

The Chinese University of Hong Kong
Computer Science, Project Design Final Report

TripShare

Version 1.2
May 4, 2018
Group 38

Victor Adair Najar Delgado 1155111952
Sergio Diaz Navarro 1155107144
Li, Chia-Chun 1155083965
Siu, Long Wa 1155049055
Chan, Tsz Fung 1155047974

Index

1. Introduction
 - 1.1 Project Overview
 - 1.2 Objective
 - 1.3 Highlights
 - 1.4 Project Statistics
2. System Architectural Design by DFDs
 - 2.1 System Architecture
 - 2.1.1 User registration
 - 2.1.2 Log in of a user
 - 2.1.3 Creating a new trip
 - 2.1.4 Like and dislike trips
 - 2.1.5 Comment section
 - 2.1.6 Page with most liked trips
 - 2.1.7 Show personal page
 - 2.1.8 Trip page and edit trips
 - 2.1.9 Show last trips (Explore trips)
 - 2.1.10 Search by country
 - 2.2 DFDs
3. Detailed Description of components by UMLs
 - 3.1 Architecture Diagram
 - 3.2 Use Cases Diagram
 - 3.3 UML Class Diagram
 - 3.4 Sequence Diagrams
4. User Interface Design
 - 4.1 Description of the User Interface
 - 4.2 Objects, Actions and Screen Images
 - 4.2.1 User registration
 - 4.2.2 Log in of a user
 - 4.2.3 Creating a new trip
 - 4.2.4 Like and dislike trips
 - 4.2.5 Comment section
 - 4.2.6 Page with most liked trips
 - 4.2.7 Show personal page
 - 4.2.8 Trip page and edit trips
 - 4.2.9 Show last trips (Explore trips)
 - 4.2.10 Search by country
5. Test
6. Lessons Learned
7. Conclusion

1. Introduction

1.1 Project Overview

With globalization, more and more people nowadays like to travel around the world to release the pressure during their short breaks or long vacations. Travelling becomes a common hobby for most of the people in the world, including all of our groupmates. Our project, namely “TripShare”, will develop an application related to travelling.

“TripShare” offers an online web-based platform for those trip lovers to share their trips and to collect the information on the destination they are likely to go. “TripShare” aims at providing the users with an experience in writing a personal travelling blog. Therefore, the main feature of “TripShare” is to personalize your own trip and share it to everyone in the world.

Sharing your own trip is the main feature of “TripShare”. In “TripShare”, users can simply post their pictures at the city they travelled with all kinds of experience, like the recommendation of their accommodation, their experience, and the type of travel. Their posts mainly focus on the personal experience and feeling instead of writing something unbiased.

Another feature is to view the information on certain tourist posts. After sharing the experience, it is worth viewing others’ opinions for your next trip. To look for information on certain tourist spots, the users can directly search for them on this website. “TripShare” will show the users the travel posts from others who have travelled there. The experience and comments give other users with a better understanding of these places.

Finally, when the users simply want to travel but have no idea on the destination, they can see the trend which is based on the ‘like’ they get from the users on the dashboard. They can plan a whole new trip without knowing any preparation in “TripShare”.

“TripShare” lets every user in the world know that finding a trip is no longer a harsh task with the aid of “TripShare”.

1.2 Objective

“TripShare” is a platform that the users can mainly share their trips. Through the shared posts, other users can then plan a new trip easily. Experienced travellers can inspire other users who are planning a trip through sharing their travelling blog. People who are planning a trip can find some resources by reading all the existed comments and post in the “TripShare”. In addition, the users can share their potential plan so that they can find a travelling partners through “TripShare”.

The ultimate goal of “TripShare” is to make travelling easier. It helps the users know more about the places they are going to travel; thus, people can prepare for their trip more properly. “TripShare” is also a kind of record in the users’ life. They store every memory in their trips in “TripShare” and share them to others. When they stop their footsteps and look back to their own life, “TripShare” will be their footprint on our Earth.

“TripShare” is actually a self-contained web application. All the resources come from the users. Like OpenRice in Hong Kong and TripAdvisor, the system only provides the users with the places they can travel. The rest of jobs will leave to the users. The difference in current application is that “TripShare” is a blog-based application. The users write a travelling blog and mark the places into the system. It will automatically share the corresponding parts to other users.

1.3 Highlight

The main feature of “TripShare” is the blogging function. The users can log in to our system with their own account and upload anything about their last trip including photos and comment. After the users uploading the photos and commenting some particular part of the trip such as food and accommodation, they can share what they uploaded to the others. Sharing is also one of the key feature of our project. The users can search and browse all the comments which shared by other users in the system. The user can also give like or dislike for each public trip. Therefore, the user can use those as a reference when they are using “TripShare” for planning for their trips. If the users have no idea about the place for their next trip, there is also a function for them. The trendy page shows the 10 top countries to travel and the most popular post of that country.

Our target users are the general public. Everyone who is interested in travelling will be our target. All the users are both customer and provider. We hope the users can get some useful information on the other post when they are planning for the trip. Also we hope that they can contribute back from writing their own comment for the trip so the other users can have more information as their reference. Once most of the user can be how we expected, the database will become more and more richer, that is our ultimate goal as well. However, to achieve our ultimate goal, we need some marketing strategy to promote our project. In the very beginning, there is not much data in our database. Therefore, our target will be some professional travelling bloggers. Not only our database get richer, but also we can promote “TripShare” by the reputation of those professional travelling bloggers. After building a rich

database, our target will shift to those who always search for the information of travelling. As they can make use of the built data from the travelling blogger to plan for their trips, and they will share “TripShare” to their friends if they found the information is useful. Therefore, the market of “TripShare” will grow bigger. Since the market of “TripShare” is already big enough after the previous stages, our target can finally shift to the general public. No matter those who are a blogger, experienced travellers or people with no experience on travelling, they are all our target. At this last stage, the whole application is self-contained. All the users will share their own trips and search for the information for their next trips. This is our ultimate goal for “TripShare”.

1.4 Project Statistics

The below is the McCabe's Cyclomatic number for our major functions. The McCabe's Cyclomatic number is calculated by the following formula: $E - N + 2*P$ where E is the number of edges of the flow graph, N is the number of nodes of the flow graph and P is the numbers of disconnected part of the flow graph.

The higher the McCabe's Cyclomatic number is, the more difficult, the more difficult for people to understand the programs. Also, the McCabe's Cyclomatic number indicates the numbers of test case to execute all paths in the program.

Class CitiesController

function	edge	node	disconnected part
new	2	3	1
create	11	11	1
edit	1	2	1
update	5	5	1
destroy	3	4	2
city_params	1	2	1
Total	23	27	7

The McCabe's Cyclomatic number in this class is $23 - 27 + 2 * 7 = 10$.

Class CommentsController

function	edge	node	disconnected part
new	2	3	1
create	12	12	1
destroy	3	4	2
comment_params	1	2	1
Total	18	21	5

The McCabe's Cyclomatic number in this class is $18 - 21 + 2 * 5 = 7$.

Class PicturesController

function	edge	node	disconnected part
new	2	3	1
create	11	11	1
edit	1	2	1
update	6	6	1
destroy	4	5	2
picture_params	1	2	1
Total	25	29	7

The McCabe's Cyclomatic number in this class is $25 - 29 + 2 * 7 = 10$.

Class StaticPagesController

function	edge	node	disconnected part
home	4	5	1
Total	4	5	1

The McCabe's Cyclomatic number in this class is $4 - 5 + 2 * 1 = 1$.

Class TripsController

function	edge	node	disconnected part
index	4	4	1
edit	1	2	1
show	5	6	1
new	1	2	1
create	8	8	1
update	5	5	1
like	7	7	1
destroy	4	5	2
trip_params	1	2	1
correct_user	4	4	1
Total	40	45	11

The McCabe's Cyclomatic number in this class is $40 - 45 + 2 * 11 = 17$.

Class UsersController

function	edge	node	disconnected part
index	1	2	1
show	2	3	1
edit	2	3	1
create	8	8	1
update	6	6	1
destroy	4	5	1
set_user	1	2	1
user_params	1	2	1
correct_user	2	3	2
Total	27	34	10

The McCabe's Cyclomatic number in this class is $27 - 34 + 2 * 10 = 13$.

2. System Architectural Design by DFDs

2.1 System architectural

The next information is a compilation of all the useful and functional requirements. After the brief description a DFD is shown with the interaction of the user and this requirements.

2.1.1 User registration

To register, the user will need to give a username, an email, and a password. The password must be given twice for confirmation.

2.1.2 Log in of a user

To login the user must insert a valid username and password.

2.1.3 Creating a new trip

The user is able to create new trips specifying the country and the dates for general use.

Then he/she needs to add a city specifying the accommodation (recommendations and comments) and information of that city (good decisions, bad decisions, what to visit?).

At the end, the user has the option to post some pictures of the trip by uploading them to the system.

2.1.4 Like and dislike trips

Every public trip has a section of likes and dislikes. Meaning that all registered users can give their votes. The number of likes and dislikes will be shown beside the buttons.

2.1.5 Comment section

Every trip has a comment section where registered users can give their opinions about that post. The comments are public for everyone, but only the users that logged in are able to post something.

2.1.6 Page with most liked trips

The home page shows the most popular places based on the number of likes. It is updated the whole time and is public for every user.

2.1.7 Show personal page

The personal page displays the user's personal trips. It has all the trips from newest to oldest, with an option to modify them (only for the owner).

2.1.8 Trip page and edit trips

Based on the personal page, every user has the opportunity to modify their old posts by clicking the button of *edit*. By doing so, the user has the ability to modify the whole trip and even add some pictures.

