Online Energy Utility Platform

Distributed Systems Laboratory, 2022-23

Student: Keresztes Beáta

Group: 30441

Lab assistant: Cristina Pop

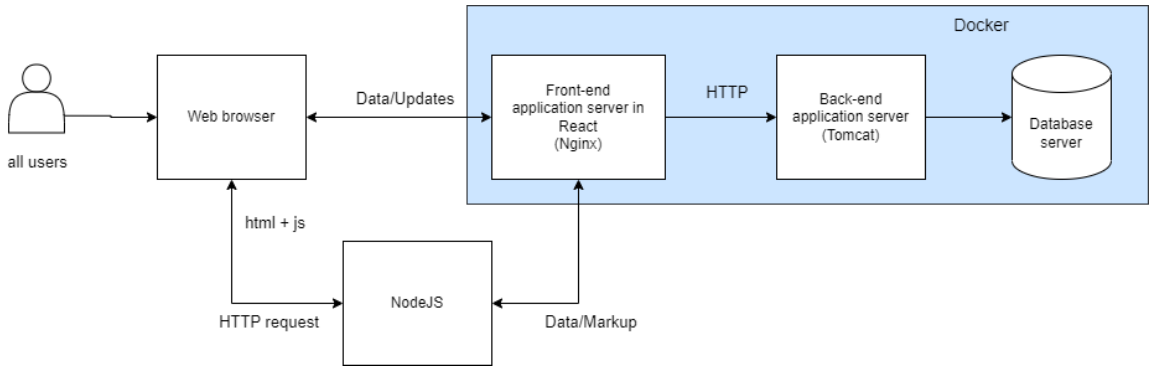
Conceptual Architecture

The main goal of the system is to provide the functionalities related to managing the energy consumption of the connected devices as well as provide a user interface for managing the devices and users, as well as visualizing the measurements data.

The responsibilities can be divided between 3 main components:

- the back-end application, responsible for providing the basic functionality
- the front-end application, responsible for providing the user interface
- the database server, responsible for persisting the data related to the users, devices and measurements.

Based on this responsibility division between the components of the system, it is obvious that the preferred architectural model for designing the system should be a layered architecture, namely the 3-tier model.

