



UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

Sisteme Distribuite

Assignment 2

Asynchronous Communication
Sensor Monitoring System and Real-
Time Notification

Nume: Timis Iulia Georgeana

Grupa:30242

Cuprins

1. Tema Proiectului
2. Functionalitate
3. Instrumente utilizate
4. Securitate
5. Designul aplicatiei
 - 5.1 Diagrama UML
 - 5.2 Baza de date
 - 5.3 Diagrama de clase
 - 5.4 Diagrama de deployment
6. Dezvoltari ulterioare
7. Concluzie
8. Bibliografie

1. Tema Proiectului

Aceasta aplicatie are drept scop ajutarea clientilor si a managerilor dintr-o firma sa gestioneze mai bine dispozitivele introduce de ei si masuratorile acestora. Exista doua tipuri de utilizatori ai aplicatiei: clientii si administratorii. Clientii pot sa isi vizualizeze lista de dispozitive introduce si masuratorile lor, iar administratorii pot crea, sterge si actualiza clientii, dispozitivele si masuratorile lor.

2. Functionalitate

O data cu pornirea aplicatiei, utilizatorul va vedea pagina de login. In cazul in care rolul utilizatorului este de client, va vedea doar o pagina cu dispozitivele lui si va putea adauga altul nou, altfel, daca este administrator, va avea functionalitati de creare, stergere, actualizare si vizualizare a tuturor clientilor, dispozitivelor si masuratorilor. O data cu apasarea butonului asociat acestor functii, el va fi redirectionat pe pagina destinata functionalitatii aleasa. De asemenea, el se poate sa apese butonul de Go Back care il va redirectiona pe pagina de Login.

3. Instrumente Utilizate

Aplicatia mea este alcatuita din 2 parti: backend si frontend.

Pentru backend am folosit Java Spring Boot in IntelliJ IDEA. Partea aceasta nu a fost dificila deoarece am mai lucrat in Java pe parcursul facultatii. Pentru frontend am folosit VisualCode pentru crearea aplicatiei. Aici am vrut sa folosesc react cu tailwind, partea de login fiind realizata astfel. Pentru functionalitatile administratorului am folosit React deoarece mi s-a parut mai usor de utilizat.

Pentru baza de date am folosit una relationala: Postgres Sql.

Pentru implementarea cerintei acestei teme, am folosit o coada care este situata deja pe cloud de la rabbitMQ si am integrat conexiunea in prima tema pentru a putea avea acces la baza de date si la toate funcționalitățile acesteia.