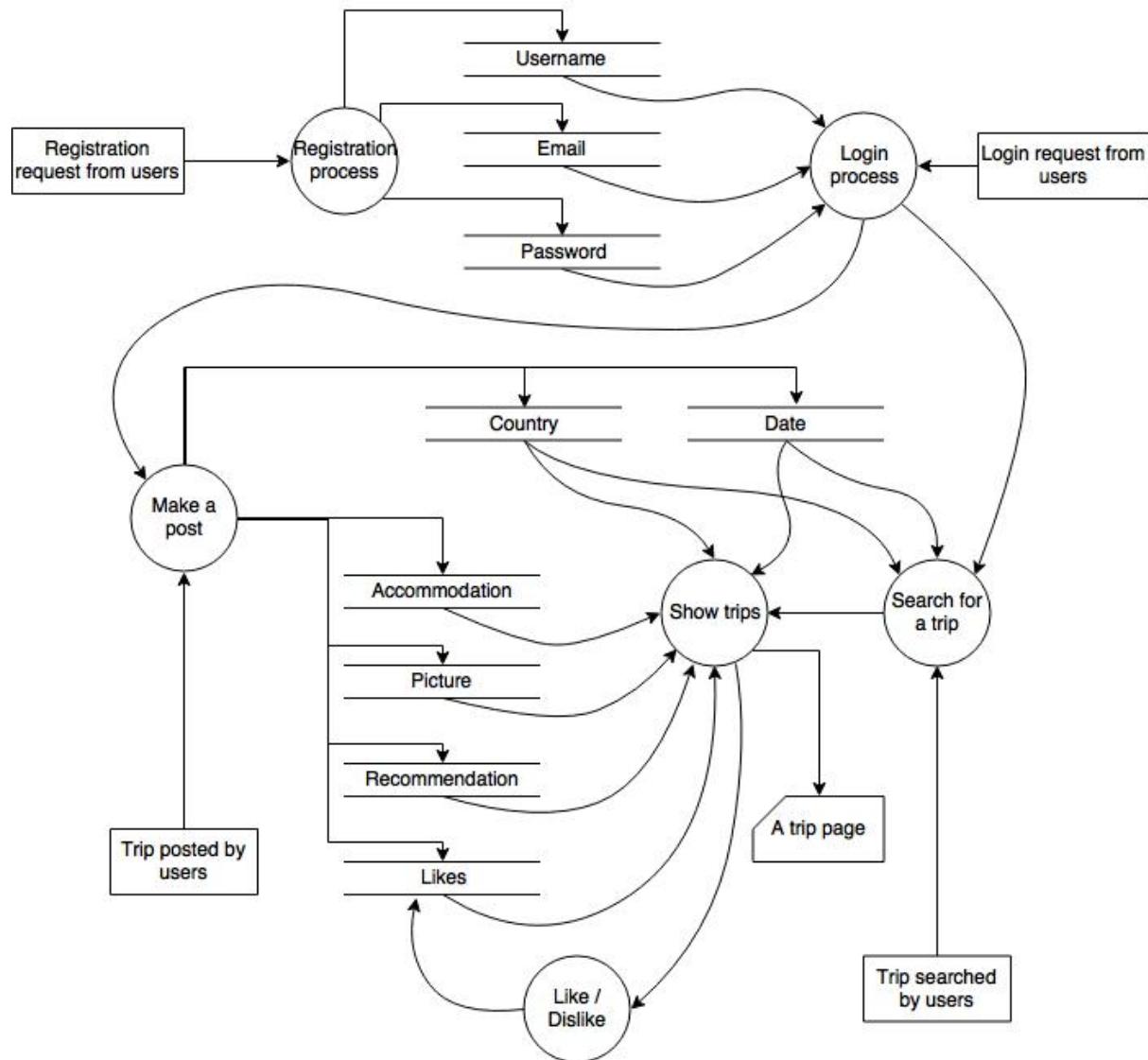
2.1.9 Show last trips (Explore trips)

The Explore page shows all the trips registered in the app. The page displays the most recent trips at the top and the oldest ones at the bottom. The search bar is also included for a better experience.

2.1.10 Search by country

A public search bar can be used to search trips for a specific country or username. The bar will display all the countries available when searching countries or an input textbox when searching usernames.

2.2 DFDs



For this DFD let's start with the registration process. When doing this the user needs a username, an email and a password, the same for the login. Once login, the user has all the options available, like making a post, like or dislike posts, making comments and search trips. As you can see from the DFD, every single element needed for a function is stored in our database.

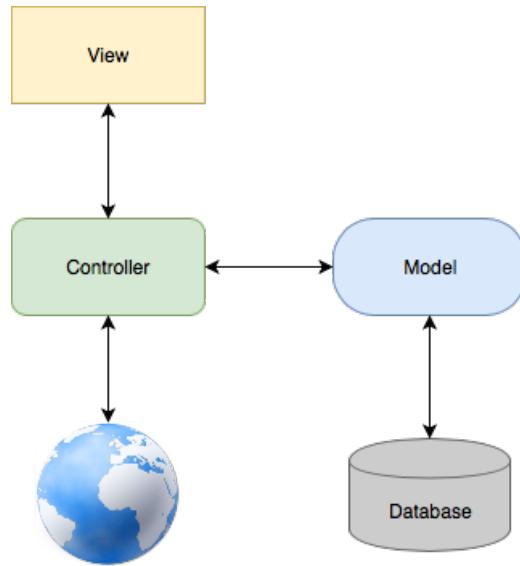
3. Detailed Description of components by UMLs

3.1 Architecture Diagram

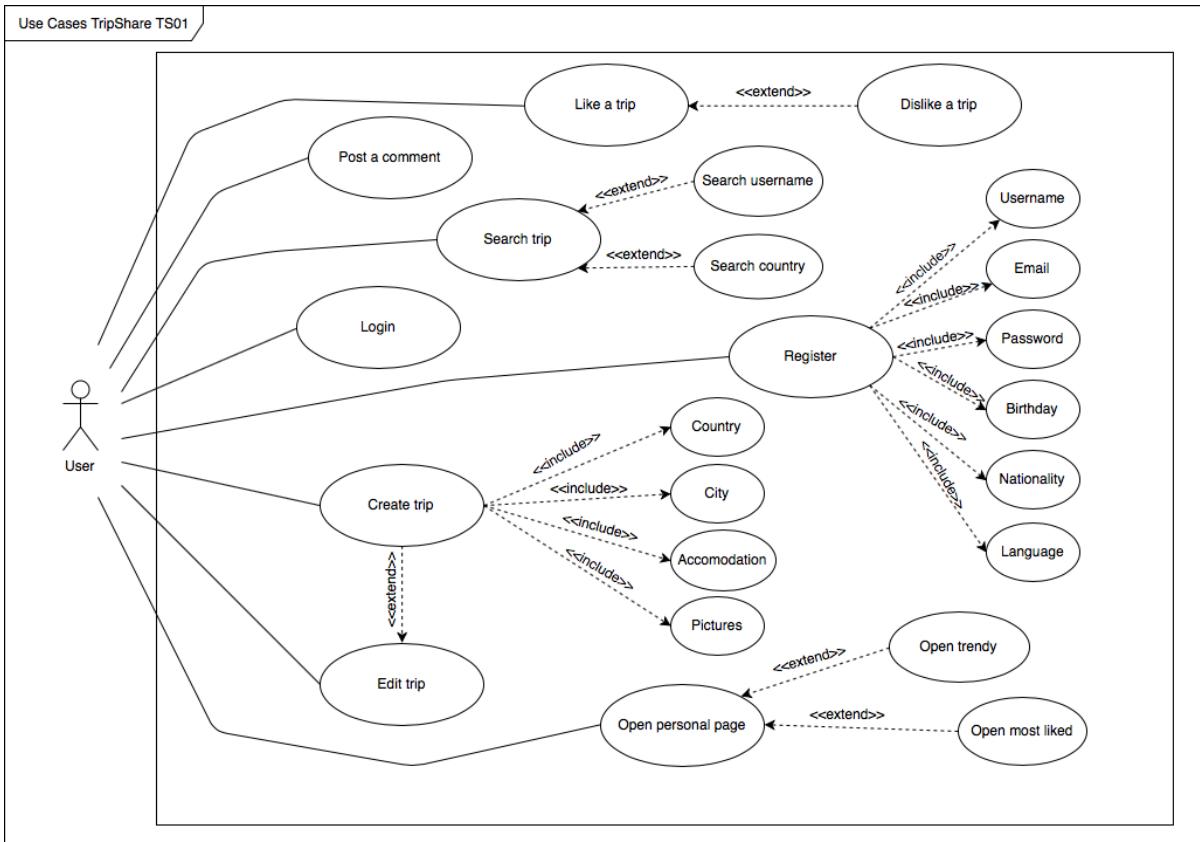
The whole application is build by using the Ruby on Rails web framework. This framework is totally written in Ruby programming language and is based on Model-view-

controller (MVC) architectural pattern, which means that it divides the software into three interconnected parts that interact with each other.

The whole process with Rails work like this: a browser sends a *request*, which is received by a web server that passes everything to Rails *controller*. After that, there are two options, the first one is to immediately render a *view* and send it back to the browser in form HTML or, the second one and most common, the controller interacts with *model*, which is a Ruby object that communicates with the *database*, then it renders the view and returns the web page to the browser as HTML. (Hartl, 2018) The next diagram shows the whole process.

