendation feature suggests schools to users based on the information they provide in their Profile Builder, such as Education level, Zone, Subjects, and Co-Curricular Activities.

Priority: High

Priority Component Ratings:

- Benefit: 9 - Provides suitable school options, directly contributing to user satisfaction.
- Penalty: 8 - Without this feature, users may struggle to find schools matching their specific needs, impacting usability.
- Cost: 6 - Moderate to high, requiring development resources for recommendation logic and updates.
- Risk: 5 - Moderate risk, mainly around accuracy and the need for updated data to ensure relevant recommendations.

4.2.5.2.2 Stimulus/Response Sequences

1. User Action: The user clicks on “RECOMMENDATIONS” in the Navbar
 - 1.1. System Response: The system generates and displays a list of recommended schools based on what users pick in their profile builder.

4.2.5.2.3 Functional Requirements

REQ-1: The website must provide personalized school recommendations based on user profile information (e.g., Education level, Zone, Subjects, Co-Curricular Activities).

REQ-2: The recommendation logic must retrieve the latest school data to ensure relevance in suggestions.

REQ-3: The website must allow users to view the full list of recommendations, with the option to explore more details about each school.

4.3 Admin Features

The Admin feature allows administrators to oversee and moderate the forum, ensuring that discussions remain constructive and adhere to community guidelines.

4.3.1 View Flagged Post

4.3.1.1 Description and Priority

Description:

- The view Flagged Post feature allows administrators to review forum posts that have been flagged by users for potential violations of community guidelines.

Priority: High

Priority Component Ratings:

- Benefit: 8 - Improves content moderation efficiency.
- Penalty: 7 - Without this feature, flagged posts may go unnoticed, diminishing user trust and community quality.
- Cost: 4 - Moderate development cost for creating a flagged posts view with associated actions.
- Risk: 5 - Moderate risk, primarily related to ensuring fair moderation and accurate tracking of flagged posts.

4.3.1.2 Stimulus/Response Sequences

1. User Action: The admin selects “View Flagged Posts”.
 - 1.1. System Response: The system displays a list of all posts flagged by users, including details such as report reason.

4.3.1.3 Functional Requirements

REQ-1: The system must provide a “View Flagged Posts” button accessible only to the admin.

REQ-2: The system must display flagged posts in a list format, including details like post content, author, and reason for the flag.

4.3.2 Unflag Posts

4.3.2.1 Description and Priority

Description:

- The unflag post feature allows the administrator to remove flagged status of a forum post after reviewing and determining that it does not violate community guidelines.

Priority: Medium

Priority Component Ratings:

- Benefit: 7 - Enhances moderation efficiency.
- Penalty: 5 - Without this feature, flagged posts deemed appropriate would continue to be flagged.

- Cost: 4 - Low development cost, as it extends existing flagging functionality.
- Risk: 3 - Low risk, as it only removes the flag status without deleting or modifying content.

4.3.2.2 Stimulus/Response Sequences

1. User Action: The admin reviews a flagged post and decides it is appropriate.
 - 1.1. System Response: The system provides an “Unflag” button to clear the flag on the post.

4.3.2.3 Functional Requirements

REQ-1: The system must provide an “Unflag” button for each flagged post that administrators can select after reviewing the content.

REQ-2: The system must update the flagged status of the post immediately upon selecting “Unflag,” removing it from the flagged post list.

4.3.3 Delete comments

4.3.3.1 Description and Priority

Description:

- The Delete comments feature allows administrators to remove inappropriate or offensive comments from forum discussions.

Priority: High

Priority Component Ratings:

- Benefit: 9 - Helps keep the forum appropriate for younger audiences by allowing prompt removal of harmful or unwanted content.
- Penalty: 9 - Without this feature, inappropriate content could persist, negatively impacting the community.
- Cost: 5 - Moderate development cost for implementing the feature.
- Risk: 6 - Moderate risk, as it involves ensuring fair moderation and handling potential disputes over deleted content.

4.3.3.2 Stimulus/Response Sequences

1. User Action: The admin identifies a comment that violates guidelines and clicks the “Delete” button.

- 1.1. System Response: The system deletes the comment from the post.

4.3.3.3 Functional Requirements

REQ-1: The system must provide a “Delete” button for each comment visible to administrators in the forum.

REQ-2: The system must update the comment immediately upon deletion, ensuring users no longer see the removed comment.

4.3.4 Delete Flagged Post

4.3.4.1 Description and Priority

Description:

- The delete post feature allows administrators to remove posts from the forum that have been flagged by other users.

Priority: High

Priority Component Ratings:

- Benefit: 9 - Helps keep the forum appropriate for younger audiences by allowing prompt removal of harmful or unwanted content.
- Penalty: 9 - inappropriate content could persist, negatively impacting the community.
- Cost: 5 - Moderate development cost for implementing the feature.
- Risk: 7 - Moderate-to-high risk due to the importance of accurate and fair content moderation and potential user disputes.

4.3.4.2 Stimulus/Response Sequences

1. User Action: The admin reviews a flagged post and clicks on the “Delete” button.
 - 1.1. System Response: The system prompts the admin for confirmation before deleting the post permanently.
2. User Action: The admin confirms the deletion by clicking “Yes, Delete” on the confirmation prompt.
 - 2.1. System Response: The system removes the post from the forum, updates the forum to reflect the deletion
3. User Action: The admin cancels deletion by clicking “Cancel” on the confirmation prompt.
 - 3.1. System Response: The system closes the confirmation prompt without deleting the post.

4.3.4.3 Functional Requirements

REQ-1: The website must provide an “Delete” button for each post, visible to the admin in the flagged posts page

REQ-2: The website must display a confirmation prompt when the admin clicks the delete button.

REQ-3: The website must require the admin to confirm or cancel deletion before proceeding.

REQ-4 Upon confirmation, the website must permanently delete the post from the forum database and remove it from all views.

REQ-5: The website must immediately update the forum page to reflect the deletion.

5. Other Nonfunctional Requirements

5.1 Legal Requirements

- 1) Ensure compliance with the Personal Data Protection Act (PDPA) when handling users' sensitive data, especially for minors.
- 2) Adhere to local education regulations and policies related to school selection and data handling, ensuring accurate and responsible information is provided to users.

5.2 Reuse Requirements

- 1) Modular Component Design: Components developed for the platform should be modular, allowing for easy reuse to enhance maintainability.

5.3 Usability Requirements

- 1) User-Centric Design: Conduct user research and usability testing to ensure the application is intuitive and meets the needs of its target audience, including parents, students, and educators.
- 2) Responsive Design: Ensure the application is responsive and accessible across various devices (desktop, tablet, mobile) and screen sizes.

5.4 Internationalisation Requirements

- 1) Language Support: Initially, the **SchoolPicker** will support English, with future releases considering multi-language support to cater to diverse user groups, including Mandarin and Malay.
- 2) Date and Time Representation: All date and time representations should adhere to the Singapore timezone (GMT+8) and accommodate local date formats.

5.5 Business Rules

- 1) User Roles and Permissions
 - a) Only Admins can manage forum activities, including viewing of flagged posts and reasons, and deletion of posts or comments of other users.
 - b) Standard Users can access and modify their own data, including their profile information and posts, but cannot view others' profile information or edit others' posts.

- c) Guest Users can only access "Schools" and "Compare Schools" functionalities, without access to "Profile Builder," "Recommend Schools," or "Forum."

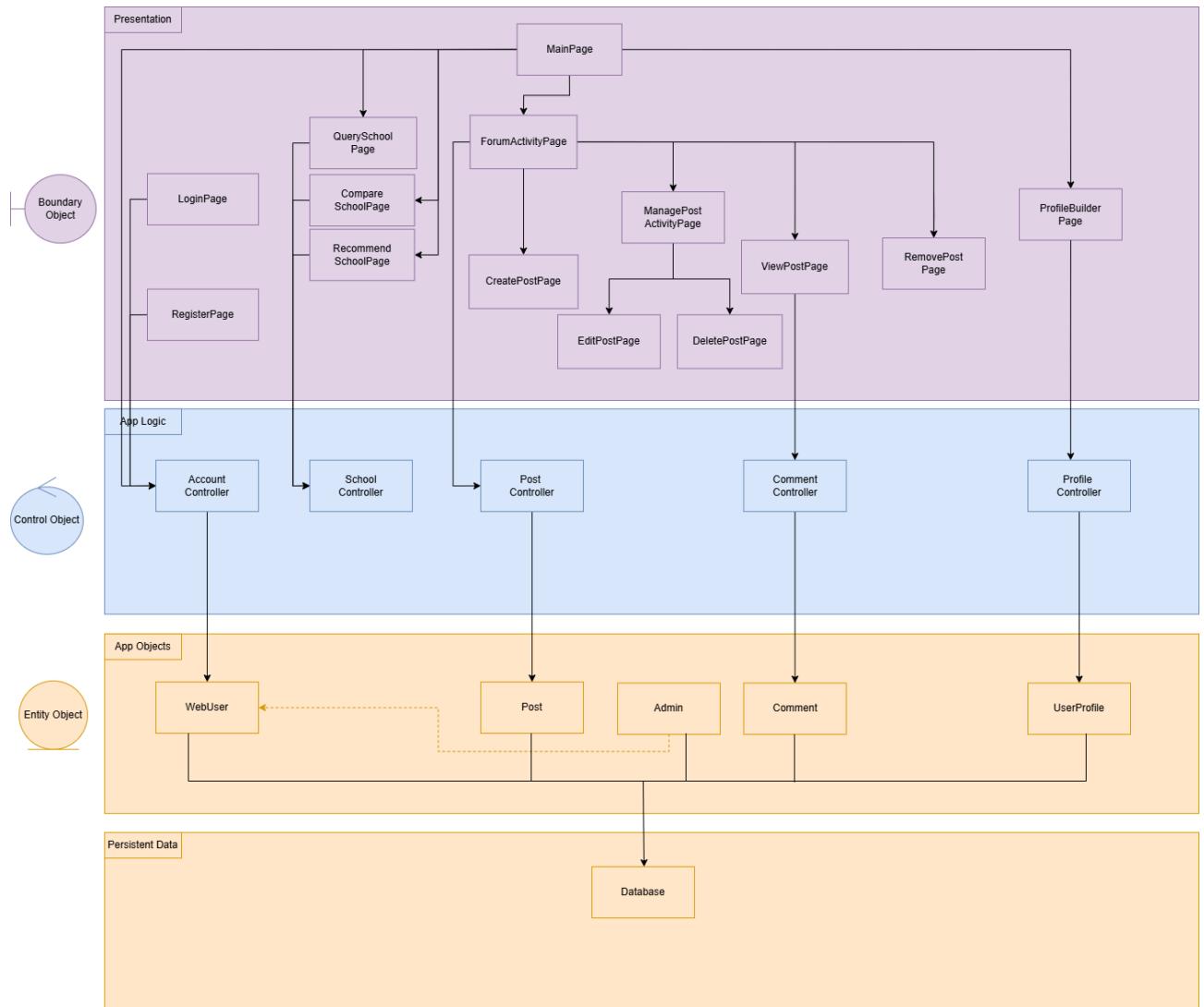
6. Other Requirements

Appendix A: Glossary

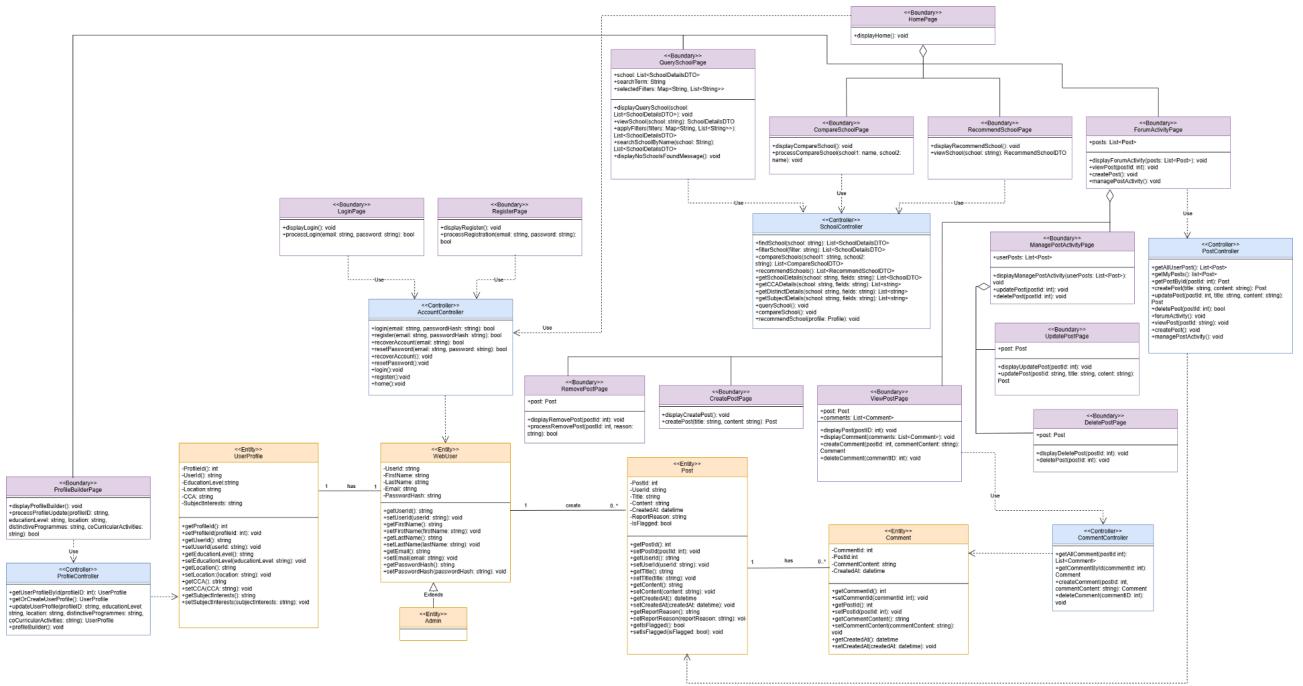
Acronym / Abbreviation	Meaning
PDPA	<p>Personal Data Protection Act</p> <ul style="list-style-type: none"> - A baseline standard for protecting personal data in Singapore
Frontend	<p>Responsible for client-side components that users interact with directly, encompassing all visual elements and user interfaces, such as text, images, buttons, and navigation. It focuses on delivering a seamless and engaging user experience through design and interactivity.</p>
Backend	<p>Responsible for server-side components that manage data, application logic, and server interactions. It handles processes like database management, authentication, and server configuration, enabling the frontend to function effectively by providing the necessary data and services behind the scenes.</p>
Forum	<p>An online platform where User can post messages, ask questions, and engage in discussions on various topics. Forums are often organized into different categories or sections based on subject matter, such as technology, hobbies, education, or health, allowing User to find and participate in conversations relevant to their interests.</p>
CCAs	<p>Co-Curricular Activities</p> <ul style="list-style-type: none"> - A vital part of a student's education in Singapore, where they are a core component of a holistic education. CCAs are non-academic activities that take place outside of the classroom, but they reinforce or supplement the classroom curriculum.

