Project Title:

“Travelverse”

Submitted in the partial fulfillment of the requirements for the award of the Degree of Bachelor of Science in Software Engineering

Submitted By

Nazmus Saqueeb Ashrafi

192-35-2878

Department of Software Engineering

Under the support & guidance of

Ms. Afsana Begum

Assistant Professor

Department of Software Engineering

Department of Software Engineering

Daffodil International University

Fall - 2022

APPROVAL (Room- 603)

This project titled on “Travelverse”, submitted by Nazmus Saqueeb Ashrafi (ID: 192-35-2878) to the Department of Software Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Software Engineering and approval as to its style and contents.

BOARD OF EXAMINERS

Chairman

Dr. Imran Mahmud
Head and Associate Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Internal Examiner 1

Afsana Begum
Assistant Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Fazla Elahe

Internal Examiner 2

Dr. Md. Fazle Elahe
Assistant Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

External Examiner

Mohammad Abu Yousuf, PhD
Professor
Institute of Information Technology
Jahangirnogor University

DECLARATION

I hereby declare that I have done this project under the supervision of Ms. Afsana Begum, Assistant Professor, Department of Software Engineering, Daffodil International University. I also declare that this project or any part of this is unique and has not been submitted elsewhere for the award of any degree.

Nazmus Saqueeb Ashrafi
192-35-2878
Department of Software Engineering
Faculty of Science & Information Technology
Daffodil International University

Certified by:

Ms. Afsana Begum
Assistant Professor
Department of Software Engineering
Faculty of Science & Information Technology
Daffodil International University

ACKNOWLEDGMENT

I thank our creator Almighty God from deep of my heart and show much gratitude toward him for his heavenly blessing as we able to submit our final year project successfully.

I also show my heartiest gratitude and accept our utmost liability to our supervisor **Ms. Afsana Begum**, Assistant Professor, Department of Software Engineering, Daffodil International University, Dhaka. My supervisor's vast knowledge in the area of web application and her enthusiastic approach toward my project has deeply encouraged us to complete this project. Her strength, her forbearance her scholastic counsel her unparalleled leadership, her passion, her profound reflection on each rough sheet and her mistakes and helpfully correcting them every time has formed my attempts into a complete project. I want to share my appreciation and gratitude to Dr. Imran Mahmud, Associate Professor and Head In-Charge, Department of Software Engineering, for his gracious look towards my project which has made it possible to complete it properly. I also thank other faculty members and staff of the Software Engineering department of Daffodil International University for their kindness and helpful attitude towards us. I would like to thank all of my classmates at Daffodil International University, who took part in this discussion while completing the course work.

At last, I must acknowledge with due respect the constant support and patience of my parents.

ABSTRACT

Travelverse is a unique approach to social media applications that let users make detailed posts regarding the location they traveled to. Friends and family can be followed and users will be able to check out spots they visited as well. The application has a unique way of presenting locations visited on a map which will let users collect memories and share them with friends and family. This way of sharing locations will surely allow users to create memories and network those memories. The application has all features of a fully fleshed-out social media application including follows, likes, comments, live notifications, real-time chat, and even a functional shop portal that will allow travelers to buy items from registered shops.

TABLE OF CONTENTS

Chapter 1. Introduction / Background

1.1. Project Overview

The aim of this project is to make something which is useful to the general public and is not commonly found or integrated into the current industry. There are many social media projects in the market today and many of them are very popular and successful. Undertaking a social media project with well fleshed out features is a great challenge. This is why I wanted to undertake such a project. Social media websites allow you to make posts that can express your feelings or vision to your chosen audience, let them communicate with friends and family and much more. But how many social media websites allow you to build your post around the places that you visited. There is Google Maps which allows you to save locations you visited and review them but it does not allow you to share the places you visited with friends and family. It allows people to write reviews, but interactivity similar to Facebook is missing. I will combine the two ideas (of Google Maps and Facebook) and create Travelverse. I will also add location bases shops to sell as the e-commerce aspect of Travelverse. Ecommerce integration is not present in Google Maps and not strongly implemented in Facebook as it offers sales through only direct message. This feature will also increase the exclusivity of Travelverse.

1.2 Project Purpose

The purpose of this project is to produce a social media application dedicated to travelers in which users will be able to create posts using the detailed location of the spot they visit. The post creation will be detailed with title, description, latitude and longitude (which will be shown on an interactive map), multiple images, moods/emojis. Users will have a powerful way of making location-based posts and saving them as memories to be shared with friends and family in the network. Friends, family, and whoever is using the network will be able to like, comment, and share posts with real time notification delivery. They will be able to chat with other users in real time. Many other features like viewing all posts on a big map, viewing your follower's posts on the map will also be integrated. This will give the users an idea of where their friends are visiting. Shops will also be a part of the social network, where users can set up shops in different locations which are most visited and sell to people online or offline. This will be a very large application with many complicated, well-built out features at the user's disposal. It will also be one of a kind with modern and beautiful UI/UX design. It will be built using modern web development technology to make the application run smoothly. To summarize this project will:

- Let users create detailed memories of the place they visited with the exact location and pictures and feelings.
- Let users share that memory with friends, family and acquaintances who follow them and let them interact with those memories by like, comment, share.
- Robust system to manage followers/followings and friend suggestions.
- Give users access to a big map which shows all the places they visited and their friends visited.

- Live real time chatting and live notifications
- Shops with e-commerce possibility
- Modern UI/UX which will please the eyes.

1.2.1 Project Goals

This project has some important goals which will allow location-based posts to be recorded, displayed, and shared in a more organized and systemic fashion. Communication and e-commerce are also major goals.

The objectives are:

Objective 1 – Create memories and view them on the map

This application will act as a memory bank for travelers. Once a place is visited, the user can use the post creator to add a detailed memory to their timeline. Title, description and dropping a pin on the map is absolutely necessary in making a post in Travelverse. Over that you can add moods and multiple images to make your post stand out and be an all-inclusive memory recorded on your timeline. But the recording process does not end there. Each and every post in Travelverse has to have a location pin. This gives us the opportunity to collect all locations/posts in an all-inclusive map. This gives the user a bird’s eye view of all the places visited by the user and recorded on the timeline. Thus, with long term usage the Travelverse website becomes like a trip dairy for the user. Then we have Target Mode which will allow the user to further pin point and view each post in details with location.

Objective 2 – Share posts with followers in the network

The posts a user make is recorded in a bigger map but that’s not the end of it. Followers of the user can see the post and interact with it. They can like the post. Unlike the post once liked. They can comment on the post and share the post. This functionality is very necessary for any modern social media website, so it will be implemented solidly in Travelverse.

This application will use a system of followers. A user can follow any other users and they will become their follower. Once they become a follower their map will contain both their post and posts from the user they followed. This is why I will make a robust system to suggest friends, and maintain followers and followings. Map pins with different colors will represent own posts and post from their followings. This will ensure all different posts are accessible to a user of this application. This will build a solid network with multiple clusters of users sharing memories with each other.

Objective 3 – Real-time communication

Real time communication is extremely essential to all social networks. Without this feature a modern social media is incomplete. I will use WebSocket to ensure real-time chat possibility for all users of Travelverse. They will get instant messages and alerts when getting new messages from any users.

Objective 4 – E-commerce integration in social network

Shops will be an essential part of this application. Users will be able to create shops in this network with a shop registration number. Once a shop is created the user can set the location for the shop. This will ensure that the shop is discoverable by the travellers when they visit the location.

Shops will have items which can be created by the shop owner. Users can visit the shop and buy items from the shop by placing an order. This will make Travelverse a place to not only create rich travel memories but also a place to shop around.

Once put into cart and checkout done, a dashboard for the seller will display sales statistics and current orders with an option to fulfill order once the items are shipped. This will ensure revenue generation using Travelverse.

1.3. Stakeholders

Stakeholders are one of the most important components of a requirement documentation. Identification of the primary and secondary stakeholders at an early stage of the project is very beneficial because it gives a clear idea about what the project aims to do and which people the project will serve.

1.3.1 The traveler/ user

Traveler is a key component in this project because a lot of features are built to assist the traveler to create a meaningful post in the system and interact with the community. Travelers must be able to:

1) Sign up, sign in, forget password.

2) Create posts with multiple images and map location

3) Visualize posts in map

4) Customize their profile

5) Like, comment, share a post and get live notifications

6) Follow/Unfollow other users

7) Shop for items from different shops

8) Chat with other travelers

1.3.2 The shop/user

Shops will allow e-commerce to happen within the network. Shops will add additional value to Travelverse because users will be allowed to purchase items or preorder items before they visit a location.

Shops must be able to:

1) Sign up, sign in, forget password.

2) Set location for shop.

3) Add items for purchase

4) View sales and sales statistics

1.3.3 The admin

The admin has an overview of the project and database through an admin dashboard. They can delete users/ shops/ posts based on information about them. Shops in the database will have a registration number which will be shown in the dashboard. If it is not a valid registration number, the shop can be deleted by the admin.

1.3.4 Hands on user of the product

All users are hands on users. But each of them has unique privileges. For example, travelers will be having all social media features at their disposal and can also buy items. Shops will have ecommerce features accessible to them. Admin will have the power to delete multiple components from the database to control the overall application.

1.4 Methodology

I have opted to use the iterative waterfall model to design this software. The feedback loop present in the iterative waterfall model was crucial to making iterative improvements to the application. In this project, the requirements did not change heavily so the iterative model was a perfect fit. I wanted to develop a minimal viable product and make improvements as necessary and therefore iterative model was a good choice in my opinion.

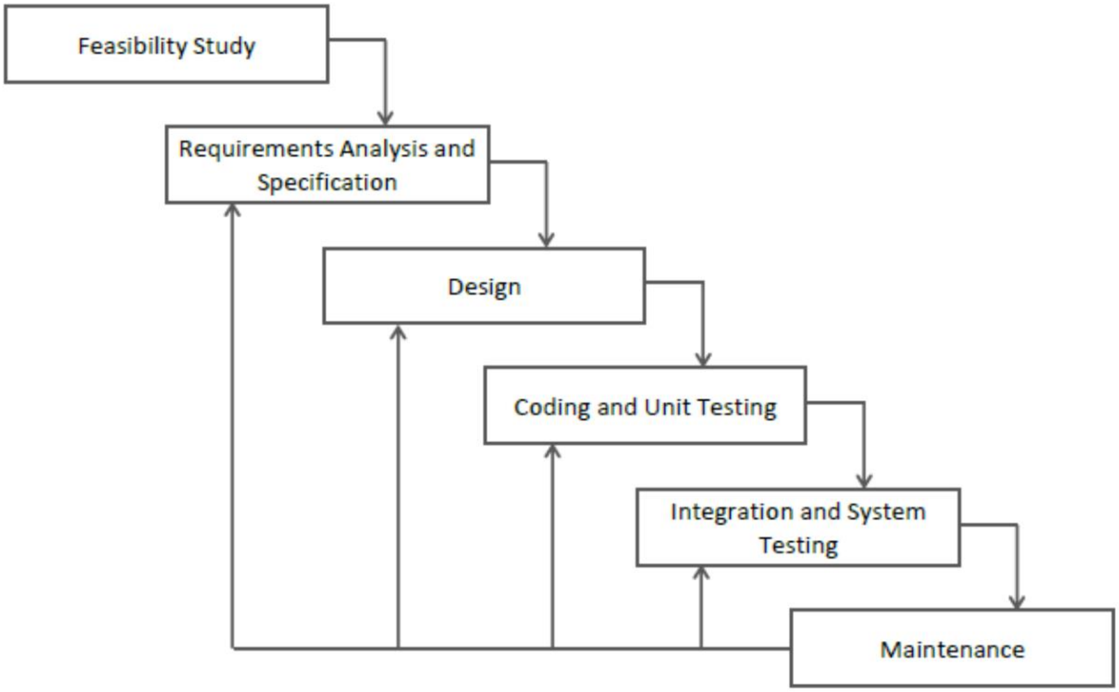

Feasibility Study
Requirements Analysis and Specification
Design
Coding and Unit Testing
Integration and System Testing
Maintenance

Figure 1.4: Iterative Waterfall Model

1.5 Proposed System Model

A simple model to visualize the system architecture is necessary for development. I have simplified the complexities of Travelverse and represented it in this simple design model.

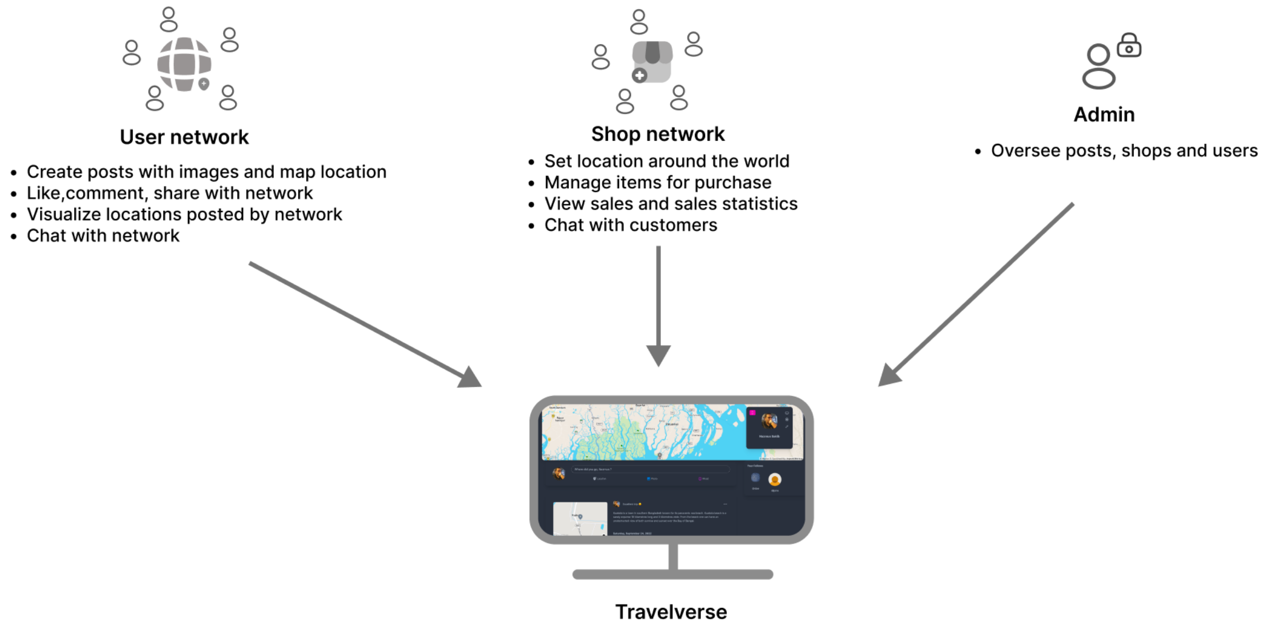
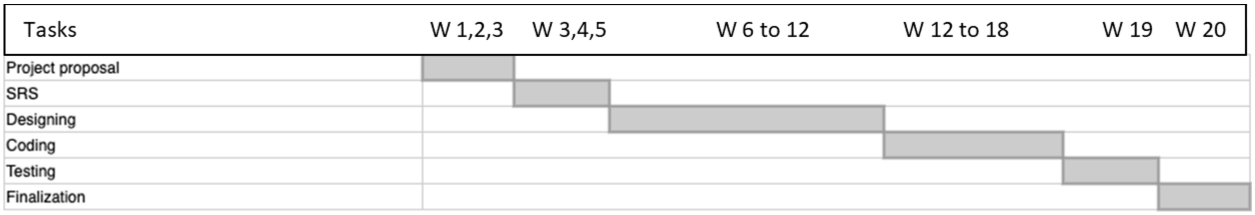

User network
• Create posts with images and map location
• Like,comment, share with network
• Visualize locations posted by network
• Chat with network
Shop network
• Set location around the world
• Manage items for purchase
• View sales and sales statistics
• Chat with customers
Admin
• Oversee posts, shops and users
Travelverse

Figure 1.5: Proposed System Model

1.6 Project schedule

1.6.1 Timescales

See Figure 1.6.1

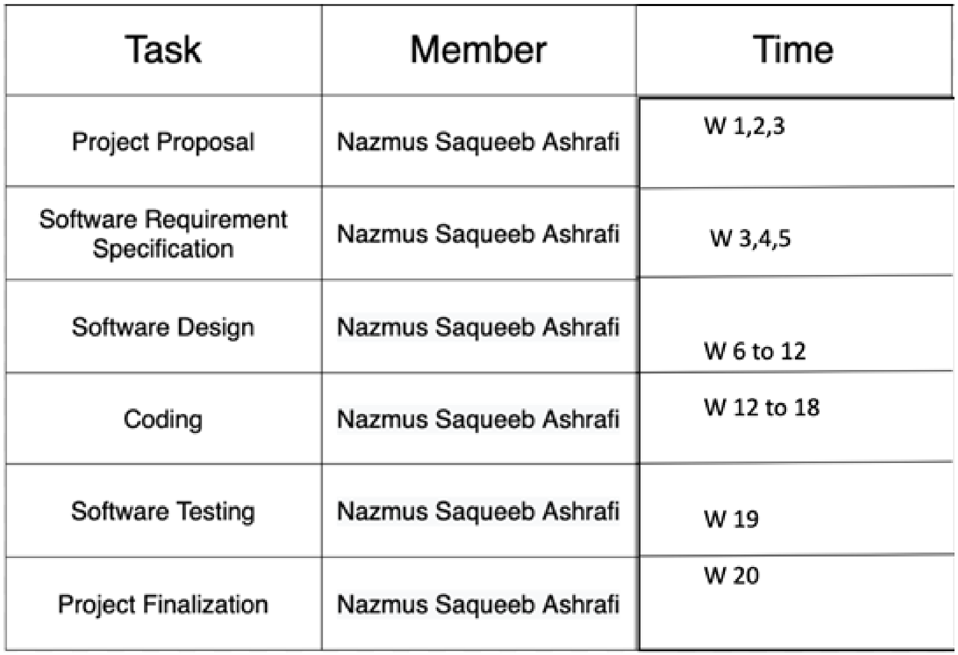


Tasks	W 1,2,3	W 3,4,5	W 6 to 12	W 12 to 18	W 19	W 20
Project proposal	▨					
SRS		▨				
Designing			▨			
Coding				▨		
Testing					▨	
Finalization						▨

Figure 1.6.1- Time frame for project implementation

1.6.2 Work distribution

See Figure 1.6.2



Task	Member	Time
Project Proposal	Nazmus Saqueeb Ashrafi	W 1,2,3
Software Requirement Specification	Nazmus Saqueeb Ashrafi	W 3,4,5
Software Design	Nazmus Saqueeb Ashrafi	W 6 to 12
Coding	Nazmus Saqueeb Ashrafi	W 12 to 18
Software Testing	Nazmus Saqueeb Ashrafi	W 19
Project Finalization	Nazmus Saqueeb Ashrafi	W 20

Figure 1.6.2 Work distribution

1.7 Project constraints

1.7.1 Memory constraint

This project aims to collect a huge amount of information from its users and organize them. It expects huge networks to be formed and lots of interactions between users. This requires a reliable and scalable database. Large scale databases are expensive. Thus, this is a big constraint for this application.

1.8 Project deliverables

1.8.1 User documentation

Some documents can be will be provided in relation to this project:

1) Printed materials: SRS documentation completed with use case descriptions.

2) Projected material: Power point presentations to present information about the software.

1.8.2 Full code

Entire codebase will be provided in a GitHub link with all detailed timeline of all updates.

https://github.com/nazmus-saqueeb-ashrafi/travelVerse-combined

Chapter 2. Requirements Engineering

2.1 Requirement elicitation

Two ways in which I collected requirement for this project are

1) From current systems – In current systems which are in place, there is a way to write reviews online for people based on location. But there is no way we can store location memories and share them with friends and family. Also, there is no solid implementation of shops in any social media based on locations. Studying current systems, I found a gap and thus I gathered requirements to fill this gap.

2) From an imaginary scenario - I created a scenario where the people are interested in a social media exclusive to travelers with shop features. I gathered the requirements that are needed to fulfill this gap and used them as requirements to make this project.

2.2. Functional requirements

2.2.1 Scope

The scope of this project is targeted towards 2 major points of view. The traveler and the shop. Project aims to serve both users with different features

The traveler can post, and do many other this related to post, like viewing the post in bigger maps, like, comment, share etc. They will follow other travelers, will live chat, get live notifications and also shop from suitable locations.

The shop can set up at a popular destination, set up products for sale and sell them to travelers.

2.2.2 Functional Requirements

The functional requirements are written in laymen's terms so that everybody without a technical background can read and understand the features and what they mean for this application. It is necessary that we follow through and fulfil all these requirements as it is essential for the completion of this application.

FR-01	Registration
Description	This feature allows travelers or shops to register as a user using their information. Registration will require information from users. Once registered the user will be able to login.
Stake holder	Traveller, Shop

FR-02	Login
Description	A traveler, shop can login to the application after registrations

Stake holder	Traveller, Shop, Admin

FR-03	Forgot password
Description	Reset password through email link if password is forgotten
Stake holder	Traveller, Shop

FR-04	User profile
Description	Traveller, Shop can update their name, email and display picture to customize their profile
Stake holder	Traveller, Shop

FR-05	Follower suggestion
Description	To create a good network among users, the system will recommend people to follow to the user who is logged in
Stake holder	Traveller, Shop

FR-06	Search for other travellers/shop
Description	Users will be able to search for each other using a searching system which lets them search the entire network.
Stake holder	Traveller, Shop

FR-07	Follow/Unfollow
Description	Users will be able to follow/ unfollow each other. Depending on who they follow, post pins will show in a user's map.
Stake holder	Traveller, Shop

FR-08	Realtime notifications
Description	Users will receive real time notifications when their post is liked, commented on or shared. They will be also able to dismiss notifications after checking.
Stake holder	Traveller, Shop

FR-09	Post

Description	Create, edit and delete posts. Posts should be able to be made using multiple images, mood emojis and location details.
Stake holder	Traveller

FR-10	Like/unlike
Description	Like and unlike post.
Stake holder	Traveller

FR-11	Comment
Description	Comment on post and be able to delete one's own comment
Stake holder	Traveller

FR-12	Share posts
Description	Share a post others made.
Stake holder	Traveller

FR-13	Messenger
Description	Show online users and being able to communicate with them in real time. Also get alerted when you receive a brand new chat.
Stake holder	Traveller, Shop

FR-14	Themes
Description	Customize your UI with the color palette you prefer
Stake holder	Traveller, Shop

FR-15	Delete account
Description	Delete account once you do not need the account any more
Stake holder	Traveller, Shop

FR-16	Target post
Description	Show all timeline posts in a big map along with related posts to create a dairy of memories in one place which is conveniently accessible.
Stake holder	Traveller

FR-17	Logout
Description	Logout after your session
Stake holder	Traveller, Shop, Admin

FR-18	Edit shop location
Description	Edit location of the shop to customize the shop
Stake holder	Shop

FR-19	Product
Description	Add products in shop for travelers to order online using item details, edit or delete products
Stake holder	Shop

FR-20	Place order

Description	View items in a shop, add to cart and place order. See all the orders placed by the user in the orders page.
Stake holder	Travelers

FR-21	Shop dashboard
Description	Be able to see sales statistics on how many orders were received versus how many orders fulfilled. Fulfill orders from the dashboard after shipping them.
Stake holder	Shop

FR-22	Total users/shops
Description	See all users in the system including shops. Get a brief idea about them and be able to delete them if need be.
Stake holder	Admin

FR-23	Total posts
Description	See all posts in the system. Get a brief idea about them and be able to delete them if need be.

Stake holder	Admin

2.3. Non-functional requirements

2.3.1 Performance Requirements

These are the non-functional dependencies which are crucial to running the program smoothly and efficiently.

2.3.2 Speed and latency requirement

Speed is needed when the users operate the application.

PR-01	Responsive system

Description	When user is using features in the application, it should come up fast.
Stake holder	Traveller, Shop

2.3.3 Precision and accuracy requirement

When an organizer makes a vaccination slot it should have an accurate date/time picker.

PR-02	Accurate post, item maker
Description	Accuracy is needed when making a post or item
Stake holder	Traveller, Shop

2.3.4 Capacity requirement

System should scale up to user’s needs.

PR-03	Initially we can have 10,000 users
Description	Using MongoDb, aim to scale up to 10,000 initially
Stake holder	Traveller, Shop

2.3.5 Dependability requirement

My system must be reliable and robust. This is very important for us to ensure because without a dependable system it is difficult to run the operations of a software smoothly. If the system is not running smoothly then there will not be any users who will be interested in using our system long term.

2.3.6 Reliability and availability

It is important for our system to be reliably. We should ensure that our system can handle downtimes and not of the data is lost due to incomplete data transactions. System should be available all the time and ensure database rules are followed.

DR-01	System must be up 24x7
Description	System should be up all the time, should be updated whenever we get a chance. System must keep user's data up to date.
Stake holder	Traveller, Shop

2.3.7 Robustness and fault tolerance requirements

System will ensure that there is no crash/minimum crash when users are using our system. Also, system can facilitate multiple users logged in.

DR-02	System allows multiple access and can recover from errors without crashing
Description	When multiple users are logged on in the system, the system does not crash and can handle the situation smoothly.
Stake holder	N/A

2.3.8 Safety critical requirements

No specific safety critical requirements needed.

2.3.9 Maintainability and supportability

Supportability is the level to which our system is designed to accommodate change easily. A system is easily maintainable if it allows our support team to operate on our software easily to solve various issues.

2.3.9.1 Maintenance Requirements

MS-01	System allows smooth updating and confirmation.
Description	When anyone tries to update information of a user our system allows it to smoothly happen.

Stake holder	Traveller, Shop

2.3.10 Supportability Requirements

Supportability is very important necessity for our system. Technical support should be able to fix any issues that arise with the system at any moment. The technical support team can face various difficulties like:

- Malfunctioning system where the system operator has to find out exactly where the problem is.
- Wrong output by the system and developers need to check the data flow which is giving the wrong output.
- Out system is hacked and we need to restore security of our system.

2.3.11 Adaptability Requirements

No specific adaptability requirements needed.

2.3.12 Security requirements

The necessary security requirements are given below:

- Software must validate that each of the users that are logging in are already a registered user. If they are not registered, they should be denied access.
- Input size of username and password must not exceed size already specified for input
- Our server should only store authorized user's data

- Encrypt sensitive information so that hackers do not get easy access.

2.3.13 Access requirements

Software must be accessed in a secure way. All the different modules of the software must follow a security protocol to access the different parts of the software.

SR-01	The system has security protocols in place
Description	All modules should go through a fixed security protocol to access part of the system.
Stake holder	Traveller, Shop, Admin

2.3.14 Integrity requirements

Encryption should be done to all passwords and relevant user details so that it is not easy for the hackers to hack our system and steal all sensitive information.

2.3.15 Privacy requirements

We should use a well-protected and secure server to host our information to ensure privacy for anyone who uses our system.

SR-02	Ensure data protection
Description	Use a secure and private server to host our data to ensure privacy of users is maintained.
Stake holder	Traveller, Shop, Admin

2.3.16 Usability and human integrity requirements

These requirements make sure that our system is easy to use by the users and they are cognitively able to use the software.

2.3.17 Ease of use requirements

System must be easy to use.

UH-01	Must be easy to use by all stakeholders
Description	All possible routes the user can take while using this software must be easy and convenient for all users.
Stake holder	Traveller, Shop, Admin

2.3.18 Understandability and politeness requirement

Software must use simple language so every stand can understand the functionalities.

UH-02	Easy to read
Description	System must be understandable by all users of the software
Stake holder	Traveller, Shop, Admin

2.3.19 Accessibility requirement

Below are the accessibility requirements of the software.

AR1: Login as Traveller

AR2: Logout as Traveller

AR3: Forgot password as Traveller

AR4: Login as Shop

AR5: Logout as Shop

AR6: Forgot password as Shop

AR7: Login as Admin

AR8: Logout as Admin

AR9: Forgot password as Admin

There should be a way user can register for an account with information. This must be done using a central authentication mechanism. Security must be provided for these accounts too.

2.3.20 User documentation

UH-03	Developer documentation
Description	System must be well documented for developer use.
Stake holder	Developer

2.3.21 Look and feel requirements

The intention of the design and how to interpret it is described in this section.

2.3.21.1 Appearance requirements

System should clearly show to the user which field is necessary and which field should be left blank. It should also show number of characters to be inputted if there is a limit. There should also be appropriate errors shown by our system in a nice format.

LF-01	Mandatory field indicators
Description	Labels for mandatory field should be clearly indicated
Stake holder	Traveller, Shop, Admin

LF-02	Error boxes
Description	Show errors in a nice format
Stake holder	Traveller, Shop, Admin

2.3.21.2 Style requirements

Modern UI/UX design is a must for this application

2.3.21.3 Operational and environmental requirements

These requirements focus on how our system will interact with the outside world and how it will behave in what circumstances.

2.3.21.4 Expected physical requirements

No expected physical requirements.

2.3.21.5 Requirements for interfacing with adjacent systems

No requirements for interfacing with adjacent systems.

2.3.21.6 Release requirements

No specific release requirements.

2.3.21.7 Legal requirements

These requirements set the rules for our system and what to do if any term is violated.

