

# DISTRIBUTED SYSTEMS

## Assignment 1

### Online Energy Utility Platform

Gramma Bianca-Alexandra

Group 30441

## Table of Contents

BACKEND .....	2
1. Introduction .....	2
2. Technology Stack .....	2
3. Diagrams.....	2
4. Architecture .....	4
FRONTEND .....	6
1. Technology .....	6
2. Architecture .....	6
BIBLIOGRAPHY .....	7

# BACKEND

## 1. Introduction

The online energy utility platform provides users with the possibility to view their devices and their energy consumption, view charts that can be filtered by date. There are two types of user: admin and client. The admin can view all the users and all the devices, add, edit or delete them. The client may only see his own devices and their charts.

## 2. Technology Stack

Programming language: Java

Backend framework: Spring

Frontend framework: AngularJS

Data storage and querying: PostgreSQL

## 3. Diagrams

The database diagram is presented in figure 1:

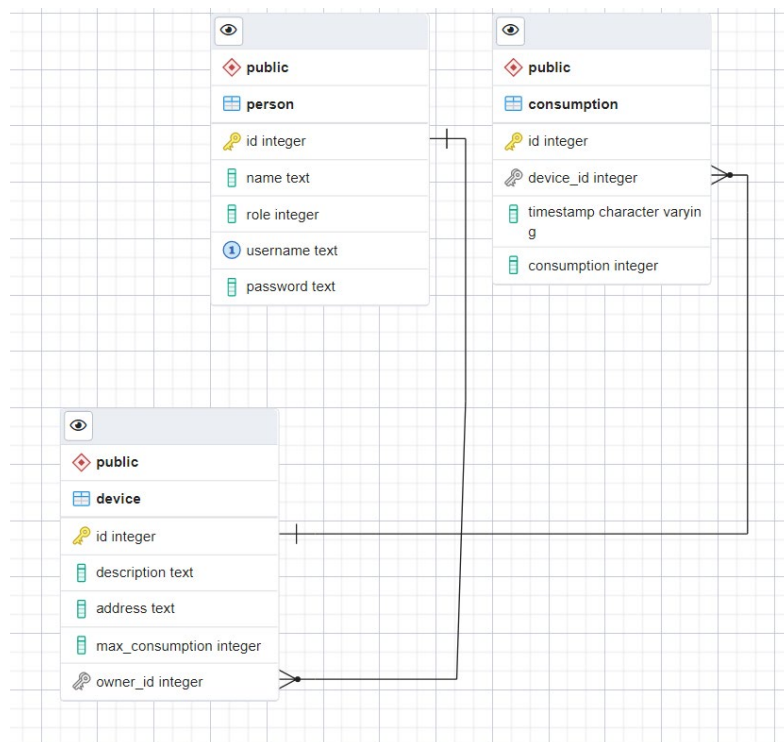


Figure 1

The package diagram is presented in figure 2:

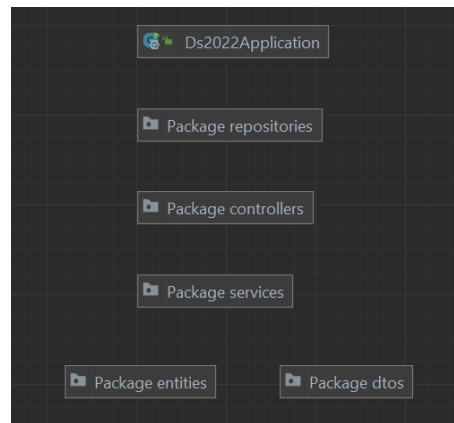


Figure 2

Figure 3 contains the conceptual architecture model:

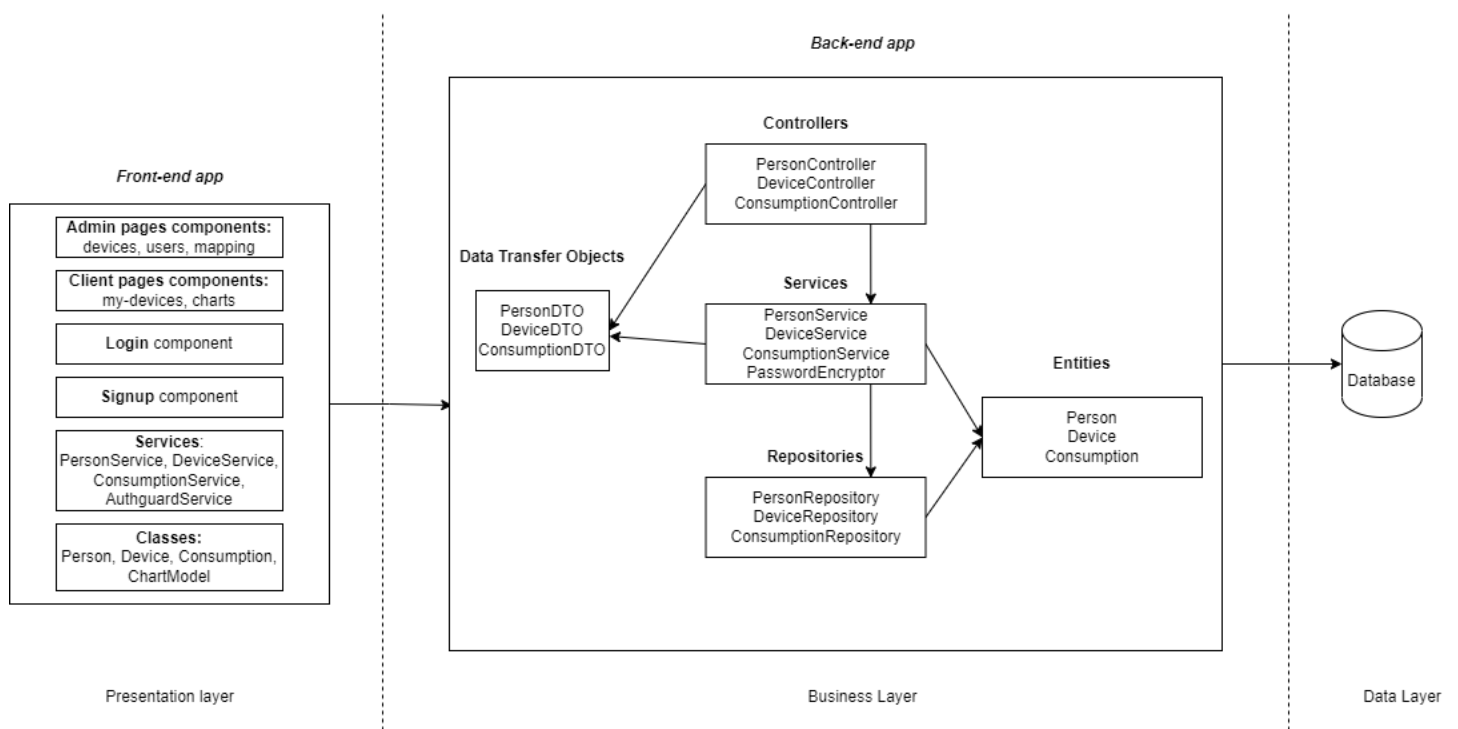


Figure 3

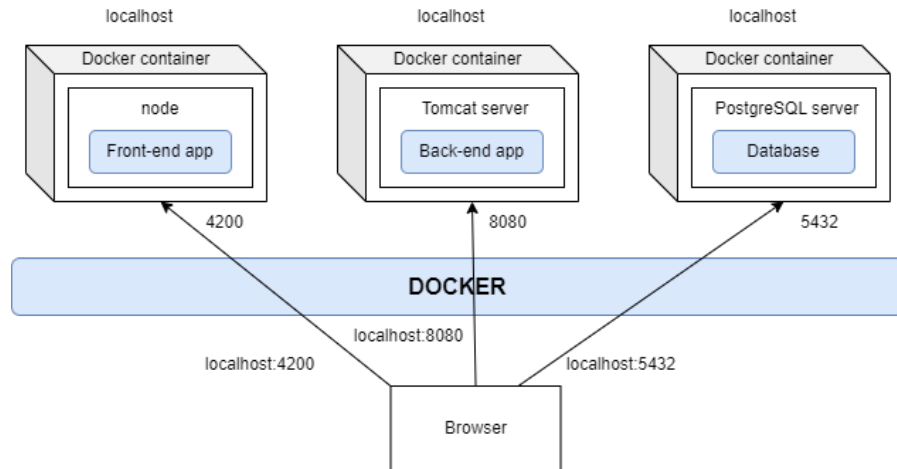


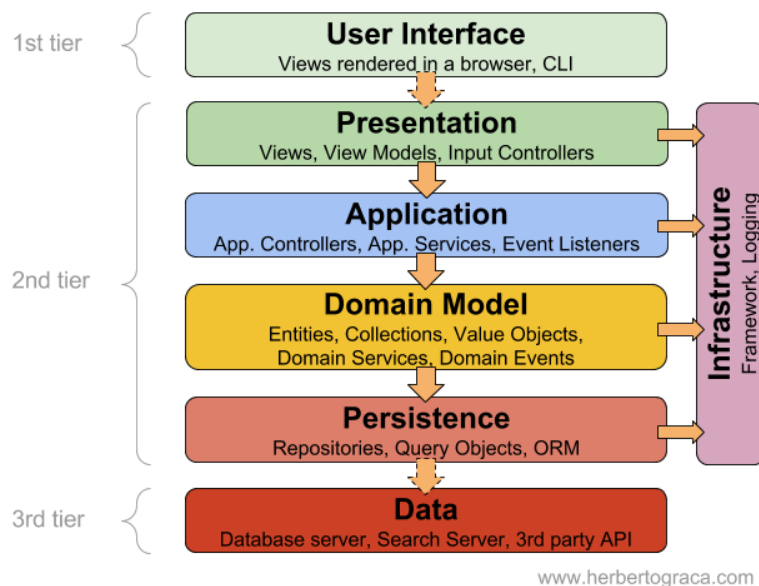
Figure 4

In figure 4, I have represented the deployment diagram. I have used a Docker container for each component.

#### 4. Architecture

The application will be implemented according to the layered architecture style. It consists of organizing modules or components with similar functionalities in horizontal layers. Each layer performs a specific role.

The layered architecture style promotes the concept of separation of concerns. It abstracts the view of the system, and each layer communicates with the layer below it.



www.herbertograca.com

The presentation layer consists of the frontend application.

In the business logic layer, I have controllers and services. The controllers are: ConsumptionController, DeviceController, PersonController. The services are: ConsumptionService, DeviceService and PersonService. I have also used a PasswordEncryptor that hashed the user password before adding it to the database.

In the model layer, I have the classes that illustrate the Person, the Device and the Consumption: one Entity and one DTO for each one.

In the data access layer, I have the repositories: ConsumptionRepository, DeviceRepository and PersonRepository.