SRS	<p>Software Requirement Specification</p> <ul style="list-style-type: none">- A detailed document that outlines the functional and non-functional requirements of a software system. It serves as a blueprint for development, providing a clear description of the software's intended features, performance metrics, user interactions, and constraints.
User	Individual who interact with the system. User may include prospective student, current student, parent or other stakeholders seeking information on secondary schools and academic programs.
System Administrator	Individual who manage the system. Have the privilege to perform specific tasks on the system normal User do not have.
Co-Curricular Activities (CCAs)	Co-Curricular Activities (CCAs) are a vital part of a student's education in Singapore, where they are a core component of a holistic education. CCAs are non-academic activities that take place outside of the classroom, but they reinforce or supplement the classroom curriculum.

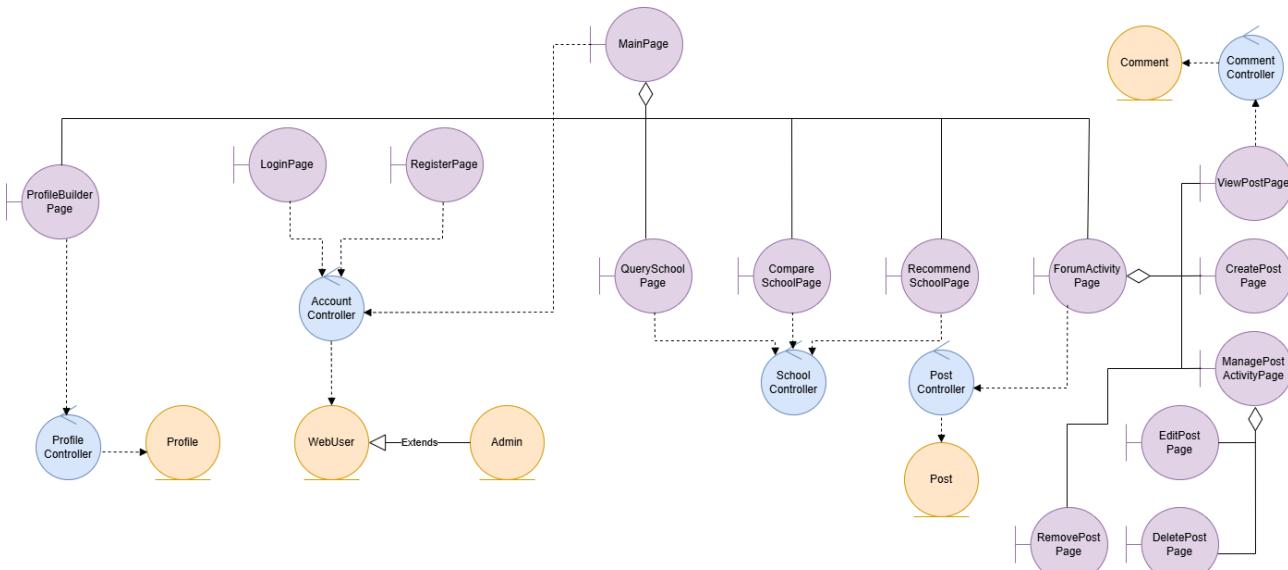
Appendix B: Analysis Models



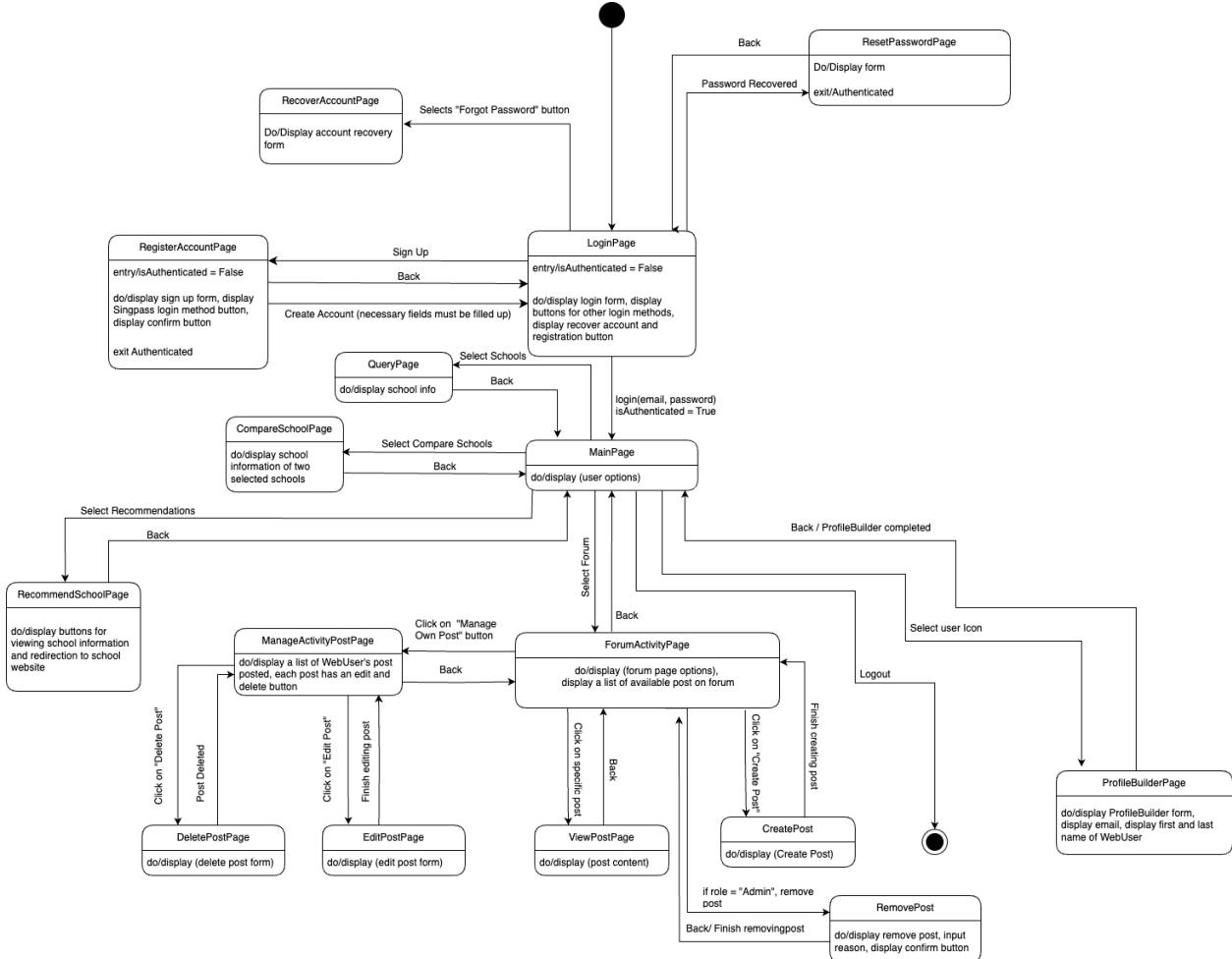
Architecture Diagram



Class Diagram



Stereotype Class Diagram

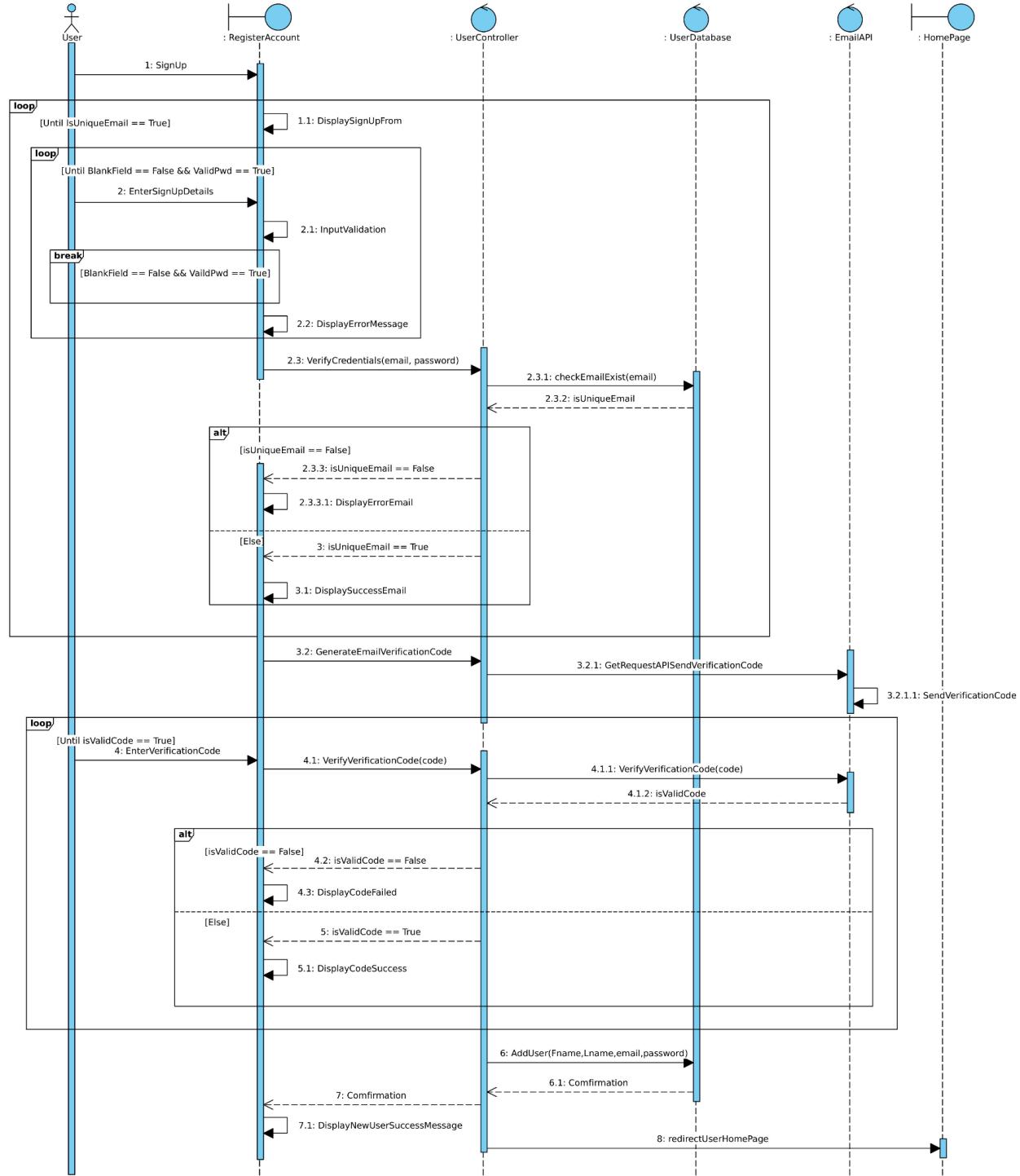


Dialog Map

