**REQUIREMENTS SPECIFICATIONS**

**P-10:ODYSSEUM**

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

|  |  |  |
| --- | --- | --- |
| **Content** | **Totals** | **Obtained** |
| Introduction & system actors | 5 | 5 |
| Use case diagram | 10 | 6 |
| Use case descriptions | 20 | 12 |
| Class diagram | 20 | 18 |
| Sequence diagram | 20 | 17 |
| State diagram | 5 | 5 |
| Non-functional requirements | 5 | 5 |
| Who did what | 5 | 5 |
| Review checklist | 5 | 5 |
| Overall formatting/template | 5 | 5 |
| Late submission penalty | -20 | - |
| GitHub Folder structure penalty | -5 | - |
| **Grand Total** | **100** | **83** |
| **General Comments/Individual Grading:** | | |

**TABLE OF CONTENTS**

1. Introduction 3
2. System Actors 4
3. Use Cases 5
   1. Use Case Diagrams 6
   2. Description of Use Cases 7
      1. Login 7
      2. Register 8
      3. Making itineraries 9
      4. Posting Itineraries 10
      5. Post Moderation by Admin Users 11
      6. AI-Based Travel/Post Suggestions 12
      7. Linking Social Media Profiles 13
      8. Searching for Locations 13
      9. Rewards System 14
      10. Adding Friends 15
      11. Making Posts 15
      12. Writing a Review 16
      13. Sending Messages 17
      14. Update User Profile 17
      15. Adding Travel Bookmarks 18
      16. View Business Analytics 19
      17. Multi-Factor Authentication 19
      18. Push Notification 20
      19. Activity Feed 21
      20. Report Content 21
4. Class Diagram 23
   1. Diagram 23
   2. Description 24
5. Users 24
6. Itinerary 24
7. Bookmark 24
8. Chat 25
9. Posts 25
10. Comments 25
11. Location 25
12. Review 25
13. Business 25
14. [Sequence Diagrams 25](#_TOC_250026)
    1. [Login 26](#_TOC_250025)
    2. [Register 26](#_TOC_250024)
    3. [Making Itineraries 26](#_TOC_250023)
    4. [Posting Itineraries 26](#_TOC_250022)
    5. [Post Moderation by Admin Users 26](#_TOC_250021)
    6. [AI-Based Travel/Post Suggestions 26](#_TOC_250020)
    7. [Linking Social Media Profiles 26](#_TOC_250019)
    8. [Searching for Locations 26](#_TOC_250018)
    9. [Rewards System 26](#_TOC_250017)
    10. [Adding Friends 26](#_TOC_250016)
    11. [Making Posts 26](#_TOC_250015)
    12. [Writing a Review 26](#_TOC_250014)
    13. [Sending Messages 26](#_TOC_250013)
    14. [Update User Profile 26](#_TOC_250012)
    15. [Adding Travel Bookmarks 26](#_TOC_250011)
    16. [View Business Analytics 26](#_TOC_250010)
    17. [Multi-Factor Authentication 26](#_TOC_250009)
    18. [Push Notification 26](#_TOC_250008)
    19. [Activity Feed 26](#_TOC_250007)
    20. [Report Content 26](#_TOC_250006)
15. [State Diagrams 27](#_TOC_250005)
    1. [Diagram details 27](#_TOC_250004)
    2. [Diagram 27](#_TOC_250003)
16. [Non-functional Requirements / Quality Attributes 28](#_TOC_250002)
17. [Who Did What? 28](#_TOC_250001)
18. [Review checklist 29](#_TOC_250000)

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.

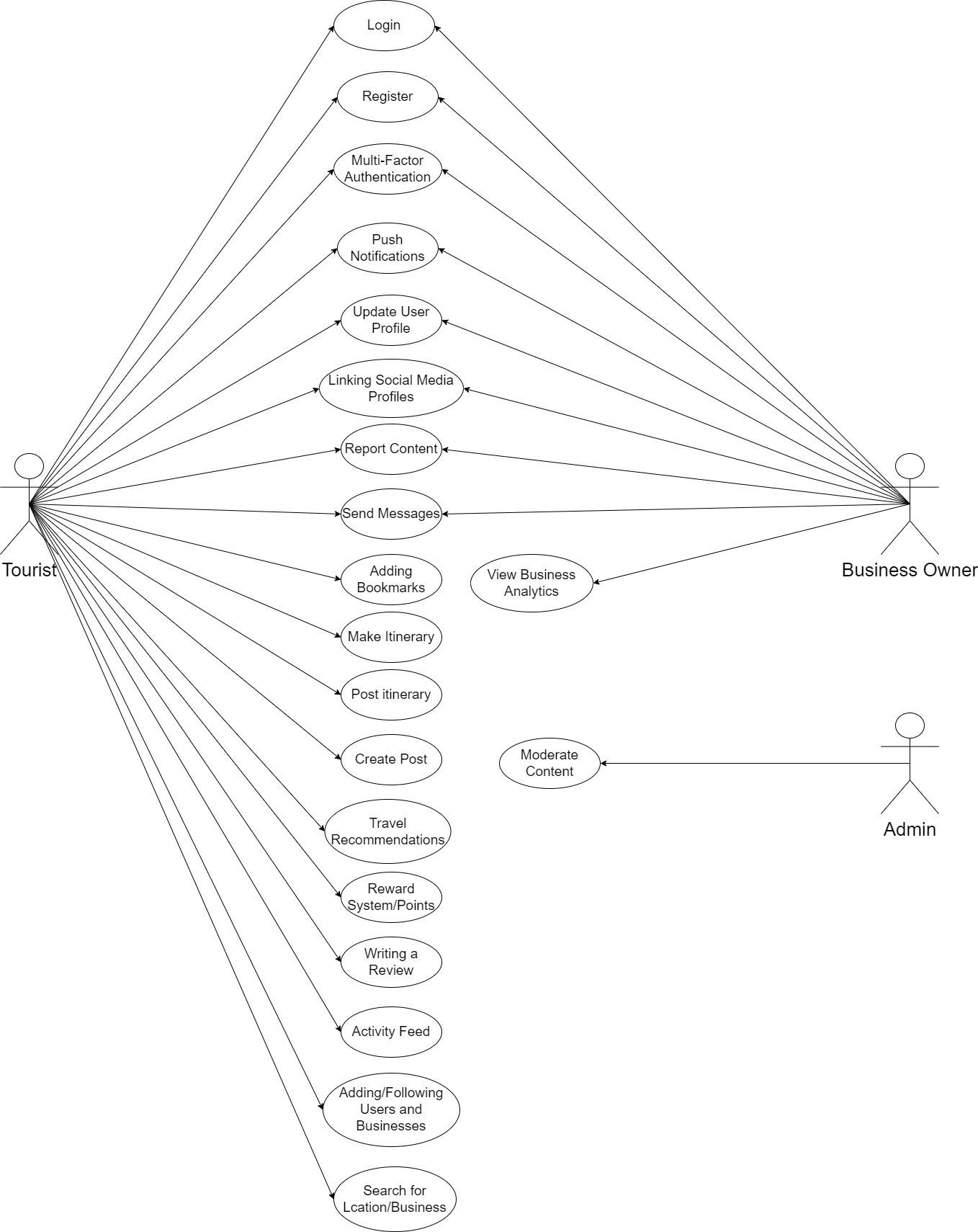
As stated above, potential users of this app include travelers themselves, administrators and local businesses which are but not limited to hotels, restaurants and tour guides.

