



Dhirubhai Ambani
Institute of Information and Communication Technology

IT314 - Software Engineering

Lab 6

Date of submission: 22-03-2023

Due date of submission: 22-03-2023

Group Members:

1. 202001063 - Bhalodiya Hem Pareshbhai
2. 202001066 - Japan Vijay Bhatt
3. 202001068 - Dhrupal Kukadia
4. 202001078 - Shashank Didwania
5. 202001081 - Ronit Jain
6. 202001083 - Patel Vedant Vipulbhai
7. 202001093 - Parmar Dhruv Jayeshbhai
8. 202001106 - Hardi Sanghani
9. 202001115 - Aditya Kothari

Project Name: Location Sharing System

Contents:

1. Domain Analysis Model
2. Boundary, Entity and Control Object
3. Sequence and Class Diagram
4. High Level System Design

Domain Analysis Model:

A domain analysis model for the “Location Sharing System” would typically include the following elements:

- **User Roles:** There are different roles for a user in a system such as end-users, administrators, moderators, etc. Each of them having different permissions and functionalities.
- **Location Data:** The system would define the type of location data that it will collect and process, such as GPS coordinates, street addresses, or landmarks.
- **Privacy and Security:** The system would provide privacy and security to the users such as data encryption, user authentication and access control.
- **Communication Protocols:** The system should define the communication protocols that it will use to receive, store and retrieve location data, such as RESTful APIs or MQTT.
- **Integration:** The model should define the integration points. The system will use an integration of systems such as mapping services, social networks, Emergency SOS, etc.
- **User Interface:** The model should define the user interface that the system will provide to end-users, i.e. a mobile application here, and the features that will be available, such as real-time location tracking, notifications, alerts, etc.
- **Performance and Scalability:** The model should define the performance and scalability requirements of the system, such as response times, data throughput, and user concurrency.
- **Data Analytics:** The model should define the data analytics capabilities of the system, such as location-based insights, user behavior, and trends.

Overall, the domain analysis model for a location sharing system should be comprehensive and cover all the important aspects of the system, from user roles and privacy to performance and analytics.

Identify Boundary, Entity and Control Object:

1. Boundary Object:

- a.** Boundary for the application would be the interface of mobile application which separates the system and the user.
- b.** It would include different features in the application such as friends meetup, navigation, live location sharing etc.