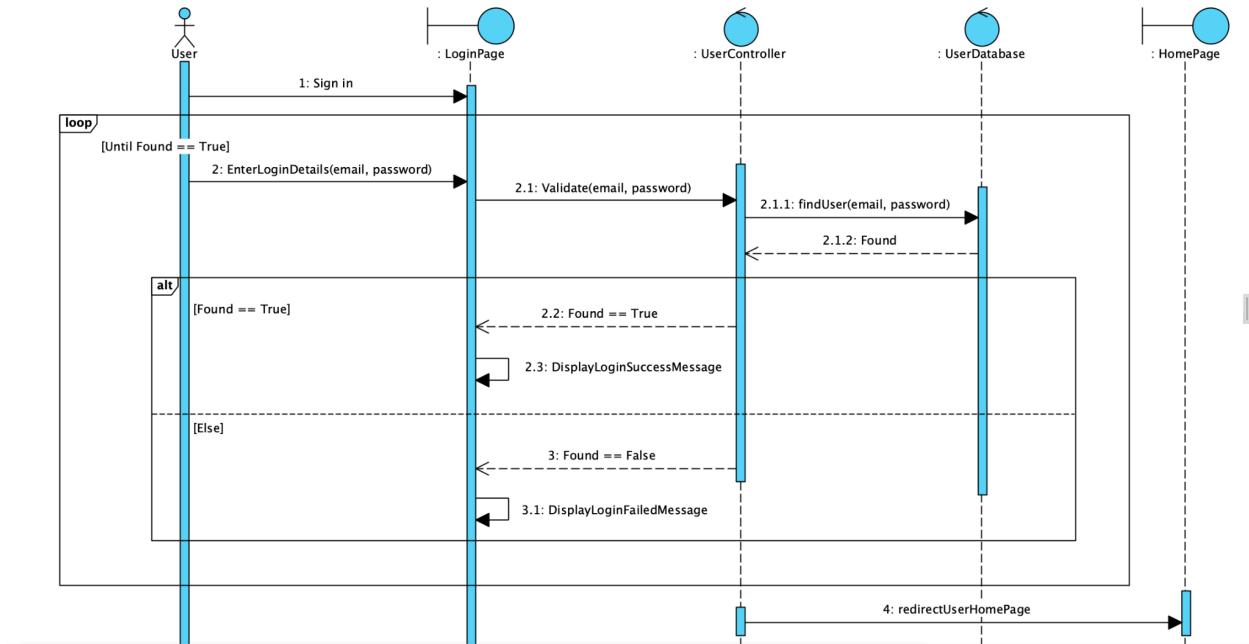
Appendix C: Sequence Diagrams

Functional Requirement 1

a. RegisterAccount

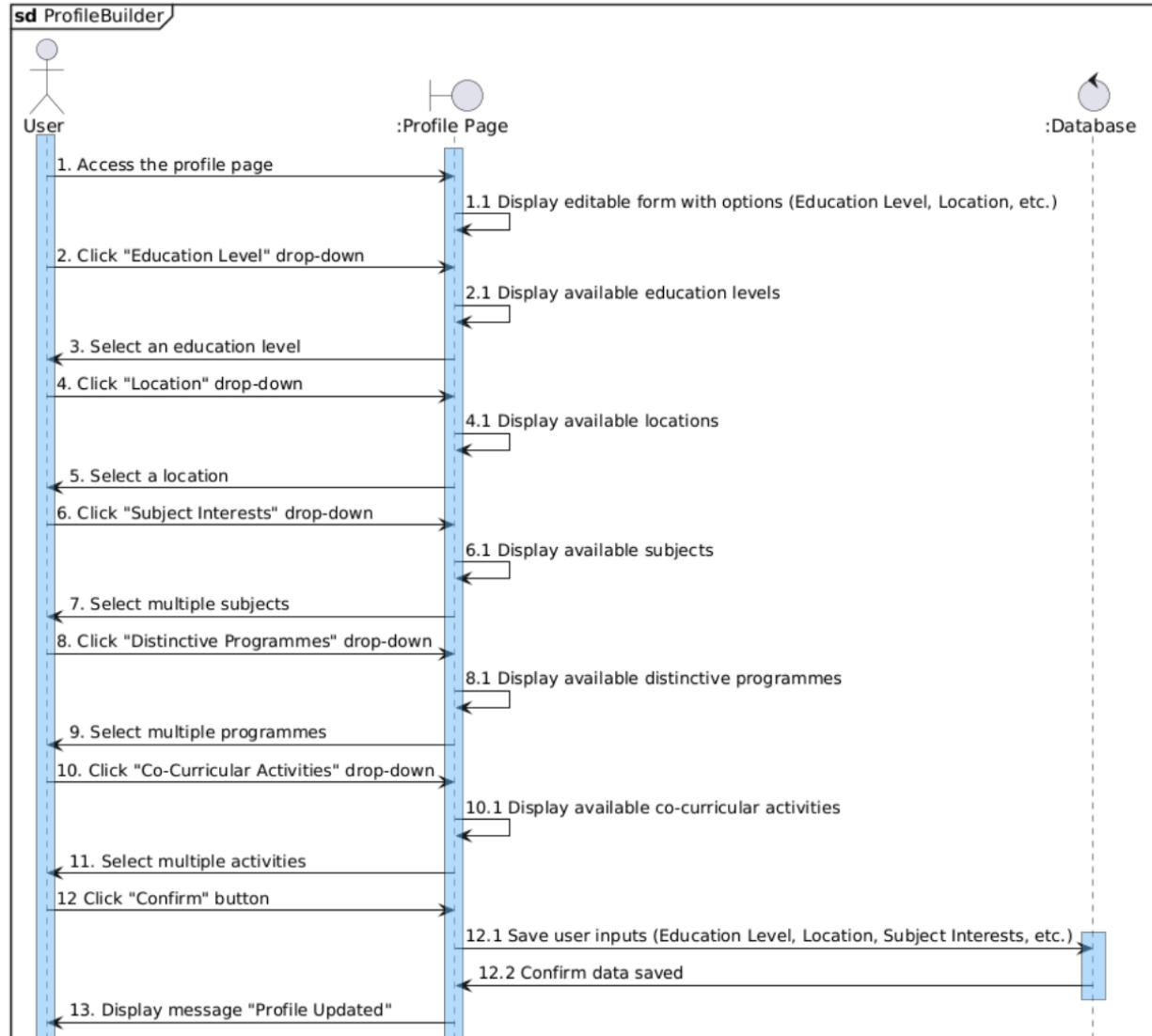


b. Login



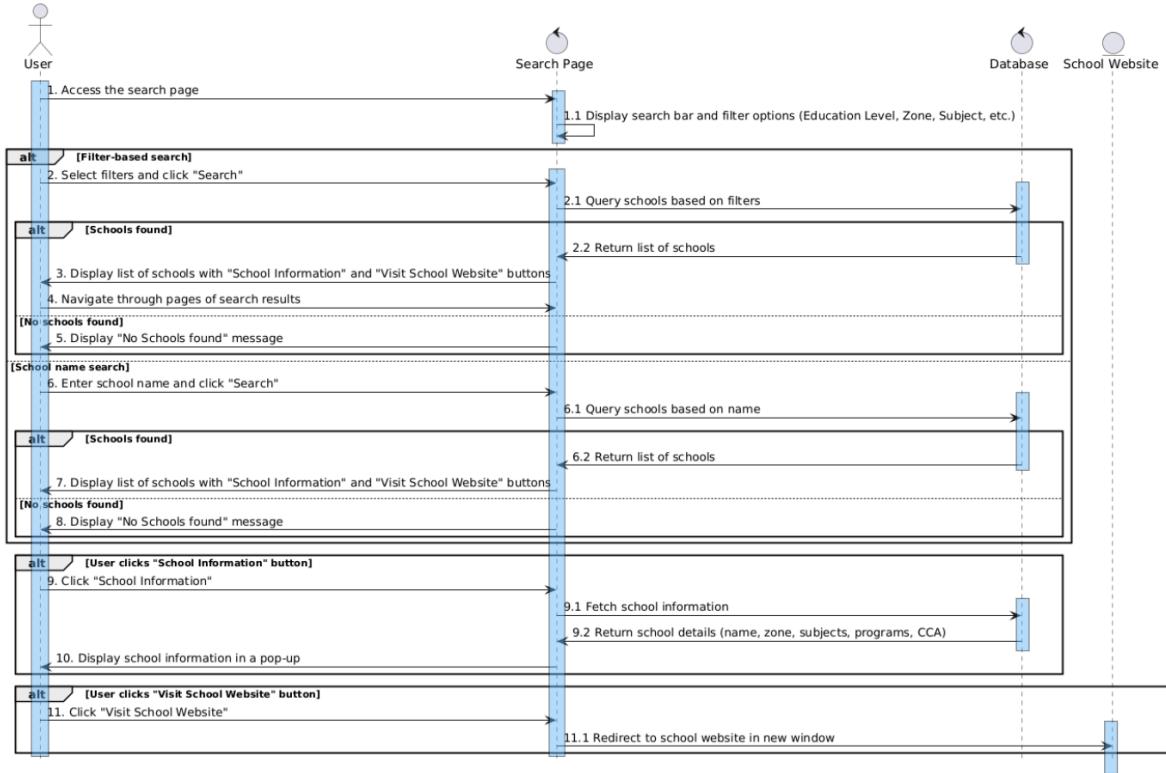
Functional Requirement 2

a. ProfileBuilderActivity



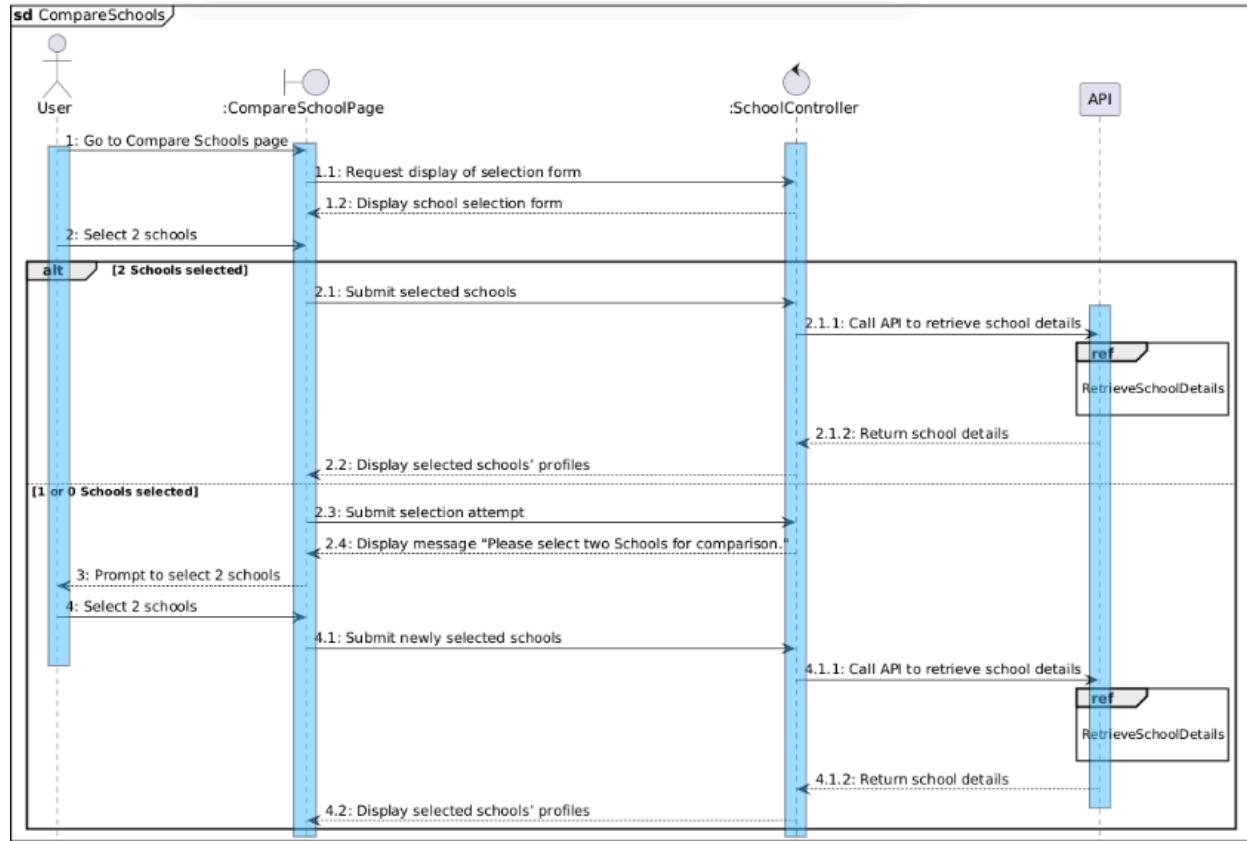
Functional Requirement 3

a. FindSchool



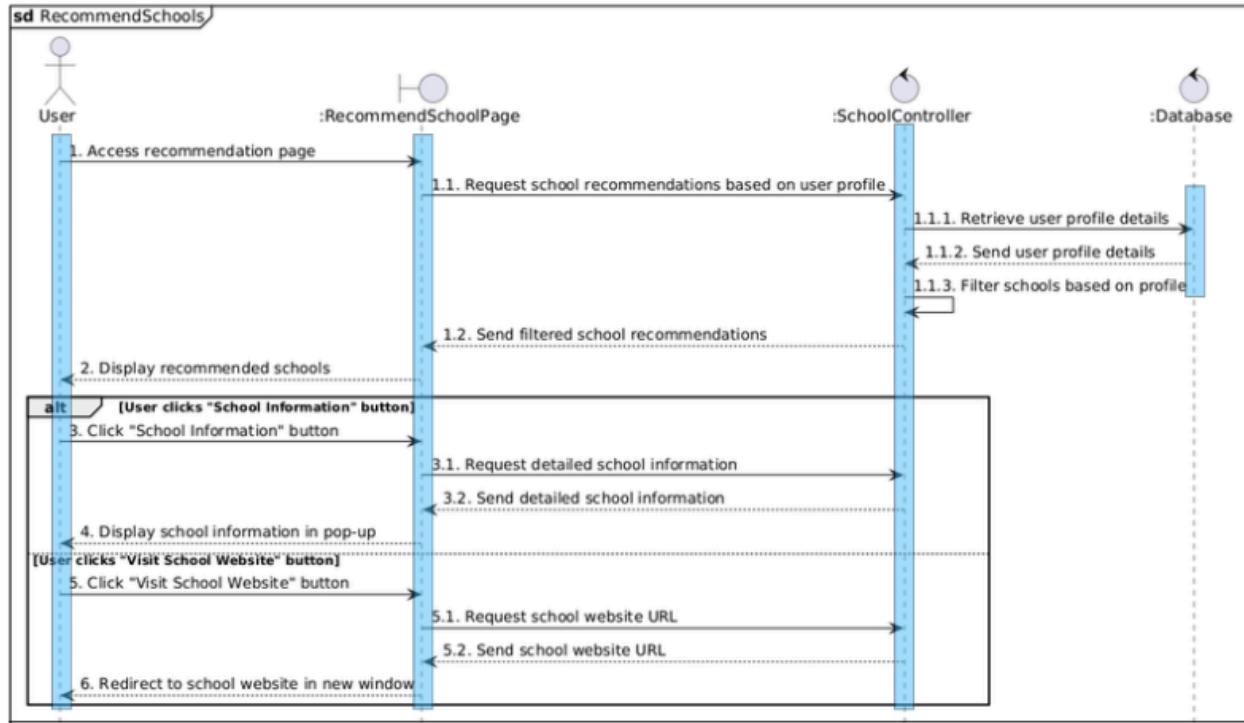
Functional Requirement 4

a. CompareSchool



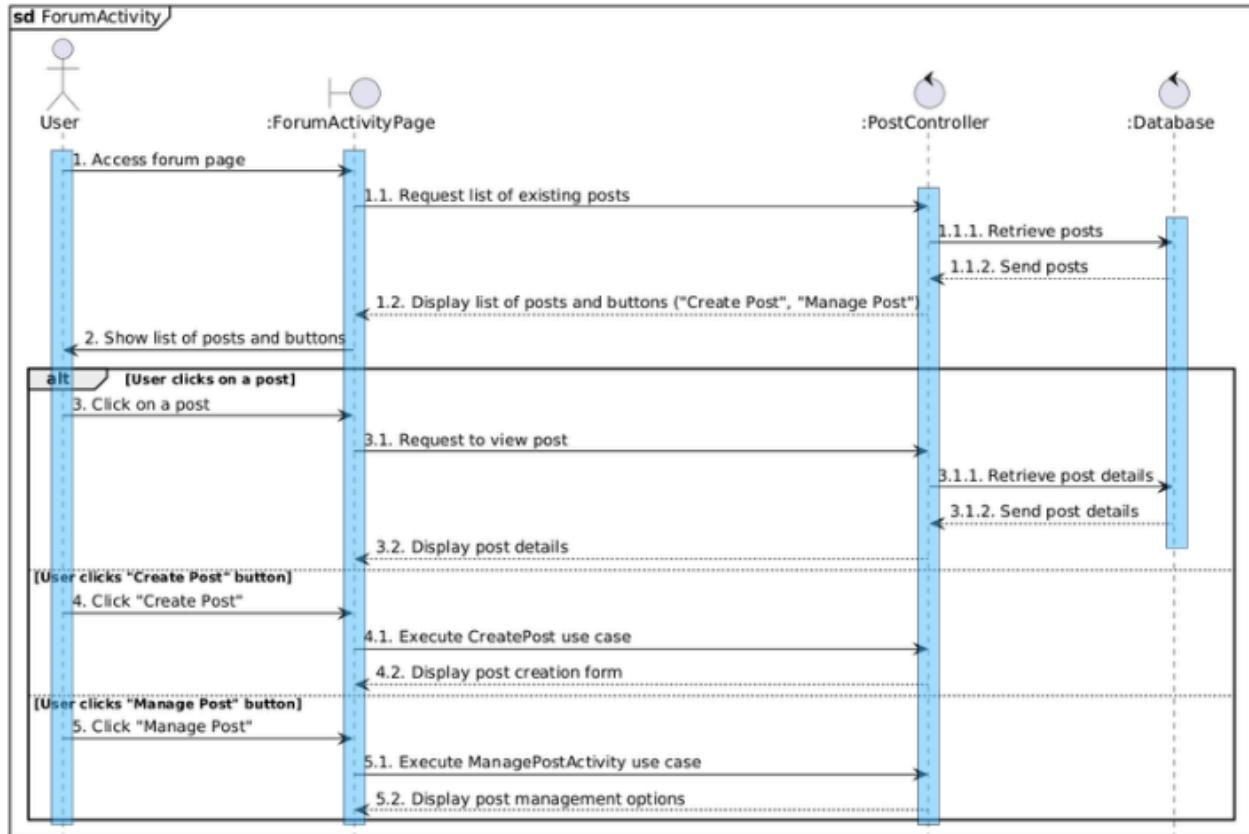
Functional Requirement 5

a. RecommendSchool

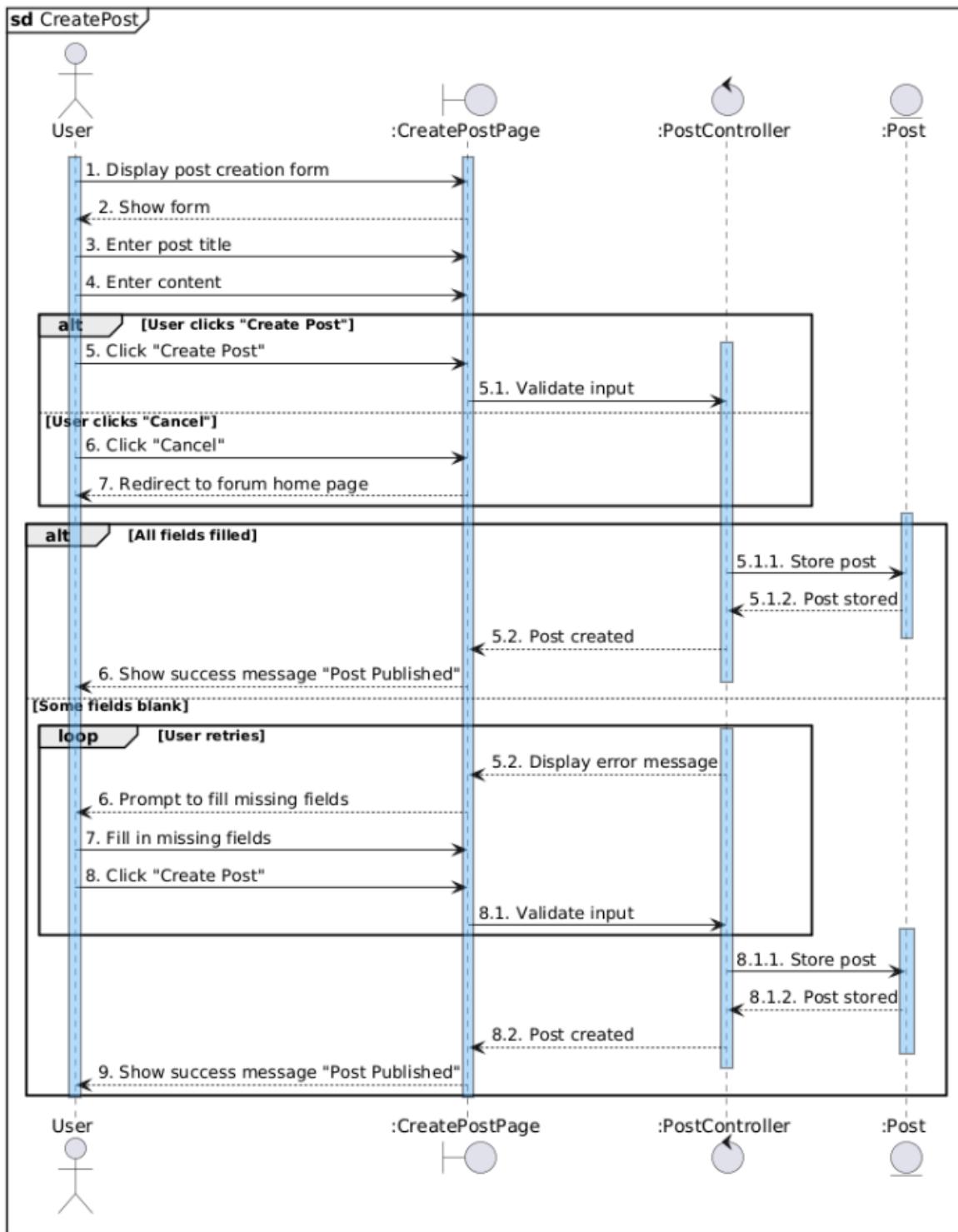


Functional Requirement 6

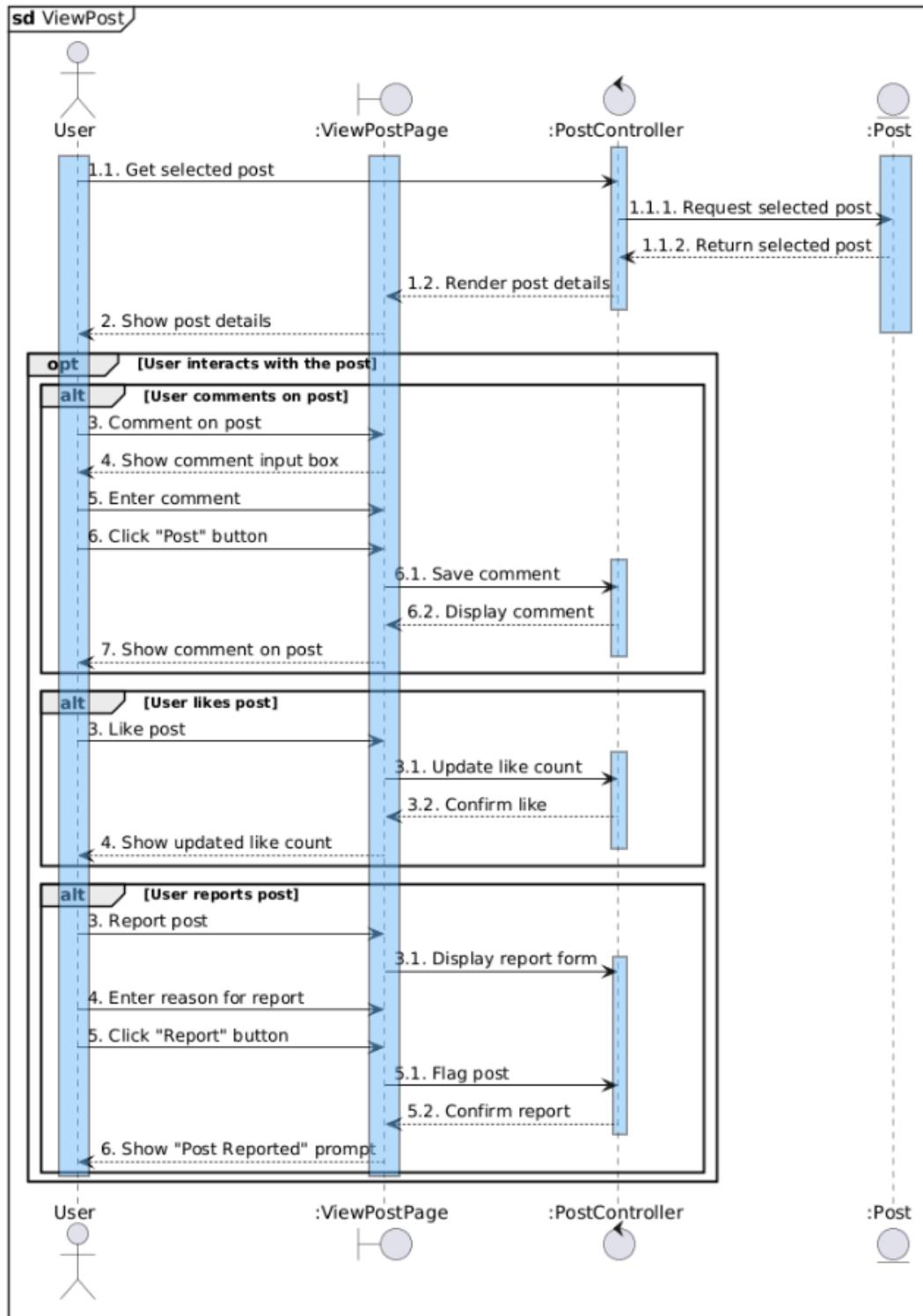
a. ForumActivity



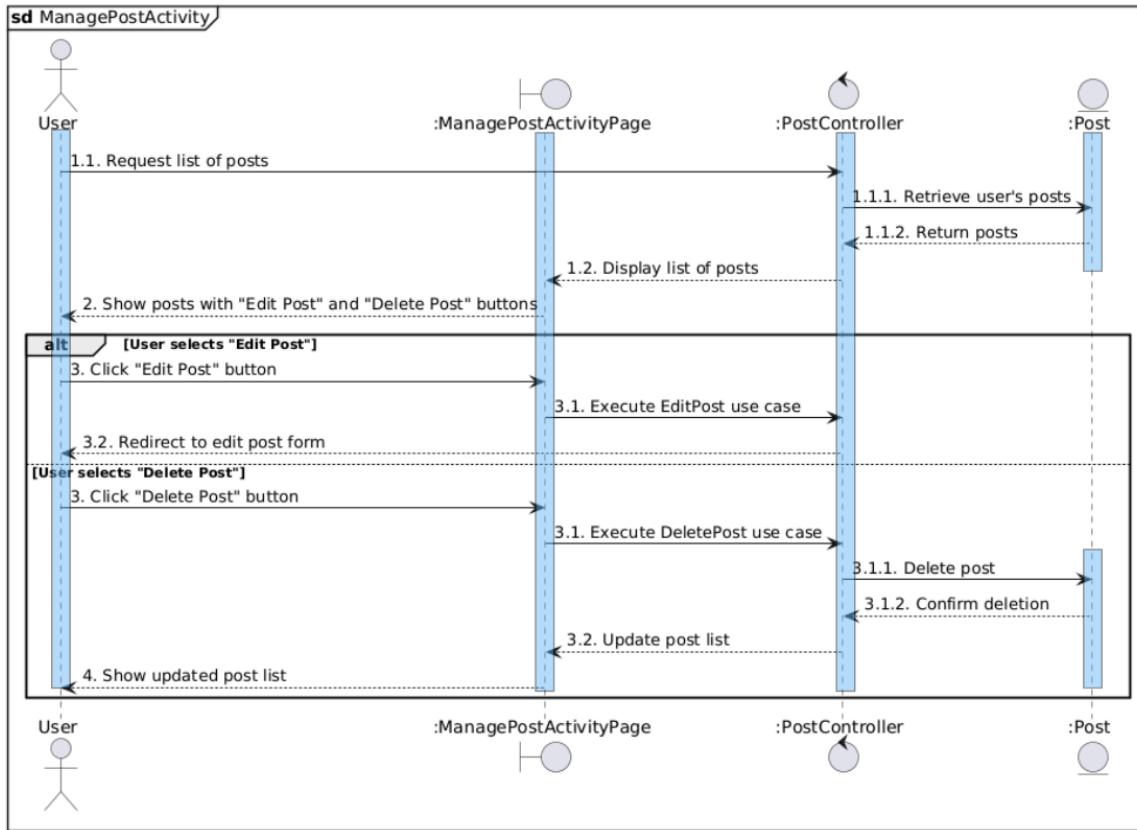
b. CreatePost



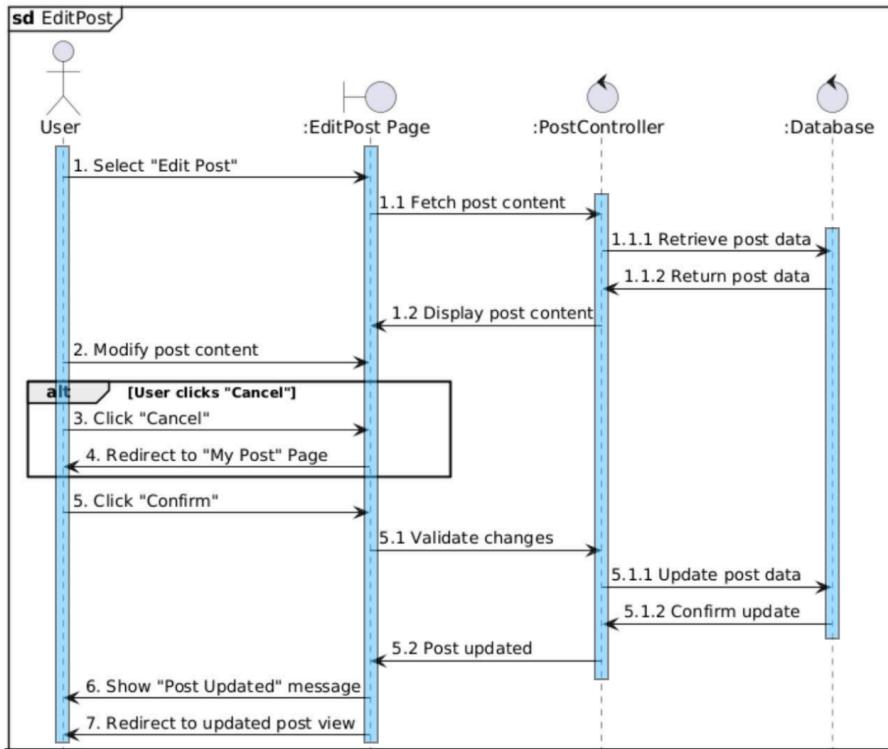
c. ViewPost



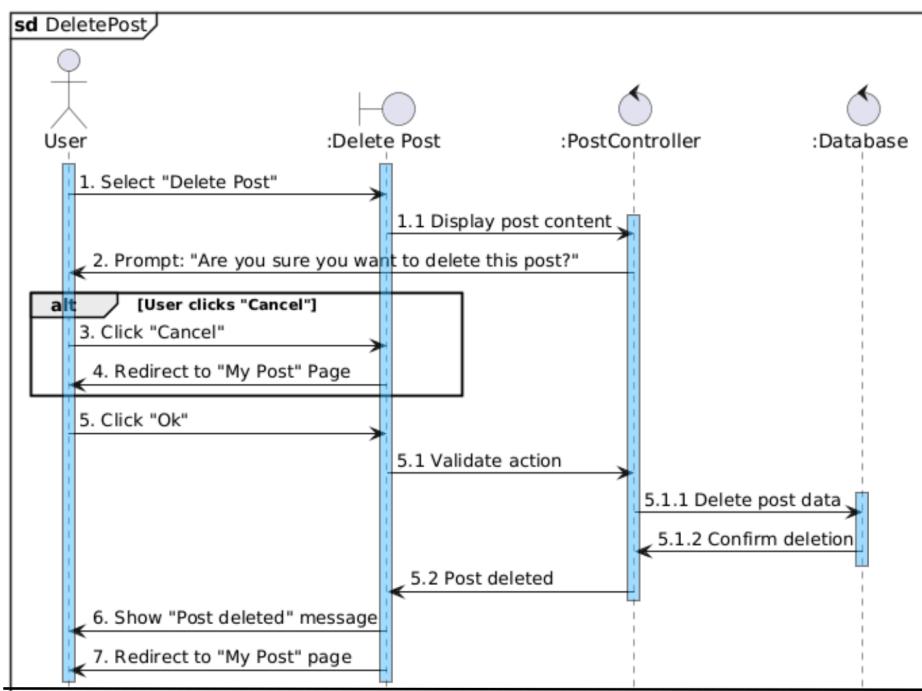
d. ManagePostActivity



e. EditPost

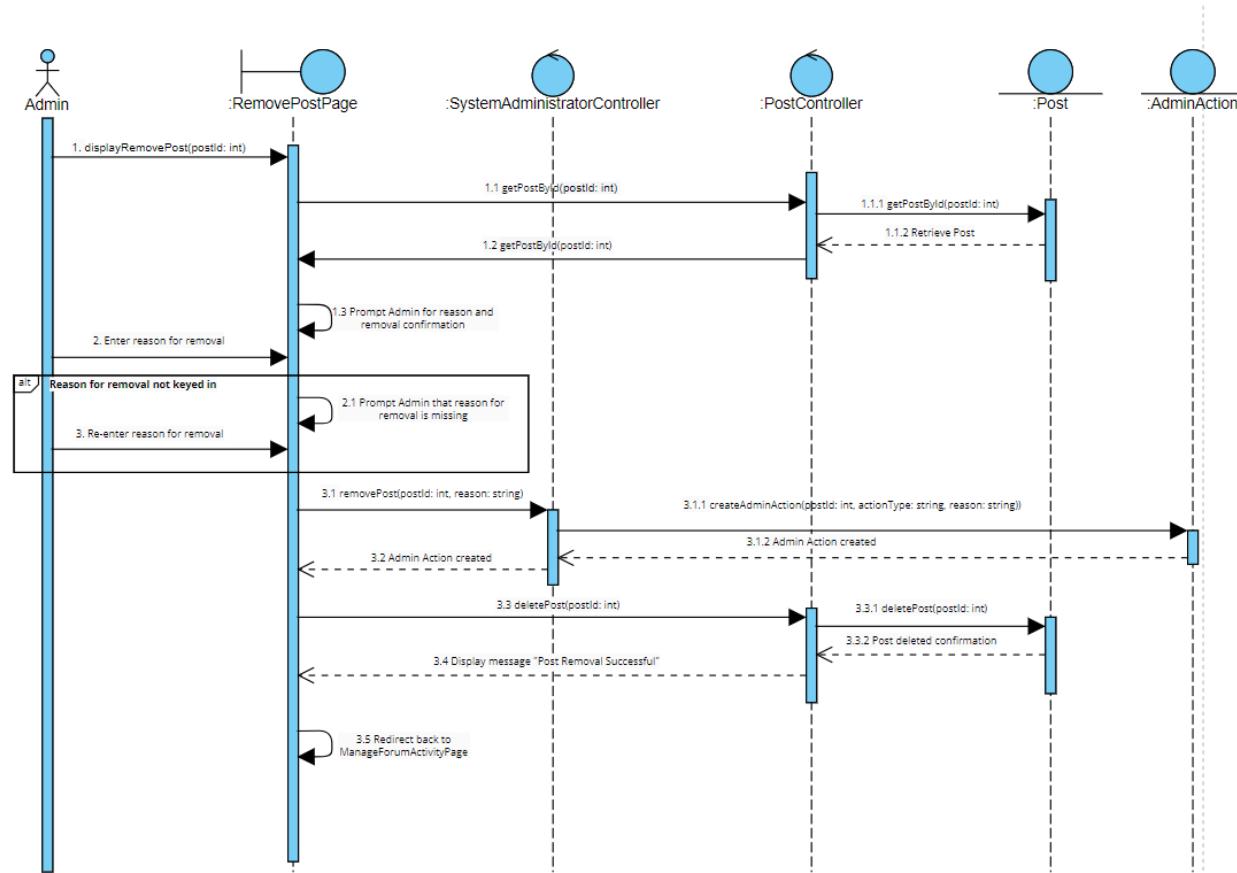


f. DeletePost



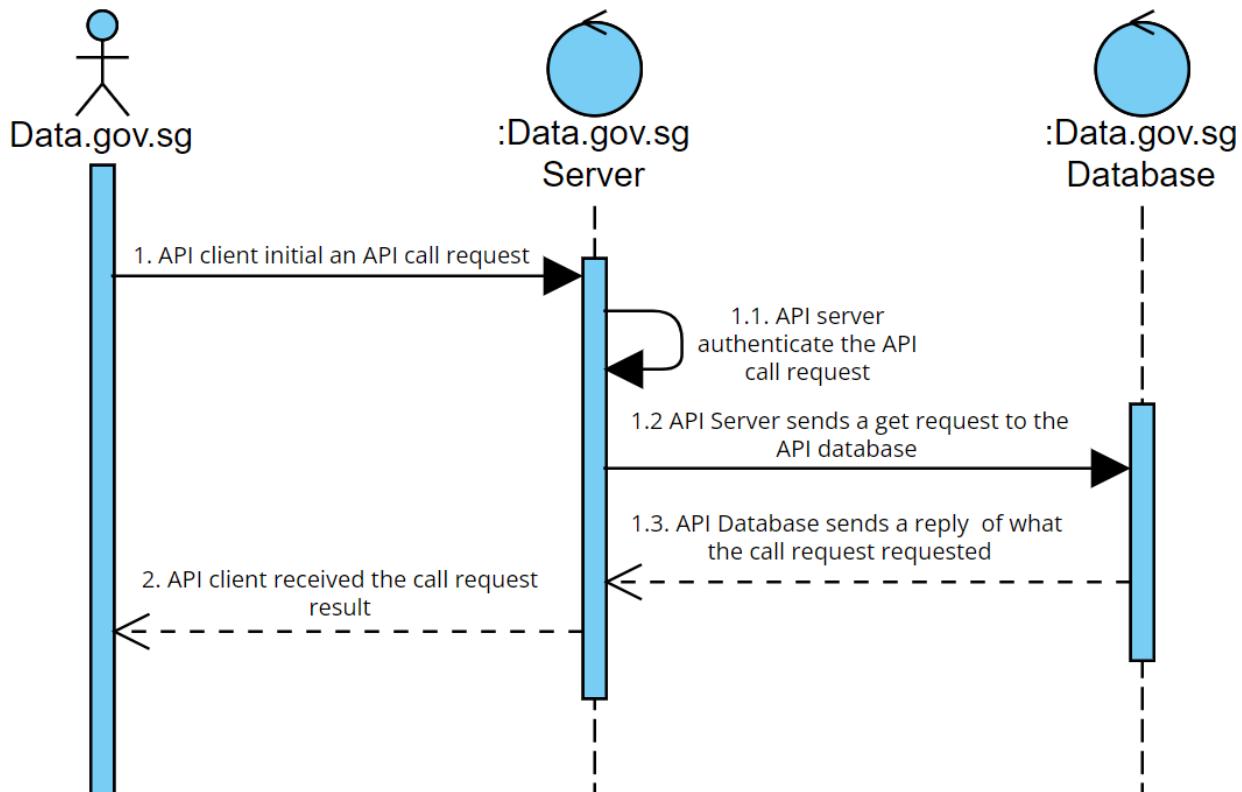
Functional Requirement 7

a. RemovePost



API Function

a. RetrieveSchoolDetails



Appendix D: Test Cases

1. Black Box Testing

I. AccountController - controller we are testing

The accountController manages any request for user authentication, including **register** and **login**.