|  |  |
| --- | --- |
| **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. |

## Use Case Diagrams]

## [No Use case for tour guide.]

## The use case names should reflect the goals that a user wants to achieve. Some of the use case names in your diagram are nouns which is not appropriate.]



* 1. **Description of Use Cases**

[Alternate course of action missing from use cases? Post conditions are not correctly written, please consult lecture slides.]

3.2.1 Login

|  |  |  |
| --- | --- | --- |
| **Identifier** | | UC-001 |
| **Purpose** | | The user enters their credentials to log in to access the app functionalities. |
| **Pre-condit ions** | | Users must have registered an account. |
| **Post-condi tions** | | The user is logged in and navigated to their general feed. |
|  | | |
| **Step #** | **Typical Course of Action** | |
| 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. | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
|  |  | |
|  |  | |
| **Step #** | **Exception Paths** | |
| 1. | User enters invalid username or password. | |
| 2. | System will deny access to the user and ask to try again. | |

|  |  |  |
| --- | --- | --- |
| **Identifier** | | UC-002 |
| **Purpose** | | User needs an account to access app functionalities. |
| **Pre-condit ions** | | App must be installed on the user’s phone. |
| **Post-condi tions** | | Account created and user can now login. |
|  | | |
| **Step #** | **Typical Course of Action** | |
| 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. | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
|  |  | |
|  |  | |
|  |  | |
| **Step #** | **Exception Paths** | |
| 1. | User supplied form data is incorrect or invalid. | |
| 2. | System sends the user an error message and asks them to input data again. | |

|  |  |  |
| --- | --- | --- |
| **Identifier** | | UC-003 |
| **Purpose** | | Allow users to create and save custom travel itineraries, including destinations, activities, accommodations, and dates. |
| **Pre-condit ions** | | User is logged into the app. |
| **Post-condi tions** | | Travel itinerary is saved to the user’s account.  Itinerary can be shared with others or accessed for further updates. |
|  | | |
| **Step #** | **Typical Course of Action** | |
| 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. | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
| 1. |  | |
| **Step #** | **Exception Paths** | |
|  | 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. | |

|  |  |  |
| --- | --- | --- |
| **Identifier** | | UC-004 |
| **Purpose** | | Allow users to share their travel itineraries from their itinerary tab, making them visible to other users for viewing and feedback. |
| **Pre-condit ions** | | User is logged into the app.  User has at least one saved itinerary. |
| **Post-condi tions** | | Itinerary is posted and visible to other users.  Other users can view, like, or comment on the shared itinerary. |
|  | | |
| **Step #** | **Typical Course of Action** | |
| 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 | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
| 1. | User can make their itinerary private again. | |
| **Step #** | **Exception Paths** | |
| 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. | |

|  |  |  |
| --- | --- | --- |
| **Identifier** | | UC-005 |
| **Purpose** | | Allow admins to review and moderate user posts for inappropriate content, spam, or violations of community guidelines. |
| **Pre-condit ions** | | 1. Admin is logged into the admin panel. 2. A user has posted content that needs moderation. |
| **Post-condi tions** | | 1. Post is either approved or flagged for removal/violations. 2. If removed, the user is notified with the reason |
|  | | |
| **Step #** | **Typical Course of Action** | |
| 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. | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
|  |  | |
| **Step #** | **Exception Paths** | |
|  | If the post was flagged incorrectly, the admin can mark it as safe, and it is restored. | |

|  |  |  |
| --- | --- | --- |
| **Identifier** | | UC-006 |
| **Purpose** | | Provide personalized travel suggestions based on user preferences, past trips, and reviews. |
| **Pre-condit ions** | | User is logged in. |
| **Post-condi tions** | | User receives a list of suggested travel destinations, activities, and accommodations and their main feed also shows posts related to these recommendations. |
|  | | |
| **Step #** | **Typical Course of Action** | |
| 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. | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
|  |  | |
| **Step #** | **Exception Paths** | |
|  | If the AI engine lacks sufficient data, generic travel suggestions are provided. | |

