Distributed Systems

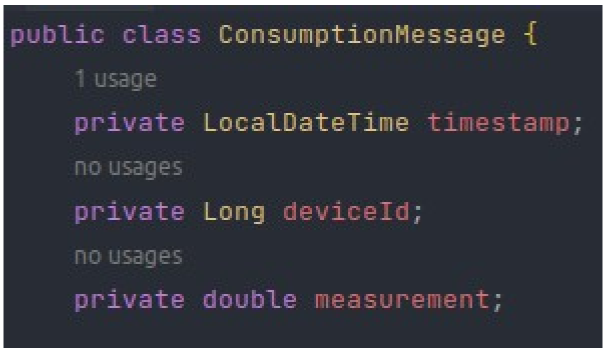
Online Energy Utility Platform

Assignment 2

Bîndilă Adrian-Ioan
30443

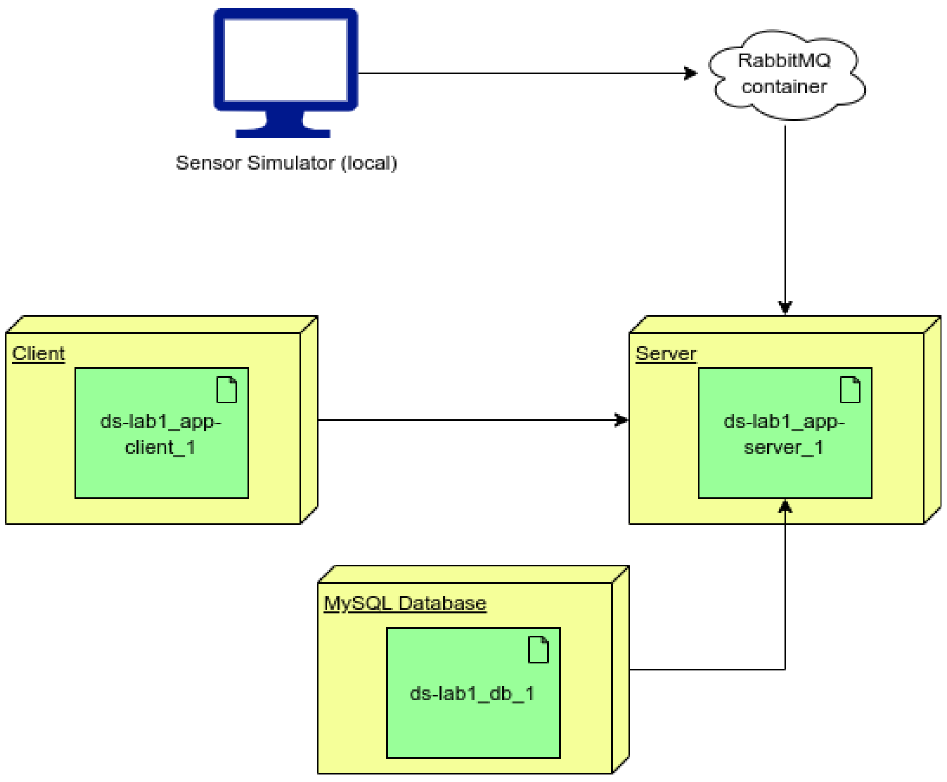
The Online Energy Monitoring Platform will now be extended in order to accommodate simulating receiving data from the devices that a user owns. In this scenario a sensor simulator module will be added, which will send data to our server. In order for the server to receive the data without having to continuously poll for it, a message broker will be used. RabbitMQ, and its Java-based implementation, called AMQP are message brokers meant to mediate between multiple real-time readings and allow their centralization through queues. As such, our sensor simulator will send energy consumption data to an existing rabbitMQ queue, while our server will listen for new data being pushed through the queue, on the other end.