When a user attempts to register, the controller requires the user to enter their first name, last name, email, password and re-confirmation of their password. Upon entering of the data, it would be validated for example whether a valid email or password is given. Prior to adding the user's password into the database, the password would be hashed.

When a user attempts to login, the controller would require the user to enter their email and password which would be validated for correctness. To authenticate the user, the controller would

get the hashed password from the database and compare it with the user's provided password. If the password matches, the user would be redirected to the home page.

II. Equivalence Class and Boundary Value Testing

Equivalence Class Testing

For the web users:

Email variables being used for the signup function cannot be repeated, all emails must be unique. Attempting to signup for a web user using an email that exists in the system database will not create a new webuser account. The email variable also checks for the format of the input to see if it is a properly formatted email input. If the email input variable is not a proper format for an email, the webuser account will not be created.

Password variables used for the webuser have a structure to follow to ensure the password created is strong (fulfills our password requirement criteria). The password restriction used in this application checks if the password is at least 8 characters long and includes the usage of at least 1 lowercase, uppercase letter and a number. Passwords not matching this requirement will be rejected and the webuser account will not be created.

First name and last name variables will be checked to see if they are empty or null. If these fields are empty, webuser will not be created.

Boundary Value Testing

1. Signup Function

Valid Equivalence Class: First name, Last name, Email, Password, Confirm Password, Terms & Conditions with correct formats and fulfills conditions.

Invalid equivalence Class: First name, Last name, Email, Password, Confirm Password, Terms & Conditions with incorrect formats or missing conditions.

2. Sign in Function

Valid Equivalence Class: Email, password with correct formats and fulfills conditions.

Invalid Equivalence Class: Email, password with incorrect formats or missing conditions.

Equivalence Class	Boundary Conditions
Password	Minimum length of 8 (Minimum boundary) Contains the following character variation: A, a, 1 Password123 (accepted) Password123 (rejected) Password (rejected) Pass1 (rejected)
Email	Must be unique (non-existent in the system) Minimum format a@b.com (Minimum boundary) Formatted with an '@', characters in between, and with a '.' and characters afterward

	user@gmail.com (valid) user.com (rejected) user@ (rejected) user@.com (rejected) jexistedalready@gmail.com (rejected, existed)
--	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------

1) Signup

Input parameters: First name, Last name, Email, Password, Confirm Password, Terms & Conditions

No .	Test Input	Expected Output	Actual Output	Pass ?
1	(All valid Inputs) First Name: "Tang" Last Name: "Guan Yu" Email: "TangGuanYu@gmail.com" Password: "Password123" Confirm Password: >Password123" Terms & Conditions: selected	Successful signup	Successful signup	Pass
2	(All Input Empty) (All invalid Inputs) First Name: "" Last Name: "" Email: "" Password: "" Confirm Password: "" Terms & Conditions: not selected	Unsuccessful signup, system notify: - Email is required -First name is required -Last name is required -Password is required - Confirmation password is required - Please tick this box if you want to proceed.	Unsuccessful signup, system notify: - Email is required -First name is required -Last name is required -Password is required - Confirmation password is required - Please tick this box if you want to proceed.	Pass
3	(All valid inputs except password) (Valid) First Name: "Guan Yu" (Valid) Last Name: "Tang" (Valid) Email: "Supergy@gmail.com" (Invalid) Password: "pass123" (Invalid) Confirm Password: "pass123" (Valid) Terms & Conditions:	Unsuccessful signup, system notify "Password did not meet requirements"	Unsuccessful signup, system notify "Password did not meet requirements"	Pass

	selected			
4	<p>(All valid inputs except confirm password)</p> <p>(Valid) First Name: "Guan Yu" (Valid) Last Name: "Tang" (Valid) Email: "Supergy@gmail.com" (Valid) Password: "Pass1234" (Invalid) Confirm Password: "Pass4567" (Valid) Terms & Conditions: selected</p>	Unsuccessful signup, system notify "Passwords do not match"	Unsuccessful signup, system notify "Passwords do not match"	Pass
5	<p>(All valid inputs except first name)</p> <p>(Invalid) First Name: "" (Valid) Last Name: "Guan Yu" (Valid) Email: "TangGuanYu@gmail.com" (Valid) Password: "Password123" (Valid) Confirm Password: "Password123" (Valid) Terms & Conditions: selected</p>	Unsuccessful signup, system notify "First name is required"	Unsuccessful signup, system notify "First name is required"	Pass
6	<p>(All valid inputs except last name)</p> <p>(Invalid) First Name: "Guan Yu" (Invalid) Last Name: "" (Valid) Email: "TangGuanYu@gmail.com" (Valid) Password: "Pword123" (Valid) Confirm Password: "Pword123" (Valid) Terms & Conditions: selected</p>	Unsuccessful signup, system notify "Last name is required"	Unsuccessful signup, system notify "Last name is required"	Pass
7	<p>(All valid inputs except email)</p> <p>(Valid) First Name: "Guan Yu" (Valid) Last Name: "Tang" (Invalid) Email: "Supergy" (Valid) Password: "Pass1234" (Valid) Password: "Pass1234" (Valid) Terms & Conditions: selected</p>	Unsuccessful signup, system notify "Please include an '@' in the email address. 'Supergy' is missing an '@'."	Unsuccessful signup, system notify "Please include an '@' in the email address. 'Supergy' is missing an '@'."	Pass

8	(All valid inputs except email) (Valid) First Name: "Guan Yu" (Valid) Last Name: "Tang" (Invalid) Email: "Supergy@" (Valid) Password: "Pass1234" (Valid) Password: "Pass1234" (Valid) Terms & Conditions: selected	Unsuccessful signup, system notify "Please enter a part following '@' in the email address. 'e@' is incomplete."	Unsuccessful signup, system notify "Please enter a part following '@' in the email address. 'e@' is incomplete."	Pass
9	(All valid inputs except email) (Valid) First Name: "Guan Yu" (Valid) Last Name: "Tang" (Invalid) Email: "Supergy@gmail" (Valid) Password: "Pass1234" (Valid) Password: "Pass1234" (Valid) Terms & Conditions: selected	Unsuccessful signup, system notify "Email is invalid"	Unsuccessful signup, system notify "Email is invalid"	Pass
10	(All valid Inputs except terms & conditions) (Valid) First Name: "Tang" (Valid) Last Name: "Guan Yu" (Valid) Email: "TangGuanYu@gmail.com" (Valid) Password: "Password123" (Valid) Confirm Password: "Password123" (Invalid) Terms & Conditions: not selected	Unsuccessful signup, system notify "Please tick this box if you want to proceed."	Unsuccessful signup, system notify "Please tick this box if you want to proceed."	Pass

2) Sign In

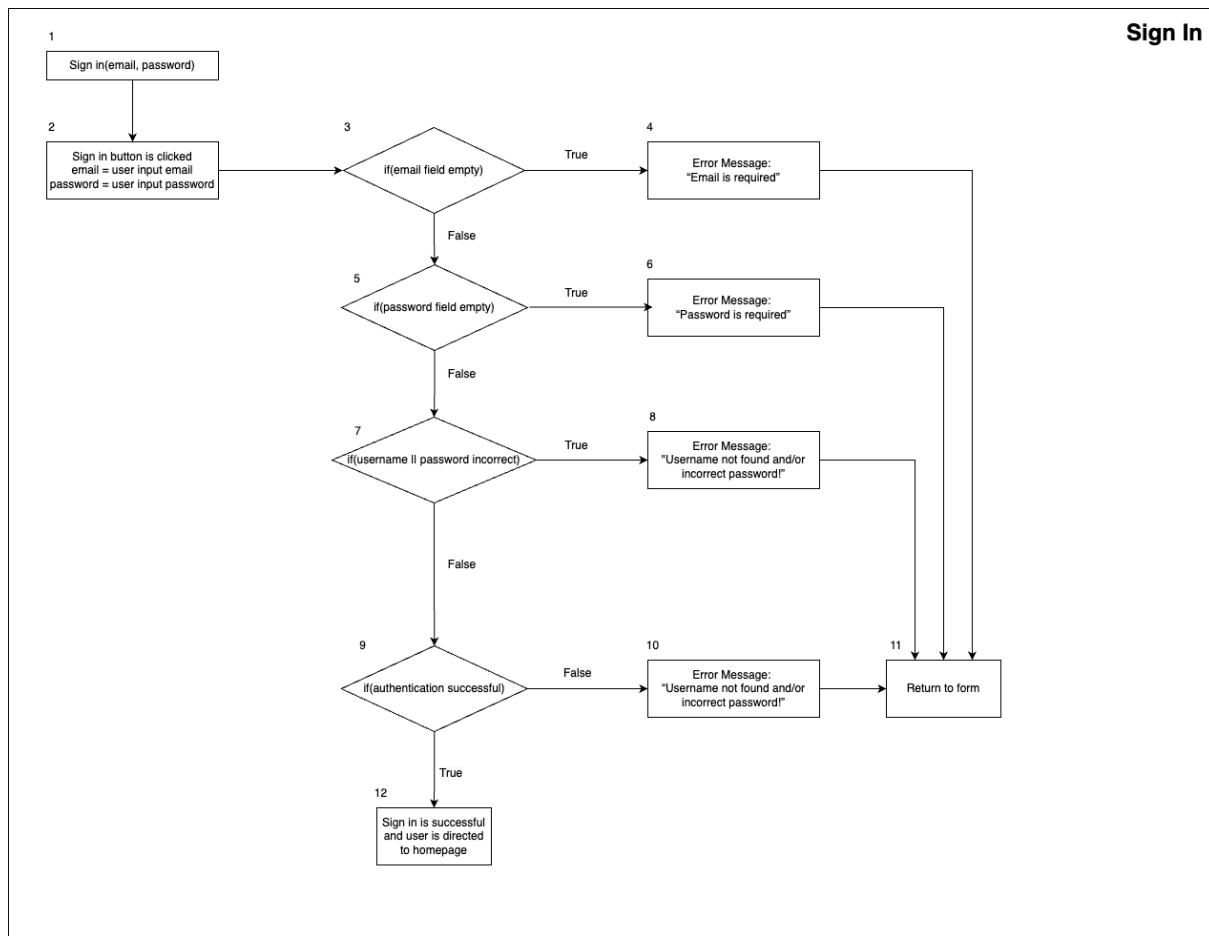
Input parameters: Email, Password

No .	Test Input	Expected Output	Actual Output	Pass ?
1	(Valid) Email: "TangGuanYu@gmail.com" (Valid) Password: "Password123"	Successful sign in	Successful sign in	Pass
2	(Invalid) Email: "TangGuanYu@gmail.com" (Invalid) Password: "Password124"	Unsuccessful sign in, system notify "Username not found and/or incorrect"	Unsuccessful sign in, system notify "Username not found and/or incorrect"	Pass

		password!"	password!"	
3	(Valid) Email: “ <u>SuperGuanYu@gmail.com</u> ” (Invalid) Password: “”	Unsuccessful sign in, system notify “Password is required”	Unsuccessful sign in, system notify “Password is required”	Pass
4	(Invalid) Email: “HandsomeGuanYu” (Valid) Password: “Password123”	Unsuccessful sign in, system notify “Email is invalid”	Unsuccessful sign in, system notify “Email is invalid”	Pass
5	(All Input Empty) (All invalid Inputs) Email: “” Password: “”	Unsuccessful sign in, system notify “Email is required” “Password is required”	Unsuccessful sign in, system notify “Email is required” “Password is required”	Pass

2. White Box Testing

Sign In



Cyclomatic Complexity = $4 + 1 = 5$

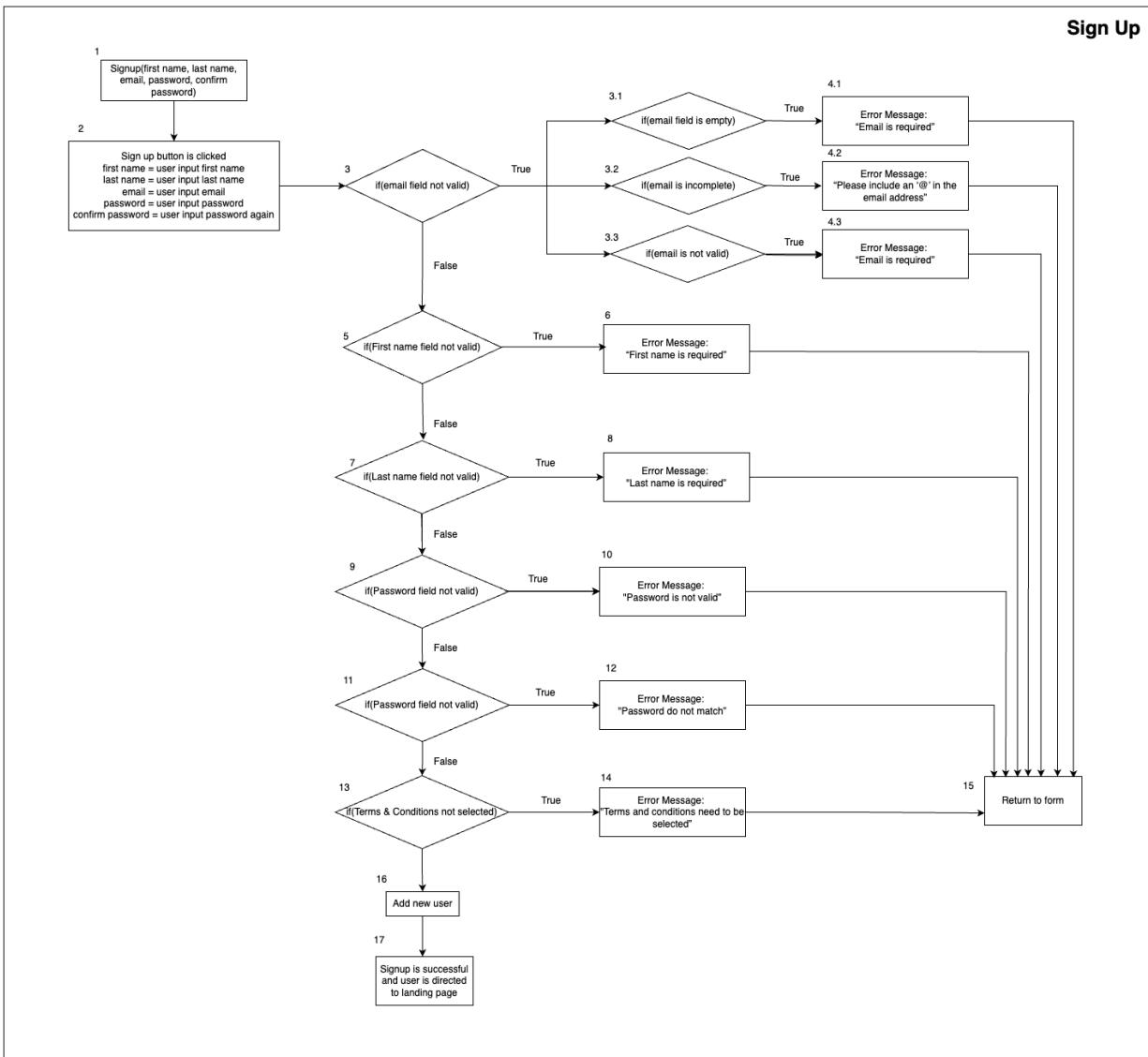
Basic Paths

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

Sign In

Path	Sequence	Description
1	1 → 2 → 3 (True) → 4 → 11	Email field is empty; displays "Email is required" error and returns to the form.
2	1 → 2 → 3 (False) → 5 (True) → 6 → 11	Email is filled, but password field is empty; displays "Password is required" error and returns.
3	1 → 2 → 3 (False) → 5 (False) → 7 (True) → 8 → 11	Email and password are filled, but authentication fails; displays "Incorrect credentials" error.
4	1 → 2 → 3 (False) → 5 (False) → 7 (False) → 9 (False) → 10 → 11	Email and password are filled; authentication is not successful, display "Username not found and/or incorrect password".
5	1 → 2 → 3 (False) → 5 (False) → 7 (False) → 9 (True) → 12	Email and password are filled; authentication is successful, and the user is directed to the homepage.

Sign Up



Cyclomatic Complexity = $8 + 1 = 9$

Basic Paths

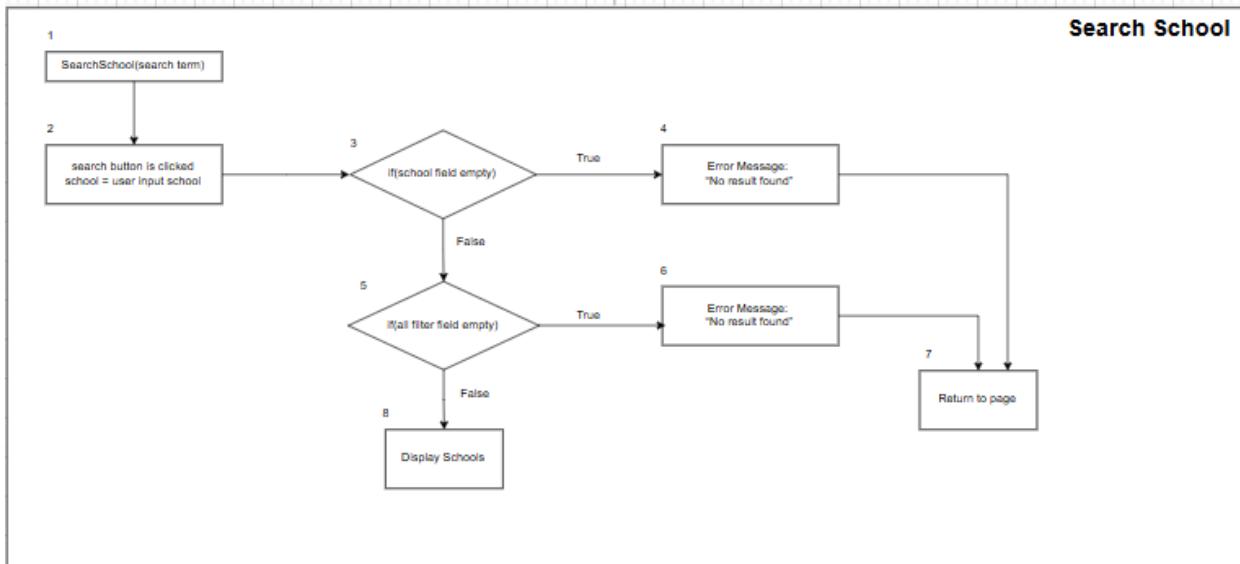
1. Basispath 1: 1, 2, 3, 3.1, 4.1, 15
2. Basispath 2: 1, 2, 3, 3.2, 4.2, 15
3. Basispath 3: 1, 2, 3, 3.3, 4.3, 15
4. Basispath 4: 1, 2, 3, 5, 6, 15
5. Basispath 5: 1, 2, 3, 5, 7, 8, 15
6. Basispath 6: 1, 2, 3, 5, 7, 9, 10, 15
7. Basispath 7: 1, 2, 3, 5, 7, 9, 11, 12, 15
8. Basispath 8: 1, 2, 3, 5, 7, 9, 11, 13, 14, 15
9. Baseline path: 1, 2, 3, 5, 7, 9, 11, 13, 16, 17

Pat	Sequence	Description
-----	----------	-------------

h		
1	1 → 2 → 3 (True at 3.1) → 4.1 → 15	The email field is empty; displays "Email is required" error and returns to the form.
2	1 → 2 → 3 (False) → 3.2 (True) → 4.2 → 15	The email field is incomplete; displays "Please include an '@' in the email address" error and returns to the form.
3	1 → 2 → 3 (False) → 3.3 (True) → 4.3 → 15	The email is provided but invalid; displays "Email is not valid" error and returns to the form.
4	1 → 2 → 3 (False) → 5 (True) → 6 → 15	The first name field is invalid; displays "First name is required" error and returns to the form.
5	1 → 2 → 3 (False) → 5 (False) → 7 (True) → 8 → 15	The last name field is invalid; displays "Last name is required" error and returns to the form.
6	1 → 2 → 3 (False) → 5 (False) → 7 (False) → 9 (True) → 10 → 15	The password field is invalid; displays "Password is not valid" error and returns to the form.
7	1 → 2 → 3 (False) → 5 (False) → 7 (False) → 9 (False) → 11 (True) → 12 → 15	The passwords do not match; displays "Password does not match" error and returns to the form.
8	1 → 2 → 3 (False) → 5 (False) → 7 (False) → 9 (False) → 11 (False) → 13 (True) → 14 → 15	The terms and conditions are not selected; displays "Terms and conditions need to be selected" error and returns to the form.
9	1 → 2 → 3 (False) → 5 (False) → 7 (False) → 9 (False) → 11 (False) → 13 (False) → 16 → 17	All fields are valid; user is successfully signed up and directed to the landing page.

3. Other Test Cases

This section contains the Whitebox testing of specific functions on the application SearchSchool



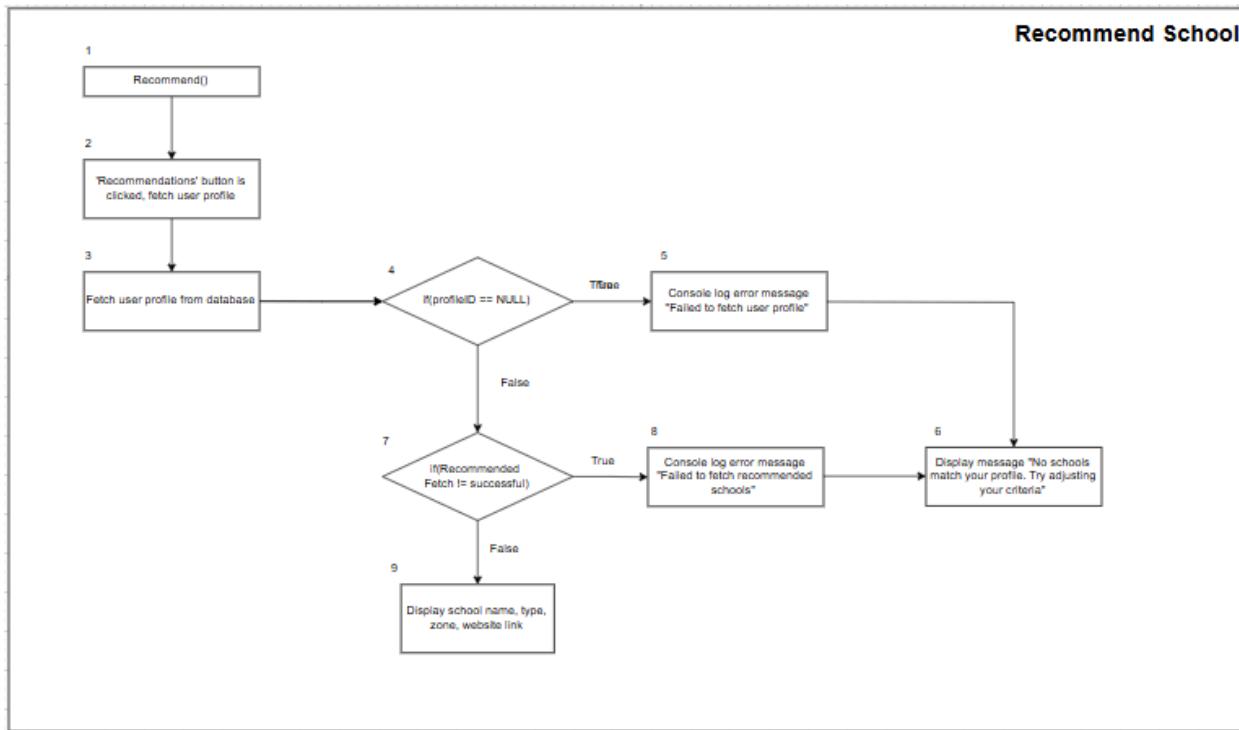
Cyclomatic Complexity = $2 + 1 = 3$

Basic Paths

1. Basispath: 1, 2, 3, 5, 8
2. Basic path: 1, 2, 3, 4, 7
3. Basic path: 1, 2, 3, 5, 6, 7

Path	Sequence	Description
1	1 -> 2 -> 3 -> 5 (False) -> 8 (False)	Webpage displays the results of schools found based on the school name field and filter results.
2	1 -> 2 -> 3 -> 4 (True) -> 7	Webpage displays "No results found" based on the school name field search.
3	1 -> 2 -> 3 -> 5 (False) -> 6 (True) -> 7	Webpage displays "No results found" based on the filter search results.

