

Final Project Report

# NowHere



CSCI3100 - Group D2

## Members:

1155109054 - Geoffrey Wong Hin (CSE CUHK)

1155114044 - Chak Sum Felix Ho (QFIN CUHK)

1155110077 - LIN Chuanfeng (CSE CUHK)

1155109863 - Leon Wong Man Him (STAT CUHK)

1155109788 - Chiu Chi Keung (CSE CUHK)

<b>1. Introduction</b>	<b>2</b>
1.1 Project Overview	2
1.2 Background	2
1.3 Objective	3
1.4 Highlights	4
1.5 Project Statistics	5
<b>2 SYSTEM ARCHITECTURAL DESIGN by DFD</b>	<b>6</b>
2.1 System Architecture	6
2.2 DFDs	8
<b>3. Detailed Description of Components by UML</b>	<b>9</b>
3.1 Overall Use Case Diagram	9
3.2 Overall Class Diagram	10
3.3 User Authentication Module	11
3.4 Profile Module	12
3.5 Search Module	14
3.6 Create Post Module	16
3.7 Edit Post Module	18
3.8 Post member management module	21
<b>4. User Interface Design</b>	<b>22</b>
4.1 Description of the User Interface	22
4.2 Objects and Actions with Screen Images	24
<b>5. Test</b>	<b>31</b>
5.1 Image Recognition with Neural Network	31
5.2 Search Bar	35
5.3 Search Bar (Homepage)	36
5.4 User Authentication	37
5.5 User Profile	41
5.6 Post Management	42
5.7 Post members management	45
<b>6. Lessons Learned</b>	<b>48</b>
<b>7. Conclusion</b>	<b>49</b>

# 1. Introduction

## 1.1 Project Overview

NowHere is a novel and user-friendly web application which provides an online platform for users to find travel partners. By combining the characteristics of self-guided trip and package tour, people can find or create their well-suited travel plan according to their preferences such as travel destination, travel style, travel period and group size. And thus, people with similar travel interests can be gathered and travel together.

### **Nowhere to Find travel partners? NowHere!**

## 1.2 Background

Nowadays, due to globalization and the economic growth of the countries, the tourism industry has become a sky-rocketing and profitable industry all around the world. Statistics showed that the industry had contributed 10.4% to the global GDP in 2018. Moreover, by investigating the data in each country, we found that many economies had the tourism sector that accounted for at least 7% of their overall GDP. Although the total revenue and performance of global tourism has decreased sharply from the beginning of 2020 due to the Covid-19 pandemic, a rebound is expected since the Covid vaccination campaigns are now underway across the world. Moreover, people are expected to travel more and willing to spend more within the post-pandemic period because of their vindictiveness towards the preventive measures lasting for a year. Therefore, we would like to provide a platform to the people who are longing for traveling.

In the current market, there are two main ways of traveling. One is the self-guided trip, and the other one is the package tour. Self-guided traveling has shown to gain overwhelming popularity over package tours due to the flexibility and autonomy over the travel plans. Such change is observed across different demographics, but it is the most significant in the younger age group, with 96% of people ages 25-34 are likely to travel independently.

However, the traditional package tour, as we believe, is an excellent way to make new friends by traveling together, but upon an important assumption, common interest. In conventional package tours, we do not have the prior information of the companions, and the traveling style of the common package tours is shopping. Therefore, it is difficult for people to get in touch deeply with their travel companions and usually regard them as strangers. However, if there is a platform containing the common interest and the traveling style of the people, it becomes easier for them to make new friends throughout the trip. Categories such as nationality, age, interests, expected expense, traveling style (culture/ nature/shopping/yoga-relax), travel period (day trip, short tour, long tour, foreign exchange) would be included in the dashboard. Then people can join or create their tailor-made trips and enjoy their group travels with ease. Currently, there are some websites and platforms that connect travelers together. An app called Backpacker helps backpackers to meet one another at the destination they are at. Another website called Travel Buddies allows travelers to browse through various profiles to find companions they would like to travel with. The community functions on those platforms will be fundamental to our platform. However, we aim to differentiate Nowhere by making the process of finding compatible companions as accessible as possible, such as adding a search function for travel destinations and other specifications, which is not present in those platforms.

Just like how companies like Expedia and Hotels.com provide cheap accommodation and traveling choices for self-guided travelers, we would like to support self-guided travelers by matching them with other self-guided travelers of similar travel plans. Whether it is for finding a new peer for future travel or seeking travel advice, we believe that connecting self-guided travelers together can empower them on their journeys.

### 1.3 Objective

Our website aims at helping people, who would like to travel during their precious holidays, but are too shy and scared to go on their own, find travel partners for a trip

of their lifetime. Organizing a trip abroad with friends is difficult with everyone's quickly changing schedule in this fast-paced society. We provide them with an option to look for strangers who would like to travel together with them.

## 1.4 Highlights

Our target users generally include everyone who loves traveling. So experienced travels who would like to meet some brand new faces on their journeys could definitely make use of our service. At the same time, we also implemented some user-friendly features that could help people who are not used to traveling a lot, quickly get up to speed.

For example, we implemented an image search feature, with the quickly developing Deep Learning Technology, that would automatically match an image input by the user with the most similar traveling destinations within our database. This helps people, who could only imagine what types of places they enjoy the most but do not have a clear idea of the exact destination, effortlessly search for a good place to go. This feature could also help people, who have a hard time searching for partners to go with them to a less well-known location, find a decent replacement.

To allow customers using our platform to get to know the strangers they are about to travel with better, we allow each user to customize their profile. They could write a short essay detailing their interests, personality, and traveling experience. We also allow users to select if they have some hobbies that could potentially cause discomfort to others, such as if they are a smoker or not. We believe these features could help reduce conflicts on the trip.

To ensure a user-friendly experience, we wish our platform could be as responsive as possible. That is why we implemented our backend on Firebase's real-time database. Upon receiving a request from one user, the database will be updated in split seconds. Other users viewing the same section could see the data being updated in real-time. This feature allows all the users to receive the most up-to-date information.

Overall, we ensure both experienced and inexperienced travelers could have a great time using our platform to find someone to accompany them on the journey.

## 1.5 Project Statistics

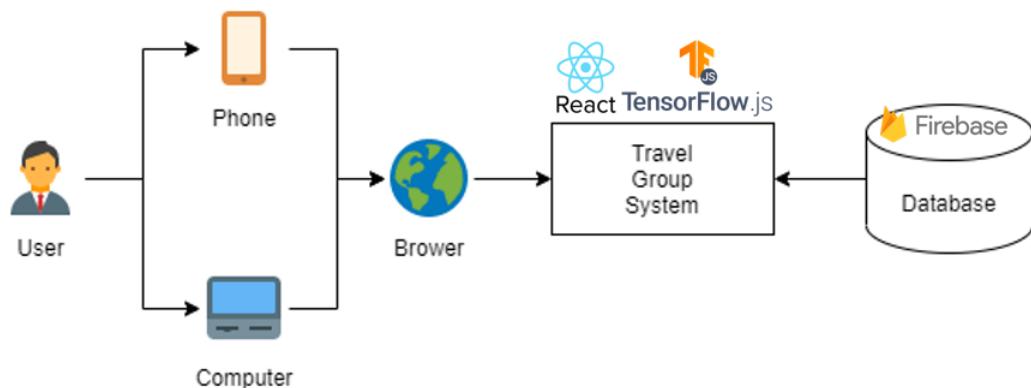
Important module	Lines of code	McCabe's number	Function	Comment
Total	3006	91	88	230
AccountDetails.js	360	3	3	12
SignIn.js	133	2	3	7
SignUp.js	155	2	2	7
CreatePost.js	194	4	5	11
EditPost.js	196	6	4	10
MyPost.js	114	6	4	10
Post.js	263	15	10	21
EditProfile.js	603	6	11	54

ImageUpload.js	227	3	9	13
Validate.js	86	16	9	4
FirebaseCrud.js	132	8	3	12
Search.js	375	7	11	31
Firebase.js	20	1	4	8
Home.js	268	6	6	16
App.js	144	6	4	14

