**Text, logo

Description automatically generated**

**DISTRIBUTED SYSTEMS**

**ASSIGNMENT 2**

**Asynchronous Communication**

**Sensor Monitoring System and**

**Real-Time Notification**

**Neag Dragoș-Ion Group 30442**

Contents

[Architecture of the application 3](#_Toc124331219)

[UML Deployment diagram 4](#_Toc124331220)

[How to run the application 5](#_Toc124331221)

# Architecture of the application

An online platform should be designed and implemented to manage users, their associated smart

energy metering devices, and the monitored data from each device. 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,

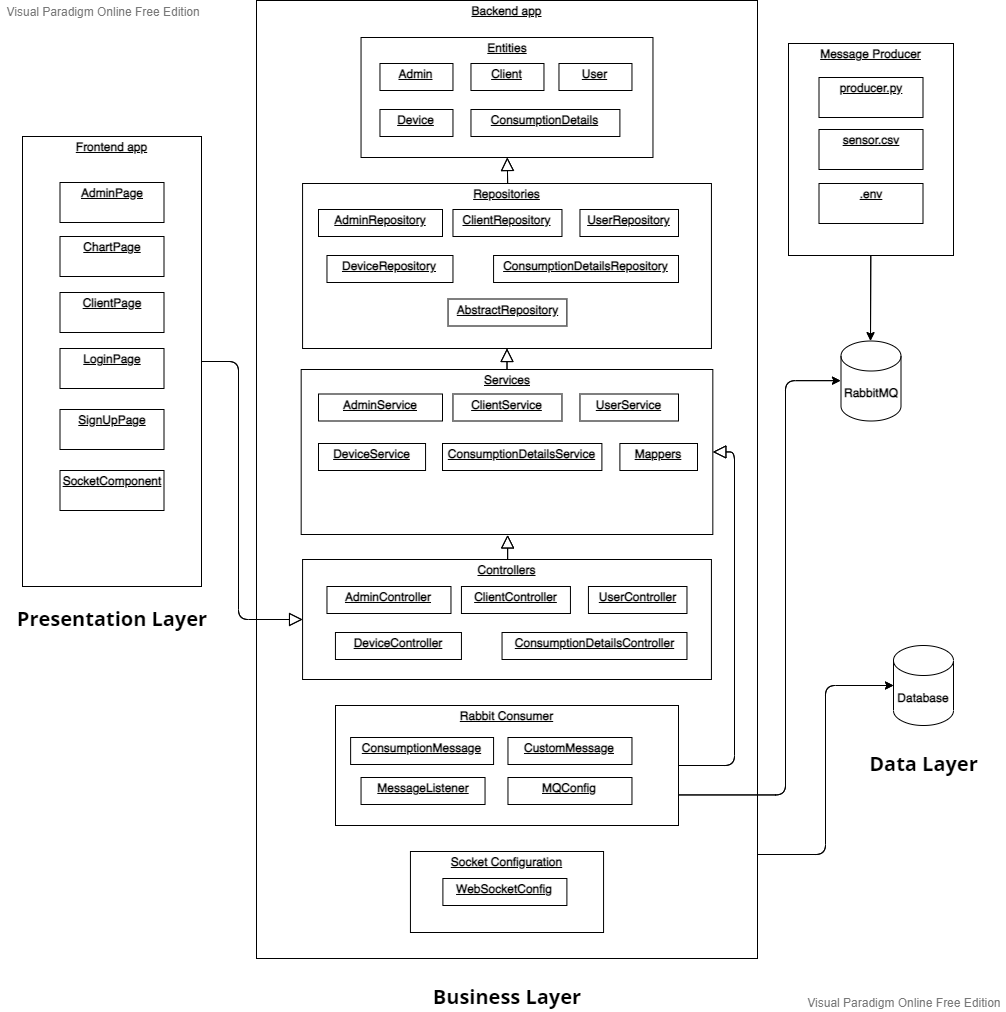
name, role: admin/client), registered smart energy metering devices (defined by ID, description,

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). After the mapping is done, for each

device the energy consumption is stored on hourly basis as tuples of the form <timestamp, energy

consumption> in the database.



The application will be an app implemented using Java. As a framework, we will use Spring Boot + Maven. As our goal is to design an app capable of administrating a database, Java provides a convenient way of building such an app. The IDE used for developing the project will be IntelliJ IDEA. Also, for the database we’ll be done in DataGrip using MySQL. This type of database covers our necessities as it is able to function smoothly with the backend of our app.

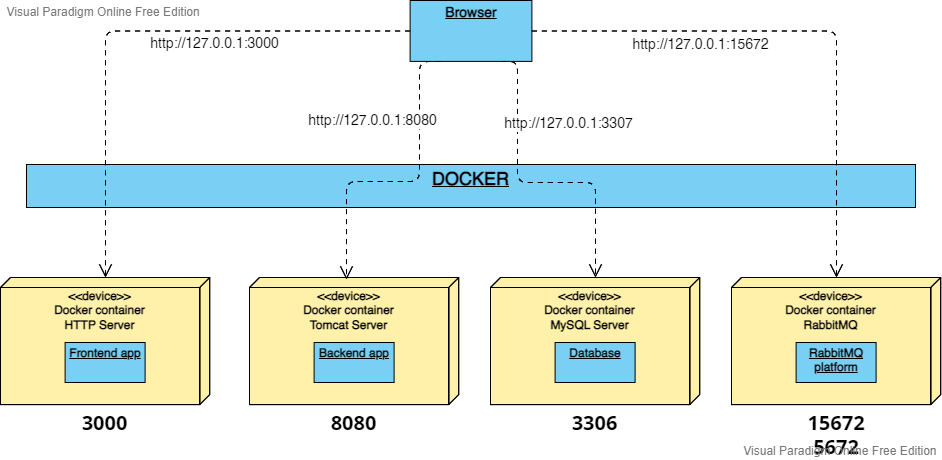
This time, a message producer was also added to supply messages read from an external source by sending them to the RabbitMQ platform, which will later provide the messages to the consumer present in the Spring application.

Java is a general-purpose, robust, secure, and object-oriented programming language. It is a high-level language, I.e., its syntax uses English like language. It was developed by Sun Microsystems in the year 1995. It is now maintained and distributed by Oracle. Java has its runtime environment and API; therefore, it is also called a platform.

Spring Framework is a Java platform that provides comprehensive infrastructure support for developing Java applications. Spring handles the infrastructure so you can focus on your application. Spring's focus on speed, simplicity, and productivity has made it the world's most popular Java framework.

RabbitMQ is lightweight and easy to deploy on premises and in the cloud. It supports multiple messaging protocols. RabbitMQ can be deployed in distributed and federated configurations to meet high-scale, high-availability requirements.

# UML Deployment diagram



# How to run the application

# How to install and run the application

First step is to install docker on the pc.

# Creating the containers

Start the docker platform.

Navigate to the backend folder in the command prompt and build the image like:

*docker build -t energy-platform-backend .*

Then navigate to the frontend folder in the command prompt and build the image like:

*docker build -t energy-platform-frontend .*

Then go back to the root folder of the project run the following command:

*docker compose up*

Wait until the mysql server will finish the initial configuration.

The backend will most likely fail due to the database linking can't be done as the mysql server is not ready in time for it.

Next, navigate to the docker desktop where we will see that the containers of the mysql server and react application are running.

We simply hit run on the backend container and it should start properly.

# Starting the application

Now navigate to (https://localhost:3000) and you should be able to interact with the app, replacing the localhost with your local IP.