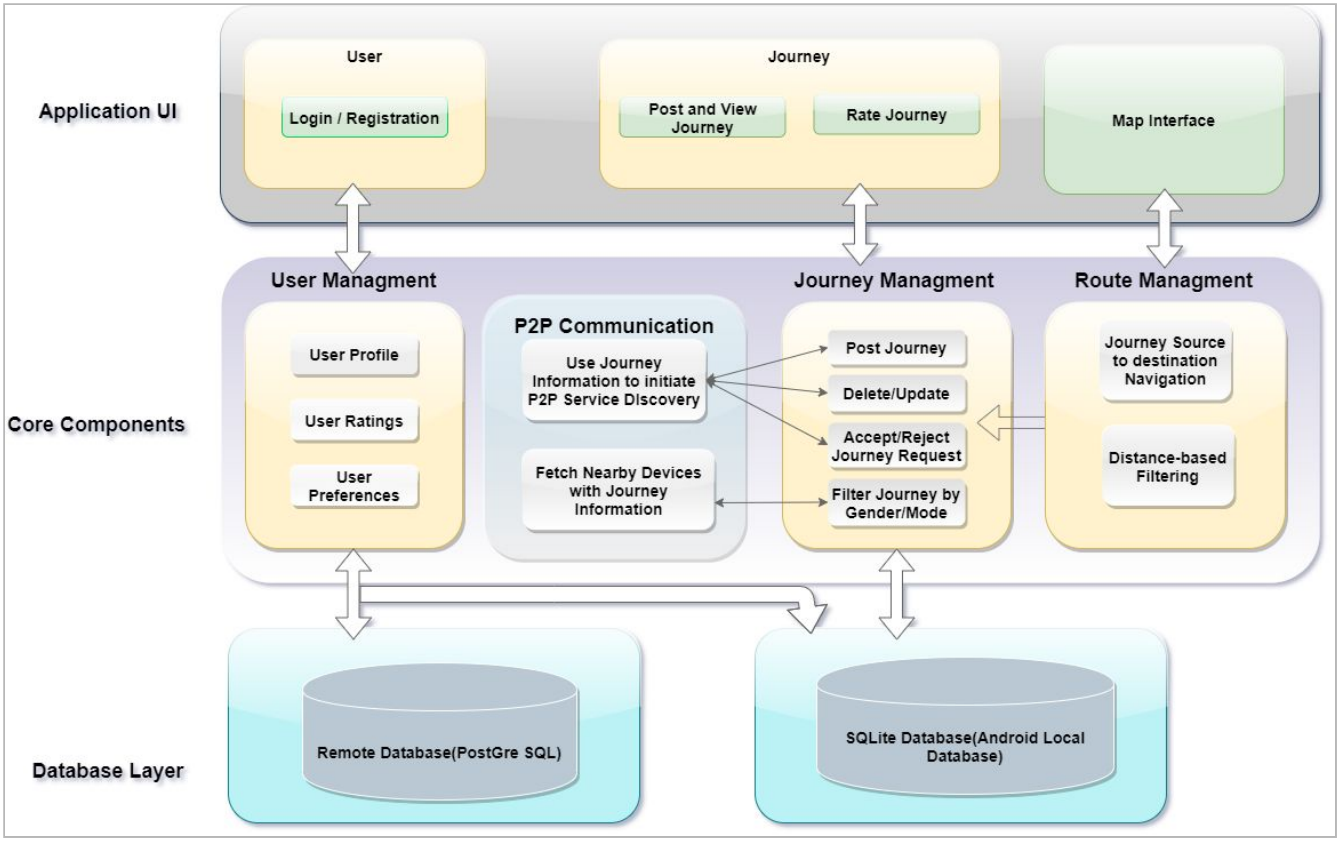
FUNCTIONAL ARCHITECTURE



Core components of Journey Sharing App:

1. **Application UI**
    a. **User**

    Various activities like login, register, find riders, post about a ride, give feedback and ratings are included in this module. The user interface includes:

    - Login/Register
        - This is done by using the registration activity and login activity. Basic information like name, email address, gender, contact number are taken as input for registering the users.
    - Post rides
        - A user can post about the rides he or she is planning to take with information like time, start and end journey locations and preferences about gender and mode of travel.
    - Show rides
        - A peer who wants to travel, can check for some available options and depending upon the preferences, a list of nearby travelers will pop up.
    - Rate and give feedback to fellow peers
        - Fellow travelers can rate each other after the trip ends.

    b. **Map Interface**

This interface is used for displaying a map that holds information like the start journey location and end journey location. It also shows real-time navigation.

2. **Core Components**
    a. **User Management**

All user-related backend API's from saving and registering user to viewing and posting ride information either on local database or server are included in this module.

- User Profile
    - A basic user profile is created on registration and basic information of the user is stored.
- User Ratings
    - Users can rate each other out of 5 after the trip is complete.
- User Preferences
    - This is used to filter out rides based on user preferences. Preferences include modes of travel as well as gender.

b. **Journey Management**

This module is responsible to post rides and select rides depending upon the filters. Option to delete and update the posted ride is available.

- Post Journey
    - Information like start and end journey location along with time and preferences to show this ride accordingly are saved.
- View Journey
    - This fetches and shows the rides on the basis of start and end location and other user preferences.
    - Preferences to show this ride accordingly are saved.
- Search/Filter Journey
    - Searching journey depending upon the preferences, location and time.
- Select Journey
    - Selecting the journey depending upon the similar journey information (start and end location), gender and mode of travel.
- Update/Delete Journey
    - Update journey information like time, preference, location or delete journey.

c. **Route Management**

This module will manage the route information.

- Route Navigation: Shows the selected route for navigation on Maps API.
- Distance-based filtering is implemented.

3. **Database Layer**
    a. **Local Database**
        i. A local database is maintained on the user's device, where the user-specific data is stored in the absence of any internet connectivity. During the peer discovery process, the local database is used to provide user data (name, gender and contact number if provided.)

b. **Remote Database**
    i. A remote database is maintained on the cloud server. When a new user registers on the application, the registration data is stored on the remote database. At the time of login, the user details are fetched from this database.

4. **P2P component**
    a. Peer Discovery: Using this interface, the user is able to look for other users who are around him and will be going to destination around the user's destination.
    b. Communication Services: Once the users or peers establish contact and accept the rides, they are able to exchange information that is encrypted.