2.3.21.8 Compliance requirements

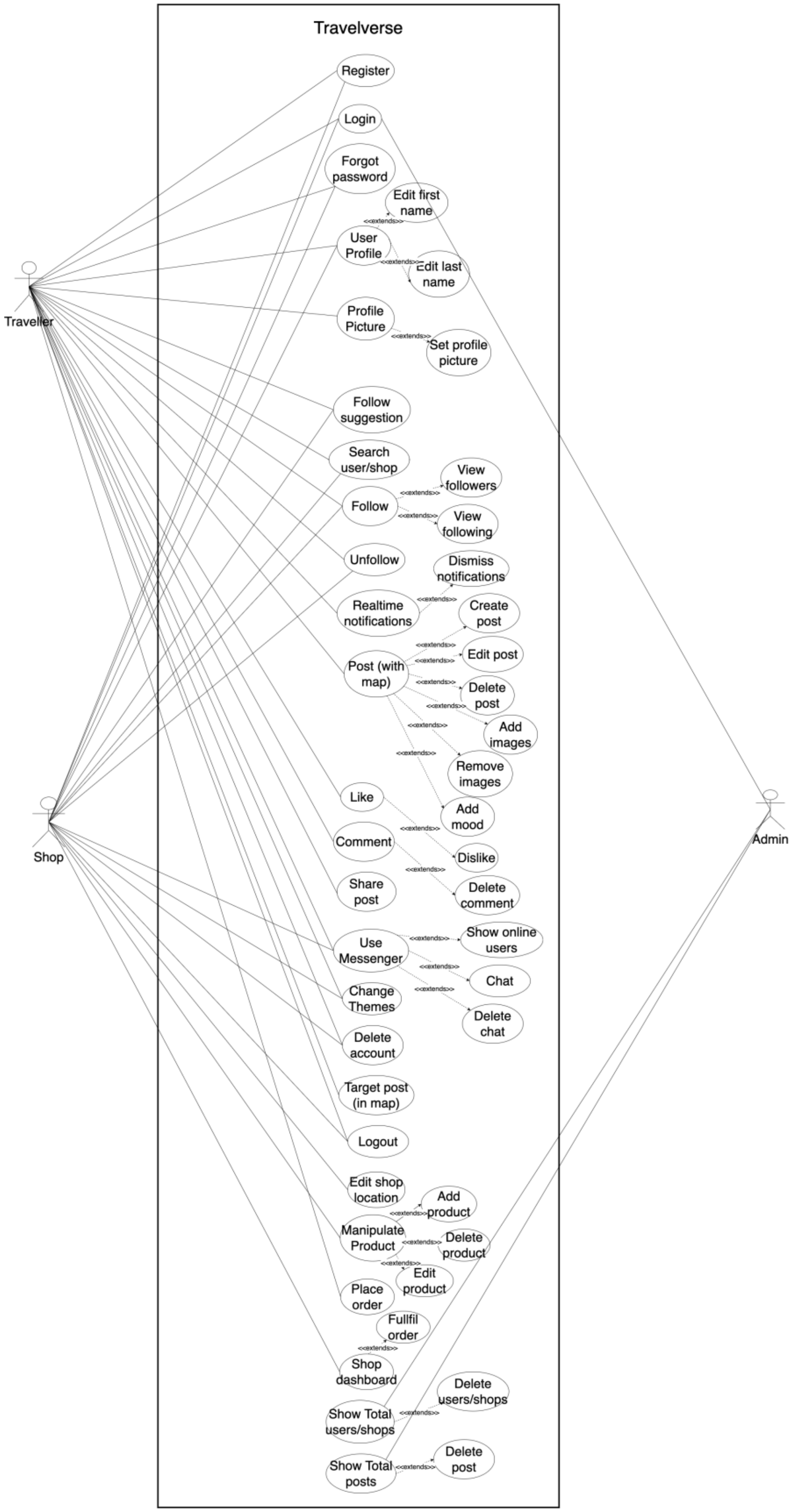
No specific compliance requirements.

2.3.21.9 Standard requirements

No specific standard requirements.

Chapter 3. System analysis and design specification

3.1 Use Case Diagram


Travelverse
Traveller
Shop
Admin
Register
Login
Forgot password
Edit first name
<<extends>>
User Profile
Edit last name
Profile Picture
Set profile picture
Follow suggestion
Search user/shop
View followers
Follow
View following
Unfollow
Dismiss notifications
Realtime notifications
Create post
Edit post
Post (with map)
Delete post
Add images
Remove images
Add mood
Like
Dislike
Comment
Delete comment
Share post
Use Messenger
Show online users
Chat
Delete chat
Change Themes
Delete account
Target post (in map)
Logout
Edit shop location
Manipulate Product
Add product
Delete product
Edit product
Place order
Fullfil order
Shop dashboard
Show Total users/shops
Delete users/shops
Show Total posts
Delete post

3.2 Use Case Descriptions

Use Case	Register	
Goal	User information is stored in the system and an account is made using the information.	
Preconditions	The information needs to be already filled in a form by the user.	
Success end condition	The filled information is saved in our system and the related used has an account in our system	
Failed end condition	The information is not saved in our system.	
Primary actors: **Secondary Actors:**	Traveller, Shop	
Trigger	User wants to make an account	
Description / Main success scenario	**Step:**	Action

	1.	User fills information in fields
	2.	User clicks create account
	3.	All user details are saved and account is successfully created
Alternatives Flows	**Step:**	Branching Action
	2a.	User already exists under that information
	2a1.	Do not let user make duplicate account
Quality Requirements	**Step:**	Requirements
	1.	Must ensure valid information is inputted.
	2.	Password should be hashed

Use Case	Sign in
Goal	The user can login if the user's information is stored in the system.
Preconditions	User should already be registered in the system.

Success end condition	User successfully logged in our system	
Failed end condition	Registered user could not be logged in our system	
Primary actors: **Secondary Actors:**	Traveller, Shop	
Trigger	User wants to login into our system	
Description / Main success scenario	**Step:**	Action
	1.	User enters the username and password
	2.	System matches username and password using authorization (see use case description – authorization)
	3.	User is taken to the appropriate page based of the kind of user.
Alternatives Flows	**Step:**	Branching Action

	2a.	User name and password does not match
	2a1.	User is bought back to the login page
Quality Requirements	**Step:**	Requirements
	1.	Ensure security

Use Case	Forgot password
Goal	Reset password using an email link if user forgot password
Preconditions	User should already be registered in the system with an email.
Success end condition	User successfully reset their password
Failed end condition	User failed to reset password
Primary actors: **Secondary Actors:**	Traveller, Shop
Trigger	User want to reset password

Description / Main success scenario	**Step:**	Action
	1.	User clicks reset password
	2.	User enters email
	3.	Email sends to user
	4.	User follows link and resets password
Alternatives Flows	**Step:**	Branching Action
	3a.	User enters wrong email
	2a1.	Error shown “Not registered”
Quality Requirements	**Step:**	Requirements
	1.	Email should be in a format

Use Case	User profile
Goal	Traveller, Shop can update their name, email and display picture to customize their profile
Preconditions	User should have a profile registered

Success end condition	User can edit desired information	
Failed end condition	User failed to edit desired information	
Primary actors: **Secondary Actors:**	Traveller, Shop	
Trigger	User clicked edit profile	
Description / Main success scenario	**Step:**	Action
	1.	User clicks edit profile
	2.	User chooses field or image to edit
	3.	Information/photo updates
Alternatives Flows	**Step:**	Branching Action
	2a.	User leaves field empty
	2a1.	Do not allow to edit
Quality Requirements	**Step:**	Requirements
	1.	Process should be smooth

Use Case	Follower suggestion	
Goal	To create a good network among users, the system will recommend people to follow to the user who is logged in	
Preconditions	User should have a profile registered	
Success end condition	Show all users who are not in user's followers list	
Failed end condition	Failed to show relevant users	
Primary actors: **Secondary Actors:**	Traveller, Shop	
Trigger	User wants to discover who to follow	
Description / Main success scenario	**Step:**	Action
	1.	Info is shown in home page
Alternatives Flows	**Step:**	Branching Action
Quality Requirements	**Step:**	Requirements
	1.	Must show valid information

Use Case	Search for other travelers/shop	
Goal	Users will be able to search for each other using a searching system which lets them search the entire network.	
Preconditions	User should have a profile registered	
Success end condition	User can search people and shops using name	
Failed end condition	Search does not show result	
Primary actors: **Secondary Actors:**	Traveller, Shop	
Trigger	User enters information in search field	
Description / Main success scenario	**Step:**	Action
	1.	User types name in search field
	2.	User profile shows up
Alternatives Flows	**Step:**	Branching Action
	2a.	No user

	2a1.	No information shown
Quality Requirements	**Step:**	Requirements

Use Case	Follow/Unfollow	
Goal	Users will be able to follow/ unfollow each other and shops. Depending on who they follow, post pins will show in a user's map.	
Preconditions	User should have a profile registered	
Success end condition	User successfully followed/unfollowed another user/shop	
Failed end condition	Unsuccessful follow/unfollow	
Primary actors: **Secondary Actors:**	Traveller, Shop	
Trigger	User clicks follow/unfollow	
Description / Main success scenario	**Step:**	Action
	1.	User click follow/unfollow

	2.	User/ shop followed or unfollowed
Alternatives Flows	**Step:**	Branching Action
Quality Requirements	**Step:**	Requirements

Use Case	Realtime notifications
Goal	Users will receive real time notifications when their post is liked, commented on or shared. They will be also able to dismiss notifications after checking.
Preconditions	User should have a profile registered
Success end condition	User receives notification immediately
Failed end condition	User does not receive any notifications
Primary actors: **Secondary Actors:**	Traveller, Shop

Trigger	Automated	
Description / Main success scenario	**Step:**	Action
	1.	User's post is liked, commented or shared by another user
	2.	User receives a notification stating this
Alternatives Flows	**Step:**	Branching Action
Quality Requirements	**Step:**	Requirements
	1.	Should be a live event

Use Case	Post
Goal	Create, edit and delete posts. Posts should be able to be made using multiple images, mood emojis and location details.
Preconditions	User should have a profile registered
Success end condition	User can create, manipulate or delete post
Failed end condition	Actions not possible

Primary actors: **Secondary Actors:**	Traveller	
Trigger	User clicks create, edit, delete post	
Description / Main success scenario	**Step:**	Action
	1.	User clicks action
	2.	Action related modal shows and asks for confirmation
	3.	Post is created, edited, deleted
Alternatives Flows	**Step:**	Branching Action
Quality Requirements	**Step:**	Requirements

Use Case	Like/unlike
Goal	User should be able to like and unlike post.

Preconditions	User should have a profile registered and they should have made a post.	
Success end condition	Post can be liked or unliked.	
Failed end condition	Post cannot be liked or unliked.	
Primary actors: **Secondary Actors:**	Traveller	
Trigger	User wants to like/unlike post	
Description / Main success scenario	**Step:**	Action
	1.	User clicks like/unlike post
	2.	Post is liked or unliked
	3.	Post maker user get a notification stating a user (with name) liked/unlike their post
Alternatives Flows	**Step:**	Branching Action
Quality Requirements	**Step:**	Requirements

1.	Notification should be a live event

Use Case	Comment	
Goal	Comment on post and be able to delete one’s own comment	
Preconditions	User should have a profile registered and they should have made a post.	
Success end condition	Post can be commented on and the comment can be deleted	
Failed end condition	Post cannot be commented on	
Primary actors: **Secondary Actors:**	Traveller	
Trigger	User wants to comment on post or delete a made comment	
Description / Main success scenario	**Step:**	Action
	1.	User makes a comment

	2.	Post is commented on
	3.	Post maker user get a notification stating a user (with name) commented on the post
Alternatives Flows	**Step:**	Branching Action
Quality Requirements	**Step:**	Requirements
	1.	Notification should be a live event

Use Case	Share posts
Goal	Share a post other have made
Preconditions	User should have a profile registered
Success end condition	Post can be shared
Failed end condition	Post cannot be shared
Primary actors:	Traveller

Secondary Actors:		
Trigger	User wants to share a post	
Description / Main success scenario	**Step:**	Action
	1.	User clicks on share
	2.	Post is shared
	3.	Owner of the post which is shared receives a notification
Alternatives Flows	**Step:**	Branching Action
Quality Requirements	**Step:**	Requirements
	1.	Notification should be a live event

Use Case	Messenger
Goal	Show online users and being able to communicate with them in real time. Also get alerted when you receive a brand new chat.

Preconditions	User should have a registered profile	
Success end condition	Users can chat with each other in real time	
Failed end condition	Live chat not possible	
Primary actors: **Secondary Actors:**	Traveller, Shop	
Trigger	Users want to chat	
Description / Main success scenario	**Step:**	Action
	1.	System shows online users
	2.	User clicks on an online user
	3.	User chats
Alternatives Flows	**Step:**	Branching Action
Quality Requirements	**Step:**	Requirements
	1.	Events should happen in real time

Use Case	Theme	
Goal	Customize your UI with the color palette you prefer	
Preconditions	User must be logged in	
Success end condition	User changes look of the website	
Failed end condition	User cannot change look of the website	
Primary actors: **Secondary Actors:**	Traveller, Shop	
Trigger	User browses and chooses theme	
Description / Main success scenario	**Step:**	Action
	1.	Click on themes
	2.	Browse themes
	3.	Choose theme
Alternatives Flows	**Step:**	Branching Action

Quality Requirements	**Step:**	Requirements

Use Case	Delete account	
Goal	Delete account once you do not need the account any more	
Preconditions	User should have an account and be logged in	
Success end condition	Account deleted	
Failed end condition	Account not deleted	
Primary actors: **Secondary Actors:**	Traveller, Shop	
Trigger	Click on delete account	
Description / Main success scenario	**Step:**	Action
	1.	Click on delete account
	2.	Modal asks for assurance
	3.	Account deleted

Alternatives Flows	**Step:**	Branching Action
	2a.	User clicks cancel
	2a1.	Account not deleted
Quality Requirements	**Step:**	Requirements

Use Case	Target post
Goal	Show all timeline posts in a big map along with related posts to create a dairy of memories in one place which is conveniently accessible.
Preconditions	User should have an account and be logged in
Success end condition	Show user’s timeline posts in a big map
Failed end condition	All timeline posts not showing in map
Primary actors: **Secondary Actors:**	Traveller

Trigger	User clicks the target button	
Description / Main success scenario	**Step:**	Action
	1.	User clicks target button
	2.	All timeline posts shown in a big map
Alternatives Flows	**Step:**	Branching Action
Quality Requirements	**Step:**	Requirements

Use Case	Logout
Goal	Logout after user finishes a session
Preconditions	User must be logged in
Success end condition	User logged out
Failed end condition	User not logged out
Primary actors:	Traveller, Shop

Secondary Actors:		
Trigger	User clicks the logout button	
Description / Main success scenario	**Step:**	Action
	1.	User clicks the logout button
	2.	User is logged out
Alternatives Flows	**Step:**	Branching Action
Quality Requirements	**Step:**	Requirements

Use Case	Edit shop location
Goal	Edit location of the shop to customize the shop
Preconditions	User should be registered as a shop
Success end condition	Shop location changed
Failed end condition	Shop location not changed

Primary actors: **Secondary Actors:**	Shop	
Trigger	User clicks the shop location button	
Description / Main success scenario	**Step:**	Action
	1.	User clicks shop location button
	2.	User chooses a location on the map
	3.	Location of the shop is set
Alternatives Flows	**Step:**	Branching Action
Quality Requirements	**Step:**	Requirements

Use Case	Product
Goal	Add products in shop for travelers to order online using item details, edit product, delete product

Preconditions	The user should be logged in as a shop	
Success end condition	A product is successfully created in the shop, can be edited or deleted	
Failed end condition	Product failed to be created under that shop, edit or delete	
Primary actors: **Secondary Actors:**	Shop	
Trigger	User wants to create a product	
Description / Main success scenario	**Step:**	Action
	1.	User logs in as shop
	2.	User click create/ edit/delete product button
	3.	User fills in product info and submits to creates a product, user get a form to edit, product is deleted
Alternatives Flows	**Step:**	Branching Action

	3a.	Product data in invalid
	3b.	User clicks cancel
	3.a.1	Show errors where data in invalid
	3.b.1	User is taken back to the main page
Quality Requirements	**Step:**	Requirements

Use Case	Place order
Goal	View items in a shop, add to cart and place order. See all the orders placed by the user in the orders page.
Preconditions	User should log in as a traveler
Success end condition	User places an order from a shop
Failed end condition	Order placement failed
Primary actors:	Traveller

Secondary Actors:		
Trigger	User want to shop	
Description / Main success scenario	**Step:**	Action
	1.	User selects products and adds to cart
	2.	User orders products
	3.	Order updates in dashboard
	4.	Shop can fulfill the order
Alternatives Flows	**Step:**	Branching Action
Quality Requirements	**Step:**	Requirements

Use Case	Shop dashboard
Goal	Be able to see sales statistics on how many orders were received versus how many orders fulfilled. Fulfill orders from the dashboard after shipping them.

Preconditions	Seller logs in	
Success end condition	The seller can see stats and also fulfill orders placed by users	
Failed end condition	Seller cannot see stats or fulfill orders	
Primary actors: **Secondary Actors:**	Shop	
Trigger	Sell is on the dashboard	
Description / Main success scenario	**Step:**	Action
	1.	Seller logs in
	2.	Seller can see stats and current orders
	3.	Seller can fulfill current orders
Alternatives Flows	**Step:**	Branching Action
Quality Requirements	**Step:**	Requirements

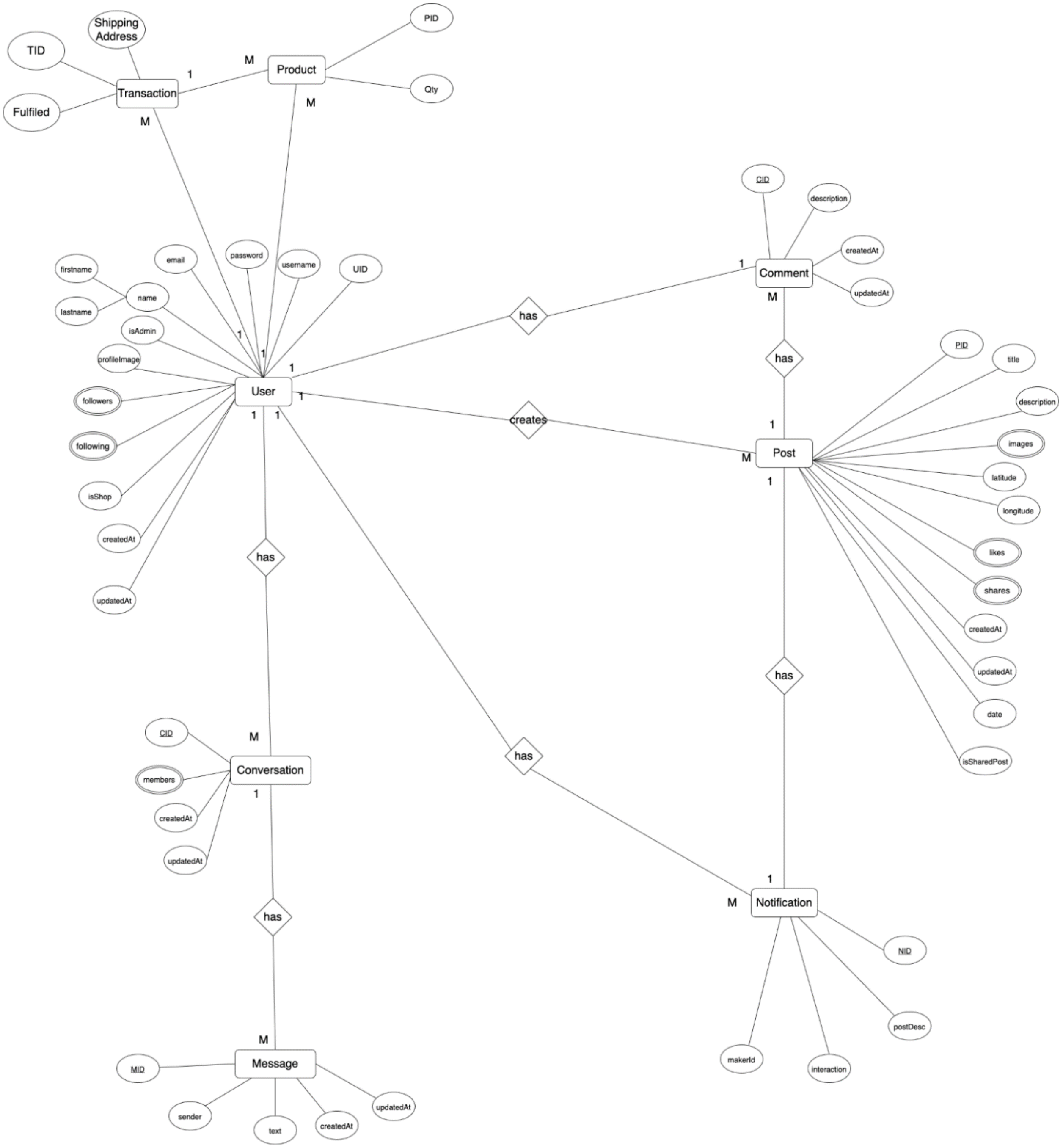
Use Case	Total users/shops	
Goal	See all users in the system including shops (with registration number). Get a brief idea about them and be able to delete them if need be.	
Preconditions	Must be logged in as admin	
Success end condition	Admin can manipulate the users/shops in the database	
Failed end condition	Admin cannot manipulate the users/shops in the database	
Primary actors: **Secondary Actors:**		
Trigger	Admin logs in and wants to manipulate database	
Description / Main success scenario	**Step:**	Action

	1.	Admin logs in
	2.	Admin can see travellers, shops (with registration number)
	3.	Admin can delete inappropriate users or content
Alternatives Flows	**Step:**	Branching Action
Quality Requirements	**Step:**	Requirements

Use Case	Total posts
Goal	See all posts in the system. Get a brief idea about them and be able to delete them if need be.
Preconditions	Must be logged in as admin
Success end condition	Admin can manipulate the posts in the database

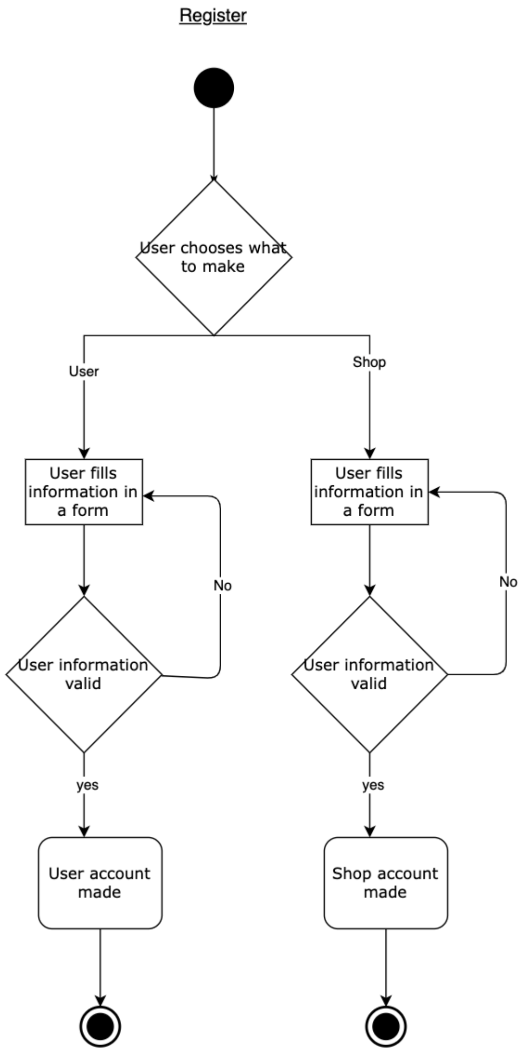
Failed end condition	Admin cannot manipulate the posts in the database	
Primary actors: **Secondary Actors:**		
Trigger	Admin logs in and wants to manipulate database	
Description / Main success scenario	**Step:**	Action
	1.	Admin logs in
	2.	Admin can see posts with post likes and comment count
	3.	Admin can delete inappropriate posts
Alternatives Flows	**Step:**	Branching Action
Quality Requirements	**Step:**	Requirements

3.3 ERD

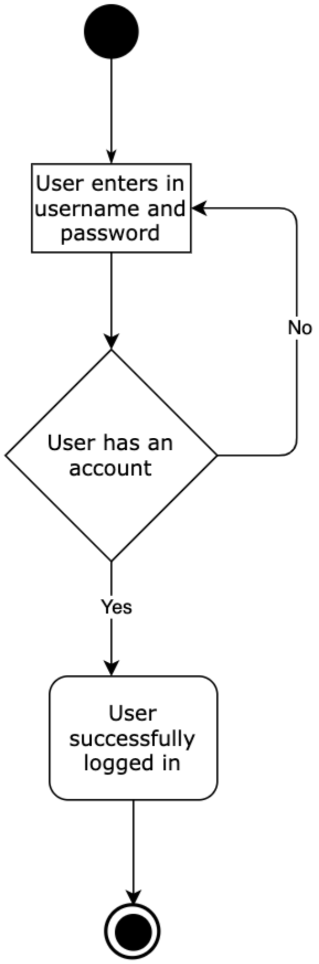


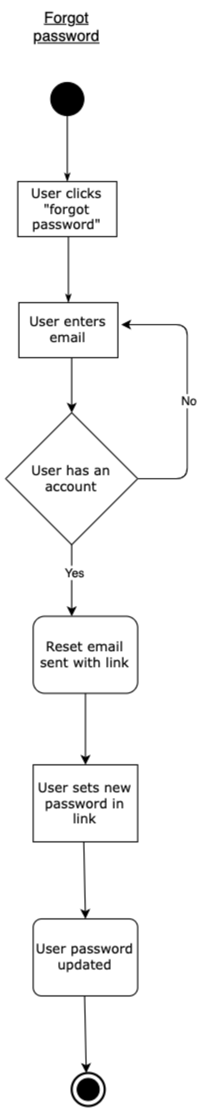
TID
Shipping Address
Fulfiled
Transaction
1
M
M
Product
M
PID
Qty
CID
description
createdAt
updatedAt
1
Comment
M
has
has
firstname
lastname
name
email
password
username
UID
isAdmin
profileImage
followers
following
isShop
createdAt
updatedAt
1
1
1
1
User
1
1
creates
1
M
Post
1
PID
title
description
images
latitude
longitude
likes
shares
createdAt
updatedAt
date
isSharedPost
has
has
has
M
CID
members
createdAt
updatedAt
Conversation
1
has
M
Message
MID
sender
text
createdAt
updatedAt
1
M
Notification
NID
postDesc
interaction
makerId

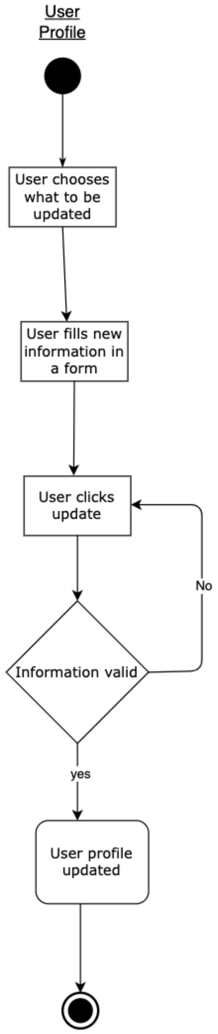
3.4 Activity diagrams

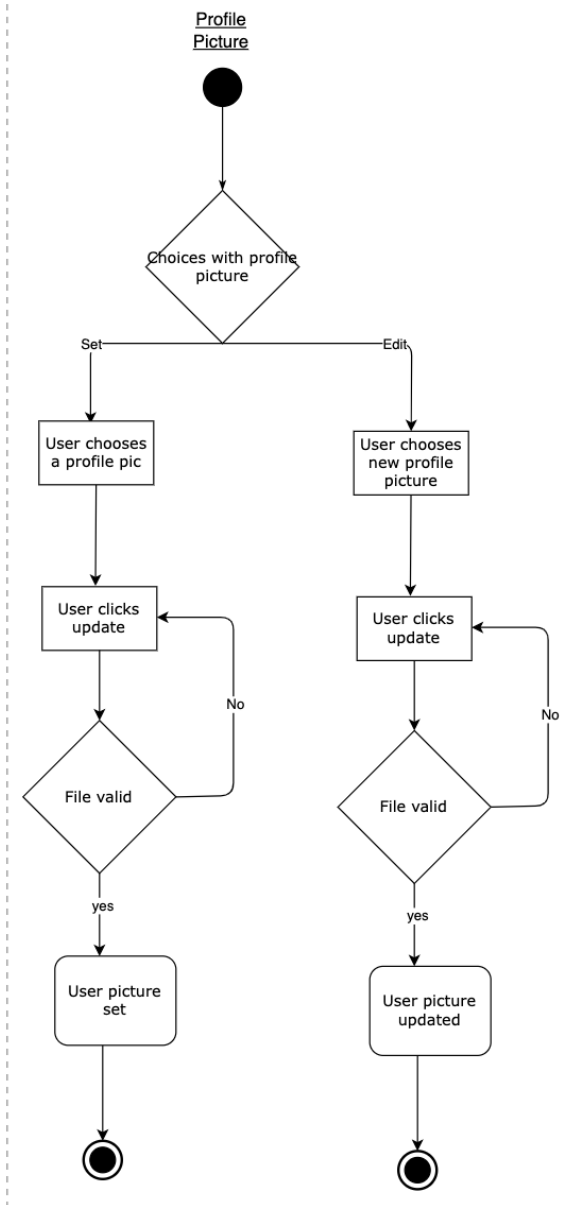


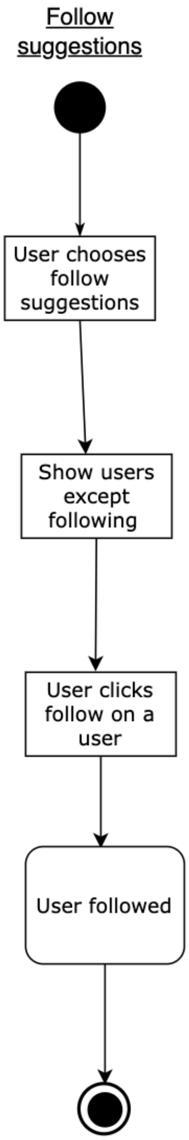
Login

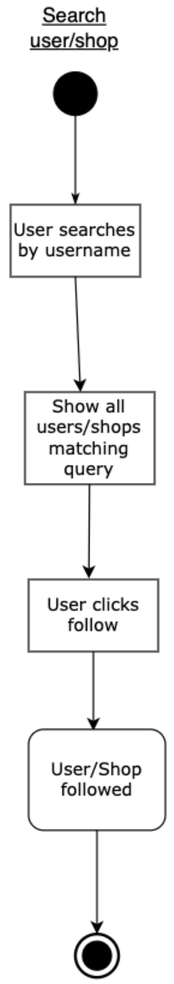

User enters in username and password
User has an account
No
Yes
User successfully logged in

