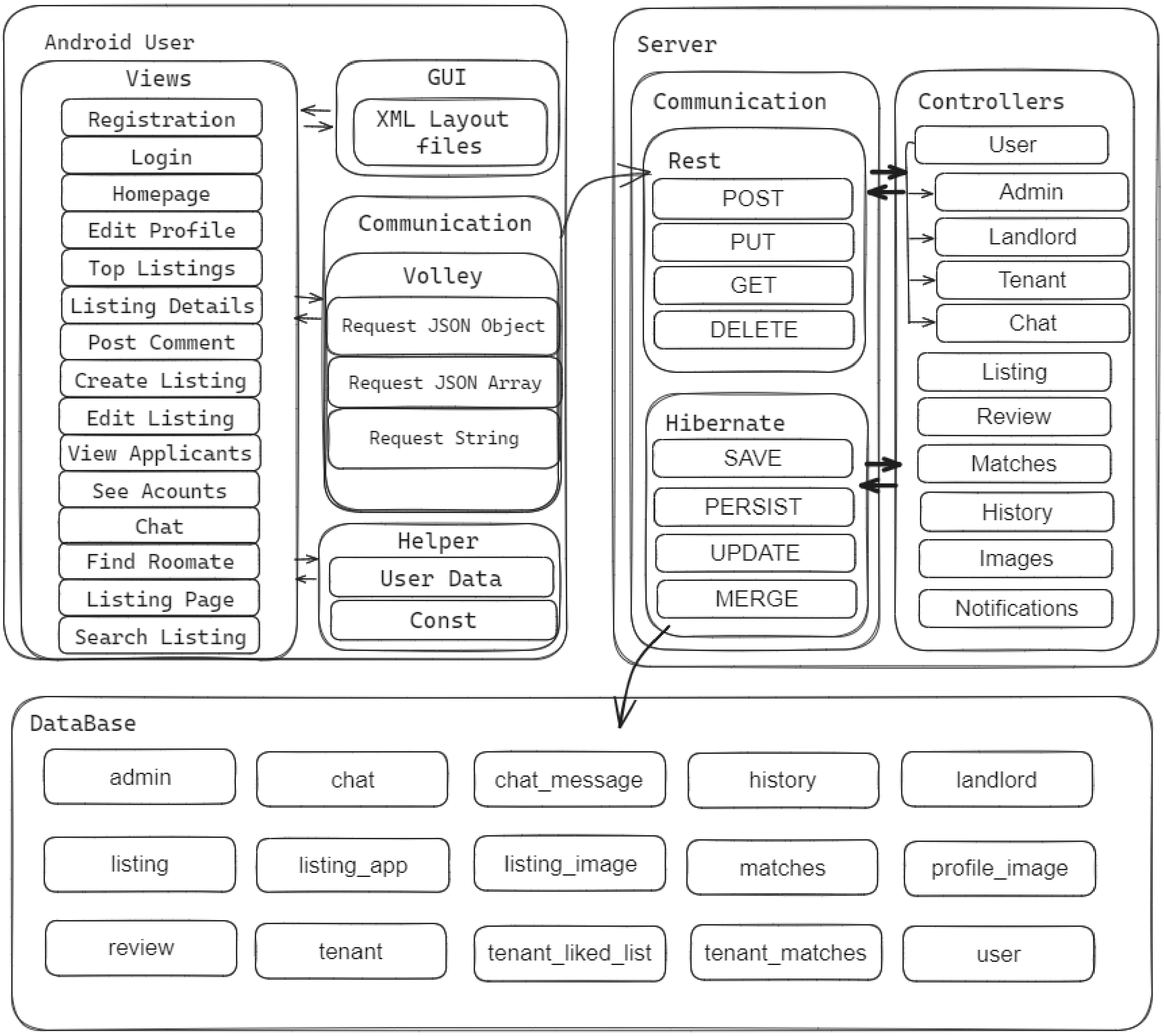
eDesign Document for <<Flat Finder >>

Group <JK-222>
Sullivan F: 25% contribution
Tyler V: 25% contribution
Carter P: 25% contribution
Brandon M: 25% contribution

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



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

Registration (User) - Creates a new account when Create button in clicked

Always takes – Name, Password Email

For Renter takes – Price Range, Location, Pets, # of rooms, # of bathrooms

Login button takes user to the login page

Login (User) - (Starting Page) Log in to your Account when Submit Button is clicked – Goes to homepage

Takes – Name and Password

For Renter – Finds Roommate Recommendations to display on homepage

Sign Up button takes user to Registration page

User -> User chat (Any User) – Implements a WebSocket to allow users to chat with one another.