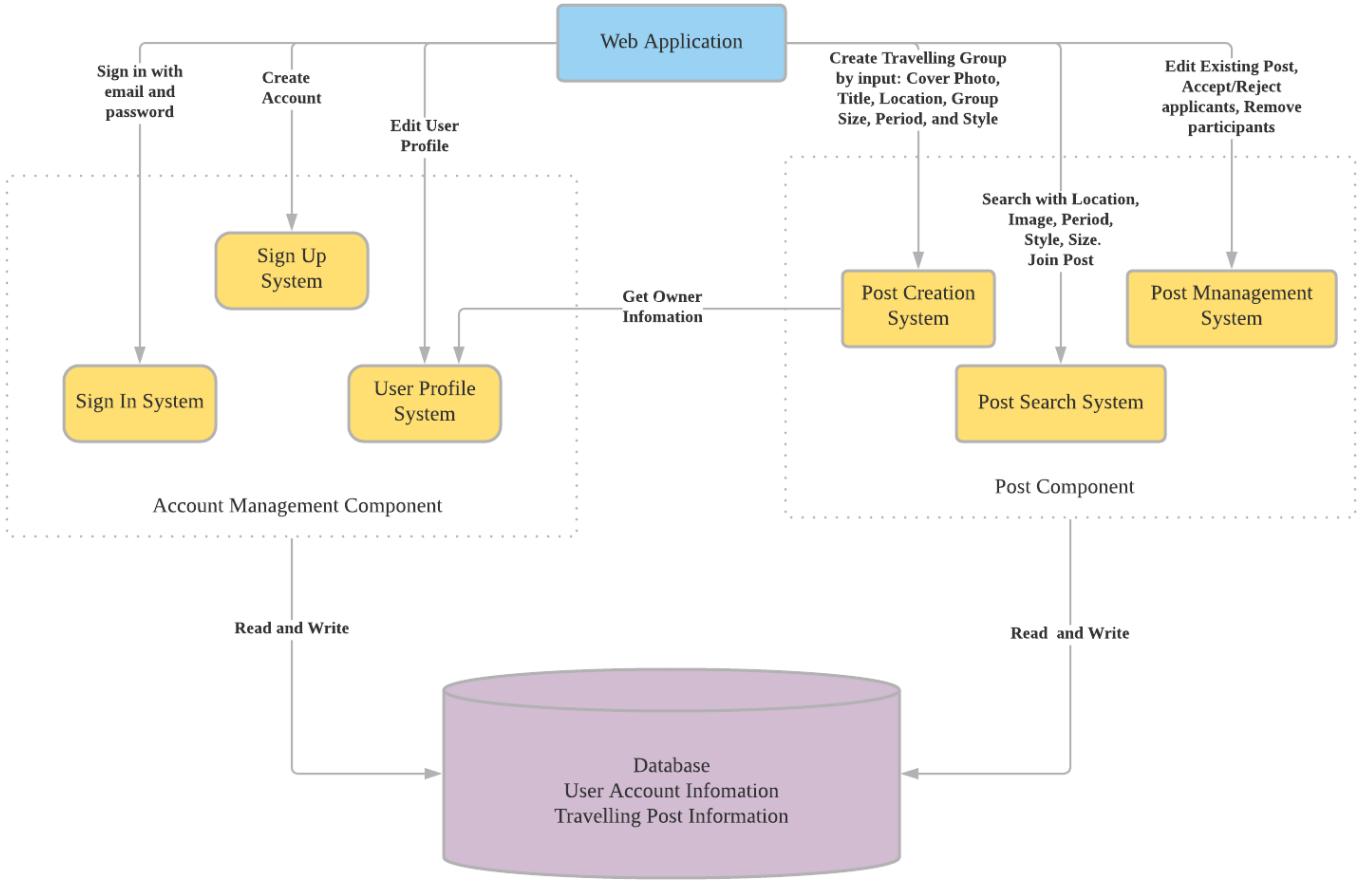
## 2 SYSTEM ARCHITECTURAL DESIGN by DFD

### 2.1 System Architecture

#### 2.1.1 Architecture Diagram

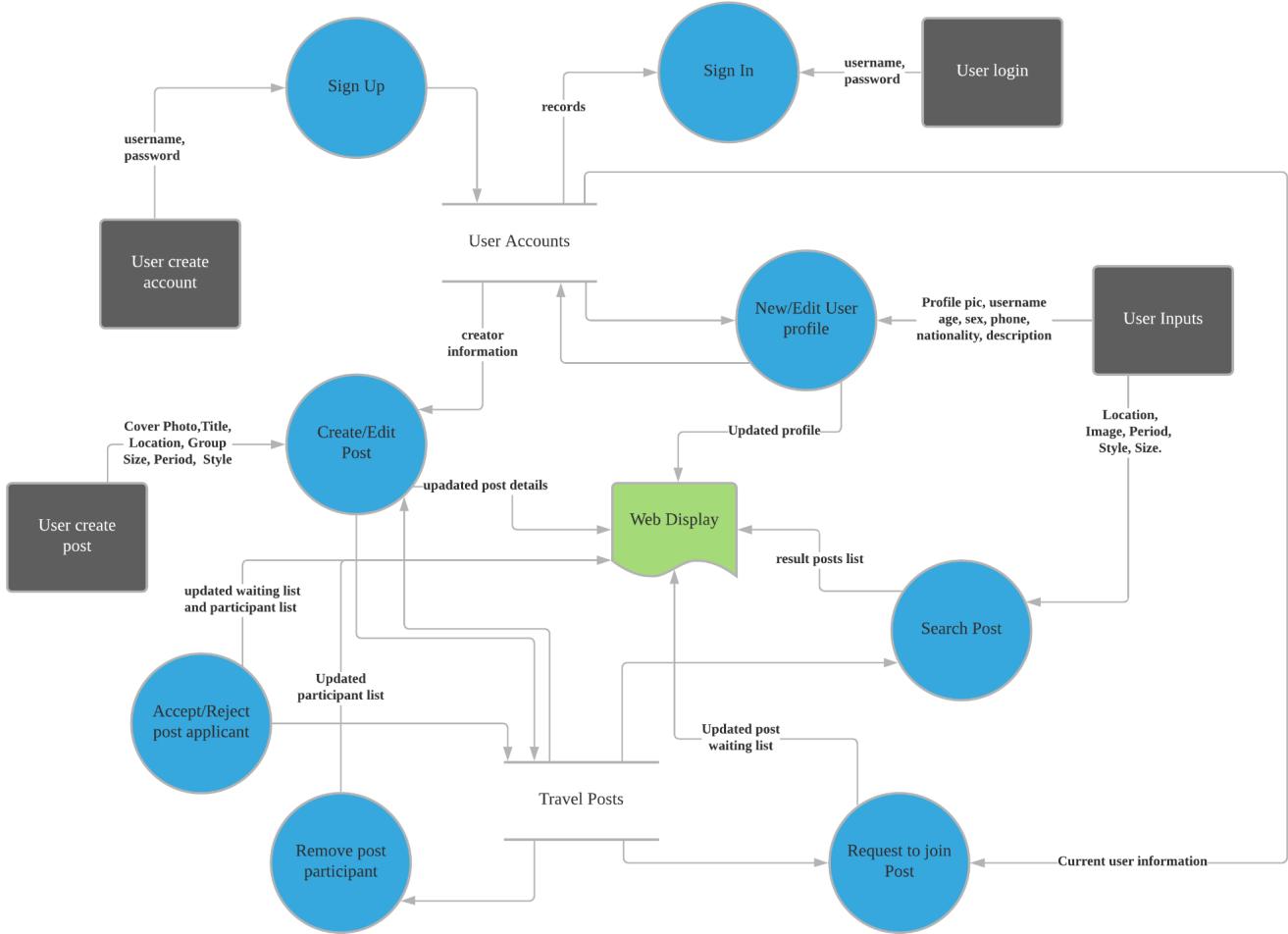


## 2.1.2 Component Diagram



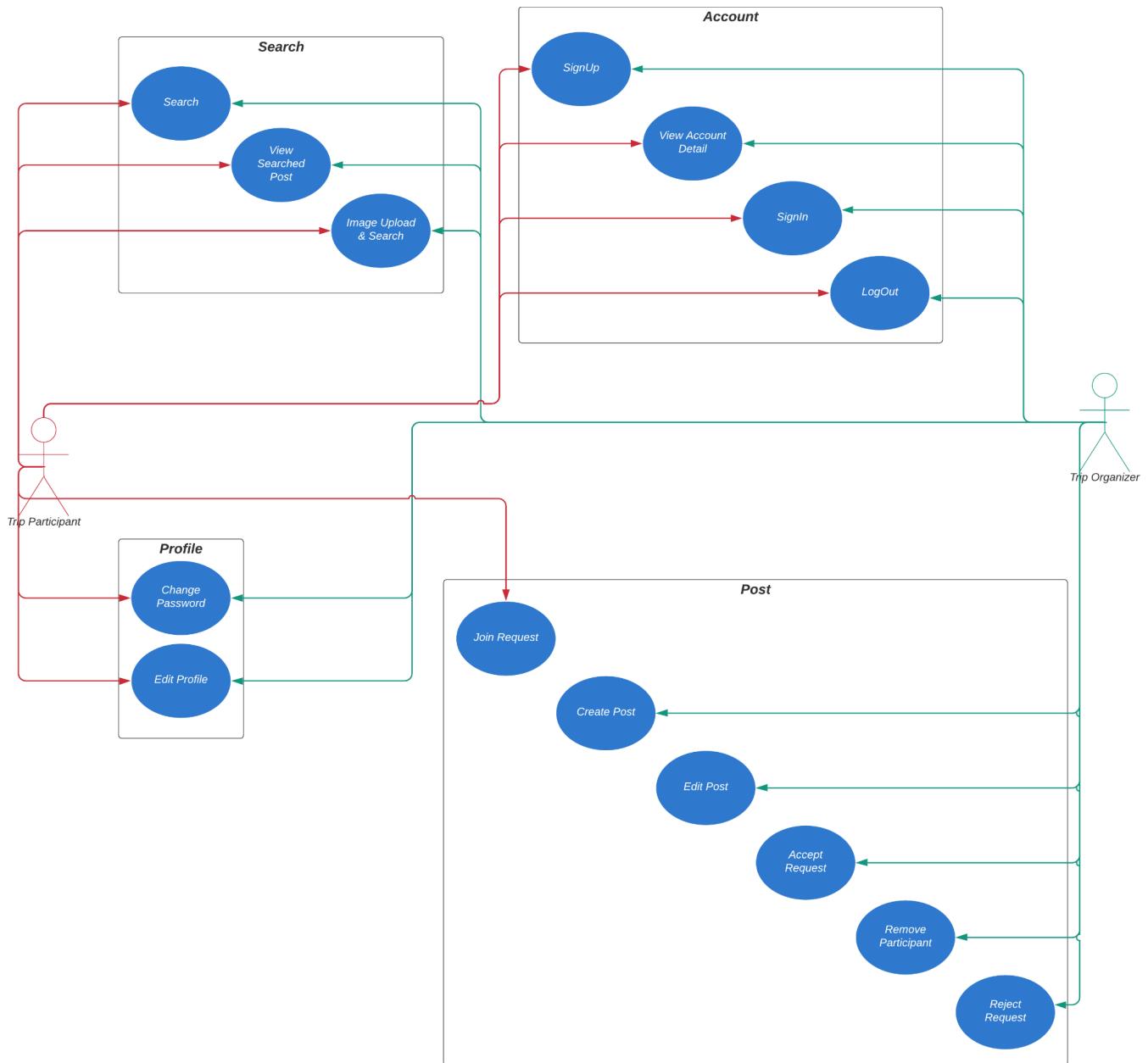
There are two key components and an online database for our web application. Each of the key components has some sub modules.

