TECHNICAL UNIVERSITY
OF CLUJ-NAPOCA
ROMANIA

Faculty of Automation
and Computer Science

# Energy Utility Platform

-Distributed Systems project-

Fazakas Borbála
Group 30441
2022

# Table of Contents

# Overview

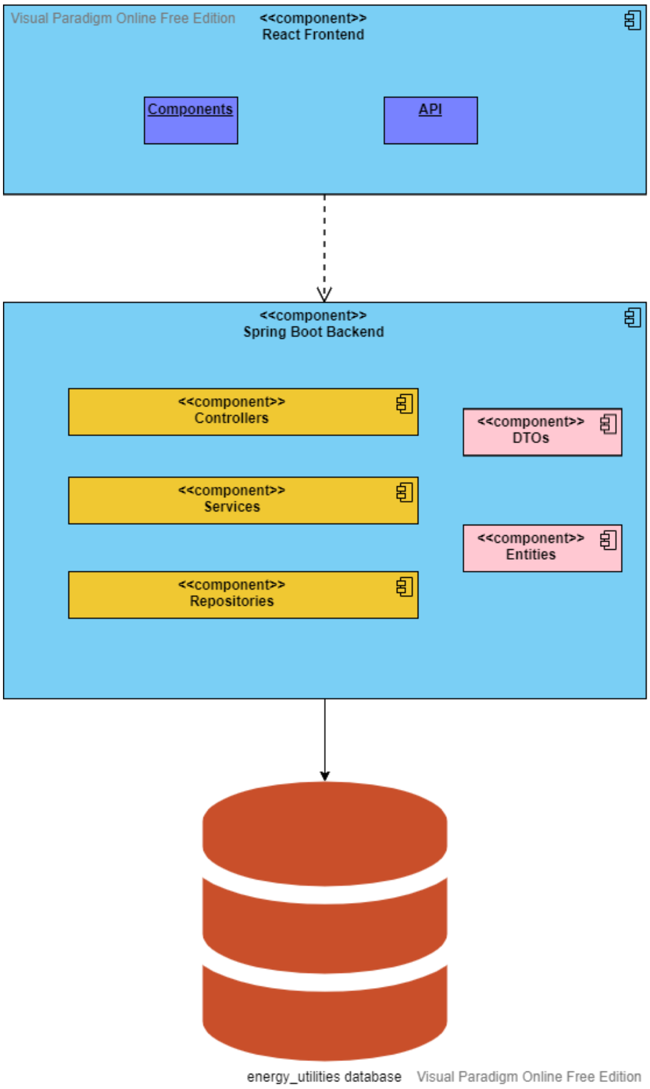
The energy utility platform is an application intended to facilitate monitoring the energy utilisation of the devices in any environment.

It supports two types of users: admin and clients, and the following functionalities:

- admins can
  - register new clients and admins
  - login
  - view the data of all clients and admins
  - update the data of existing clients or admins
  - delete any clients or admins (except themselves)
  - register new devices (includes specifying the client who owns the device)
  - view the data of all the existing devices
  - update the data of existing devices
  - delete any devices
- clients can
  - register themselves
  - login
  - view their associated devices
  - view the plot of the energy consumption of any of their associated devices, for any day

# Conceptual Architecture

## Overview



The application has a standard 3-Tier architecture, with:

- a Postgres database server, for persistent data storage
- a Spring Boot backend, for the business logic and the data access
- a ReactJs frontend, for the presentation

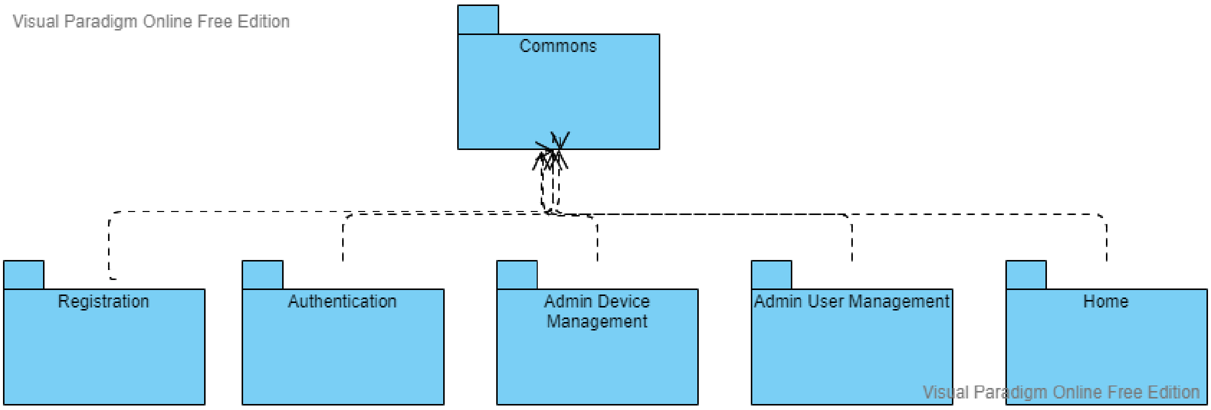
The backend offers a Rest API for accessing its services. The frontend uses this API to perform all the data retrievals and updates, through HTTP requests.

