

## REQUIREMENTS SPECIFICATIONS

**P-10:O**DYSSEUM

STUDENT ID	NAME
25100283	MUHAMMAD AFFAN NAVED
25100225	MOHAMMAD HAROON KHAWAJA
25100212	SHAHREZ AEZAD
25100097	PIR M. SHAHRAIZ CHISHTY
25100023	LUQMAN AADIL

## TABLE OF CONTENTS

1. Introduction	3
2. System Actors	4
3. Use Cases	5
3.1 Use Case Diagrams	6
3.2 Description of Use Cases	7
3.2.1 Login	7
3.2.2 Register	8
3.2.3 Making itineraries	9
3.2.4 Posting Itineraries	10
3.2.5 Post Moderation by Admin Users	11
3.2.6 AI-Based Travel/Post Suggestions	12
3.2.7 Linking Social Media Profiles	13
3.2.8 Searching for Locations	13
3.2.9 Rewards System	14
3.2.10 Adding Friends	15
3.2.11 Making Posts	15
3.2.12 Writing a Review	16
3.2.13 Sending Messages	17
3.2.14 Update User Profile	17
3.2.15 Adding Travel Bookmarks	18
3.2.16 View Business Analytics	19
3.2.17 Multi-Factor Authentication	19
3.2.18 Push Notification	20
3.2.19 Activity Feed	21
3.2.20 Report Content	21
4. Class Diagram	23
4.1 Diagram	23
4.2 Description	24
1. Users	24
2. Itinerary	24
3. Bookmark	24

4. Chat	25
5. Posts	25
6. Comments	25
7. Location	25
8. Review	25
9. Business	25
5. Sequence Diagrams	25
5.1 Login	26
5.2 Register	26
5.3 Making Itineraries	26
5.4 Posting Itineraries	26
5.5 Post Moderation by Admin Users	26
5.6 AI-Based Travel/Post Suggestions	26
5.7 Linking Social Media Profiles	26
5.8 Searching for Locations	26
5.9 Rewards System	26
5.10 Adding Friends	26
5.11 Making Posts	26
5.12 Writing a Review	26
5.13 Sending Messages	26
5.14 Update User Profile	26
5.15 Adding Travel Bookmarks	26
5.16 View Business Analytics	26
5.17 Multi-Factor Authentication	26
5.18 Push Notification	26
5.19 Activity Feed	26
5.20 Report Content	26
6. State Diagrams	27
6.1 Diagram details	27
6.2 Diagram	27
7. Non-functional Requirements / Quality Attributes	28
8. Who Did What?	28
9. Review checklist	29

## **1. Introduction**

This project aims to develop a travel/social network application to help travelers plan their next trip using just one app. Rather than relying on blogs and pages from different online outlets, the app would be a one-stop solution for all travelers. The application will provide users a platform to search for various tourist destinations they may be interested in visiting and what these destinations have to offer such as accommodation, sightseeing, dining, nightlife, historical sites, and tour guides. Combining all these services onto one platform would improve the travel experience and will allow users to make well informed decisions based on information on the destinations..

Nowadays, travelers face a fundamental problem: finding accurate and relevant information. They have to rely on large commercial travel agencies that only have profit-driven goals or on personal connections that provide limited details and advice. This gives travelers an experience far from fulfilling, while local businesses gain limited benefits. The purpose of this app is to serve as a networking app to connect like-minded travelers and local service providers such that both parties benefit, with travelers having a fulfilling experience visiting their destinations and the local business being given an opportunity to boost the economic growth in the region.

As stated above, potential users of this app include travelers themselves, administrators and local businesses which are but not limited to hotels, restaurants and tour guides.

## 2. System Actors

Actor Name	Description
Tourist	The primary user of the application who can create itineraries , explore places, and connect with other travelers and local service providers. They can view recommendations, rate locations, leave reviews, and share their travel experiences with the community.
Business Owner	Businesses such as hotels, restaurants, shopping centers etc. They use the app to promote their services and engage with travelers by providing information, answering inquiries, and receiving feedback.
Tour Guides	Government or private tourism bodies that promote regions, cultural heritage, and local attractions. Similar to business owners they promote services and provide information regarding them to interested tourists.
Administrator	Responsible for managing and maintaining the platform. This actor oversees user management, content moderation, and system performance to ensure smooth operation.

### 3. Use Cases

#### 3.1 Use Case Diagrams



## 3.2 Description of Use Cases

### 3.2.1 Login

<b>Identifier</b>	UC-001
<b>Purpose</b>	The user enters their credentials to log in to access the app functionalities.
<b>Pre-conditions</b>	Users must have registered an account.
<b>Post-conditions</b>	The user is logged in and navigated to their general feed.
<b>Step #</b>	<b>Typical Course of Action</b>
1.	The user opens the app and is presented with the login screen.
2.	The user enters their email/username and password.
3.	The backend system receives these credentials.
4.	The server verifies the credentials with the database.
5.	If credentials are valid, the system will authenticate the user.
6.	The user will be redirected to their main feed.
<b>Step #</b>	<b>Alternate Courses of Action</b>
<b>Step #</b>	<b>Exception Paths</b>
1.	User enters invalid username or password.
2.	System will deny access to the user and ask to try again.

### 3.2.2 Register

<b>Identifier</b>	UC-002
<b>Purpose</b>	User needs an account to access app functionalities.
<b>Pre-conditions</b>	App must be installed on the user's phone.
<b>Post-conditions</b>	Account created and user can now login.
<b>Step #</b>	<b>Typical Course of Action</b>
1.	User opens the app and selects the button to register an account.
2.	Users enter their name, email, username, password and submit it to the system.
3.	System verifies the form data and if correct, sends users a verification email to confirm the account.
4.	Once email is verified, the user can now log in and is prompted for location preferences for personalized recommendations.
5.	Users can now access the entire app.
<b>Step #</b>	<b>Alternate Courses of Action</b>
<b>Step #</b>	<b>Exception Paths</b>
1.	User supplied form data is incorrect or invalid.
2.	System sends the user an error message and asks them to input data again.



### 3.2.3 Making itineraries

<b>Identifier</b>	UC-003
<b>Purpose</b>	Allow users to create and save custom travel itineraries, including destinations, activities, accommodations, and dates.
<b>Pre-conditions</b>	User is logged into the app.
<b>Post-conditions</b>	Travel itinerary is saved to the user's account. Itinerary can be shared with others or accessed for further updates.
<b>Step #</b>	<b>Typical Course of Action</b>
1.	User navigates to the "Create Itinerary" section.
2.	User selects destinations, dates, and activities.
3.	User adds accommodations, restaurants, and other points of interest.
4.	User reviews and saves the itinerary.
5.	System confirms the itinerary is saved successfully and provides sharing options.
<b>Step #</b>	<b>Alternate Courses of Action</b>
1.	
<b>Step #</b>	<b>Exception Paths</b>
	If the internet connection is lost, the system notifies the user and saves progress locally, resuming once connection is restored.
	If invalid data is entered (e.g., conflicting dates), an error message is displayed, and the user is prompted to correct the information.

### 3.2.4 Posting Itineraries

<b>Identifier</b>	UC-004
<b>Purpose</b>	Allow users to share their travel itineraries from their itinerary tab, making them visible to other users for viewing and feedback.
<b>Pre-conditions</b>	User is logged into the app. User has at least one saved itinerary.
<b>Post-conditions</b>	Itinerary is posted and visible to other users. Other users can view, like, or comment on the shared itinerary.
<b>Step #</b>	<b>Typical Course of Action</b>
1.	User navigates to the "Itineraries" tab.
2.	User selects an itinerary to share.
3.	User adds optional descriptions or tags.
4.	User confirms sharing.
5.	System posts the itinerary to the feed or designated audience.
6.	System notifies the user of successful sharing
<b>Step #</b>	<b>Alternate Courses of Action</b>
1.	User can make their itinerary private again.
<b>Step #</b>	<b>Exception Paths</b>
1.	If there is a network issue, the system notifies the user that sharing failed, and the action is queued to retry when the connection is restored.

### 3.2.5 Post Moderation by Admin Users

<b>Identifier</b>	UC-005
<b>Purpose</b>	Allow admins to review and moderate user posts for inappropriate content, spam, or violations of community guidelines.
<b>Pre-conditions</b>	<ol style="list-style-type: none"><li>1. Admin is logged into the admin panel.</li><li>2. A user has posted content that needs moderation.</li></ol>
<b>Post-conditions</b>	<ol style="list-style-type: none"><li>1. Post is either approved or flagged for removal/violations.</li><li>2. If removed, the user is notified with the reason</li></ol>
<b>Step #</b>	<b>Typical Course of Action</b>
1.	Admin navigates to the moderation panel.
2.	Admin reviews flagged posts.
3.	Admin selects the action: approve, warn, or remove.
4.	System applies the selected action.
5.	Admin documents the action if necessary and notifies the user of the action.
<b>Step #</b>	<b>Alternate Courses of Action</b>
<b>Step #</b>	<b>Exception Paths</b>
	If the post was flagged incorrectly, the admin can mark it as safe, and it is restored.

### 3.2.6 AI-Based Travel/Post Suggestions

<b>Identifier</b>	UC-006
<b>Purpose</b>	Provide personalized travel suggestions based on user preferences, past trips, and reviews.
<b>Pre-conditions</b>	User is logged in.
<b>Post-conditions</b>	User receives a list of suggested travel destinations, activities, and accommodations and their main feed also shows posts related to these recommendations.
<b>Step #</b>	<b>Typical Course of Action</b>
1.	User logs in and is taken to their main feed.
2.	User goes to their explore page.
3.	AI backend analyzes user viewing history and preferences.
4.	System generates suggestions and displays them to the user.
<b>Step #</b>	<b>Alternate Courses of Action</b>
<b>Step #</b>	<b>Exception Paths</b>
	If the AI engine lacks sufficient data, generic travel suggestions are provided.

### 3.2.7 Linking Social Media Profiles

<b>Identifier</b>	UC-007
<b>Purpose</b>	Allow users to link their external social media accounts to the app for sharing content or synchronizing profiles.
<b>Pre-conditions</b>	<ol style="list-style-type: none"> <li>1. User is logged into the app.</li> <li>2. User has external social media accounts they wish to link.</li> </ol>
<b>Post-conditions</b>	<ol style="list-style-type: none"> <li>1. Social media profiles are successfully linked.</li> <li>2. User can share content directly from the app to their linked social media accounts</li> </ol>
<b>Step #</b>	<b>Typical Course of Action</b>

1.	User navigates to "Settings" > "Link Social Profiles".
2.	User selects the platform to link (e.g., Facebook, Tiktok).
3.	User authenticates with the external platform.
4.	System confirms the profile is linked.
<b>Step #</b>	<b>Alternate Courses of Action</b>
	User can choose to cancel the process and not link any social media account.
<b>Step #</b>	<b>Exception Paths</b>
	If the authentication fails, the system prompts the user to retry.

### 3.2.8 Searching for Locations

<b>Identifier</b>	UC-008
<b>Purpose</b>	Allow users to search for travel destinations, restaurants, hotels, and points of interest.
<b>Pre-conditions</b>	User is logged into the app.
<b>Post-conditions</b>	Search results are displayed based on user input. User can select a location for more details or add it to an itinerary.
<b>Step #</b>	<b>Typical Course of Action</b>
1.	User enters a location or keyword in the search bar.
2.	System retrieves and displays results.
3.	User selects a location for more information.
4.	System displays location details
<b>Step #</b>	<b>Alternate Courses of Action</b>
	User can apply filters (e.g., price, distance) to refine results.
<b>Step #</b>	<b>Exception Paths</b>
	If no results are found, the system displays "Not found".

### 3.2.9 Rewards System

<b>Identifier</b>	UC-009
<b>Purpose</b>	Encourage user engagement by offering rewards for activities like posting reviews, adding friends, and traveling.
<b>Pre-conditions</b>	<ol style="list-style-type: none"> <li>1. User is logged into the app.</li> <li>2. User performs reward-eligible actions (e.g., writing reviews, visiting locations).</li> </ol>
<b>Post-conditions</b>	<ol style="list-style-type: none"> <li>1. Rewards are earned and credited to the user's account.</li> <li>2. User can redeem rewards for discounts, badges, or other benefits.</li> </ol>
<b>Step #</b>	<b>Typical Course of Action</b>
1.	User completes an eligible action (e.g., post, review, etc.).
2.	System credits the user with reward points based on user app usage.
3.	User views their reward points balance.
4.	User redeems points for rewards
<b>Step #</b>	<b>Alternate Courses of Action</b>
<b>Step #</b>	<b>Exception Paths</b>
	If a reward redemption fails, the system prompts the user to retry.

### 3.2.10 Adding Friends

<b>Identifier</b>	UC-010
<b>Purpose</b>	Allow users to connect with friends or other travelers on the platform.
<b>Pre-conditions</b>	User is logged into the app.
<b>Post-conditions</b>	User can see their followings posts and other public credentials.
<b>Step #</b>	<b>Typical Course of Action</b>
1.	User searches for another user by name or profile.
2.	User clicks on the follow button to follow them.

3.	System receives this action and updates the database accordingly.
<b>Step #</b>	<b>Alternate Courses of Action</b>
	User can press unfollow if they do not wish to see their posts anymore.
<b>Step #</b>	<b>Exception Paths</b>
	If the user has blocked the other user, the system prevents following them.

### 3.2.11 Making Posts

<b>Identifier</b>	UC-011
<b>Purpose</b>	Allow users to share posts about their travels, itineraries, or reviews.
<b>Pre-conditions</b>	User is logged into the app.
<b>Post-conditions</b>	<ol style="list-style-type: none"> <li>1. Post is created and shared with the user's followers.</li> <li>2. Post appears in the user's feed and others can interact with it.</li> </ol>
<b>Step #</b>	<b>Typical Course of Action</b>
1.	User navigates to the post creation section.
2.	User writes a post and adds media (photos, videos, etc.).
3.	User selects visibility (public, friends, etc.).
4.	System posts the content to the user's feed.
<b>Step #</b>	<b>Alternate Courses of Action</b>
	User can remove a post they have already made.
<b>Step #</b>	<b>Exception Paths</b>

### 3.2.12 Writing a Review

<b>Identifier</b>	UC-012
<b>Purpose</b>	Allow users to write and post reviews for destinations, restaurants, and accommodations.
<b>Pre-conditions</b>	<ol style="list-style-type: none"> <li>1. User is logged into the app.</li> <li>2. User has visited or interacted with the location being reviewed.</li> </ol>
<b>Post-conditions</b>	<ol style="list-style-type: none"> <li>1. Review is posted publicly or to selected audiences.</li> <li>2. Other users can view and rate the review.</li> <li>3. Businesses can now view the user reviews.</li> </ol>
<b>Step #</b>	<b>Typical Course of Action</b>
1.	User navigates to the review section of a location or a business.
2.	User writes a review and rates the location or business..
3.	User submits the review.
4.	System posts the review to the location page and user's activity feed.
<b>Step #</b>	<b>Alternate Courses of Action</b>
	User can delete or edit their review.
<b>Step #</b>	<b>Exception Paths</b>
	If the user doesn't complete all fields (e.g., rating), the system prompts them to finish before submitting.

### 3.2.13 Sending Messages

<b>Identifier</b>	UC-013
<b>Purpose</b>	Allow users to send direct messages to other users or group chats.
<b>Pre-conditions</b>	User is logged into the app.
<b>Post-conditions</b>	<ol style="list-style-type: none"> <li>1. Messages are sent and received.</li> <li>2. Chat history is updated in real-time.</li> </ol>
<b>Step #</b>	<b>Typical Course of Action</b>
1.	User navigates to the messaging section.



2.	User selects a contact or group.
3.	User composes and sends a message with any media if they want to.
4.	System delivers the message in real-time.
5.	Recipient(s) receives a notification.
<b>Step #</b>	<b>Alternate Courses of Action</b>
	User can edit or delete the message.
<b>Step #</b>	<b>Exception Paths</b>
	If the user has blocked the other user, the system prevents sending messages.

### 3.2.14 Update User Profile

<b>Identifier</b>	UC-014
<b>Purpose</b>	Allow users to update their personal information, preferences, and travel interests.
<b>Pre-conditions</b>	User is logged into the app.
<b>Post-conditions</b>	User profile is updated with the new information.
<b>Step #</b>	<b>Typical Course of Action</b>
1.	User navigates to "Profile Settings".
2.	User edits personal information (e.g., bio, interests).
3.	User saves changes.
4.	System updates the profile and confirms the changes.
<b>Step #</b>	<b>Alternate Courses of Action</b>
	User can cancel the profile update.
<b>Step #</b>	<b>Exception Paths</b>

### 3.2.15 Adding Travel Bookmarks

<b>Identifier</b>	UC-015
<b>Purpose</b>	Allow users to save or bookmark locations for future reference.
<b>Pre-conditions</b>	User is logged into the app.
<b>Post-conditions</b>	<ol style="list-style-type: none"> <li>1. Location is saved to the user's bookmarks.</li> <li>2. Bookmarked locations can be accessed later in the profile or itinerary builder.</li> </ol>
<b>Step #</b>	<b>Typical Course of Action</b>
1.	User selects a location
2.	User clicks the "Bookmark" button.
3.	System saves the location to the user's bookmarks list.
4.	User can access the bookmarks later from their profile.
<b>Step #</b>	<b>Alternate Courses of Action</b>
	User can remove the bookmarks.
<b>Step #</b>	<b>Exception Paths</b>

### 3.2.16 View Business Analytics

<b>Identifier</b>	UC-016
<b>Purpose</b>	Allow business users to view analytics related to their listings, including engagement, reviews, and demographics.
<b>Pre-conditions</b>	<ol style="list-style-type: none"> <li>1. User has a registered business account.</li> <li>2. User is logged into the app.</li> <li>3. User has listed a business on the platform.</li> </ol>
<b>Post-conditions</b>	Analytics data is displayed (views, interactions, demographics).
<b>Step #</b>	<b>Typical Course of Action</b>
1.	Business user logs into their account.

2.	User navigates to the "Business Analytics" section.
3.	System displays relevant metrics for the user's business (e.g., number of views, engagement rates, user demographics).
4.	Business user can export analytics data for further analysis
<b>Step #</b>	<b>Alternate Courses of Action</b>
<b>Step #</b>	<b>Exception Paths</b>
	If there is no data available, the system displays a "no data available" message.

### 3.2.17 Multi-Factor Authentication

<b>Identifier</b>	UC-017
<b>Purpose</b>	To enhance account security by requiring a second factor (e.g., OTP, email) in addition to a password.
<b>Pre-conditions</b>	<ol style="list-style-type: none"> <li>1. User has an active account.</li> <li>2. MFA is enabled on the account.</li> </ol>
<b>Post-conditions</b>	<ol style="list-style-type: none"> <li>1. User successfully logs in after entering the second factor.</li> <li>2. If the second factor is incorrect, access is denied.</li> </ol>
<b>Step #</b>	<b>Typical Course of Action</b>
1.	User logs in with their username and password.
2.	If the username and password are correct, System prompts the user for a second authentication factor (e.g., OTP sent via SMS).
3.	User enters the correct MFA code.
4.	System verifies the MFA code and grants access.
<b>Step #</b>	<b>Alternate Courses of Action</b>
	User selects "Resend OTP" if they do not receive the code. User can use backup codes if the MFA device is unavailable.
<b>Step #</b>	<b>Exception Paths</b>
	If the MFA code is incorrect, the system denies access and requests the user to try again.

### 3.2.18 Push Notification

<b>Identifier</b>	UC-018
<b>Purpose</b>	Notify users about important events, such as new messages, friend requests, and nearby places of interest.
<b>Pre-conditions</b>	User has an account and push notifications access is enabled.
<b>Post-conditions</b>	Notification appears on the user's device.
<b>Step #</b>	<b>Typical Course of Action</b>
1.	A notification triggering event occurs (e.g., new message, recommendation).
2.	System sends a push notification to the user's device.
3.	Notification appears on the user's screen.
4.	User clicks the notification to open the app..
<b>Step #</b>	<b>Alternate Courses of Action</b>
	User can mute specific notifications or categories in the settings.
<b>Step #</b>	<b>Exception Paths</b>

### 3.2.19 Activity Feed

<b>Identifier</b>	UC-019
<b>Purpose</b>	Display a personalized activity feed showing updates on other users and businesses the user follows.
<b>Pre-conditions</b>	<ol style="list-style-type: none"> <li>1. User is logged into the app.</li> <li>2. User has a following list.</li> </ol>
<b>Post-conditions</b>	User sees the latest activities and posts from friends, businesses, or nearby locations
<b>Step #</b>	<b>Typical Course of Action</b>
1.	User is shown the activity feed screen upon login.
2.	System fetches relevant data. (posts, promotions etc)

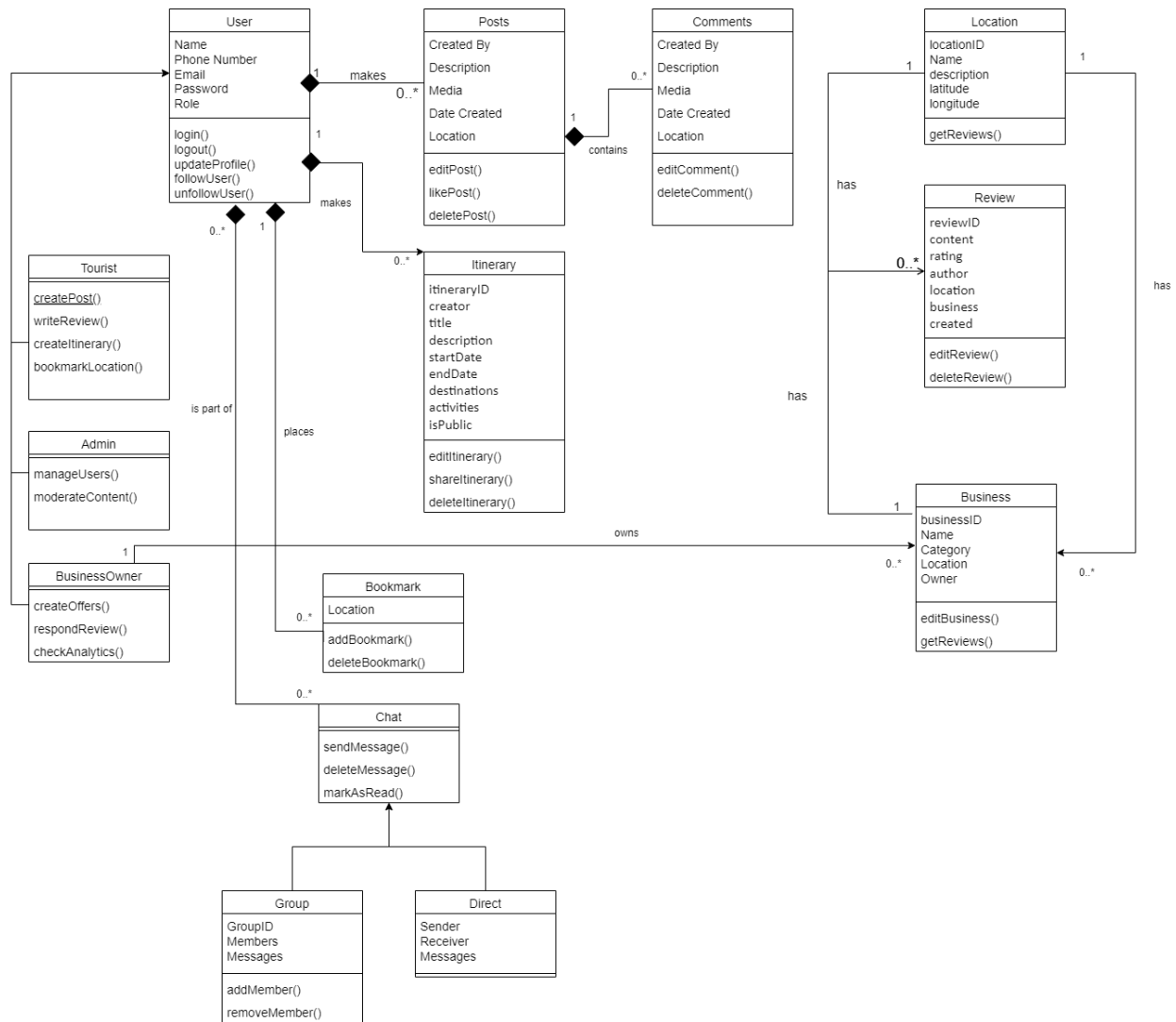
3.	User scrolls through the activity feed.
4.	User can interact with posts (e.g., like, comment, or share).
<b>Step #</b>	<b>Alternate Courses of Action</b>
<b>Step #</b>	<b>Exception Paths</b>
	If there are no new activities, the system displays a "No new updates" message.

### 3.2.20 Report Content

<b>Identifier</b>	UC-020
<b>Purpose</b>	Allow users to report content that violates platform rules, such as offensive posts, spam, or illegal activities.
<b>Pre-conditions</b>	<ol style="list-style-type: none"> <li>1. User is logged into the app.</li> <li>2. User has encountered content they deem inappropriate.</li> </ol>
<b>Post-conditions</b>	<ol style="list-style-type: none"> <li>1. Reported content is flagged for review.</li> <li>2. Admins review the content and take appropriate action (e.g., content removal, user warnings).</li> </ol>
<b>Step #</b>	<b>Typical Course of Action</b>
1.	User finds inappropriate content.
2.	User selects the "Report" option for the content.
3.	System prompts the user to select a reason for reporting (e.g., harassment, spam).
4.	User submits the report.
5.	System logs the report and notifies moderators.
<b>Step #</b>	<b>Alternate Courses of Action</b>
<b>Step #</b>	<b>Exception Paths</b>

## 4. Class Diagram

### 4.1 Diagram



## 4.2 Description

<Give brief description/purpose of each class in the class diagram. Give readable names to classes, attributes and operations.>

### 1. Users

This represents the base class for all users (Tourist, Admin, Business Owner). Contains common attributes and methods shared by all users. Each user can login and register (Admin does not need to register).

#### a. *Tourists*

These are the primary users of the app. They can create posts and itineraries, comment on other posts, search for locations, leave reviews and bookmark them. They can chat with other users and businesses and follow them.

#### b. *Admin*

These users are responsible for monitoring the app health and making sure that its users follow the guidelines of the app. Their main responsibilities are checking app status, moderating posts and managing users.

#### c. *Business Owners*

These users are the secondary users of the app. They can create an account and their business pages and advertise their services to incoming tourists. They can chat with others, check user reviews and monitor overall business analytics.

### 2. Itinerary

Itinerary allows tourists to create and manage their travel plans by adding locations, activities, and dates. Users can choose to make them public or private.

### 3. Bookmark

Bookmark allows users to save certain locations they are interested in visiting if they plan to visit sometime in the future or want to recommend it to others.

#### 4. Chat

Represents the chat functionality for direct and group communication between users.

##### *a. Direct*

This subclass is to represent direct one on one communication between two users.

##### *b. Group*

This subclass represents communication between multiple members.

#### 5. Posts

Posts class will allow users to create posts about their experiences so that other viewers can check them out.

#### 6. Comments

Comments class represents a list of comments left by users on a certain post.

#### 7. Location

This represents tourist destinations for which users can make posts about, create itineraries, leave reviews and bookmark.

#### 8. Review

This class represents reviews left by users for a certain location or a business in that location for others to view.

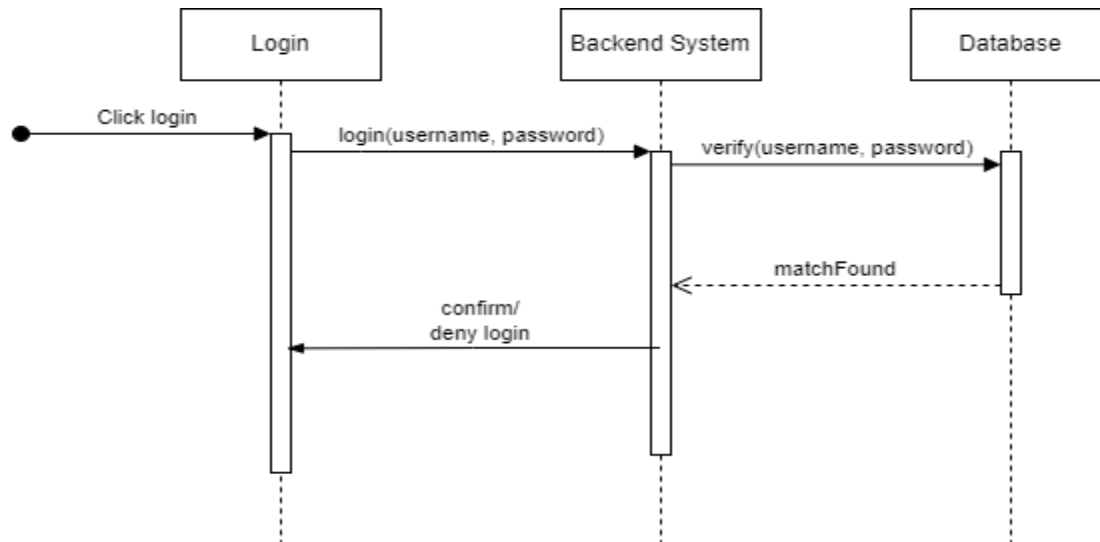
#### 9. Business

Business class represents a business in a certain location owned by a certain business owner. Users can view businesses, chat and write a review about them. Meanwhile business owners can advertise their services and check overall performance.

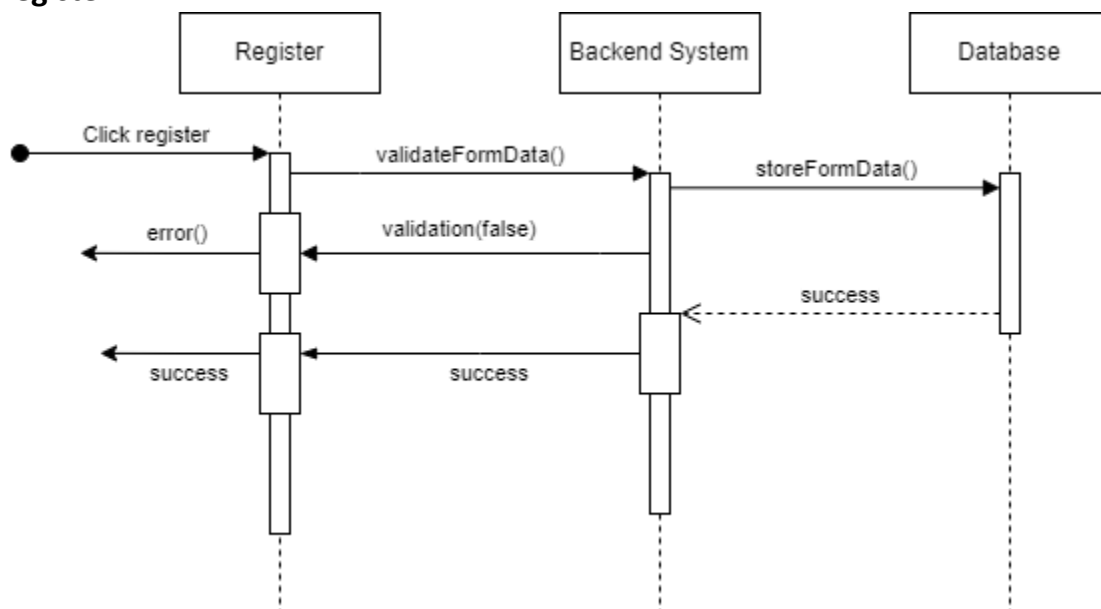


## 5. Sequence Diagrams

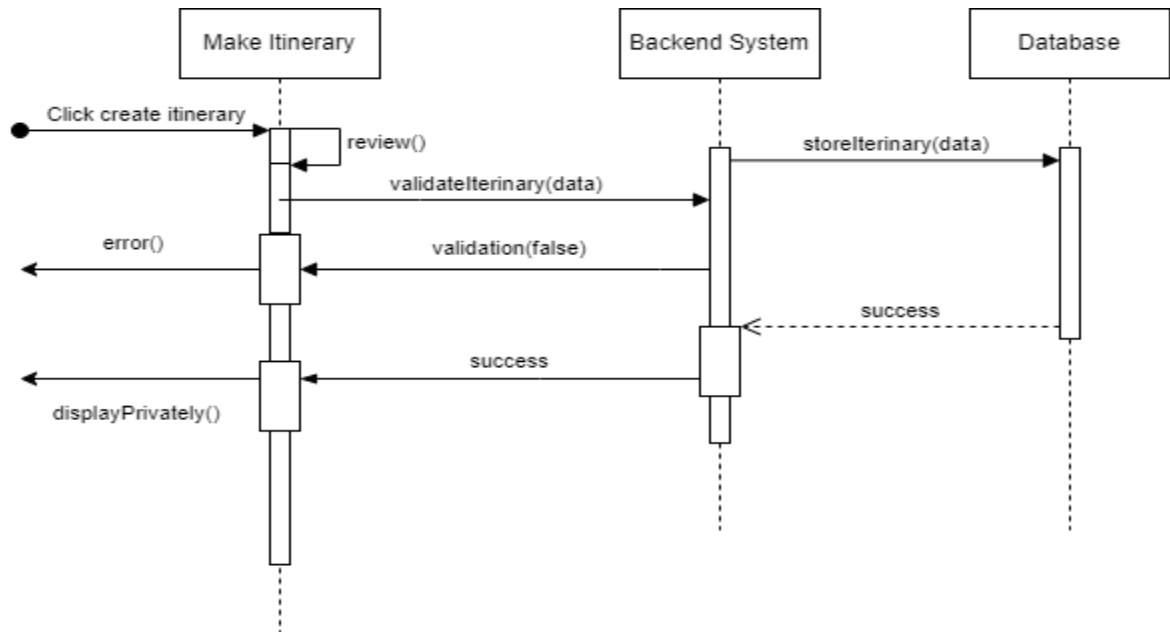
### 5.1 Login



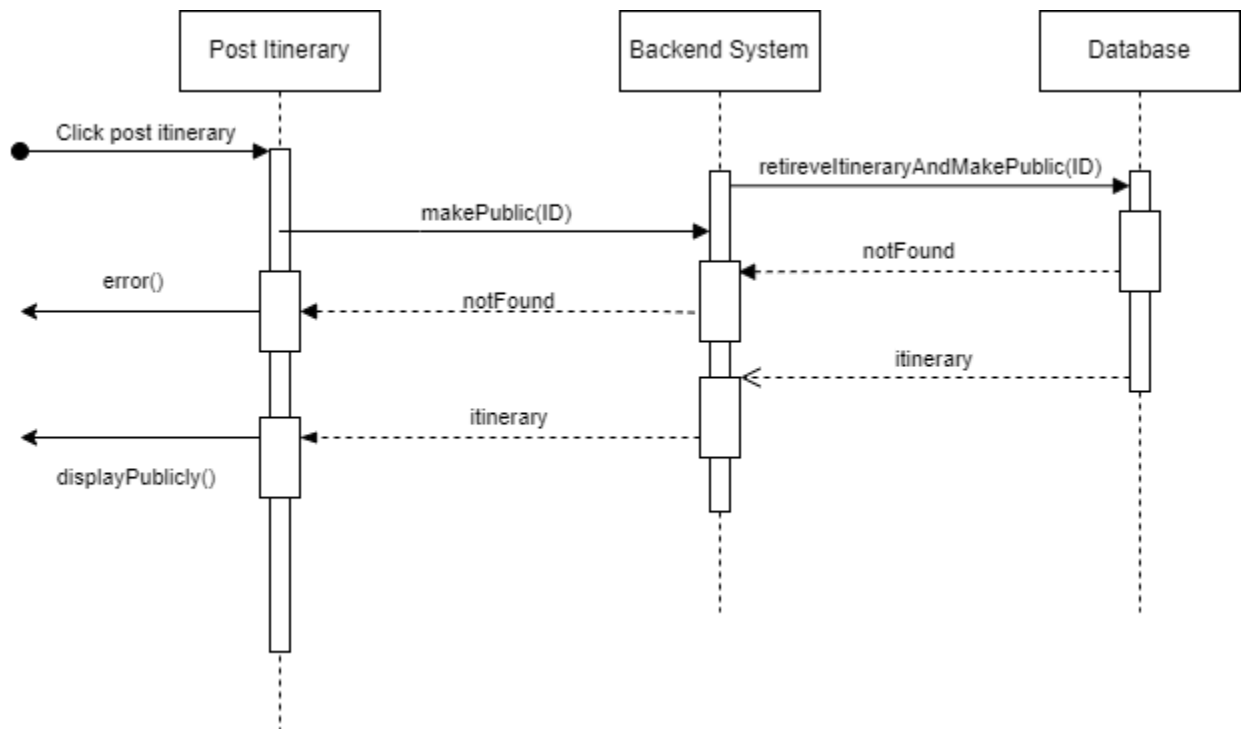
### 5.2 Register



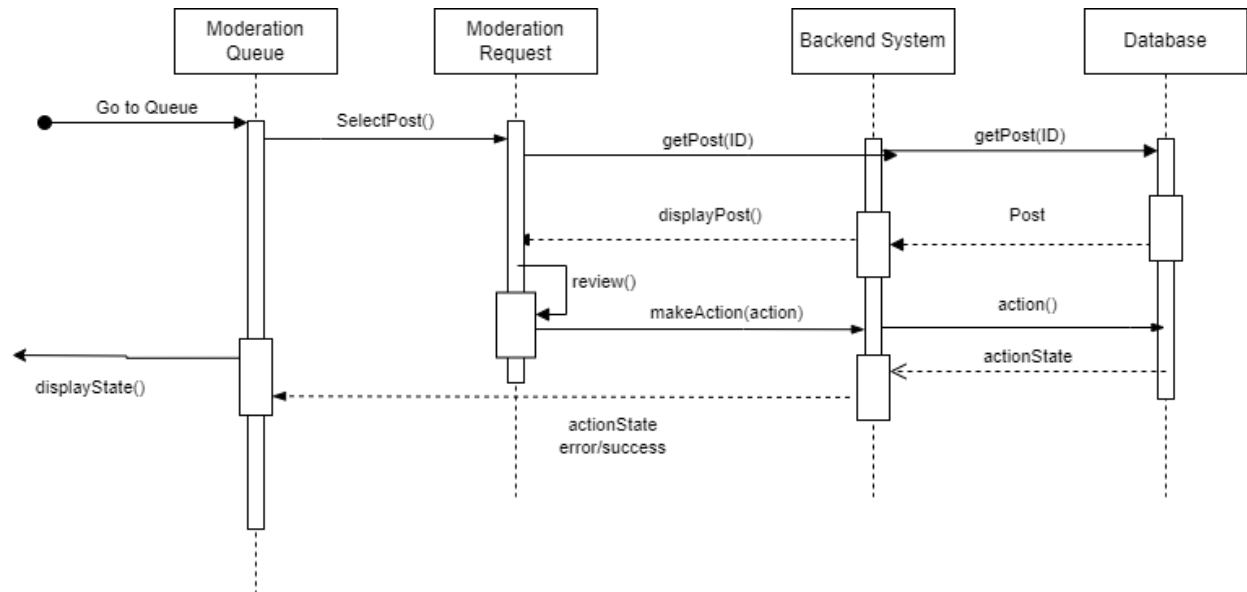
### 5.3 Making Itineraries



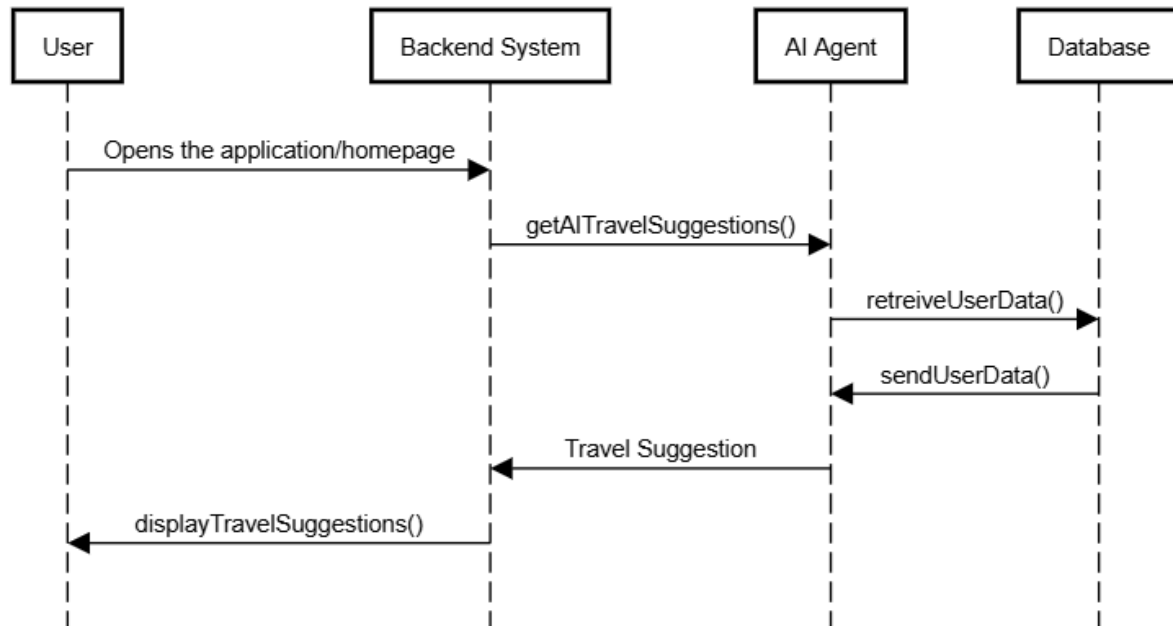
### 5.4 Posting Itineraries



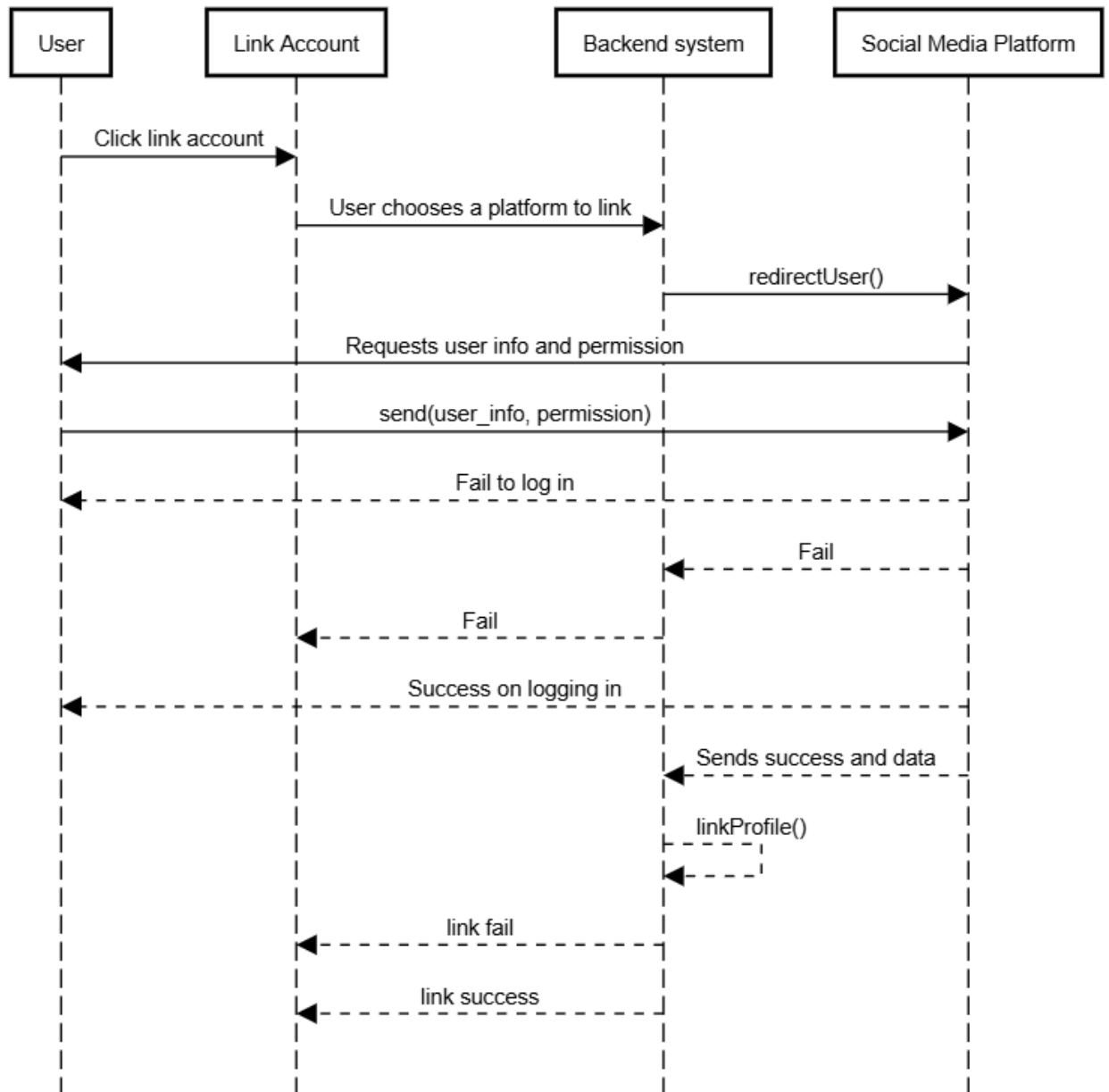
## 5.5 Post Moderation by Admin Users



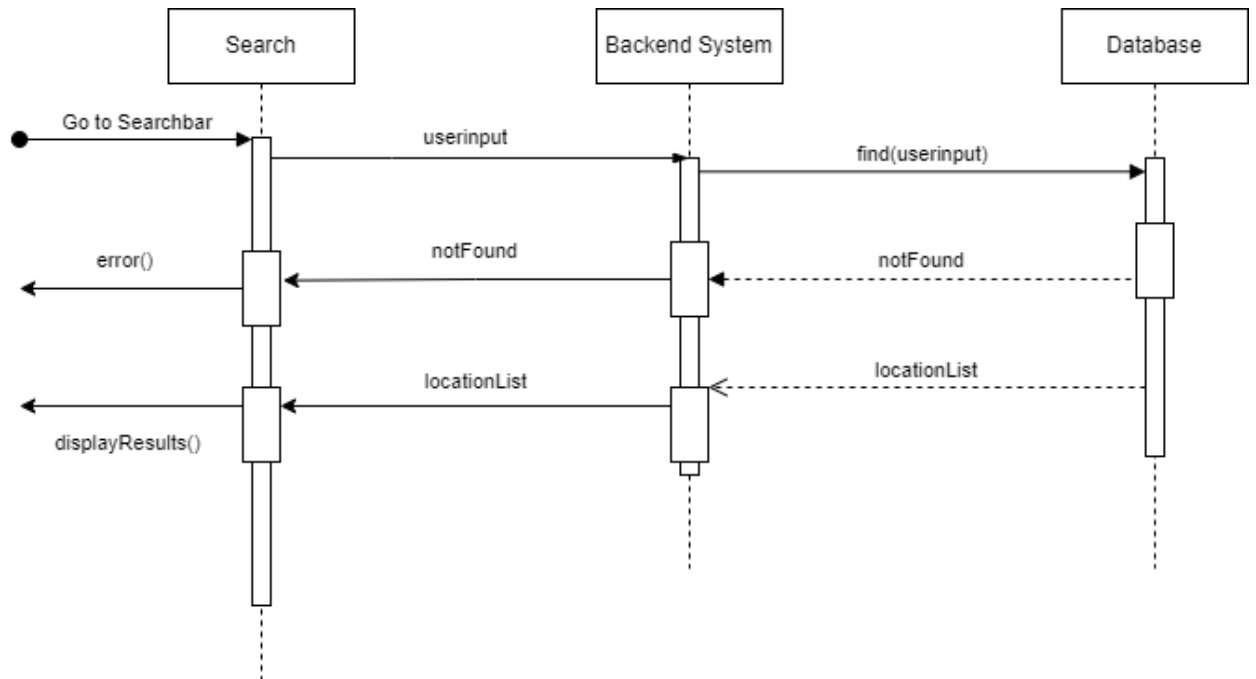
## 5.6 AI-Based Travel/Post Suggestions



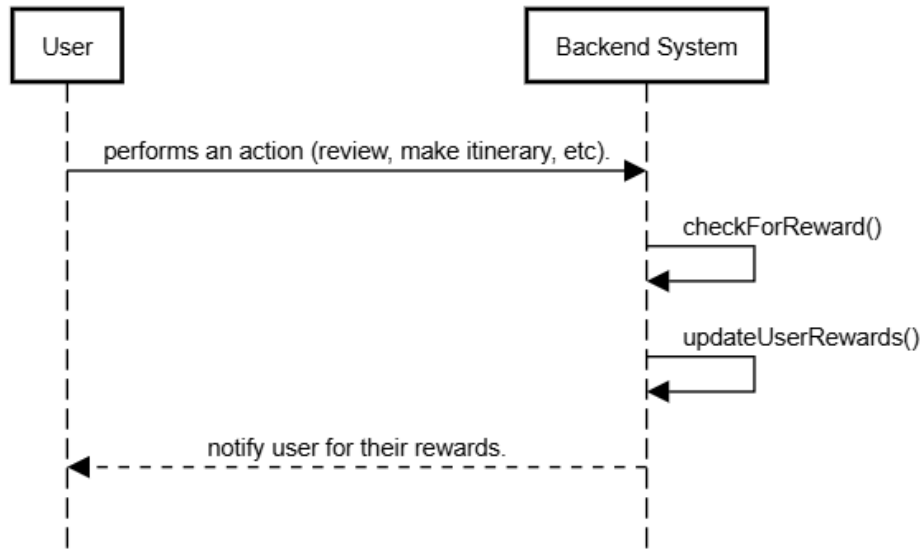
## 5.7 Linking Social Media Profiles



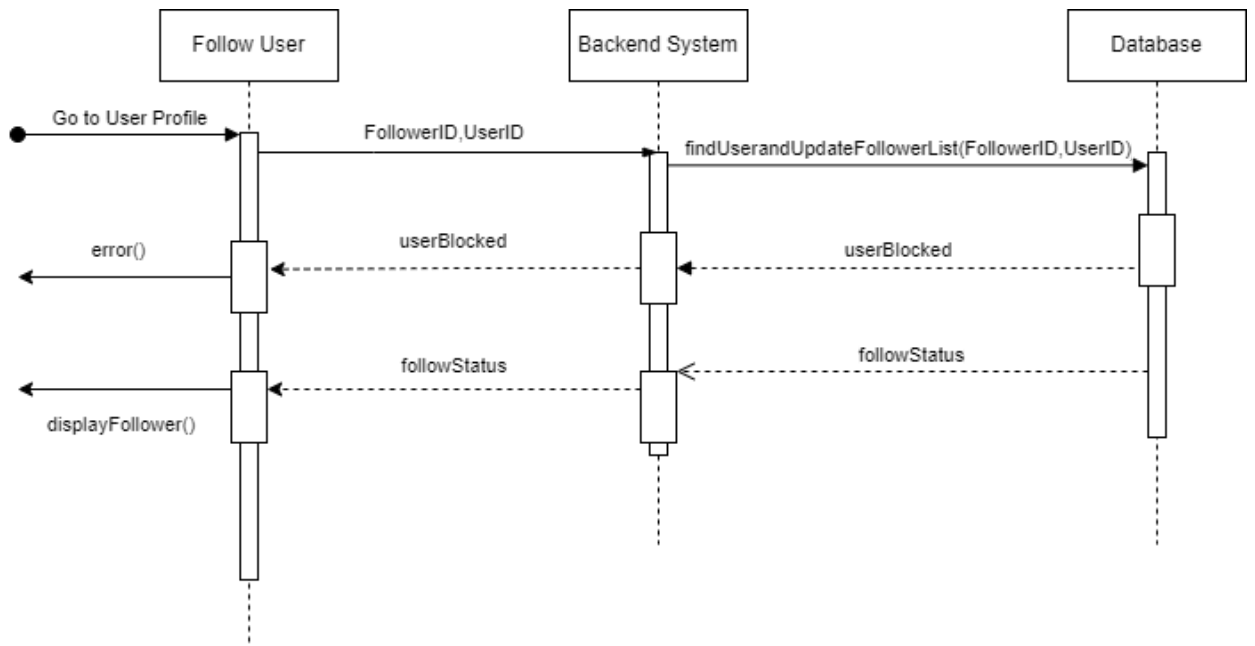
## 5.8 Searching for Locations



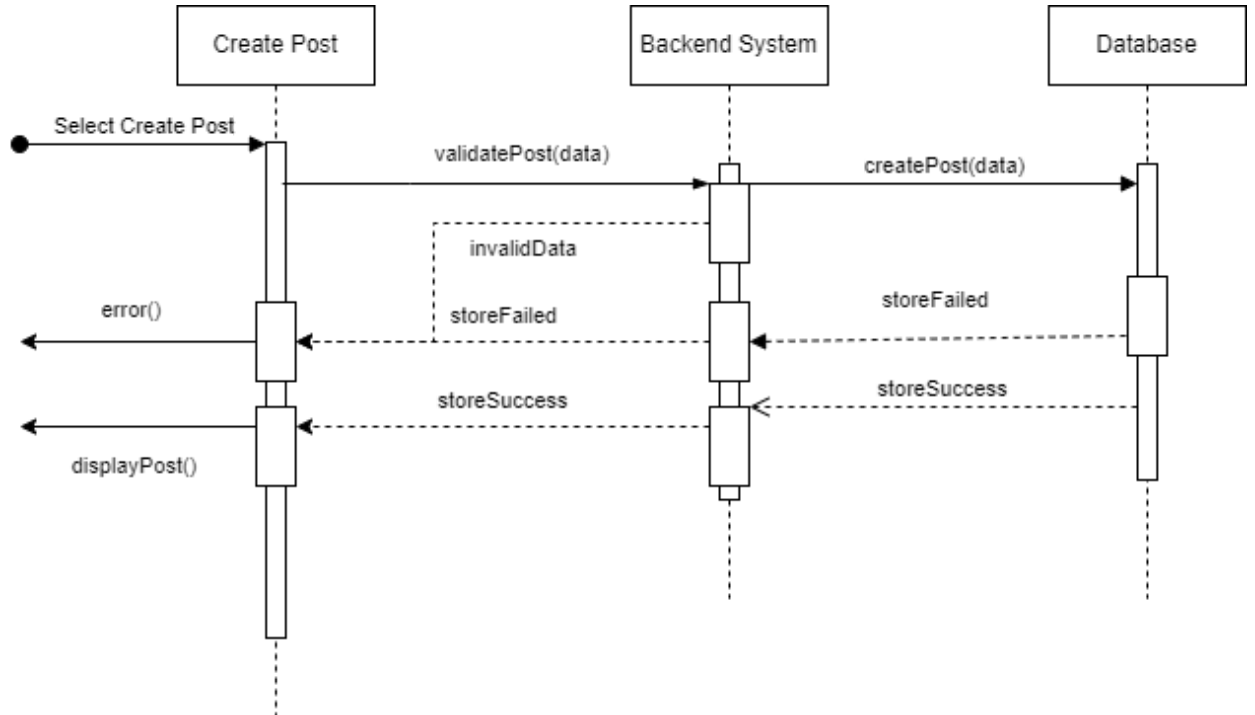
## 5.9 Rewards System



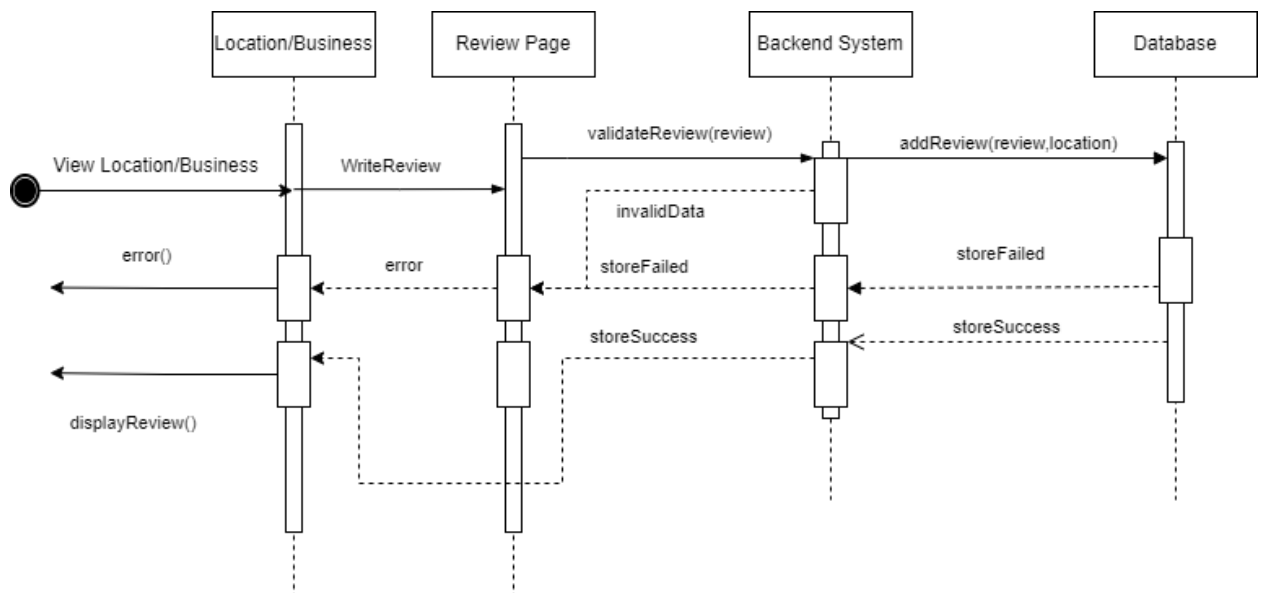
## 5.10 Adding Friends



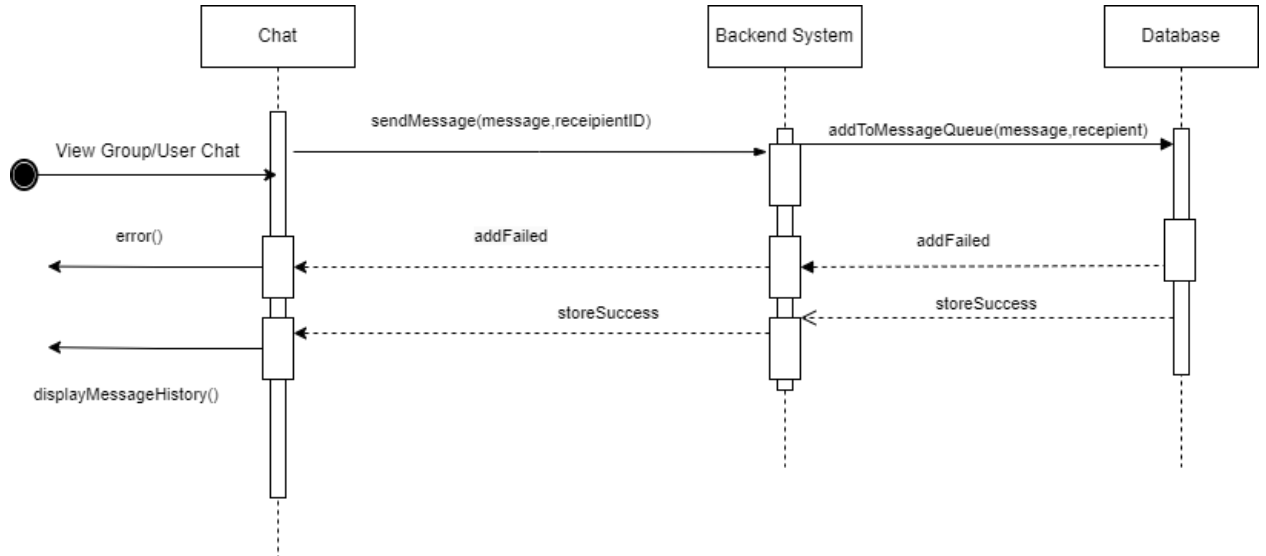
## 5.11 Making Posts



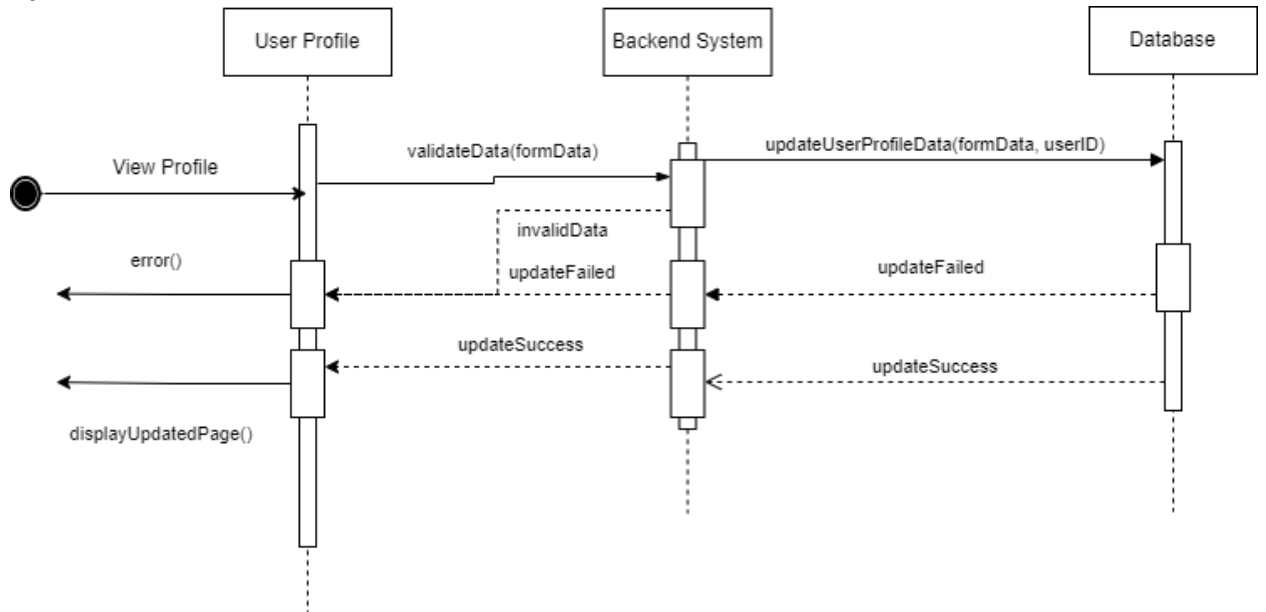
## 5.12 Writing a Review



### 5.13 Sending Messages

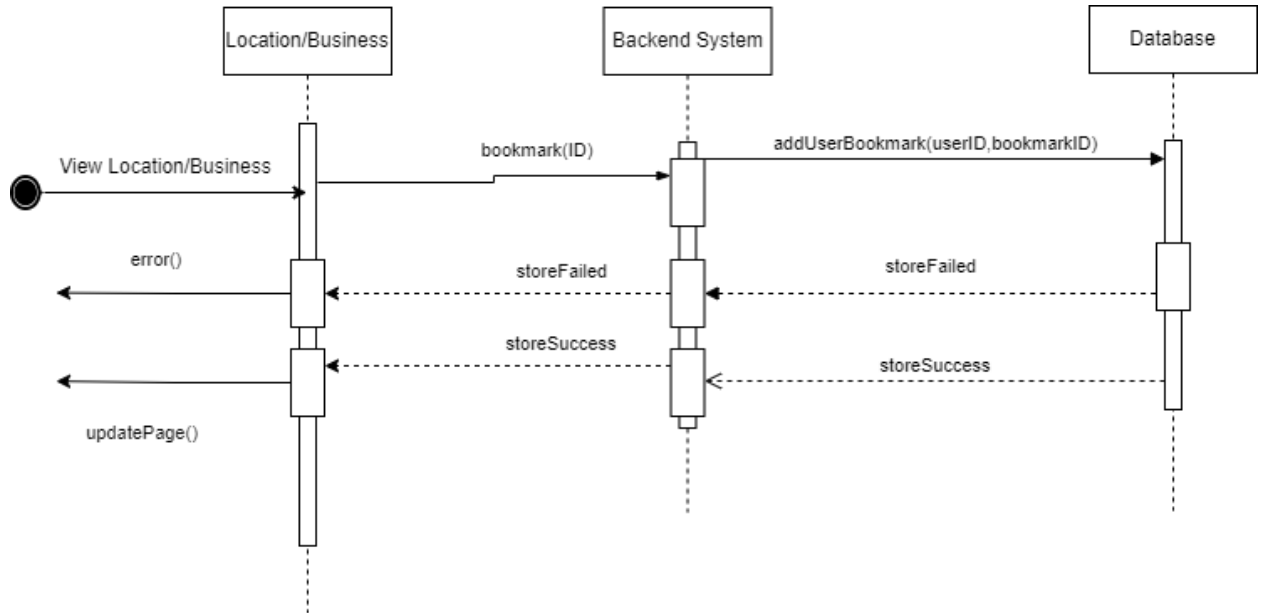


### 5.14 Update User Profile

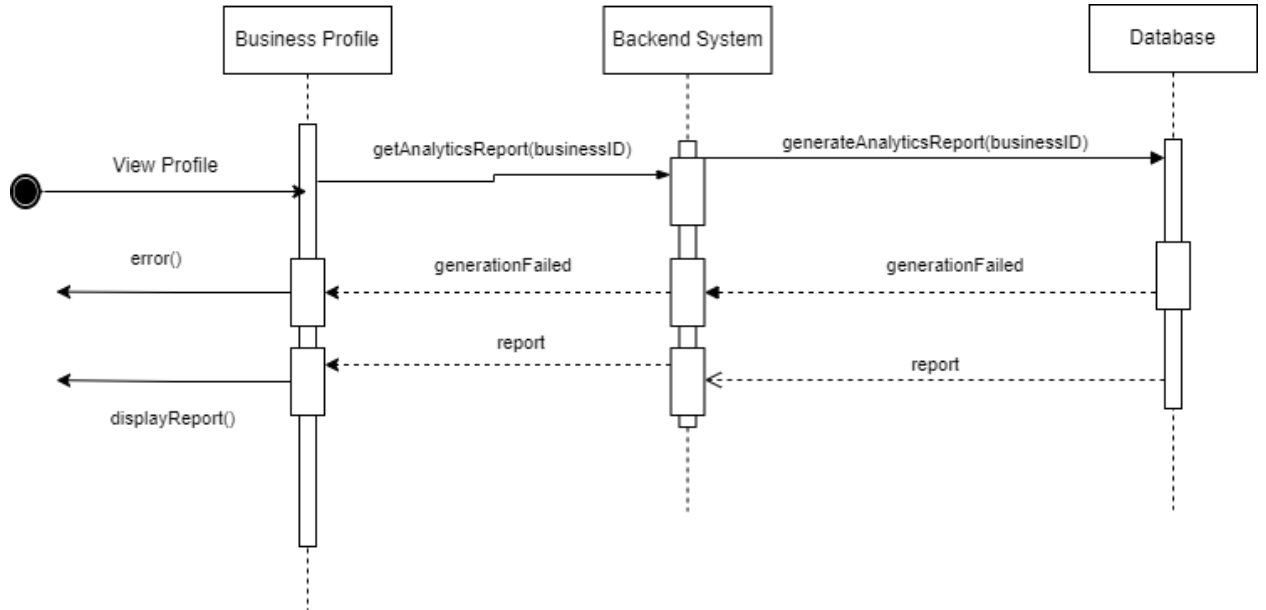




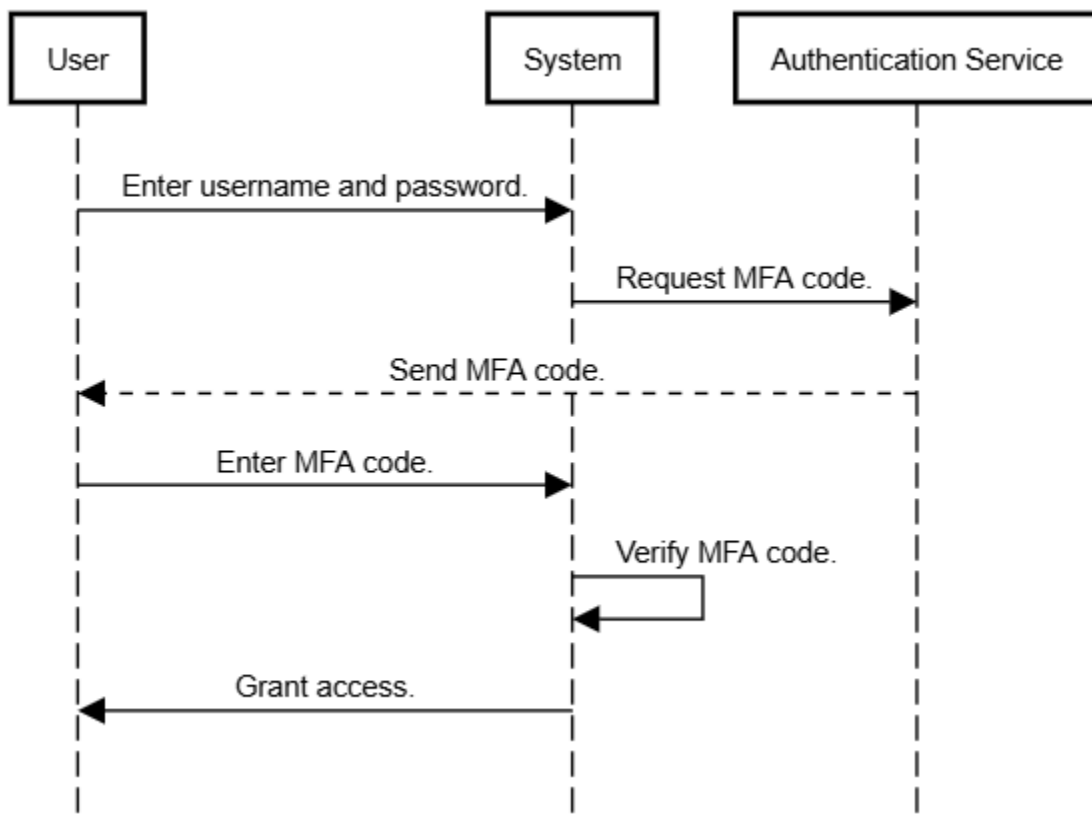
### 5.15 Adding Travel Bookmarks



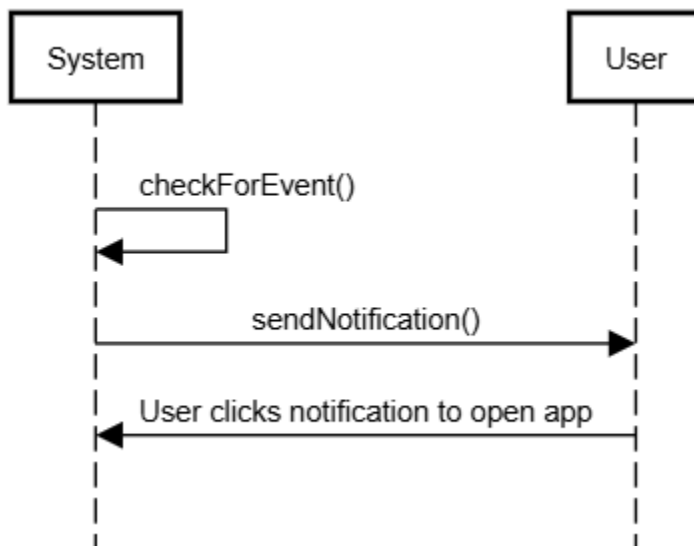
### 5.16 View Business Analytics



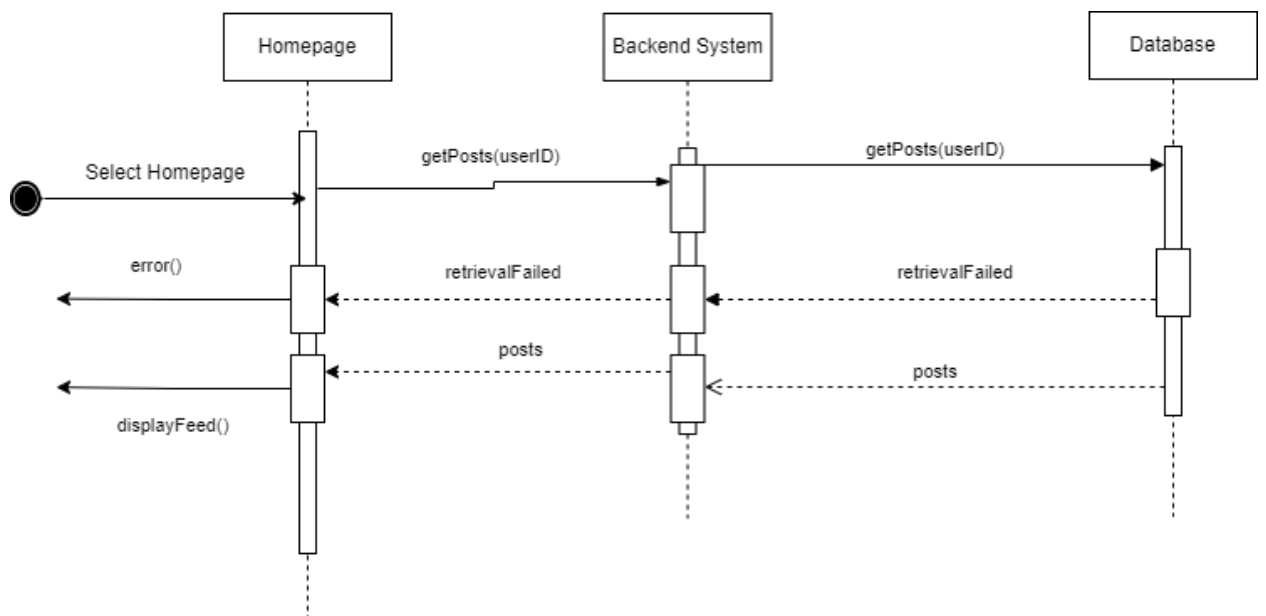
## 5.17 Multi-Factor Authentication



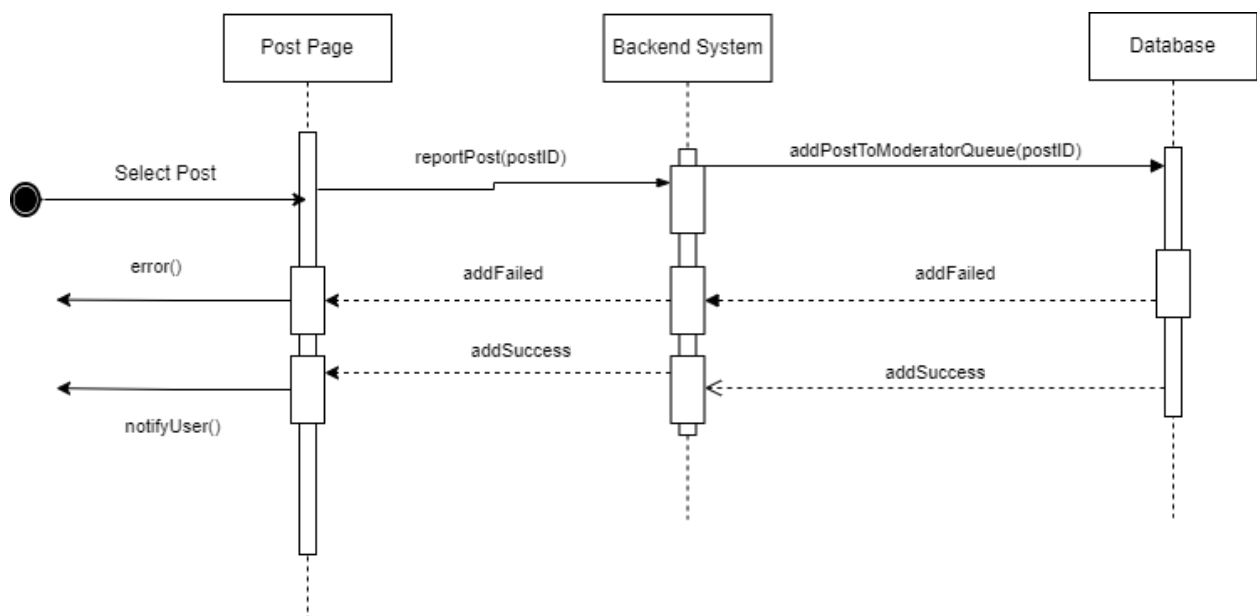