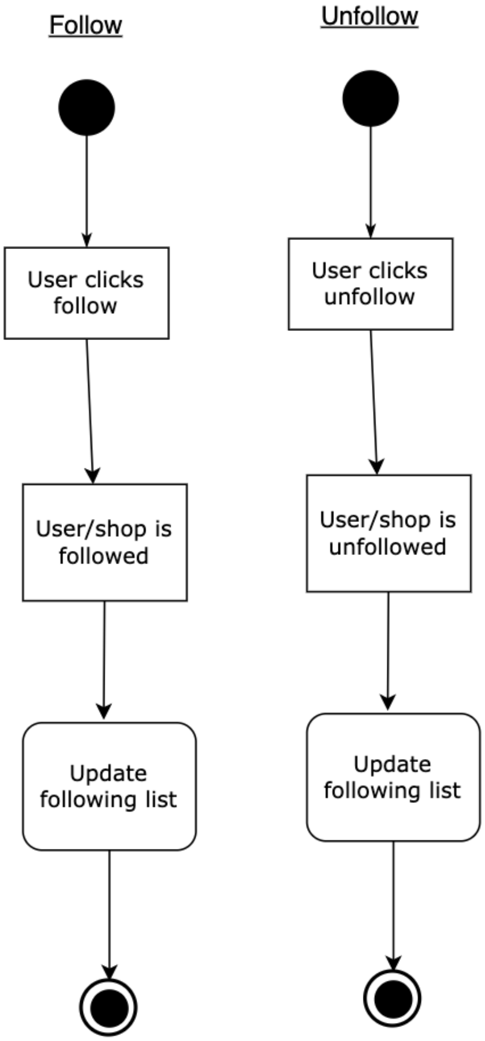

Forgot password
User clicks "forgot password"
User enters email
No
User has an account
Yes
Reset email sent with link
User sets new password in link
User password updated

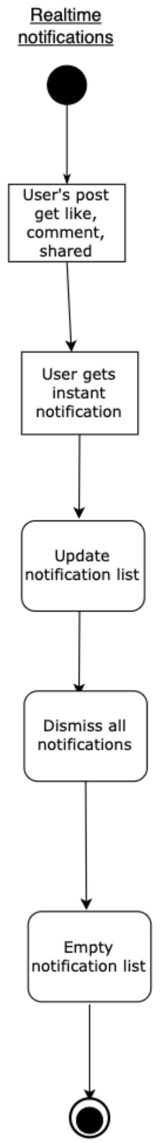

User Profile
User chooses what to be updated
User fills new information in a form
User clicks update
No
Information valid
yes
User profile updated

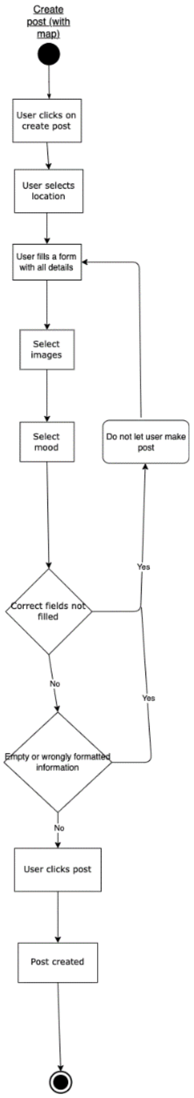

Profile Picture
Choices with profile picture
Set
Edit
User chooses a profile pic
User chooses new profile picture
User clicks update
User clicks update
No
No
File valid
File valid
yes
yes
User picture set
User picture updated

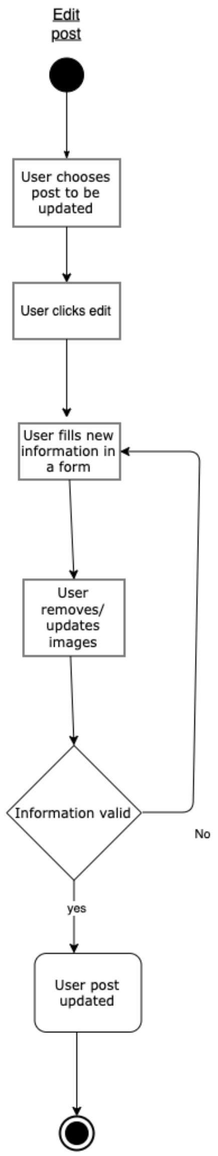

Follow suggestions
User chooses follow suggestions
Show users except following
User clicks follow on a user
User followed

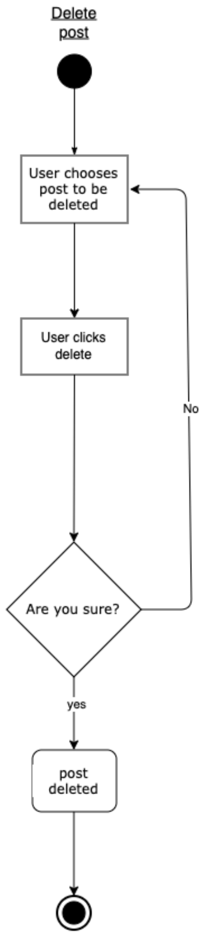

Search user/shop
User searches by username
Show all users/shops matching query
User clicks follow
User/Shop followed

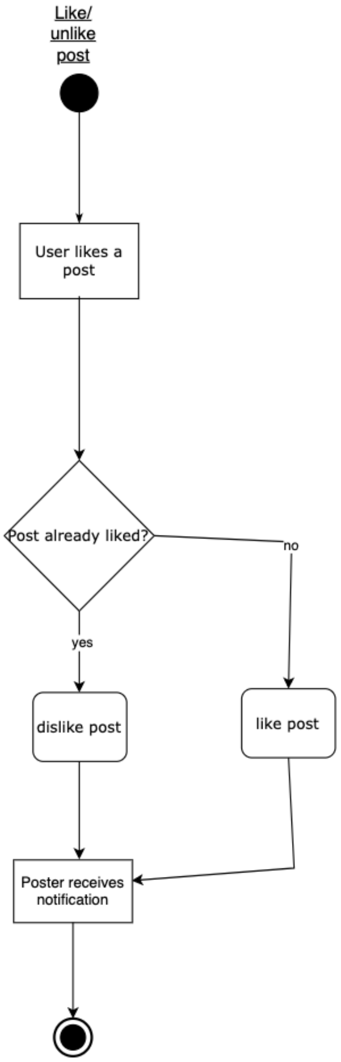

Follow
Unfollow
User clicks follow
User/shop is followed
Update following list
User clicks unfollow
User/shop is unfollowed
Update following list


Realtime notifications
User's post get like, comment, shared
User gets instant notification
Update notification list
Dismiss all notifications
Empty notification list

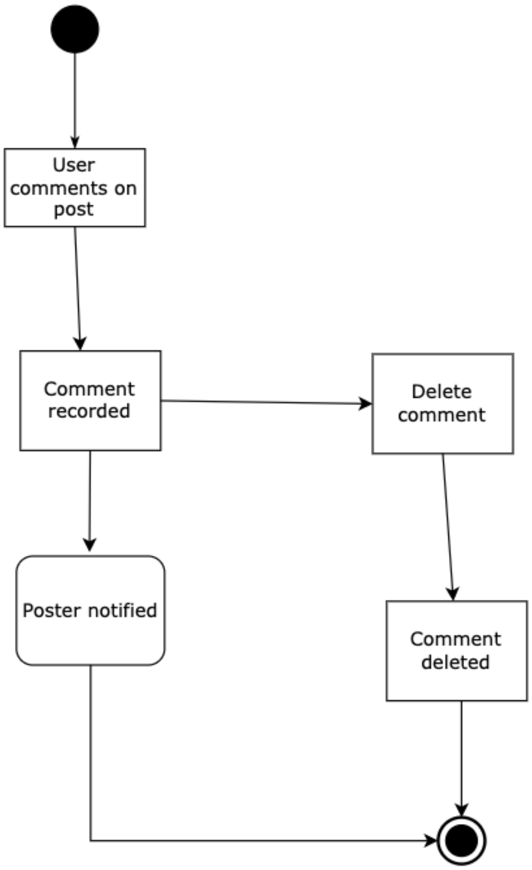

Create post (with map)
User clicks on create post
User selects location
User fills a form with all details
Select images
Select mood
Do not let user make post
Yes
Correct fields not filled
No
Yes
Empty or wrongly formatted information
No
User clicks post
Post created

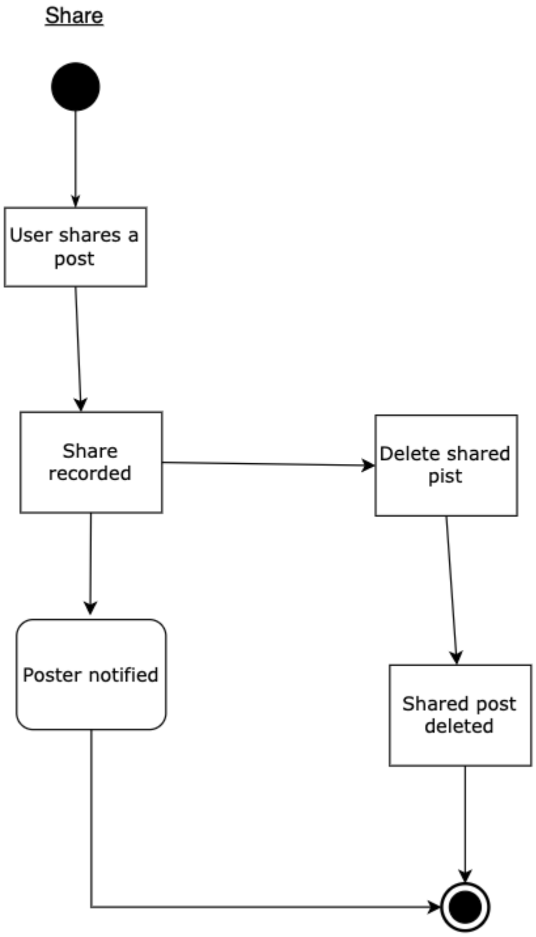

Edit post
User chooses post to be updated
User clicks edit
User fills new information in a form
User removes/ updates images
Information valid
No
yes
User post updated

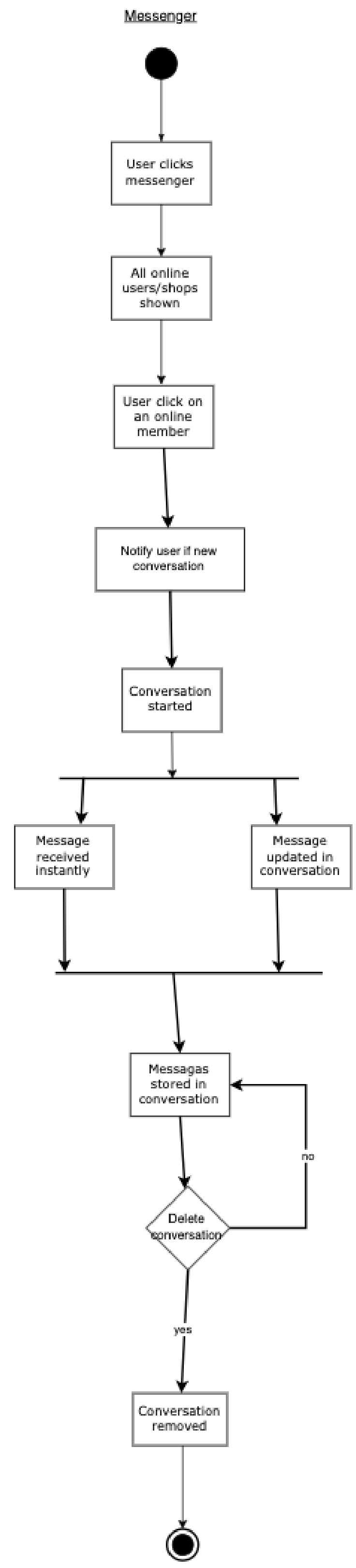

Delete post
User chooses post to be deleted
User clicks delete
No
Are you sure?
yes
post deleted


Like/
unlike
post
User likes a post
Post already liked?
no
yes
dislike post
like post
Poster receives notification

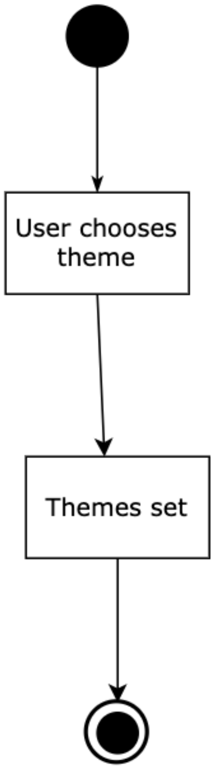
Comment

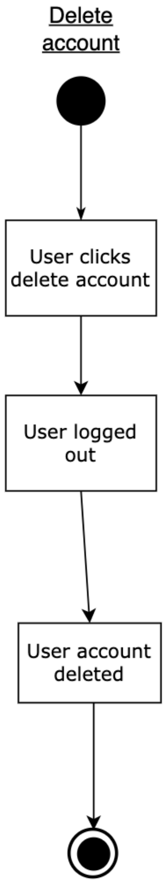

User comments on post
Comment recorded
Delete comment
Poster notified
Comment deleted

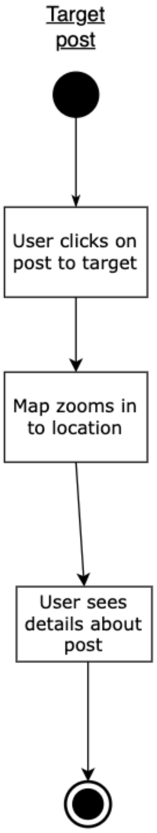

Share
User shares a post
Share recorded
Delete shared pist
Poster notified
Shared post deleted

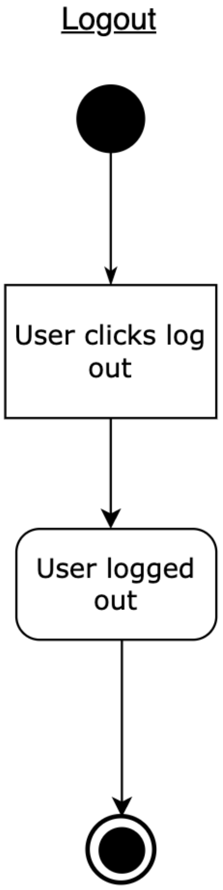

Messenger
User clicks messenger
All online users/shops shown
User click on an online member
Notify user if new conversation
Conversation started
Message received instantly
Message updated in conversation
Messagas stored in conversation
no
Delete conversation
yes
Conversation removed

Themes

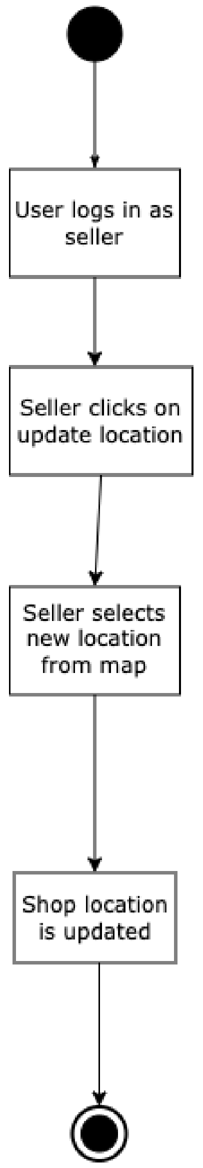


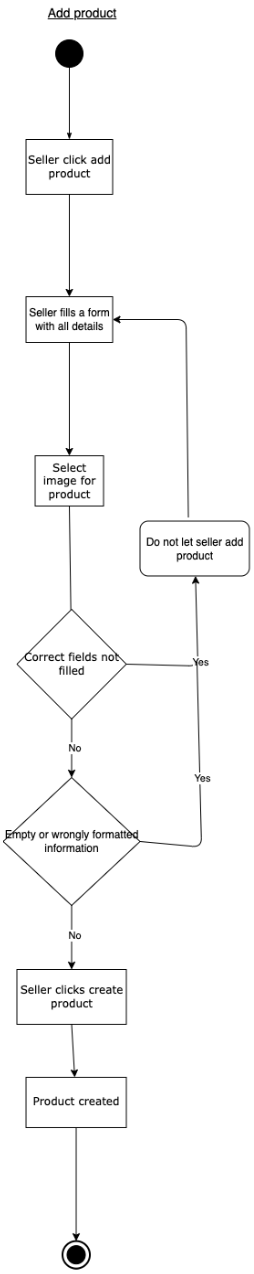

Delete account
User clicks delete account
User logged out
User account deleted


Target post
User clicks on post to target
Map zooms in to location
User sees details about post

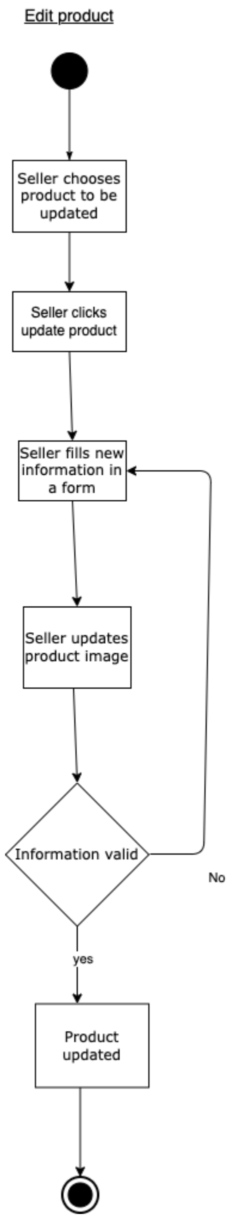

Logout
User clicks log out
User logged out

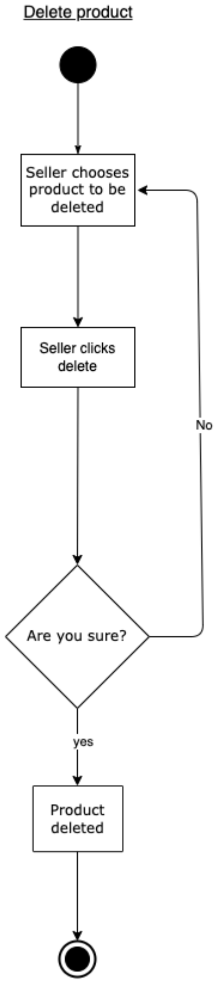
Edit shop location

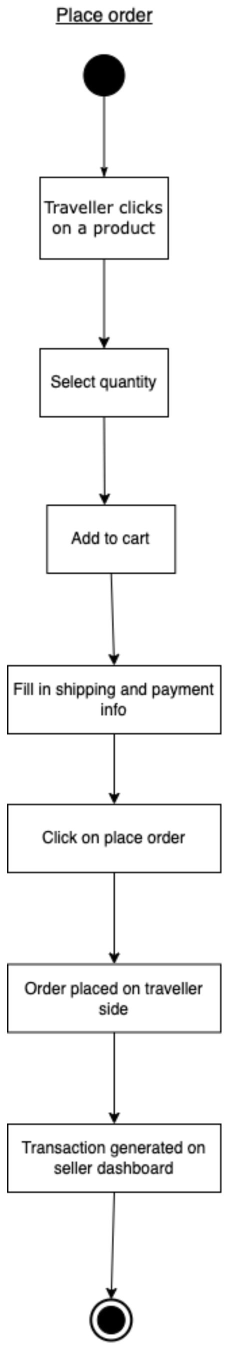

User logs in as seller
Seller clicks on update location
Seller selects new location from map
Shop location is updated



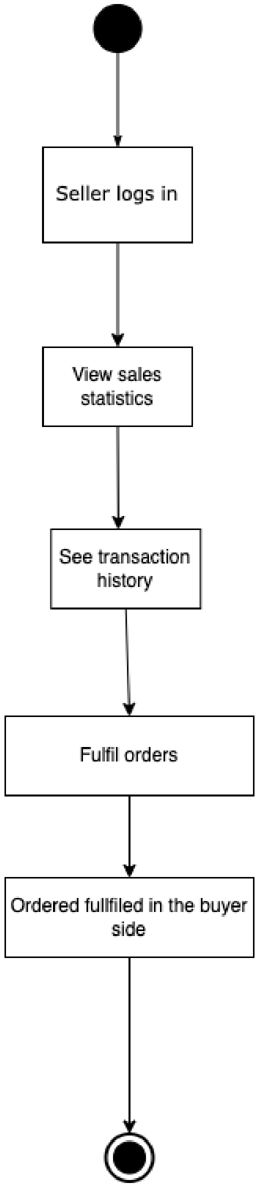
Add product
Seller click add product
Seller fills a form with all details
Select image for product
Do not let seller add product
Correct fields not filled
Yes
No
Yes
Empty or wrongly formatted information
No
Seller clicks create product
Product created

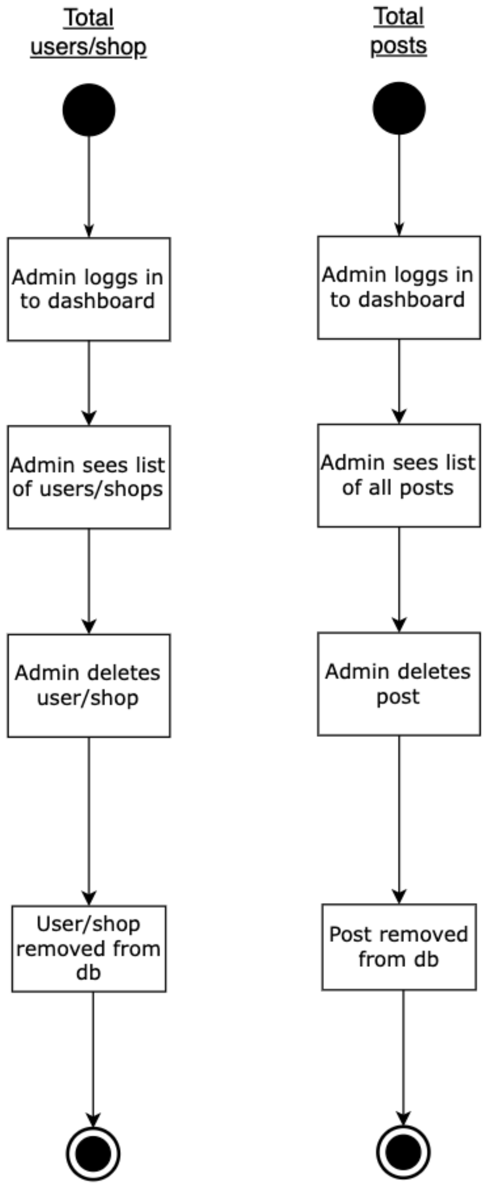

Edit product
Seller chooses product to be updated
Seller clicks update product
Seller fills new information in a form
Seller updates product image
Information valid
No
yes
Product updated


Delete product
Seller chooses product to be deleted
Seller clicks delete
No
Are you sure?
yes
Product deleted

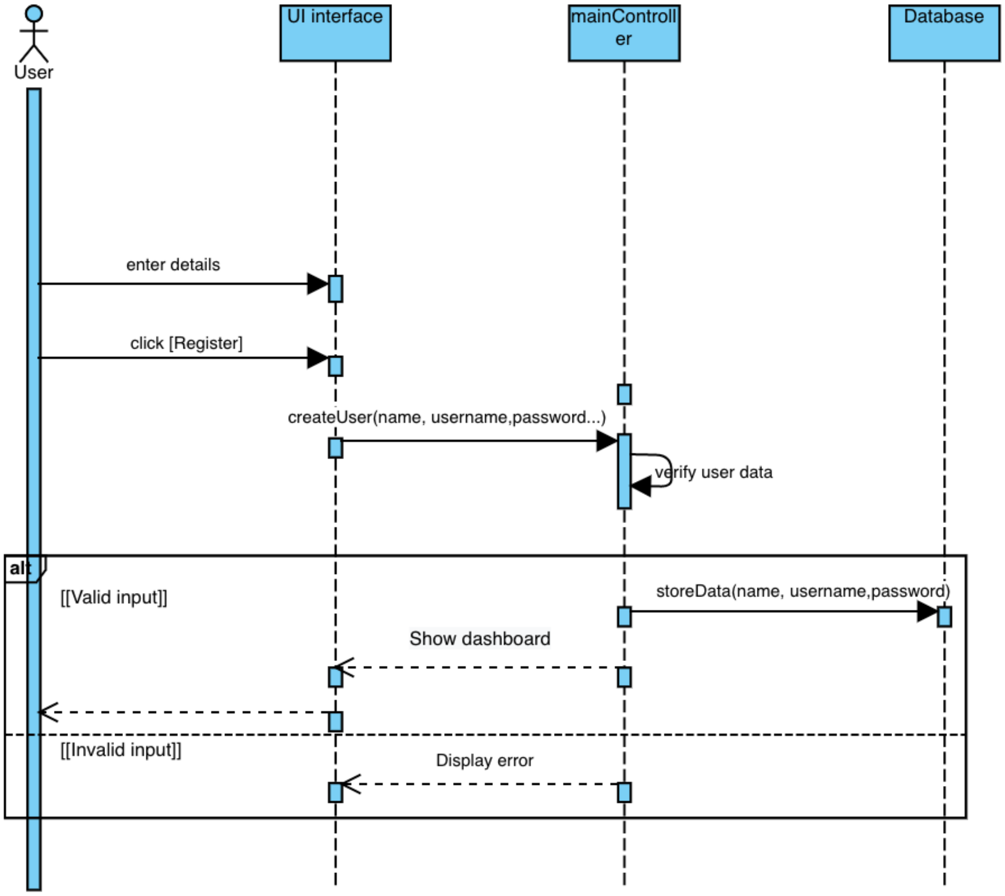

Place order
Traveller clicks on a product
Select quantity
Add to cart
Fill in shipping and payment info
Click on place order
Order placed on traveller side
Transaction generated on seller dashboard

Shop dashboard

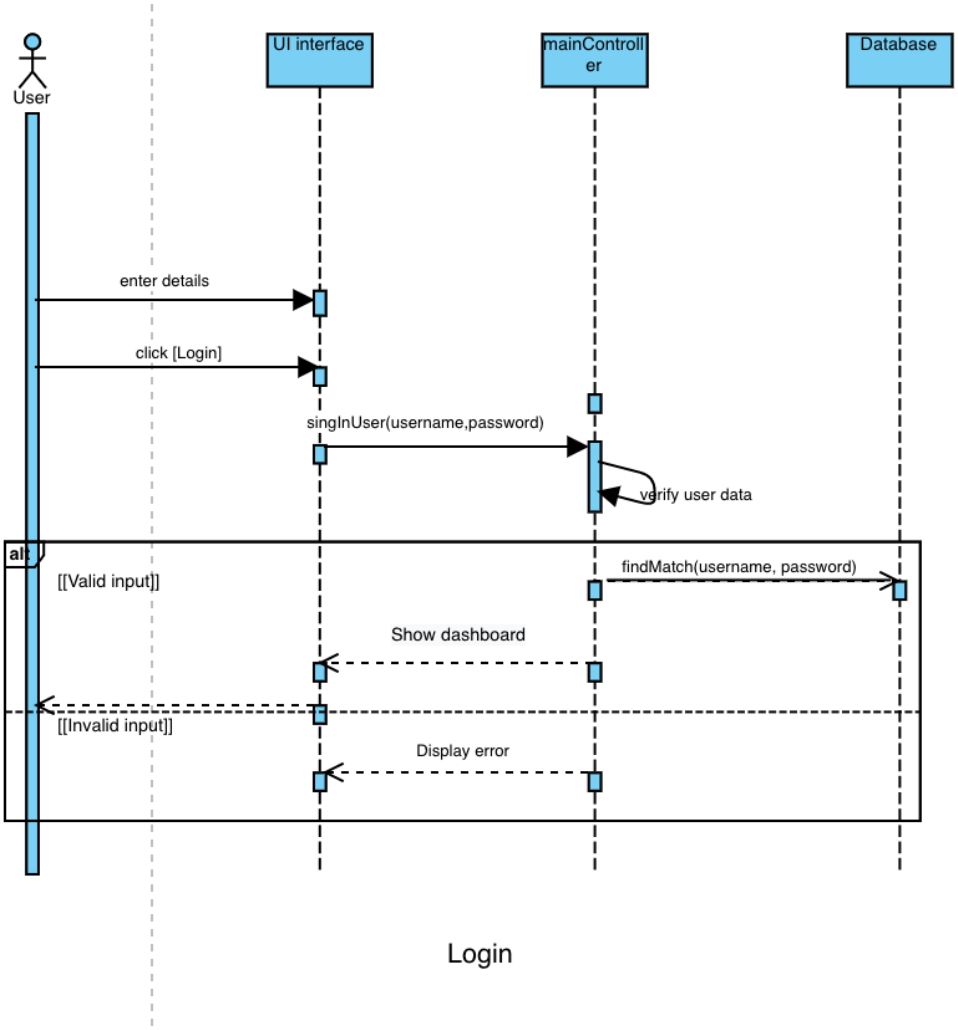



Total users/shop
Admin loggs in to dashboard
Admin sees list of users/shops
Admin deletes user/shop
User/shop removed from db
Total posts
Admin loggs in to dashboard
Admin sees list of all posts
Admin deletes post
Post removed from db

