Design Document for CyDine

Group 2_fatema_1

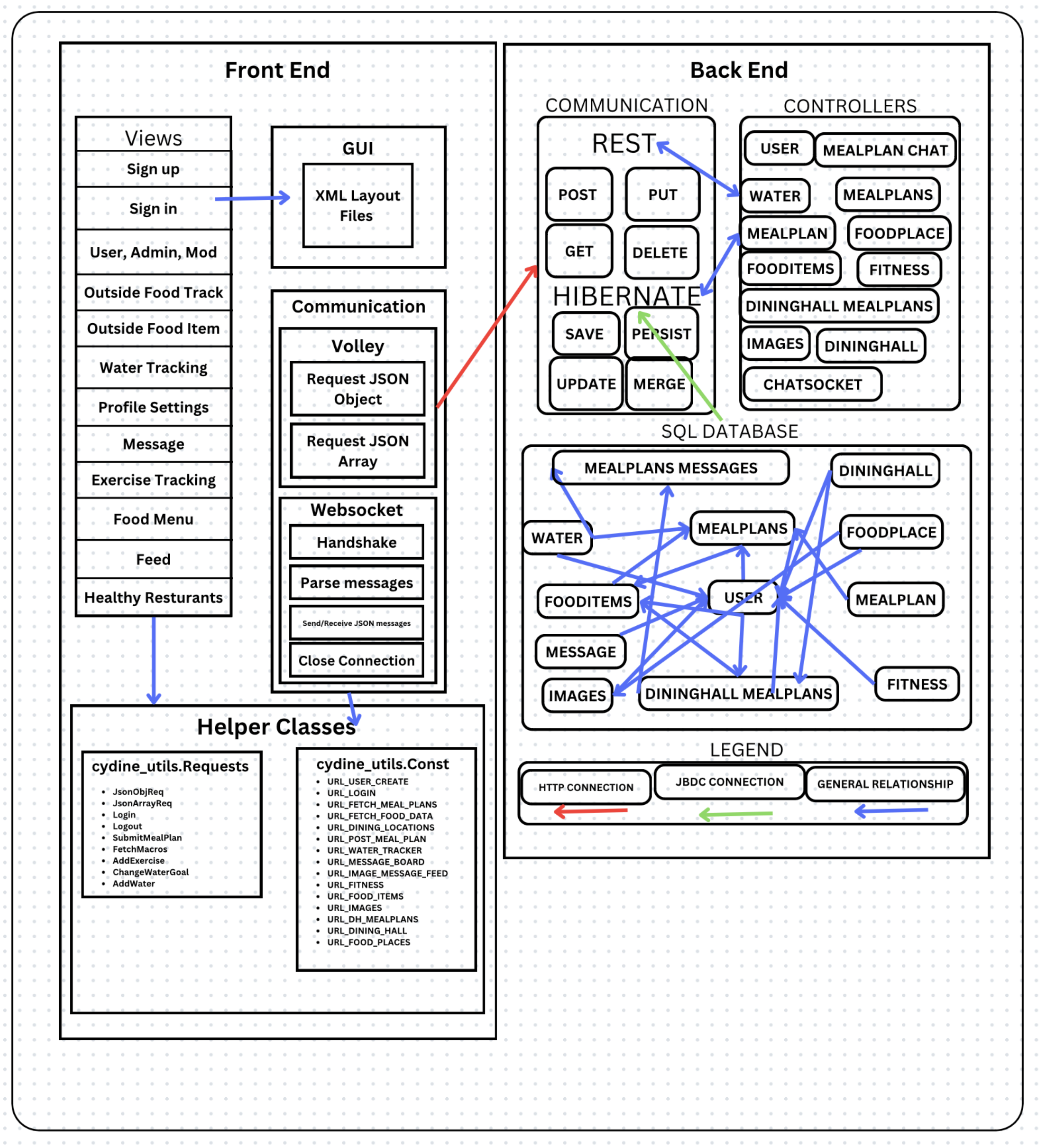
Asray Gopa: 25% contribution

Issmle Bekri: 25% contribution

Arjun Patel: 25% contribution

Akhil Pallem: 25% contribution

PUT THE BLOCK DIAGRAM PICTURE ON THIS PAGE! (Create the picture using pencil or drawIO)



Design Description

Frontend

Sign Up (User)

- User will be able to create an account and asked to prompt:
    - EditText: First Name, Last Name, Email, Password, Confirm Password
    - Button: Sign Up
- Upon clicking on Sign Up button there will be a POST request to server

Sign In (User, Admin, Mod)

- User, Admin, Mod will be able to sign in and asked to prompt:
    - EditText: Email, Password
    - Button: Sign In
- Upon clicking Sign In if the username and password is valid it will take them to the home screen.