View Notifications (Any User) - Implements a WebSocket so users can view notifications immediately. Things like application requests/responses or pending chats

Edit Profile (Any User) - Update the information for your page

Takes – Same info as registration, but only changes data fields that have an input

Leads to Login Page only

Homepage (Renter) – Buttons leading to other pages

List of roommate recommendations, each has a button navigating to "Other Tenant Details"

Other Tenant Details (Renter) – Shows details pertaining to one other renter, navigates to chat page

Recommended Listings (Renter) - List of recommended listings

Each listing containing buttons for like, comment, and navigation to "view listing details" page

View Listing Details (Renter) - Shows details pertaining to one listing including image

Navigation to apply to listing

Apply to Listing (Renter)- Sets an application flag connecting the user to that listing

Sends notification to Landlord to "review application"

Homepage (Landlord) - Buttons leading to other pages

Create Listing (Landlord) - Allows landlords to post listings

Review Applications (Landlord) - Allows landlords to approve or deny applicants to their listings

Backend:

Communication:

The backend of our application uses REST api and CRUD operations to allow the frontend to communicate with the database. These operations include:

POST: Send information and parameters to create an item to be stored in the database.

GET: Receive information about a certain item that is specified upon request.

PUT: Send information to update an already existing item in the database.

DELETE: Request an item to be removed from the database.

Controllers:

The controllers contain the URLs and mappings that allow the frontend to communicate with the database given the URL. Our controller include:

CHAT: Controls the WebSocket that allows for users to communicate with each other while using the app.

IMAGE: Allows the front end to retrieve images that are stored in the team server.

LISTING: Lets landlords create and manage their listing to advertise to tenants.

NOTIFICATIONS: Controls the WebSocket that allows for notifications to be sent to tenants and landlords, i.e., tenants are notified when rent is due.

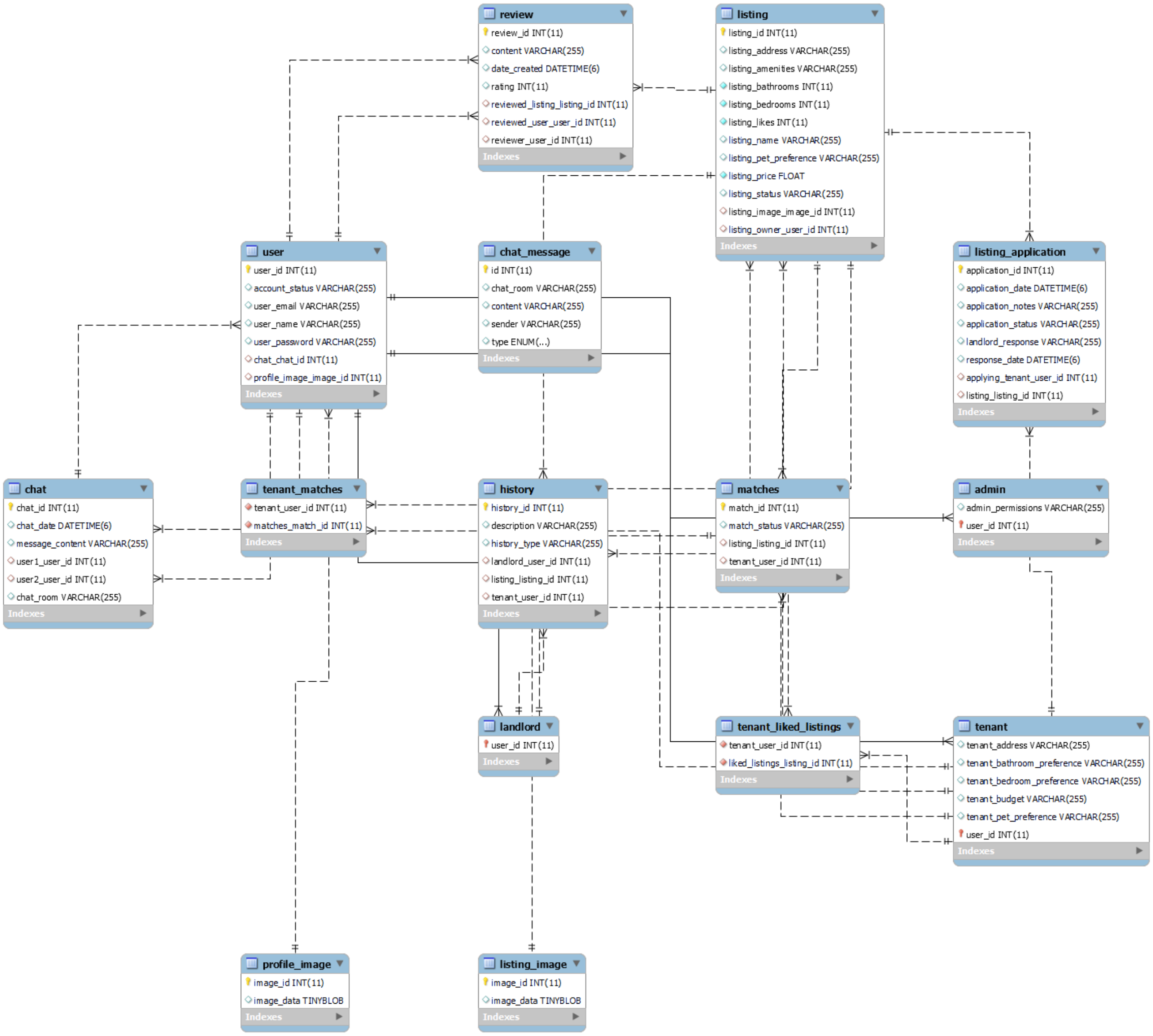
REVIEWS: Handles reviews that are published by tenants when reviewing a landlord or listing,