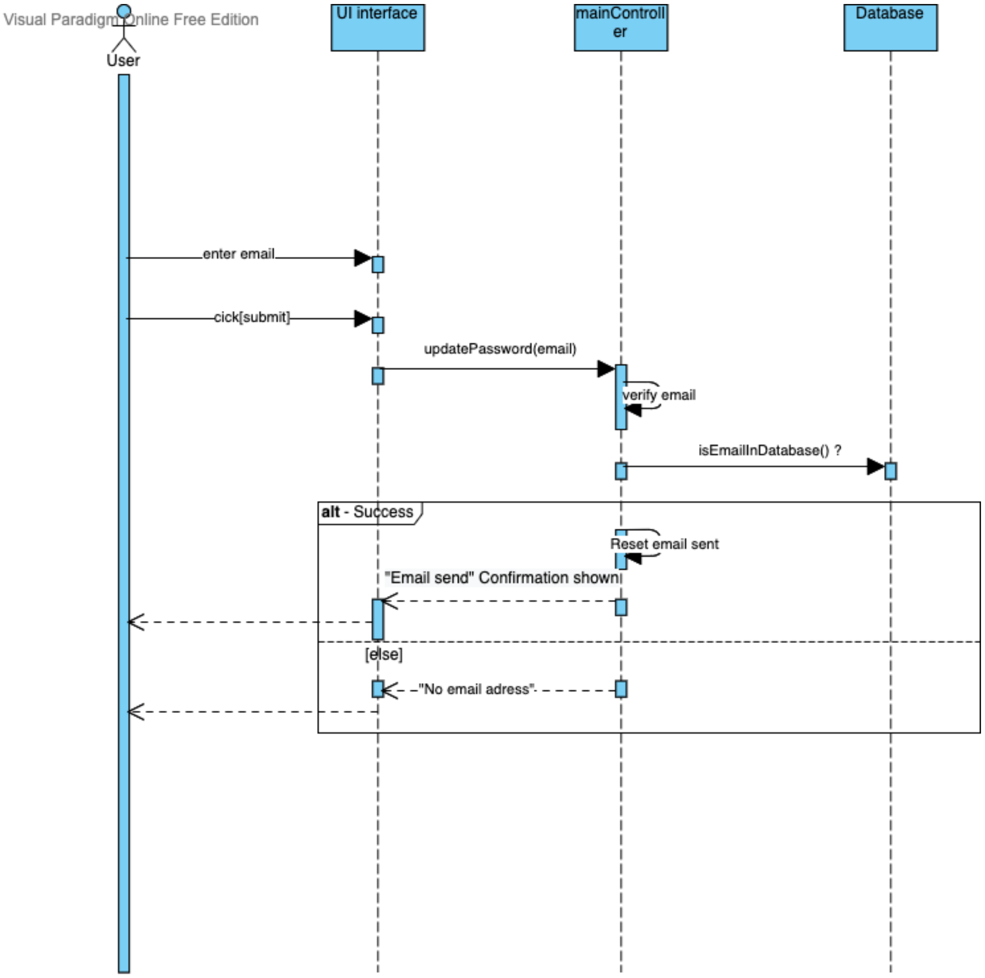
3.5 Sequence Diagrams



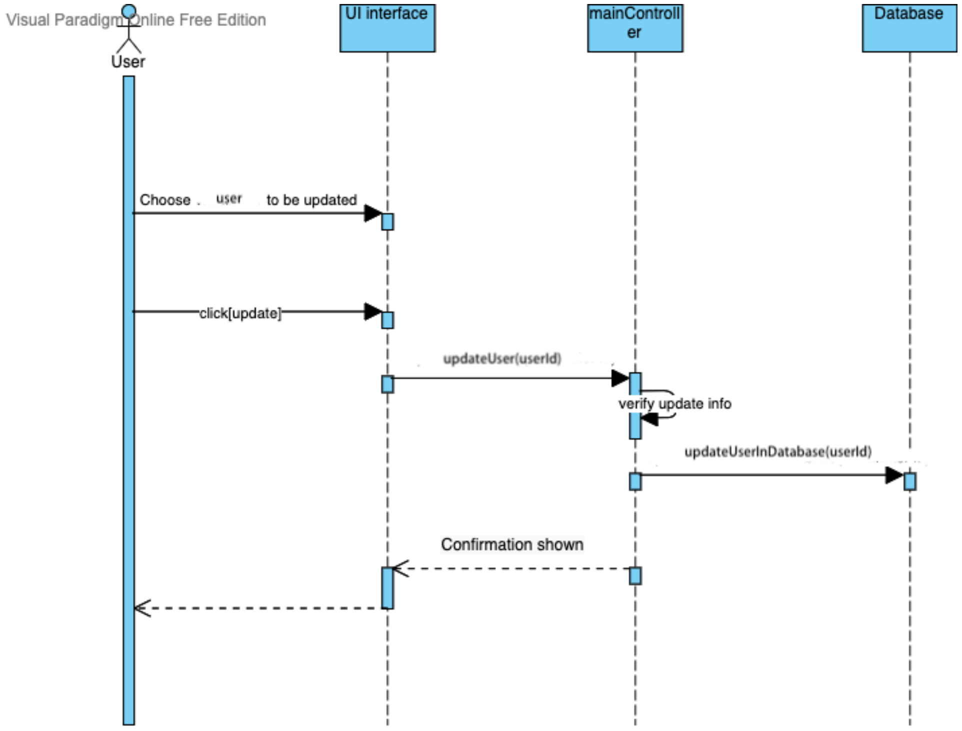
Register


User
UI interface
mainController
Database
enter details
click [Login]
singInUser(username,password)
verify user data
alt
[[Valid input]]
findMatch(username, password)
Show dashboard
[[Invalid input]]
Display error

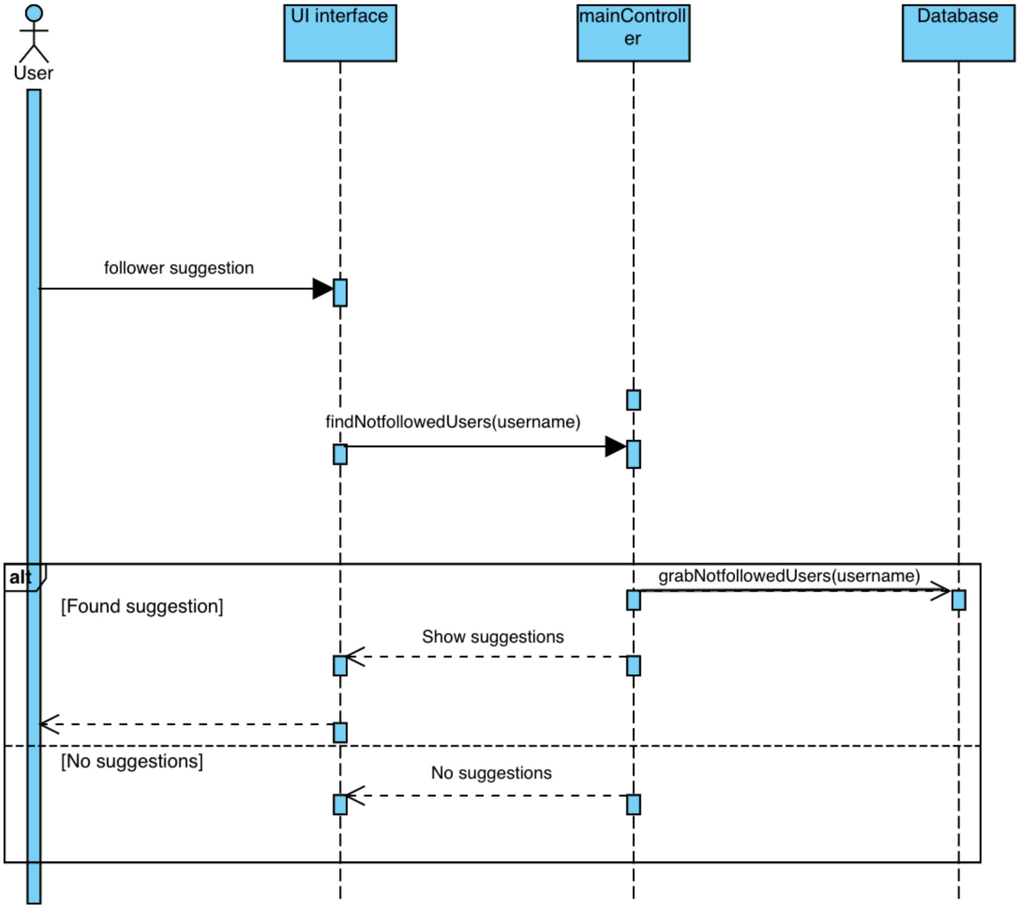
Login



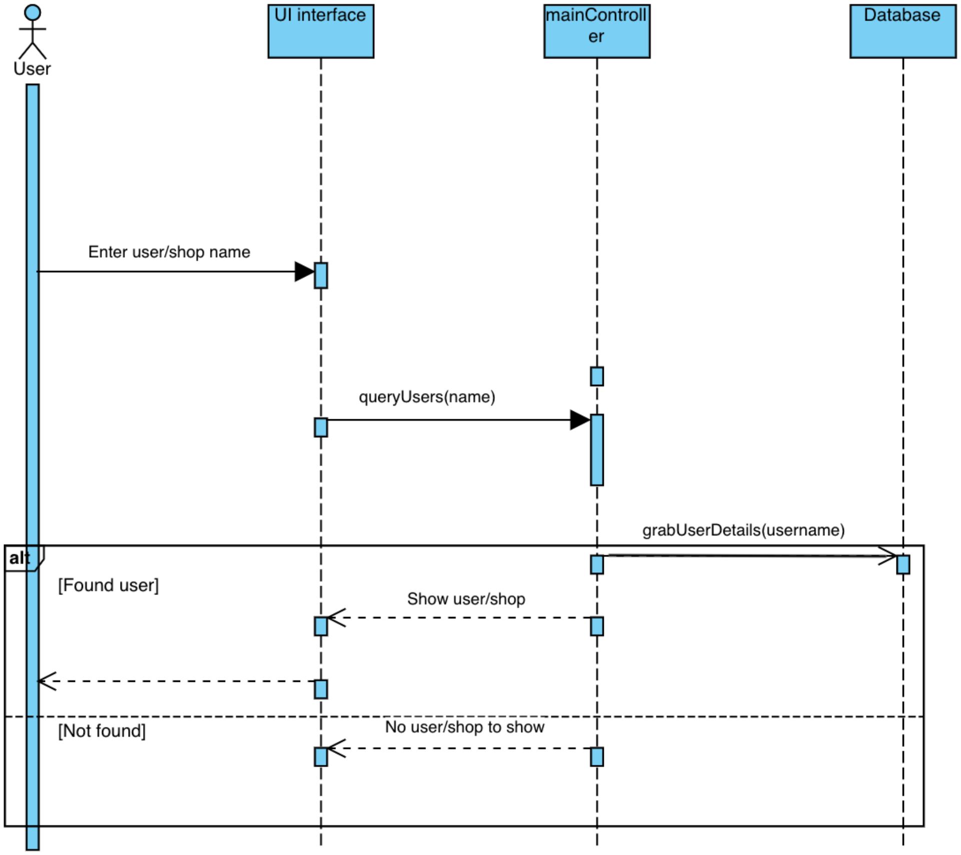
Forgot password



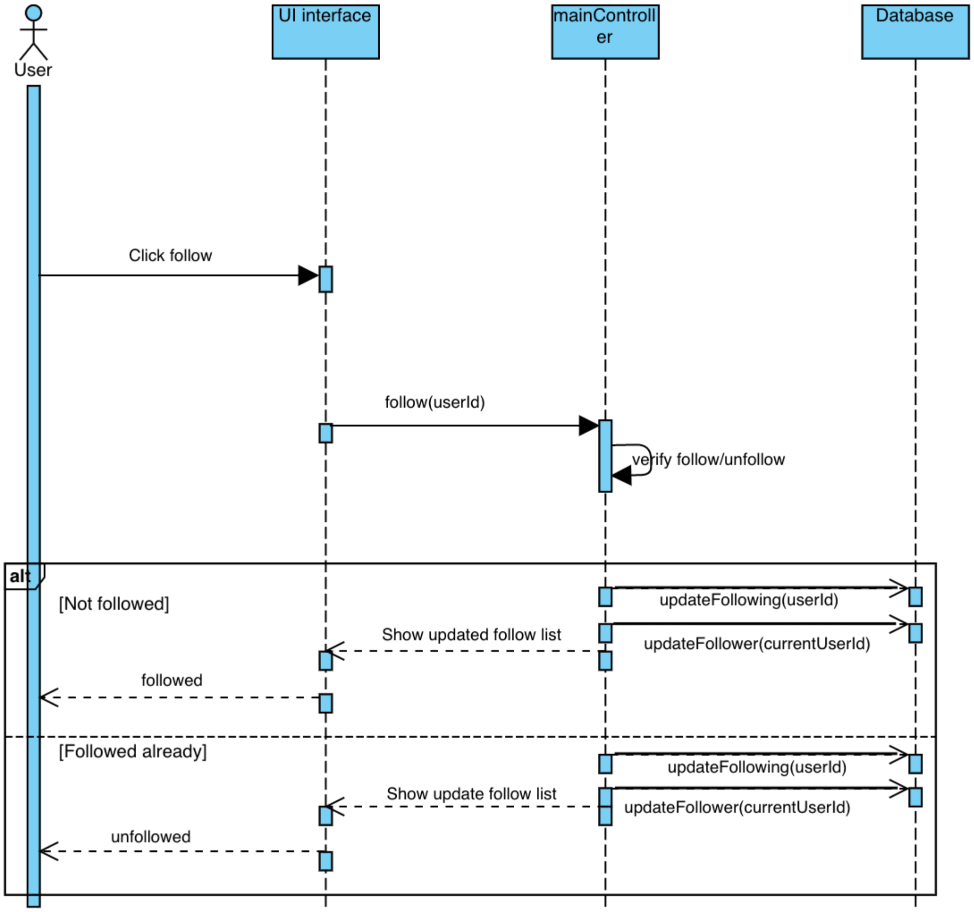
Update user profile (lastname, firstname,profile picture)



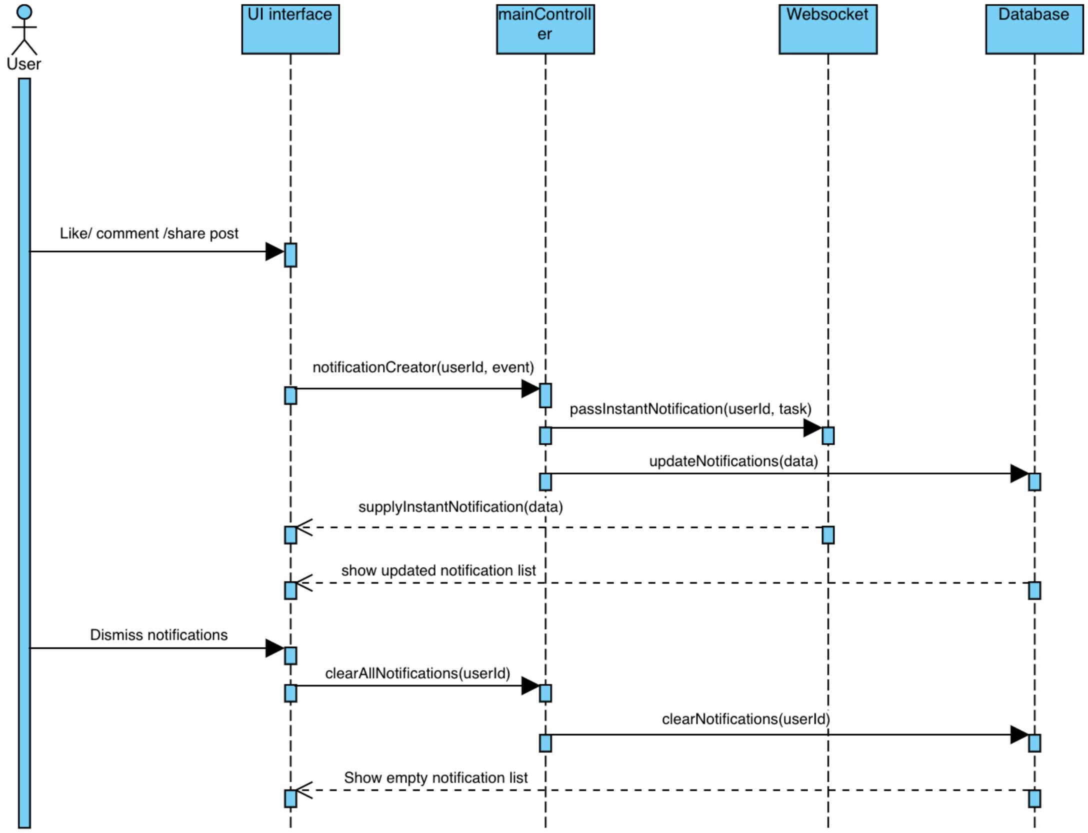
Follow suggestions



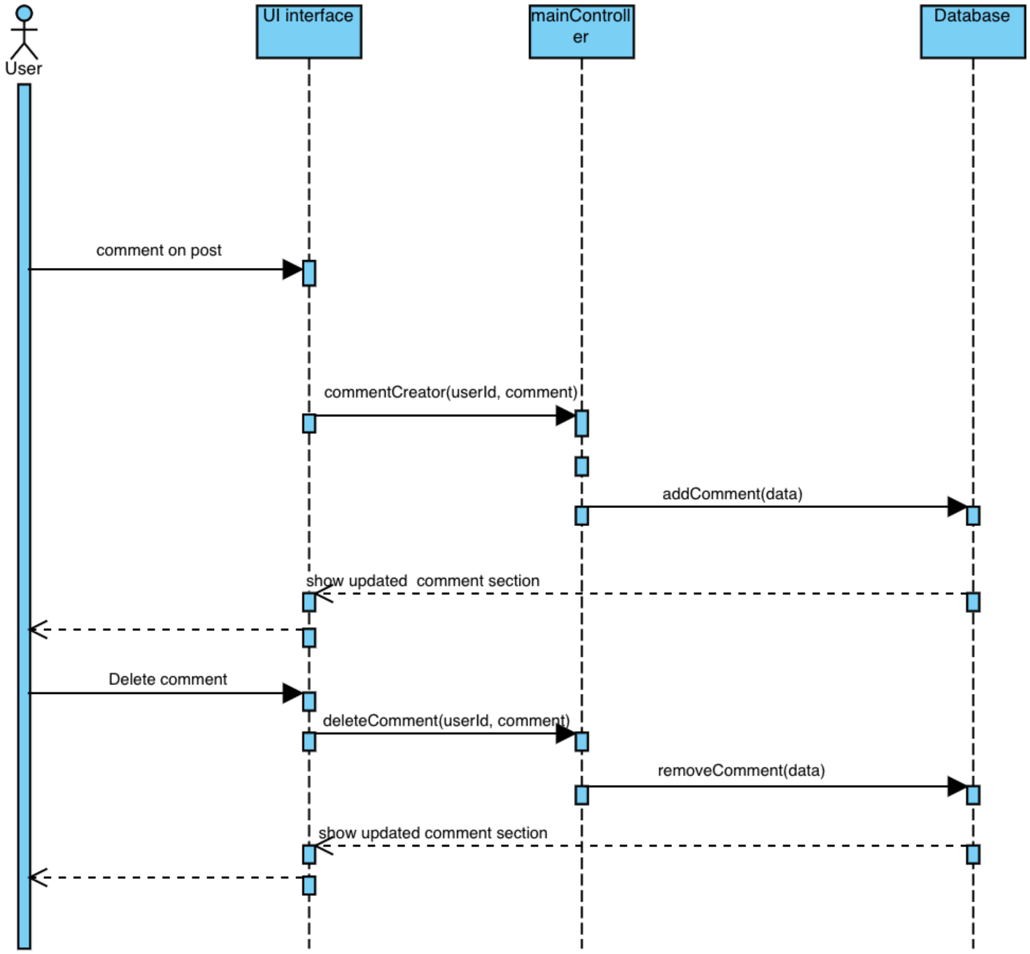
Search user/shop



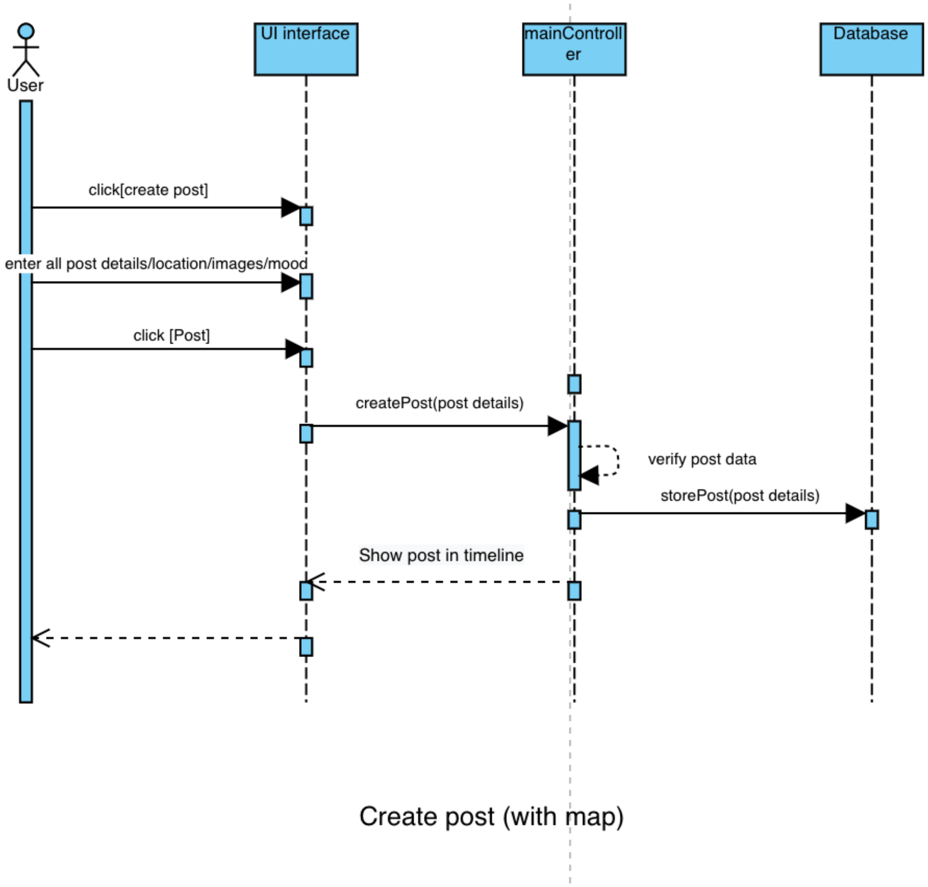
Follow/Unfollow


User
UI interface
mainControll
er
Websocket
Database
Like/ comment /share post
notificationCreator(userId, event)
passInstantNotification(userId, task)
updateNotifications(data)
supplyInstantNotification(data)
show updated notification list
Dismiss notifications
clearAllNotifications(userId)
clearNotifications(userId)
Show empty notification list

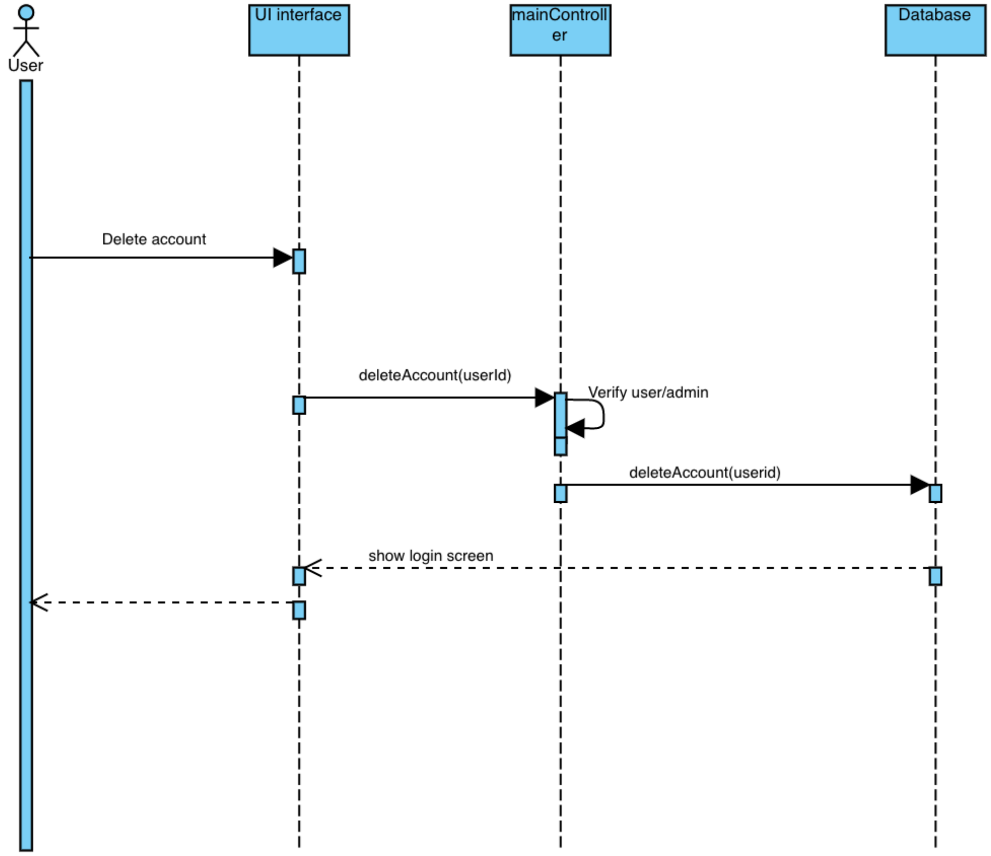
Realtime notification


User
UI interface
mainControll
er
Database
comment on post
commentCreator(userId, comment)
addComment(data)
show updated comment section
Delete comment
deleteComment(userId, comment)
removeComment(data)
show updated comment section

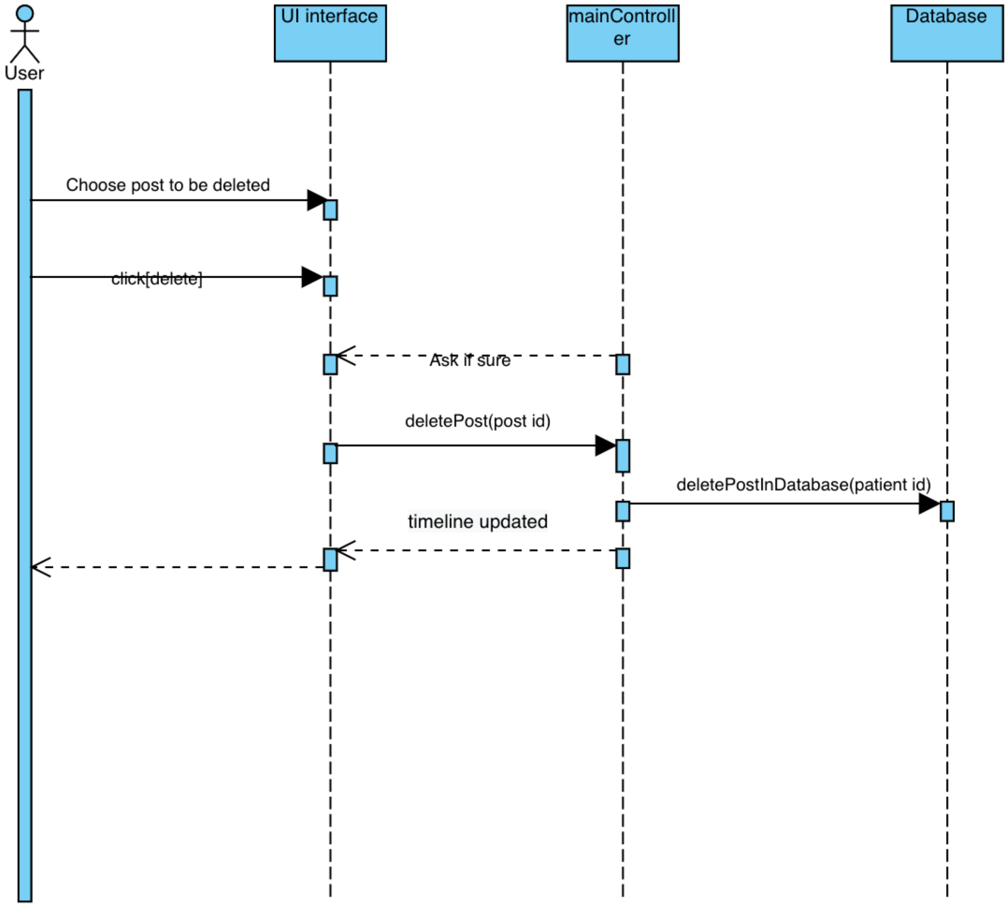
Add comment/ delete comment



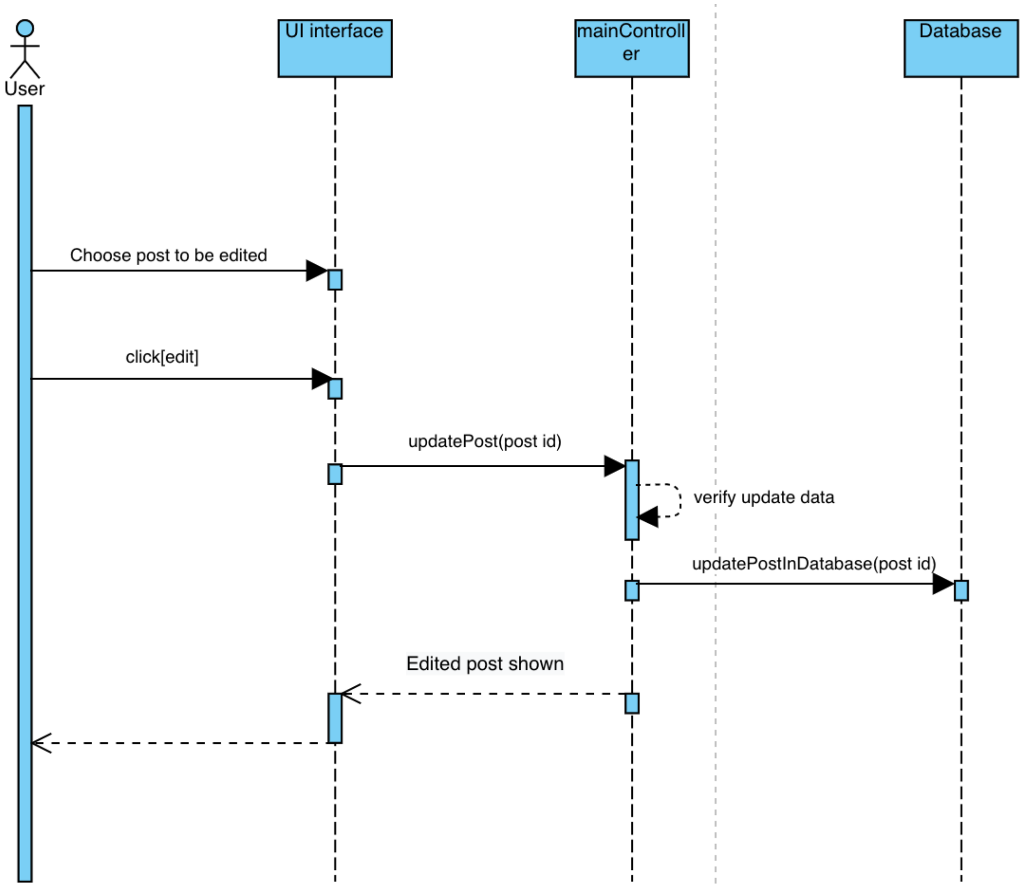
Create post (with map)