AdminAccount (Admin)

- Generates a page with active users and list of accounts:
    - Button: User Management
    - TextView: Active users
- Upon clicking user management admin will see the list of users using a GET request.
    - SelectButton: used to delete selected users

ModAccount (Mod)

- Generates a page with live message view and a list of reported messages:
    - Button: Reported Messages
    - TextView: Active Reports
- Upon clicking the button will see the list of messages using a GET request.
    - SelectButton: used to delete selected reported messages
    - BanUserButton: used to delete selected users who have been reported

Backend

Communication

The backend uses mappings to update the database based on information sent to the given Mappings in the URL which include:

- **Post:** send information on an item to be added to the database based on the controller
- **Get:** request information, often with an identifier for the specific item requested from
- the database and then display the requested information to the user
- **Put:** send information to update a specific item in the database
- **Delete:** send an identifier to delete a specific item from the database

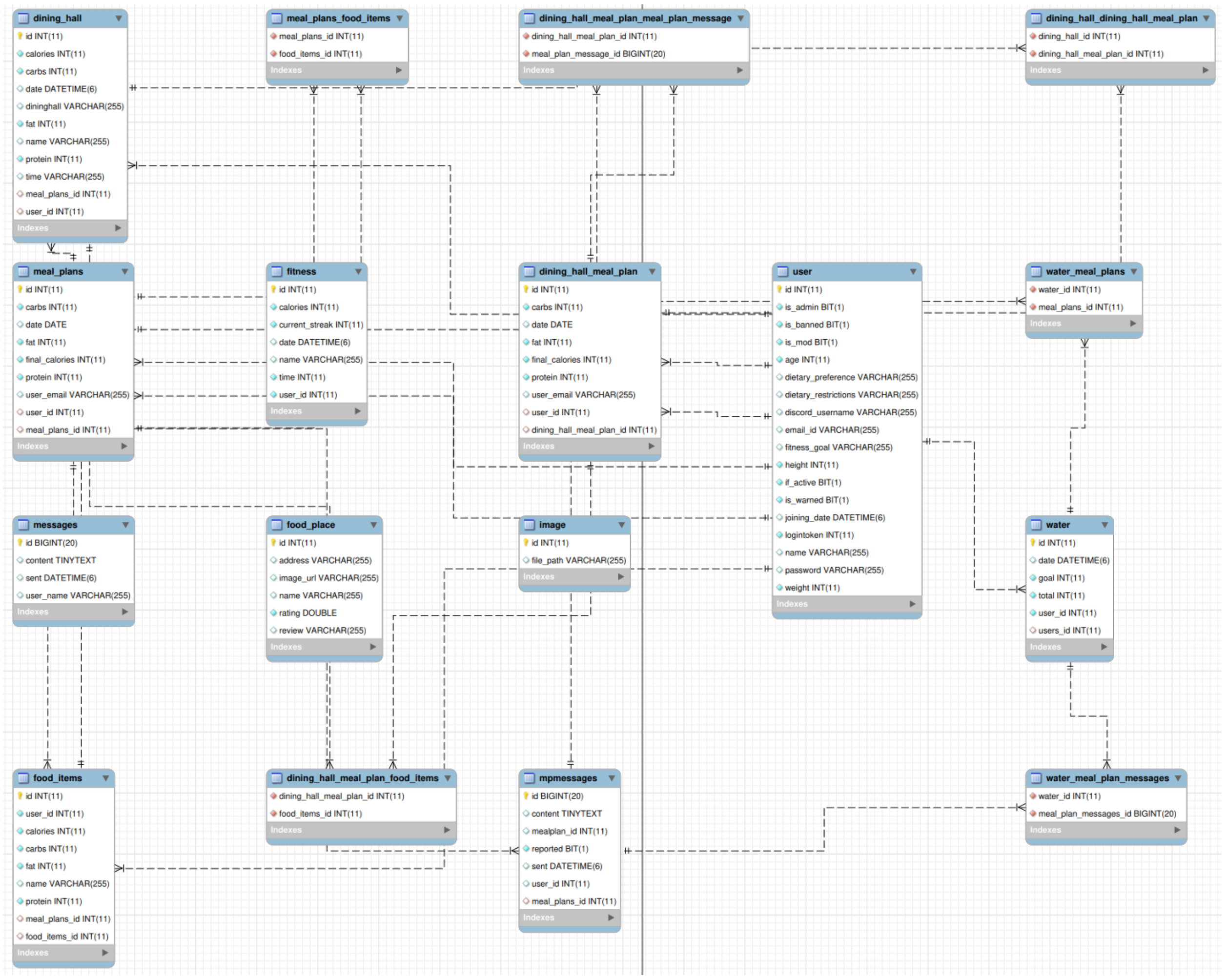
Controllers

The controllers contain the mappings for communication between frontend and the database which include:

- **User:** Contains the above mappings to create users, which contain one-to-many relationships for MealPlans, DiningHall MealPlan, and Food Items
- **Fitness:** Contains the above mappings to create fitness logs, which contain many-to-many relationships for Users
- **Food Place:** Contains the above mappings to create food places and reviews and ratings in ames with a map, which contain Many-to-many relationships for Users and Images
- **MealPlans:** Contains the above mappings to create meal plans, which contain Many-to-many relationships for Users

- **DiningHall:** Contains the above mappings to create foods from dining halls, which contain Many-to-one relationships for Users
- **Image:** Contains the above mappings to creates images to use in chat, which contain Many-to-one relationships for Users
- **FoodItems:** Contains the above mappings to create food items from outside sources, which contain Many-to-one relationships for MealPlans
- **MealPlan Chat:** Contains the above mappings to create MealPlans which can be displayed in chat, which contain Many-to-one relationships for Users
- **Chat Socket:** Contains the above mappings to create message display area and web socket, which contain Many-to-Many relationships for users
- **Water:** Contains the above mappings to create water logs for users, which contain Many-to-One relationships for Users

PUT THE TABLE RELATIONSHIPS DIAGRAM on this fourth page! (Create the picture using MySQLWorkbench)


dining_hall
id INT(11)
calories INT(11)
carbs INT(11)
date DATETIME(6)
dininghall VARCHAR(255)
fat INT(11)
name VARCHAR(255)
protein INT(11)
time VARCHAR(255)
meal_plans_id INT(11)
user_id INT(11)
Indexes
meal_plans_food_items
meal_plans_id INT(11)
food_items_id INT(11)
Indexes
dining_hall_meal_plan_meal_plan_message
dining_hall_meal_plan_id INT(11)
meal_plan_message_id BIGINT(20)
Indexes
dining_hall_dining_hall_meal_plan
dining_hall_id INT(11)
dining_hall_meal_plan_id INT(11)
Indexes
meal_plans
id INT(11)
carbs INT(11)
date DATE
fat INT(11)
final_calories INT(11)
protein INT(11)
user_email VARCHAR(255)
user_id INT(11)
meal_plans_id INT(11)
Indexes
fitness
id INT(11)
calories INT(11)
current_streak INT(11)
date DATETIME(6)
name VARCHAR(255)
time INT(11)
user_id INT(11)
Indexes
dining_hall_meal_plan
id INT(11)
carbs INT(11)
date DATE
fat INT(11)
final_calories INT(11)
protein INT(11)
user_email VARCHAR(255)
user_id INT(11)
dining_hall_meal_plan_id INT(11)
Indexes
user
id INT(11)
is_admin BIT(1)
is_banned BIT(1)
is_mod BIT(1)
age INT(11)
dietary_preference VARCHAR(255)
dietary_restrictions VARCHAR(255)
discord_username VARCHAR(255)
email_id VARCHAR(255)
fitness_goal VARCHAR(255)
height INT(11)
if_active BIT(1)
is_warned BIT(1)
joining_date DATETIME(6)
logintoken INT(11)
name VARCHAR(255)
password VARCHAR(255)
weight INT(11)
Indexes
water_meal_plans
water_id INT(11)
meal_plans_id INT(11)
Indexes
messages
id BIGINT(20)
content TINYTEXT
sent DATETIME(6)
user_name VARCHAR(255)
Indexes
food_place
id INT(11)
address VARCHAR(255)
image_url VARCHAR(255)
name VARCHAR(255)
rating DOUBLE
review VARCHAR(255)
Indexes
image
id INT(11)
file_path VARCHAR(255)
Indexes
water
id INT(11)
date DATETIME(6)
goal INT(11)
total INT(11)
user_id INT(11)
users_id INT(11)
Indexes
food_items
id INT(11)
user_id INT(11)
calories INT(11)
carbs INT(11)
fat INT(11)
name VARCHAR(255)
protein INT(11)
meal_plans_id INT(11)
food_items_id INT(11)
Indexes
dining_hall_meal_plan_food_items
dining_hall_meal_plan_id INT(11)
food_items_id INT(11)
Indexes
mpmessages
id BIGINT(20)
content TINYTEXT
mealplan_id INT(11)
reported BIT(1)
sent DATETIME(6)
user_id INT(11)
meal_plans_id INT(11)
Indexes
water_meal_plan_messages
water_id INT(11)
meal_plan_messages_id BIGINT(20)
Indexes