* + 1. Linking Social Media Profiles

|  |  |  |
| --- | --- | --- |
| **Identifier** | | UC-007 |
| **Purpose** | | Allow users to link their external social media accounts to the app for sharing content or synchronizing profiles. |
| **Pre-condit ions** | | 1. User is logged into the app. 2. User has external social media accounts they wish to link. |
| **Post-condi tions** | | 1. Social media profiles are successfully linked. 2. User can share content directly from the app to their linked social media accounts |
|  | | |
| **Step #** | **Typical Course of Action** | |

|  |  |
| --- | --- |
| 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. |
|  | |
| **Step #** | **Alternate Courses of Action** |
|  | User can choose to cancel the process and not link any social media account. |
| **Step #** | **Exception Paths** |
|  | If the authentication fails, the system prompts the user to retry. |

* + 1. Searching for Locations

|  |  |  |
| --- | --- | --- |
| **Identifier** | | UC-008 |
| **Purpose** | | Allow users to search for travel destinations, restaurants, hotels, and points of interest. |
| **Pre-condit ions** | | User is logged into the app. |
| **Post-condi tions** | | Search results are displayed based on user input.  User can select a location for more details or add it to an itinerary. |
|  | | |
| **Step #** | **Typical Course of Action** | |
| 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 | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
|  | User can apply filters (e.g., price, distance) to refine results. | |
| **Step #** | **Exception Paths** | |
|  | If no results are found, the system displays “Not found”. | |

|  |  |  |
| --- | --- | --- |
| **Identifier** | | UC-009 |
| **Purpose** | | Encourage user engagement by offering rewards for activities like posting reviews, adding friends, and traveling. |
| **Pre-condit ions** | | 1. User is logged into the app. 2. User performs reward-eligible actions (e.g., writing reviews, visiting lcoations). |
| **Post-condi tions** | | 1. Rewards are earned and credited to the user’s account. 2. User can redeem rewards for discounts, badges, or other benefits. |
|  | | |
| **Step #** | **Typical Course of Action** | |
| 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 | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
|  |  | |
| **Step #** | **Exception Paths** | |
|  | If a reward redemption fails, the system prompts the user to retry. | |

* + 1. Adding Friends

|  |  |  |
| --- | --- | --- |
| **Identifier** | | UC-010 |
| **Purpose** | | Allow users to connect with friends or other travelers on the platform. |
| **Pre-condit ions** | | User is logged into the app. |
| **Post-condi tions** | | User can see their followings posts and other public credentials. |
|  | | |
| **Step #** | **Typical Course of Action** | |
| 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. |
|  | |
| **Step #** | **Alternate Courses of Action** |
|  | User can press unfollow if they do not wish to see their posts anymore. |
| **Step #** | **Exception Paths** |
|  | If the user has blocked the other user, the system prevents following them. |

* + 1. Making Posts

|  |  |  |
| --- | --- | --- |
| **Identifier** | | UC-011 |
| **Purpose** | | Allow users to share posts about their travels, itineraries, or reviews. |
| **Pre-condit ions** | | User is logged into the app. |
| **Post-condi tions** | | 1. Post is created and shared with the user’s followers. 2. Post appears in the user’s feed and others can interact with it. |
|  | | |
| **Step #** | **Typical Course of Action** | |
| 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. | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
|  | User can remove a post they have already made. | |
| **Step #** | **Exception Paths** | |
|  |  | |

|  |  |  |
| --- | --- | --- |
| **Identifier** | | UC-012 |
| **Purpose** | | Allow users to write and post reviews for destinations, restaurants, and accommodations. |
| **Pre-condit ions** | | 1. User is logged into the app. 2. User has visited or interacted with the location being reviewed. |
| **Post-condi tions** | | 1. Review is posted publicly or to selected audiences. 2. Other users can view and rate the review. 3. Businesses can now view the user reviews. |
|  | | |
| **Step #** | **Typical Course of Action** | |
| 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. | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
|  | User can delete or edit their review. | |
| **Step #** | **Exception Paths** | |
|  | If the user doesn’t complete all fields (e.g., rating), the system prompts them to finish before submitting. | |

* + 1. Sending Messages

|  |  |  |
| --- | --- | --- |
| **Identifier** | | UC-013 |
| **Purpose** | | Allow users to send direct messages to other users or group chats. |
| **Pre-condit ions** | | User is logged into the app. |
| **Post-condi tions** | | 1. Messages are sent and received. 2. Chat history is updated in real-time. |
|  | | |
| **Step #** | **Typical Course of Action** | |
| 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. |
|  | |
| **Step #** | **Alternate Courses of Action** |
|  | User can edit or delete the message. |
| **Step #** | **Exception Paths** |
|  | If the user has blocked the other user, the system prevents sending messages. |

* + 1. Update User Profile

|  |  |  |
| --- | --- | --- |
| **Identifier** | | UC-014 |
| **Purpose** | | Allow users to update their personal information, preferences, and travel interests. |
| **Pre-condit ions** | | User is logged into the app. |
| **Post-condi tions** | | User profile is updated with the new information. |
|  | | |
| **Step #** | **Typical Course of Action** | |
| 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. | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
|  | User can cancel the profile update. | |
| **Step #** | **Exception Paths** | |
|  |  | |

|  |  |  |
| --- | --- | --- |
| **Identifier** | | UC-015 |
| **Purpose** | | Allow users to save or bookmark locations for future reference. |
| **Pre-condit ions** | | User is logged into the app. |
| **Post-condi tions** | | 1. Location is saved to the user’s bookmarks. 2. Bookmarked locations can be accessed later in the profile or itinerary builder. |
|  | | |
| **Step #** | **Typical Course of Action** | |
| 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. | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
|  | User can remove the bookmarks. | |
| **Step #** | **Exception Paths** | |
|  |  | |

* + 1. View Business Analytics

|  |  |  |
| --- | --- | --- |
| **Identifier** | | UC-016 |
| **Purpose** | | Allow business users to view analytics related to their listings, including engagement, reviews, and demographics. |
| **Pre-condit ions** | | 1. User has a registered business account. 2. User is logged into the app. 3. User has listed a business on the platform. |
| **Post-condi tions** | | Analytics data is displayed (views, interactions, demographics). |
|  | | |
| **Step #** | **Typical Course of Action** | |
| 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 |
|  | |
| **Step #** | **Alternate Courses of Action** |
|  |  |
| **Step #** | **Exception Paths** |
|  | If there is no data available, the system displays a "no data available" message. |