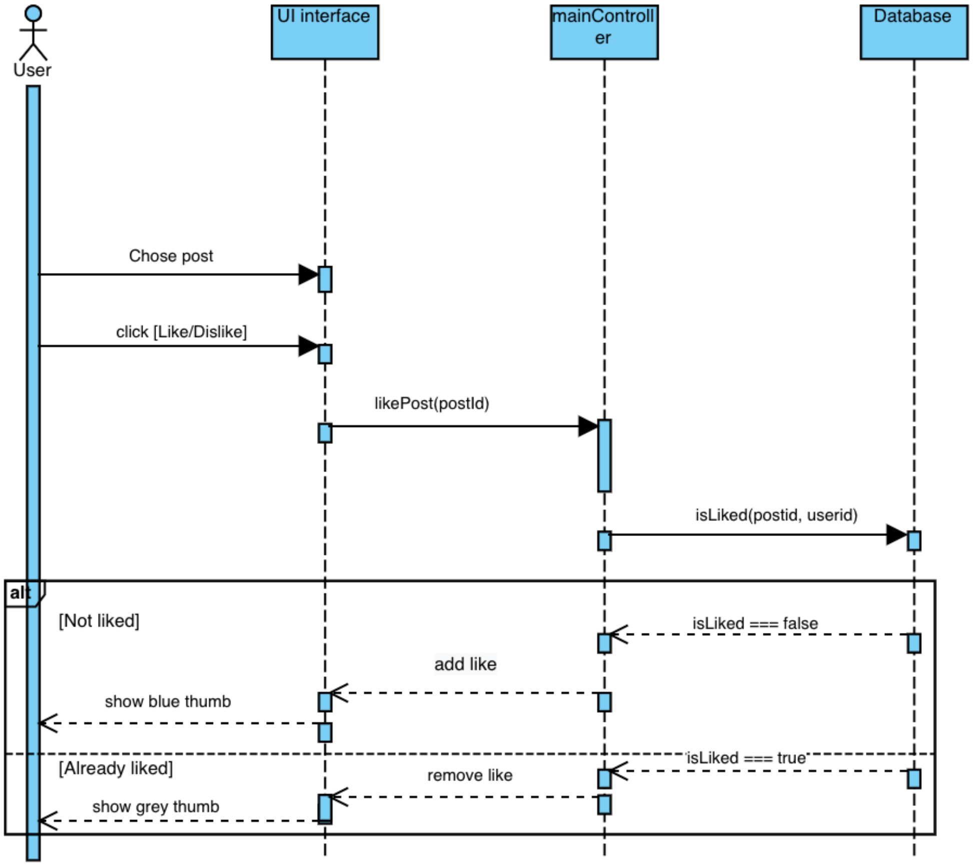
Delete account


User
UI interface
mainControll
er
Database
Choose post to be deleted
click[delete]
Ask if sure
deletePost(post id)
deletePostInDatabase(patient id)
timeline updated

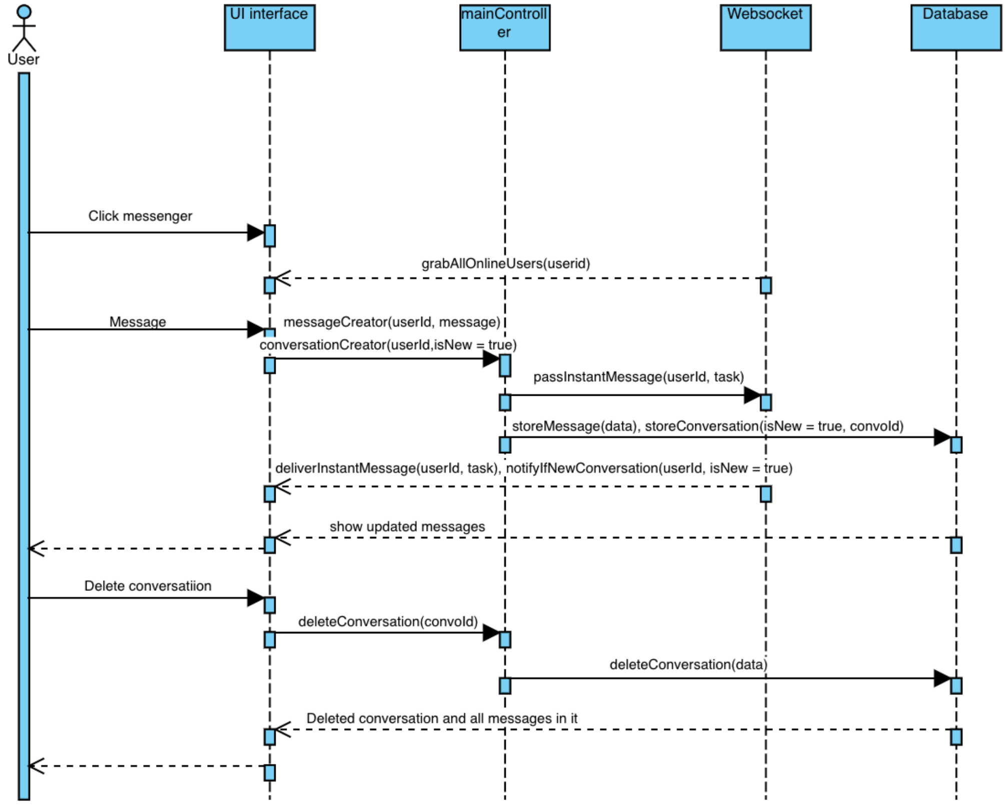
Delete post


User
UI interface
mainController
Database
Choose post to be edited
click[edit]
updatePost(post id)
verify update data
updatePostInDatabase(post id)
Edited post shown

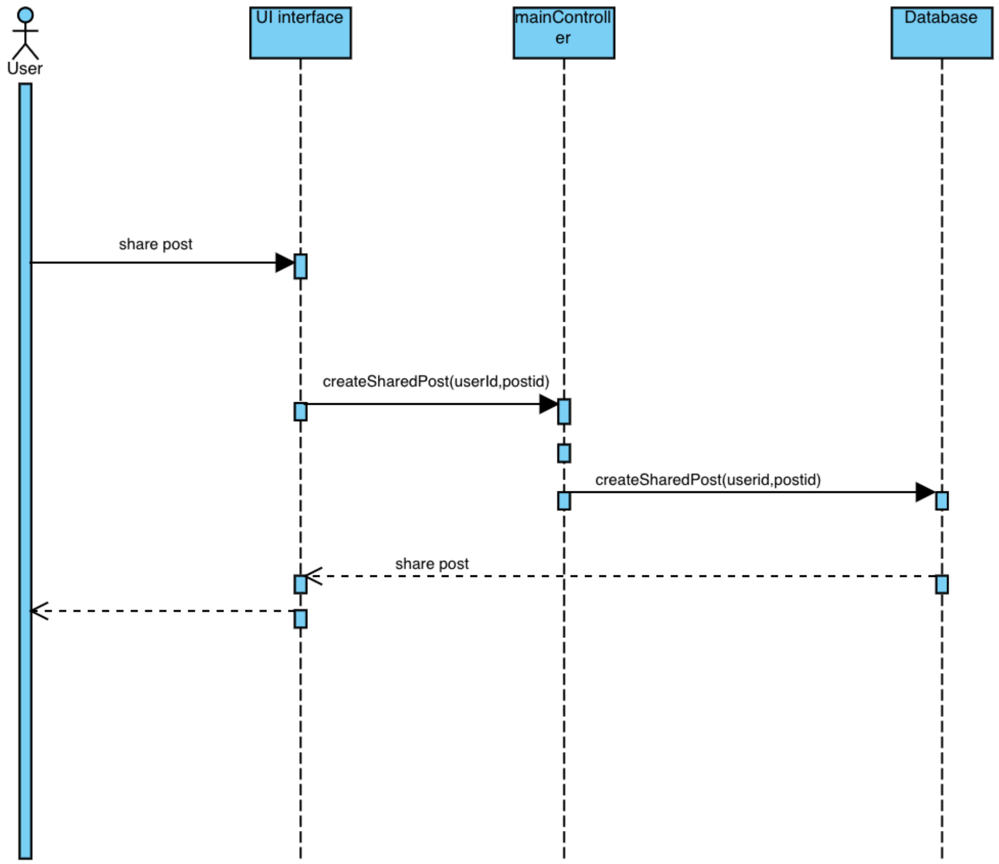
Edit post



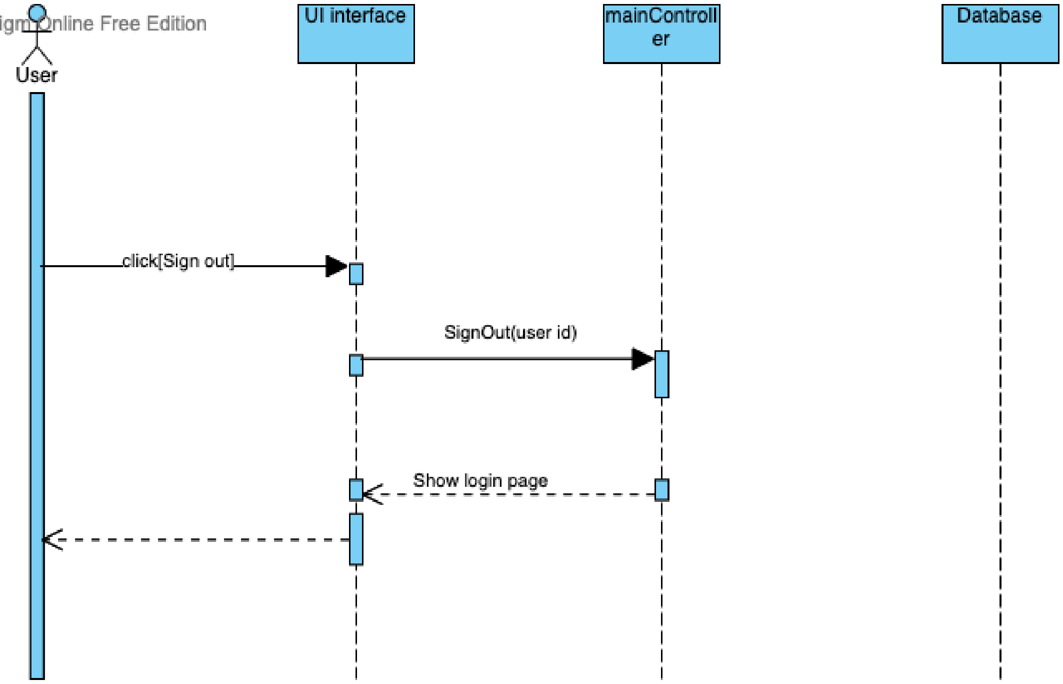
Like/ dislike


User
UI interface
mainControll er
Websocket
Database
Click messenger
grabAllOnlineUsers(userid)
Message
messageCreator(userId, message)
conversationCreator(userId,isNew = true)
passInstantMessage(userId, task)
storeMessage(data), storeConversation(isNew = true, convoId)
deliverInstantMessage(userId, task), notifyIfNewConversation(userId, isNew = true)
show updated messages
Delete conversatiion
deleteConversation(convoId)
deleteConversation(data)
Deleted conversation and all messages in it

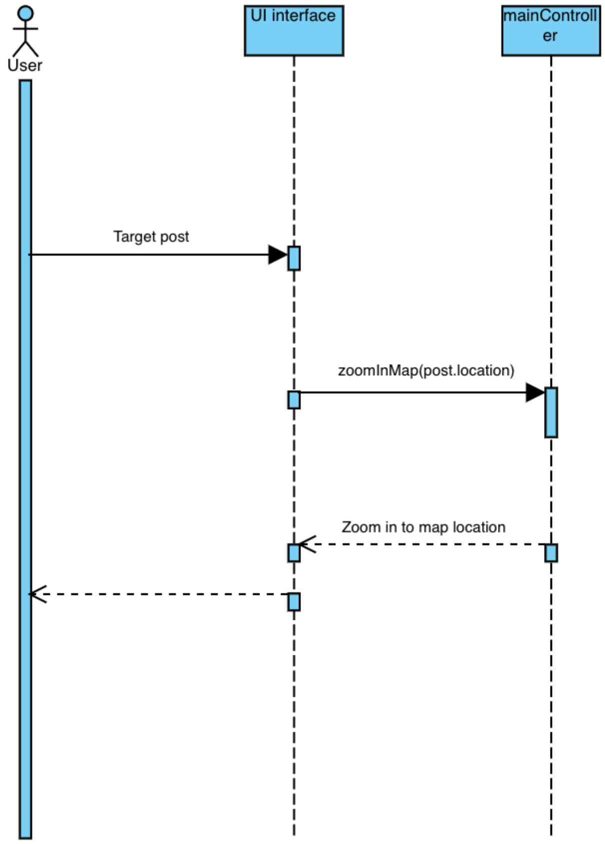
Messenger


User
UI interface
mainController
Database
share post
createSharedPost(userId,postid)
createSharedPost(userid,postid)
share post

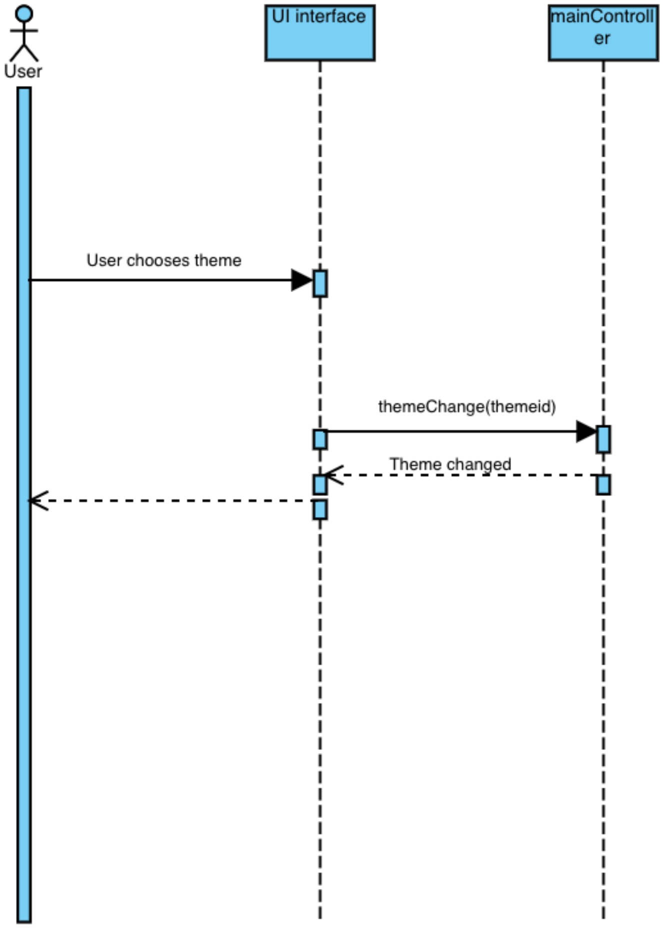
Share post


User
UI interface
mainController
Database
click[Sign out]
SignOut(user id)
Show login page

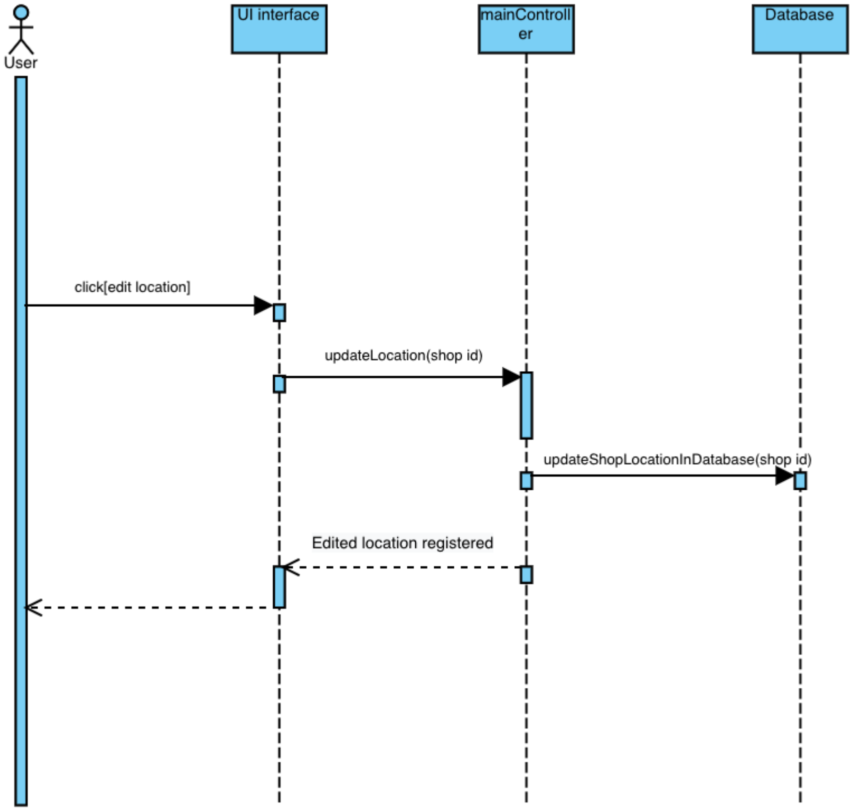
Sign out



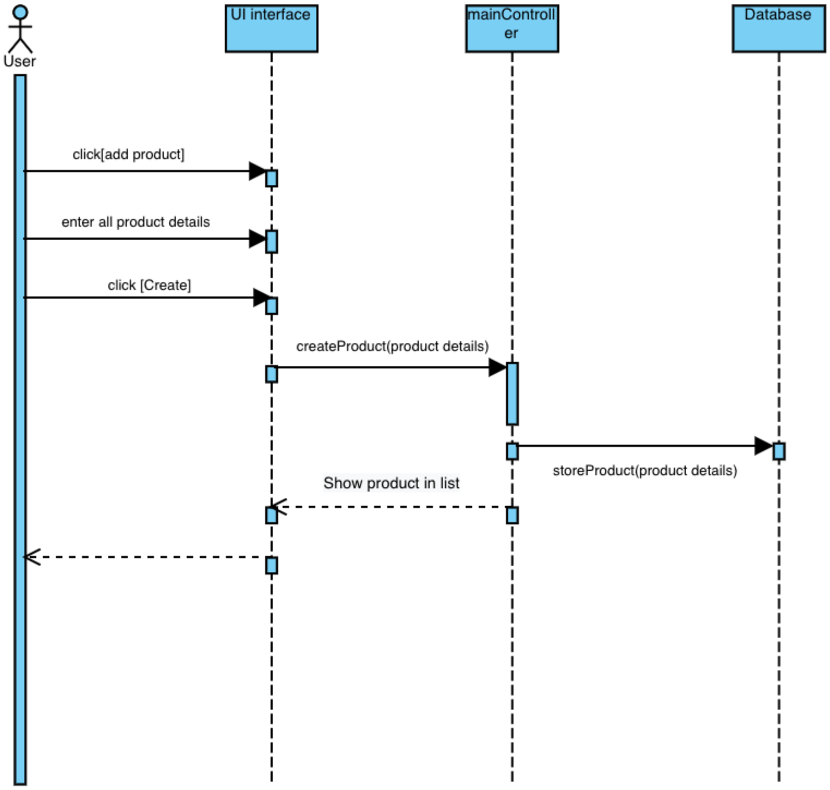
Target post (on map)


User
UI interface
mainController
User chooses theme
themeChange(themeid)
Theme changed

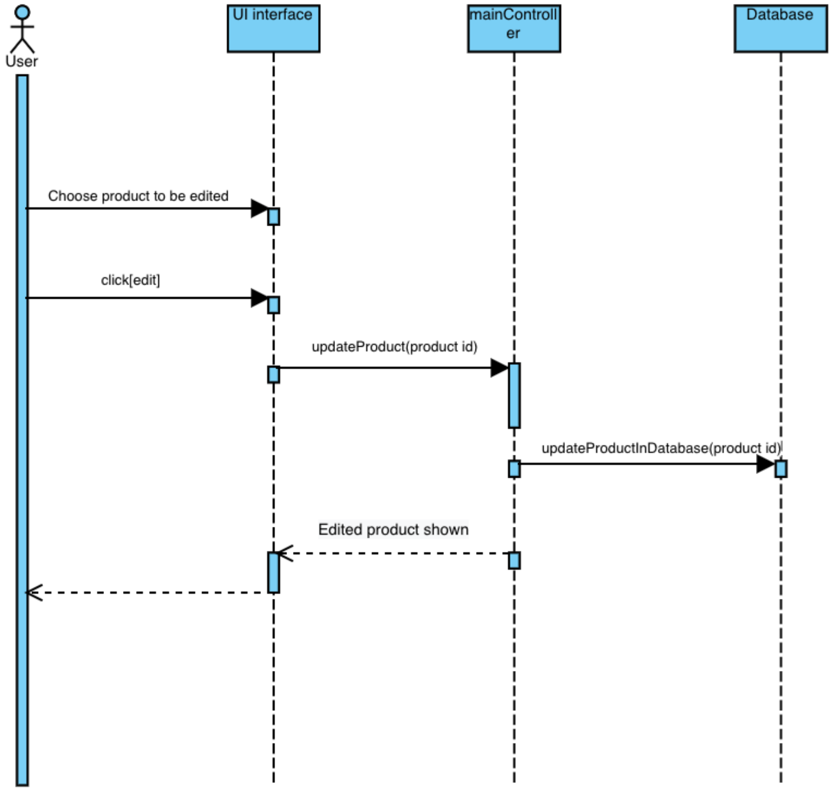
Theme



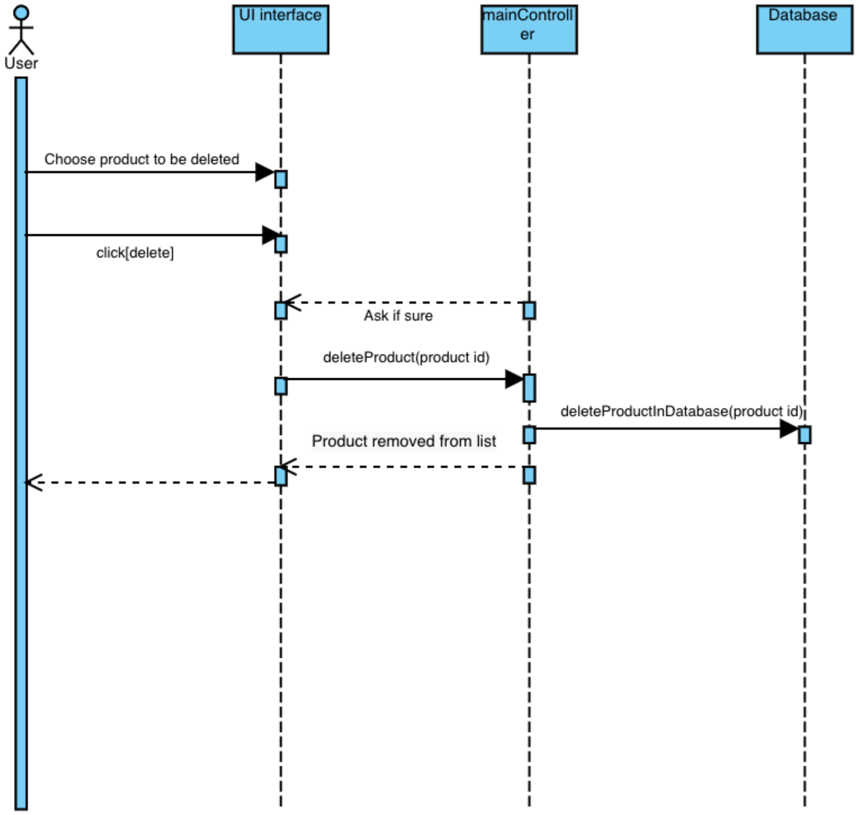
Edit shop locaton


User
UI interface
mainController
Database
click[add product]
enter all product details
click [Create]
createProduct(product details)
storeProduct(product details)
Show product in list

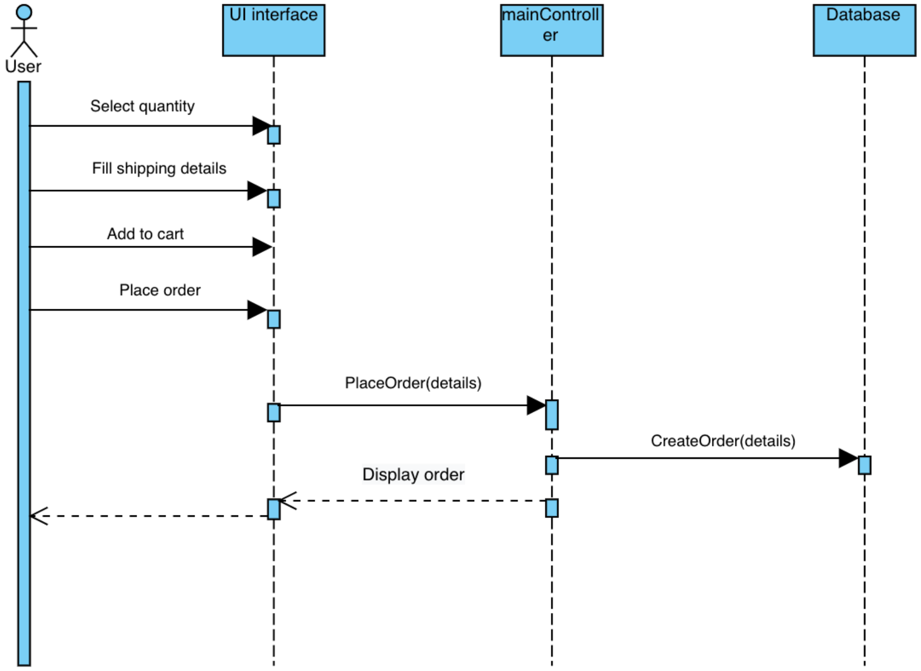
Add product


User
UI interface
mainControll
er
Database
Choose product to be edited
click[edit]
updateProduct(product id)
updateProductInDatabase(product id)
Edited product shown

