

SISTEME DISTRIBUITE

Assignment 1

Online Energy Utility Platform

Student: Talpos Andreea Georgiana

Grupa: 30643

**Diagrama bazei de date**

Diagram, schematic

Description automatically generated

Baza de date conține 3 tabele: users – în care sunt stocate datele despre utilizatorii platformei, device – conține datele despre device-urile asociate utilizatorilor și measures – cu informații despre energia consumată a fiecărui device. Între tabelele users și device există o legătură “one to many”, adică fiecare utilizator poate avea unul sau mai multe dispozitive asociate, iar între device și measures am creat, de asemenea, tot o legătură “one to many” – fiecare device având mai multe măsurători pentru diferite momente de timp.

In tabela cu masuratori, daca aplicatia Message Producer este activata, se vor adauga incontinuu valori reprezentand valorile citite de pe coada cu ajutorul RabbitMq.

**Arhitectura conceptuală**

Diagram

Description automatically generated

Sistemul este structurat pe trei nivele: nivelul de prezentare, logica de business si persistenta. Utilizatorii interactioneaza cu interfata aplicatiei, unde, pentru a se conecta trebuie sa-si introduca corect credentialele, dupa care sunt redirectionati pe pagina corespunzatoare rolului: Administrator sau Client. In functie de rol, un utilizator poate face actiuni diferite: clientul isi poate gestiona contul personal si lista de device-uri asociate, iar administratorul se ocupa de gestiunea clientilor si a dispozitivelor – facand asocieri intre dispozitive si clienti.

In plus fata de Assignment1, am adaugat o aplicatie Desktop care are rolul de MessageProducer. In aceasta aplicatie, sunt citite date dintr-un fisier – sensor.csv si sunt incarcate pe o coada prin intermediul serviciului RabbitMq. Acest proces se executa pentru fiecare utilizator conectat, implementarea fiind facuta cu thread-uri. Aceste valori reprezinta masuratori ale energiei consumate de deviceu-rile utilizatorului.

Utilizatorul este notificat de fiecare data cand o noua valoare a fost citita pentru device-ul sau daca valoarea maxima pentru device-ul respectiv a fost depasita. Pentru aceasta functionalitate am folosit WebSocket.

**Diagrama deployment**

Diagram

Description automatically generated

Docker este un cadru deschis de dezvoltare a aplicațiilor, conceput ca instrument de ajutor pentru DevOps și dezvoltatori. Folosind Docker, dezvoltatorii pot construi, împacheta, expedia și rula cu ușurință aplicații sub formă de containere simple, portabile și autosuficiente, care pot rula, practic, oriunde. Containerele le permit dezvoltatorilor să împacheteze o aplicație cu toate dependențele sale și să o implementeze ca pe o singură unitate. Oferind containere de aplicații construite în prealabil și autosustenabile, dezvoltatorii se pot concentra asupra codului aplicației și utilizării, fără a se preocupa de sistemul de operare de bază sau de sistemul de implementare.

Pentru deployment-ul local al aplicatiei am utilizat Docker, am creat cate o imagine pentru backend, frontend si postgres si am rulat aplicatia intr-un container docker. Daca procesul de rulare este pornit, aplicatia poate fi accesata din browser de pe <http://localhost:4200> fara a mai rula aplicatia din IDE.

Pentru rularea aplicatiei:

* Trebuie facut build la fiecare imagine pentru a ne asigura ca e ultima versiune incarcata
  + docker build -t backend .
  + docker build -t frontend .
  + docker build -t postgres .
* Pe urma rulam comenzile:
  + docker-compose build
  + docker-compose up