UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

Sisteme Distribuite

-Partea 1-

Platformă de utilitate energetică online

NUME: Gavra
PRENUME: Anamaria
GRUPA: 302644

UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

Table of Contents

Cerințele rezolvate

Această fază a proiectului presupune utilizarea serviciilor REST pentru crearea unei aplicații backend și framework-uri bazate pe JavaScript pentru aplicația client; acestea având ca scop proiectarea și implementarea unei platforme online pentru a gestiona utilizatorii, dispozitivele de contorizare a energiei inteligente asociate acestora și datele monitorizate de pe fiecare dispozitiv.

Sistemul poate fi accesat de două tipuri de utilizatori după un proces de conectare: administrator și clienți. Administratorul poate efectua operațiuni CRUD pe conturile de utilizator, dispozitivele inteligente de contorizare a energiei înregistrate, și privind maparea utilizatorilor pe dispozitive. După efectuarea cartografierii, pentru fiecare dispozitiv, consumul de energie este stocat pe oră în baza de date.

Arhitectura conceptuală a platformei online.

Aceasta aplicație a fost creeata la baza unei arhitecturi de tipul “client-server”; astfel, are in componenta sa două aplicații: Prima parte are rolul de a implementa operatiile specifice unui **server** iar cealaltă este specifica unui utilizator de tip **client**.

Arhitectura client / server este o arhitectură de calcul producător / consumator în care serverul acționează ca producător și client ca consumator. Serverul găzduiește și oferă la cerere servicii de înaltă performanță, consumatoare de calcul. Aceste servicii pot include acces la aplicație, stocare, partajare fișiere, acces la imprimantă și / sau acces direct la puterea de calcul brută a serverului.

Arhitectura client / server funcționează atunci când computerul client trimite o solicitare de resursă sau proces către server prin intermediul conexiunii de rețea, care este apoi procesată și livrată clientului. Un computer server poate gestiona mai mulți clienți simultan, în timp ce un client poate fi conectat la mai multe servere simultan, fiecare oferind un set diferit de servicii. În forma sa cea mai simplă, internetul se bazează, de asemenea, pe arhitectura client / server, unde serverele web servesc mulți utilizatori simultan cu date de site.Astfel, ambele aplicații sunt structurate pe mai multe straturi, avand un stil arhitectural stratificat.

Stilul arhitectural stratificat este unul dintre cele mai comune stiluri arhitecturale. Ideea din spatele arhitecturii stratificate este că modulele sau componentele cu funcționalități similare sunt organizate în straturi orizontale. Ca urmare, fiecare strat îndeplinește un rol specific în cadrul aplicației. Stilul de arhitectură stratificat abstrage viziunea sistemului ca întreg, oferind în același timp suficiente detalii pentru a înțelege rolurile și responsabilitățile straturilor individuale și relația dintre ele.