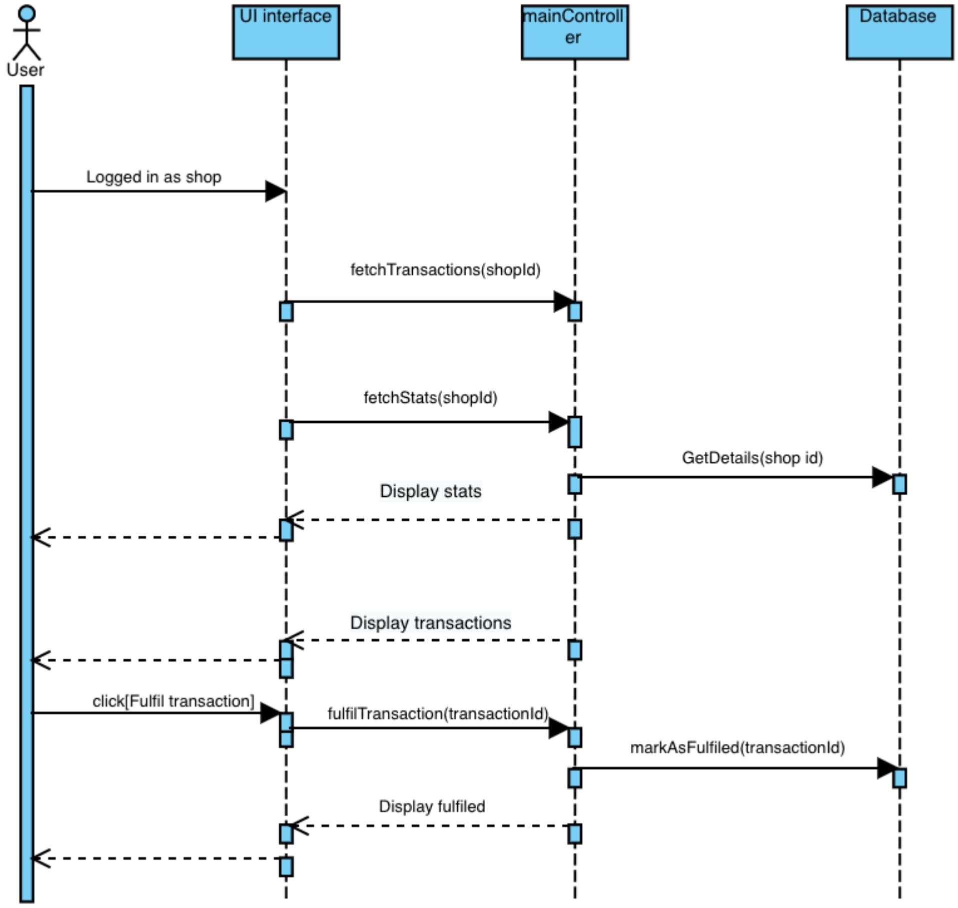
Edit product


User
UI interface
mainController
Database
Choose product to be deleted
click[delete]
Ask if sure
deleteProduct(product id)
deleteProductInDatabase(product id)
Product removed from list

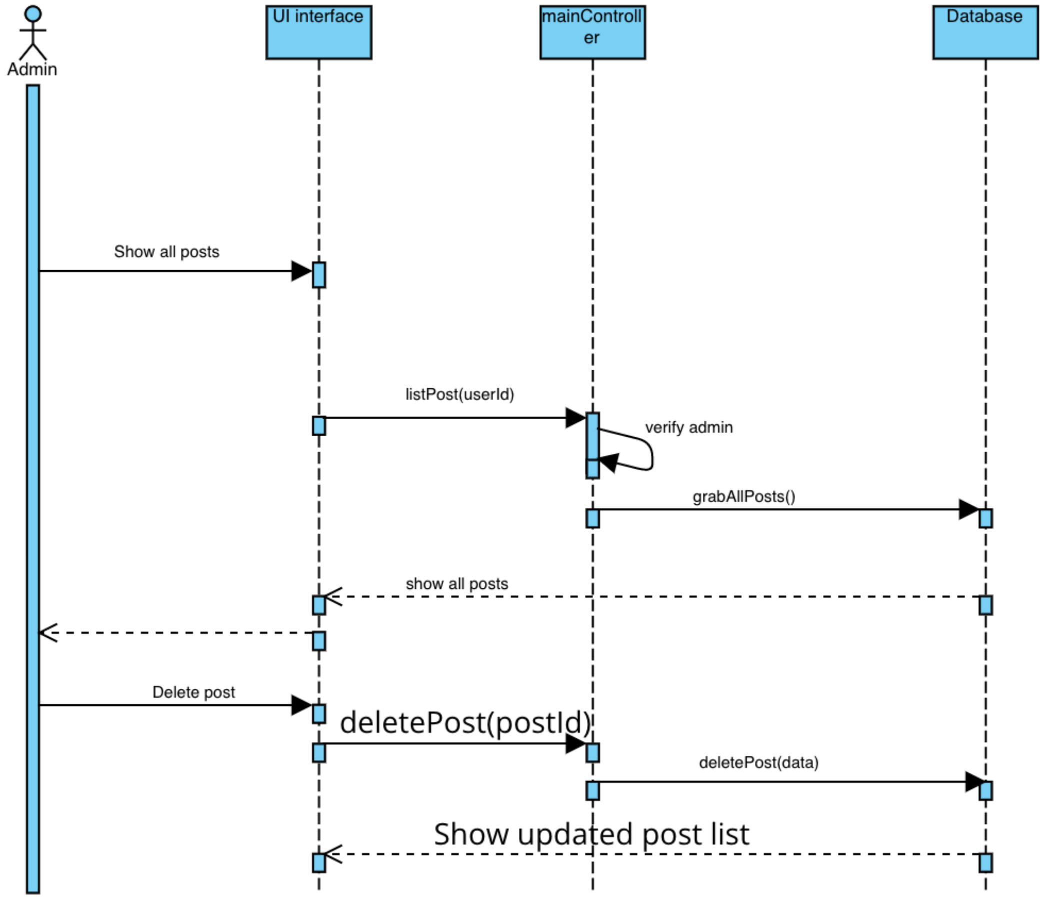
Delete product


User
UI interface
mainControll
er
Database
Select quantity
Fill shipping details
Add to cart
Place order
PlaceOrder(details)
CreateOrder(details)
Display order

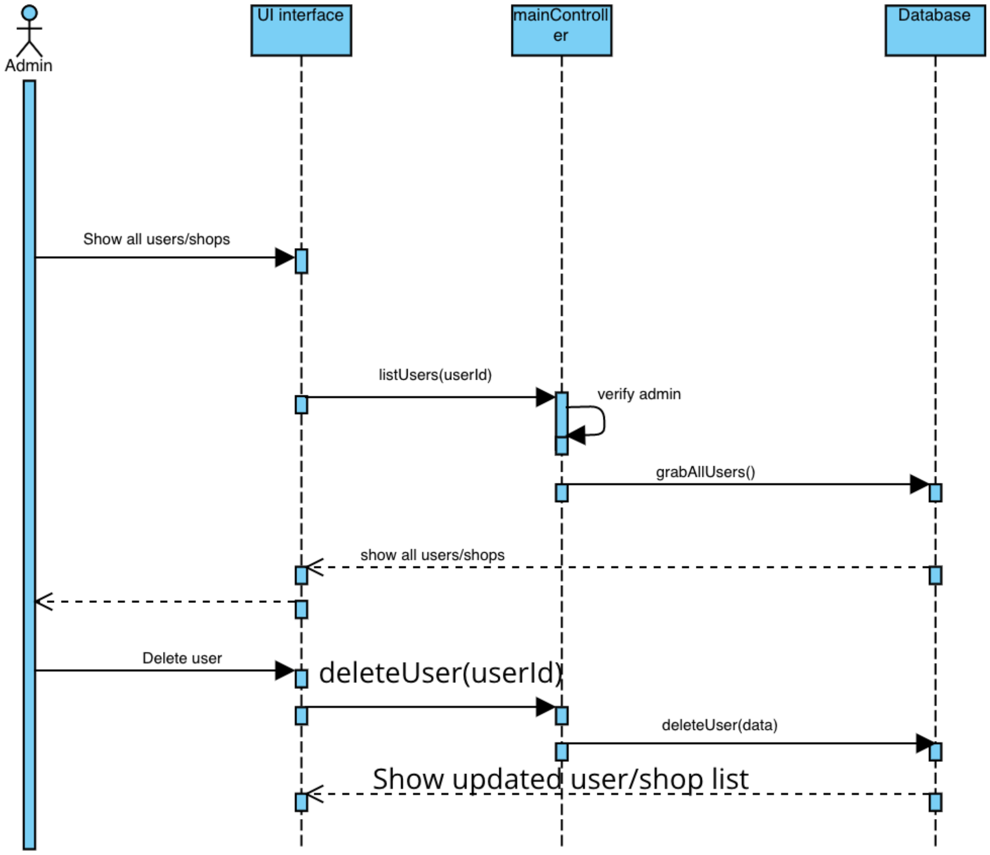
Place order



Shop dashboard


Admin
UI interface
mainController
Database
Show all posts
listPost(userId)
verify admin
grabAllPosts()
show all posts
Delete post
deletePost(postId)
deletePost(data)
Show updated post list

Total posts



Total user/shop

Chapter 4. Testing

4.1 Test Approach

I have used Blackbox testing to ensure the quality of my system. I focused on functional testing and structural testing mainly. Black Box Testing (also called functional testing) is testing that ignores the internal mechanism of a system or component and focuses solely on the outputs generated in response to selected inputs and execution conditions.

4.2 Pass/Fail Criteria

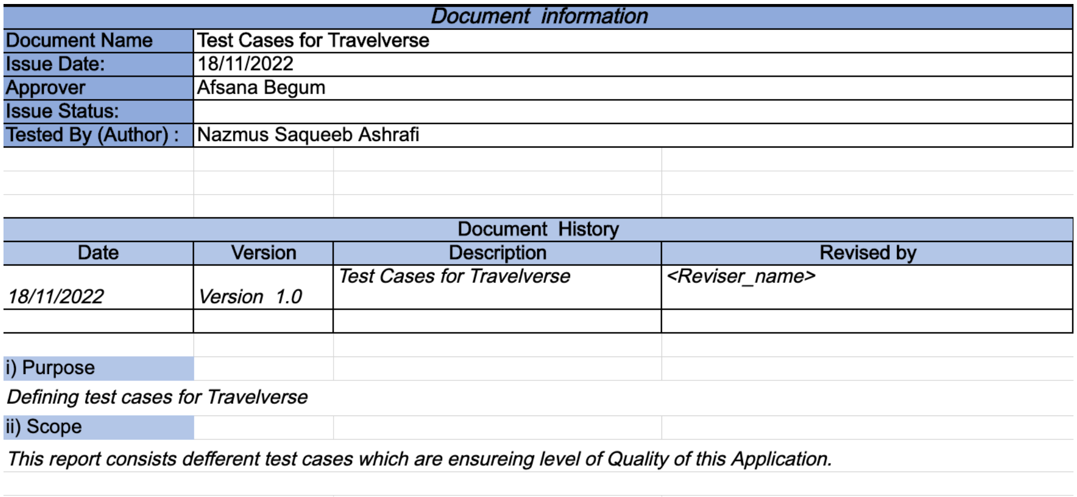
I have defined some pass/fail criteria to find the proficiency of the software.

· System crash is considered a failure case.

· If any criteria pass 100% of testing, then it is considered as pass criteria only.

· Which data can't be displayed to the system properly, then it is also to be considered as fail criteria.

4.3 Test Cases and Priority



Document information	
Document Name	Test Cases for Travelverse
Issue Date:	18/11/2022
Approver	Afsana Begum
Issue Status:	
Tested By (Author) :	Nazmus Saqueeb Ashrafi

Document History			
Date	Version	Description	Revised by
18/11/2022	*Version 1.0*	*Test Cases for Travelverse*	*<Reviser_name>*

i) Purpose

Defining test cases for Travelverse

ii) Scope

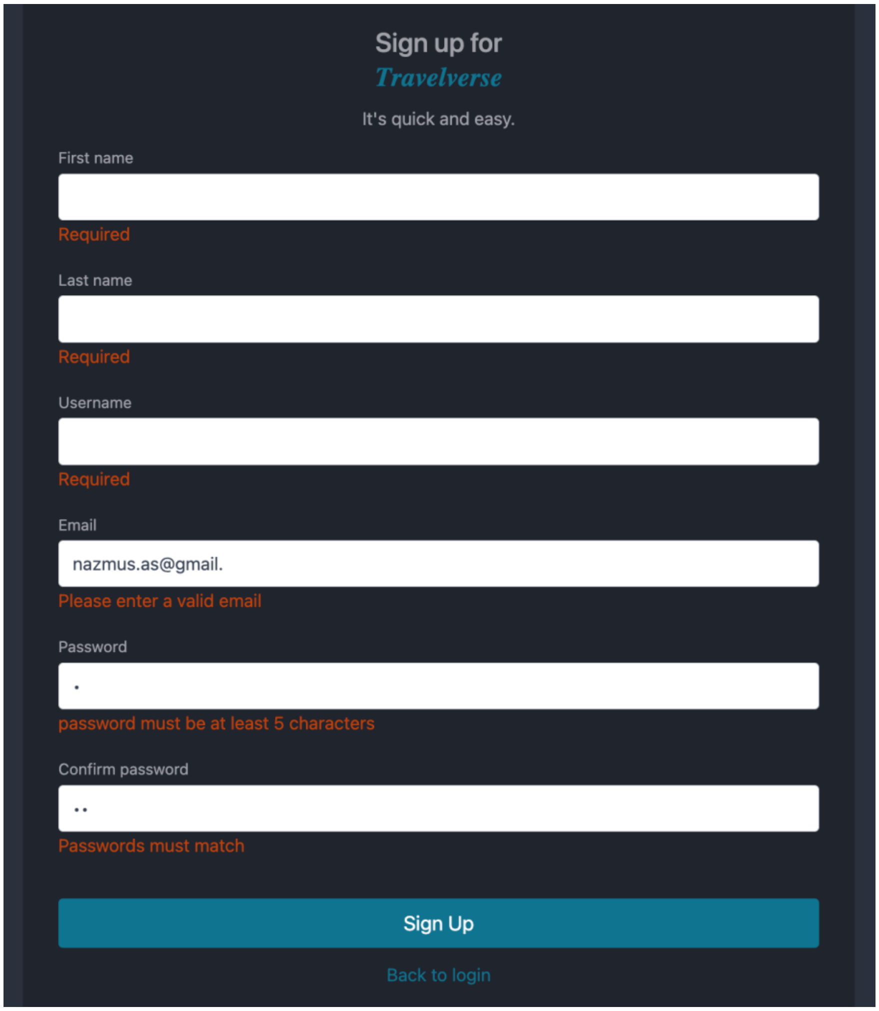
This report consists defferent test cases which are ensureing level of Quality of this Application.

Test Case ID#	**Use Case ID#**	**Use Case Name**	**Priority**
TC-01	UC01	Registration	1
TC-02	UC02	Login	1
TC-03	UC03	Forgot password	1
TC-04	UC04	User profile	1
TC-05	UC05	Follower suggestion	2
TC-06	UC06	Search for other travelers/shop	2
TC-07	UC07	Follow/Unfollow	1
TC-08	UC08	Realtime notifications	3
TC-09	UC09	Post	1
TC-10	UC10	Like/unlike	1
TC-11	UC11	Comment	1
TC-12	UC12	Share posts	2
TC-13	UC13	Messenger	2
TC-14	UC14	Themes	3
TC-15	UC15	Delete account	1
TC-16	UC16	Target post	2
TC-17	UC17	Logout	1

TC-18	UC18	Edit shop location	1
TC-19	UC19	Product	1
TC-20	UC20	Place order	1
TC-21	UC21	Shop dashboard	2
TC-22	UC22	Total users/shops	1
TC-23	UC23	Total Posts	1

Chapter 5. User manual

5.1 Registration (traveler)



Sign up for
Travelverse

It's quick and easy.

First name

Required

Last name

Required

Username

Required

Email

nazmus.as@gmail.

Please enter a valid email

Password

•

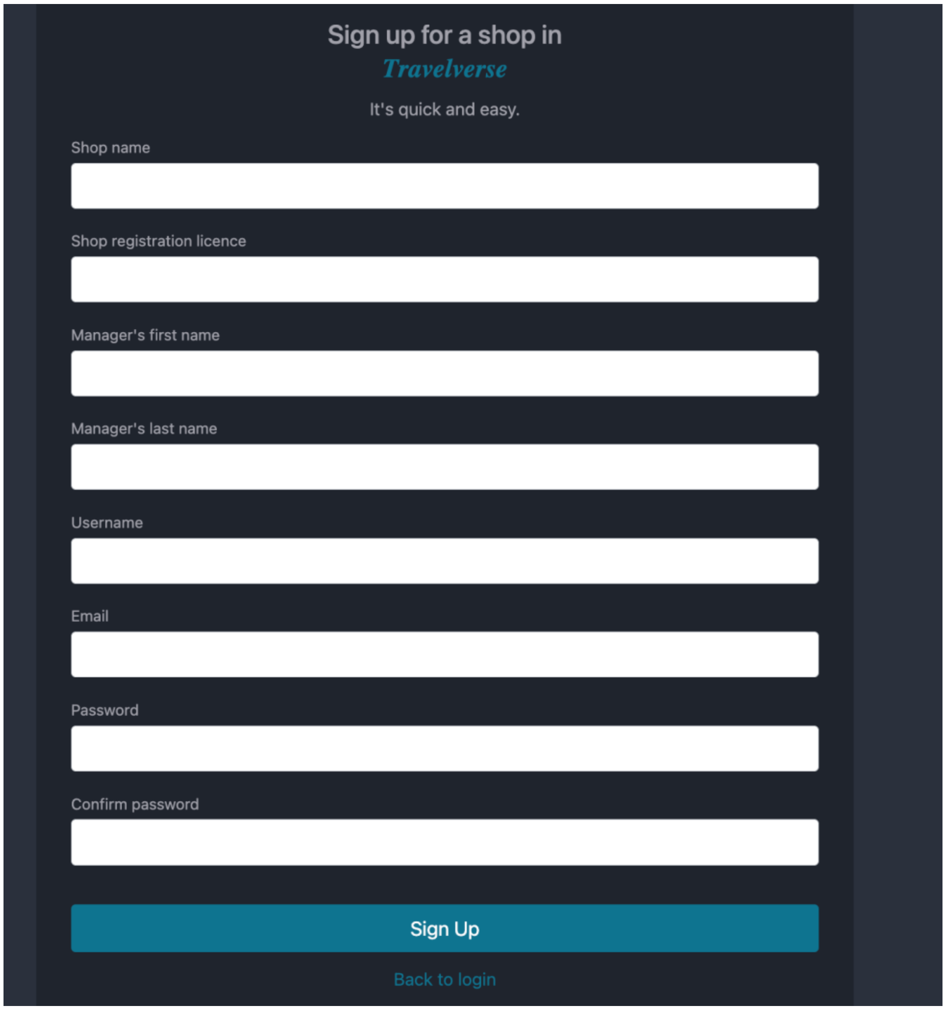
password must be at least 5 characters

Confirm password

••

Passwords must match

5.2 Registration (shop)



Sign up for a shop in

Travelverse

It's quick and easy.

Shop name

Shop registration licence

Manager's first name

Manager's last name

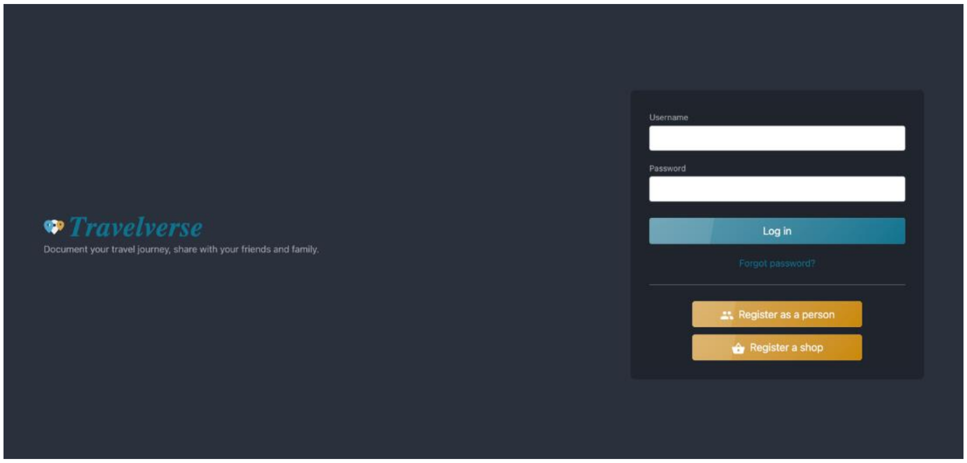
Username

Email

Password

Confirm password

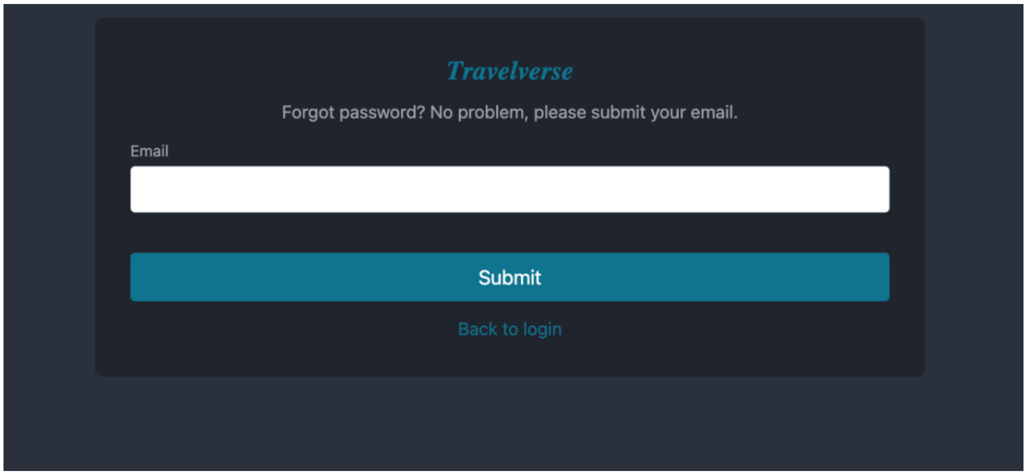
5.3 Log in

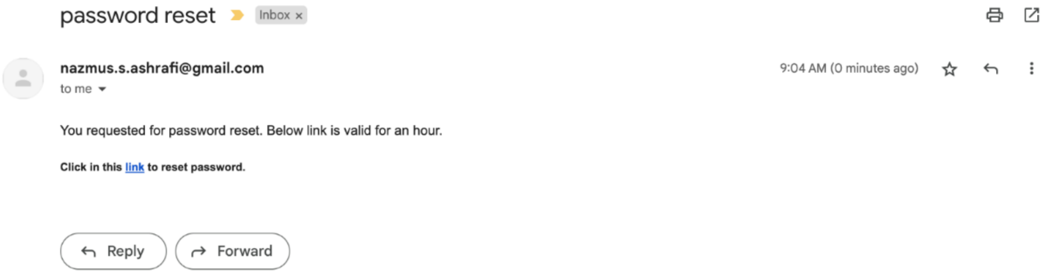


Travelverse

Document your travel journey, share with your friends and family.

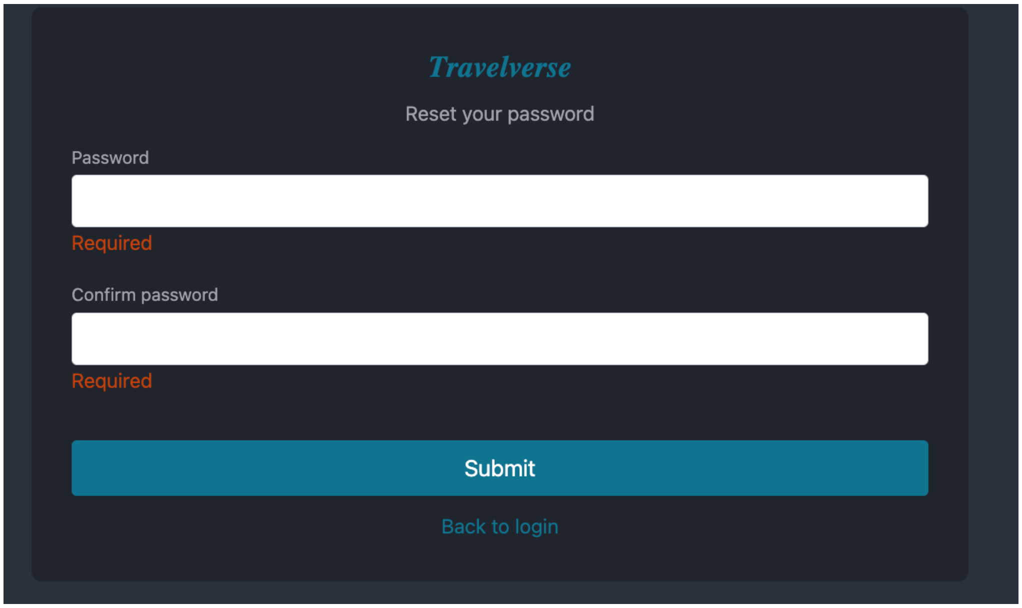
5.4 Forgot password





You requested for password reset. Below link is valid for an hour.

Click in this link to reset password.



Travelverse

Reset your password

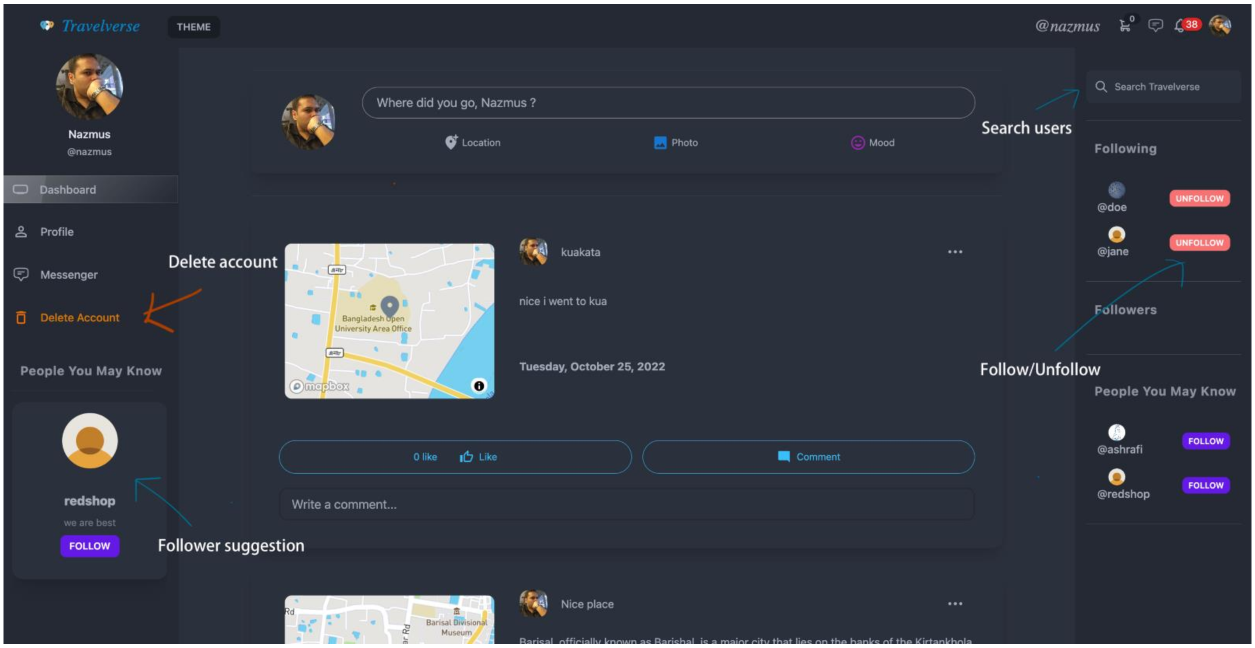
Password

Required

Confirm password

Required

5.5 Dashboard



Delete account

Search users

Follow/Unfollow

Follower suggestion

Where did you go, Nazmus ?

kuakata

nice i went to kua

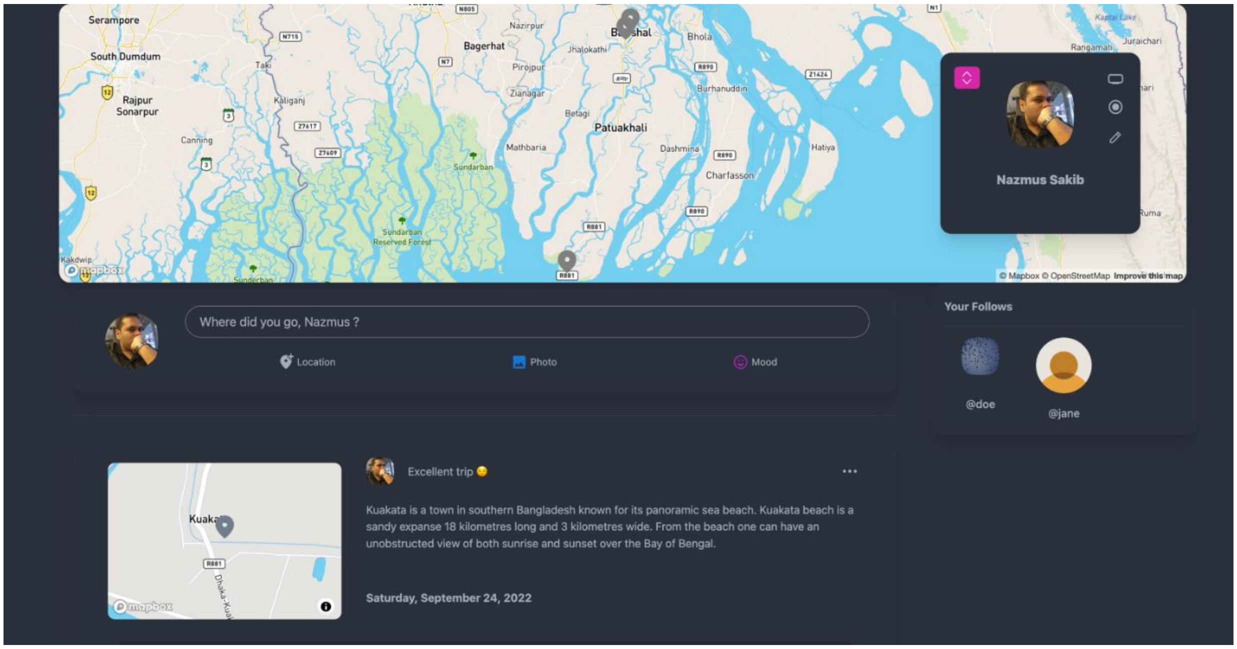
Tuesday, October 25, 2022

Write a comment...