## Front End

As specified above, the front end is implemented with react and it is responsible for providing an easy-to-use user interface for the end users. Thus, it offers pages for

- home (the presentation of the app)

- registering
- login
- managing the devices, for the admins (CRUD operations)
- managing the users, for the admins (CRUD operations)

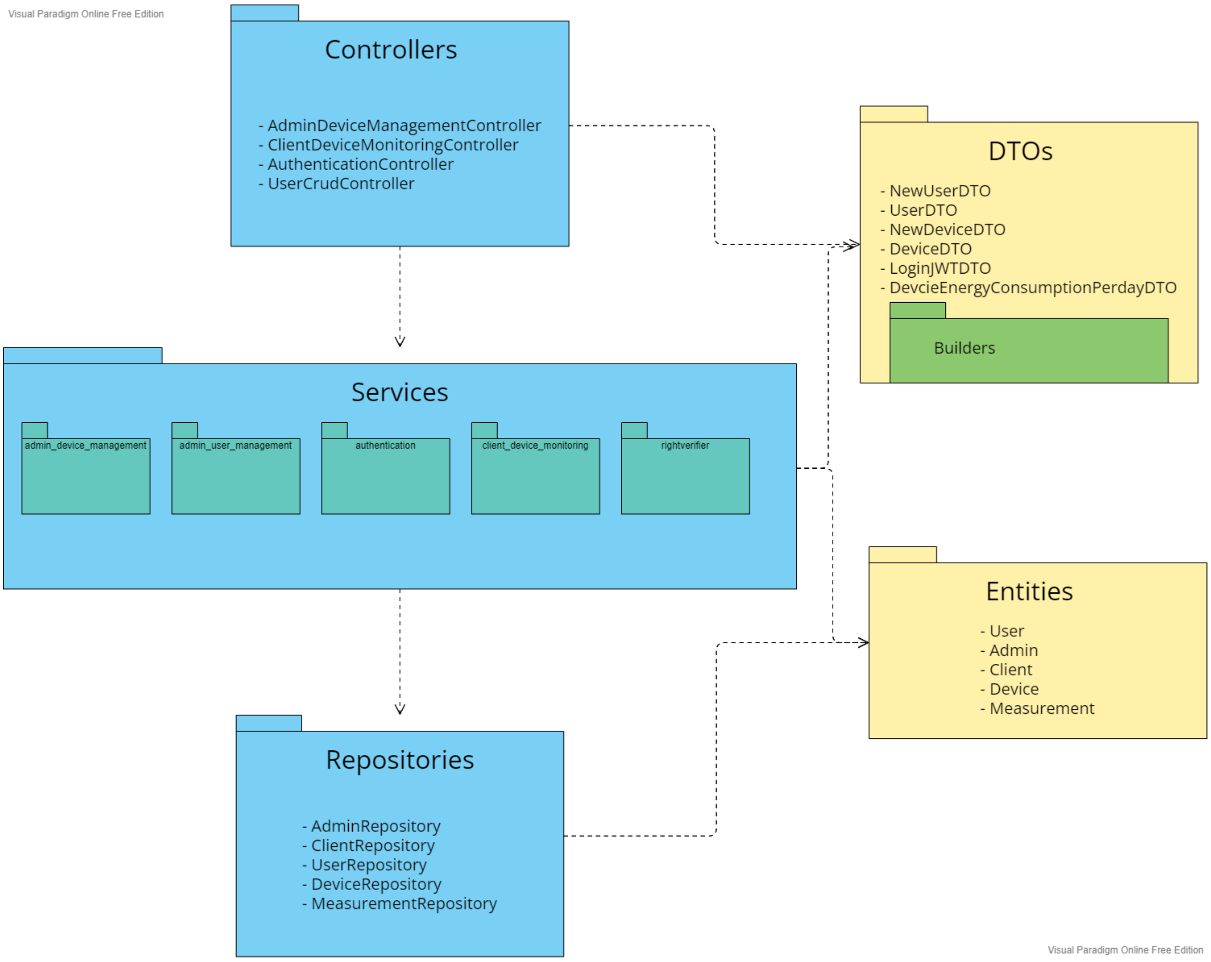


## Back End

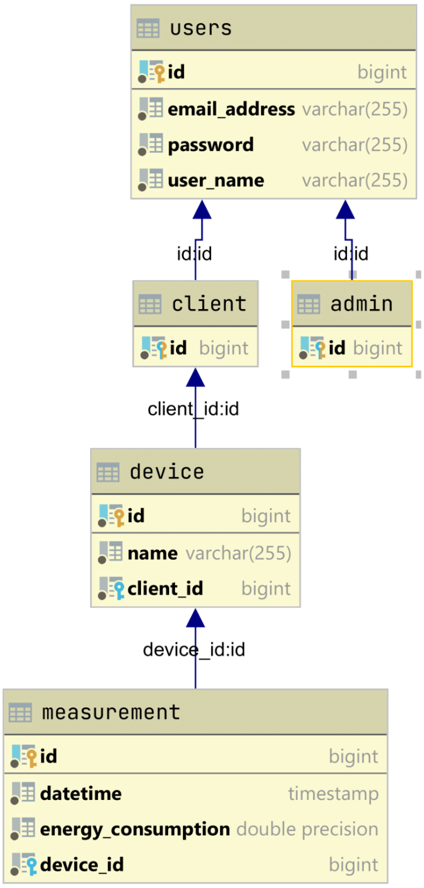
The back end is where all the business logic is implemented. It is structured in layers itself, with