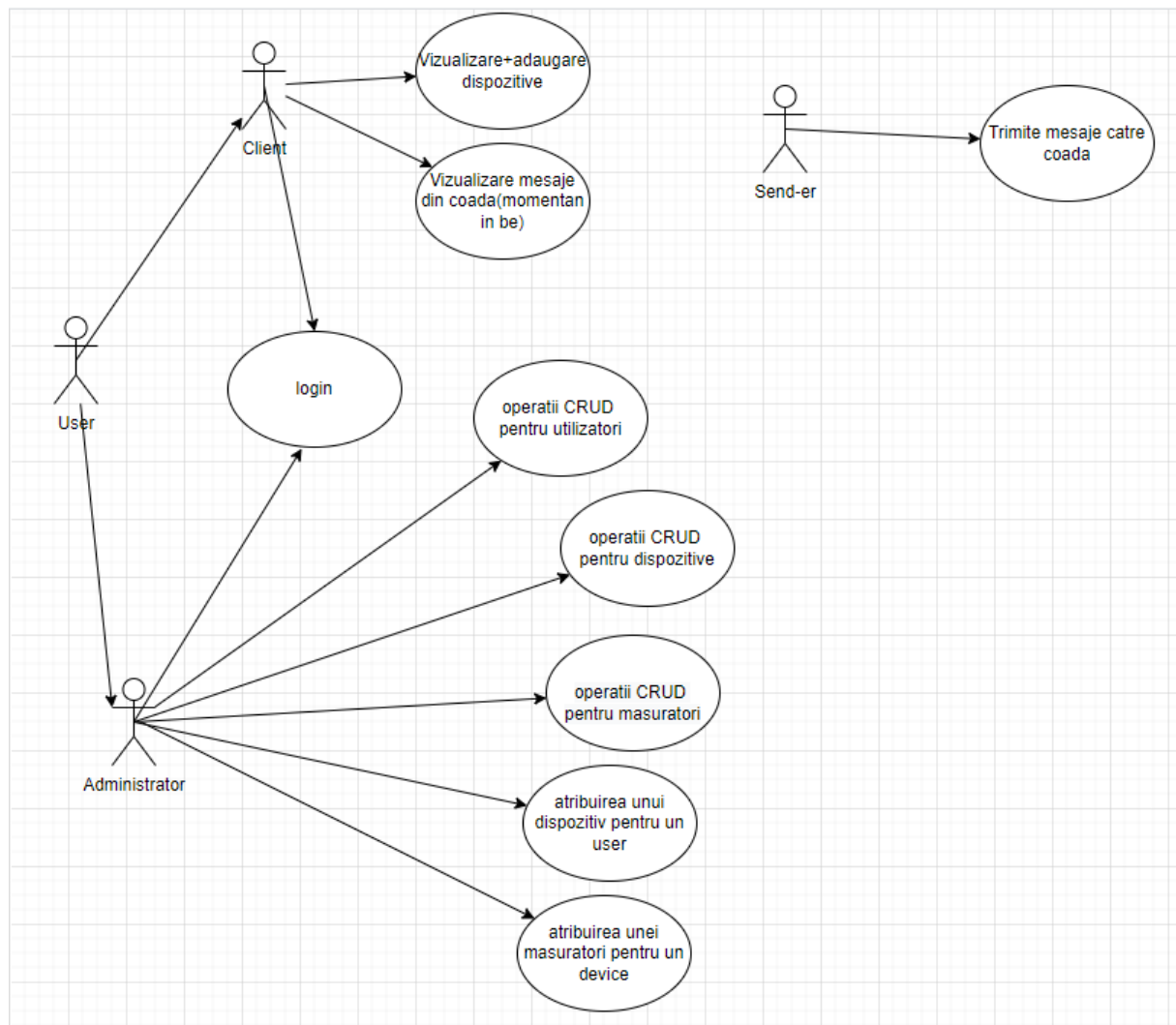
4. Securitate

Pentru Securitate am folosit sessionStorage. In cazul in care utilizatorul nu este logat, el nu are permisiunea de a accesa alte pagini ale aplicatiei. De asemenea, un utilizator cu rolul de client nu poate accesa paginile cu

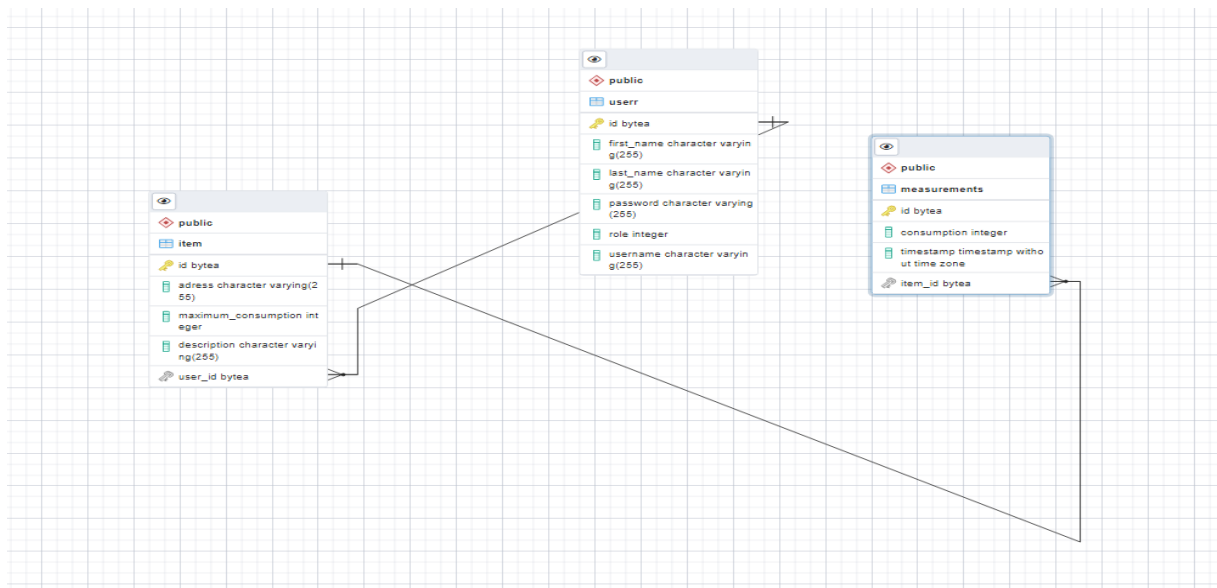
functionalitati ale administratorului. Acesta verifica daca id-ul celui care doreste sa acceseze aplicatia este al unui administrator, si daca nu, nu ofera permisiunea de access clientului.