The Sensor simulator module is configured using a text file, in which we simply insert the id of the device for which the measurements belong. The simulator is started and will randomly select an energy reading from the file sensors.csv. The reading will be sent once every 10 minutes, however this is adjustable within the code. A message sent from the sensor simulator through to the message queue will be contained within the EnergyConsumptionTimestamp object:

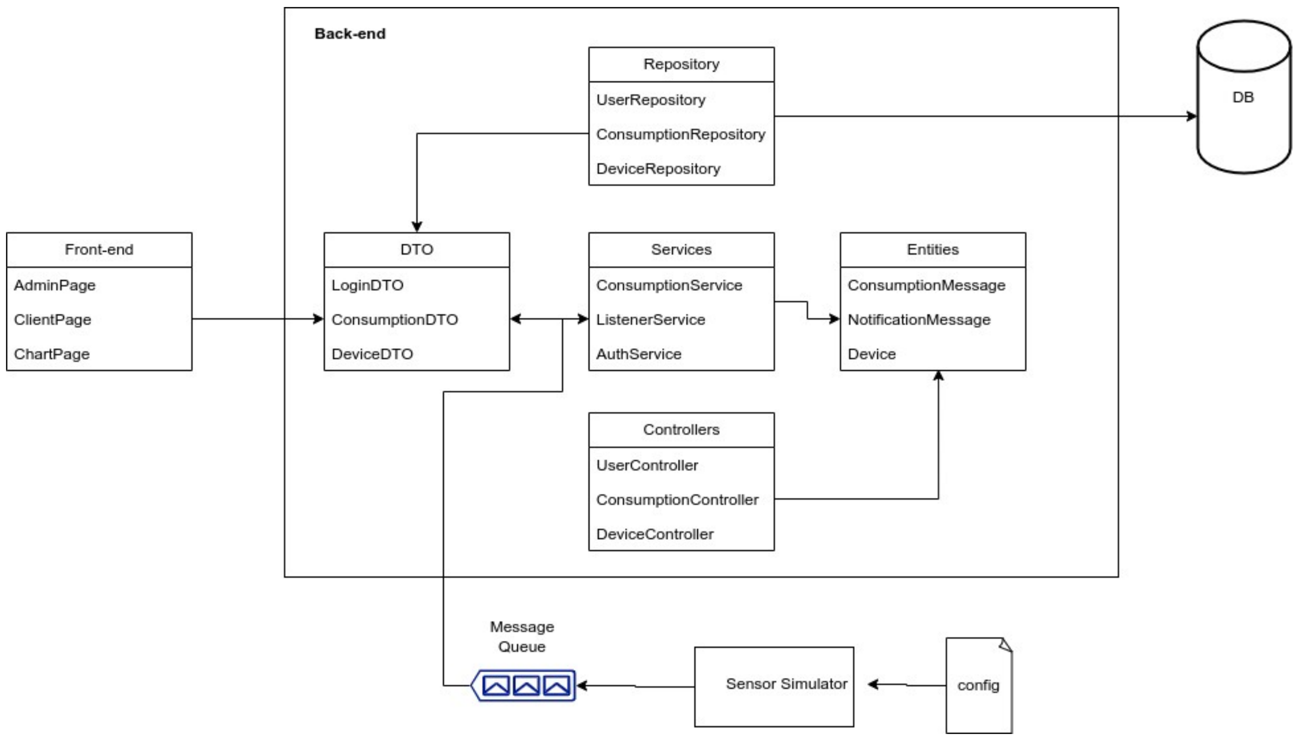


```
public class ConsumptionMessage {
    1 usage
    private LocalDateTime timestamp;
    no usages
    private Long deviceId;
    no usages
    private double measurement;
```

This object is converted into JSON using the ModelMapper object from the Jackson library. This allows us to send it into plain text to our server, via the RabbitMQ message broker. Once it arrives, the message is intercepted by a ListenerService, which then converts it into an EnergyConsumptionTimestamp object, which is then saved into our MySQL Database, while also being converted into a Notification message, in order to send a notification with information to the client about which device has been exceeding the consumption limit.



Using Websockets, the user is notified by the server if a device has exceeded the maximum hourly consumption limit. As such, the platform is designed to provide real-time notifications to users in a web application. It uses WebSockets as the communication protocol to achieve low latency and high-performance notifications. From this point of view, each user is subscribed to a topic, which represents a channel through which messages are sent. Since we intend to notify only the user in possession of the device which has exceeded consumption, we create a topic which is based on the username of each user. As such a logged in user receives notifications only for their devices.



Back-end
Repository
UserRepository
ConsumptionRepository
DeviceRepository
DB
Front-end
AdminPage
ClientPage
ChartPage
DTO
LoginDTO
ConsumptionDTO
DeviceDTO
Services
ConsumptionService
ListenerService
AuthService
Entities
ConsumptionMessage
NotificationMessage
Device
Controllers
UserController
ConsumptionController
DeviceController
Message Queue
Sensor Simulator
config