- the controllers being responsible for taking up and handling HTTP requests from the clients
- the services implementing the business rules
- the repositories creating the connection with the database

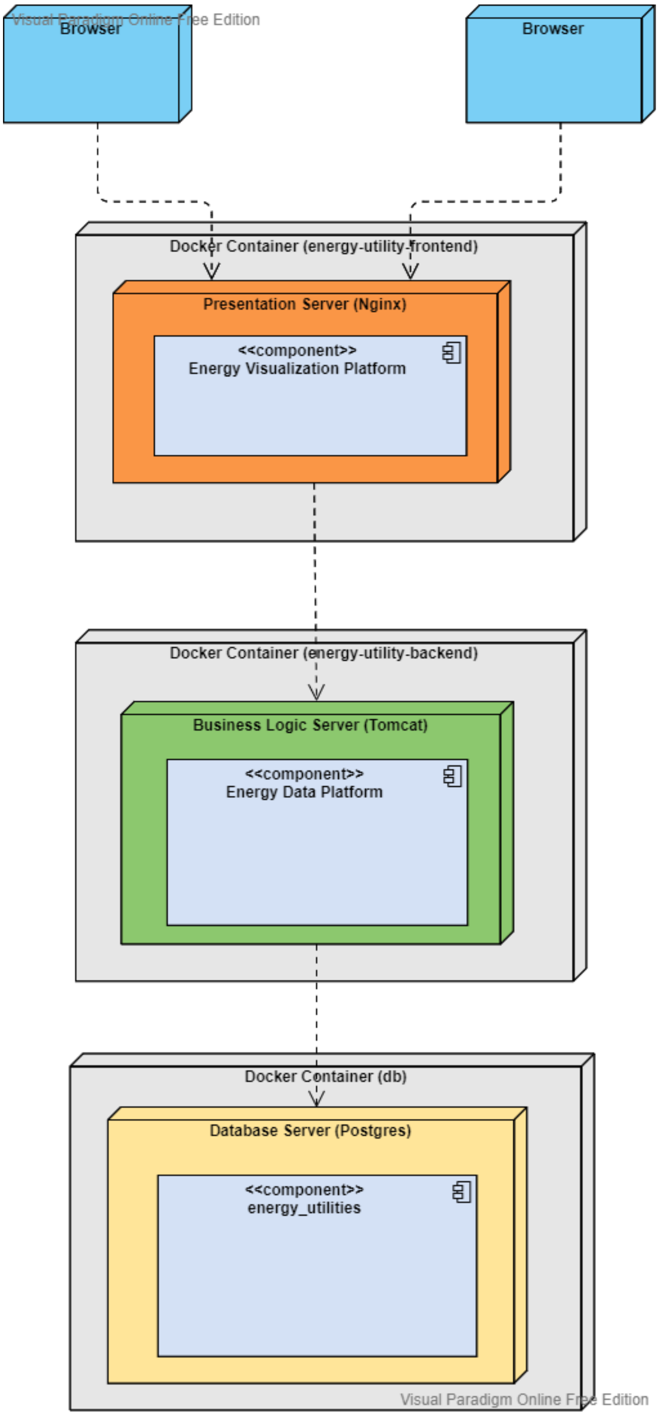
The controllers communicate via light-weight DTOs (data transfer objects) with the clients and the services, but the services convert the DTOs to entities, in order to mirror the data representation in the underlying database.


