Block Diagram for R8MyApartment

Group MK-12(4_kabir_2)

Anish Nag: 25% contribution
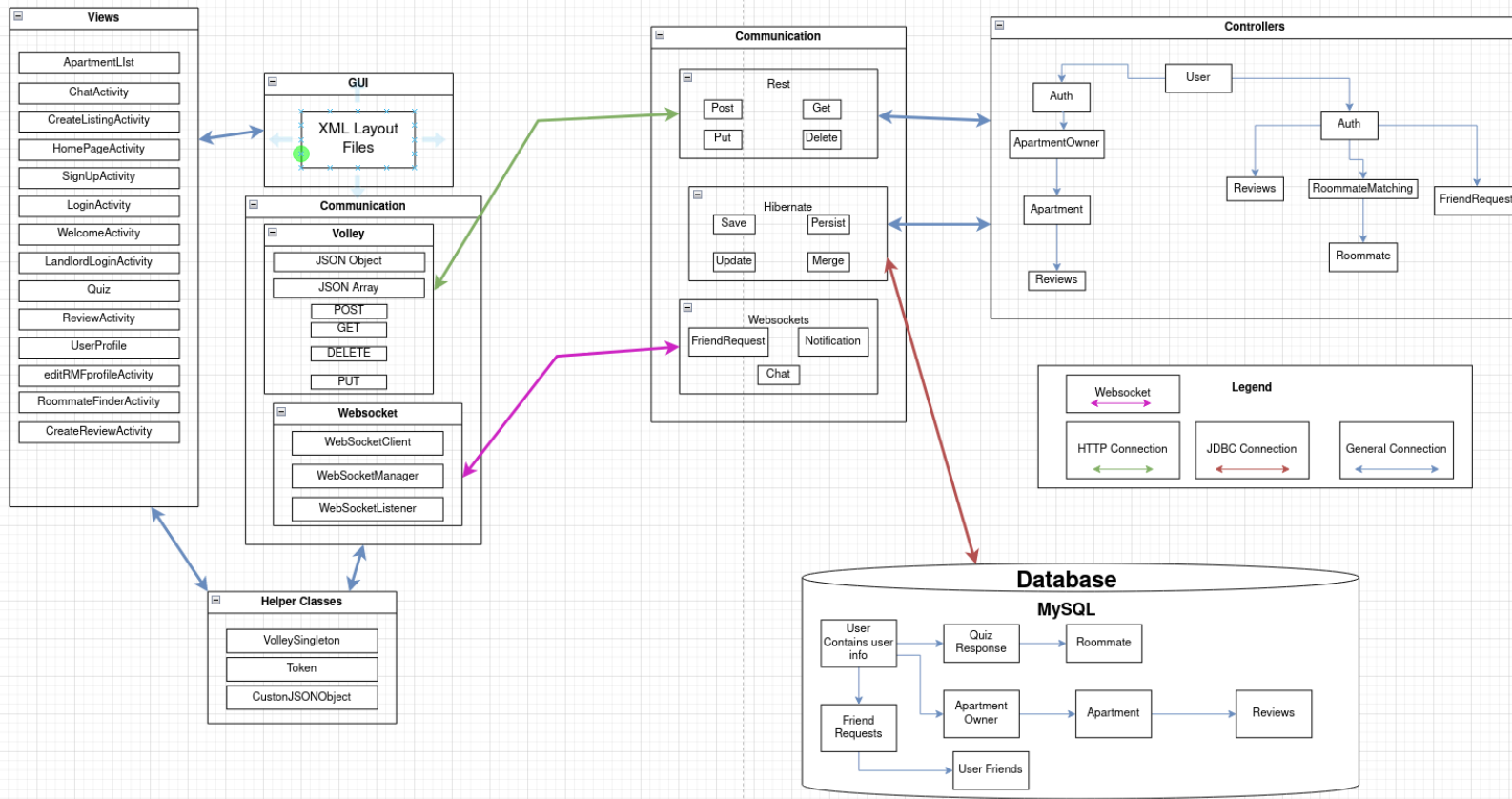
Declan Creadon: 25% contribution

Vinicius Soares: 25% contribution

Justin Rohde: 25% contribution

BLOCK DIAGRAM SCHEMATIC

Frontend

SignUpActivity(User, Landlord)

- For user signup, the user is prompted to enter information for the following fields:
    - EditText: FirstName
    - EditText: LastName
    - EditText: Email
    - EditText: birthDate
    - EditText: phoneNumber
    - EditText: password
    - EditText: confirmPassword
    - button: sign up
- At the end of these fields, there is a button labeled, sign up. Once this button is pressed, 'SignUpActivity' takes this information and forms a JSON object that is then sent to the server using a POST request.
- Landlord signup is a lot more simple, it only requires an email and password. Once these are entered then it sends a POST request to the server attempting to add the user to the database.

Login(User, Landlord)

- If an attempt as user is made, the following elements are shown:
    - EditText: username
    - EditText: password
    - button: login
- Once the login button is pressed 'Login' takes the information that was just entered and makes a POST request to the server. The server then takes this post request and returns a successful response if the user exists.
- If an attempt as a landlord is made, it is the same process as above except the landlord uses email as a parameter instead of the username. So, this one creates a POST request to the server with the email and username of the landlord attempting to sign in.

Quiz(User)

- If the current user is attempting to look for roommates, they are required to take a roommate finder quiz. When the user goes to take the quiz the following is presented with a scroll allowing users to choose 0-10 on their preferences.
  - SeekBar: hostingBar
  - SeekBar: petBar
  - SeekBar: smokingBar
  - SeekBar: organizedBar
  - SeekBar: guestsOverBar
  - SeekBar: noiseBar
  - SeekBar: cleanlinessBar
  - EditText: desiredUsername
  - button: submit
- Once the button is pressed the 'Quiz' takes the results and the entered username to create a JSON Object that is sent to the server as a POST request. The server then stores this in a quiz_responses table to be called upon.

Backend

Controllers:

The controllers allow for communication between our frontend and backend. Our controllers include the following:

- **User Management**
  - User Controller: Has endpoints for all CRUDL operations for users.
  - Auth Controller: Takes username and password from a post request and checks whether the username matches the entered password in our database.
  - Profile Controller: For our roommate profile feature, this has endpoints for all CRUDL operations.
  - Security Controller: Our security controller allows certain endpoints to be accessed only by our front end and no third party.
- **Property Management**

- Apartment Controller: Contains endpoints for creating apartments, editing, deleting, and listing apartments. Note that not just any user can create an apartment listing, they have to have an apartment owner account to create a listing.
- Listing Controller: Listing controller will list all apartments with their associated reviews. Also contains all CRUDL operations.
- Review Controller: Contains endpoints for CRUDL operations for reviews. A review cannot be created unless that apartment name is within the SQL database.
- Property Owner Controller: This contains all the CRUDL endpoints for the property owner account.
- **Roommate Features**
  - Roommate Controller: Contains endpoints for all CRUDL operations for roommates.
  - Matching Controller: Endpoints for
  - Quiz Controller: Has endpoints for all CRUDL operations with our roommate quiz.
  - Compatibility controller
- **Websocket Controllers**
  - Chat Server
  - Friend Request Server
  - Notification Server
  - Websocket Configuration

MYSQL TABLES

user_frien...
- user_id INT(11)
- friend_id INT(11)
- Indexes

user
- id INT(11)
- birth_date VARCHAR(255)
- email VARCHAR(255)
- first_name VARCHAR(255)
- last_name VARCHAR(255)
- password VARCHAR(255)
- phone_number VARCHAR(25...
- username VARCHAR(255)
- Indexes

apartment
- id BIGINT(20)
- address VARCHAR(255)
- amenities VARCHAR(255)
- contact_email VARCHAR(255)
- contact_phone VARCHAR(255)
- name VARCHAR(255)
- price DOUBLE
- owner_id BIGINT(20)
- Indexes

password
- id INT(11)
- password VARCHAR(255)
- Indexes

quiz_responses
- id BIGINT(20)
- cleanliness INT(11)
- hosting INT(11)
- liking_pets INT(11)
- morning_person INT(11)
- noise_level INT(11)
- organization_skills INT(1...
- people_over INT(11)
- smoking INT(11)
- username VARCHAR(255)
- Indexes

roommates
- id BIGINT(20)
- age INT(11)
- budget_range VARCHAR(25...
- cleanliness INT(11)
- hosting INT(11)
- lifestyle VARCHAR(255)
- match_percentage INT(11)
- morning_person INT(11)
- name VARCHAR(255)
- noise INT(11)
- occupation VARCHAR(255)
- organized INT(11)
- people_over INT(11)
- pet INT(11)
- smoking INT(11)
- username VARCHAR(255)
- Indexes

friend_reque...
- id INT(11)
- status ENUM(...)
- receiver_id INT(11)
- sender_id INT(11)
- Indexes

reviews
- id BIGINT(20)
- comment VARCHAR(255)
- rating INT(11)
- review_text VARCHAR(25...
- username VARCHAR(255)
- apartment_id BIGINT(20)
- user_id INT(11)
- Indexes

room_roommat...
- room_id BIGINT(20)
- roommate_id BIGINT(20)
- Indexes

chat_messages
- id BIGINT(20)
- content VARCHAR(1000)
- is_edited BIT(1)
- edited_at DATETIME(6)
- read_at DATETIME(6)
- recipient_username VARCHAR(25...
- sender_username VARCHAR(255)
- sent_at DATETIME(6)
- status ENUM(...)
- receiver_username VARCHAR(255)
- timestamp DATETIME(6)
- Indexes

apartment_owne...
- id BIGINT(20)
- email VARCHAR(255)
- name VARCHAR(255)
- password VARCHAR(255)
- Indexes

chat_message
- id BIGINT(20)
- content VARCHAR(255)
- receiver_username VARCHAR(25...
- sender_username VARCHAR(255)
- status ENUM(...)
- timestamp DATETIME(6)
- Indexes

amenities
- id BIGINT(20)
- name VARCHAR(255)
- Indexes

rooms
- id BIGINT(20)
- owner_id BIGINT(2...
- Indexes

roommate_frien...
- roommate_id BIGINT(20)
- friend_id BIGINT(20)
- Indexes