USER: Provides basic functionalities for all users that may be present on the app including landlords and tenants.

TENANT: Allows for additional features specific to tenants such as listing preferences.

LANDLORDS: Allows for additional features specific to landlords such their listings and reviews.

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


review
review_id INT(11)
content VARCHAR(255)
date_created DATETIME(6)
rating INT(11)
reviewed_listing_listing_id INT(11)
reviewed_user_user_id INT(11)
reviewer_user_id INT(11)
Indexes
listing
listing_id INT(11)
listing_address VARCHAR(255)
listing_amenities VARCHAR(255)
listing_bathrooms INT(11)
listing_bedrooms INT(11)
listing_likes INT(11)
listing_name VARCHAR(255)
listing_pet_preference VARCHAR(255)
listing_price FLOAT
listing_status VARCHAR(255)
listing_image_image_id INT(11)
listing_owner_user_id INT(11)
Indexes
user
user_id INT(11)
account_status VARCHAR(255)
user_email VARCHAR(255)
user_name VARCHAR(255)
user_password VARCHAR(255)
chat_chat_id INT(11)
profile_image_image_id INT(11)
Indexes
chat_message
id INT(11)
chat_room VARCHAR(255)
content VARCHAR(255)
sender VARCHAR(255)
type ENUM(...)
Indexes
listing_application
application_id INT(11)
application_date DATETIME(6)
application_notes VARCHAR(255)
application_status VARCHAR(255)
landlord_response VARCHAR(255)
response_date DATETIME(6)
applying_tenant_user_id INT(11)
listing_listing_id INT(11)
Indexes
chat
chat_id INT(11)
chat_date DATETIME(6)
message_content VARCHAR(255)
user1_user_id INT(11)
user2_user_id INT(11)
chat_room VARCHAR(255)
Indexes
tenant_matches
tenant_user_id INT(11)
matches_match_id INT(11)
Indexes
history
history_id INT(11)
description VARCHAR(255)
history_type VARCHAR(255)
landlord_user_id INT(11)
listing_listing_id INT(11)
tenant_user_id INT(11)
Indexes
matches
match_id INT(11)
match_status VARCHAR(255)
listing_listing_id INT(11)
tenant_user_id INT(11)
Indexes
admin
admin_permissions VARCHAR(255)
user_id INT(11)
Indexes
landlord
user_id INT(11)
Indexes
tenant_liked_listings
tenant_user_id INT(11)
liked_listings_listing_id INT(11)
Indexes
tenant
tenant_address VARCHAR(255)
tenant_bathroom_preference VARCHAR(255)
tenant_bedroom_preference VARCHAR(255)
tenant_budget VARCHAR(255)
tenant_pet_preference VARCHAR(255)
user_id INT(11)
Indexes
profile_image
image_id INT(11)
image_data TINYBLOB
Indexes
listing_image
image_id INT(11)
image_data TINYBLOB
Indexes