2. Entity Object:

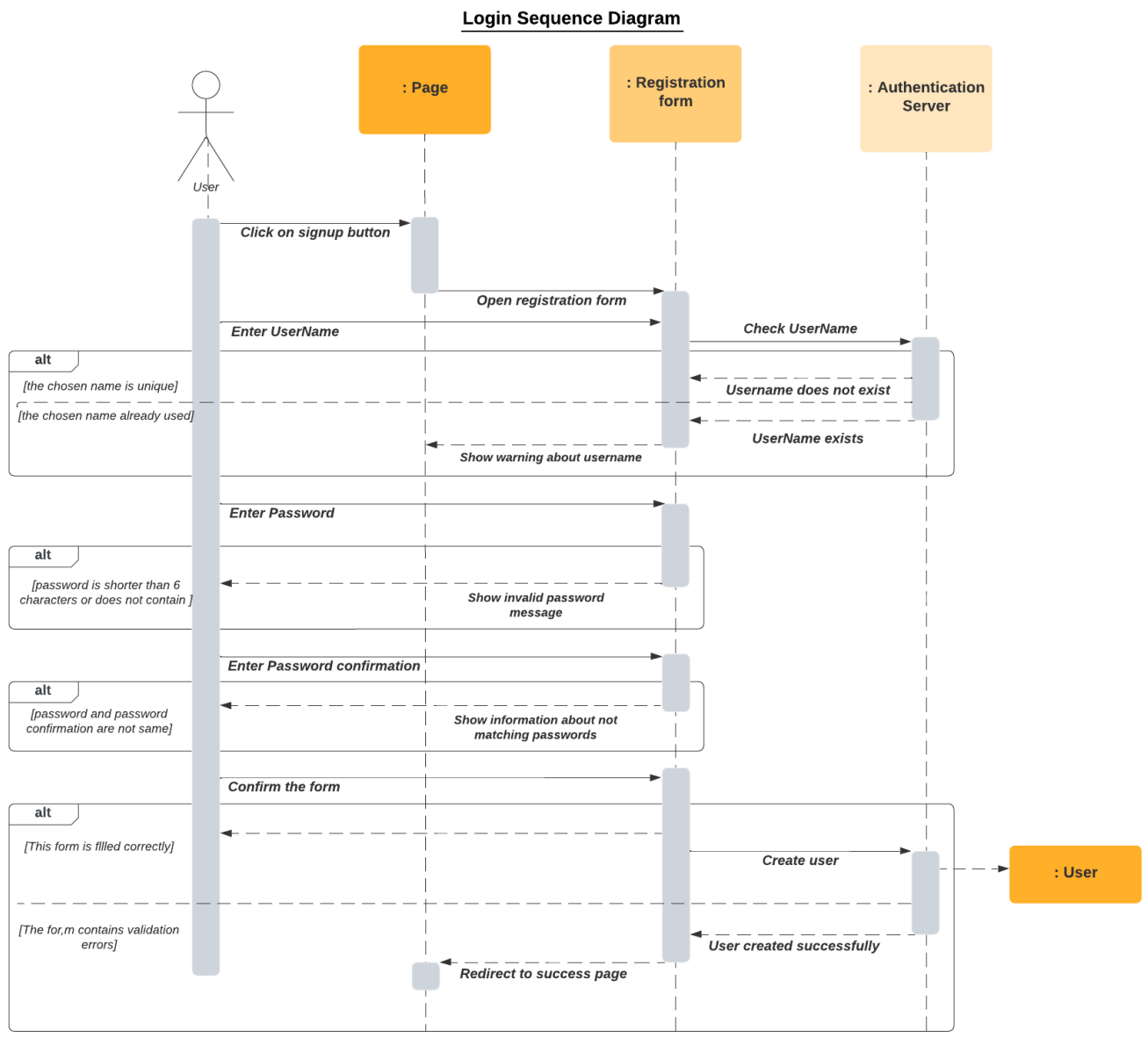
- a.** There are several entities associated with the application. This includes databases of saved locations, restaurants, hotel, sos places.
- b.** It would also include database of friends and family, database of sos contacts and api of google maps
- c.** Additionally the entities would include the hardware and software supporting the system.

3. Control Object:

- a.** Control objects would be the options and features that generate data or control the database features.
- b.** Database for friends and family contacts will be accessible through phones contacts.

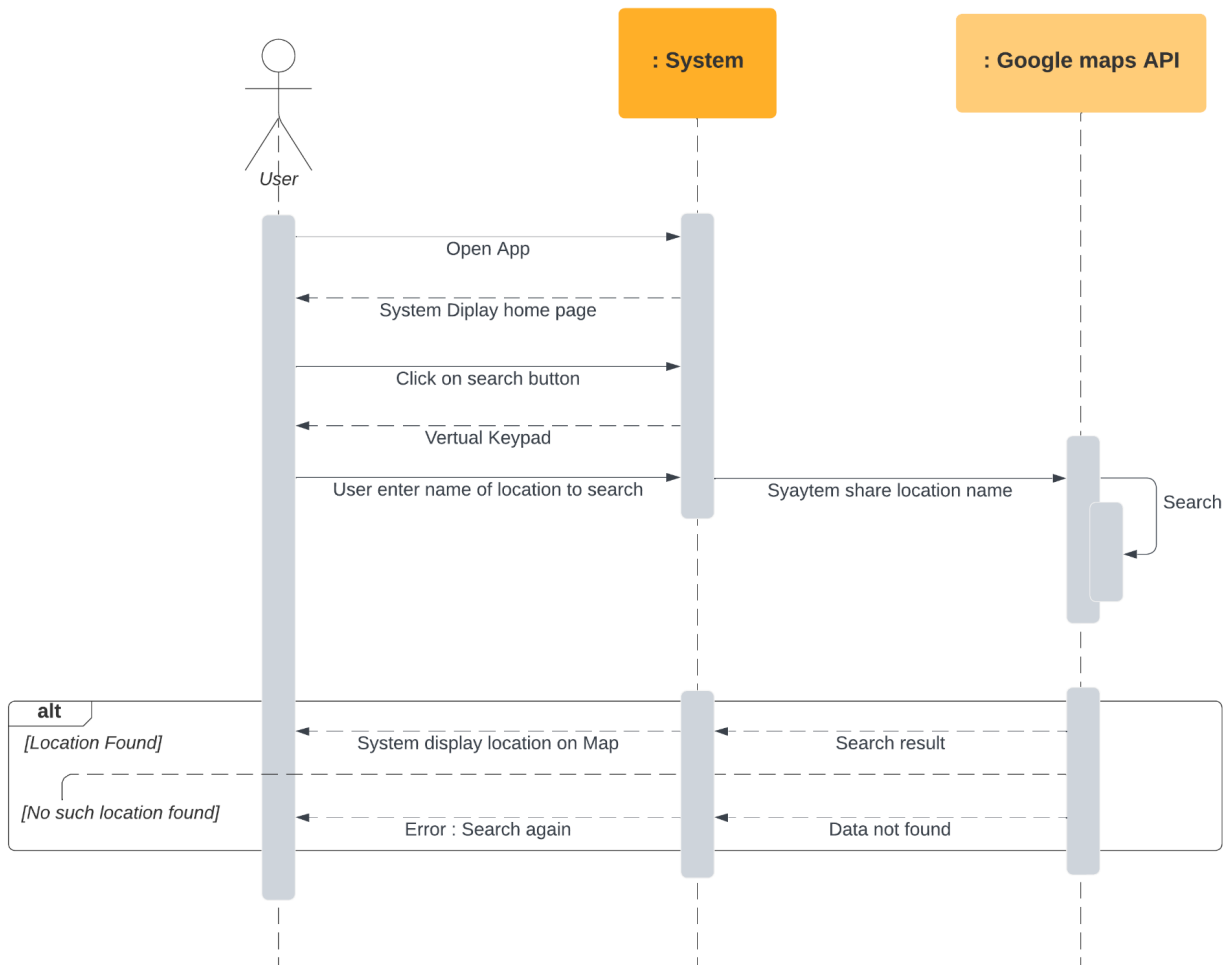
Sequence Diagram:

1. Login

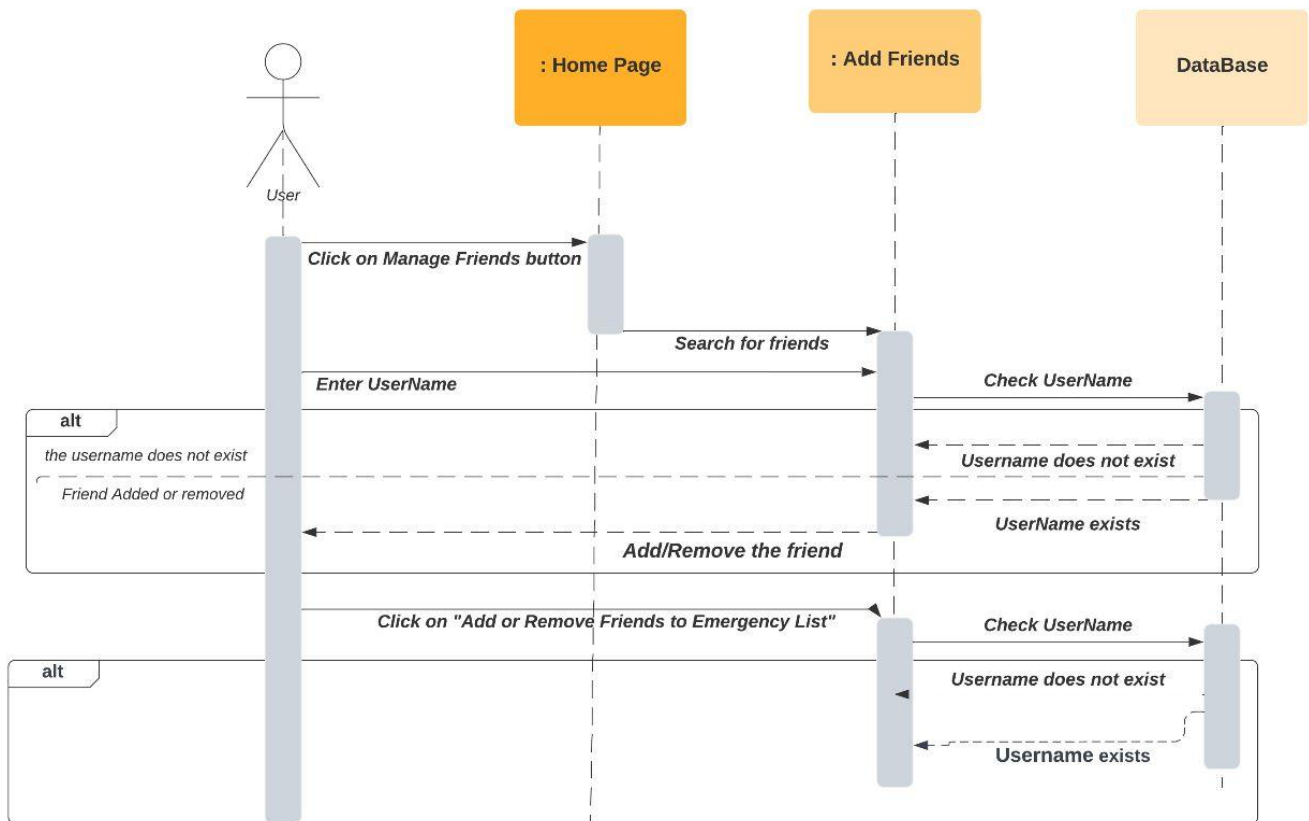


2. Search Location

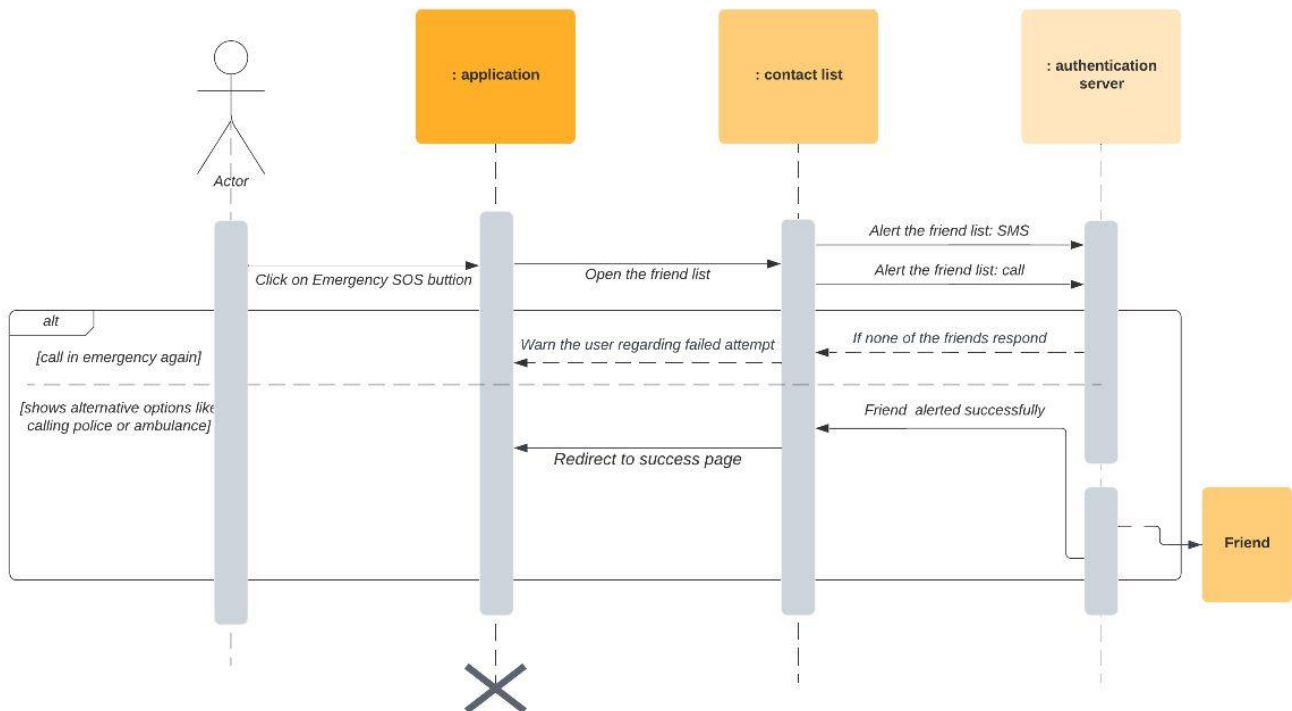
Search Location Sequence Diagram



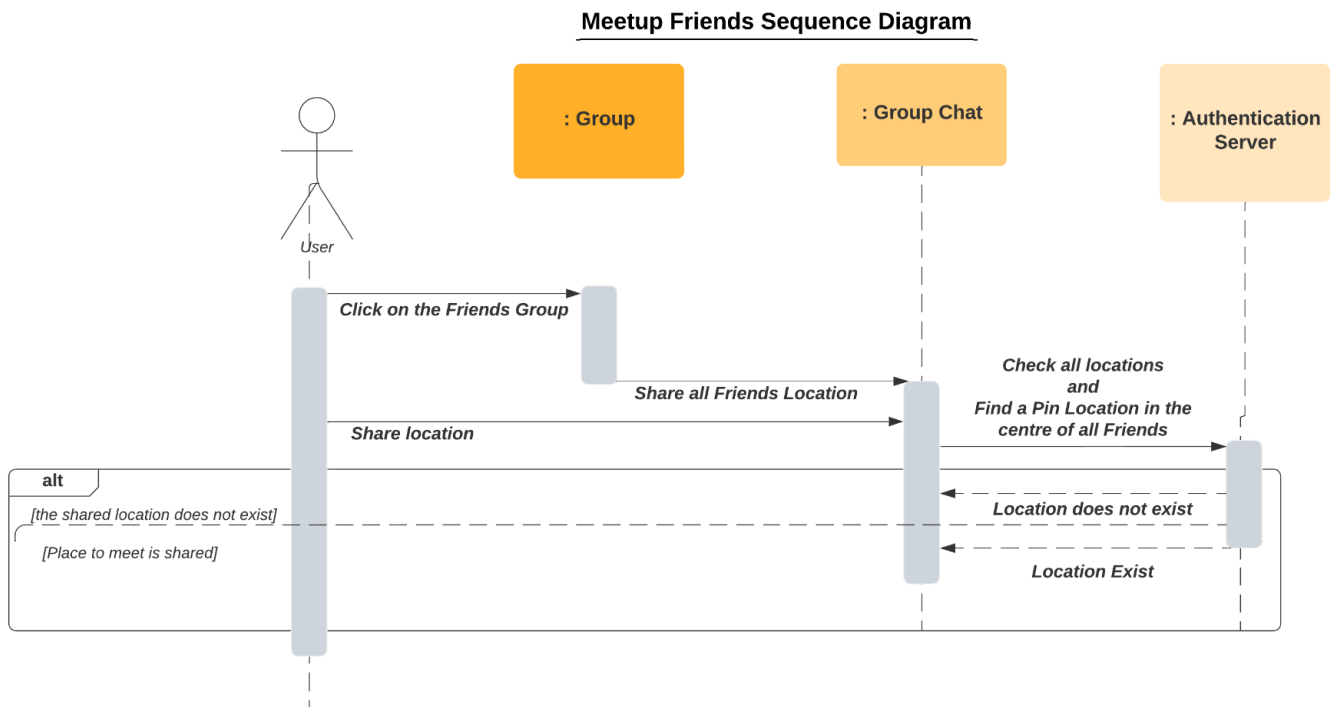
3. Manage Friend



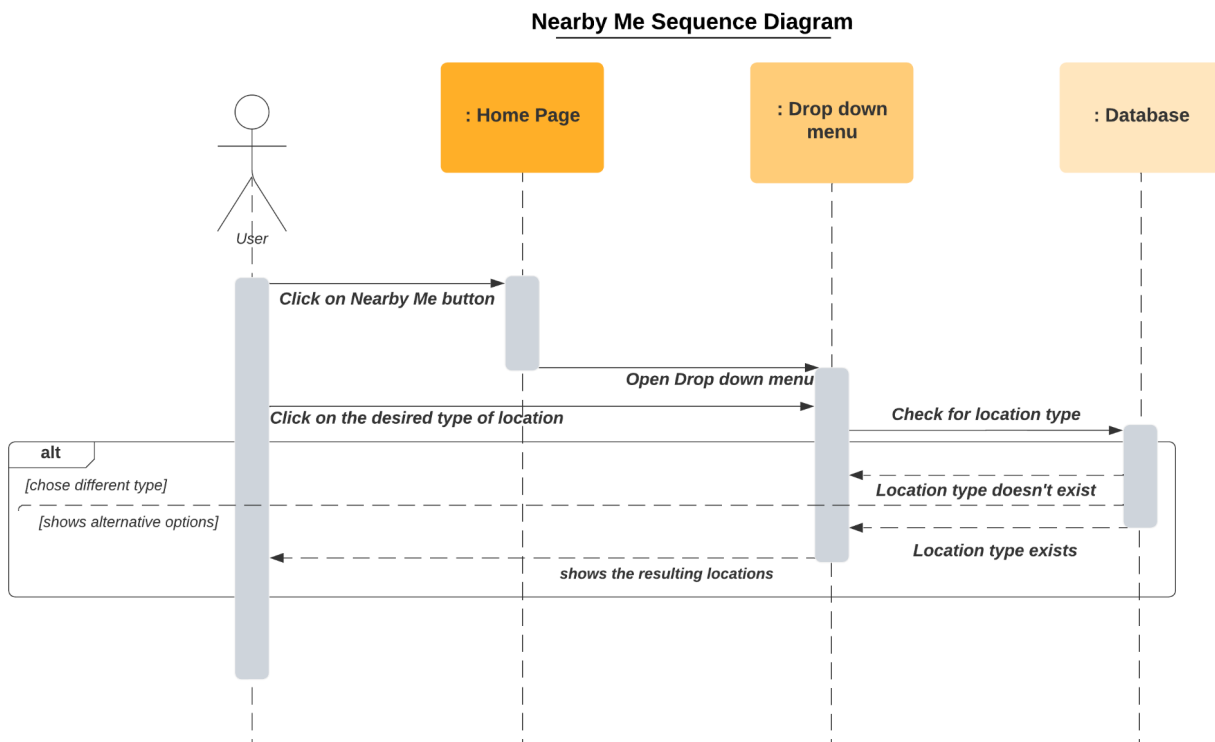
4. Emergency SOS



5. Meetup Friends

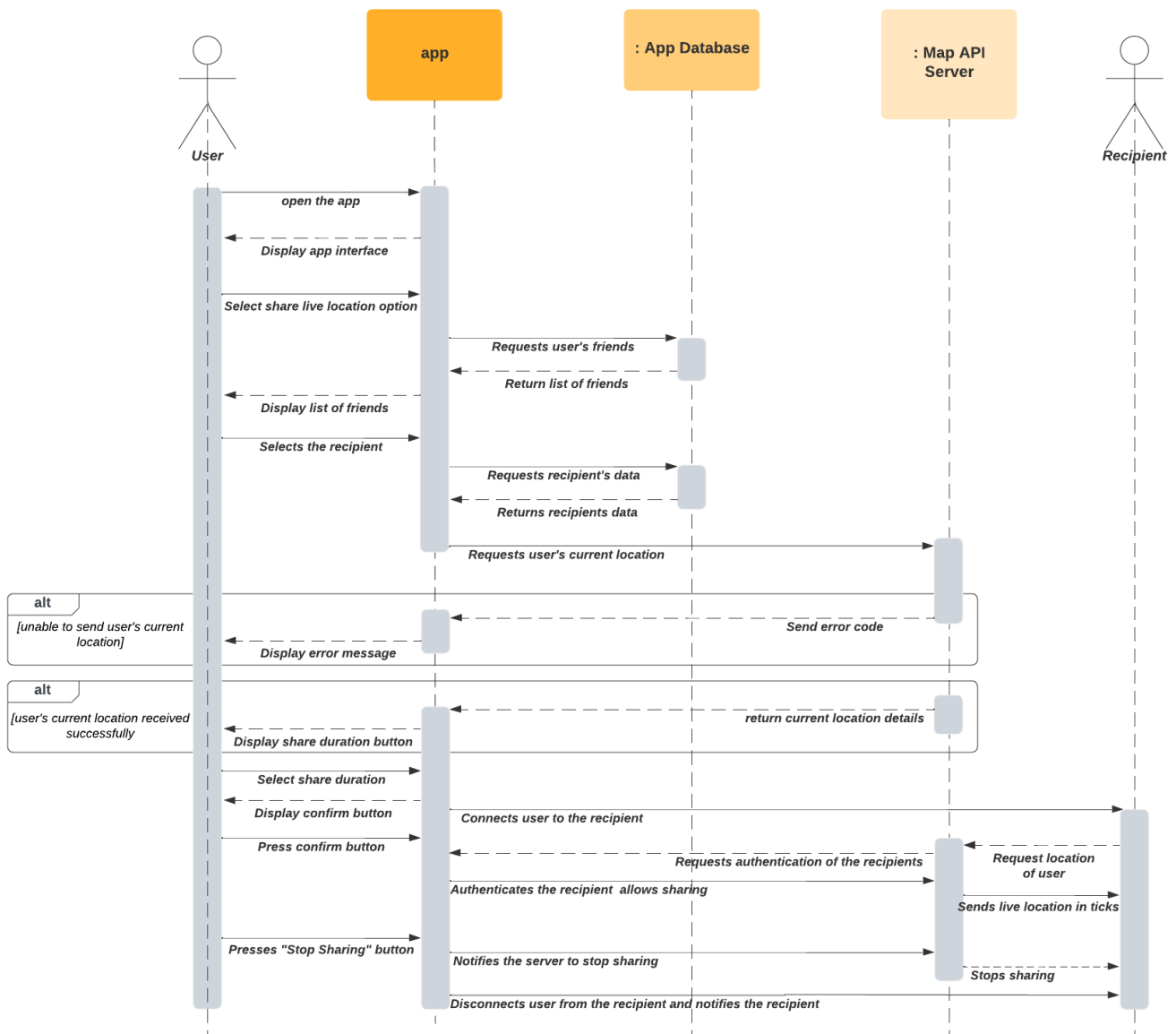


6. Nearby me

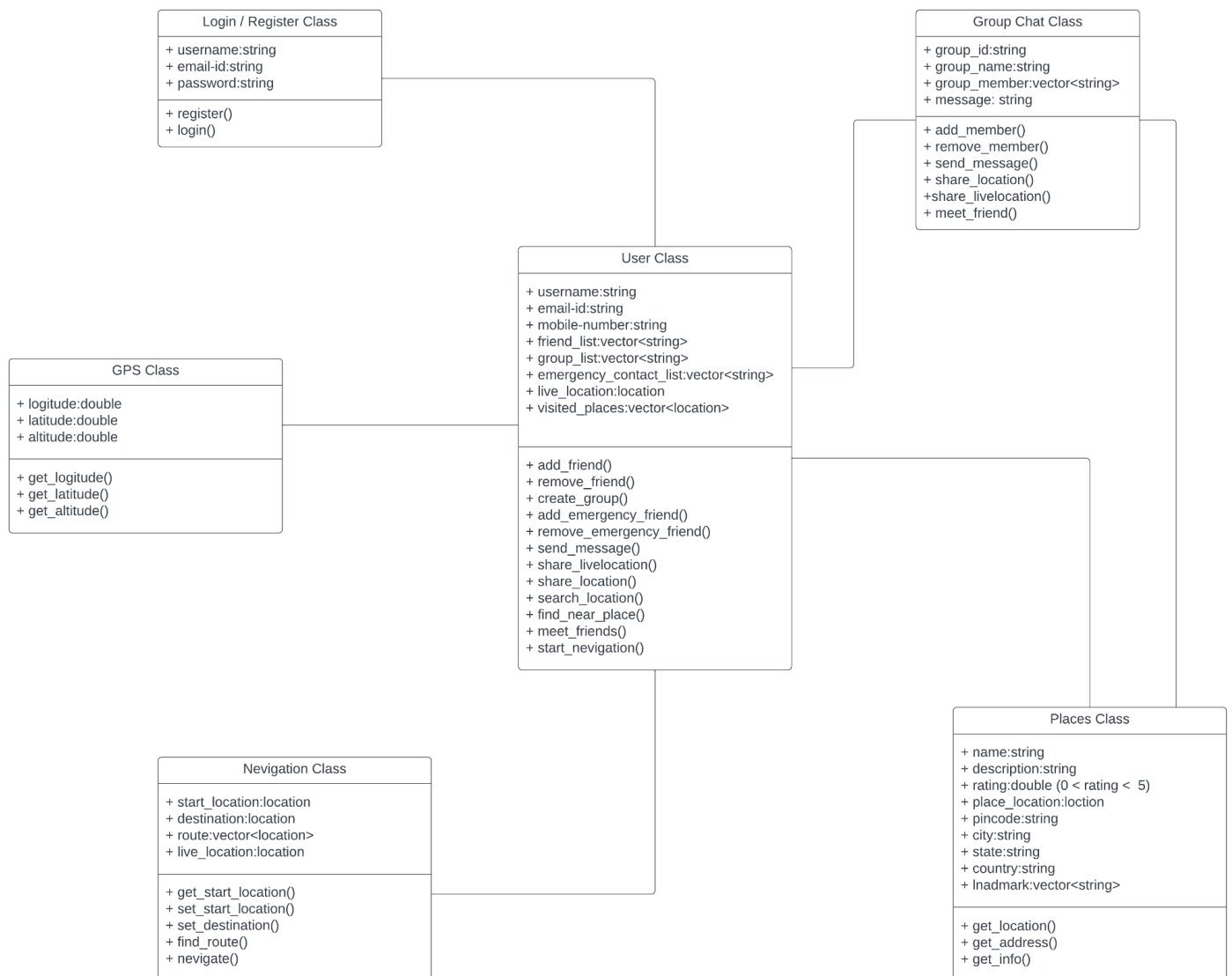


7. Share Live Location

Share live location Sequence Diagram



Class Diagram



High Level System Design

Architecture:

A client-server architecture would be suitable for this project. The client would be the mobile application that users interact with, and the server would be a centralized location where user data is stored and processed. The client application would send location updates and other data to the server, and the server would process this data and send it back to the appropriate users. The communication between the client and server would be through RESTful API or similar web services. The use of cloud-based technologies such as AWS or Google Cloud Platform could provide the necessary resources and scalability for the project. Overall, this high-level design provides a solid foundation for a location sharing project that is scalable, secure, and user-friendly.

This architecture consists of a presentation tier, an application tier, and a data storage tier.

Presentation Tier:

It is responsible for handling user interface and user input. This tier includes the following subsystems:

- **User Interface:** This subsystem includes the app interface that allow users to interact with the app.
- **Client-side Logic:** This subsystem includes the Dart and other client-side code that handles user input validation and other user interactions.

Application Tier:

The application tier is responsible for handling the business logic and processing user input. This tier includes the following subsystems:

- **Controller:** This subsystem is responsible for managing the flow of data between the user interface and the data storage tier.
- **Business Logic:** This subsystem is responsible for implementing the business logic of the app, such as user registration, friends list, and managing sos.
- **Security:** This subsystem is responsible for implementing security features, such as user authentication and authorization.

Data Storage Tier:

The data storage tier is responsible for storing and managing data. This tier includes the following subsystems:

- **Data Access Layer:** This subsystem is responsible for accessing and retrieving data from the database.
- **Database:** This subsystem is responsible for storing all of the website's data, such as user information, forum data, topic data, post data, tag data, and ranking data.
- **Security:** The system should be secure and protect sensitive employee data. Access to the system should be restricted based on user roles and permissions. The system should also be backed up regularly to prevent data loss in case of a system failure.
- **Integration:** The system should be able to integrate with other HR systems and tools, such as applicant tracking systems, performance management software, and benefits administration platforms.

Subsystems:

1. User Interface (UI) subsystem:

This subsystem would consist of the user-facing components of the application, including the login and registration screens, the home screen where users can select contacts to share their location with, and the settings screen where users can customize their preferences. The UI subsystem would be responsible for handling user input and presenting information in a clear and intuitive manner.

2. Location Tracking subsystem:

This subsystem would be responsible for tracking the user's location using GPS or other location-based technologies. The subsystem would need to be efficient and accurate, as well as capable of handling location data from multiple users. The location data would be stored in a database and presented to the user in a meaningful way, such as on a map or in a list of recent locations.

3. Communication subsystem:

This subsystem would be responsible for enabling communication between users in the application. It would handle sending and receiving location updates and other messages between users. The communication subsystem would need to be fast and reliable, as well as secure to prevent unauthorized access to user data.

4. Database subsystem:

This subsystem would be responsible for storing and managing user data, including location data, contact information, and preferences. The database would need to be scalable to handle large amounts of data and support fast retrieval and updating of data. The data would need to be secured using encryption and other security measures to prevent unauthorized access.

5. Security subsystem:

This subsystem would be responsible for ensuring the security and privacy of user data. It would include measures such as authentication and encryption to prevent unauthorized access to user data. The security subsystem would need to be constantly updated and monitored to protect against new security threats.