Chat System for Customer Support

Assignment 3

Name: Groza Dominic

Group: 30443

Teaching Assistant: Mitrea Dan

# Conceptual architecture of the online platform

The entities contained within the platform are:

* User
* Device
* Measurement

For each user, we have two roles:

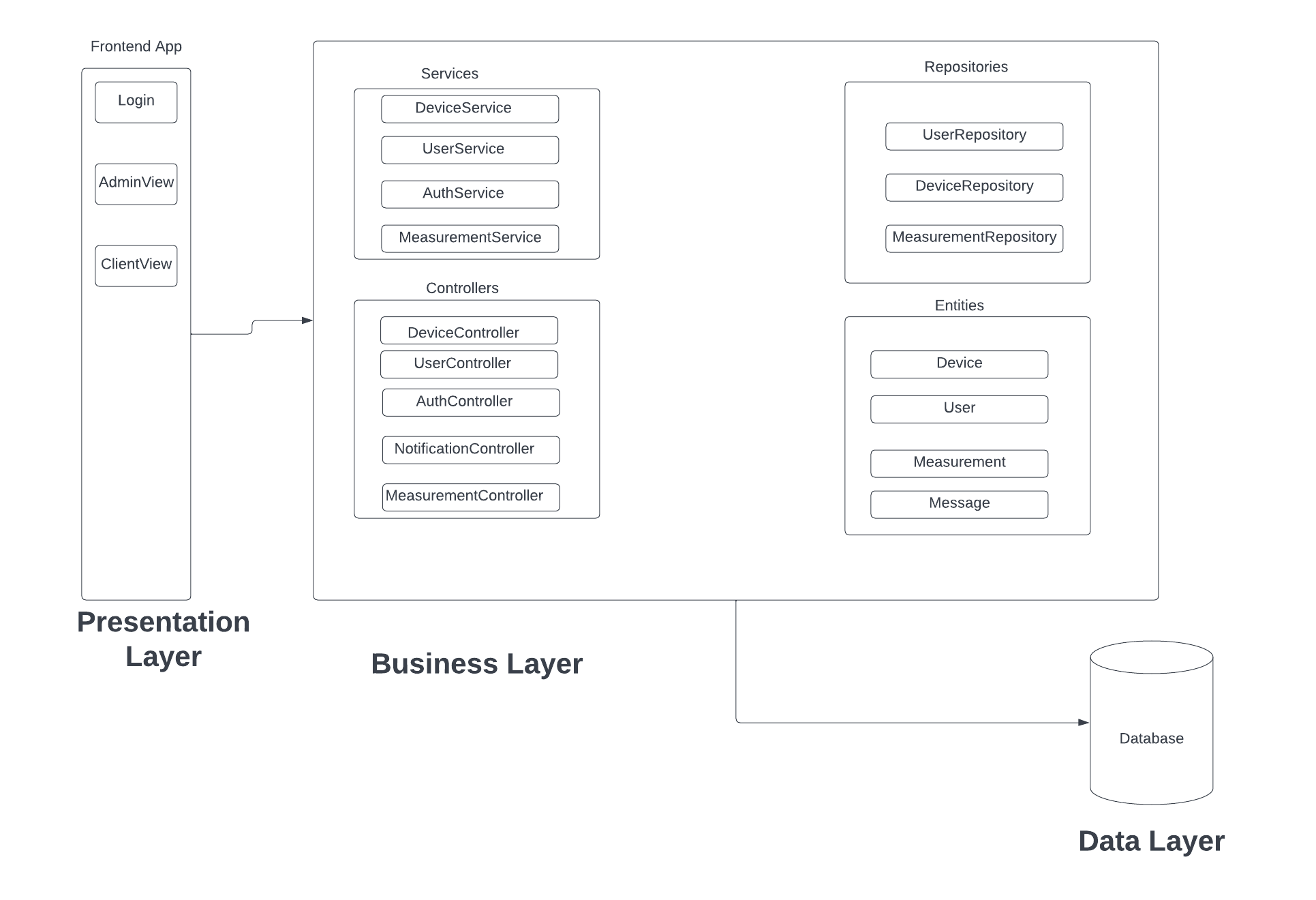
* Client
* Administrator

Each client can:

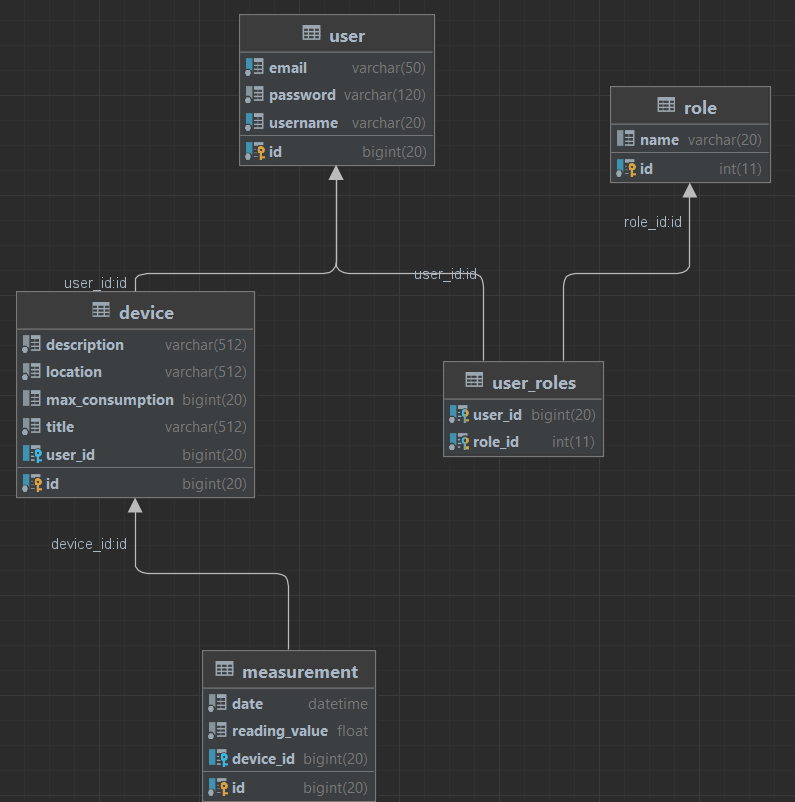
* View all his/her devices
* Generate charts containing the energy consumption [kWh] for each hour in a selected day
* Chat with any other user logged in the platform, provided that their username is known

Each administrator can:

* Create, edit, delete & view all users
* Create, edit, delete devices
* Assign a device to a user
* Chat with any other user logged in the platform, provided that their username is known



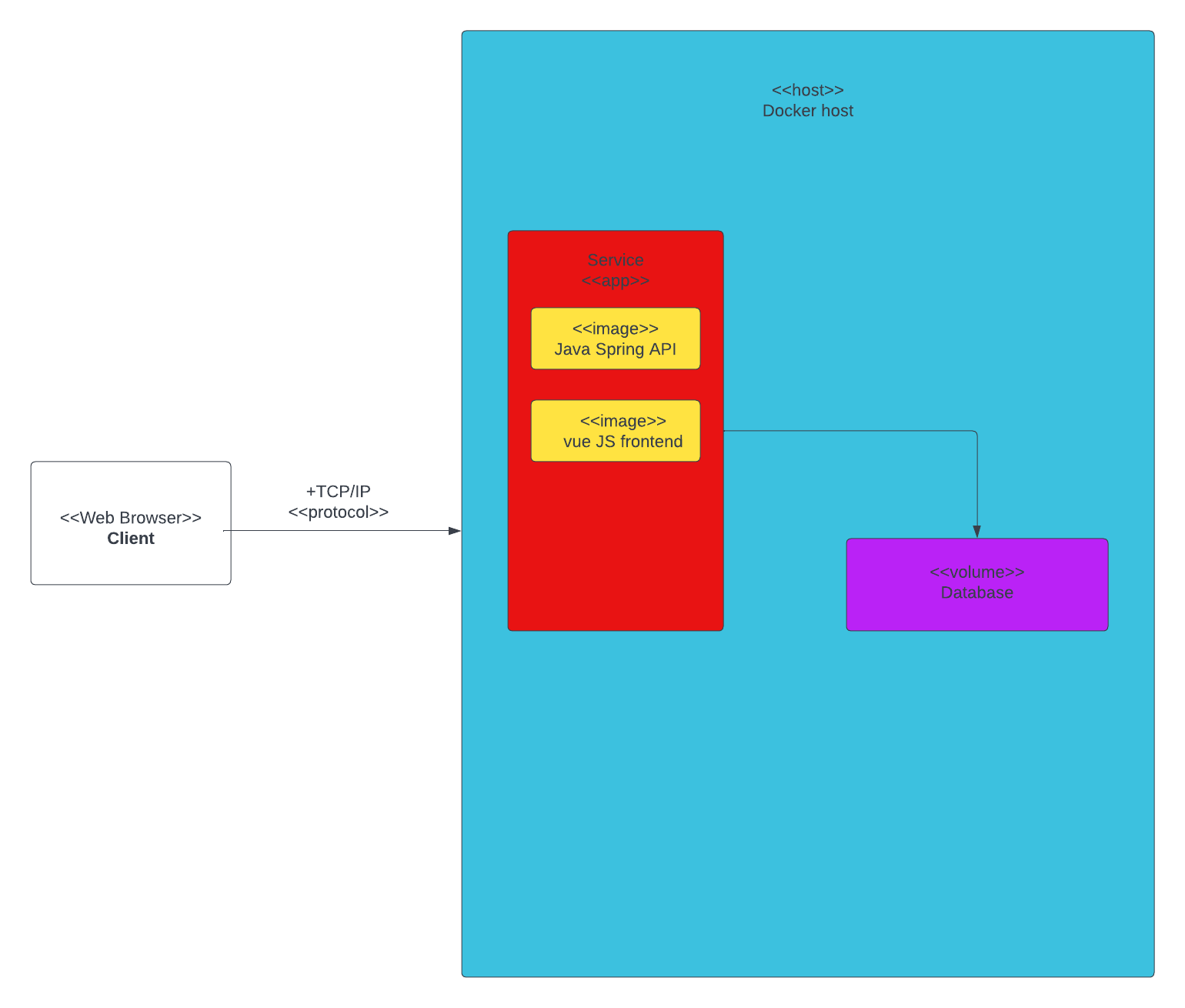
# Database design



Websockets

In this assignment, real-time chat functionality is implemented using WebSockets. Chat messages are sent in real-time to clients of the application, leveraging the push notifications that were implemented in the previous assignment. The protocol used for messaging is STOMP (Simple Oriented Messaging Protocol). To receive messages, the user must be logged in and have a valid session, which is used to identify them by their username. To send messages to a specific user, the sender must also provide their username and the recipient's username. This ensures that the recipient can see who sent the message, similar to how it is displayed in most other chat applications.

# UML Deployment Diagram



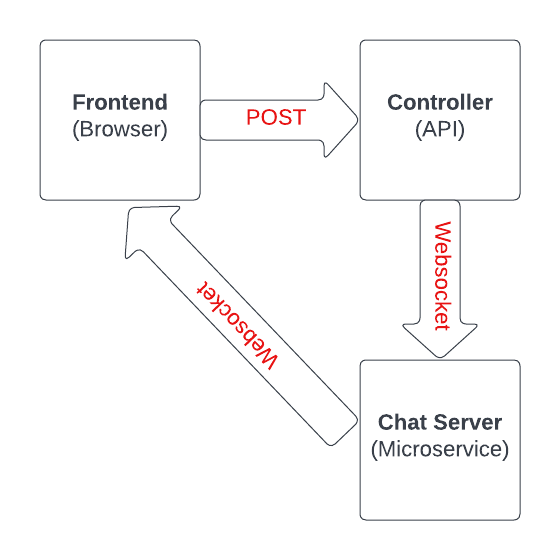
Port 80

Port 8090

Port 3306

Port 80

Port 8090



# Readme

* docker-compose build
* cd frontend
* docker-compose build
* cd ../
* docker-compose up