## 2.2 DFDs

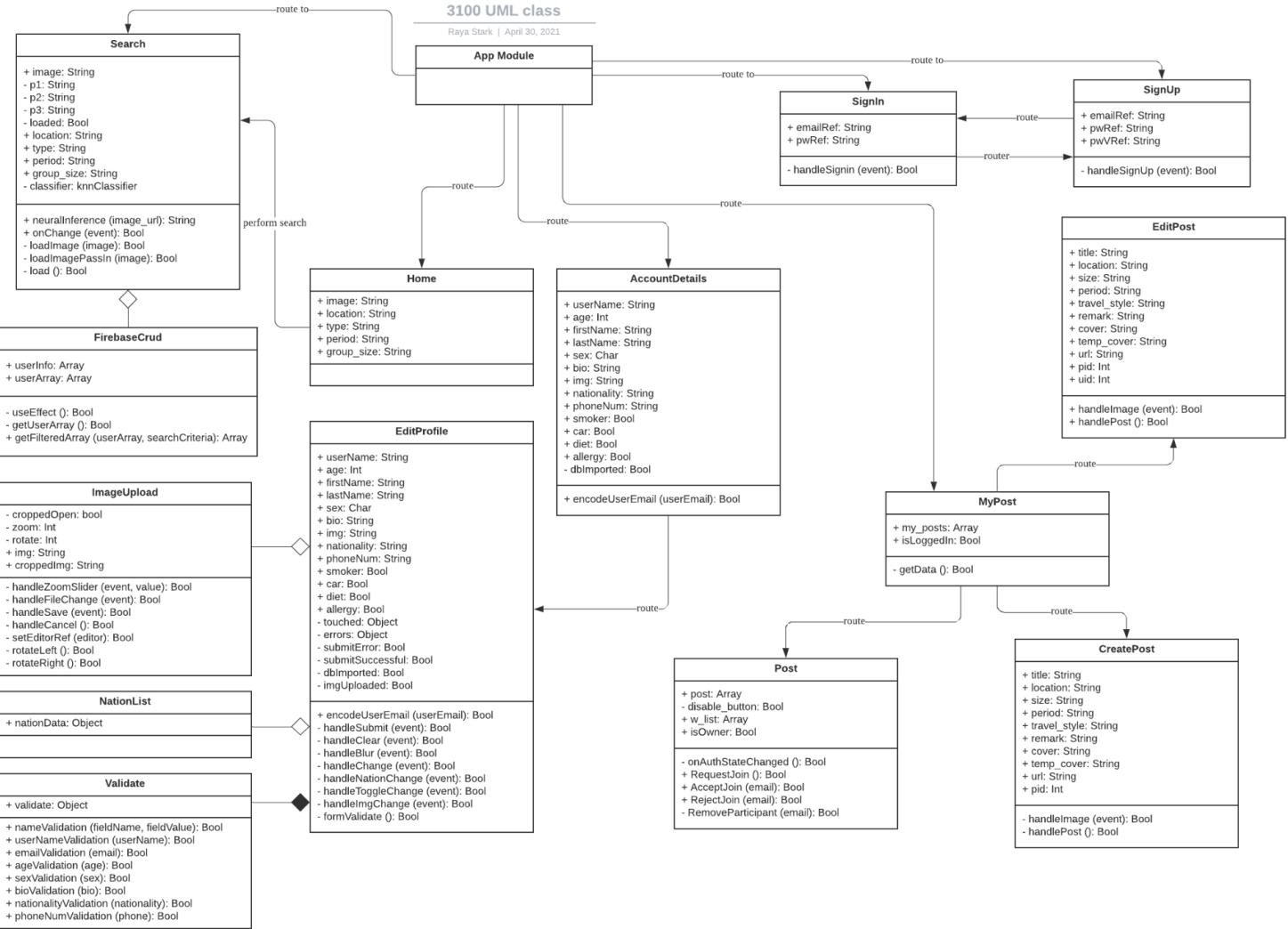


### 3. Detailed Description of Components by UML

#### 3.1 Overall Use Case Diagram

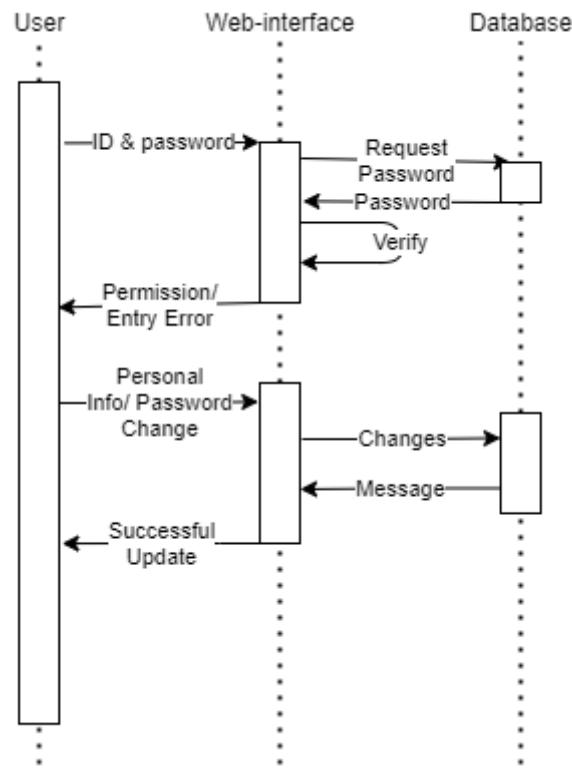


## 3.2 Overall Class Diagram



### 3.3 User Authentication Module

#### 3.3.1 UML Sequence Diagram for Account

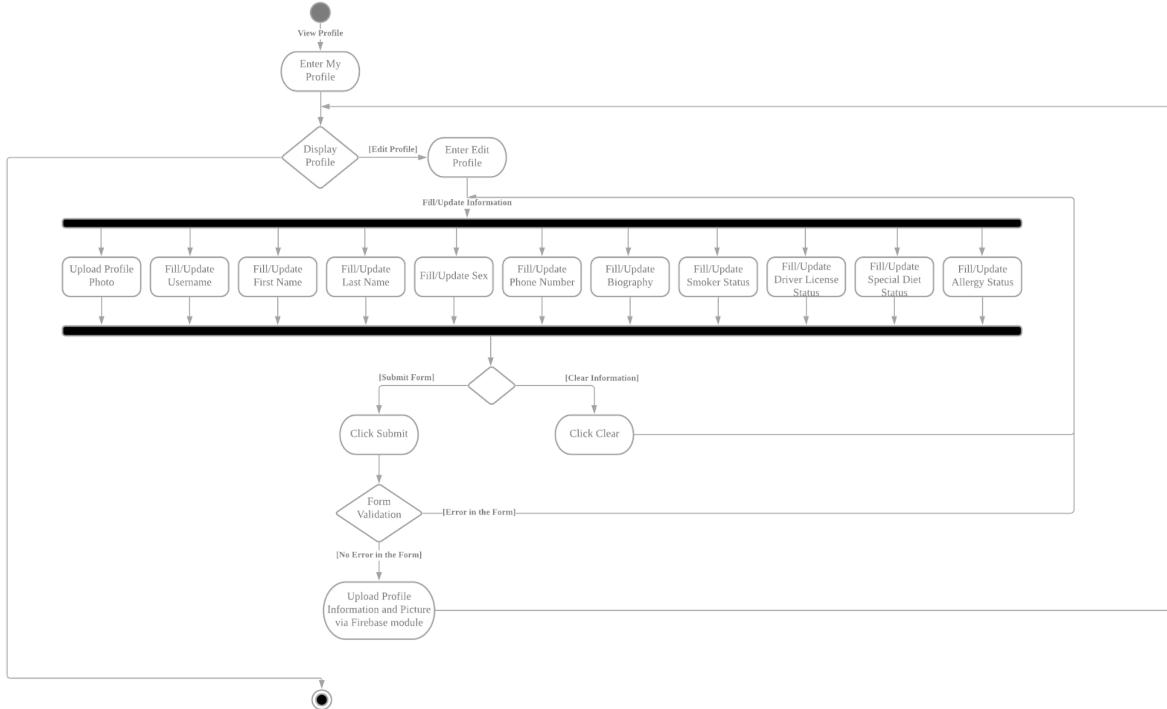


#### 3.3.2 Functionality

Let the user sign in with email and password to pass the authentication procedure.

## 3.4 Profile Module

### 3.4.1 UML Activity Diagram



### 3.4.2 Functionality

Users view or edit the profile to update their account's information. All profiles are created and edited via Edit Profile, which will upload the validated information to Firebase. The My Profile page will display the information stored on Firebase.

#### Step-By-Step Description:

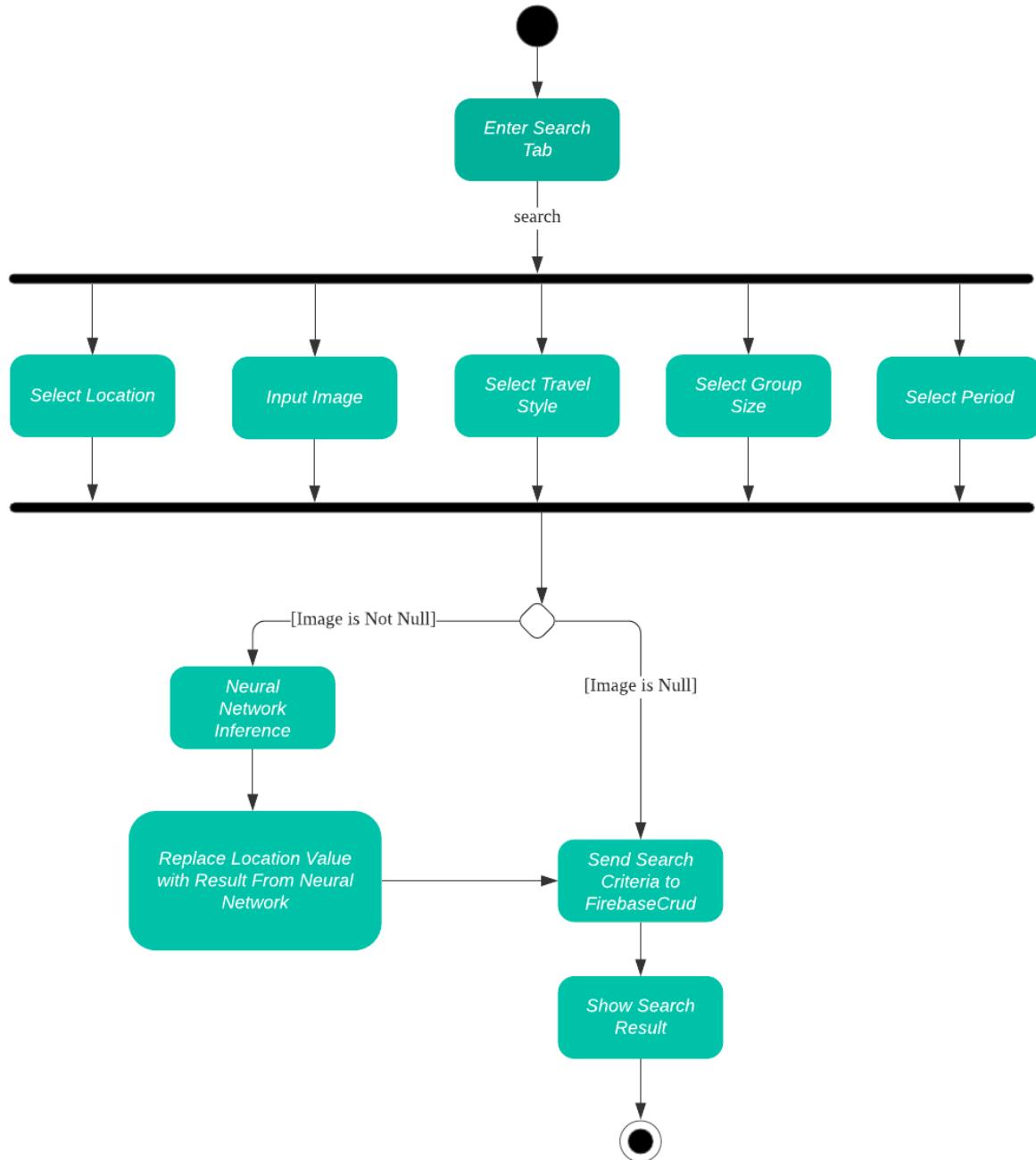
1. The user enters the My Profile tab.
2. The user sees their own profile.
3. If the user wishes to create or update their profile, the user enters Edit Profile .
4. The users fill out all the fills in the form and upload a profile photo if they wish to.
5. If the user wishes to start with a blank profile, the user clicks Clear.
6. If the user is satisfied with what they filled out, the user clicks Submit. If all information is valid, their profile will be updated on Firebase. Else, their submission is rejected and they have to amend the errors before submitting.
7. The updated profile will now be displayed in the My Profile tab.

### 3.4.3 Procedures and Functions

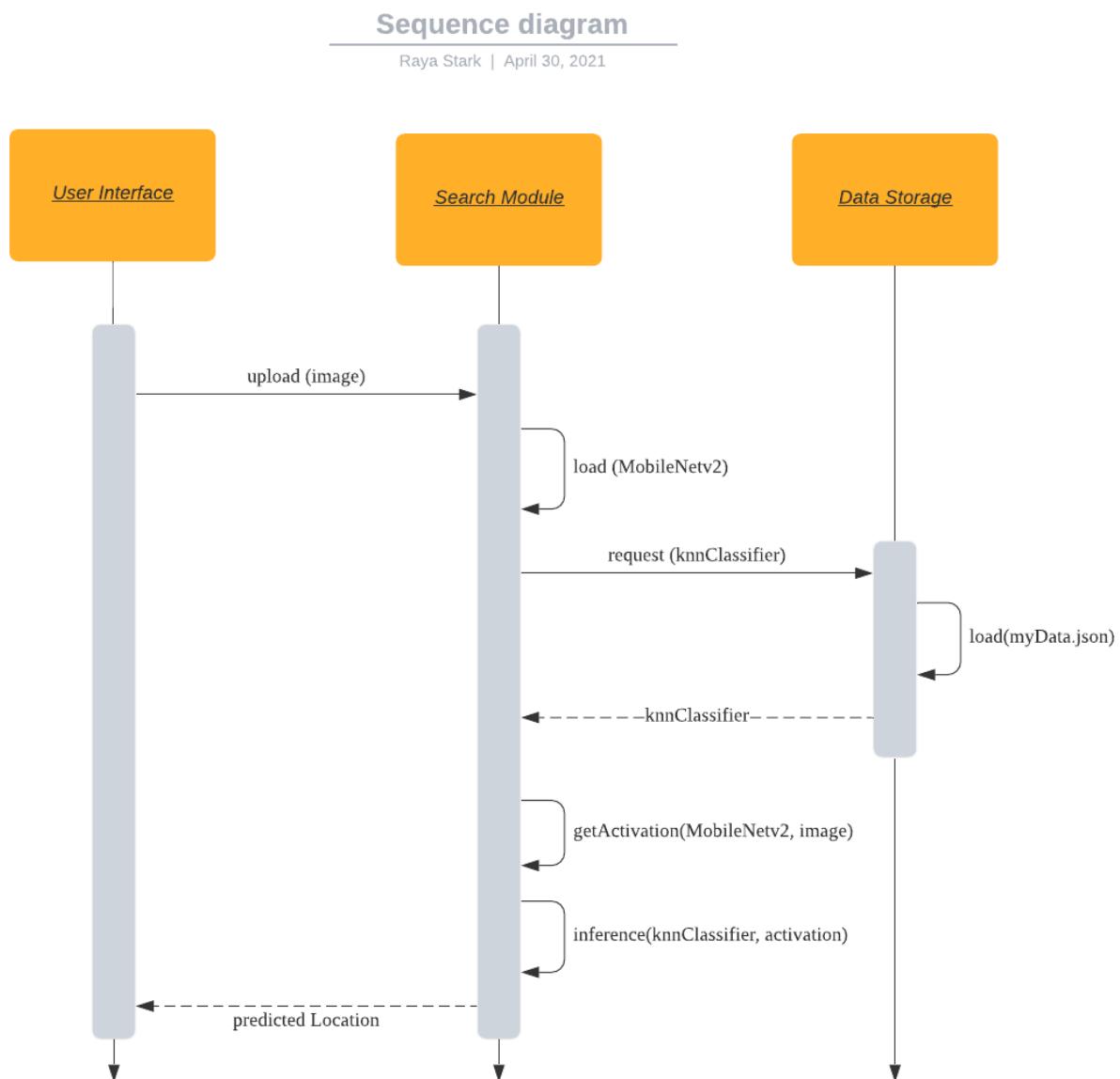
Function/ Procedure	Parameters	Descriptions
componentDidMount		Loads the profile information from Firebase if it exists
encodeUserEmail	userEmail	Helper function that edits the userEmail to make it a valid key for Firebase
formValidate		Help function that verifies if there are errors in the Edit Profile form upon submission
validate	nam, value	Collections of helper functions that validates each fill during editing
handleSubmit	event	Uploads validated profile information to Firebase upon submission of Edit Profile
handleClear	event	Clears all fills in the Edit Profile tab
handleBlur	event	Display errors in the Edit Profile form if there are.
handleChange/ handleNationChange/ handleToggleChange/ handleImgChange	event, value	Stores the inputs from the fills as states in preparation for handleSubmit and validation
imageUpload		Handles the upload procedure of the profile photo

## 3.5 Search Module

### 3.5.1 UML Activity Diagram for Overall Searching



### 3.5.2 UML Sequence Diagram for Image Search



### 3.5.3 Functionality

Users select search criteria from our autocomplete search bar. A real-time database filters the results, and shows them to users.

#### Step-By-Step Description:

1. The user enters the search tab.
2. The user sees all the existing posts available in the database.

The user enters search criteria.

If the user uploaded an image, we would perform neural network inference to replace the “Location” criteria with the most similar location in the image.

We send the search result to a sub-module, FirebaseCrud, that connects to our Firebase backend. It returns posts that satisfy all the search criteria if one or more search criteria are specified.

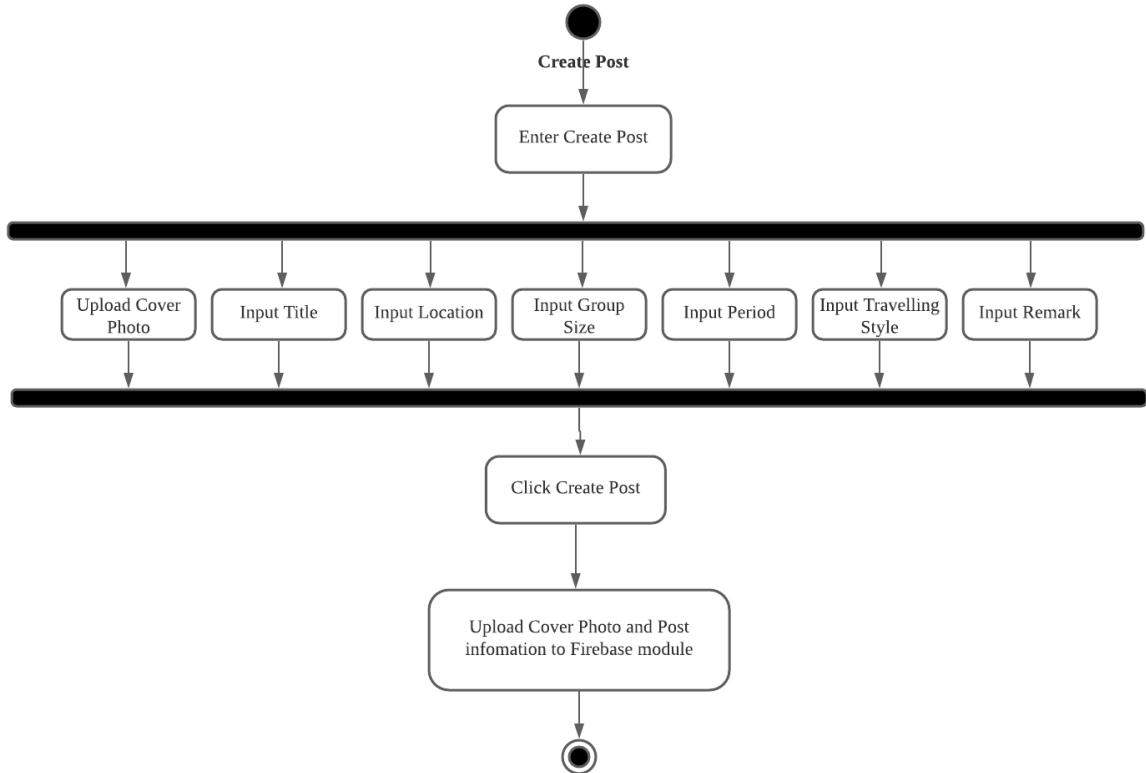
The filtered result is displayed to the user.

### 3.5.4 Procedures and Functions

Function/ Procedure	Parameters	Descriptions
neurallInference	image_url	Receive the url of an image uploaded by user, return a String of the most similar location.
onChange → {onChangeLocation, onChangeType, onChangePeriod, onChangeSize}		Update user search bar input to the states.
loadImage	image	Convert the image uploaded by user input url format and update the url onto the states.
loadImagePassIn	image	Handle the image uploaded by the user that was passed in from the Home module. Convert to url and update it onto the states.
load		Helper function for loading the MobileNetV2 model that is used in image search.

## 3.6 Create Post Module

### 3.6.1 UML Activity Diagram



### 3.6.2 Functionality

Users create posts with details. All post details will be uploaded.

**Step-By-Step Description:**

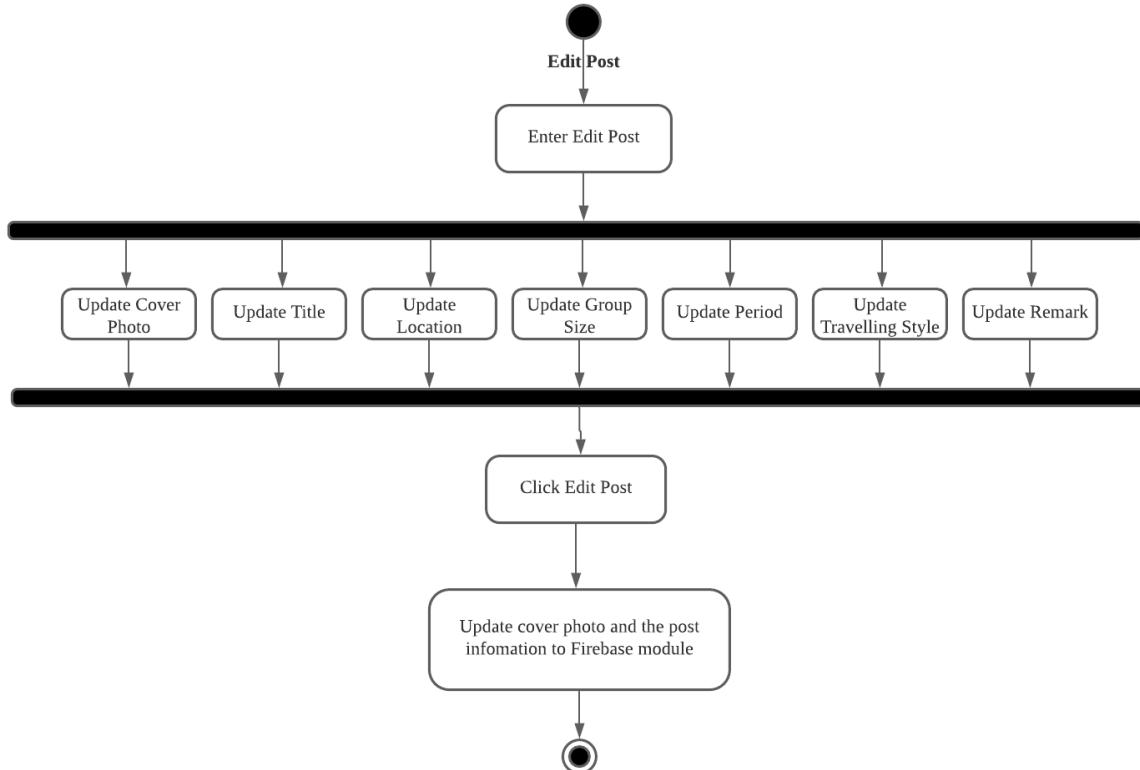
1. The user enters the Create New Post tab.
2. The user sees all the columns that are needed to be filled.
3. The user fills the details of the post and upload a cover photo if they want to.
4. If the user finished filling the details of the post, the user clicks the “Create Post” button.
5. The post details will now be shown on the post list.

