Department of Computer Science

Technical University of Cluj-Napoca

Distributed Systems

Integrated Energy Management System
Laboratory project

Name: Bogdan-Mihai Doia
Group: 30442

Teaching Assistant: Gabriel Antonesi

Contents

Chapter 1

Project Specification

The project involves developing a microservices-based web application with an Angular frontend and a backend composed of two Spring Boot microservices. The application enables user and device management functionalities, including CRUD operations and role-based access. The primary goal is to provide a scalable architecture, allowing each component to function independently while also working together as a complete system.

The application has the following features:

User Authentication: Users must authenticate to access the system and cannot access parts of the app they are not meant to without logging in and proper rights. This is done using a cookie that keeps the logged user's id and role.

Admin: Can view all users and devices in the dashboard. Can perform Create operations on both the databases. Can Edit or Delete users or devices. Can assign a specific device to a user. When a user is deleted the association between them and a device if it exists is also deleted by an API call from the user microservice to the device one.

Client: Can view in the dashboard the devices it has assigned by the admin.

Chapter 2

Implementation Details

The web application is developed with a microservices architecture. The backend is built using Spring Boot, where each microservice is dedicated to a specific domain(User Management and Device Management).

Each microservice has its own MySQL database, with user data stored separately from device data. The databases are accessed via Spring Data JPA.

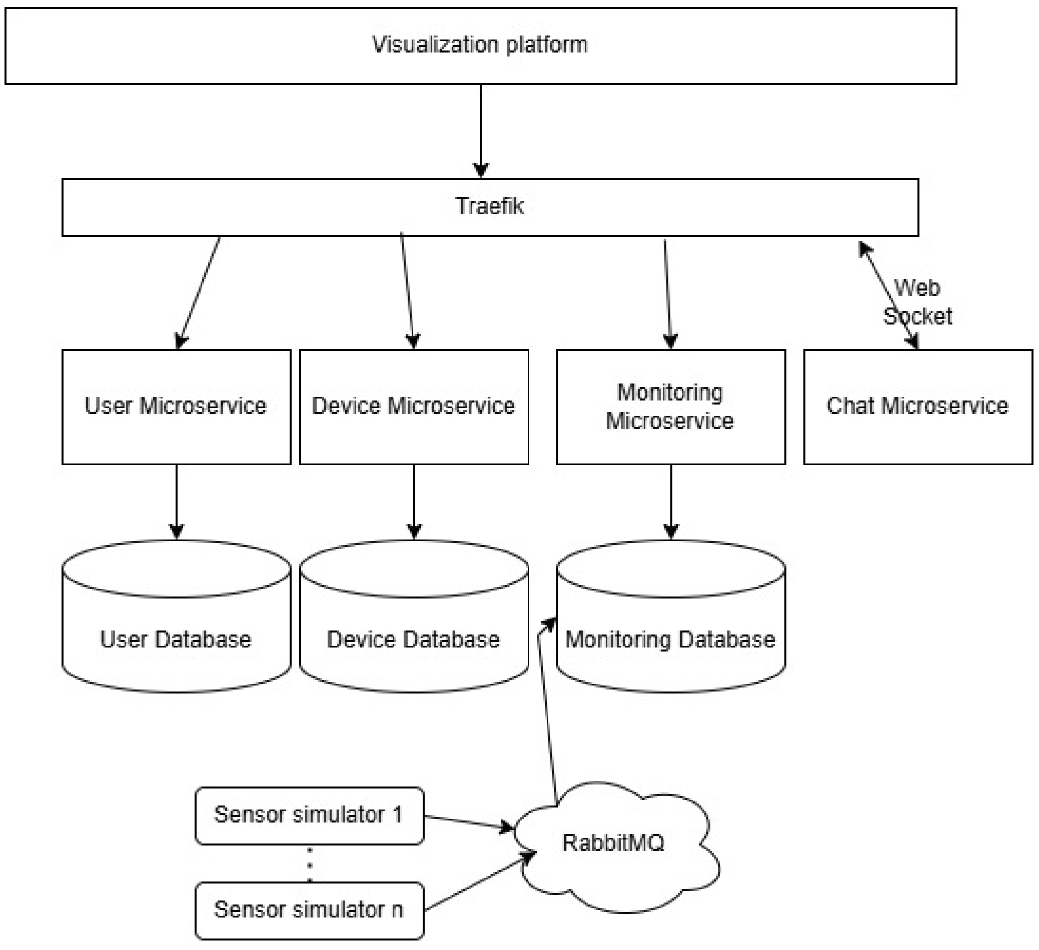
Docker is used to encapsulate each microservice, the frontend, and the databases. Docker Compose organizes these containers, ensuring they start and interact correctly.

The frontend is built in Angular. It communicates with the backend through HTTP requests and displays different views based on the user's role (admin or client). Authentication is handled with cookies storing essential information like user ID and role to manage access control throughout the application.

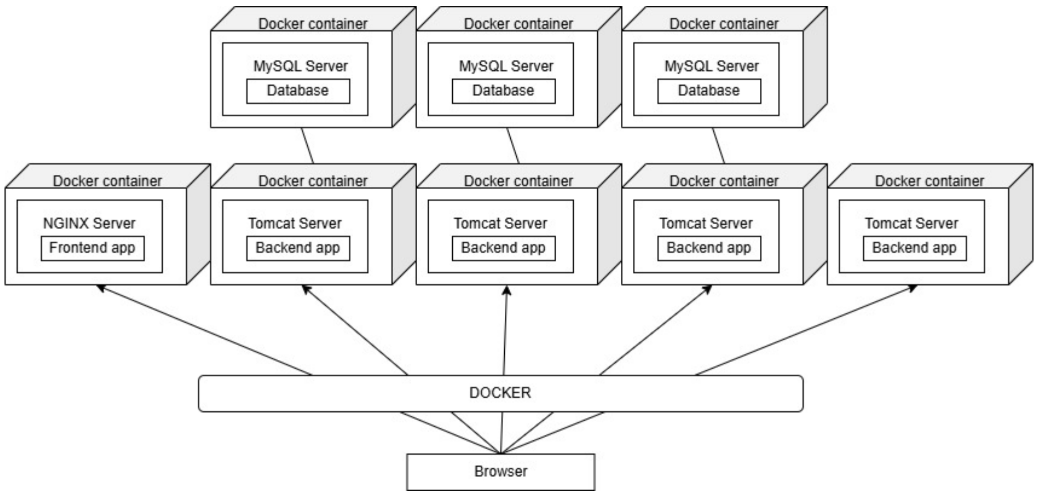
Chapter 3

Diagrams

3.1 Conceptual Architecture



3.2 Deployment Diagram



Chapter 4

Readme

4.1 Build and Execution Steps

To build and execute the application, follow these steps:

1. In each of the microservice projects, open a command prompt in the folder containing the `pom.xml` file and run the following command:

   ```
   mvnw clean package
   ```

   This will build the JAR files for each microservice.

2. Ensure a Dockerfile is present for both microservices as well as for the frontend. The `docker-compose.yml` file should be located in the top-level folder of the project.

3. Open a WSL terminal and run the following command:

   ```
   docker-compose up --build
   ```

   Wait for the build process to complete and confirm that both microservices are running. To populate the two databases, use the following command for each:

   ```
   docker exec -it project_mysql-users_1 mysql -u root -p
   ```

   Then enter the MySQL commands needed to populate the databases. Repeat this for both databases.

4. Open a web browser and navigate to `http://localhost:4200/` to access the application.