* + 1. Multi-Factor Authentication

|  |  |  |
| --- | --- | --- |
| **Identifier** | | UC-017 |
| **Purpose** | | To enhance account security by requiring a second factor (e.g., OTP, email) in addition to a password. |
| **Pre-condit ions** | | 1. User has an active account. 2. MFA is enabled on the account. |
| **Post-condi tions** | | 1. User successfully logs in after entering the second factor. 2. If the second factor is incorrect, access is denied. |
|  | | |
| **Step #** | **Typical Course of Action** | |
| 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. | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
|  | User selects "Resend OTP" if they do not receive the code. User can use backup codes if the MFA device is unavailable. | |
| **Step #** | **Exception Paths** | |
|  | If the MFA code is incorrect, the system denies access and requests the user to try again. | |

|  |  |  |
| --- | --- | --- |
| **Identifier** | | UC-018 |
| **Purpose** | | Notify users about important events, such as new messages, friend requests, and nearby places of interest. |
| **Pre-condit ions** | | User has an account and push notifications access is enabled. |
| **Post-condi tions** | | Notification appears on the user's device. |
|  | | |
| **Step #** | **Typical Course of Action** | |
| 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.. | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
|  | User can mute specific notifications or categories in the settings. | |
| **Step #** | **Exception Paths** | |
|  |  | |

* + 1. Activity Feed

|  |  |  |
| --- | --- | --- |
| **Identifier** | | UC-019 |
| **Purpose** | | Display a personalized activity feed showing updates on other users and businesses the user follows. |
| **Pre-condit ions** | | 1. User is logged into the app. 2. User has a following list. |
| **Post-condi tions** | | User sees the latest activities and posts from friends, businesses, or nearby locations |
|  | | |
| **Step #** | **Typical Course of Action** | |
| 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). |
|  | |
| **Step #** | **Alternate Courses of Action** |
|  |  |
| **Step #** | **Exception Paths** |
|  | If there are no new activities, the system displays a "No new updates" message. |

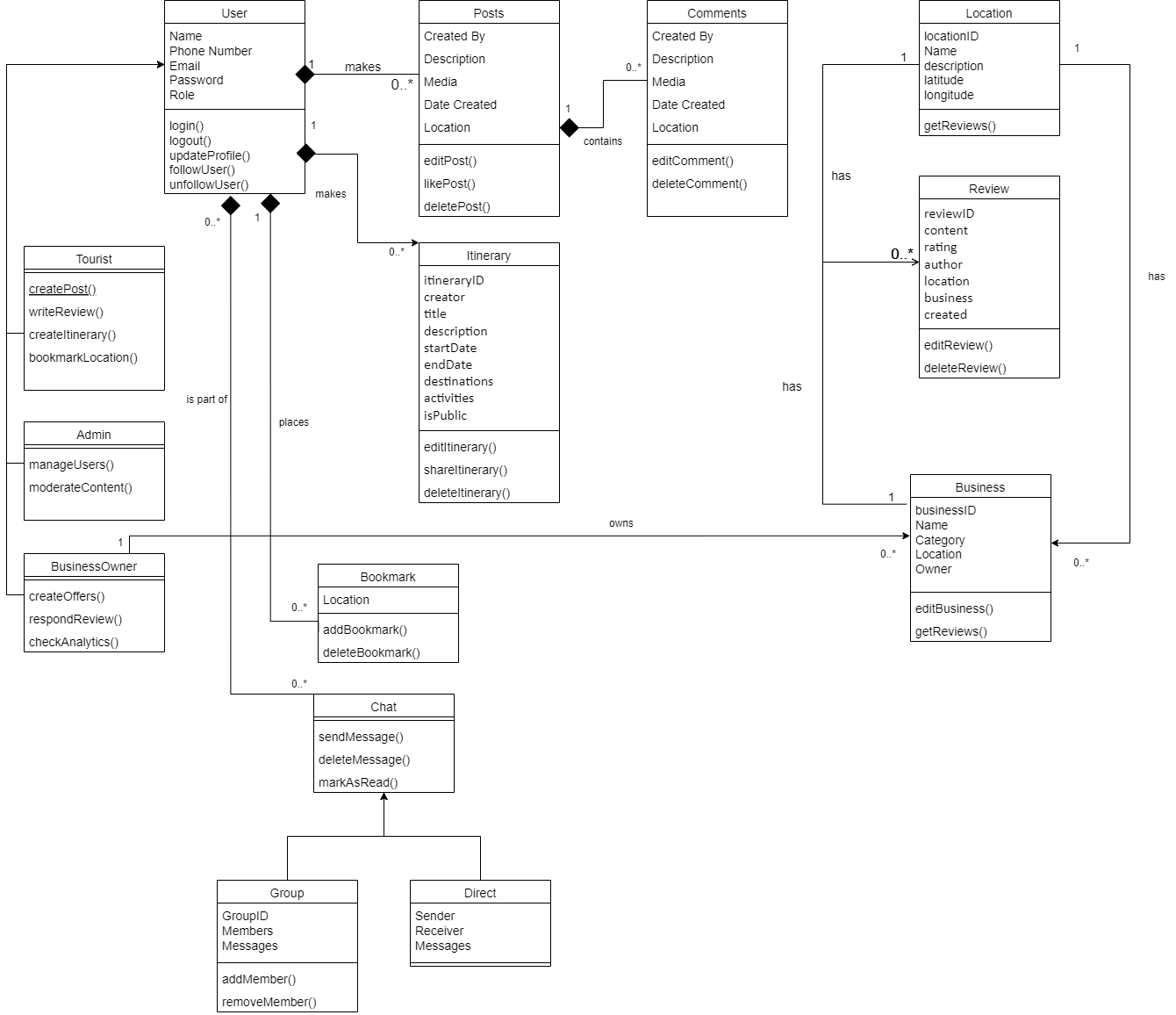
* + 1. Report Content

|  |  |  |
| --- | --- | --- |
| **Identifier** | | UC-020 |
| **Purpose** | | Allow users to report content that violates platform rules, such as offensive posts, spam, or illegal activities. |
| **Pre-condit ions** | | 1. User is logged into the app. 2. User has encountered content they deem inappropriate. |
| **Post-condi tions** | | 1. Reported content is flagged for review. 2. Admins review the content and take appropriate action (e.g., content removal, user warnings). |
|  | | |
| **Step #** | **Typical Course of Action** | |
| 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. | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
|  |  | |
| **Step #** | **Exception Paths** | |
|  |  | |

# Class Diagram

## Diagram

[The user roles should be modelled in a way that rights can be granted and revoked on need basis.]



