

---

## Block Diagram and API of ItineraryBuddy

---

Group MS3\_3

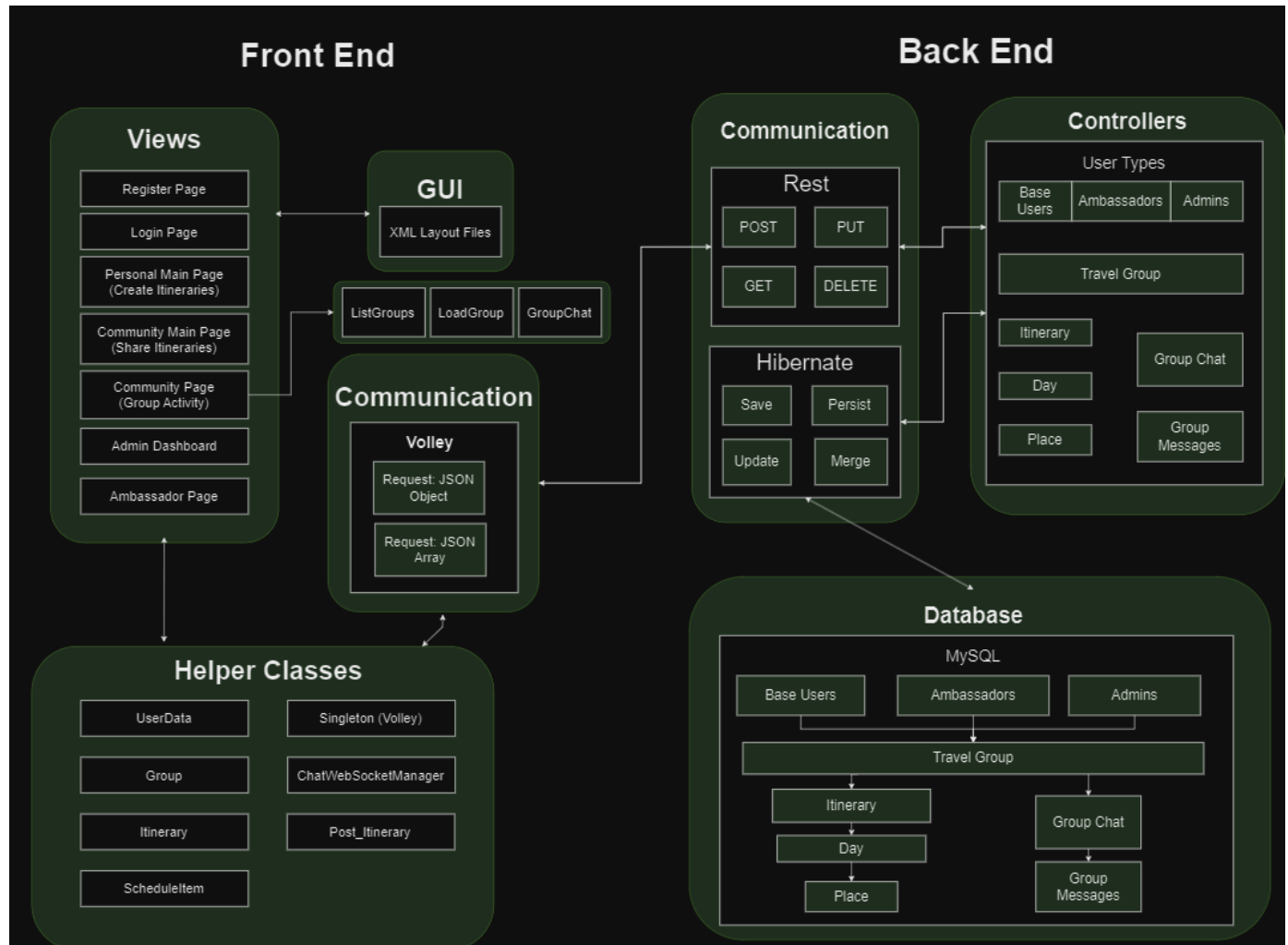
Aina Azman: 25% contribution

Justin Sebahar: 25% contribution

Noah Pater: 25% contribution

Jackson Collalti: 25% contribution

# Block Diagram



# Design Description

Use this third page to describe complex parts of your design.

## Frontend

**User Login/data (User, Admin, Ambassador):** Upon login, a GET request is made for the User object given the entered credentials. The response is then saved to a static data field in the UserData class, where there are various getters to retrieve the many data elements of the user, such as usernames, groups, etc, in various parts of the program.

**Personal Mainpage (User, Ambassador):** After login, this will be the first page the users see. This page allows users to create personal itineraries with the following elements:

- EditText: Destination
- EditText: StartDate
- EditText: EndDate
- Button: CreateItinerary

After creating an Itinerary, it will be assigned eight digits of unique self-generated tripCode and the number of days obtained by deducting the StartDate from the EndDate. All the stated information will be stored in the back end. Users will then be able to click on the created itineraries to update the schedule of the day, which will receive the following elements in ArrayList:

- EditText: Time
- EditText: Place
- EditText: Note

**Groups (User, Ambassador):** Upon opening the group page on the dashboard, a helper method of UserData is called to retrieve the list of group IDs a user is a member of from the UserData JSON. It then makes a GET request for each group to instantiate a Group object for each group the user is in. It is then added to a list to be displayed, and the user can click into the group to see the details, make changes, chat, etc. Data is updated as needed by PUT requests. All requests are made via the TravelGroupController.

**Community Mainpage (User, Ambassador):** This page allows users to post their itineraries to be shared with other users. The itineraries to be posted will obtain the necessary information from the personal main page data to be shared with others. This page utilizes WebSocket for real-time communication, efficiency, and scalability.

## **Backend**

**Itinerary Class (Itinerary, Day, Place):** Itinerary objects have their basic values and a unique trip code used to share your itinerary with friends and family. Itineraries also incorporate a list of days to track where one will be on a specific day, respective to the spot in the array. Day objects also incorporate their list of places to visit on that given day. Itineraries, days, and places all have their respective controllers and repositories.

**User Class (User, Ambassador, Admin):** Users are associated by their unique username, email, password, state and city. They have variables for the number of posts and likes they have. They have a list of group codes that allows them to join itineraries with other users. Ambassadors are the same as users, except they have permission to create groups for users. Admins have permissions to everything on the platform; they can revoke permissions and ambassador titles, delete and update itineraries/groups, and change a user's information.

**TravelGroup Class:** TravelGroup objects are the group for an itinerary. An ambassador will create it and be saved as the group creator and can add users to said group in a list. Each group also has a name, destination, and code.

# Tables and Fields