### 3.6.3 Procedures and Functions

Function/ Procedure	Parameters	Descriptions
componentWillMount		Get the largest Post ID from Firebase
handlePost		Upload the cover photo to the Firebase Storage and updates the post information to Firebase Database
changeInput → {changeInputTitle, changeInputLocation, changeInputSize, changeInputPeriod, changeInputTravelStyle, changeInputRemark }	event	Update post details from user input to the states.
handleImage	event	Update the cover photo to the state and display the photo on the interface

## 3.7 Edit Post Module

### 3.7.1 UML Activity Diagram



### 3.7.2 Functionality

Users edit the post with details. All post details will be uploaded.

**Step-By-Step Description:**

1. The user enters the Edit Post tab.
2. The user sees all the columns of post details.
3. The user updates the details of the post and updates a cover photo if they want to.
4. If the user finished updating the details of the post, the user clicks the “Save Changes” button.
5. The post details will now be updated and shown on the post list.

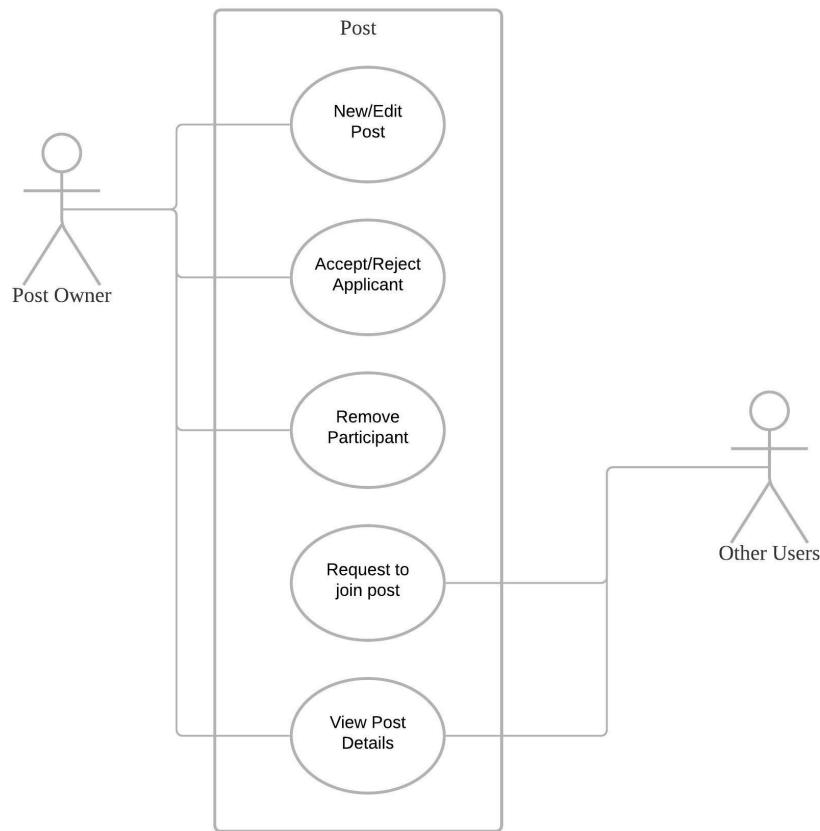
### 3.7.3 Procedures and Functions

Function/ Procedure	Parameters	Descriptions
componentWillMount		Get the current post details from Firebase
handlePost		Update the cover photo to the Firebase and updates the post information to Firebase

changeInput → {changeInputTitle, changeInputLocation, changeInputSize, changeInputPeriod, changeInputTravelStyle, changeInputRemark }	event	Update post details from user input to the states.
handleImage	event	Update the cover photo to the state and display the photo on the interface

## 3.8 Post member management module

### 3.8.1 UML USE-CASE Diagram



### 3.8.2 Functionality

The post owner can manage the members for the post.

### 3.8.3 Detail Description

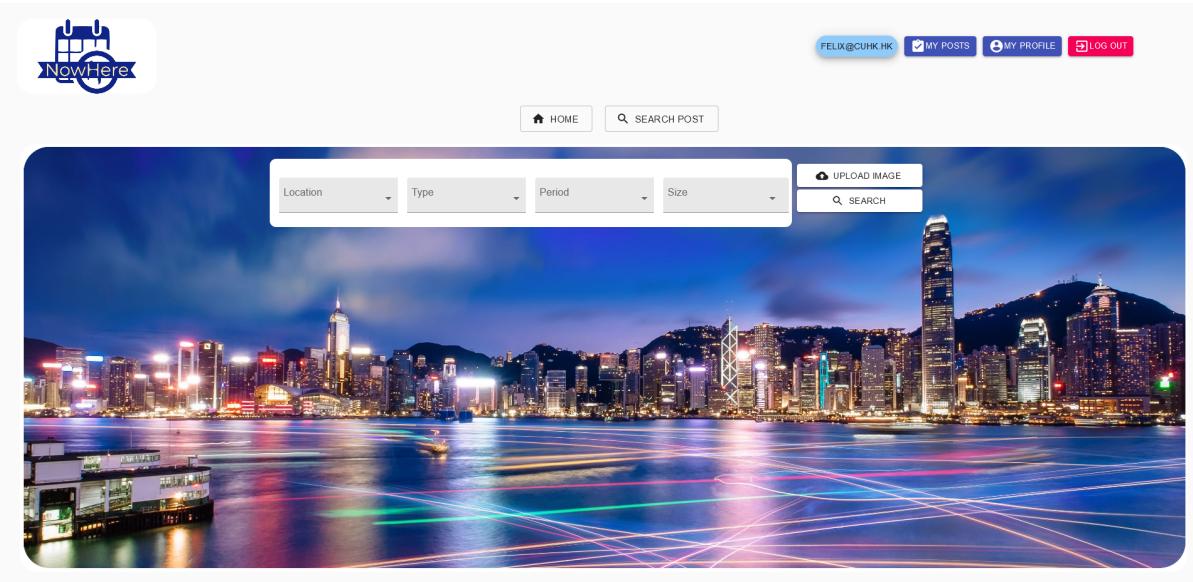
Users can click the “Request join” button to put himself/herself in the waiting list of the post. The button is not visible to the post owner as the owner cannot be put in the waiting list.

In the participant list there is a remove button for each entry which is only visible to the owner. Similarly, for each entry in the waiting list, there is a “Accept” and a “Reject” button which are only visible to the owner.

## 4. User Interface Design

### 4.1 Description of the User Interface

Our aim with the user interface design is to make it as minimalistic as possible to let the users navigate through our website comfortably. There are two sets of navigation bars in our website, with the one in the center corresponding to the main function of our website and the one in the top right corresponding to the subsidiary functions. The purpose of creating two sets of navigation bars is to separate the functions for visibility purposes and make it more convenient for the users to access the common functions.



To attain minimalism in the website, we used MaterialUI to decorate our website. We use intuitive colors such as dark blue and red in the buttons of the navigation bar to highlight the function each button corresponds to, such as red for logging out. The use of color is supplemented with the use of icons for the buttons, such that the function of the buttons can easily be identified.

The design of the website is consistent for every page, which includes the layout, fonts, color etc. We also try to create layers of information for display, such that only the information relevant to the user is displayed at a time.

LOCATION	TRAVEL STYLE	PERIOD	SIZE		
	Hong Kong	Shopping	Weeks-Trip	2-4	<a href="#">VIEW POST</a>
	Hong Kong	Sporty	Exchange(student)	5-8	<a href="#">VIEW POST</a>
	Hong Kong	Sporty	Long-Trip	8+	<a href="#">VIEW POST</a>
	United States	Sporty	Long-Trip	2-4	<a href="#">VIEW POST</a>
	United States	Cultural	Weeks-Trip	5-8	<a href="#">VIEW POST</a>
	United States	Shopping	Day-Trip	8+	<a href="#">VIEW POST</a>
	India	Cultural	Weeks-Trip	2-4	<a href="#">VIEW POST</a>

 NowHere
FELIX@CUHK HK
[MY POSTS](#)
[MY PROFILE](#)
[LOG OUT](#)

[HOME](#)
[SEARCH POST](#)





Participant:None

**Title: Let's Shopping in Hong Kong**

Creator: jim@cuhk.hk  
 Location: Hong Kong  
 Group Size: 2-4  
 Period: Weeks-Trip  
 Travelling Style: Shopping  
 Remark: If you are interested, join me!

[REQUEST JOIN](#) [RANDOM REQUEST\(FOR DEV\)](#)

Waiting List:None

For example, we only have minimal information in the search post so that only the basic details are displayed upon search. If the user is interested in the post, then the user can get further information about the post by viewing the post. This makes it easier for the user to find what they want from the website.

All feedback from the system, such as error messages, are also displayed to the user when required. We make sure that they are displayed where it is relevant, and contains the information that is intuitive for the user to understand. As such, we can achieve a balance between good user experience and the system requirement.

The screenshot shows a user profile creation form. The fields and their current values are:

- Username:** felix123
- First Name:** Felix
- Last Name:** Ho
- Age:** twenty one (with validation error: "Please enter a number")
- Phone Number:** 3567890123 (with validation error: "Please enter a valid phone number. I e: xxxx-xxx")
- Sex:** Male
- Nationality:** Hong Kong
- Description:** hi all I love travel
- Smoker?** No  Yes
- Have Driver's License?** No  Yes
- Allergies?** No  Yes
- Have a special dietary requirements?** No  Yes

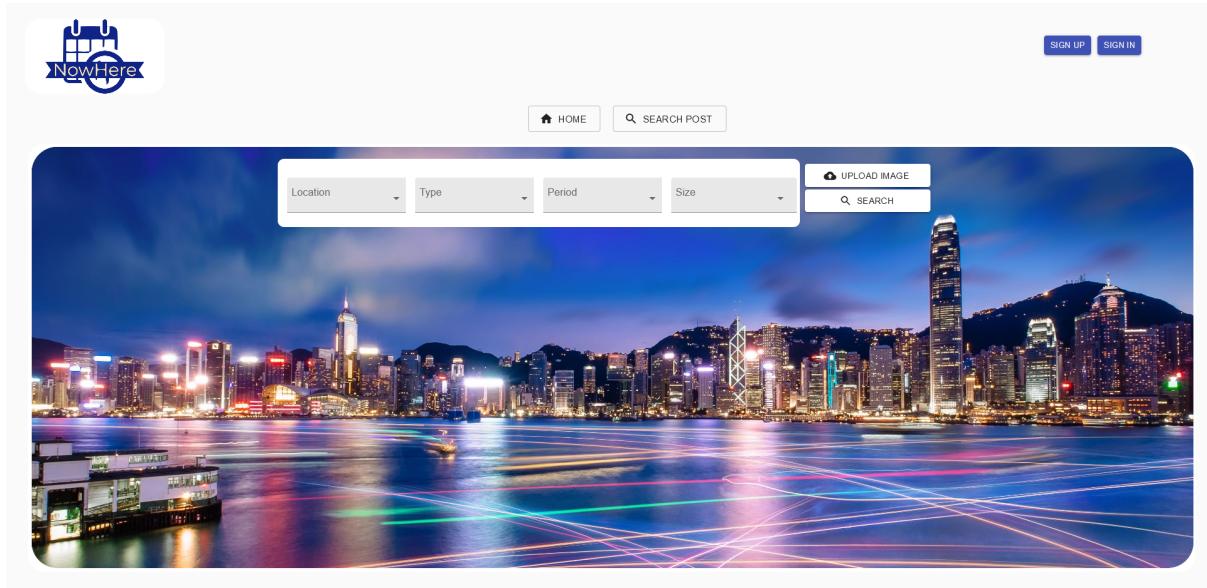
At the bottom, there are two buttons: **SUBMIT** (in blue) and **CLEAR** (in grey). A red error message is displayed: "Error detected in profile submission. Please update your information correctly."

