

**DISTRIBUTED SYSTEMS**

**Assignment 1**

**Request-reply**

**Communication Paradigm**

**Online Energy Utility Platform**

***Name:*** *Munteanu Cezar-Lucian*

***Grupa:*** *30242*

**Cuprins**

1. Specificatiile Proiectului
2. Instrumente utilizate
3. Functionalitati
4. Securitate
5. Design-ul architectural

* Baza de date
* Diagrama de clase
* Diagrama de deployment
* Diagrama UML

1. Dezvoltari ulterioare
2. Concluzie
3. Bibliografie

**Specificatiile proiectului**

S-a droit crearea unei aplicatii care sa ajute la manevrarea dispozitivelor si a masurilor effectuate de acestea dintr-o firma, cat si a utilizatorilor acestor dispositive. Aplicatia are doua tipuri de utilizatori: administrator si utilizator normal. Administratorul poate vizualiza toate dispozitivele si toate masuratorile si sa efectueze operatii CRUD asupra tuturor entitatilor din aplicatie, iar utilizatorul poate doar sa vizualizeze divice-urile si masuratorile sale. Ulterior se cere simularea unor senzori prin citirea unor date dintr-un fisier de tip .csv si incarcarea mesajelor intr-o coada de mesaje. Doi client se vor conecta la aceasta coada si vor putea citi un mesaj de notificare in cazul in care consumption-ul pentru device este prea mare.

**Instrumente folosite**

Pe partea de backend s-a folosit Java Spring Boot. Aici limbajul de programare folosit este Java. Acest limbaj imi este unul cunoscut avand mai multe proiecte anterioare realizate cu ajutorul sau.

Pe partea de frontend s-a folosit React. Ca limbaj de programare pentru React am folosit sablonul cu typescript deoarece am fost mai comun cu el decat cel cu javascript in ciuda faptului ca cele doua seamana foarte bine.

Pentru baza de date s-a folosit Postgres SQL. In ciuda faptului ca a fost prima mea interactiune cu aceasta baza de date, m-am descurcat foarte bine sa o folosesc.

Pentru implementarea pattern-ului de tip Sender-Reciever am folosit o implementare pe cloud cu ajutorul RabbitMQ si am integrat conexiunea in tema precedenta pentru a avea access la data base si toate functionalitatile precedente.

**Functionalitati**

La un prim contact cu aplicatia, utilizatorului i se afiseaza pagina de log in. Acesta se poate loga daca are deja un cont. Daca nu, pe header-ul paginii exista un buton pentru register, prin care utilizatorul va fi redirectionat catre o pagina de register. De mentionat este faptul ca toti utilizatorii care se inregistreaza din aplicatie pot fi doar utilizatori de tip USER. Adminii pot fi creati doar de alti admini.

Dupa ce s-a efectuat logarea, daca utilizatorul este de tip ADMIN acesta va fi redirectionat pe pagina de USERI unde v-a putea efectua operatii de tip CRUD asupra utilizatorilor. Daca utilizatorul este de tip USER acesta va fi redirectionat catre pagina cu divice-uri. Atat administratorii cat si utilizatorii normali au in header dupa logare butoane ce le permit navigarea catre paginile destinate divace-urilor si measurement-urilor, administratorul avand in plus un buton pentru navigarea catre pagina de users. De mentionat este faptul ca utilizatorii pot vedea pe pagina de devices doar device-urile corespunzatoare lor, iar administratorii pot vedea toate device-urile din baza de date.

Pentru simularea senzorilor am folosit 2 dispozitive din data base care au 2 utilizatori assignati Masuratorile sunt facute din 10 in 10 minute in baza de date si pentru a se determina daca energy consumption este mai mare decat current consumption se va lua valoarea maxima de la 6 masuratori si se va compara cu max consumption. Daca valoarea este depasita se va notifcia user-ul prin intermediul unui mesaj de tip alert.

**Securitate**

Din punct de vedere al securitatii, orice utilizator nu poate accesa orice pagina a aplicatiei daca nu este logat, acesta fiind redirectionat de fiecare data catre pagina de log in. Odata logat, utilizatorul poate accesa paginile aplicatiei prin URL sau prin butoanele corespunzatoare. De mentionat este faptul ca utilizatorii normali nu pot accesa paginile destinate utilizatorilor de tip administrator.

**Designul architectural**

* **Baza de date**

Cum am precizat mai sus, pentru baza de date am folosit postgres. Aici am create 4 tabele

*Tabela de users –* aceasta tabela contine toti utilizatorii aplicatiei;

*Tabela de role –* aceasta tabela contine toate rolurile utilizatorilor aplicatiei, iar intre ea si tabela de users exista o relatie de one to one;

*Tabela de devices –* aceasta tabela contine toate device-urile existente;

*Tabela de measurements –* aceasta tabela contine toate measurements-urile corespunzatoare fiecarui device;

Graphical user interface, application

Description automatically generated

* **Diagrama Use-case**

Diagram

Description automatically generated

* **Diagrama UML/clase**

A screenshot of a computer

Description automatically generated with medium confidence

**Dezvoltari ulterioare**

Ca si imbunatatiri pentru aplicatie se poate face o Securitate mai solida pe partea de backend. Pe partea de front se pot face imbunatatiri pe partea de interfata grafica cat si modul cum functioneaza aceasta.

**Concluzie**

In opinia mea, aceasta aplicatie este una foarte folositoare pentru firmele ce au dispositive de genul.

Pe partea de implementare, a fost destul de greut avand in vedere ca nu eram foarte familiarizat cu anumite librarii din spring boot.

Pe partea de deploy am intampinat mai multe dificultati, dar in final am reusit.