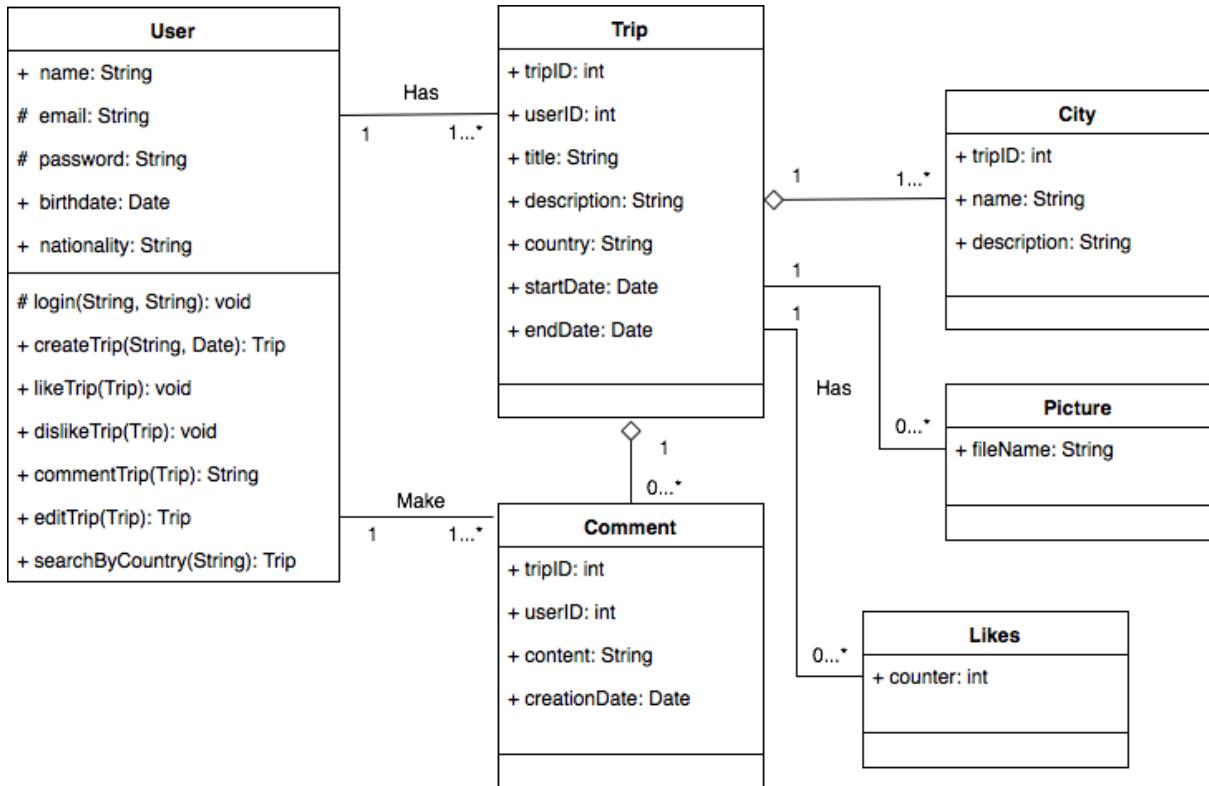


3.2 Use Cases Diagram



Code	TS01
Name	Use Cases TripShare
Actors	User
Description	<p>This web application lets the users know more about a travel experience as well as publish their own trips to share with the community.</p> <p>Every user can register, log in, create a trip, edit an existing trip, search trips, search countries, search other users, like or dislike a post, post a comment, and open the personal page, trendy page or most liked page.</p>

3.3 UML Class Diagram



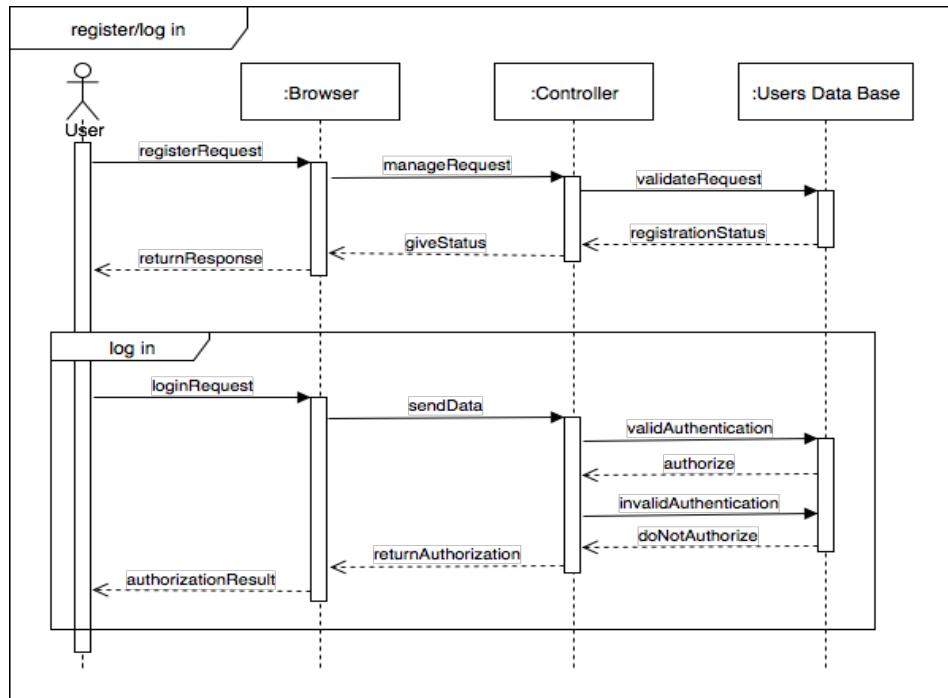
For our web application we have 6 different classes related with each other. The most important one is the User class. This class is the representation of all the people that uses the application. The user represented here is the type of user that needs to register into the system, so it is also the type of user that has all the options available. As you can see, the User class is the only one with functions, since is the one that “creates” all of the rest, but at the same time the rest are independent from it. Next, the Trip class. This class is the one that has all the information of a trip and the one that can handle different functions from the User. By doing this this class uses the rest of the classes left: Comment, Likes, Picture and City.

3.4 Sequence Diagrams

3.4.1 To register new user and log in:

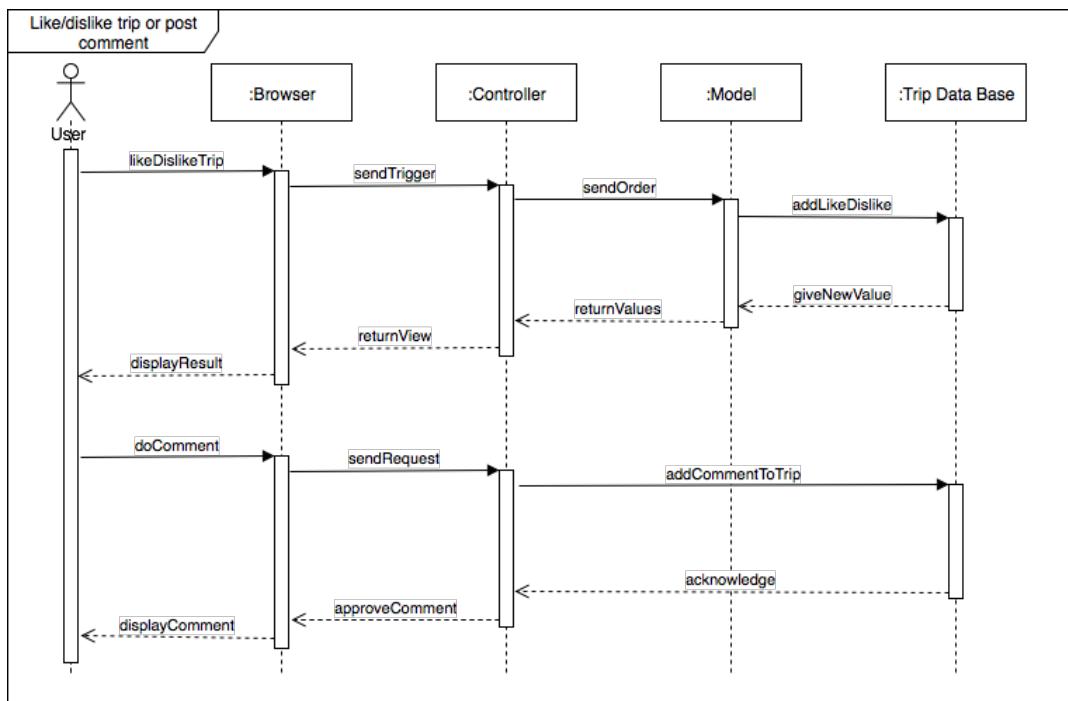
In order to register a new user, the user needs to send a register request using the browser with all the information previously filled. After that the browser sends the request to the Controller which sends the request for validation to the Users Database. If no information is repeated, then the Users Database sends a confirmation. As for

the login request, the user gives the email and password which are sent for verification.



