

# Block Diagrams & API

App Name: Dinder

Team - SD 322

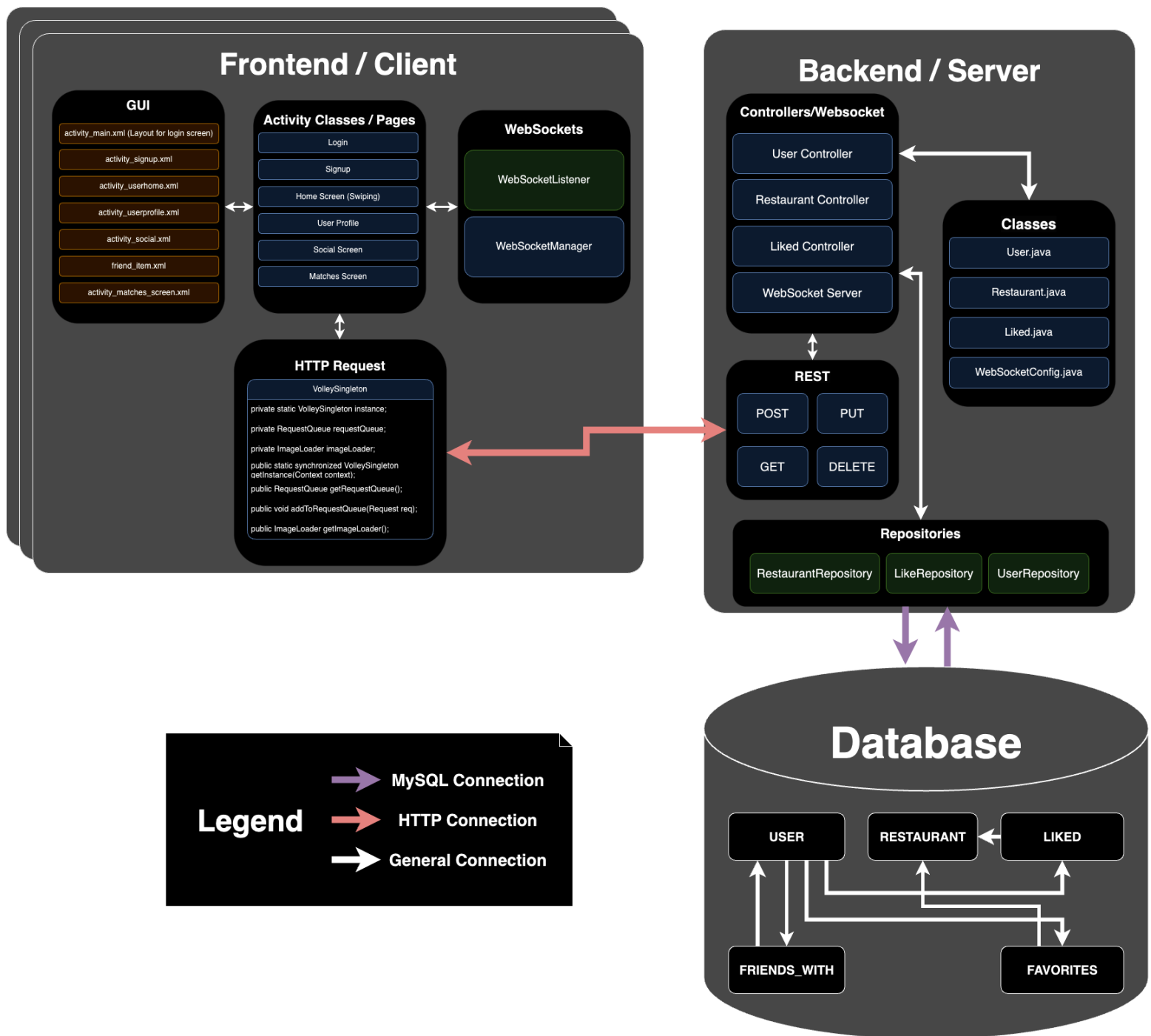
Ethan: 25%

Eli: 25%

Lali: 25%

Jesse: 25%

# Block Diagram



# Design Description

## Frontend (currently implemented)

**Login:** On the log-in page the user can log-in by entering their username and password, while the log-in request is pending, a loading indicator or symbol will be displayed over the screen.

**SignUp:** If the user doesn't have an account, they can sign up with their info on the sign-up screen.

**Homepage:** From the home page the user will be shown a restaurant which they can either swipe left on to dislike or swipe right on to like. The user can also tap on the restaurant's image to go to its profile. If the user doesn't want to immediately start swiping, they can navigate to the other screens via a navigation bar at the bottom of the screen. While on the home page if the user finds a match, a notification will pop up on the screen informing them of it.

**Social Screen:** This screen shows the user the other users they are friends with

**Matches Screen:** Shows the user what restaurants they have been matched with based on their likes, or their groups' likes

**User Profile Screen:** This screen allows the user to update their preferences, and view their recent likes

**Restaurant Profile Screen:** This screen provides the user with more specific information about the restaurant they tapped on.

## Backend (Currently implemented)

### Communication

The backend uses HTTP requests to update the database based on information sent to the given mappings' URLs includes:

**Post:** Send information on an item to be added to the database.

**Get:** Request information, often with an identifier for the specific item requested from the database.

**Put:** Send information to update a specific item in the database.

**Delete:** Send an identifier to delete a specific item from the database.

The backend also utilizes a websocket connection to allow for two way communication. The websocket has four main functions:

**Groups:** The websocket is used to group together users into Dinder sessions

**Likes:** Information about liked restaurants are sent through the websocket and added to the liked database

**Matches:** When all users in a group have liked a restaurant a websocket message is sent to notify the group of a match

**Favorites:** Restaurants that users have been matched with are tracked for later display

### Controllers

#### User Controller:

Manages user authentication and login.. Allows for the creation of new users and modification of existing ones. This controller also handles friend requests

#### Restaurant Controller:

Manages restaurant-related data, such as information and menus. Accesses the external Yelp API and searches for specific user requests and limitations

## List of Tables and their fields