1 Frontend

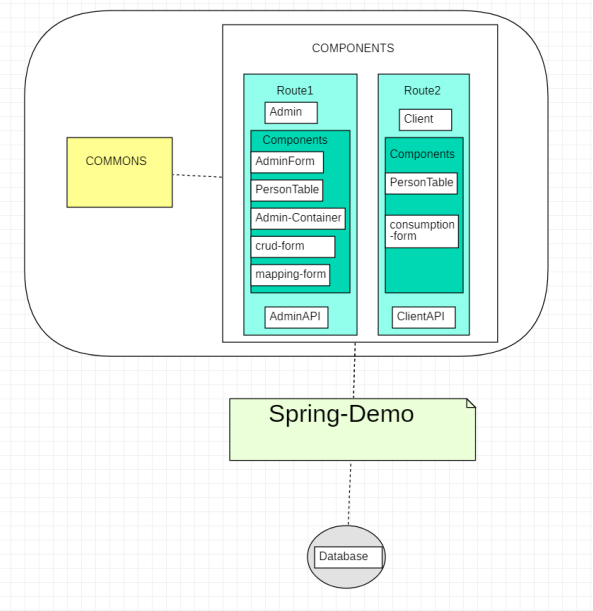

Figure 1: Frontend

2 Backend

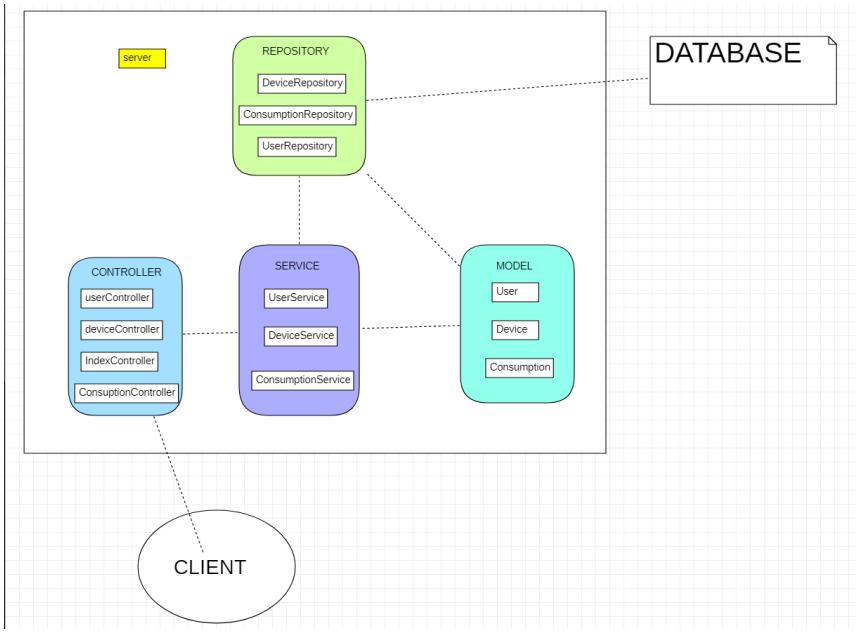

Figure 2: Back-end

Proiectare Bazei de date.

Pentru salvarea datelor am utilizat un sistem de baze de date relationale si anume PostgrreSQL.

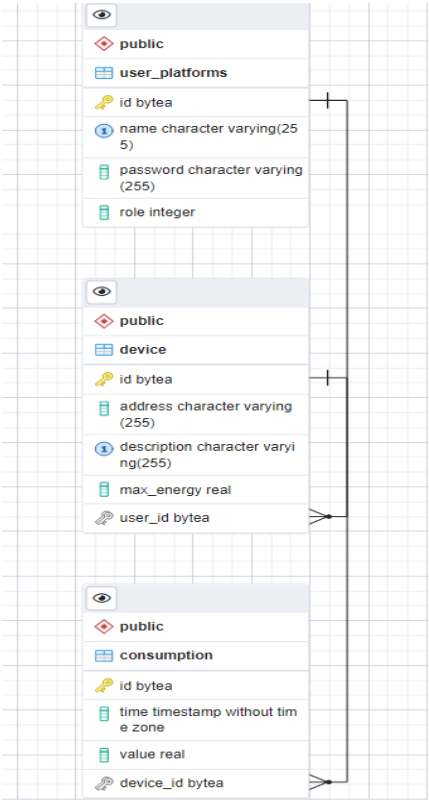

Figure 3: Database

Tabele

Baza de date a acestui proiect contine trei tabele:

- Tabelul “user_platforms” care contine credentialele utilizate de fiecare user pentru autentificare, dar si rolul acestuia.
- Tabelul “device” in care sunt memorate date specifice dispozitivelor inteligente care sunt monitorizate (descrierea, adresa la care este amplasat, consumul maxim de energie /ora)
- Tabelul “consumption” care memoreaza valorile masuratorilor periodice ale fiecarui device.

Relatii

Intre tabelele “user_platforms” si “Device” exista o relatie de tipul “one to many”, deoarece fiecare utilizator poate sa aiba unul sau mai multe duspozitive atribuite, dar fiecare device poate sa aiba un singur utilizator.

Acelasi tip de relatie exista si intre tabelele “device”, respectiv “consumption”, intrucat fiecarui dispozitiv ii corespund mai multe masuratori, dar o masuratoare este caracteristica unui singur dispozitiv inteligent.

Diagrama UML de Deployment

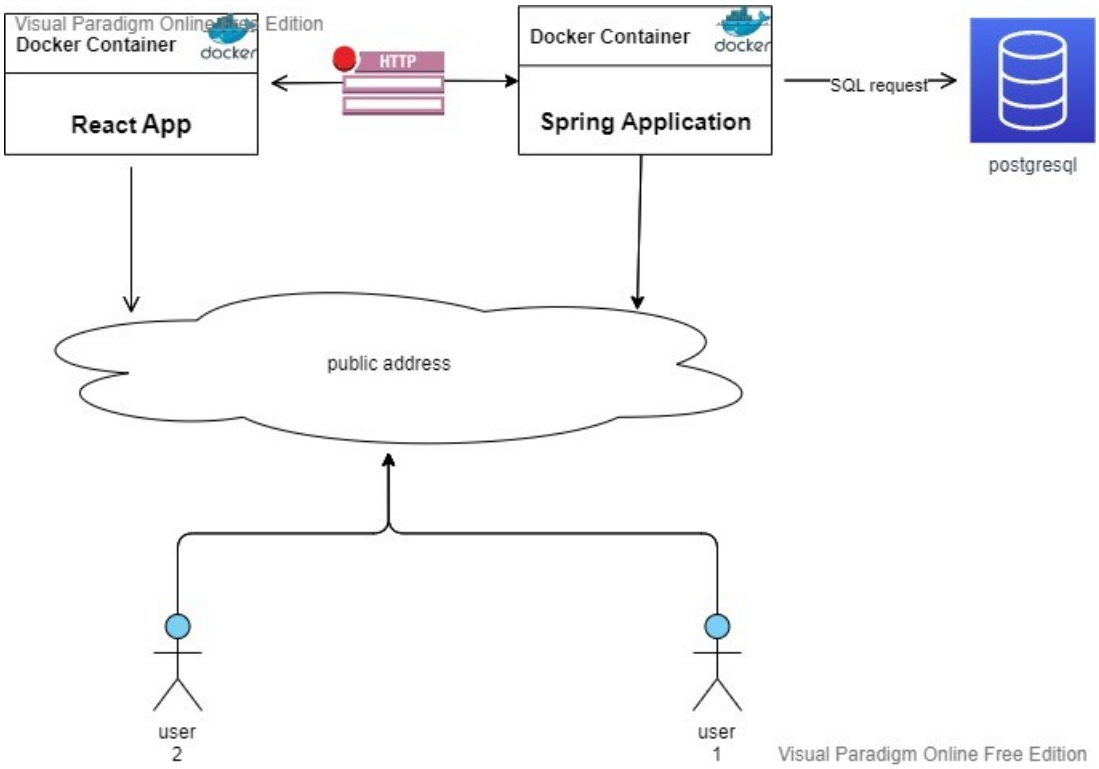

Figure 4: Deployment diagram

Considerente de construcție și execuție

Se lansează pe rând în execuție cele doua componente principale ale proiectului: clientul și serverul.

1 Baza de date

Pentru a putea porni proiectul, avem nevoie de o baza de date, creeata cu ajutorul sistemului de baze de date PostgreSQL, cu denumirea „energy”

2 Backend Spring Boot

Partea de backend a fost implementata în limbajul Java 11.

Pasii pentru pornirea proiectului sunt următorii:

- descarcat codul sursa de pe GitHub
- deschiderea proiectului într-un mediu de dezvoltare integrat ca IntelliJ IDEA
- setarea variabilelor de conexiune la baza de date (ip, username, password, etc.)

Aplicația ruleaza pe portul 8080. În cazul în care acesta este ocupat se poate întrerupe executia procesului de pe acel port deschizand „Command Prompt” ca administrator și ruland comenzile:

- „netstat -ano |findstr 8080” - pentru găsirea procesului ce ruleaza pe portul 8080
- „taskkill /PID id /F” - pentru oprirea procesului găsit anterior

3 Frontend React

Pentru realizarea acestei părți s-a utilizat Node Js, versiunea 14.

Pentru a putea rula proictul, trebuie urmați următorii pași:

- descarcat codul sursa de pe GitHub
- deschiderea proiectului într-un mediu de dezvoltare integrat ca IntelliJ IDEA sau Visual Studio Code
- deschiderea unui terminal și instalarea pachetelor necesare, executand comanda: npm install
- se lansează aplicația cu ajutorul comenzii : npm start
- acesarea din browser a adresei: localhost:3000