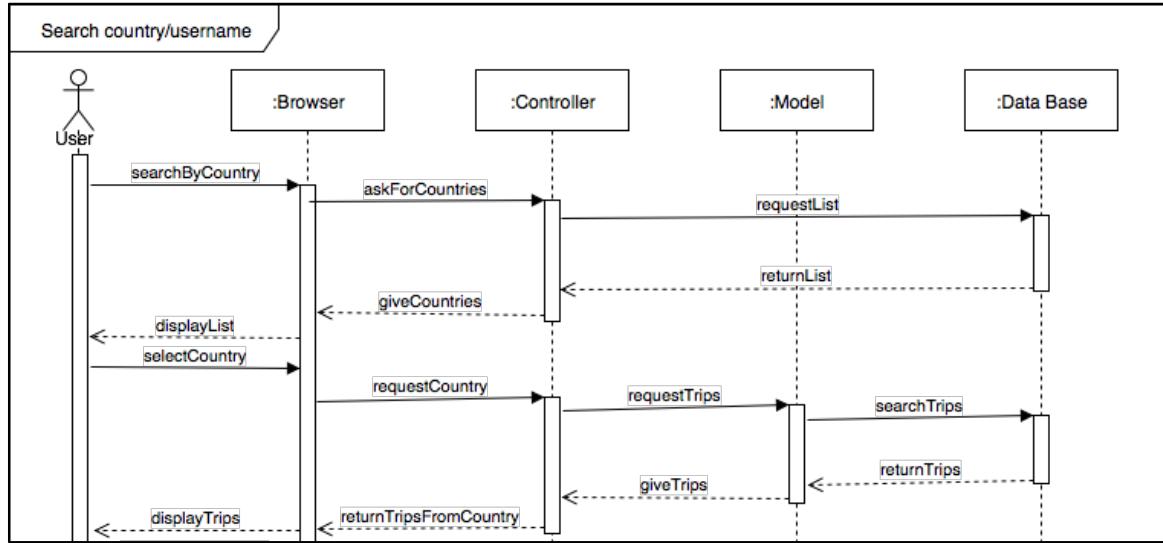
3.4.2 To like or dislike a trip and comment:

Once in the trip the user clicks on like or dislike, after that, the request passes through the Controller and the Model just to add a like or subtract it. The same goes for comments, the only difference is that the Database stores a String instead of just adding a number.



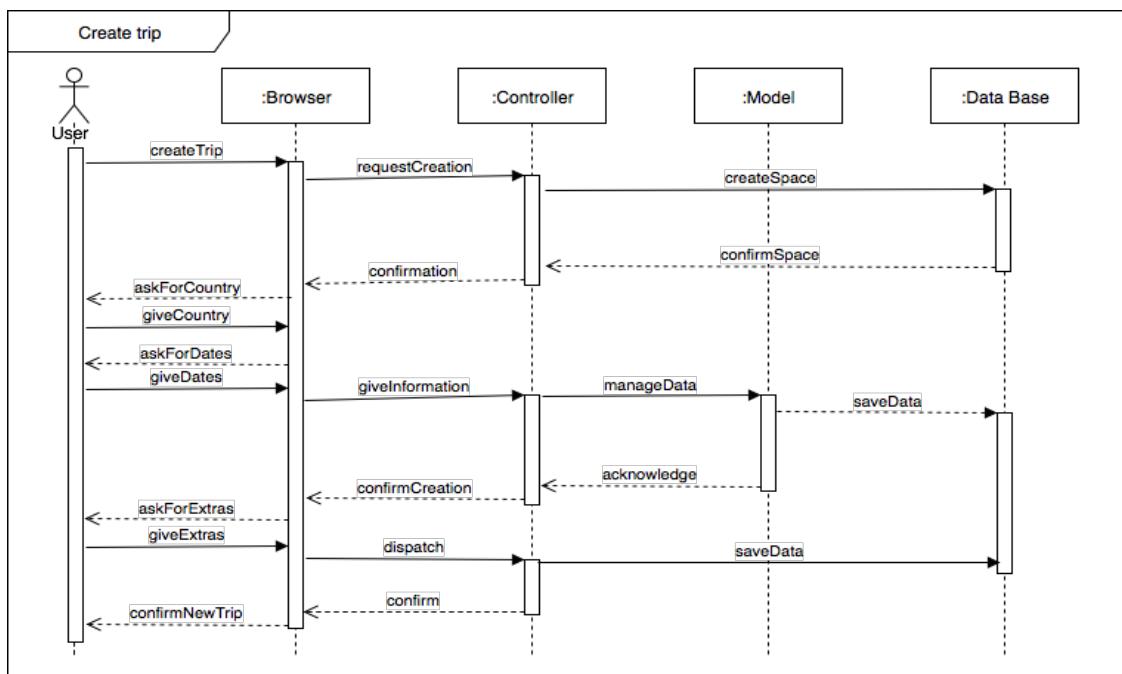
3.4.3 To search a trip by country:

When searching for a country, the search bar gives the user some options to search about, it is just option of the user to take that prediction or search by they own. Once they search it, the Database will give all the trips related to that country.



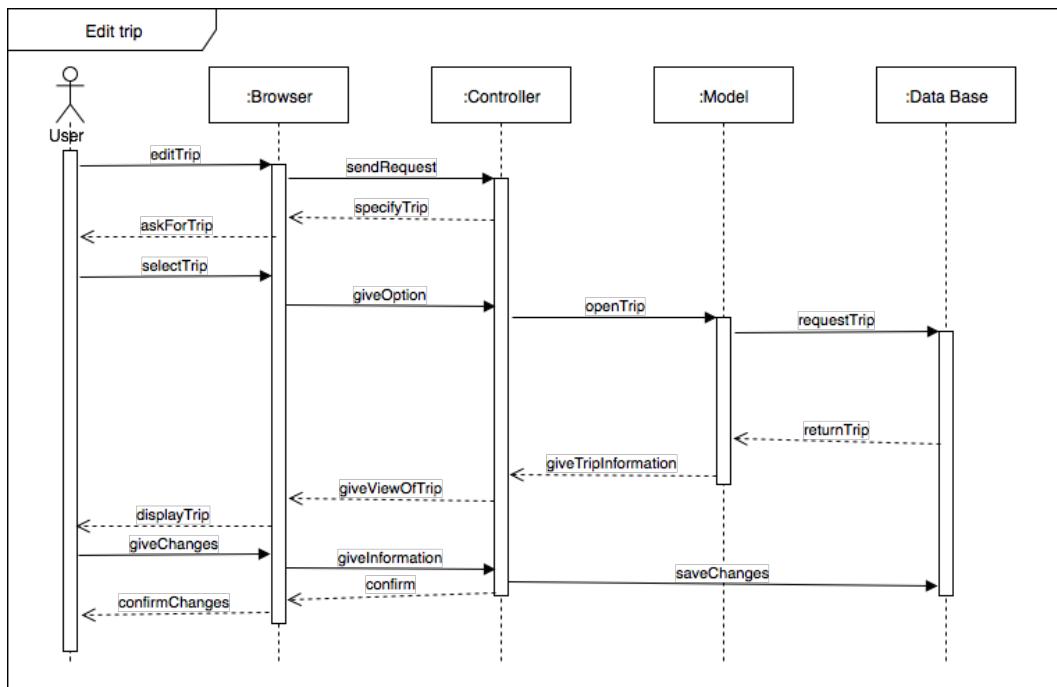
3.4.4 To create a trip:

In order to create a trip the user needs to request a form. Once accepted, the Database will give a form for the user to fill; this interaction is just with the user and the browser. When the basic information is filled, the user send that information through the browser to the Controller which then gives the data to the Model that manages it and selects the information that is going to be stored in the Database. If the user wants to add more data, the same procedure is followed.



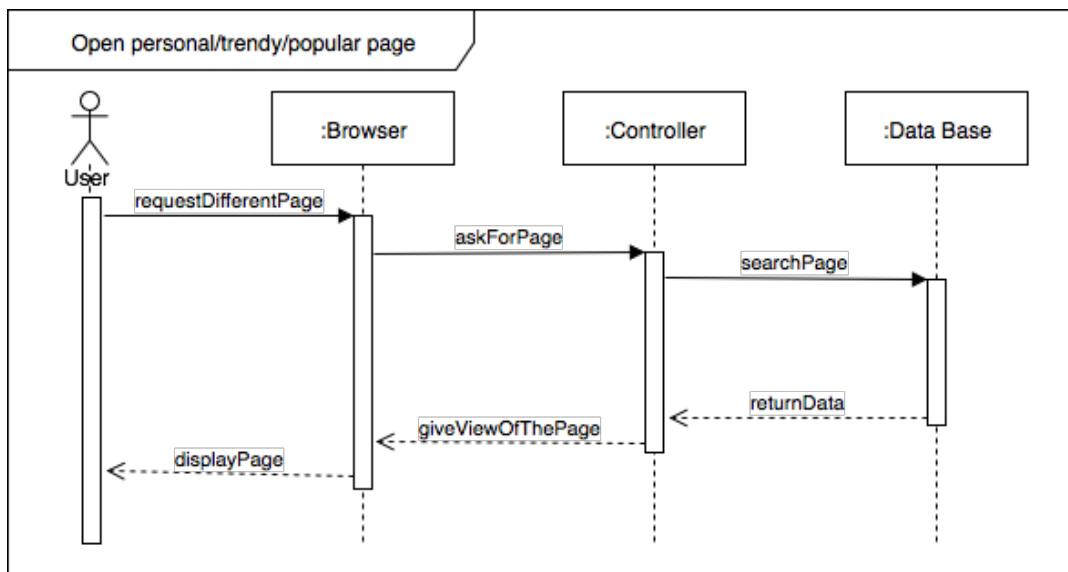
3.4.5 To edit a trip:

When a user wants to edit a trip, the option of doing it is once the user is inside the trip. Only the owner can do this, and to do so, the user has to select which trip he/she wants to edit. When the choice is made, the Database returns all the information of the trip, which is displayed on the browser. That information could be changed by the user, and when submitted, it goes directly to the Database without checking because the trip was already stored in the system.