5. Designul Aplicatiei

5.1 Diagrama UML

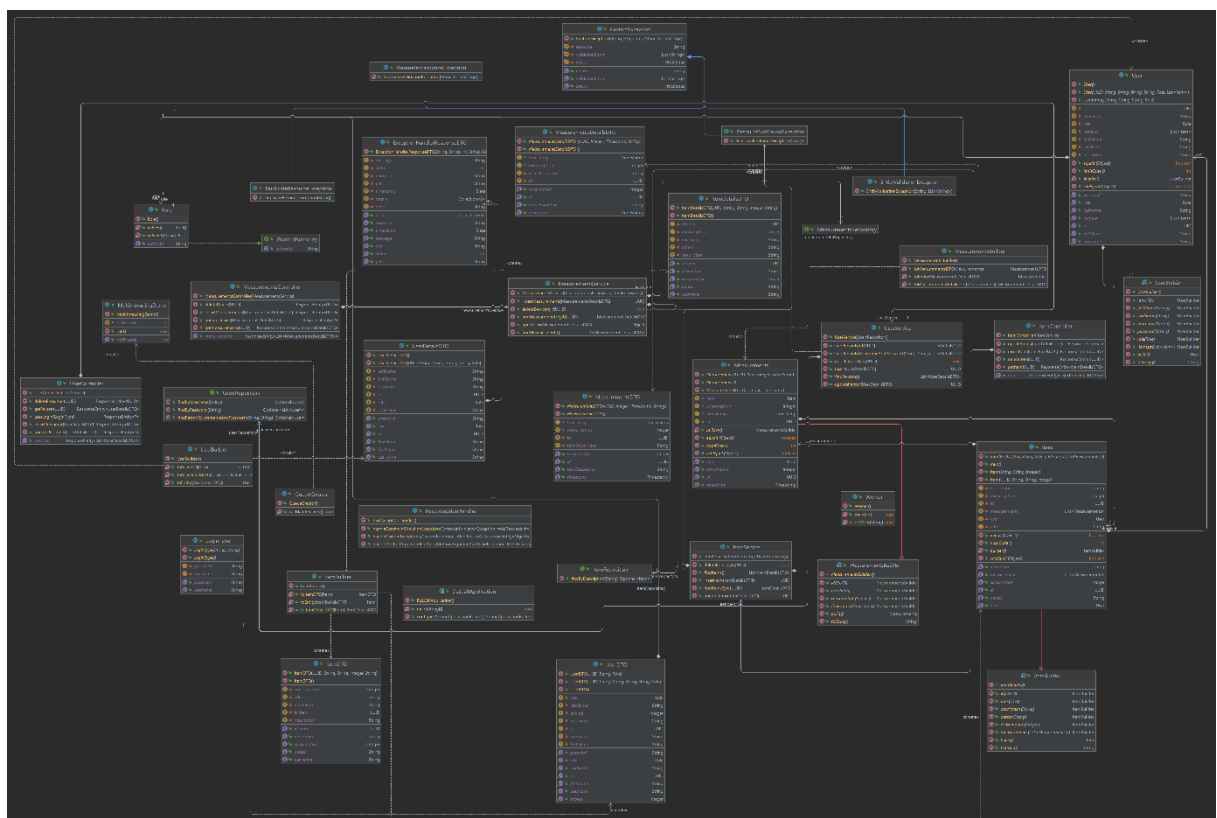


5.2 Baza de date

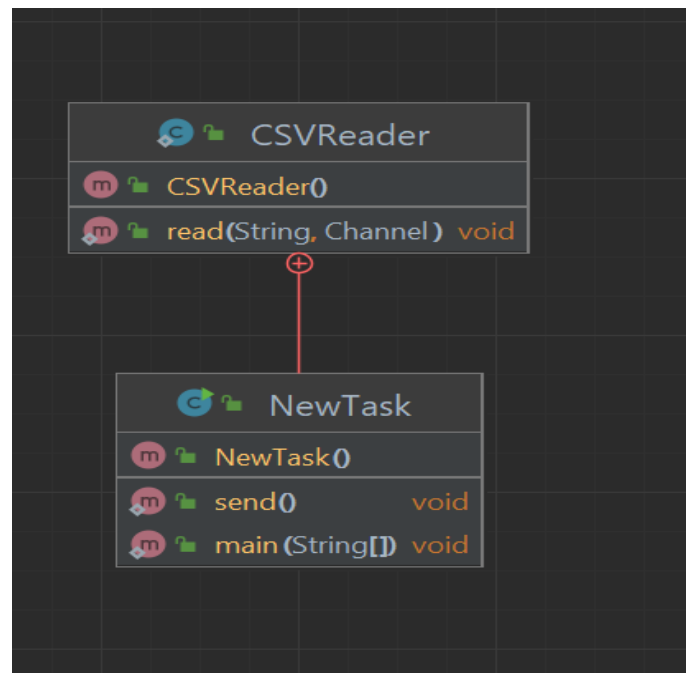


5.3 Diagrama de clase

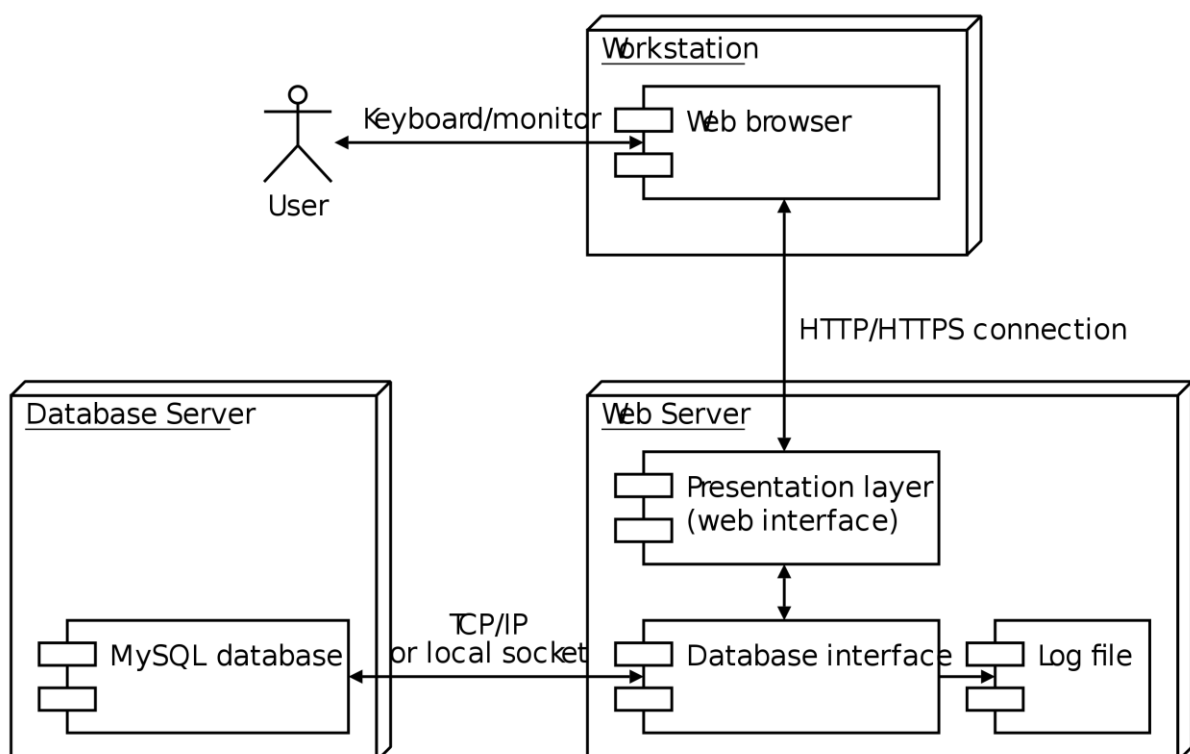
Aplicatia care cotine Reciver-ul



Aplicatia care contine Sender-ul



5.4 Diagrama de deployment



6. Dezvoltari ulterioare

Pe langa pagina de login, mai putem adauga o pagina de register pentru ca utilizatorul sa isi poata crea un cont. De asemenea, designu-ul aplicatiei poate fi gandit altfel, deoarece in modul in care este prezentat acum, este mai greu de utilizat de catre o persoana care utilizeaza prima oara aplicatia.

7. Concluzie

Personal, mi-a placut foarte mult sa muncesc la aceasta aplicatie deoarece am invatat foarte multe lucruri noi. A fost destul de grea partea de deploy deoarece am avut destul de multe erori in realizarea ei.

8. Bibliografie

<https://blog.loginradius.com/engineering/guest-post/modern-login-signup-form-tailwindcss-react/>

https://www.youtube.com/watch?v=R_sbuXBCviQ&t=3792s&ab_channel=CodeforInterview

<https://www.rabbitmq.com/tutorials/tutorial-two-java.html>

<https://www.cloudamqp.com/docs/java.html>

<https://techvidvan.com/tutorials/read-csv-file-in-java/>

W3School + Stack overflow