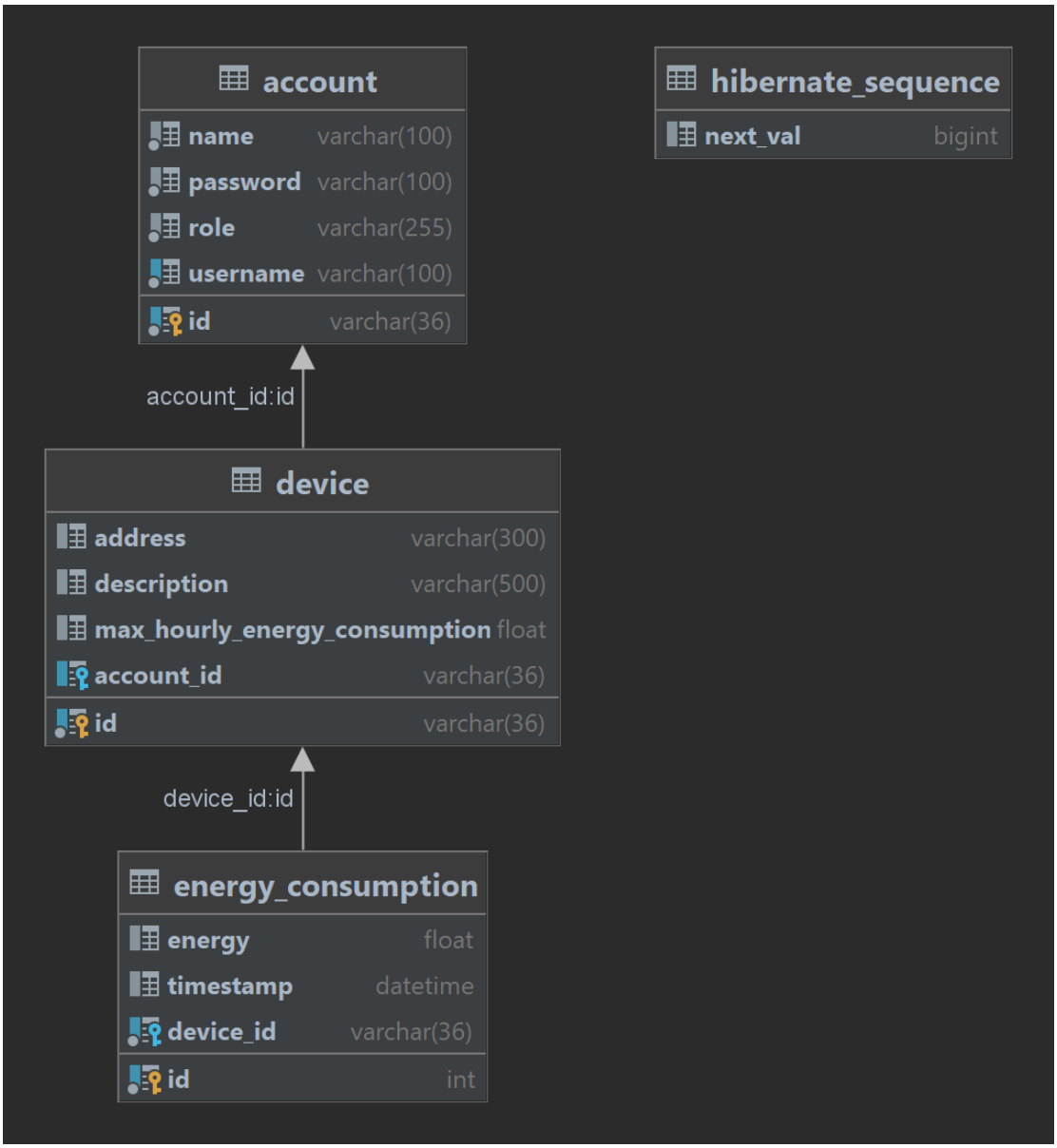


DB Design

The database contains 3 entities which hold information about the users, the smart devices and the corresponding energy consumption on an hourly basis.

There is a one-to-many relationship between the user and device entities, a user can have 0 or more devices associated but a device can only belong to 1 user. Similarly, there is a one-to-many relationship between the devices and the energy_consumption entities, as a device can have multiple data collected from it every hour, and each collected data belongs to only 1 device, measured at a given timestamp.

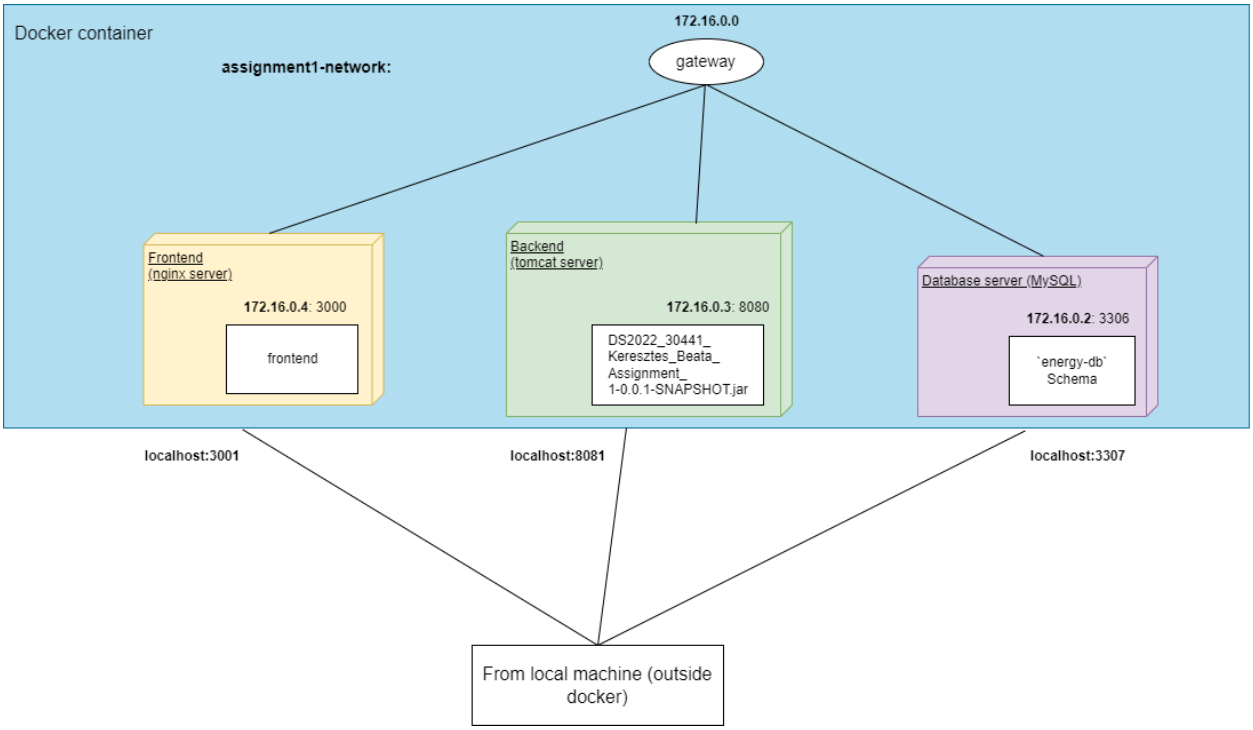
It also contains an aggregate view of the joint table between the devices and the energy consumption in order to collect more easily the total amount of energy consumed by multiple devices belonging to the same user.