# FRONTEND

## 1. Technology

I have implemented the front-end part of the project using Angular and the TypeScript programming language.

In order to obtain an aesthetically pleasing user interface, I have used primeng – a set of open source native Angular UI components.

## 2. Architecture

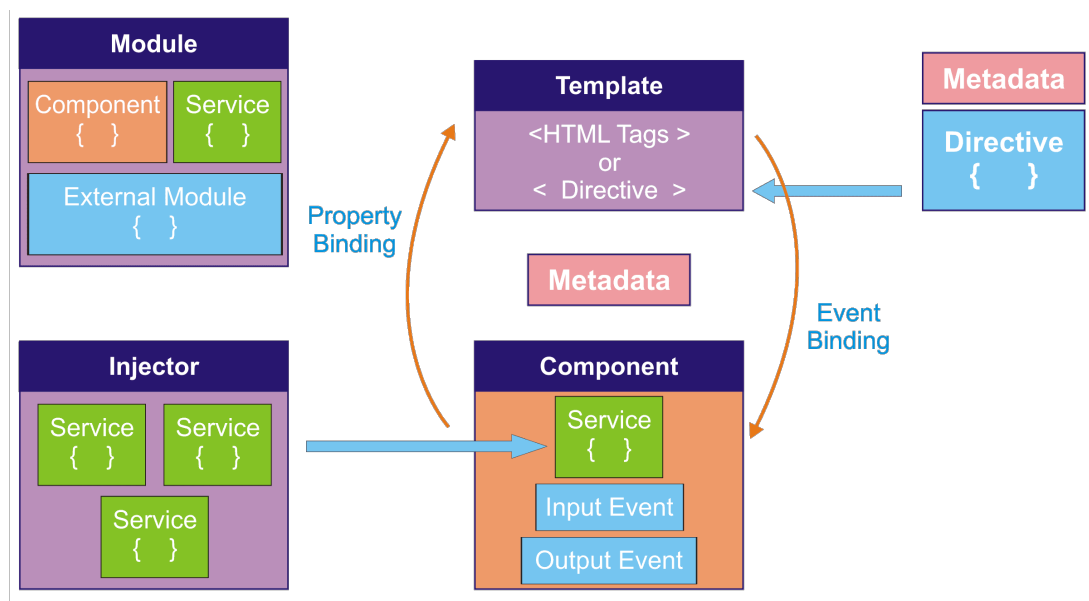
The main components of an Angular application are: components, classes and services.

A component has 4 types of files associated with it:

- A .html file -> the structure of the webpage portion represented by the component
- A .scss file -> the styling of the page
- A .ts file -> typescript code that defines the behaviour of the component, functions, communication with the services
- A .spec.ts file -> used for testing

A class contains data fields and has the same structure as the DTO model received from the REST API.

Angular services are objects that get instantiated just once during the lifetime of an application. They contain methods that maintain data throughout the life of an application and communicate with the backend through REST API calls.



## **BIBLIOGRAPHY**

<https://www.geeksforgeeks.org/spring-boot-architecture/>

<https://angular.io/start>

<https://spring.io/guides/tutorials/spring-security-and-angular-js/>

<https://docs.docker.com/get-started/>