3.4.6 To open the personal, home, or explore page:

When clicking on a tab, the browser asks the Controller for that page, once the Database has it, it will return all data to load the page.



4. User Interface Design

4.1 Description of the User Interface

As the main feature of “TripShare” is sharing the trips and getting information about the countries, the design should be efficient, concise and user-friendly.

In purpose of attracting user to use “TripShare”, efficient is one of the most important part in user interface design. The design will guide the user in every step making each process fast and reaching a page or opening a trip in less than 3 clicks after logging in. Also, concise is one of the characteristics that attracts users.

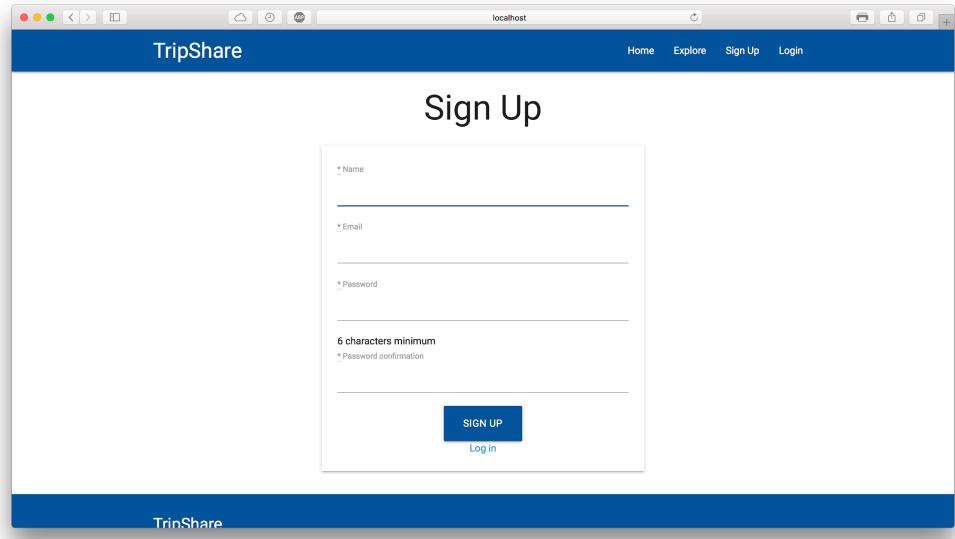
As we mentioned, “TripShare” is focusing on sharing the trips and getting trips’ information, we decided only to make some simple function. In the page, there is not that much information, but only the comment and the pictures from the owner of the post and the comment and likes/dislikes from other users. That helps the users to get the information that they want easier and faster. User-friendly is characteristic of our interface design as well.

To attract users, it must not be difficult to manipulate. All the function in “TripShare” is easy to manipulate. Most of the function only need users to click a button or enter words. For example, if the users want to like or dislike the trips, just click a button of “like” or “dislike”. If the users want to comment for the trips, just need to enter the sentence in the text box and click the enter button. Therefore, the users do not need to take a long time to study how to use our application.

4.2 Object, Actions and Screen images

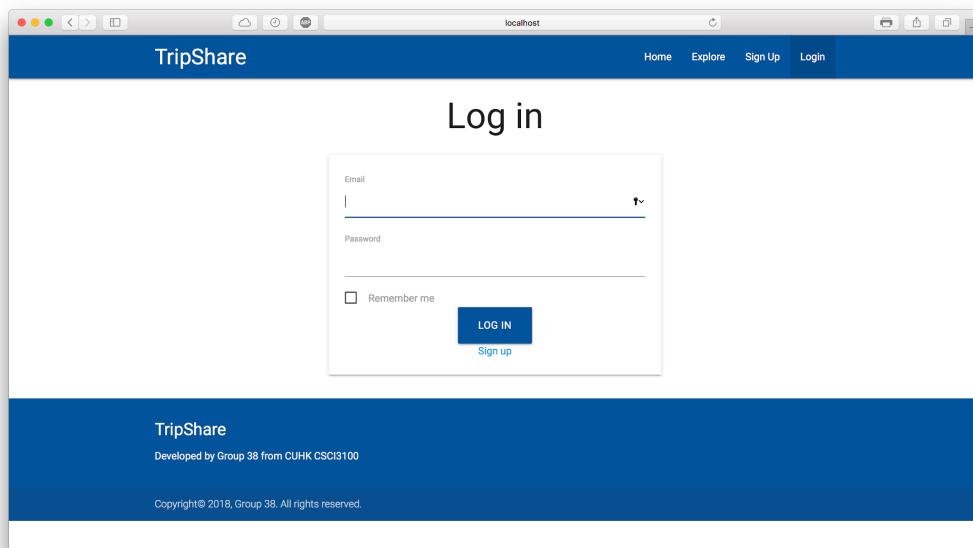
4.2.1 User registration

Users can register in the Sign Up tab by giving their name, a valid email and a valid password. If the email is not valid or is already registered, the user would not be able to register. Also, if the password is less than 6 characters or the password confirmation is wrong, the user would not be able to register.



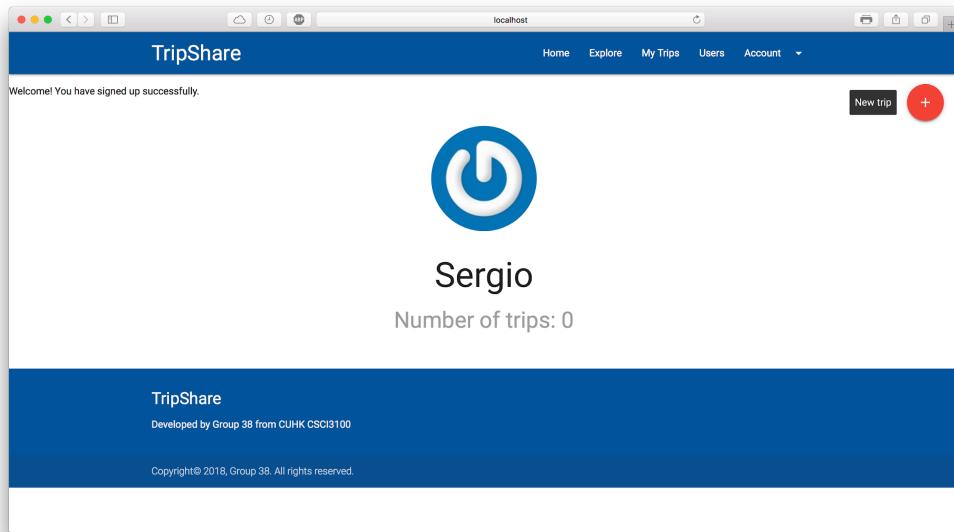
4.2.2 Log in of a user

A registered user can log in at any moment by providing the registered email and correct password. After login, the My Trips tab and Users tab are available for the user.

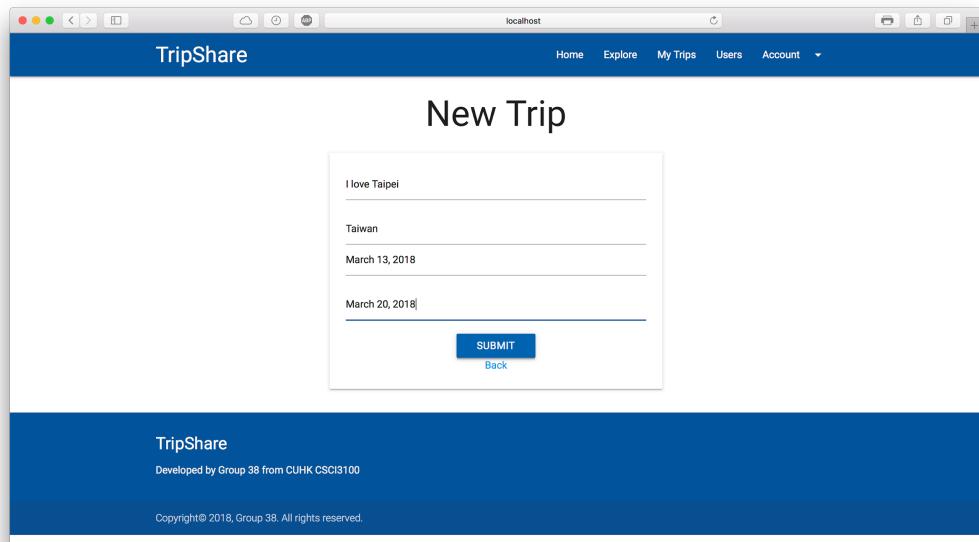


4.2.3 Creating a new trip

The users can create a new trip after clicking the “New trip” button. This button is located at the upper right corner in the “My Trips” page.

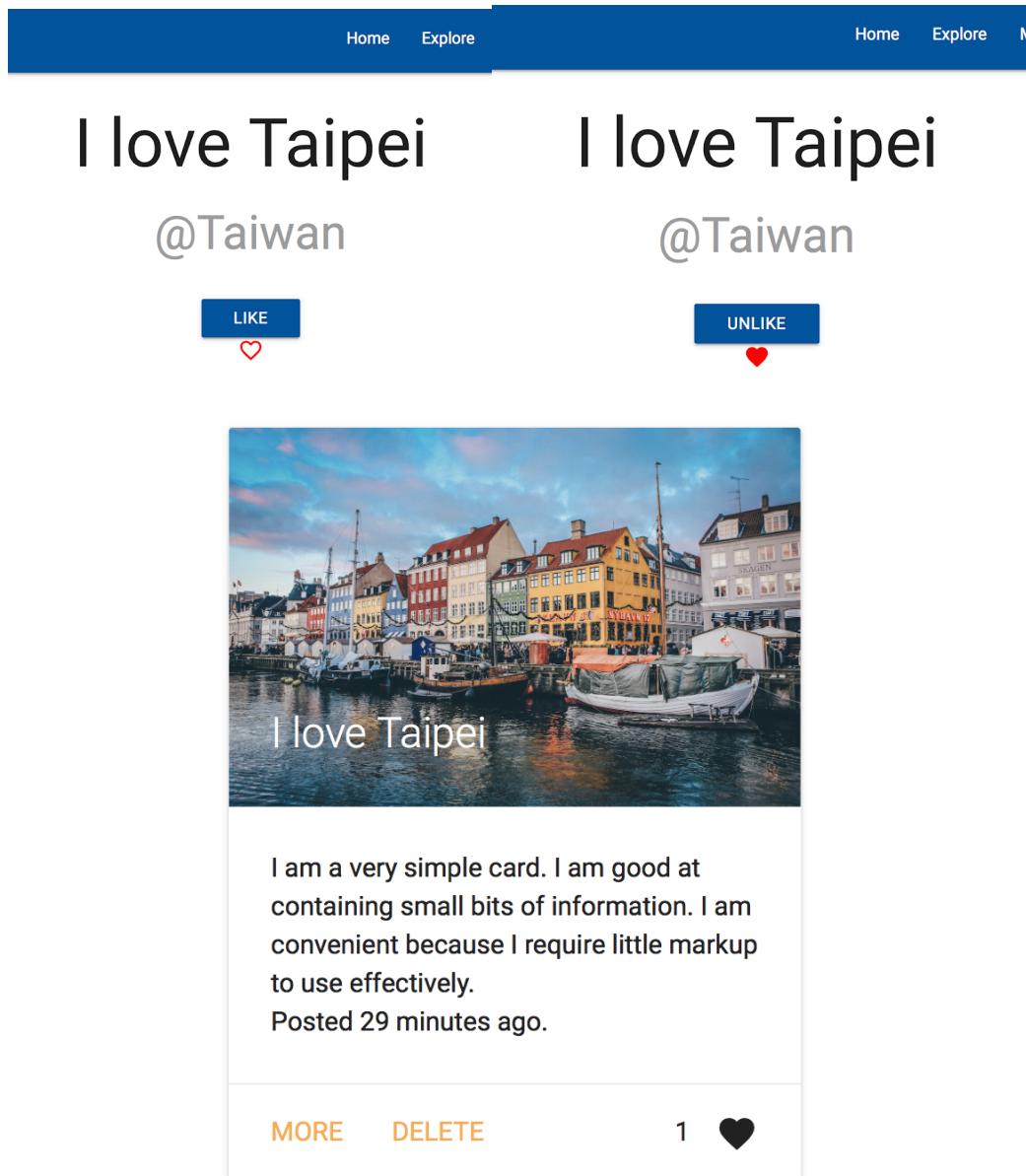


After clicking the button, the “New Trip” page appears. Now the user should input a name for the trip, the country, start date and finish date. By clicking submit, the application adds this trip to the user’s personal account and is also available for any user to see.



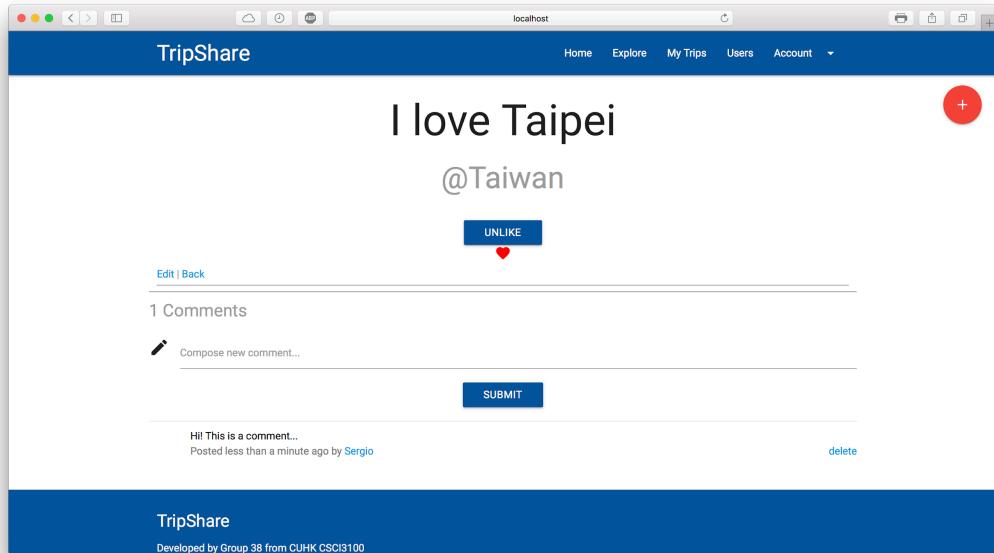
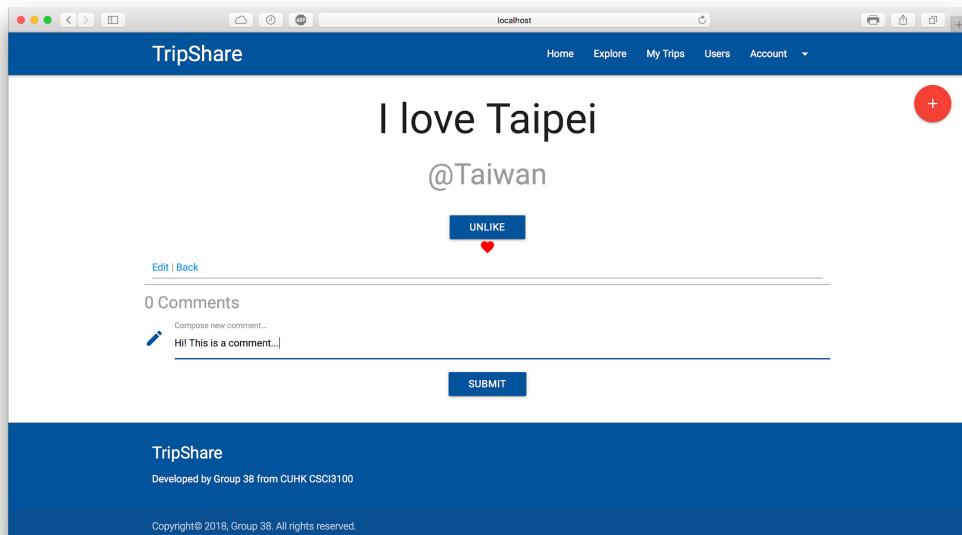
4.2.4 Like and disliked trips

When looking at a trip, a registered user is able to like or dislike the trip. A counter for that trip adds or subtracts the number of likes and displays them on the overview of the trip.



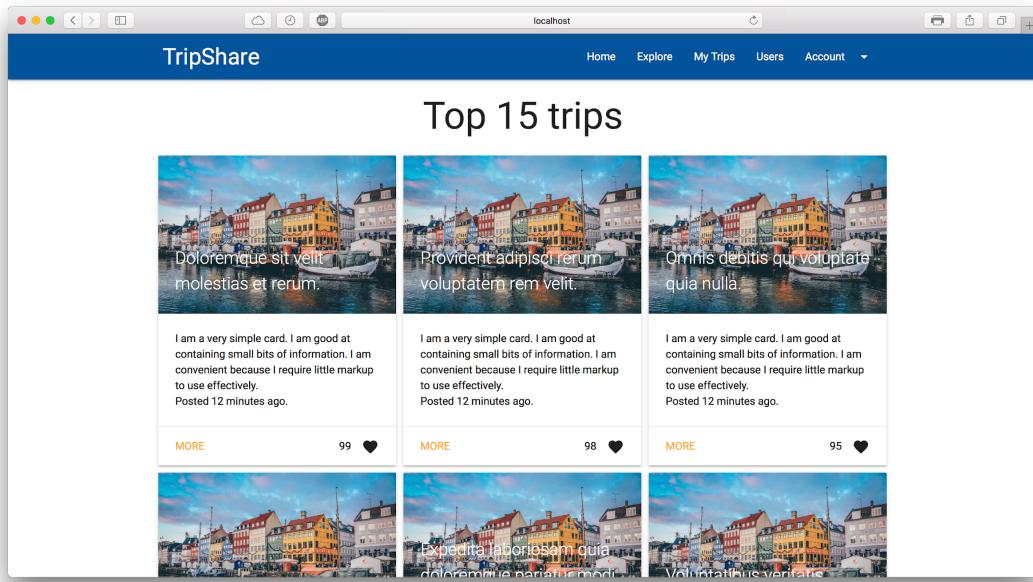
4.2.5 Comment section

When looking at a trip, there is always a comment section showing all the comments that registered users made. In that same section, a registered user is able to make a comment and submit it. The comment shows a the message, who made it and an option to delete it (only for the comment owner).