Diagrams

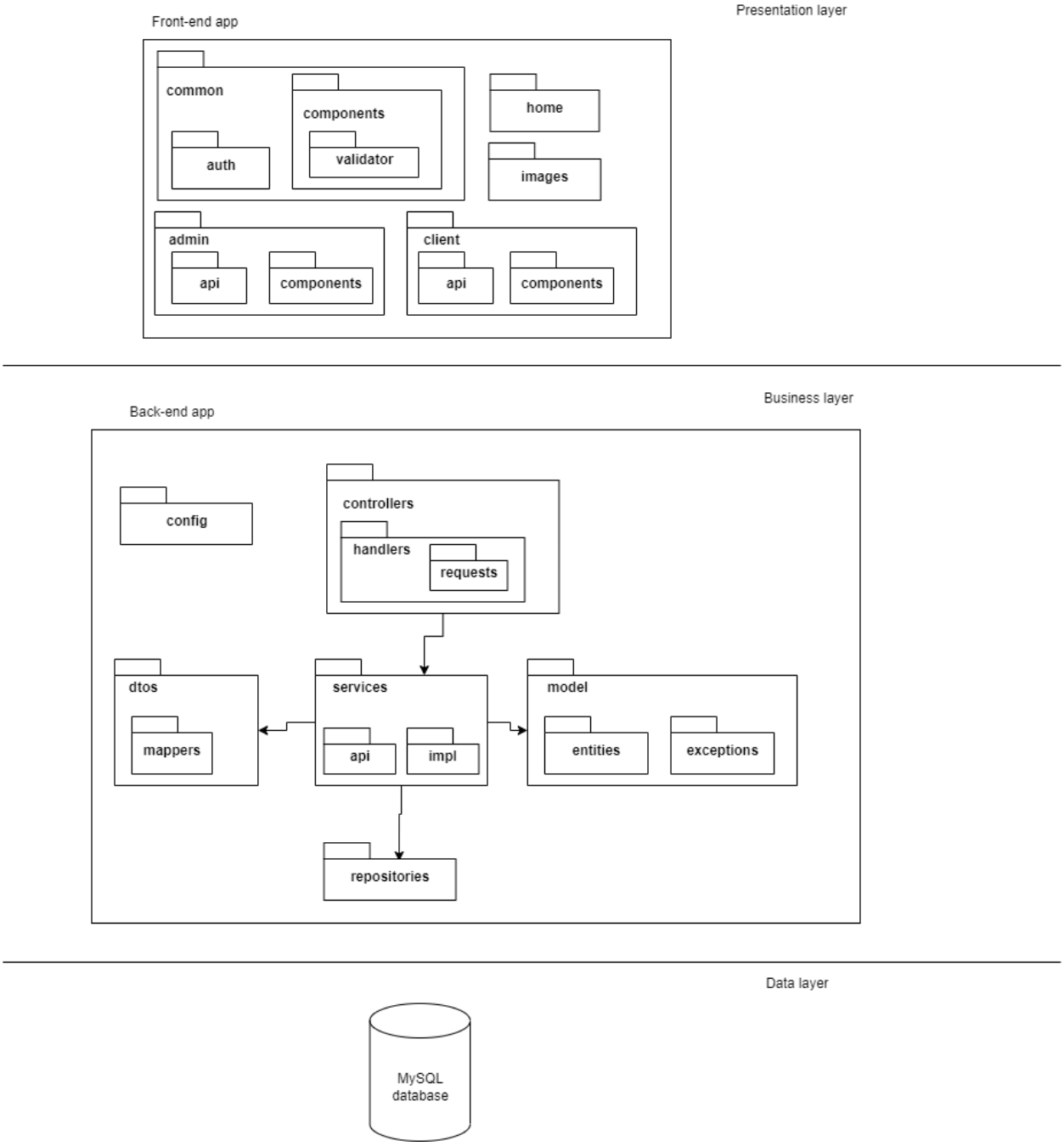
Deployment Diagram

A high-level view of the main components (hardware and software) of the system, modeling the physical architecture of the system or how the processing/services are distributed among the nodes:



Package Diagram

A lower-level view of the system’s components, down to the package level:



Class diagram

A fine-grained view of the system's architecture, down to the level of the individual classes: