



**UNIVERSITATEA TEHNICĂ**  
DIN CLUJ-NAPOCA

# **Sisteme Distribuite**

## **Assignment 3**

Remote Procedure Call (RPC) Chat  
System for Client Support

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 administratorilor de a comunica intre ei. Programul presupune dezvoltarea unui sistem de chat pentru a oferi support clientilor platformei energetice in cazul in care acestia au intrebari legate de consumul lor de energie.

## 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 web sockets. Protocolul WebSocket permite interactiunea între un browser web (sau altă aplicație client ) și un server web cu o supraîncărcare mai mică decât alternativele half-duplex, cum ar fi pollingul HTTP , facilitând transferul de date în timp real de la și către server.

## 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

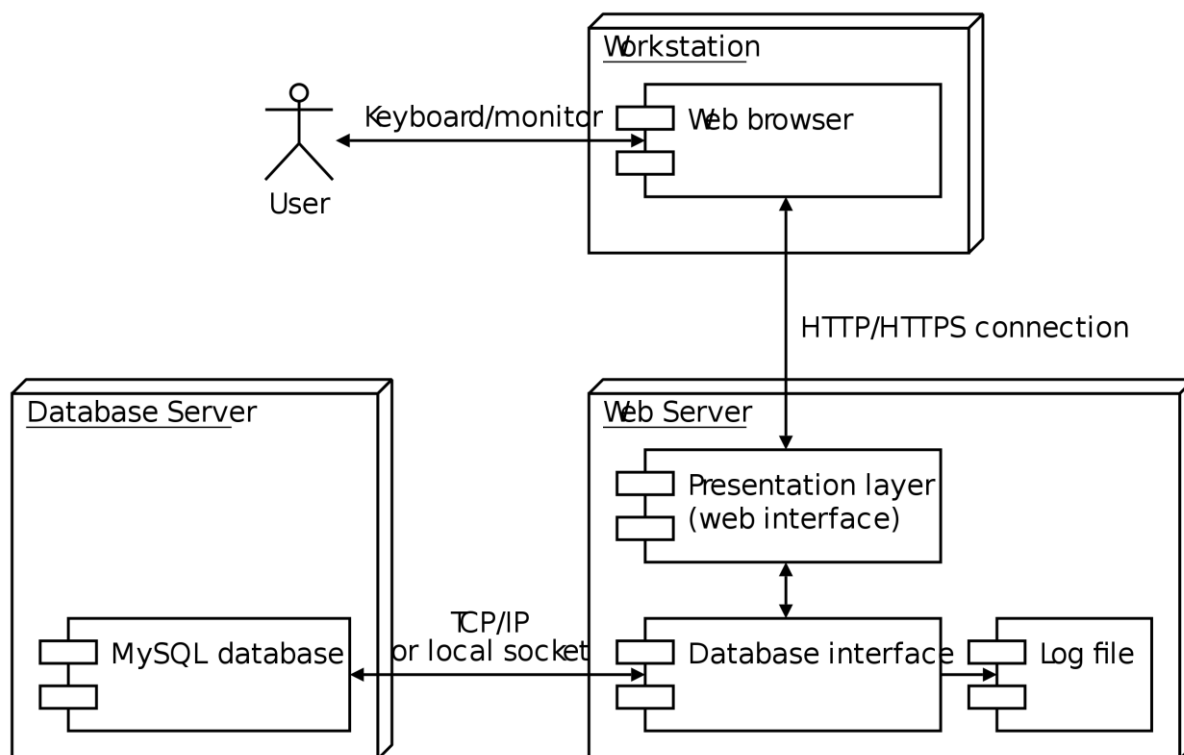
### 5.1 Diagrama UML



### 5.2 Baza de date



## 5.4 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.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/>

[https://www.youtube.com/watch?v=o\\_IjEDAuo8Y&t=4s&ab\\_channel=InvolveInInnovation](https://www.youtube.com/watch?v=o_IjEDAuo8Y&t=4s&ab_channel=InvolveInInnovation)

W3School + Stack overflow