MINISTERUL EDUCAȚIEI ȘI CERCETĂRII ȘTIINȚIFICE

UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

Facultatea de Automatica si Calculatoare

Sectia Calculatoare

Asynchronous Communication Sensor Monitoring System and Real-Time Notification

Documentation

Zilai Denis

Grupa 30244; Anul 4, sem 1

Distributed Systems

Contents

1. Project objective

Implement a component for Assignment 1 application based on a message broker middleware that gathers data from the smart metering devices, pre-processes the data to compute the hourly energy consumption and stores it in the database.

2. The analysis of the problem, modeling, scenario, use cases

Functional requirements:

➢ The message broker allows Smart Metering Device Simulator to act as messages producer and send data tuples in a JSON format.

➢ The message consumer component of the system processes each message and notifies asynchronously using WebSockets the client application.

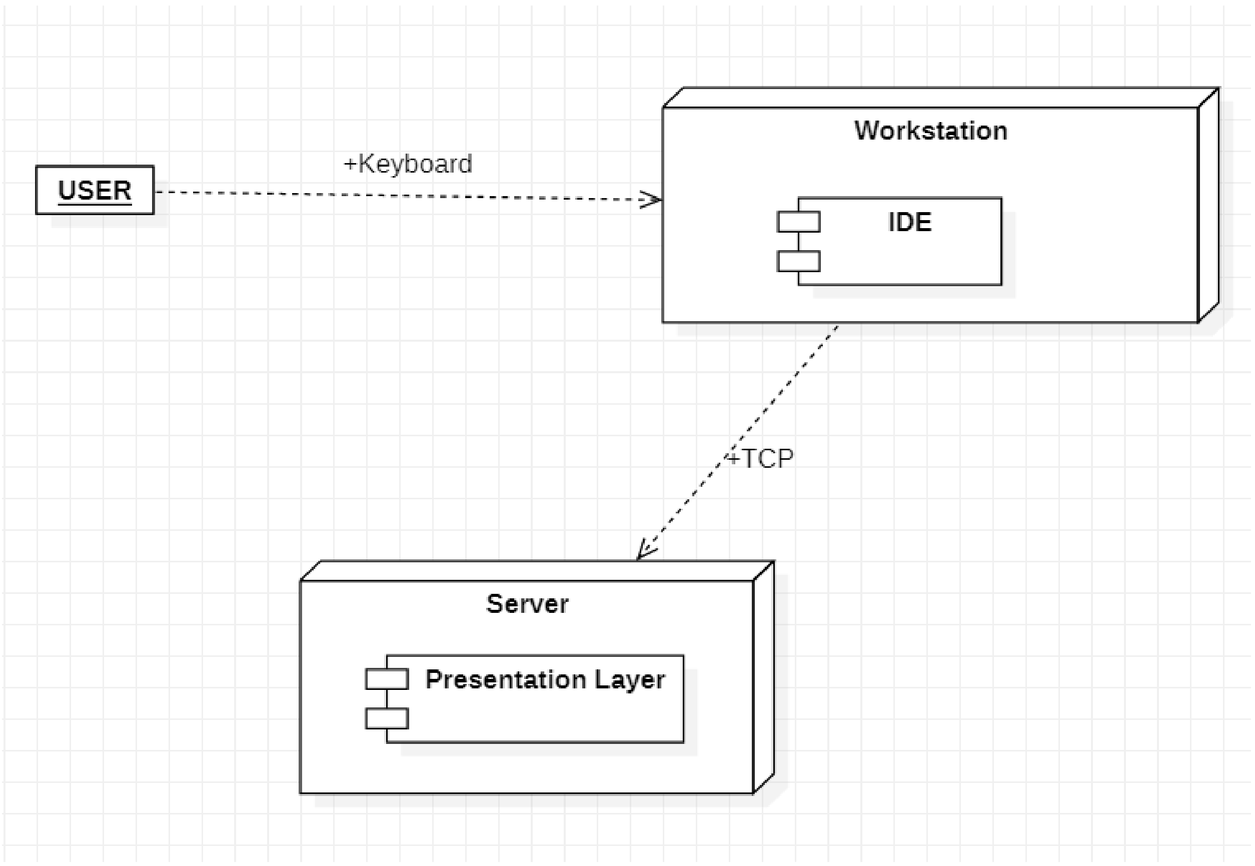
3. Design (design decisions, UML diagrams, class design, packages, User Interface)

The system was implemented using Web Sockets technology and Java Spring, and as frontend javaScript has been used, with HTML and CSS for a simple interface. Classes and modules:

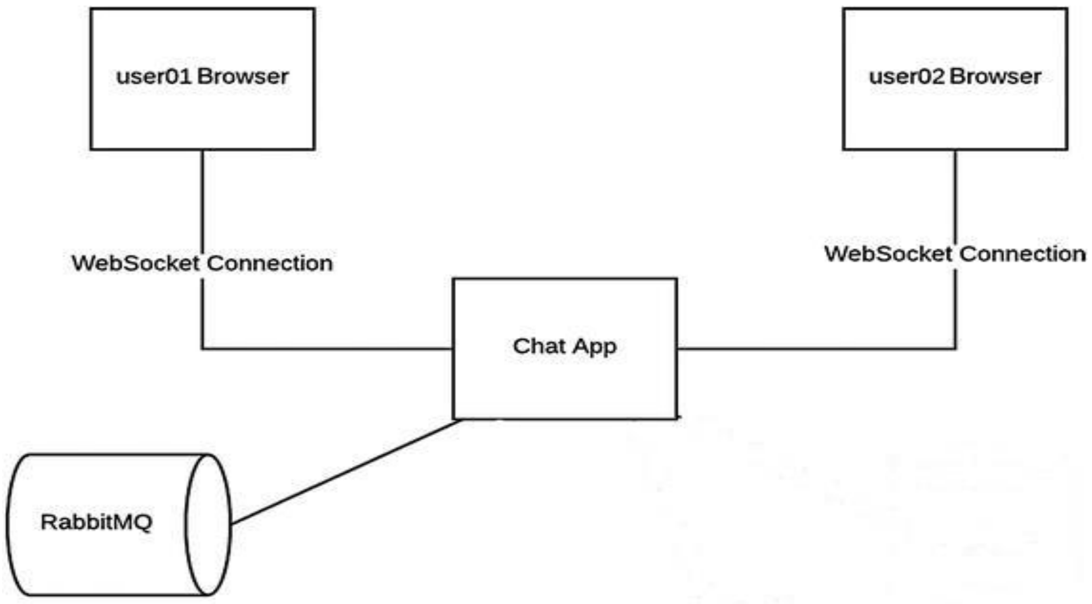
- Sender module: this module reads the data from the csv file and sends it to a rabbitmq queue
- Consumer: this class takes the data from the queue and sends it to the application, so that the data can be processed
- Main class: the class that runs the spring application
- AMQPController class: represents basically the server part of the application
- SensorData class: this class contains the object used for the encapsulation of the sensor data
- SocketTextHandler class: represents the broker (the part that is in charge of transmitting the messages

For the frontend part of the application, the technologies used are: HTML, CSS for a simple interface, and JavaScript for the functionality.
An important functionality of the application is the possibility of having multiple users using the application at the same time.

Deployment:



Architecture:



4. README

To run this application, start the sender module to start sending the data to the rabbitmq queue. Then, start the main class of the Assignment3Application, open a browser and access localhost:8080 to be logged in as a client. Afterwards, start the Consumer class to consume the data from the queue and process it. The consumed data can be seen in the console, as well in the browser console.