## 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).

* 1. *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.

* 1. *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.

* 1. *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.

1. 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.

1. 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.

1. Chat

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

* 1. *Direct*

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

* 1. *Group*

This subclass represents communication between multiple members.

1. Posts

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

1. Comments

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

1. Location

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

1. Review

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

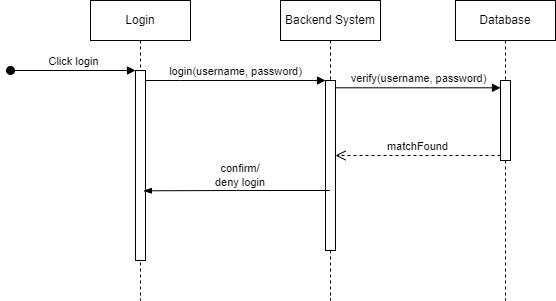
1. Business

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.

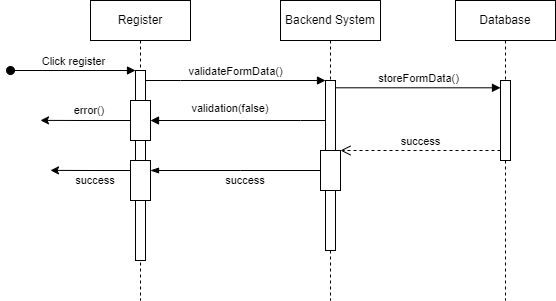
# Sequence Diagrams

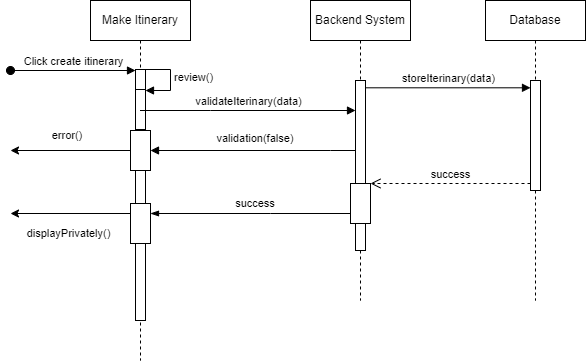
The sequence diagrams must have classes from the Class diagram you created earlier in this document. These diagrams must show the interactions among the system classes. In all of the sequence diagrams that you have created, does not contain classes above.]

## Login

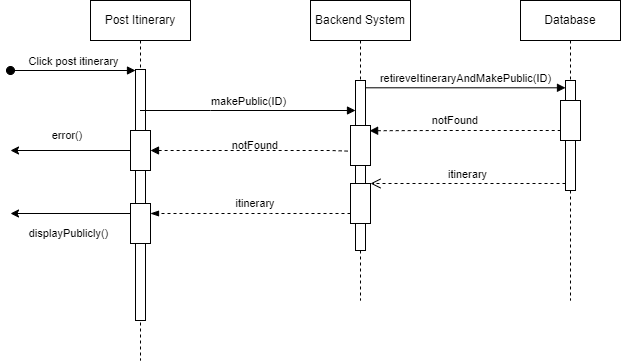


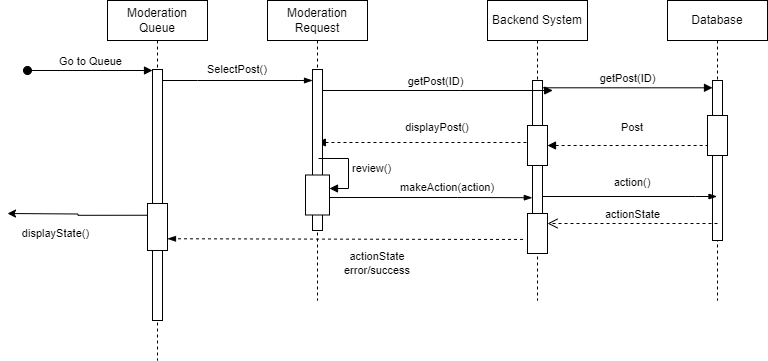
## Register



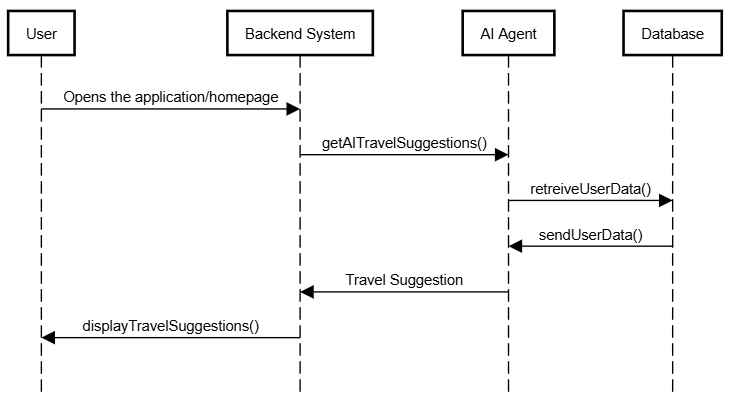


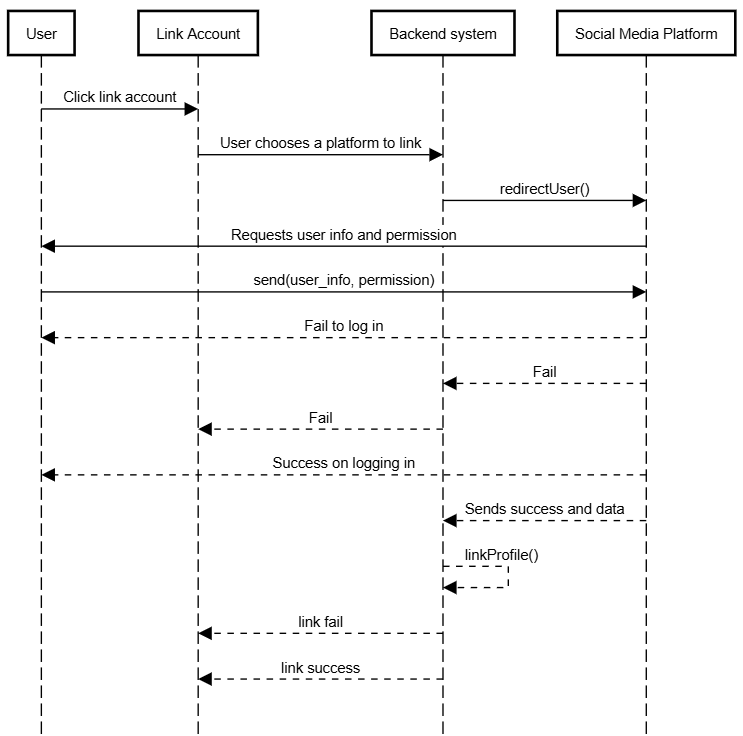
## 5.4 Posting Itineraries

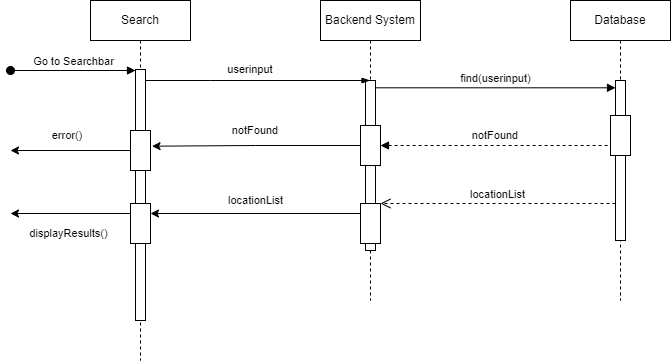


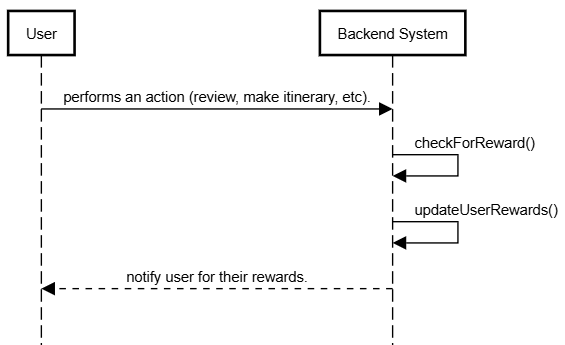


## 5.6 AI-Based Travel/Post Suggestions

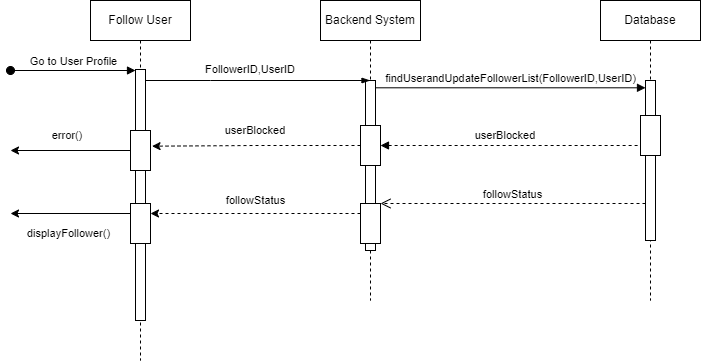


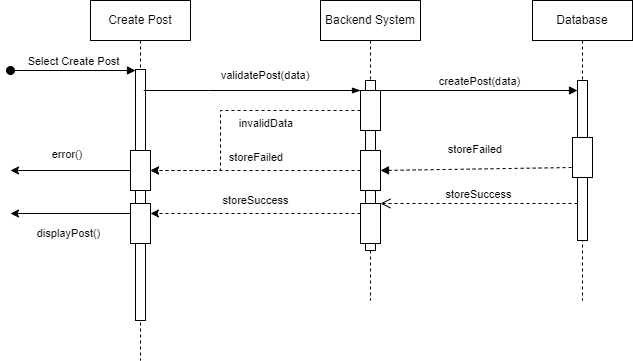




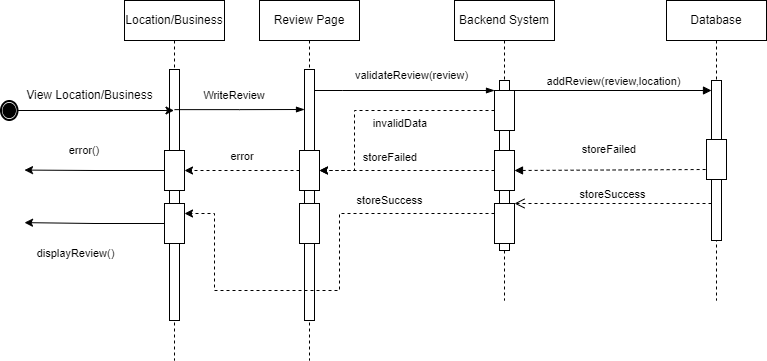


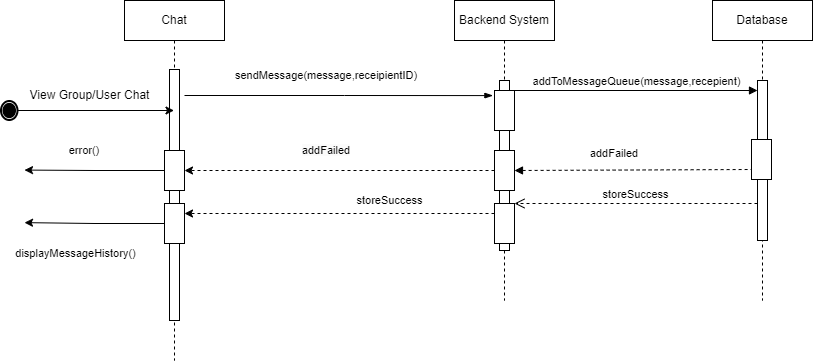
## 5.10 Adding Friends



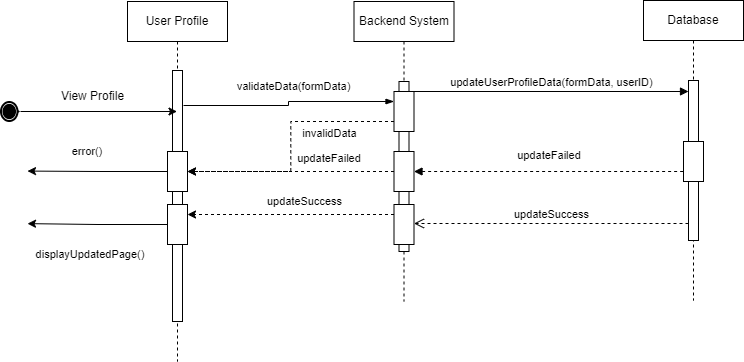


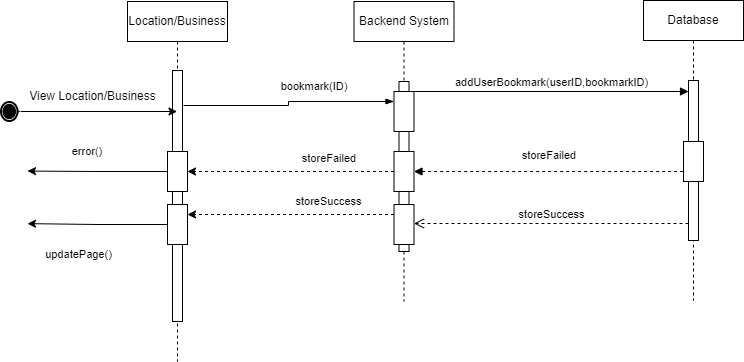
## 5.12 Writing a Review



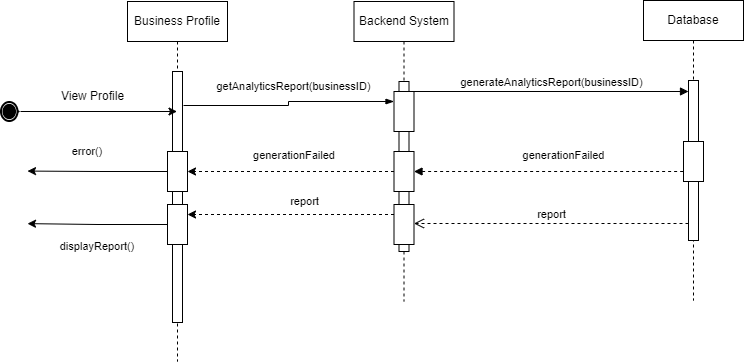


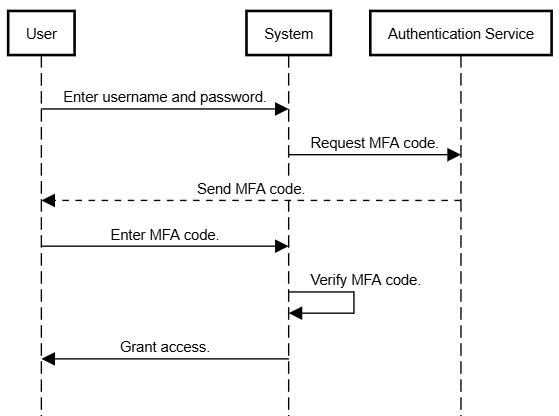
## 5.14 Update User Profile

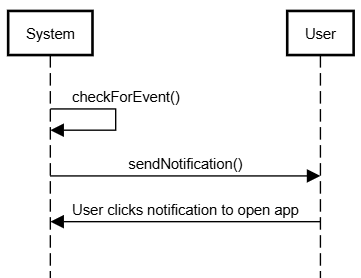


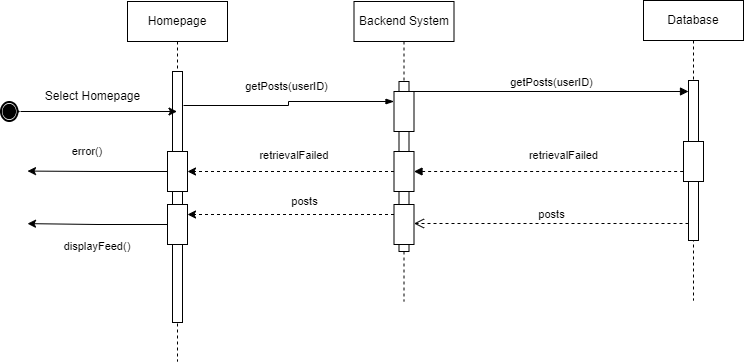


## 5.16 View Business Analytics

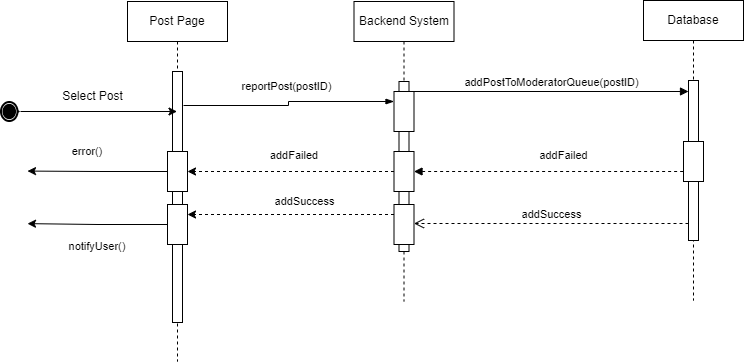








## 5.20 Report Content



# State Diagrams

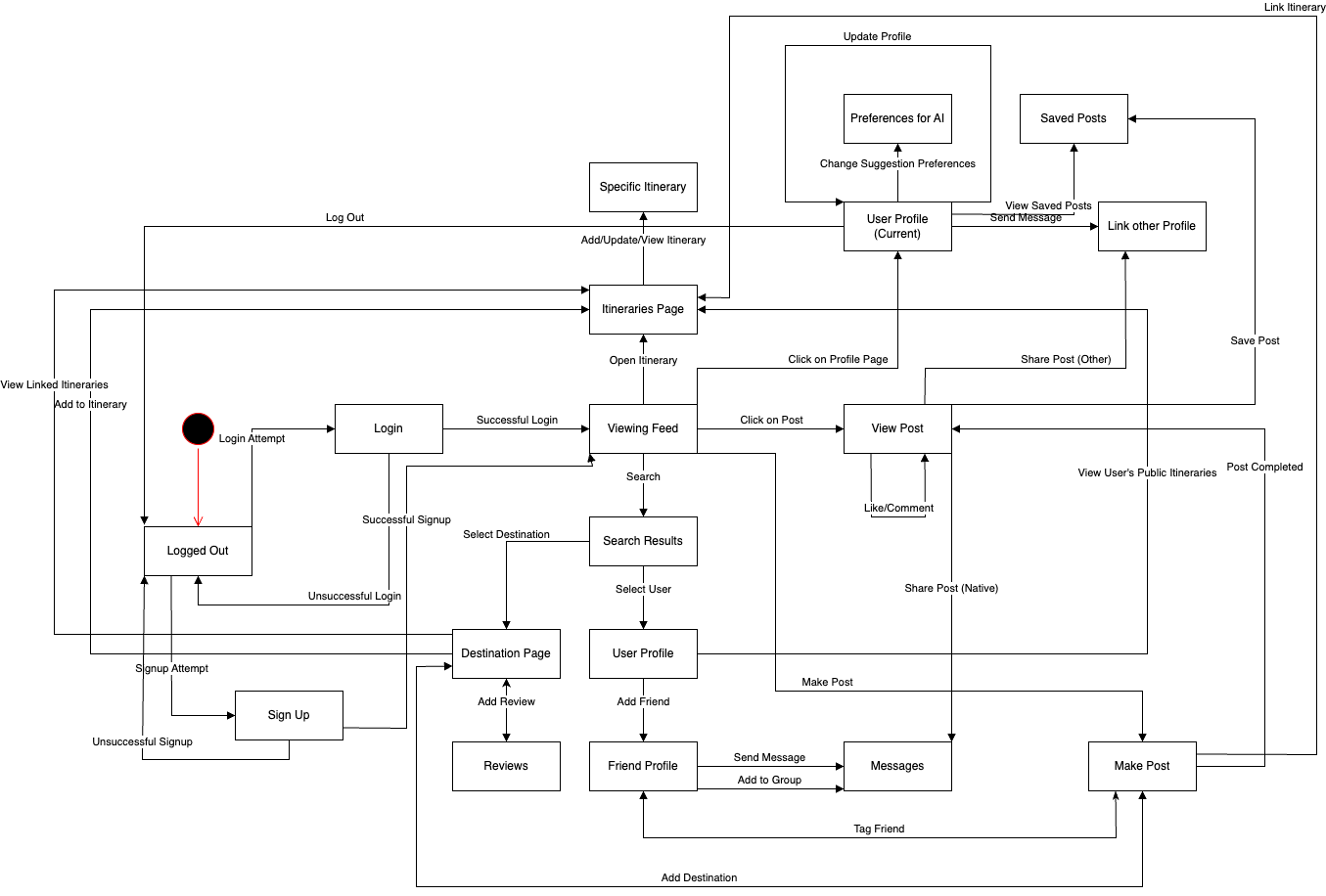
