**Assignment 3**

**Energy Utility Platform**

Universitatea Tehnica din Cluj-Napoca

Facultatea de Automatica si Calculatoare

Sisteme distribuite

Onisa Bianca-Maria

Grupa 30644

1. Arhitectura conceptuala

Aceasta aplicatie reprezinta o platforma online de monitorizare a unui sistem de energie. Proiectul consta acum in 5 mari parti: baza de date, backend, frontend (de la primul assignment) si partea de Message Producer (aplicatia desktop care simuleaza un senzor, citind datele dintr-un fisier csv) si Message Broker (care este RabbitMQ, preluand datele de la Message Producer si pregatindu-le intr-o coada pentru a fi mai apoi procesate de catre backend). Platforma suporta 2 tipuri de utilizatori: administrator si client.

Administratorul are dreptul de a vizualiza, sterge, adauga sau modifica datele utilizatorilor, dispozitivelor si consumului de energie. Clientul isi poate vizualiza dispozitivele personale si consumul lor de energie sub forma unui chart.

Acestia se vor autentifica in aplicatie printr-un formular de login, dupa care vor fi redirectionati la pagina aferenta rolului. Administratorii nu vor avea dreptul sa acceseze paginile clientilor, iar la randul lor nici clientii nu vor avea dreptul sa acceseze paginile administratorilor. Desi administratorul are dreptul sa modifice datele userilor, acesta nu le va putea modifica parola si nici vizualiza.

Frontend-ul consta in urmatoarele pachete: home (contine pagina initiala cu cateva informatii referitoare la platforma), login (pagina unde utilizatorii isi vor introduce credentialele pentru a se loga in aplicatie), users (contine pagina cu tabelul user-ilor, form-ul pentru adaugarea unui user nou si cel pentru modificarea unui user deja existent in baza de date), device (contine pagina cu tabelul device-urilor, form-ul pentru adaugarea unui device nou si cel pentru modificarea unui device deja existent in baza de date), energyconsumption (contine pagina cu tabelul consumurilor de energie, form-ul pentru adaugarea unui consum nou, cel pentru modificarea unui consum si pagina folosita de clienti pentru a vizualiza chart-ul cu informatiile despre consumul de energie al dispozitivelor sale) si chat (contine autentificarea in chat folosind un nume si pagina chat-ului, cu mesajele private si mesajele de grup).

Message Producer este o aplicatie desktop simpla, care citeste datele din fisierul csv si o data la 5 secunde trimite cu ajutorul thread-urilor cate un set de date in format json la coada din RabbitMQ.

Message Broker-ul este RabbitMQ, care va prelua datele de la Message Producer, le va adauga intr-o coada, iar backend-ul le va citi la randul lui.

Backend-ul contine urmatoarele pachete: controllers (cererile facute de catre frontend vor fi trimise controller-elor), services (vor procesa cererile; este un layer intermediar intre cererile clientului si baza de date), repositories (acestea vor lucra cu baza de date prin intermediul query-urilor), security (clasele care manipuleaza credentialele si cripteaza parola), entities (obiectele bazei de date) si dtos (fac legatura intre repositories si entities). Diferenta fata de assignment-ul 1 este faptul ca controller-ul de energy-consumption contine si un Rabbit Listener. Acest listener “asculta” coada corespunzatoare din RabbitMQ si in momentul in care se adauga date in aceasta, datele vor fi parsate si adaugate in baza de date. In cazul in care senzorul transmite o valoare a consumului mai mare decat valoarea admisa, se va afisa un mesaj in timp real cu ajutorul web socket-urilor. Mai cotine si pachetul “chat” aferent assignment-ului 3, unde avem metodele folosite pentru primirea mesajelor private si publice din cadrul chat-ului, mesaje transmise cu ajutorul web socket-urilor.

1. Database design

A picture containing text, receipt

Description automatically generated

Fiecare user, device si energy-consumption se identifica unic printr-un ID. Un user poate sa detina 0, unul sau mai multe device-uri, iar un device poate sa aiba inregistrate 0, una sau mai multe energy-consumptions. Legatura dintre user si device se face printr-un foreign key, device-ul continue id-ul user-ului caruia ii apartine. In cazul in care un user se sterge din baza de date, campul de id\_user din device-ul sau va ramane gol. La fel si in cazul device si energy-consumption.

1. UML Deployment Diagram

Diagram

Description automatically generated

Clientul acceseaza aplicatia de frontend (React) prin intermediul unul browser Web. Cererea va fi trimisa catre backend (Spring boot cu REST services) care o va procesa. Aici va trece prin urmatoarele pachete: controller, service si repository. Din repository, cererea va fi trimisa catre baza de date, care va citi sau modifica informatiile aferente. Backend-ul va intoarce un raspuns catre frontend, iar clientul va putea vedea rezultatul actiunilor sale.

Senzorul este o aplicatie desktop careia nu i se face deploy. Va citi datele din fisierul csv si le va trimite Massage Broker-ului, adica aplicatiei RabbitMQ, care deja se afla pe site-ul web [CloudAMQP - RabbitMQ as a Service](https://www.cloudamqp.com/) si nu are nevoie de deploy. De aici backend-ul va prelua datele si le va procesa corespunzator.

1. Readme file

Pentru a porni aplicatia, va fi nevoie sa rulam initial backend-ul, iar dupa aceea frontend-ul folosind in terminal comanda “npm start”. Prima pagina va arata asa:

Graphical user interface, website

Description automatically generated

Cand utilizatorul doreste sa se logheze, formul care se va deschide va fi urmatorul (unde se vor introduce credentialele):

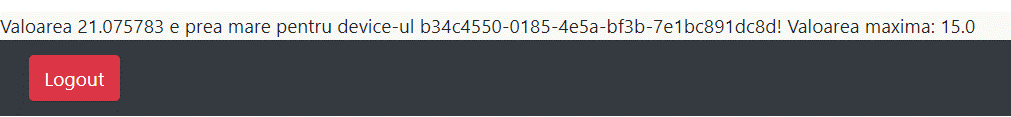
Graphical user interface, text, application, email

Description automatically generated

In cazul in care acestea sunt corecte, utilizatorul va fi redirectionat la pagina aferenta rolului sau (client sau admin). Daca credentialele nu sunt corecte, utilizatorul va fi retrimis la pagina initiala.

Utilizatorul logat nu va avea dreptul sa acceseze alte pagini care nu ii apartin sau pe care nu are drepturi.

Datele primate de la sensor vor fi citite si adaugate in baza de date. In momentul in care se citeste o valoare a consumului mai mare decat valoarea maxima permisa, se va afisa un mesaj in timp real care avertizeaza user-ul:



Userii au posibilitatea de a comunica intre ei, atat prin mesaje private (trimise unui anumit user), cat si prin mesaje publice (prin intermediul unui chatroom).

Acestia se vor autentifica in chat printr-un nume ales de ei:

A picture containing text, businesscard, screenshot

Description automatically generated

Odata intrati in chat, vor putea primi si trimite mesaje celorlalti useri sau in chatroom-ul public:

Graphical user interface, text, application

Description automatically generated