Nice place

Barisal, officially known as Barishal, is a major city that lies on the banks of the Kirtankhola

5.6 Profile

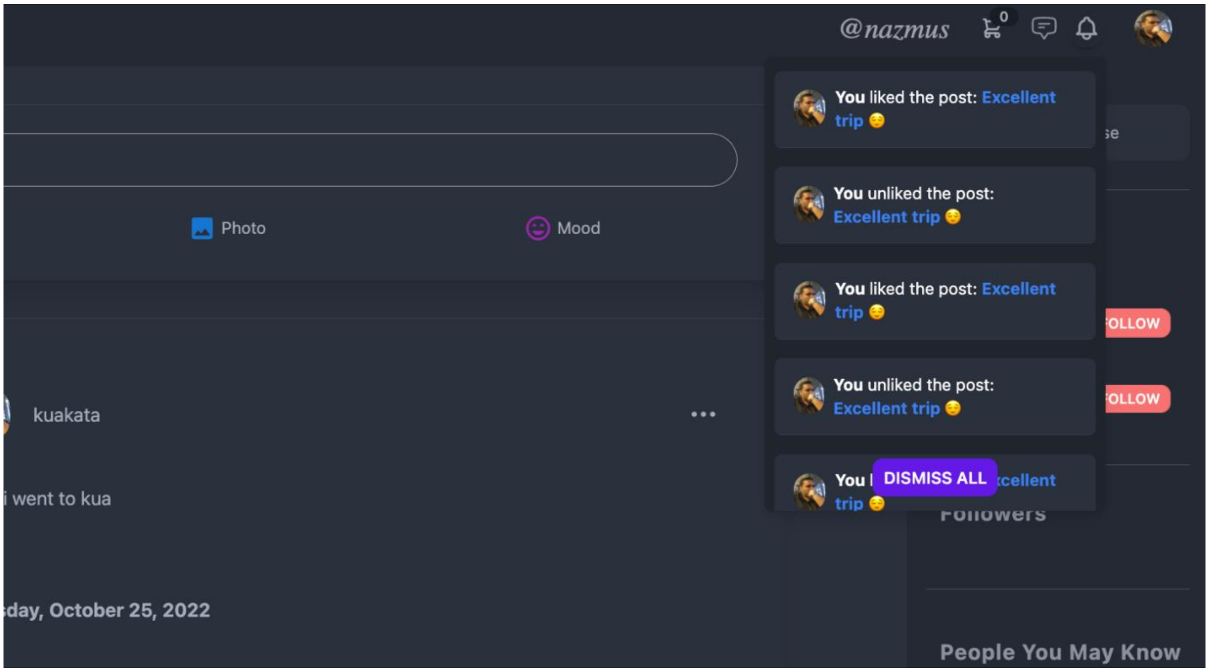


Excellent trip 😔

Kuakata is a town in southern Bangladesh known for its panoramic sea beach. Kuakata beach is a sandy expanse 18 kilometres long and 3 kilometres wide. From the beach one can have an unobstructed view of both sunrise and sunset over the Bay of Bengal.

Saturday, September 24, 2022

5.7 Live notification



You liked the post: Excellent trip 😌

You unliked the post: Excellent trip 😌

You liked the post: Excellent trip 😌

You unliked the post: Excellent trip 😌

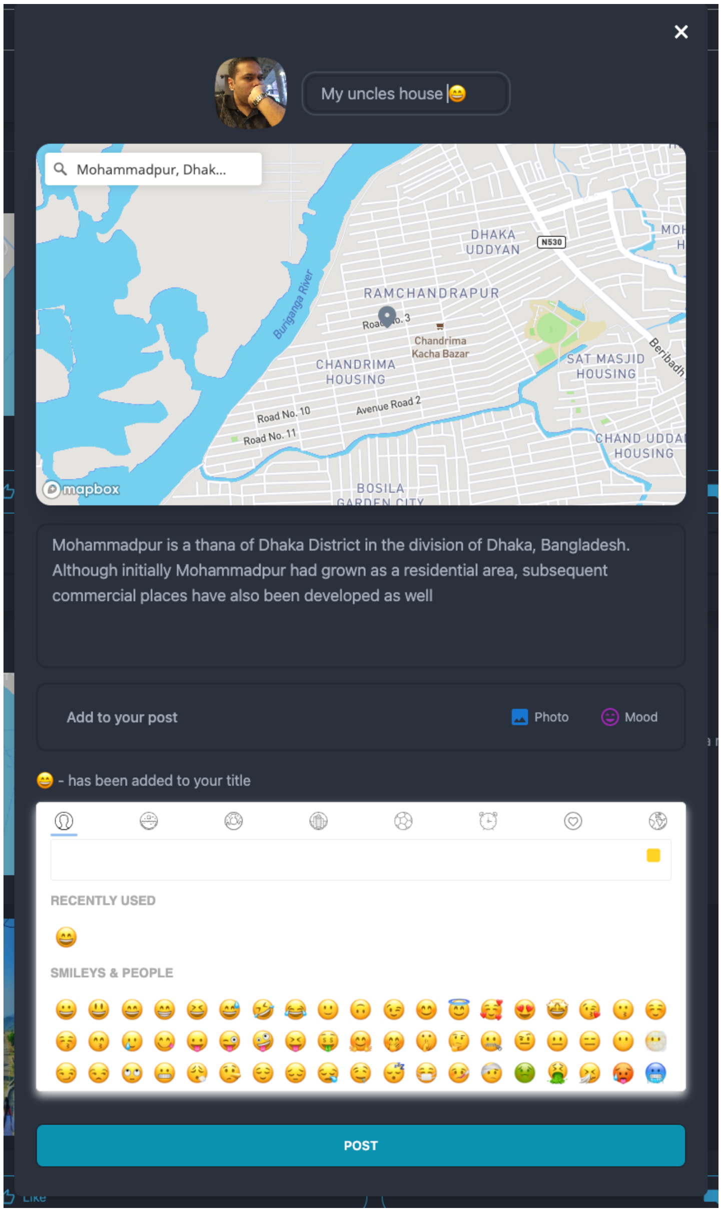
You l xcellent trip 😌

kuakata

i went to kua

day, October 25, 2022

5.8 Detailed post maker



My uncles house 😄

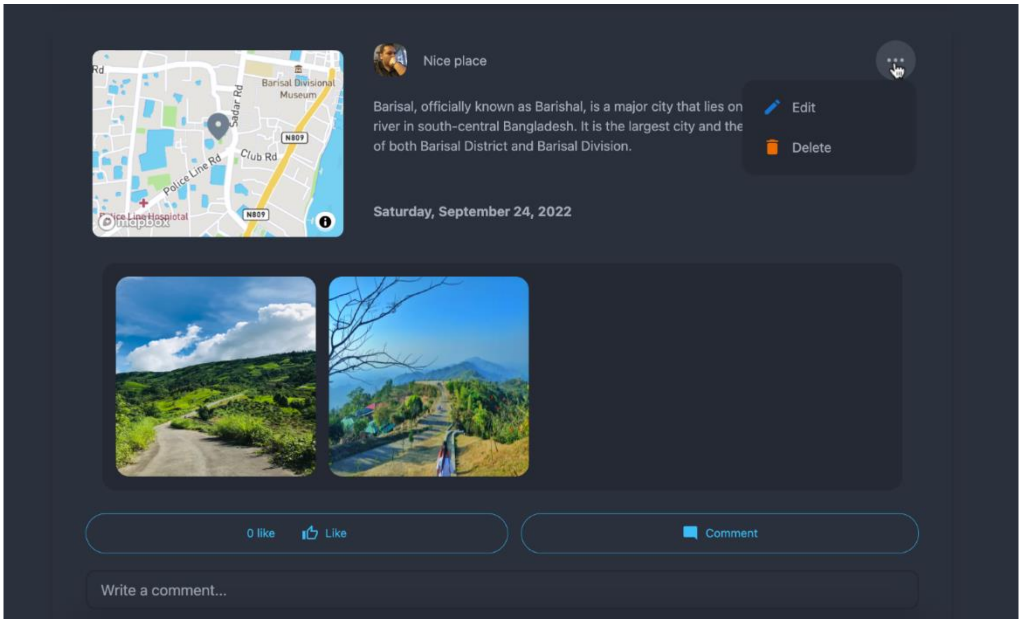
Mohammadpur, Dhak...

Mohammadpur is a thana of Dhaka District in the division of Dhaka, Bangladesh. Although initially Mohammadpur had grown as a residential area, subsequent commercial places have also been developed as well

Add to your post

😄 - has been added to your title

5.9 Post, Post Edit, Post Delete

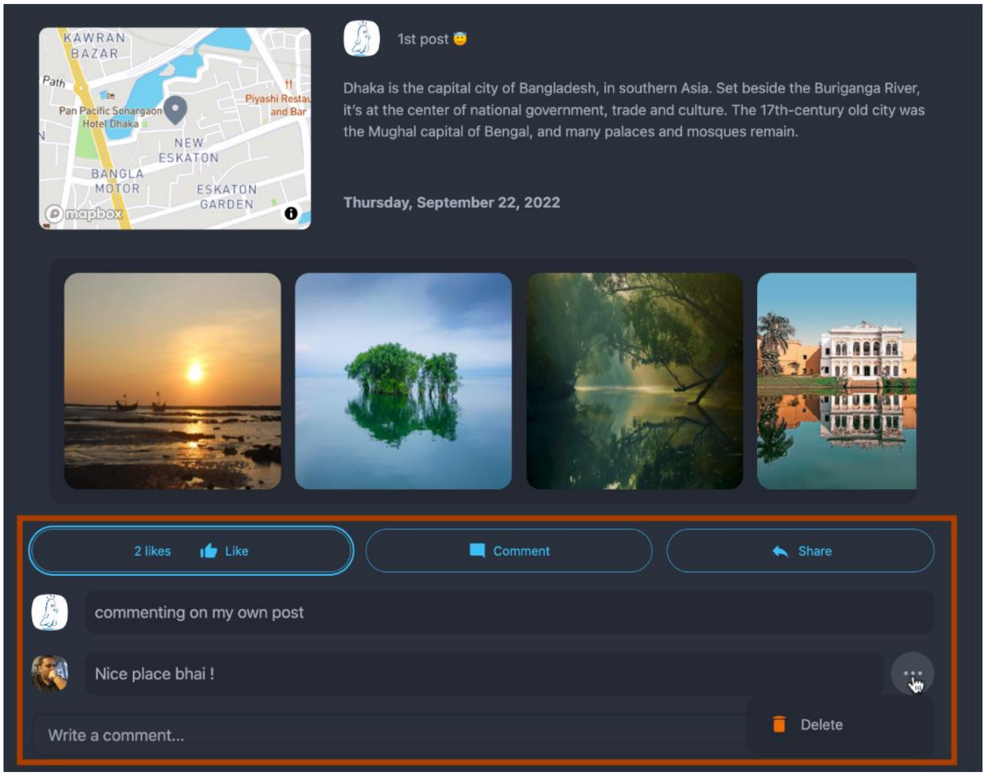


Nice place

Barisal, officially known as Barishal, is a major city that lies on
river in south-central Bangladesh. It is the largest city and the
of both Barisal District and Barisal Division.

Saturday, September 24, 2022

5.10 Comments and likes


1st post 😇

Dhaka is the capital city of Bangladesh, in southern Asia. Set beside the Buriganga River, it's at the center of national government, trade and culture. The 17th-century old city was the Mughal capital of Bengal, and many palaces and mosques remain.

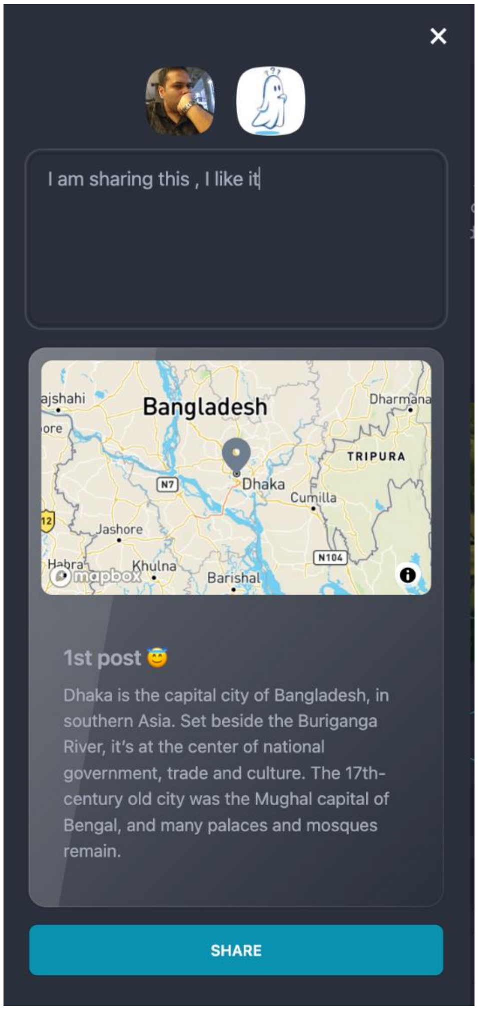
Thursday, September 22, 2022

commenting on my own post

Nice place bhai !

Write a comment...

5.11 Share post maker

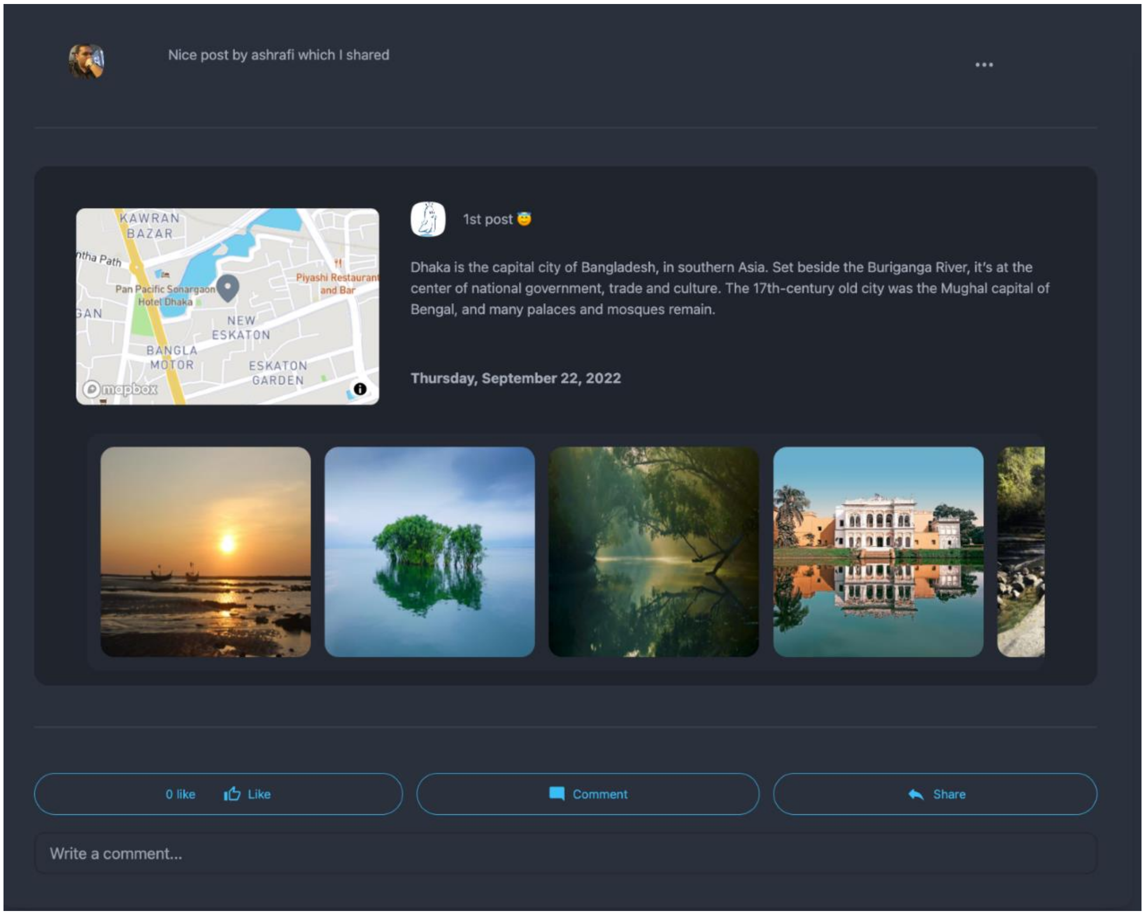


I am sharing this , I like it

1st post 😇

Dhaka is the capital city of Bangladesh, in southern Asia. Set beside the Buriganga River, it's at the center of national government, trade and culture. The 17th-century old city was the Mughal capital of Bengal, and many palaces and mosques remain.

5.12 Shared post



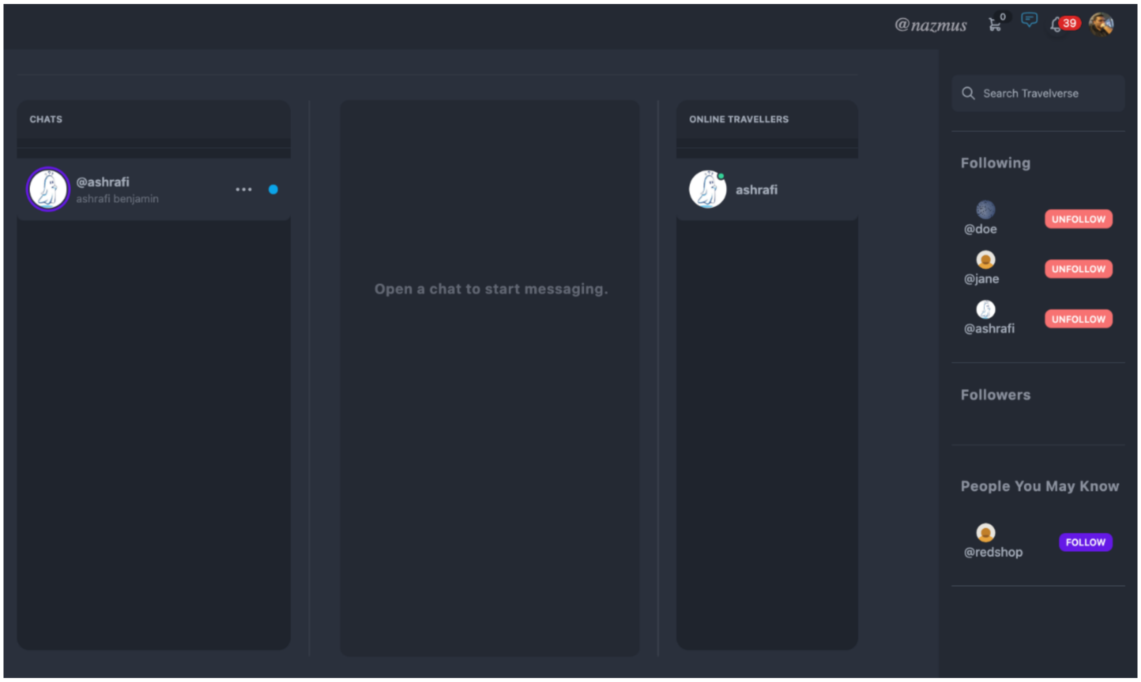
Nice post by ashrafi which I shared

1st post 😇

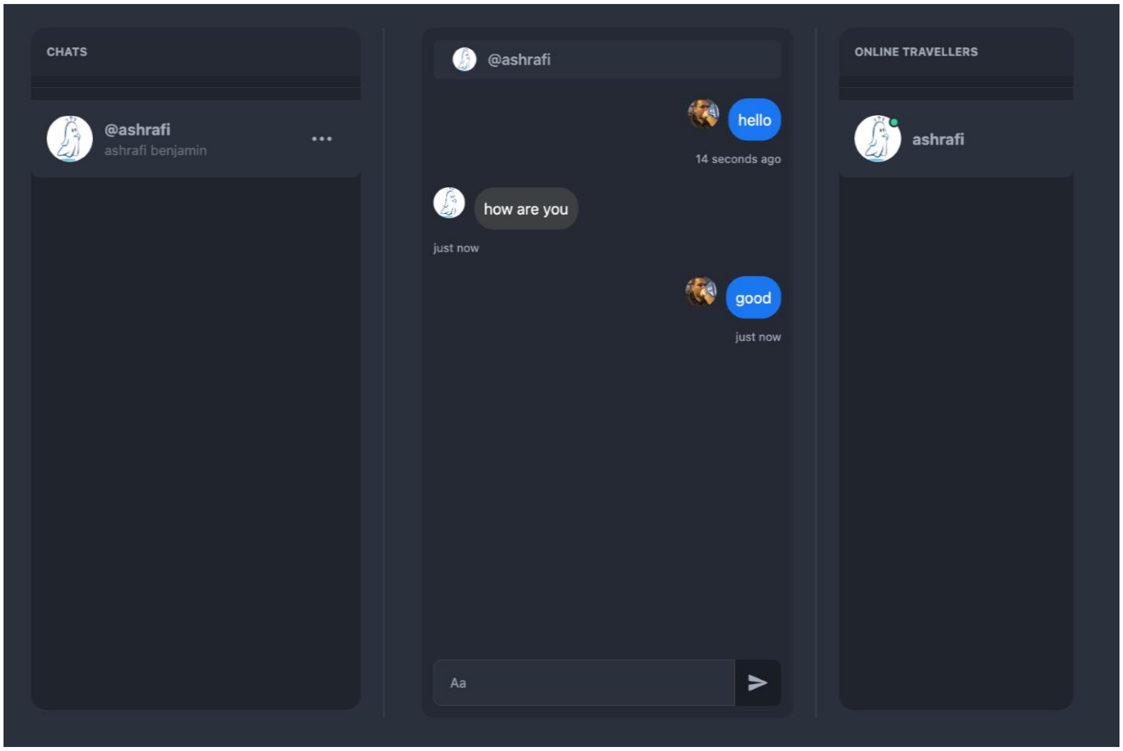
Dhaka is the capital city of Bangladesh, in southern Asia. Set beside the Buriganga River, it's at the center of national government, trade and culture. The 17th-century old city was the Mughal capital of Bengal, and many palaces and mosques remain.

Thursday, September 22, 2022

5.13 Messenger



Open a chat to start messaging.



@ashrafi

hello

14 seconds ago

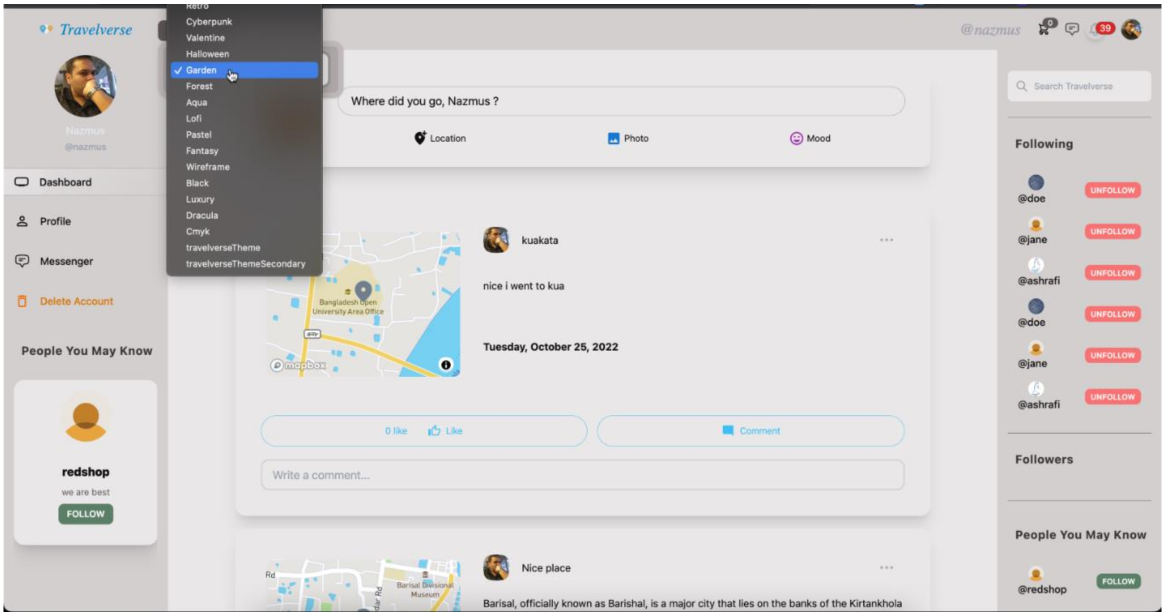
how are you

just now

good

just now

5.14 Themes



kuakata

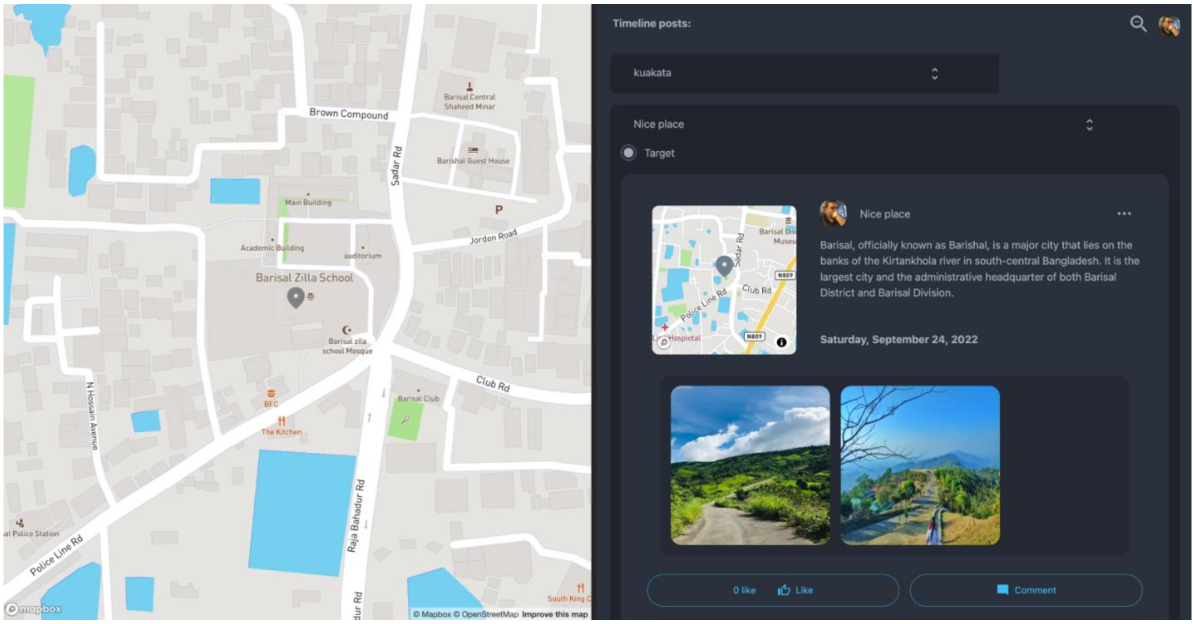
nice i went to kua

Tuesday, October 25, 2022

Nice place

Barisal, officially known as Barishal, is a major city that lies on the banks of the Kirtankhola

5.15 Target mode

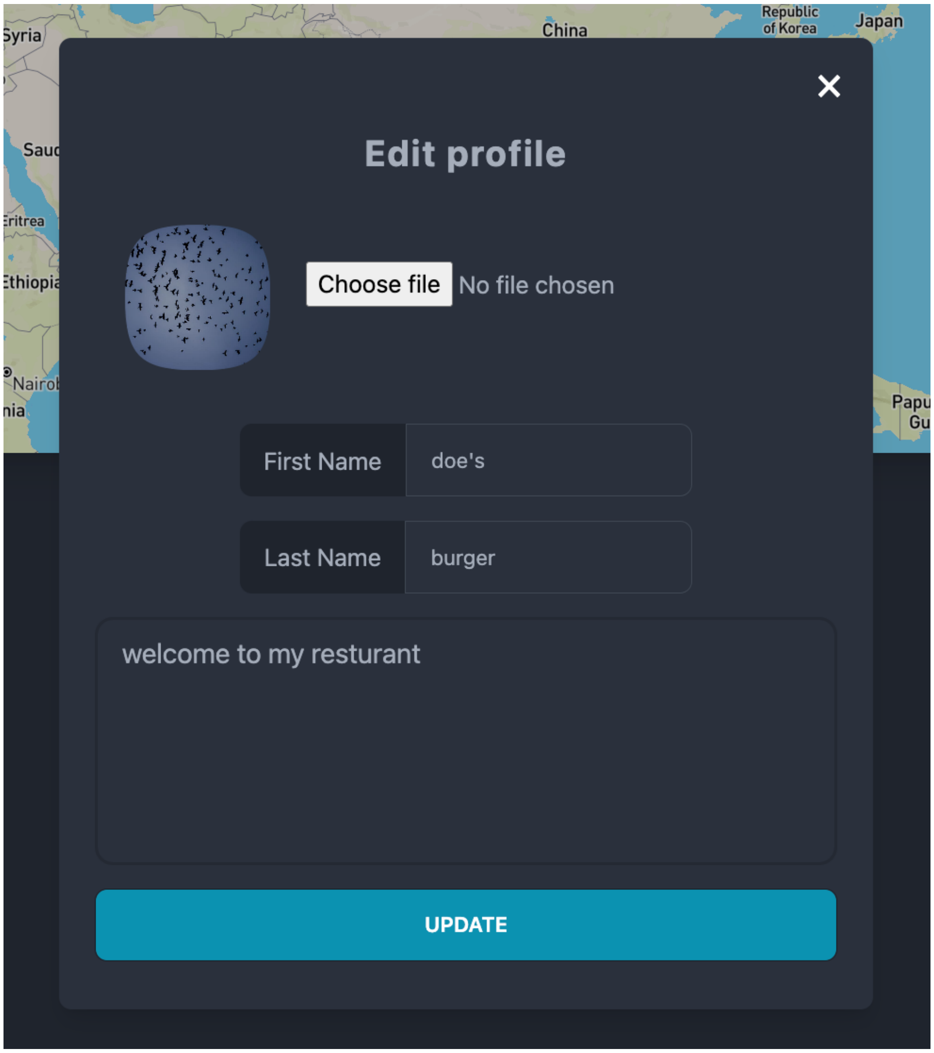


Nice place

Barisal, officially known as Barishal, is a major city that lies on the banks of the Kirtankhola river in south-central Bangladesh. It is the largest city and the administrative headquarter of both Barisal District and Barisal Division.

Saturday, September 24, 2022

5.16 Edit profile



Edit profile

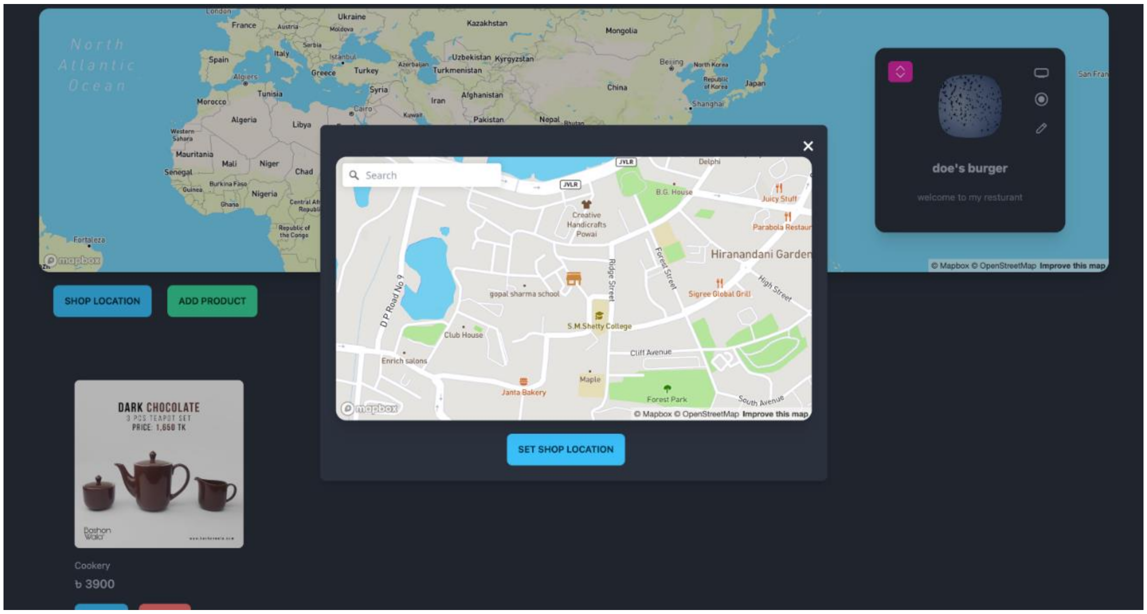
No file chosen

First Name: doe's

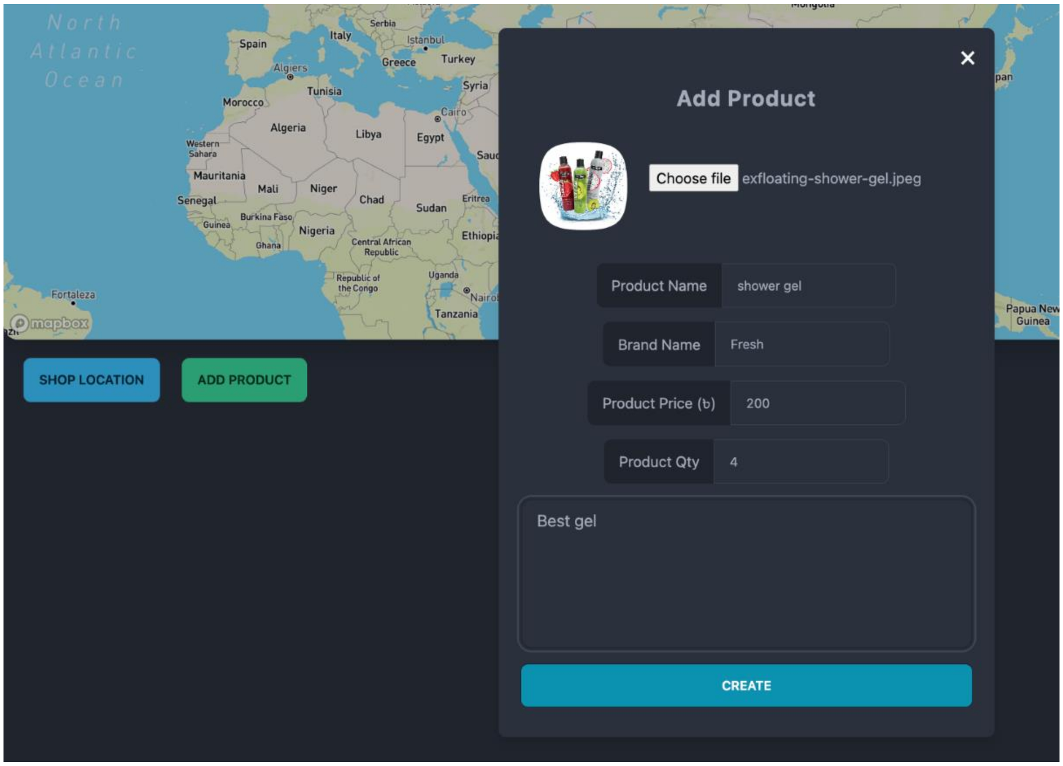
Last Name: burger

welcome to my resturant

5.17 Set shop location



5.18 Add product (seller side)



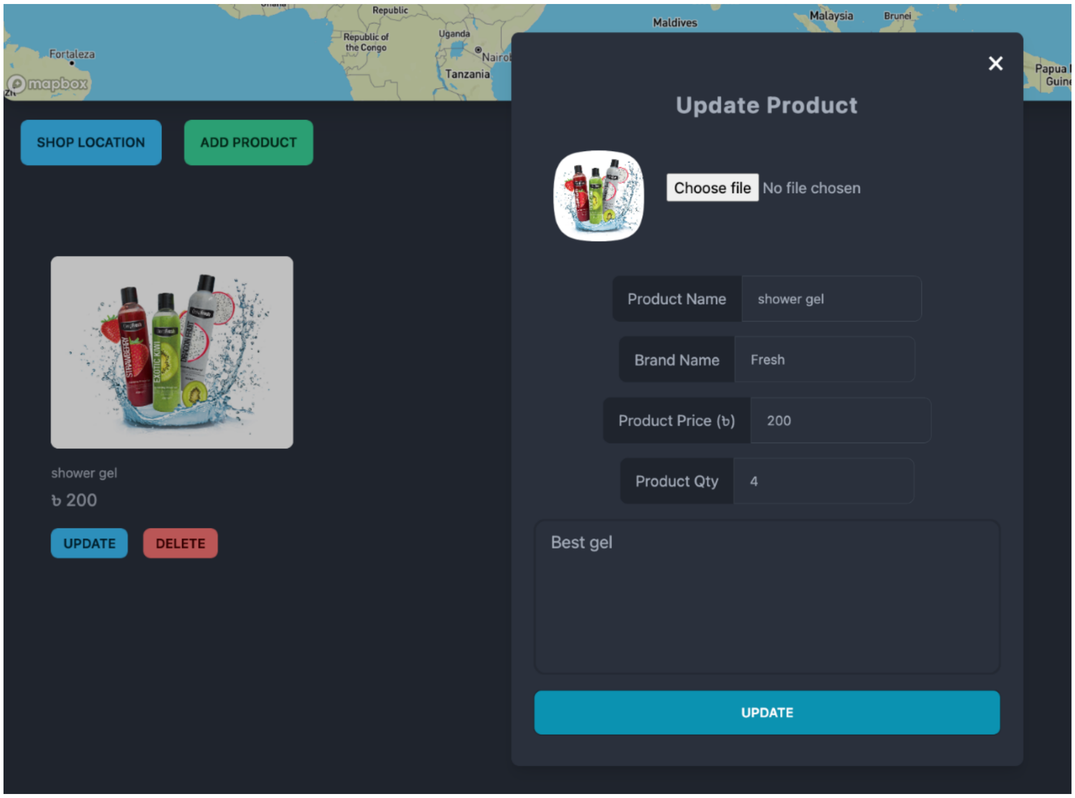
Add Product

exfloating-shower-gel.jpeg

Product Name	shower gel
Brand Name	Fresh
Product Price (৳)	200
Product Qty	4

Best gel

5.19 Update product (seller side)



shower gel

৳ 200

Update Product

No file chosen

Product Name: shower gel

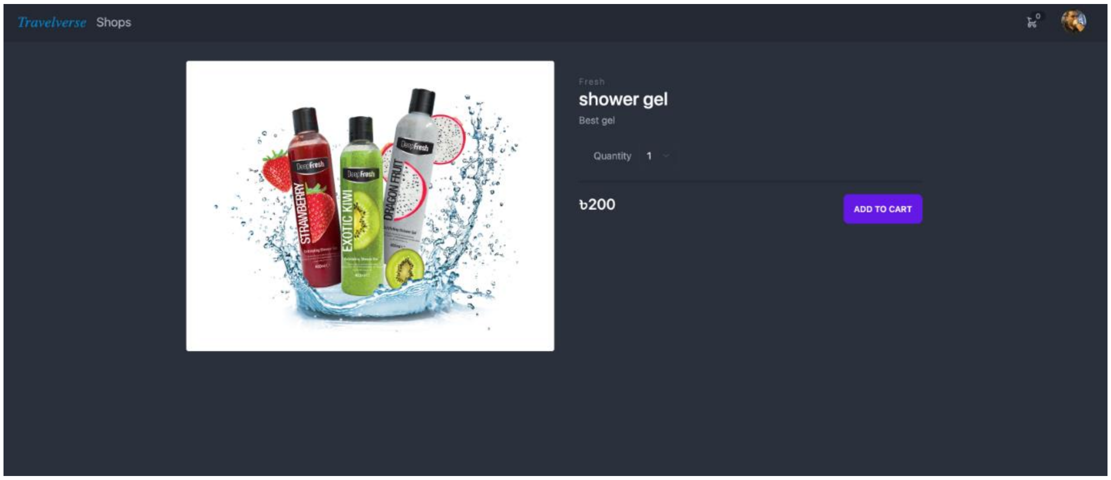
Brand Name: Fresh

Product Price (৳): 200

Product Qty: 4

Best gel

5.20 Add product to cart (customer side)



Fresh

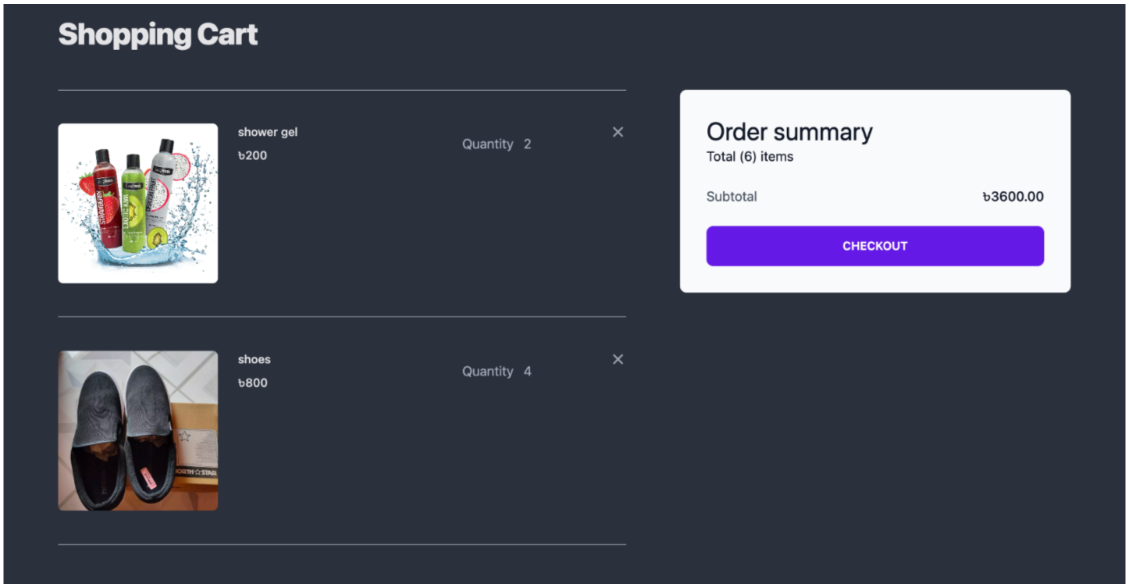
shower gel

Best gel

Quantity 1

৳200

5.21 Shopping cart (customer side)



Shopping Cart

shower gel

৳200

Quantity 2

×

shoes

৳800

Quantity 4

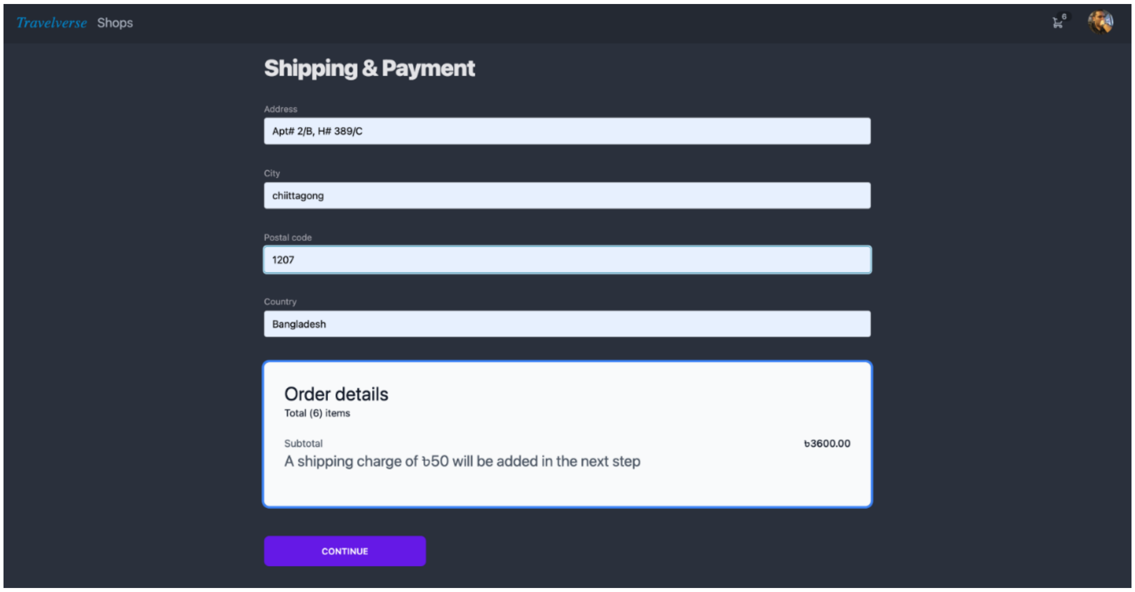
×

Order summary

Total (6) items

Subtotal ৳3600.00

5.22 Shipping and payment (customer side)



Shipping & Payment

Address
Apt# 2/B, H# 389/C

City
chiittagong

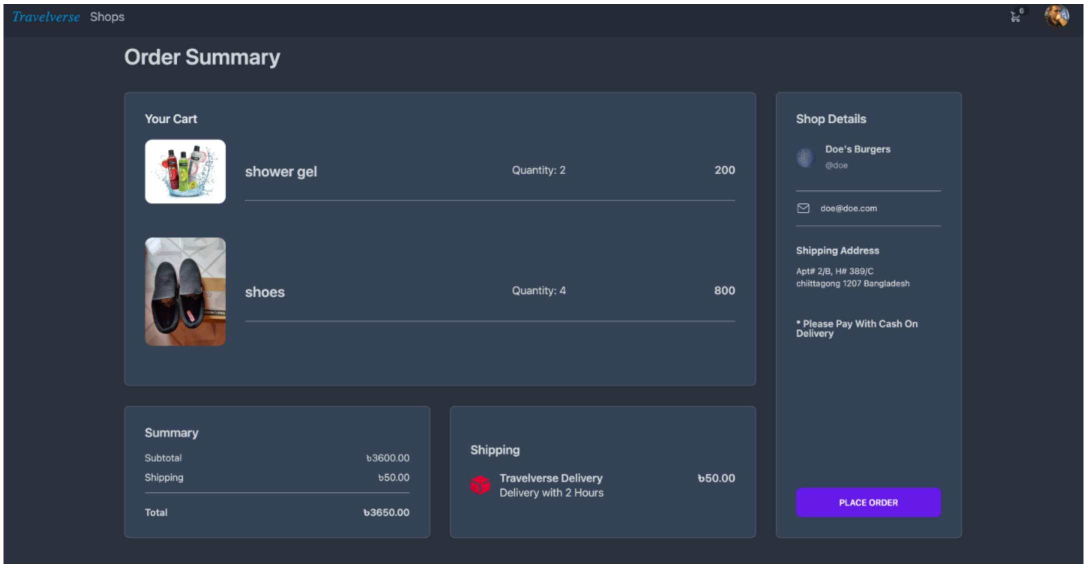
Postal code
1207

Country
Bangladesh

Order details
Total (6) items

Subtotal ৳3600.00
A shipping charge of ৳50 will be added in the next step

5.23 Order summary (customer side)



Order Summary

Your Cart

shower gel	Quantity: 2	200
shoes	Quantity: 4	800

Shop Details

Doe's Burgers
@doe

doe@doe.com

Shipping Address

Apt# 2/B, H# 389/C
chiittagong 1207 Bangladesh

* Please Pay With Cash On Delivery

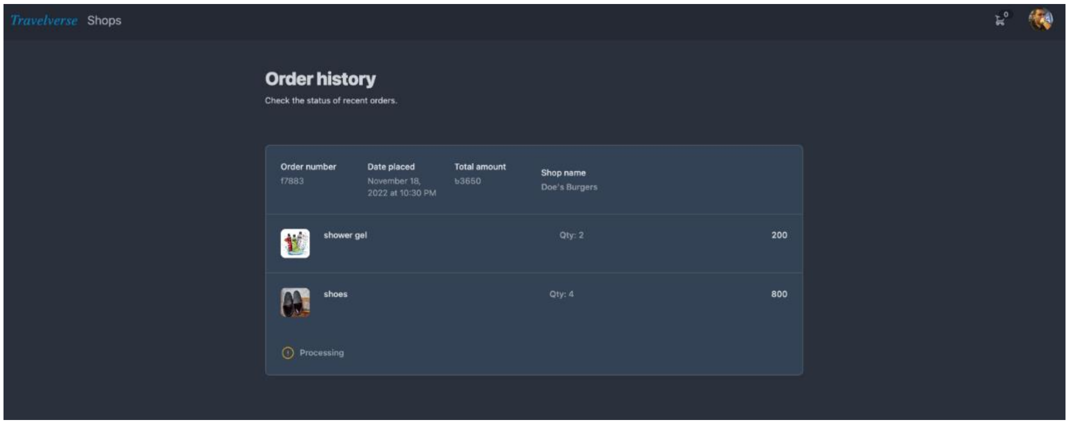
Summary

Subtotal	৳3600.00
Shipping	৳50.00
Total	৳3650.00

Shipping

Travelverse Delivery ৳50.00
Delivery with 2 Hours

5.24 Order history (customer side)



Order history

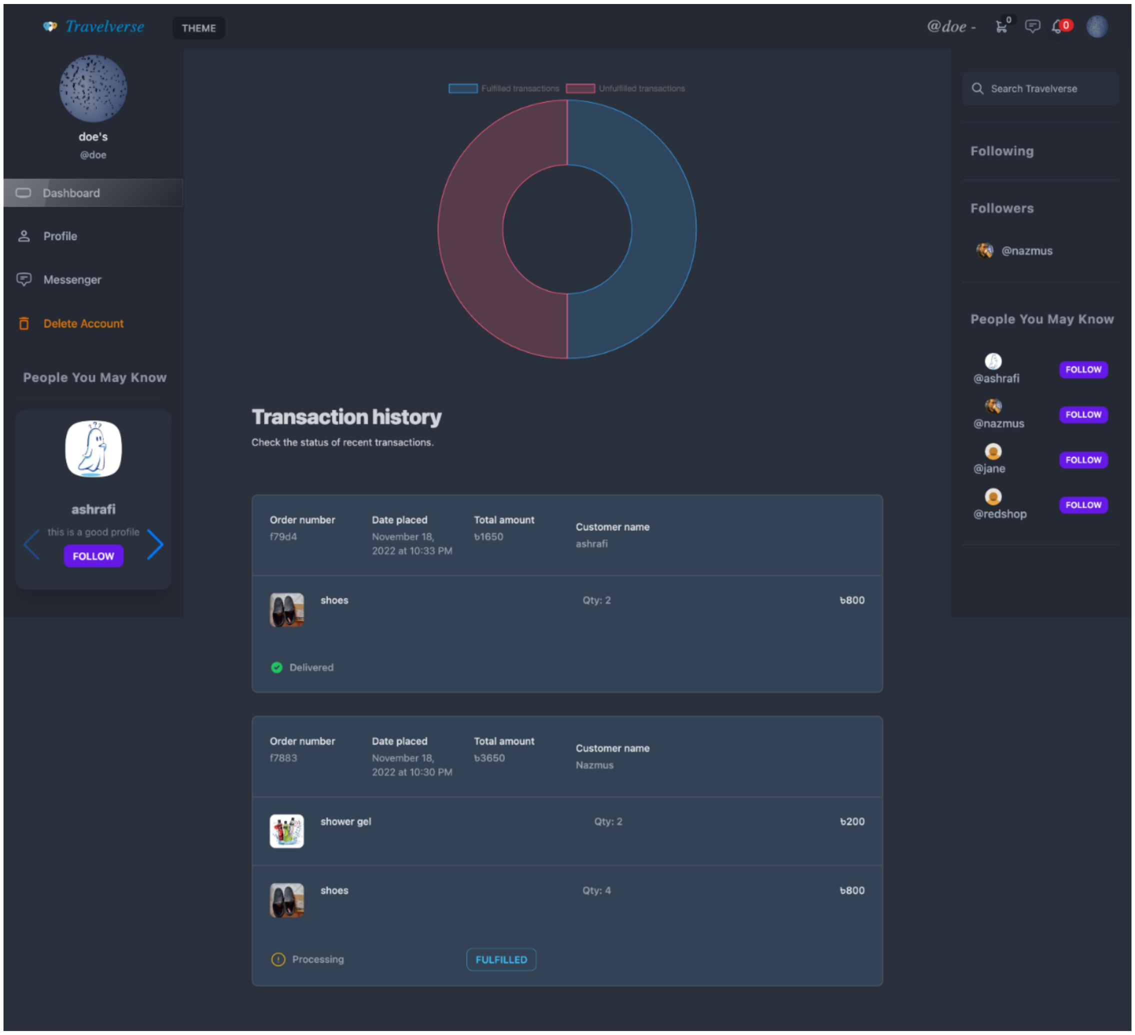
Check the status of recent orders.

Order number	Date placed	Total amount	Shop name
f7883	November 18, 2022 at 10:30 PM	৳3650	Doe's Burgers

shower gel	Qty: 2	200
shoes	Qty: 4	800

Processing

5.25 Transaction history and statistics (seller side)



Transaction history

Check the status of recent transactions.

Order number	Date placed	Total amount	Customer name
f79d4	November 18, 2022 at 10:33 PM	৳1650	ashrafi

shoes	Qty: 2	৳800

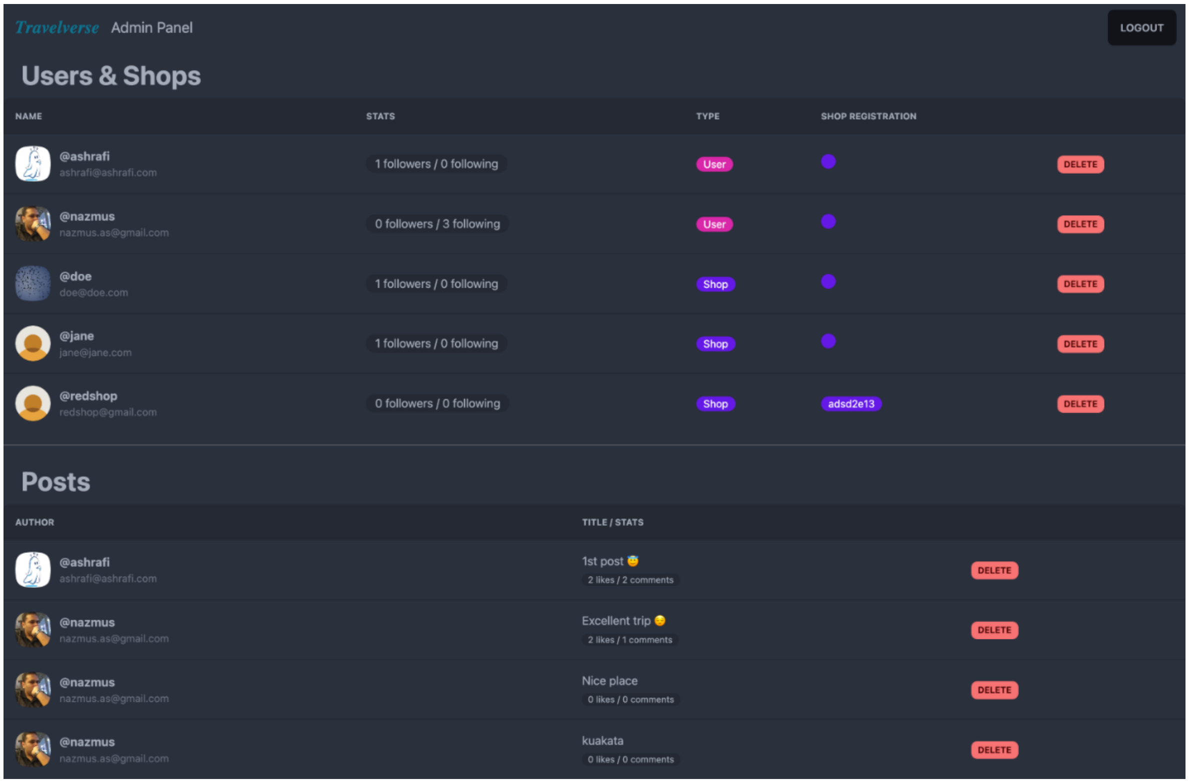
✓ Delivered

Order number	Date placed	Total amount	Customer name
f7883	November 18, 2022 at 10:30 PM	৳3650	Nazmus

shower gel	Qty: 2	৳200
shoes	Qty: 4	৳800

Processing FULFILLED

5.26 Admin dashboard (admin)



Users & Shops

NAME	STATS	TYPE	SHOP REGISTRATION	
@ashrafi ashrafi@ashrafi.com	1 followers / 0 following	User	●	DELETE
@nazmus nazmus.as@gmail.com	0 followers / 3 following	User	●	DELETE
@doe doe@doe.com	1 followers / 0 following	Shop	●	DELETE
@jane jane@jane.com	1 followers / 0 following	Shop	●	DELETE
@redshop redshop@gmail.com	0 followers / 0 following	Shop	adsd2e13	DELETE

Posts

AUTHOR	TITLE / STATS	
@ashrafi ashrafi@ashrafi.com	1st post 😇 2 likes / 2 comments	DELETE
@nazmus nazmus.as@gmail.com	Excellent trip 😔 2 likes / 1 comments	DELETE
@nazmus nazmus.as@gmail.com	Nice place 0 likes / 0 comments	DELETE
@nazmus nazmus.as@gmail.com	kuakata 0 likes / 0 comments	DELETE

Chapter 6. Project summary

Travelverse is a comprehensive application with lots of unique features and fast performance. My goal of making a unique social media is complete and hopefully people will find it easy and useful to use. The application provides a way to record travelling experiences in a unique way that is fun and can be shared with friends and family. It lets users communicate with other traveler who are like minded and also want to record their travelling experiences around the words in real time. It also creates a community of sellers and buyers and lets transactions happen in a way that is beneficial to the community growth.

6.1 GitHub link

https://github.com/nazmus-saqueeb-ashrafi/travelVerse-combined

6.2 Limitations

- Huge server size needed to support big user base
- Security can be improved by making routes more protected
- Not completely mobile responsive

6.3 Tools and Technology

Frontend technologies:

- HTML
- CSS
- React
- Redux for state management
- Tailwind CSS for design
- Socket.IO for real time events

Backend technologies:

- Node JS
- Express

Database

- Mongo Db
- Firebase (for images)

IDE

- Visual Studio Code

6.4 Future scope

- Add mobile responsiveness
- Build a mobile application based on this backend
- Add fully fleshed out ecommerce functionalities
- Add group chat in messenger
- Add ways to advertise on the platform
- Online payment integration