## 5.18 Push Notification



## 5.19 Activity Feed



## 5.20 Report Content



## **6. State Diagrams**

<Repeat the following if you need to draw state diagrams of multiple objects>

### **6.1 Diagram details**

<Name of the object whose state is being shown along with details of states and messages.>

### **6.2 Diagram**

<Use standard UML notation>

.....

## 7. Non-functional Requirements / Quality Attributes

Sr#	Requirements
1	At any time during its execution, the system must not utilize more than 250MB of memory to ensure efficient resource usage and memory-related issues.
2	In the event of a failure, the system must restore to normal operations within 2-3 minutes of a failure to minimize downtime and maintain user satisfaction.
3	The system must handle at least 10,000 users without performance degradation.
4	During peak usage, the system must respond to user requests within 7-8 seconds.
5	The system must be highly available, maintaining an uptime of 99.9%, especially during peak travel seasons.
6	The system must be able to run on both IOS and Android devices to cater to a wide range of users across different platforms.
7	The system must be easily usable by all demographics, maintaining accessibility for a wide user base.
8	The system must be scalable to accommodate the increase or decrease in traffic.
9	The system must be able to handle up to 400-500 requests at peak usage.
10	The system must be designed to be easily maintainable to facilitate testing, ongoing updates, and support.
11	The system must ensure the data is well protected, ensuring data integrity through backups and redundancy measures.

## 8. Who Did What?

Name of the Team Member	Tasks done
Muhammad Affan Naved	Section 3, 4, 5 (half)
Shahrez Aezad	Section 6
Mohammad Haroon Khawaja	Section 5 (half), 3.2
Pir M. Shahraiz Chishty	
Lugman Aadil	

## 9. Review checklist

Before submission of this deliverable, the team must perform an internal review. Each team member will review one or more sections of the deliverable.

Section Title	Reviewer Name(s)
Muhammad Affan Naved	Section 2,4,5
Shahrez Aezad	
Mohammad Haroon Khawaja	