## Diagram details

The following diagram represents the flow and transitions of the Tourist user class. Initially the user would be in a Logged Out state, the user may then attempt to sign up (new user) or login (returning user). In the case of both, an unsuccessful attempt would take the user to the initial state and a successful attempt would take the user to their viewing feed. The viewing feed state would allow for a user to browse public posts by all users matching their interests, private posts by friends, suggestions for itineraries and destinations, friend suggestions and more. The user may perform actions such as searching, which may be for users - allowing for a user to view a user’s content (if public), add them as a friend and if successful, send them messages or add them to a group - and may be for destinations; allowing for users to view specific destinations, read and write reviews, view public itineraries that include said destination and add them to a specific itinerary (new or existing).

The users may also choose to interact with a particular post, liking, commenting, saving or sharing. They may also select their own profile, update their preferences - used by the AI for suggestions and matching - and may also update their profile and change settings while also being able to view previously saved posts. From the User Profile State, the users would also be able to view, update or create new itineraries. From here, the user may also log out.

From the View Feed state, the users may also choose to make a new post in which they can tag friends, add destinations and link itineraries.

## Diagram



|  |  |
| --- | --- |
| **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. |

|  |  |
| --- | --- |
| **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 |  |
| Luqman 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 2,4,5,6 |
| Shahrez Aezad | Section1, 6, 7 |
| Mohammad Haroon Khawaja | Section 3,4,5 |
|  |  |