Controllers
- AdminDeviceManagementController
- ClientDeviceMonitoringController
- AuthenticationController
- UserCrudController
DTOs
- NewUserDTO
- UserDTO
- NewDeviceDTO
- DeviceDTO
- LoginJWTDTO
- DevcieEnergyConsumptionPerdayDTO
Builders
Services
admin_device_management
admin_user_management
authentication
client_device_monitoring
rightverifier
Entities
- User
- Admin
- Client
- Device
- Measurement
Repositories
- AdminRepository
- ClientRepository
- UserRepository
- DeviceRepository
- MeasurementRepository

# Database Design



# UML Deployment diagram


Visual Paradigm Online Free Edition
Browser
Browser
Docker Container (energy-utility-frontend)
Presentation Server (Nginx)
<<component>>
Energy Visualization Platform
Docker Container (energy-utility-backend)
Business Logic Server (Tomcat)
<<component>>
Energy Data Platform
Docker Container (db)
Database Server (Postgres)
<<component>>
energy_utilities
Visual Paradigm Online Free Edition

The database. the front end and the back-end need to be deployed in separate docker containers, as shown in the next section.

# Instructions for running the app

To start up the application with Docker, please follow these steps:

(Prerequisites: Docker and git installed)

1. Clone the repository

```
> git clone https://github.com/bori00/EnergyUtilityPlatform.git
```

2. enter the directory (corresponding to the back-end)

```
> cd EnergyUtilityPlatform
```

3. build the image for the backend and the database

```
> sudo docker build -t energy-utility-backend .
```

4. start the image for the backend

```
> sudo docker-compose up
```

5. enter the directory of the frontend

```
> cd react-client
```

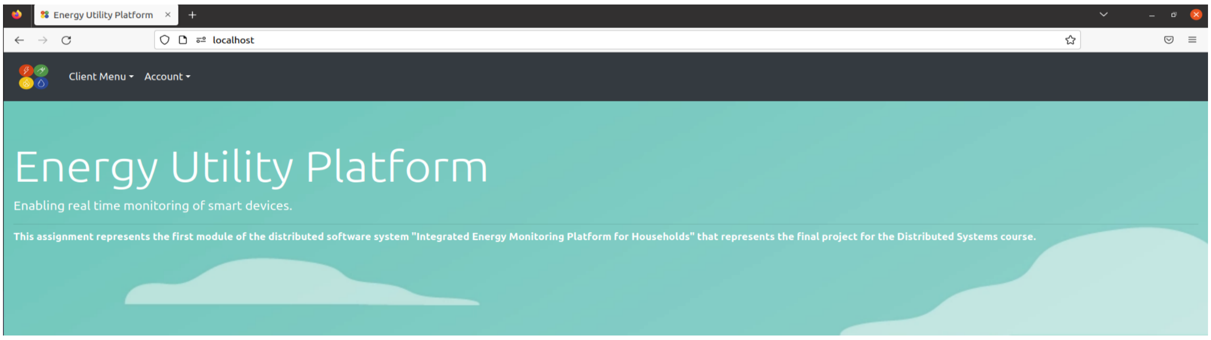
6. build the image for the frontend

```
> sudo docker build -t energy-utility-frontend .
```

7. start the image for the frontend

```
> sudo docker-compose up
```

8. Access the application at *lohalcost*



Energy Utility Platform

Enabling real time monitoring of smart devices.

This assignment represents the first module of the distributed software system "Integrated Energy Monitoring Platform for Households" that represents the final project for the Distributed Systems course.