DOCUMENTATATION

ASSIGNMENT 1: ENERGY MANAGEMENT SERVICE

STUDENT NAME: PELLE ANDREI
GROUP: 30441

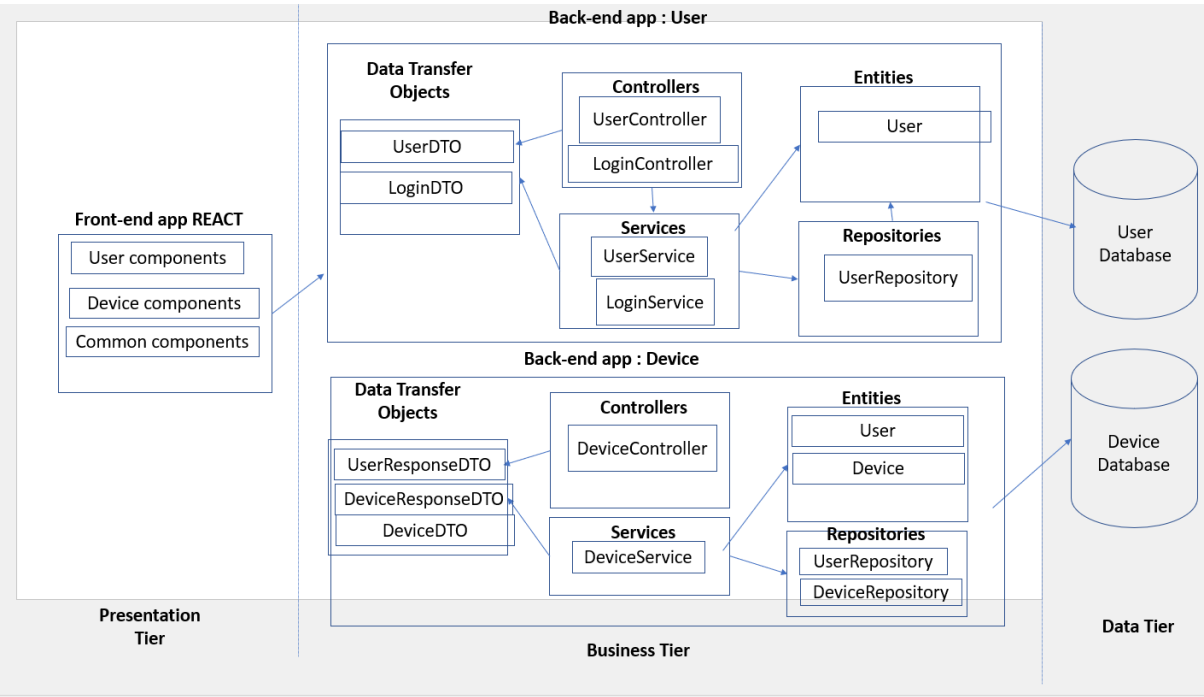
TABLE OF CONTENTS

1. Objectives

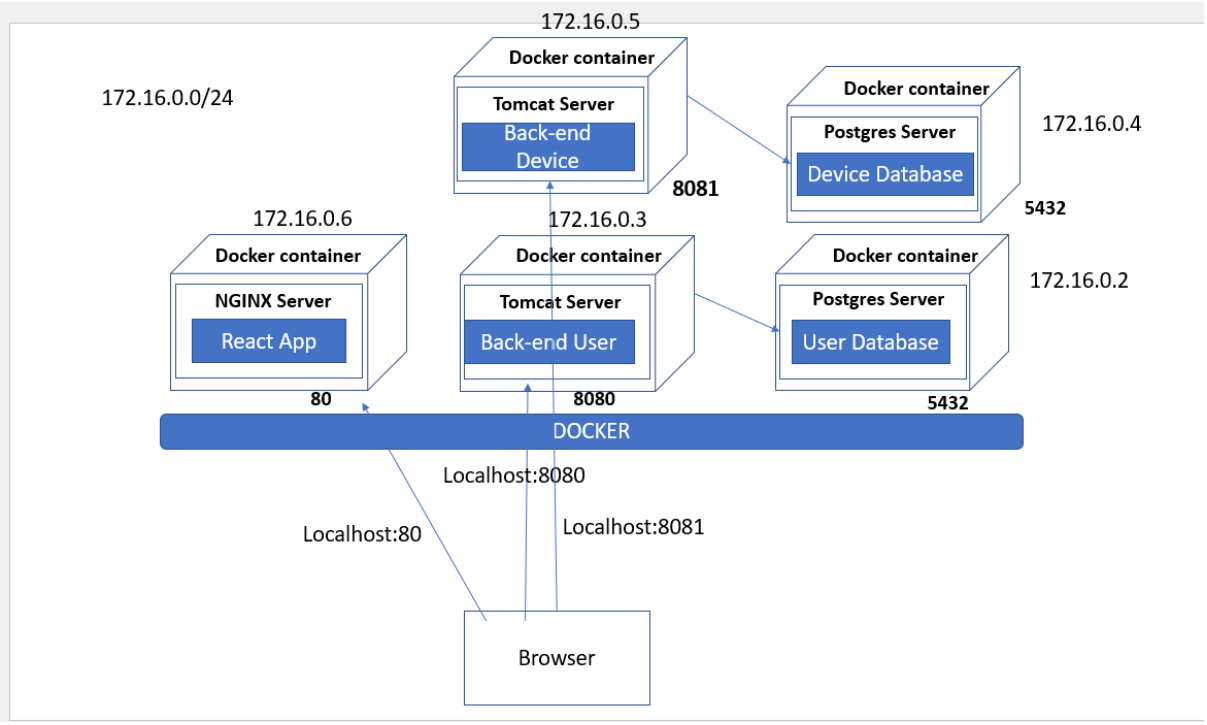
The task was to develop an Energy Management System that consists of a frontend and two microservices designed to manage users and their associated smart energy metering devices. The system can be accessed by two types of users after a login process: administrator (manager), and clients. The administrator can perform CRUD (Create-Read-Update-Delete) operations on user accounts (defined by ID, name, role: admin/client), smart energy metering devices (defined by ID, description, address, maximum hourly energy consumption), and on the mapping of users to devices (each user can own one or more smart devices in different locations).

2. Problem analysis, modelling, use cases

Conceptual architecture:



Deployment diagram:



3. Design and Implementation

In order to solve the problem of synchronizing the databases, I chose to send a POST/DELETE request from the user microservice to the device microservice any time such action is necessary. This ensures consistency between the databases, and the Transactional annotation allows for rollback in case something goes wrong. All backends are secure, device using just a simple filter that forwards to user, and user is protected by Spring Security alongside JWT token authentication.

The databases do not expose any ports outside of the docker network. This is for security purposes, as the browser does not need direct access to the databases.

4. Build and execution instructions

Build Instructions

Prerequisites

- Docker installed on your machine.

Building Images

In each folder starting with **assignment1**, there is a Dockerfile responsible for building images not found by default (e.g., PostgreSQL). Use the following commands to build these images:

```
cd assignment1-user-management
docker build -t my-user-image:latest .

cd ../assignment1-device-management
docker build -t my-device-image:latest .

cd ../assignment1-frontend
docker build -t my-react-image:latest .\
```

Execution Instructions

Starting the Microservices

Run the following command to start the microservices:

```
docker-compose up -d
```

This will start the PostgreSQL databases, Spring Boot applications, and the React frontend in detached mode.

Stopping the Microservices

To stop the services, run:

```
docker-compose down
```

Accessing Services

- User Management App: http://localhost:8080
- Device Management App: http://localhost:8081
- React Frontend: http://localhost

Stopping the Microservices

To stop the services, run:

```
docker-compose down
```

Folder Structure

```
assignment1-device-management/
assignment1-frontend/
assignment1-user-management/
.gitignore
docker-compose.yml
README.md
Doc_Pelle_Andrei Assignment_1.docx
Model_Arhitectura_Conceptuala.pptx
```

This structure organizes the microservices and relevant files.
For additional details, refer to the documentation files in the repository.