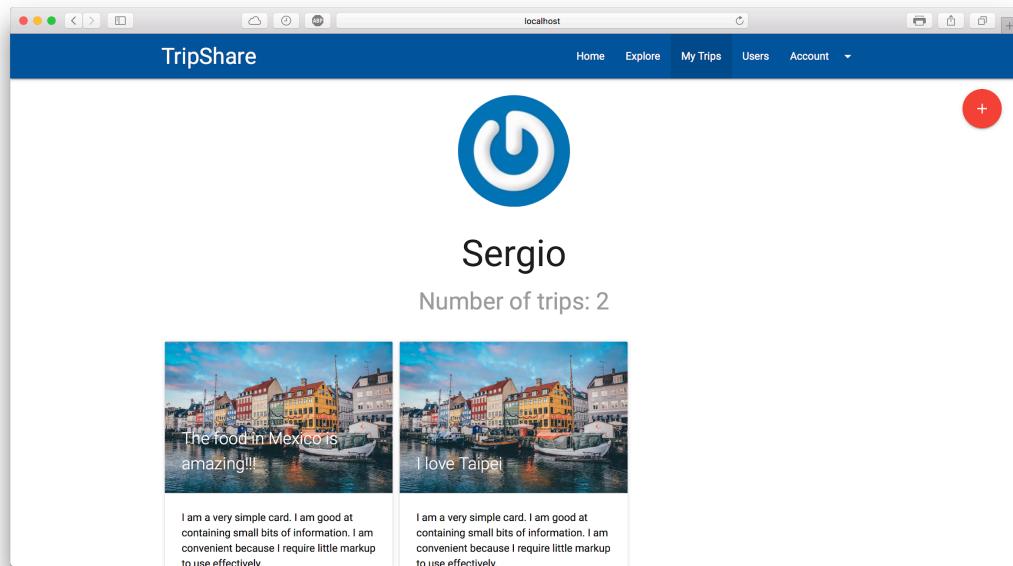
4.2.6 Page with most liked trips

The “Home” page is divided in two parts: the first one is a search bar which is described later, the second part is the top 15 trips. This section shows the most liked trips in order of likes. In this page, any user, including not registered ones, can take a look at the displayed trips and all their details.



4.2.7 Show personal page

The “My Trips” page displays all the trips made by the user. This page is only accessible by the owner, which can modify and delete any trips that he/she likes. This screen is reachable by other users (that are registered) finding the user on the “Users” page, the difference is that other users can’t make changes to these trips.



4.2.8 Trip page and edit trips

The following image shows how a trip page looks like. When the owner of the trip opens this page, he/she is able to add pictures, cities or even edit the current information.

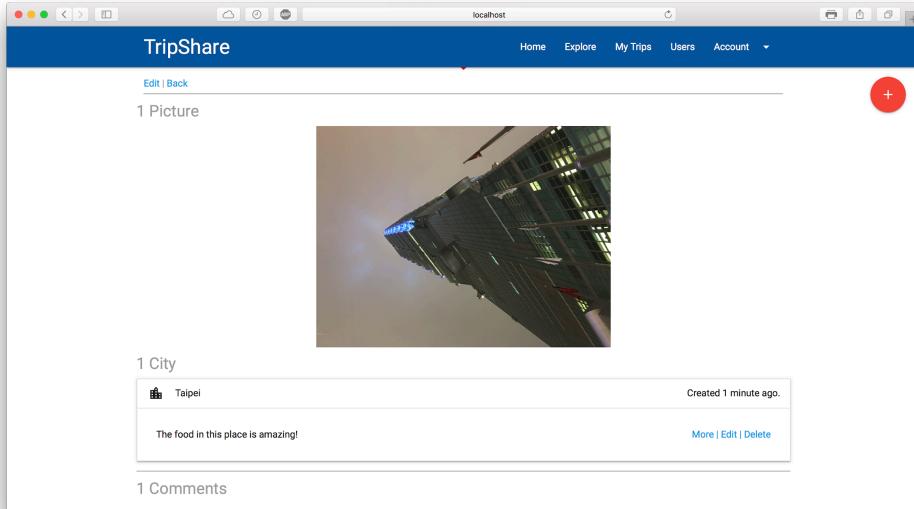
The screenshot shows a web browser window for 'TripShare' on a Mac OS X system. The main content area displays a city profile for 'Taipei' with the subtitle '@Taiwan'. At the top right are three circular buttons: a red one with a white plus sign, a blue one labeled 'PICTUR', and a blue one labeled 'CITY'. Below the title are 'UNLIKE' and 'LIKE' buttons. A link 'Edit | Back' is present. Underneath is a section for 'Comments' with a 'Compose new comment...' input field and a 'SUBMIT' button. A single comment from 'Sergio' is listed, posted 'about 1 hour ago'. The footer contains the 'TripShare' logo and developer information: 'Developed by Group 38 from CUHK CSCI3100'.

When adding a city, the application asks for the name of the city and a description, after submission the page shows the new city with the description. Any city can be modified or deleted after adding it. The more option displays a page with the information of the city in a bigger format.

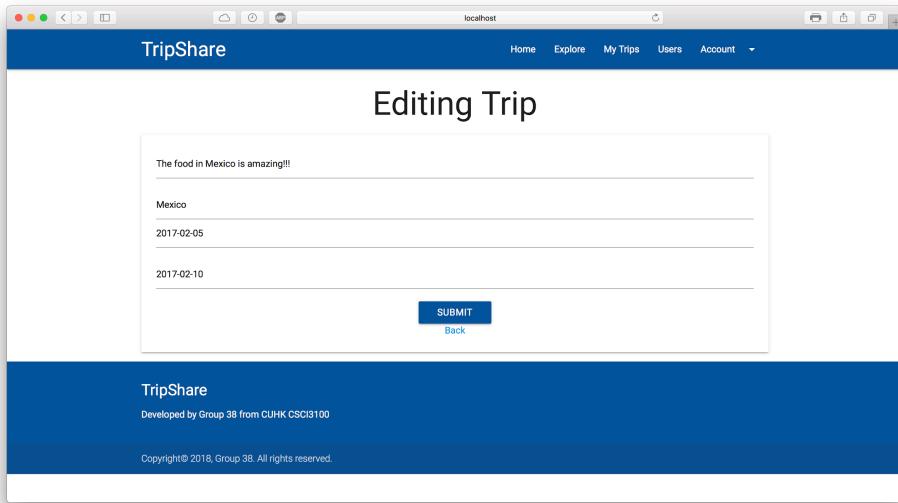
This screenshot shows the same 'TripShare' application interface, but the main content is now a detailed view of the 'Taipei' city profile. It includes the city name, a thumbnail image, a description ('The food in this place is amazing!'), and creation timestamp ('Created less than a minute ago'). It also includes 'More | Edit | Delete' links. The rest of the interface is identical to the first screenshot, including the header, sidebar buttons, and footer.

This screenshot shows a simplified version of the city profile for 'Taipei'. It only displays the city name and a brief description ('The food in this place is amazing!'). The rest of the interface, including the header, sidebar buttons, and footer, is identical to the previous screenshots.

As mentioned before, a user could also add multiple pictures to the trip, by clicking add picture. A new page will come out and the user should select the image, after submitting it, it will appear on the trip page.



If the users wants to edit the whole trip, he/she can select the edit option at the top left side and a new page with the information of the trip appears. Any modification is stored in the database.



4.2.9 Show last trips (Explore trips)

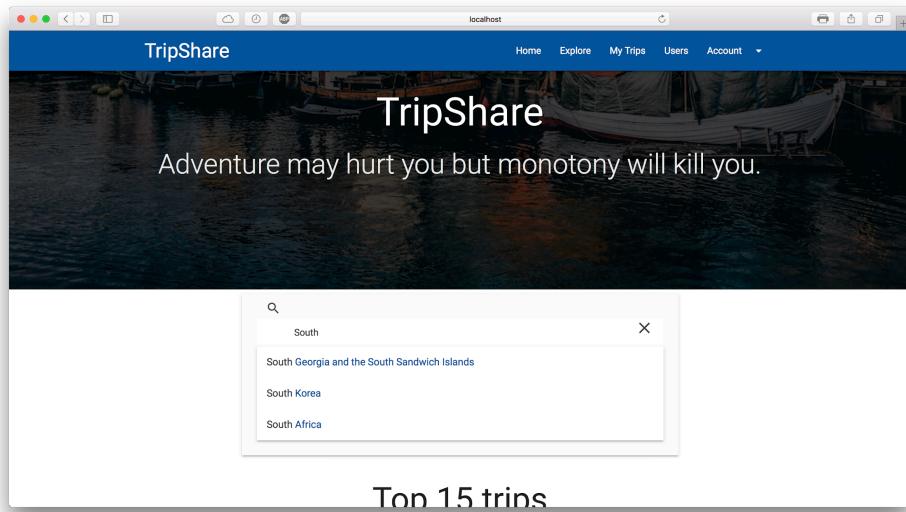
The “Explore” page shows all the trips of all users in the database. Everytime a new trip is added, this page will display the country, the title, the owner and the number of likes. The list is ordered with the most recent trips at the top and the oldest ones at the bottom. A big number at the top of the page shows the number of experiences in the application.

The screenshot shows the TripShare application's Home page. At the top, there is a navigation bar with links for Home, Explore, My Trips, Users, and Account. Below the navigation bar, a large heading says "Number of experiences: 62". Underneath this, there is a search bar with the placeholder "Search Trips". A table follows, displaying six rows of travel experiences. The columns are Country, Title, Owner, and Likes.

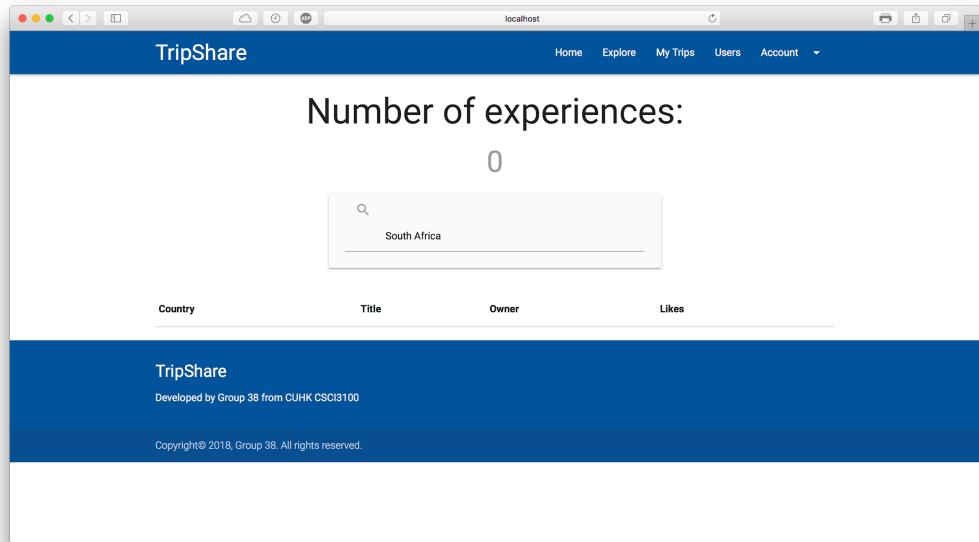
Country	Title	Owner	Likes
Mexico	The food in Mexico is amazing!!	Sergio	1
Taiwan	I love Taipei	Sergio	1
Tokelau	Sed repellat aspernatur id.	Ida Heaney	11
Vietnam	Atque laborum vero debitis rerum ut nisi qui ea.	Mrs. Judson Dickinson	58
Brazil	Sint consequuntur numquam dolores ea.	Roy Beier	19
Libyan Arab Jamahiriya	Voluptatibus veritatis voluptatem ut.	two	92

4.2.10 Search by country

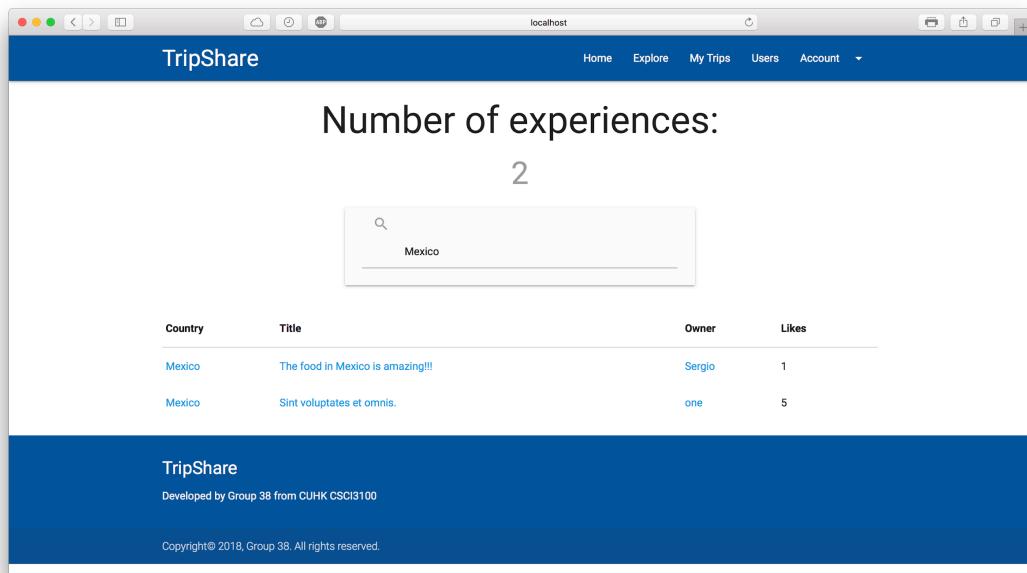
In the application we included two search bars. They both work the same way but they are in different locations. The first one is located in the “Home” page and the second one in the “Explore” page. This search bar helps users to find a trip related to the country that they choose. The search bar uses a prediction option to help the users find the country that they need, as shown in the next picture.



When clicking enter, if the country has no records, the page will show an empty list saying that the number of experiences is 0.



However, if the country has records, the page shows all the trips related to that country and also displays the number of experiences.



5. Test

Testing strategy – Black-box testing

In this project, we do functional testing to test if our web application works as what we expect. According to our requirement specification, it includes functional requirements and non-functional requirements. However, in this part, we will focus on the functional requirements. There are ten main functions specified, which includes:

- (1) register

- (2) log in
- (3) create a new trip
- (4) like
- (5) comment
- (6) a page with the trend based on the number of likes
- (7) personal page showing their trips ordered by the time it is posted
- (8) modification function to their previous post
- (9) show the trendy countries
- (10) a search bar with search function by country or by username.

Then, we do the test function by function and step by step. Also, as our application offer membership to our users, we test it in two different levels, which are “guest” and “member.”

To begin with, we go through all the function with “guest.”

Firstly, we open the website and make sure (1) the registration function works. The function matches the requirement when it successes to check if the username, email, and password are valid. In the stage, we haven’t adopted the optional information listed in the specification. After filling the blanks, we successfully register a user, and the system automatically logs in to the newly-created account. However, to test (2) the login function, we log out and log in again, which make sure it works and checks membership properly. To test with “guest,” we log out one more time.

Next, for (3) create a new trip, (4) like a trip, and (5) comment functions, they are all for “member.” So, for “guest,” we have to make sure they are banned to do so. In their pages, they may not see the tab for creating trips, but they may see there are a common block and a heart for like function. To make it user-friendly, we will redirect them to registration page if they try to like or comment on a trip.

Then, (6) the feed on the home page shows the trip with the most likes. We check if the “guest” can properly view the feed.

For (7) personal page and (8) modification function to their previous post, they are also the functions for “member”. So, we make sure the functions are hidden from “guest” and they will not see these tabs on their site.

Next, for (9) trendy countries, we check if they appear properly in “guest” site.

Finally, we check if (10) the search bar works well and if the auto-complete functions.

We complete the first level of our “guest”, and the next stage is “member.”

We checked (1) the register function and (2) log in function with “guest” before. So, firstly, for “member,” we have to make sure the two functions are properly hidden from them.

After logging into the account, we registered before, we check if the tabs on the upper side of the website are different from those of “guest.” Then, to test (3) create a new trip, we have to make sure (7) personal page works. So, we click into the profile of the “member.” It shows as what we expected, that is, no any trip record has been posted. Therefore, we try to create a new trip by clicking the button on the upper right corner. Then, we add a title and a city for the trip. The system will redirect the “member” back to his profile page and we can see one trip is added.

For better testing (4) like a trip, and (5) comment functions, we open an incognito window and log in to the “member” pre-built. And in the meanwhile, we can test (10) the search bar with “member” to see if we can get find the trip we just created. After finding it, we click into it and test the like and comment function. Then, we jump back to the first member, and we can see the number of like increase and the comment appear in the comment block.

Then, (6) the feed on the home page shows the trip with the most likes. We check if the “member” can properly view the feed.

For (8) modification function to their previous post, we test it by the trip we created just now. We click on the edit button on the card of the trip, and we test if it changes after we change its title and description.

Finally, it is (9) trendy countries. We check if they appear properly in “member” site.

The above steps are the functional testing for our web application. In the process of testing, we fix some bugs and handle the exceptions due to the difference of levels of users.

On the other hand, an additional penetration test was performed in order to protect the application from attacking. Brakeman is used in the penetration test

6. Lessons Learned

As we had to do a project design document before we started coding, which is different from the other courses. In the project design document, we did some preparation works such as stating requirement specification, drawing data flow diagrams, architectural diagrams, use case diagrams, UML class diagrams and sequence diagrams. Those are what we had not tried to do for other projects before and those are learned from this course. Also, after we finished the coding, we learned how to do testing for the application.

Since the main theme of our project is travel, which is a common interest of our group members, all of us are enjoying to participate in this project.

As we already had some ideas about sharing our travelling experiences, it is not too difficult for us to confirm what functions should be added into the application. After we had confirmed all the details about the application, we focused on coding and testing. The coding part is more easier than we thought due to the great preparation of the project. Generally, we enjoy doing this project since it is what we interested in.

If we can re-do the project again, we think we can add more function into the application. Although there is many functions in the application, most of them are not innovative enough. Most of the functions are already been used in different application or website, such as like/dislike, comment, searching. We should think about more new functions to make our application be different from the others.

7. Conclusion

In the introduction part stated, the thing that inspired us to start with this project is our common interest -- traveling. When we were discussing about the topic of the project, all of us agree using travelling as the main theme of the project by mere coincidence. Since we found that many people in Hong Kong like traveling over the world, and there is no a convenient way for them to get the information about other cities, therefore we decided to do

a blog which let people to share their own trips to the others and to get information from the others to plan their schedule for the future trips.

After we decided to use traveling as our topic, we referred to many different travel blogs on the internet, and conclude what functions should we put into our application. Therefore, the functions in the application are from our carefully selected.

When the application was finished, all of us are satisfy to our work. As we mentioned many times in this report, the main features of this application is sharing our trips and getting information for the future trips, we think our application can fulfill our ultimate goal.

Although the application is seem to be a finished product, there is a room for improvement. As the lesson mentioned, only 40% of the cost is used for preliminary work, there is 60% of the cost remains for maintainability. Despite the fact that we have no chance to publish the application and to keep doing for the maintainability part, we learned how to do the better preparation part so that we can be more easy to code and maintain the application.

8. References

- [1] Hartl, M. (2016). *Ruby on Rails Tutorial: Learn Web Development with Rails*.
- [2] CSCI3100-Group38 (2018). *Project Design Document*.