Lastly, our website is also implemented as a single-page application (SPA) to achieve smooth transitions between pages of the website. Most of the pages in our website are light weighted so choosing to implement it as a SPA is an appropriate choice that is in-line with the minimalist design.

## 4.2 Objects and Actions with Screen Images

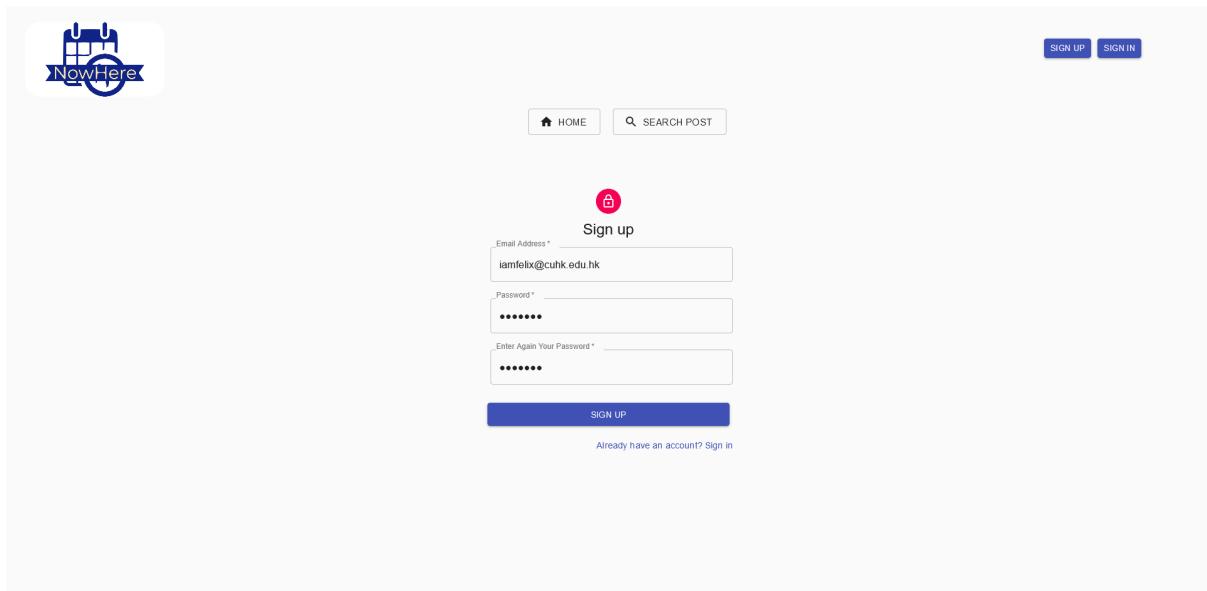
### 4.2.1 Home Page

Our home page mimics the style of travel related websites by providing the search post function. Users who are not signed-in can have a preview of what the website has to offer through the search bar. Searching posts on this page is equivalent to searching within the search post page. While the user is not signed-in, they can only access sign-up and sign-in page the top-right navigation bar.



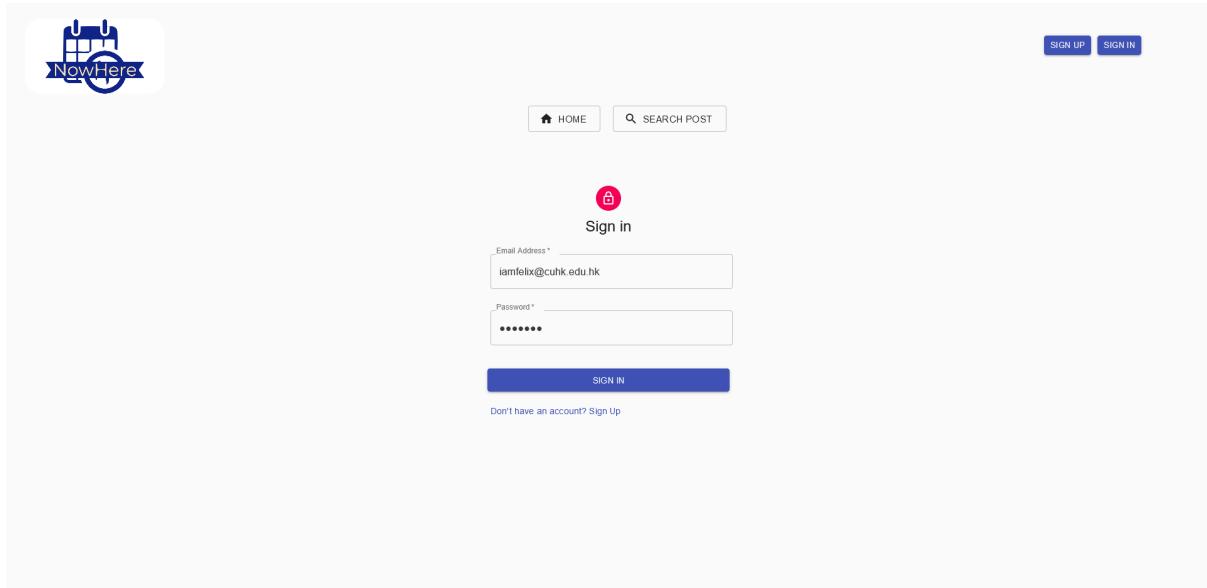
#### 4.2.2 Sign up

All users will need to create an account using their email address if they wish to join any trip they are interested in. Users need to create a password that is strong enough, i.e. password with capitalized and numeric characters.



#### 4.2.3 Sign in

Once the account is created, the user can sign in to the website by inputting the email and password the account. New users will be redirected to the My Profile page for inputting further information.



#### 4.2.4 My Profile

In this page, users can check their account details that they have inputted when they first created the account. For new users, the page will be blank since no profile was created. To create one, edit the profile via the button.

The screenshot shows the profile editing interface for a user named 'Felix Ho'. At the top, there's a navigation bar with links for 'HOME', 'MY POSTS', 'MY PROFILE', and 'LOG OUT'. Below the navigation is a search bar labeled 'SEARCH POST'.

The main area displays 'Felix Ho's Profile' and a large profile picture of a scenic landscape with mountains and water. There's also a 'EDIT PROFILE' button.

The profile form fields include:

- Username:** felix123
- Name:** Felix Ho
- Age:** 21
- Sex:** Male
- Phone Number:** 55678901
- Nationality:** Hong Kong
- Description:** hi all I love travel
- Other Information:** A section with checkboxes for Smoker, Driver's License, Allergies, and Special Dietary Requirements. 'Allergies' and 'Special Dietary Requirements' are checked.

#### 4.2.5 Edit Profile

Users can customize their profile by filling out the form with their information, such as name, age, phone number, etc. Users can upload a profile image of their own if they wish to not use the system's default image. The page is equipped with a basic image editor so that any image can be used as the profile image. The form also validates the information and will inform the user upon success or failed submission.

The screenshot shows the 'Edit Profile' page of the NowHere website. At the top right, there are links for 'FELIX@CUHK HK', 'MY POSTS' (with a checkmark icon), 'MY PROFILE' (with a person icon), and 'LOG OUT' (with a red square icon). Below these are 'HOME' and 'SEARCH POST' buttons.

The main area is titled 'Edit Profile'. It features a large image of a lake and mountains. Below the image is a blue button labeled 'UPLOAD PROFILE PICTURE' with a cloud icon.

The profile form includes the following fields:

- UserName \***: felix123
- First Name \***: Felix
- Last Name \***: Ho
- Age \***: 21
- Phone Number \***: 55678901
- Sex \***: Male
- Nationality \***: Hong Kong
- Description \***: hi all I love travel
- Smoker?**: No  Yes
- Have Driver's License?**: No  Yes
- Allergies?**: No  Yes
- Have a special dietary requirements?**: No  Yes

At the bottom are 'SUBMIT' and 'CLEAR' buttons.

#### 4.2.6 Search Post

All posts made by the community can be searched on this page. Aside from the basic search method that filters based on categories, our website also supports image search if the user wishes to do so.

The screenshot shows the NowHere homepage. At the top right, there are links for 'FELIX@CUHK HK', 'MY POSTS' (with a checkmark icon), 'MY PROFILE' (with a person icon), and 'LOG OUT'. Below these are buttons for 'HOME' and 'SEARCH POST'. A search bar with dropdown menus for 'Location', 'Type', 'Period', and 'Size' is followed by 'UPLOAD IMAGE' and 'SEARCH' buttons. The main content area displays a grid of travel posts. Each post includes a thumbnail image, location (e.g., Hong Kong, United States), travel style (e.g., Shopping, Sporty, Cultural), period (e.g., Weeks-Trip, Exchange(student), Long-Trip), group size (e.g., 2-4, 5-8, 8+), and a 'VIEW POST' button.

#### 4.2.7 Post Details

Each post contains more information for interested users to see. This page is also where interested users can request to join the trip if they wish to. Note that if the user who is accessing the post is also the creator of the posts, they have the ability to add or remove users from the waiting queue and participant list.

The screenshot shows a detailed view of a travel post. At the top right are links for 'FELIX@CUHK HK', 'MY POSTS' (with a checkmark icon), 'MY PROFILE' (with a person icon), and 'LOG OUT'. Below are buttons for 'HOME' and 'SEARCH POST'. The main content features a large horizontal image of a green, hilly landscape. Below it is a smaller vertical image of a city skyline at night. To the left of the images, the post title is 'Title: Let's Shopping in Hong Kong'. Below the title are details: 'Creator: jim@cuhk.hk', 'Location: Hong Kong', 'Group Size: 2-4', 'Period: Weeks-Trip', 'Travelling Style: Shopping', and 'Remark: If you are interested, join me!'. At the bottom are buttons for 'REQUEST JOIN' and 'RANDOM REQUEST(FOR DEV)'. At the very bottom, it says 'Participant:None' and 'Waiting List:None'.

#### 4.2.8 My Posts

Users can view the posts they have made. Each post leads to the corresponding post details page and can also be edited as well. Users can also create new posts from here.

The screenshot shows the NowHere homepage with a header featuring the logo and navigation links: FELIX@CUHK.HK, MY POSTS, MY PROFILE, and LOG OUT. Below the header are buttons for HOME and SEARCH POST. Three travel posts are displayed in a grid:

- Taj Mahal**: Location: India, Travel Style: Cultural, Group size: 2-4. Includes a photo of the Taj Mahal.
- Cricket Tournament**: Location: India, Travel Style: Sporty, Group size: 5-8. Includes a photo of a cricket player.
- Night Mall Shopping Trip**: Location: India, Travel Style: Shopping, Group size: 2-4. Includes a photo of a night market.

A red "CREATE NEW POST" button is located at the bottom left of the main content area.

#### 4.2.9 Create/Edit Post

Users can create posts for others to join via this page. Users need to input title, location, group size, duration, travelling style and remark of the trip to create a post. They can also upload a photo relevant to the trip if they wish.

The screenshot shows the "Create / Edit Post Page" with the following fields:

- Cover Photo: A placeholder image of the Taj Mahal with a "Browse..." button and a note "No file selected."
- Title: Taj Mahal
- Location: India
- Group Size: 2-4
- Period: Weeks-Trip
- Travelling Style: Cultural
- Remark: "Explore the culture of India in a 5 day trip!"

A blue "SAVE CHANGES" button is at the bottom right.

## 5. Test

### 5.1 Image Recognition with Neural Network

#### 5.1.1 Purpose

We would like to test if our neural network can predict unseen locations and classify them into the correct classes. Since Machine Learning applications behave differently compared with normal programs, we do not expect 100% accuracy from the testing result, and the validation procedure is also different.

We adopted 5-Fold Validation in our project. This means 20% of the overall data is withheld from training the Machine Learning application. The withheld data is then used to measure the robustness and classification performance of our model.

#### 5.1.2 Testing Procedure

The following images for validation are uploaded to the neural network. The predicted result is compared with the expected using a confusion matrix.

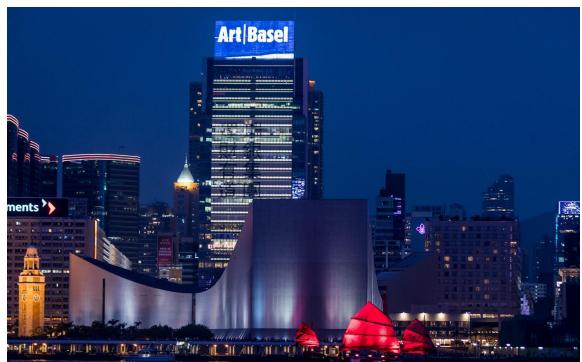
Images for Validation (5-Fold)	Expected Output
	{location: Africa}
	{location: Africa}



{location: Britain}



{location: Britain}



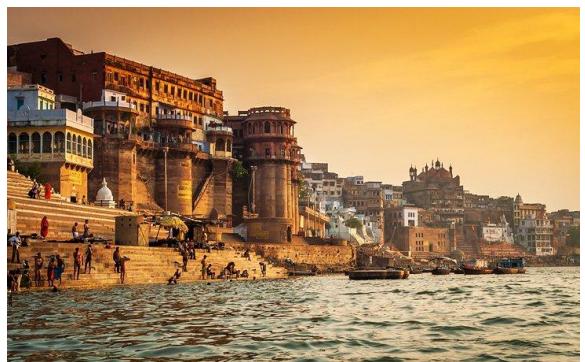
{location: Hong Kong}



{location: Hong Kong}



{location: India}



{location: India}



{location: India}



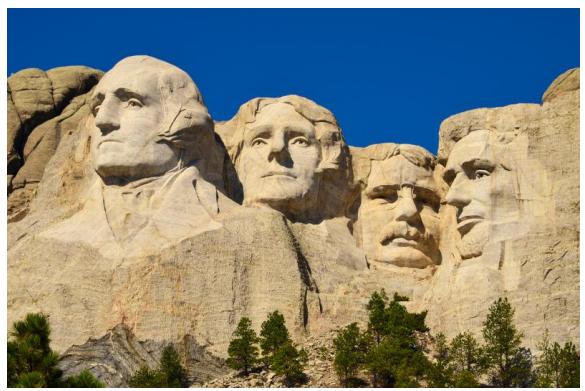
{location: Japan}



{location: Japan}



{location: United States}



{location: United States}

### 5.2.3 Test Result

Our machine learning approach scored an accuracy of 70% from 5-fold validation. This is marginally above the baseline of random guessing (%) and direct image comparison (openCV template matching with sum square difference). This is overall satisfactory but still has room for improvement in terms of dataset size. (Current dataset size = 62 Images, 6 Categories).

	Random Guessing	Direct Image Comparison	Machine Learning
Accuracy	16.7%	38.5%	69.2%
Overall Performance Summary			

## 5.2 Search Bar

### 5.2.1 Purpose

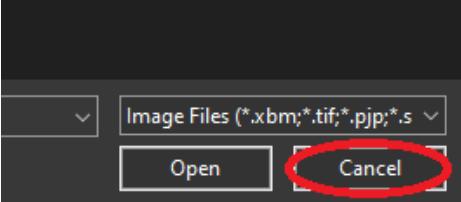
To test if our search bar can perform searching with multiple criteria.

- When no criteria is given, all existing posts are displayed
- We only support exact matching. However, the user can type the input partially, and we will return options for autocomplete.

### 5.2.2 Testing procedure

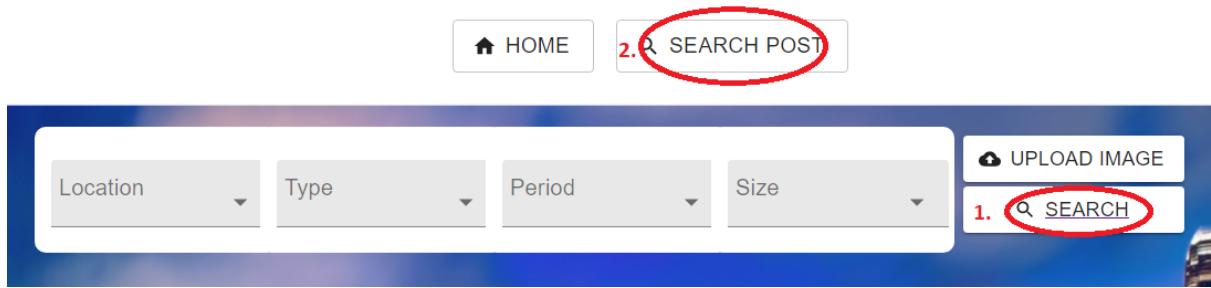
Different values are filled into the search bar. We observe if the displayed posts match the search criteria.

ID	Search Criteria	Expected Response	Remarks
1	{location: null, type: null, period: null, group_size: null, image: null}	All posts displayed	No search criteria
2	{location: "", type: "", period: "", group_size: "", image: null}	All posts displayed	Empty String as search criteria
3	{location: "Britain", type: "Sporty", period: "Day-Trip", group_size: "2-4", image: null}	Only posts matching all the provided criteria are displayed	Multiple search criteria
4	{location: "!@#\$%", type: null, period: null, group_size: null, image: null}	No posts displayed	Non-exact search result
5	{location: "Japan", type: null, period: null, group_size: null, image: null} → {location: null, type: null, period: null, group_size: null, image: null}	Japan posts only → All posts displayed	Removing existing search criteria
6	{location: "Japan", type: null, period: null, group_size: null, image: "Britain/10.jpg"} →	Japan posts only → Britain posts only	Neural Network search result would overwrite existing

	{location: "Britain", type: null, period: null, group_size: null, image: "Britain/10.jpg"}		location search criteria
7	 <p>Click the cancel button during image upload</p>	Search results no change	Cancelling initiated image upload

## 5.3 Search Bar (Homepage)

### 5.3.1 Purpose



We replicated the search feature from the search tab to the home page. The search bar has the exact behavior. But the users have to click the “Search” button (Button 1.) in order to be teleported to the search tab and see the result.

On the other hand, the user can choose to simply route to the search tab without keeping what they have input in the Home page by clicking the “Search Post” tab switch button (Button 2.).

### 5.3.2 Testing procedure

We replicated the testing procedures with ID: 1 - 4 & 7 from **section 5.2.2** since they should behave the same way. We then observe the behavior of the website by using different buttons.

ID	Search Criteria/ Buttons	Expected Response	Remarks
1	{location: "Britain", type: "Sporty", period: "Day-Trip", group_size: "2-4", image: null} +  Tab Switch (Button 2.)	All posts displayed	Using Tab Switch does not transfer user input.
2	{location: "Japan", type: null, period: null, group_size: null, image: "Britain/10.jpg"} +  Search (Button 1.)	Initially Japan posts are displayed. After the neural network inference completes, Britain posts are displayed	It takes a moment of loading before inference can be completed and replacing existing location search.

## 5.4 User Authentication

### 5.4.1 User sign up

#### Purpose

To test if the user can sign up with email and password. To maintain the security and stability. The user has to input an email address in valid format, a strong enough password (at least 6 characters or numbers) and type in the exact same password again to prevent mistypes.

#### Testing procedure

Fill in different contents in the 3 input boxes and press the “SIGN UP” button.

Each case must perform exactly the same as the expected output to pass the test.

Email Address \*

Password \*

Enter Again Your Password \*

**SIGN UP**

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

The following table shows the 7 test cases.

ID	Inputs	Expected Response	Remarks
1	All empty	Pop up window: The email address is badly formatted	Sign up failed
2	Email Address: <a href="mailto:testing_email55@gmail.com">testing_email55@gmail.com</a> Password: pw1234 PW verifier: pw1234	Pop up window: The email address is badly formatted	Sign up failed
3	Email Address: <a href="mailto:testing_email55@gmail.com">testing_email55@gmail.com</a> Other 2 boxes: (empty)	Pop up window: The password is too weak	Sign up failed

4	Email Address: <a href="mailto:testing_email55@gmail.com">testing_email55@gmail.com</a> Password: 12 PW verifier: 12	Pop up window: The password is too weak	Sign up failed
5	Email Address: <a href="mailto:testing_email55@gmail.com">testing_email55@gmail.com</a> Password: pw1234 PW verifier: (empty)	Pop up window: The two passwords are different, please try again	Sign up failed
6	Email Address: <a href="mailto:testing_email55@gmail.com">testing_email55@gmail.com</a> Password: pw1234 PW verifier: 1234	Pop up window: The two passwords are different, please try again	Sign up failed
7	Email Address: <a href="mailto:testing_email55@gmail.com">testing_email55@gmail.com</a> Password: pw1234 PW verifier: pw1234	1.Pop up window: Signed up successfully with email <a href="mailto:testing_email55@gmail.com">testing_email55@gmail.com</a> , Please fill in more information about you.  2.Direct to Edit Profile page	Sign up succeed

## 5.4.2 User sign in

### Purpose

To test if the user can sign in with email and password. To maintain the security and stability. The user has to input an email address and a password that exactly match with a record in the Firebase server.

### Testing procedure

Fill in different contents in the 3 input boxes and press the “SIGN IN” button.

Each case must perform exactly the same as the expected output to pass the test.

Assume the user owns an account with email: [testing\\_email55@gmail.com](mailto:testing_email55@gmail.com) and password: pw1234



## Sign in

Email Address \*

testing\_email55@gmail.com

Password \*

.....

SIGN IN

[Don't have an account? Sign Up](#)

The following table shows the 7 test cases.

ID	Inputs	Expected Response	Remarks
1	All empty	Pop up window: The email address is badly formatted	Sign in failed
2	Email Address: <a href="mailto:testing_email55@gmail.com">testing_email55@gmail.com</a> Password: pw1234	Pop up window: There is no user record corresponding to this identifier. The user may have been deleted.	Sign in failed
3	Email Address: <a href="mailto:testing_email55@gmail.com">testing_email55@gmail.com</a> Password: (empty)	Pop up window: Wrong Password	Sign in failed

4	Email Address: <a href="mailto:testing_email55@gmail.com">testing_email55@gmail.com</a> Password: 12	Pop up window: Wrong Password	Sign in failed
5	Email Address: <a href="mailto:testing_email55@gmail.com">testing_email55@gmail.com</a> Password: pw1234	Direct to Home page	Sign in succeed

## 5.5 User Profile

### 5.5.1 Edit Profile

#### Purpose

To test if the user can create and edit their profile. All profile information is validated before submission to Firebase.

The screenshot displays the 'Edit Profile' interface. At the top, there is a placeholder profile picture showing a scenic view of a lake and mountains. Below the picture is a blue button labeled 'UPLOAD PROFILE PICTURE'. The form contains several input fields:

- UserName \***: felix123
- First Name \***: Felix
- Last Name \***: Ho
- Age \***: 21
- Phone Number \***: 55678901
- Sex \***: Male
- Nationality \***: Hong Kong
- Description \***: hi all I love travel

At the bottom of the form, there are two rows of questions with radio buttons:

- Smoker?** No  Yes
- Have Driver's License?** No  Yes
- Allergies?** No  Yes
- Have a special dietary requirements?** No  Yes

At the very bottom are two buttons: a blue 'SUBMIT' button on the left and a grey 'CLEAR' button on the right.

#### Testing procedure

Fill in different contents in the 9 fills of the Edit Profile tab and press the “Submit” button. Each case must perform exactly the same as the expected output to pass the test. Unless specified below, the fills will be filled out the same way as above.

ID	Inputs	Expected Response	Remarks
1	All empty	Pop up message: Error detected in profile submission. Please update your information correctly.	Submission failed
2	First Name: a	Pop up message: First Name needs to be at least two characters.	Error detected before submission
3	Last Name: a	Pop up message: Last Name needs to be at least two characters	Error detected before submission
4	Age: Twenty one	Pop up message: Please enter a number	Error detected before submission
5	Age: 11	Pop up message: Age must be at least 18	Error detected before submission
6	Phone Number: 123	Pop up message: Please enter a valid phone number. i.e: xxxx-xxxx	Error detected before submission
7	Description: a	Pop up message: Your description must at least have 2 words	Error detected before submission
8	Same as photo	Pop up message: Successfully updated felix123's Profile!	Submission succeed

## 5.6 Post Management

### 5.6.1 Create Post

#### Purpose

To test if the user can create a post if the user misses the important information of the post.

The screenshot shows a 'Create Post' form with the following fields:

- Cover Photo: Choose File (no file selected)
- Title: Title
- Location: Location
- Group Size: 2-4
- Period: Weeks-Trip
- Travelling Style: Sporty
- Remark: Enter something...

A blue 'CREATE POST' button is located at the bottom right of the form.

## Testing procedure

Fill in different contents in the Create New Post tab and press the “Create Post” button. Each case must perform exactly the same as the expected output to pass the test.

ID	Inputs	Expected Response	Remarks
1	{Cover Photo:null, Title: null, location: null, Group_size: "2-4"(default), period: "Weeks-Trip"(default), Travelling_Style: "Sporty"(default), Remark:null}	Pop up message: Please fill the title and location.	Failed Creating post
2	{Cover Photo:null, Title: "Going to Hong Kong", location: null, Group_size: "2-4"(default), period: "Weeks-Trip"(default), Travelling_Style: "Sporty"(default), Remark:null}	Pop up message: Please fill the title and location	Failed Creating post
3	{Cover Photo:null, Title: null, location: "Hong Kong", Group_size: "2-4"(default), period: "Weeks-Trip"(default), Travelling_Style: "Sporty"(default), Remark:null}	Pop up message: Please fill the title and location	Failed Creating post

4	{Cover Photo:null, Title: "Going to Hong Kong", location: "Hong Kong", Group_size: "2-4"(default), period: "Weeks-Trip"(default), Travelling_Style: "Sporty"(default), Remark:null)	Pop up message: Successfully created the post!	Succeed Creating post
5	{Cover Photo:hongkong.jpg, Title: "Going to Hong Kong", location: "Hong Kong", Group_size: "2-4"(default), period: "Weeks-Trip"(default), Travelling_Style: "Sporty"(default), Remark:"Join me if you are interested!"}	Pop up message: Successfully created the post!	Succeed Creating post

### 5.6.2 Edit Post

#### Purpose

To test if the user can edit the post if the user misses the important information of the post.

**Edit Post**

Cover Photo:	
	<input type="button" value="Choose File"/> no file selected
Title:	<input type="text" value="Going to Hong Kong"/>
Location:	<input type="text" value="Hong Kong"/>
Group Size:	<input type="text" value="2-4"/>
Period:	<input type="text" value="Weeks-Trip"/>
Travelling Style:	<input type="text" value="Sporty"/>
Remark:	<input type="text" value="Join me if you are interested!"/>
<input type="button" value="SAVE CHANGES"/>	

#### Testing procedure

Update the different contents in the Edit Post tab and press the “Save Changes” button. The post for testing cases will be used from **section 5.6.1** Each case must perform exactly the same as the expected output to pass the test.

ID	Input Changes	Expected Response	Remarks
1	Title: “Going to Hong Kong” → Title: null	Pop up message: Please fill the title and location.	Failed Editing post
2	Location: “Hong Kong” → Location: null	Pop up message: Please fill the title and location	Failed Editing post
3	Title: “Going to Hong Kong” → Title: null, Location: “Hong Kong” → Location: null	Pop up message: Please fill the title and location	Failed Editing post
4	Cover Photo: hongkong.jpg → Cover Photo: VictoriaHarbour.jpg	Pop up message: Successfully updated the post!	Succeed Editing post
5	Group_Size: “2-4” → Group_Size: “5-8”	Pop up message: Successfully updated the post!	Succeed Editing post
6	Period: “Weeks-Trip” → Period: “Long-Trip”	Pop up message: Successfully updated the post!	Succeed Editing post
7	Travelling Style: “Sporty” → Travelling Style: “Culture”	Pop up message: Successfully updated the post!	Succeed Editing post
8	Remark: “Join me if you are interested!” → Remark: “Hello”	Pop up message: Successfully updated the post!	Succeed Editing post

## 5.7 Post members management

### 5.7.1 User request to join a post

#### Purpose

To test if the user can request to join a button by hitting the “Request Join” button in a post page. The owner cannot request to join his own post, so the button will not be shown if the current user is the owner. The user cannot request to join if he/she is already in the waiting list.



### Title: Let's Shopping in Hong Kong

Creator: jim@cuhk.hk

Location: Hong Kong

Group Size:2-4

Period: Weeks-Trip

Travelling Style: Shopping

Remark: If you are interested, join me!

[REQUEST JOIN](#)

[RANDOM REQUEST\(FOR DEV\)](#)

Participant:None

Waiting List:

jack@cuhk.hk

dummy@dummy.com5

### Testing procedure

In a post detail page, press the “Request button” to join a post. Here we test under the condition that the current user is not the post owner.

## Title: Let's Shopping in Hong Kong

Creator: [jim@cuhk.hk](mailto:jim@cuhk.hk)

Location: Hong Kong

Group Size:2-4

Period: Weeks-Trip

Travelling Style: Shopping

Remark: If you are interested, join me!

[REQUEST JOIN](#)

[RANDOM REQUEST\(FOR DEV\)](#)

ID	Condition & Inputs	Expected Response	Remarks
1	Condition: User not logged in Owner: <a href="mailto:jim@cuhk.hk">jim@cuhk.hk</a> User: (Null) click “Request Join” button	Pop up window: Please sign in first	Request failed

2	Condition: User logged in Owner: <a href="mailto:jim@cuhk.hk">jim@cuhk.hk</a> User: <a href="mailto:jack@cuhk.hk">jack@cuhk.hk</a> click “Request Join” button	Current User’s email appears in the waiting list	Request succeeded
3	Condition: User already in the waiting list Owner: <a href="mailto:jim@cuhk.hk">jim@cuhk.hk</a> User: <a href="mailto:jack@cuhk.hk">jack@cuhk.hk</a> click “Request Join” button	Pop up window: You’re already in the waiting list!	Request failed

### 5.7.2 Post owner manage waiting list & participant list

#### Purpose

To test if the owner of the post can accept/reject the applicants in the waiting list, and remove participants in the participant list. Here we test under the condition that the current user is the post owner.



#### Participant:

potato@cuhk.hk [REMOVE](#)

dummy@dummy.com3 [REMOVE](#)

dummy@dummy.com2 [REMOVE](#)

#### Waiting List:

dummy@dummy.com4 [ACCEPT](#) [REJECT](#)

dummy@dummy.com5 [ACCEPT](#) [REJECT](#)

#### Testing procedure

In a post detail page,

1. press the “Accept” button to add an applicant from waiting list to participant list;
2. press “Reject ” button to remove an applicant from waiting list;
3. press “Remove” button to remove an participant from the participant list.

ID	Inputs	Expected Response
1	click “Accept” button besides an applicant	Corresponding applicant disappears from the waiting list and appears in the participant list
2	click “Reject” button besides an applicant	Corresponding applicant disappears from the waiting list
3	click “Remove” button beside an participant	Corresponding applicant disappears from the participant list

## 6. Lessons Learned

I believe the biggest difference between programming exercises and software development is that modulation and program readability is no longer a “best practice”, but a necessity. This project involves a workload that is pretty much impossible for an individual. In order to cooperate with 5 people, and combine the coding effort together seamlessly, the modules need to be coded with the convenience for other programmers in mind. This required us to enforce good programming practice such as in-line comments and clear program logic which would indeed be a very useful habits for a programmer.

This project also provided a valuable experience of making a web-based application. Some of our group mates had experience coding websites before, but never an extensive webpage that has backend support and customer usage in mind. The communication between a frontend that supports interactive feedback and a cloud-based real-time storage system has proven to be challenging but also a great learning experience.

With the quickly developing deep learning technology, we agreed that implementing a function based on that would be a great selling point of our application. We implemented our image search feature using TensorFlowJS with ease. This makes us appreciate the solid foundation that researchers had left us to implement

softwares more easily. Although the coding part does not post too many issues, the data collection phase turns out to be rather tedious. I had a few experiences using deep learning in research areas. However, the dataset is often available for public use and I do not need to collect data myself. In this project, we had to manually crawl the internet for suitable images used in location matching. This process ended up taking longer than we expected. This gives us better ideas on how to better allocate time for each phase of software development if the neural network is involved in the project again. And we do believe neural networks will become more and more applicable in different software systems as the technology matures.

## 7. Conclusion

In conclusion, NowHere is a user-friendly online platform for users to find travel partners. We set up the web application using CSS, React.js, Tensorflow with Firebase server as backend. Users can create a travel group post or join another travel group post. Users can create a post with details of their travelling destinations, such as cover photo, travelling style, location and period of travelling. Users can view the posts and request to join the travelling groups if they are interested. The post owner can choose to accept or reject the joining request. Moreover, users can search posts by their preferences. We have provided an image search function for users to upload an image. Then, the system will suggest a location that is similar to the uploaded image.

**Nowhere to Find travel partners? NowHere!**