Recommended School



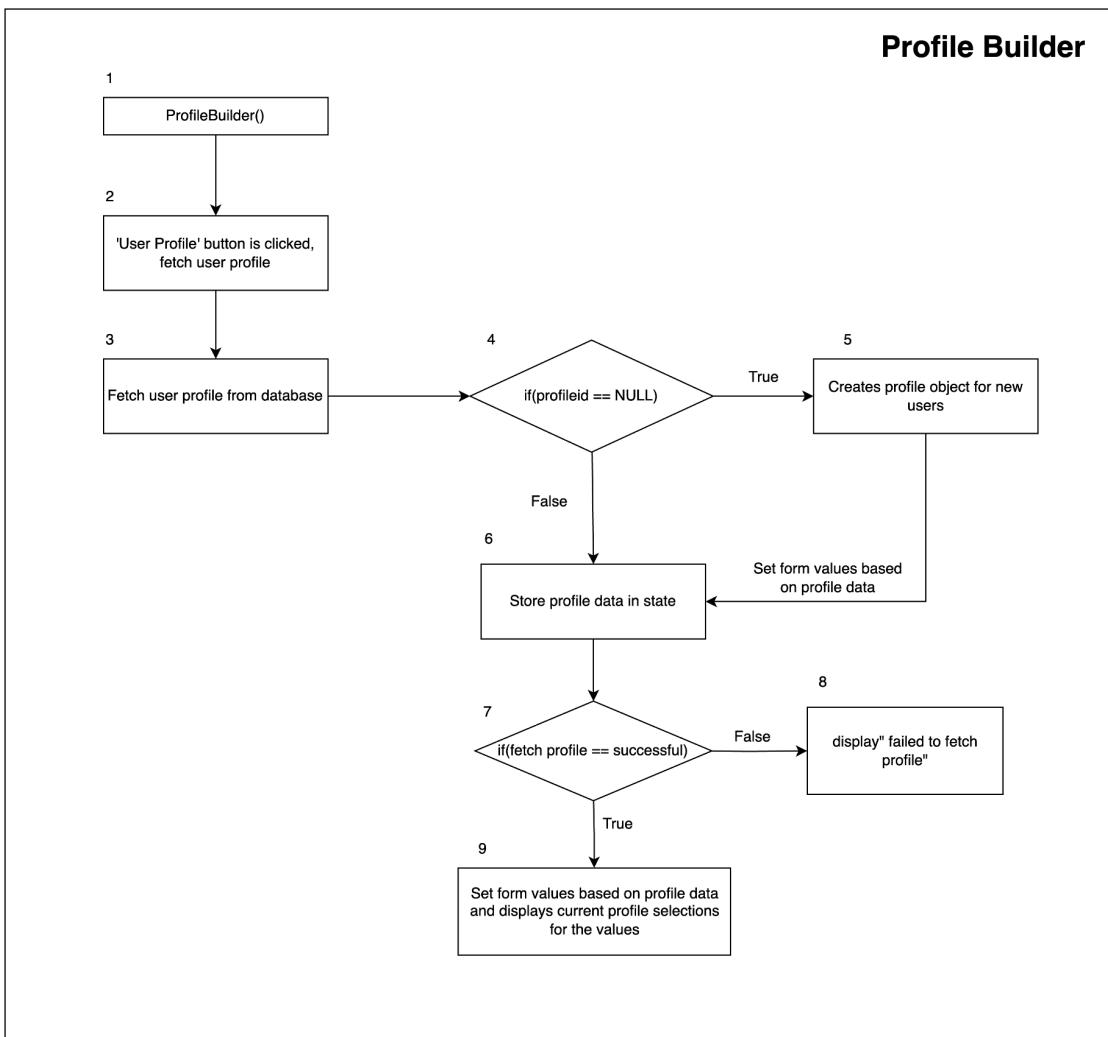
Cyclomatic Complexity = $2 + 1 = 3$

Basic Paths

1. Basispath 1: 1,2,3,4,7,9
2. Basispath 2: 1,2,3,4,5,6
3. Baseline path: 1,2,3,4,7,9

Path	Sequence	Description
1	1 -> 2 -> 3 -> 4(False) -> 7(False) -> 9	User profile created by user and recommendation displayed to user
2	1 -> 2 -> 3 -> 4(False) -> 5(True) -> 6	User Profile not created
3	1 -> 2 -> 3 -> 4(False) -> 7(True) -> 8(True) -> 6	User Profile exists, but no schools found under User's criteria

Profile Builder



Cyclomatic Complexity = $2 + 1 = 3$

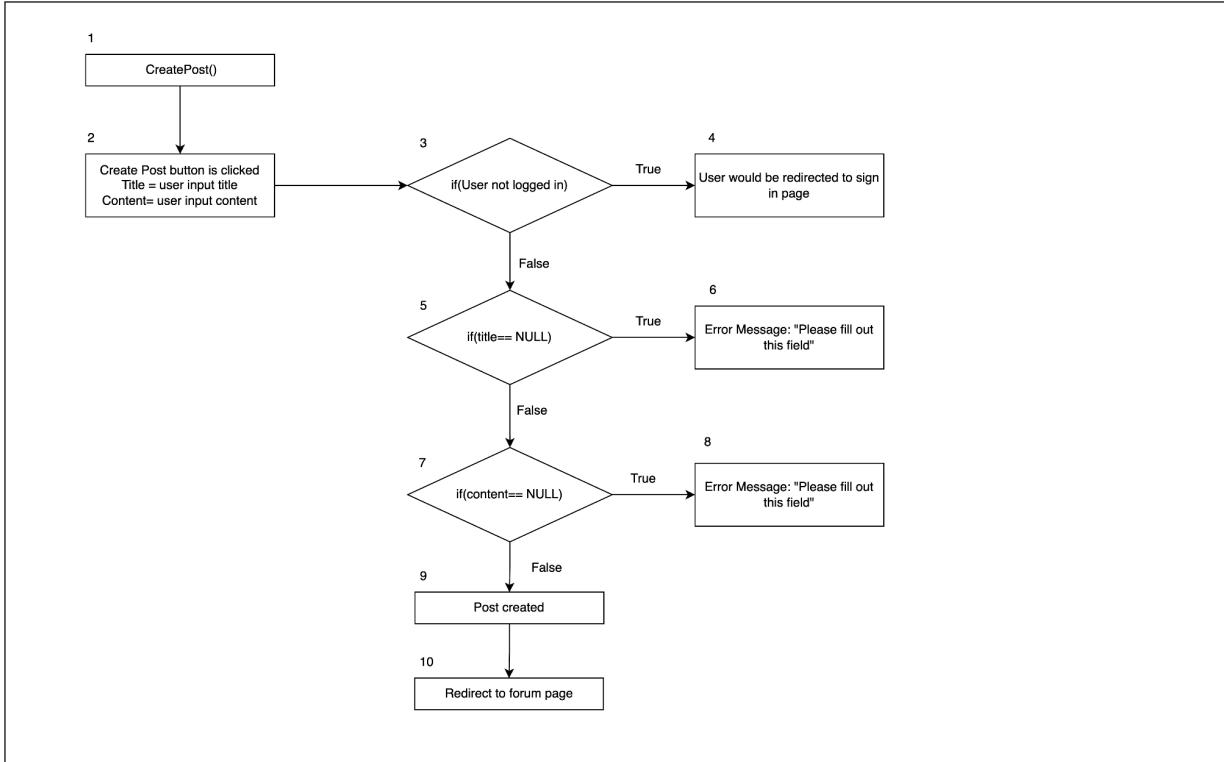
Basic Paths

1. Basis path 1: 1,2,3,4,5,9
2. Basis path 2: 1,2,3,4,6,7,8
3. Baseline path: 1,2,3,4,6,7,9

Path	Sequence	Description
1	1 → 2 → 3 → 4 (True) → 5 → 9	The profile ID is NULL, indicating a new user; creates a profile object for the new user and sets form values.
2	1 → 2 → 3 → 4 (False) → 6 → 7 (True) → 9	The profile ID exists, meaning it is an existing user; fetches profile data, stores it in state, and sets form values.

3	$1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \text{ (False)} \rightarrow 6 \rightarrow 7 \text{ (False)} \rightarrow 8$	The profile ID exists, but fetching profile data fails; displays an error message "Failed to fetch profile."
---	-------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------

Create Post



Cyclomatic Complexity = $3 + 1 = 4$

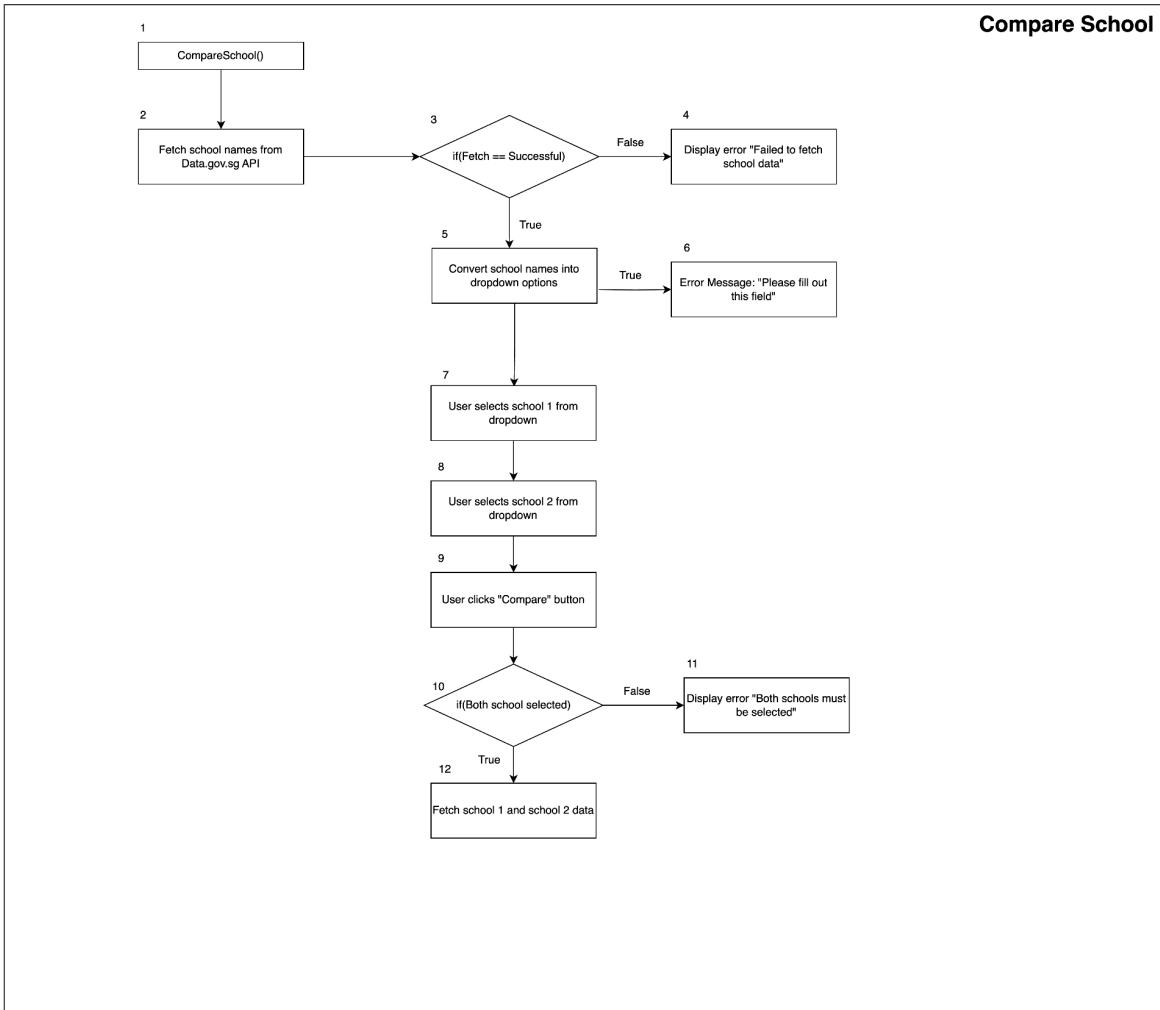
Basic Paths

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

Path	Sequence	Description
1	$1 \rightarrow 2 \rightarrow 3 \text{ (True)} \rightarrow 4$	The user is not logged in, so they are redirected to the sign-in page.
2	$1 \rightarrow 2 \rightarrow 3 \text{ (False)} \rightarrow 5 \text{ (True)} \rightarrow 6$	The user is logged in, but the title field is empty; an error message "Please fill out this field" is displayed.

3	$1 \rightarrow 2 \rightarrow 3 \text{ (False)} \rightarrow 5 \text{ (False)} \rightarrow 6 \text{ (True)} \rightarrow 8$	The user is logged in, the title is provided, but the content field is empty; an error message "Please fill out this field" is displayed.
4	$1 \rightarrow 2 \rightarrow 3 \text{ (False)} \rightarrow 5 \text{ (False)} \rightarrow 7 \text{ (False)} \rightarrow 9 \rightarrow 10$	The user is logged in, both the title and content fields are filled; the post is successfully created, and the user is redirected to the forum page.

Compare School



Cyclomatic Complexity = $2 + 1 = 3$

Basic Paths

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

Path	Sequence	Description
1	1 → 2 → 3 (False) → 4	The fetch from the Data.gov.sg API fails, and an error message "Failed to fetch school data" is displayed.
2	1 → 2 → 3 (True) → 5 → 6	The fetch is successful, but dropdown options are empty; an error message "Please fill out this field" is shown.
3	1 → 2 → 3 (True) → 5 → 7 → 8 → 9 → 10 (False) → 11	Both schools are not selected when the user clicks "Compare"; an error message "Both schools must be selected" is displayed
4	1 → 2 → 3 (True) → 5 → 7 → 8 → 9 → 10 (True) → 12	The fetch is successful, both schools are selected, and the "Compare" button is clicked; data for both schools is fetched.

Source:

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