

# Online Energy Utility Platform Assignment 1

Profesor îndrumător:  
Alex Rancea

Student: Nicoară Marusea Ioana  
Grupa: 30242

Universitatea Tehnică din Cluj-Napoca  
Facultatea de Automatică și Calculatoare

Disciplina: Sisteme Distribuite

## 1. Cerintele aplicatiei

Prima a proiectului consta in realizarea unei paltforme online ce are ca scop a gestiona intr-un mod inteligent si eficient clientii acestei platforme si dispozitivele acestora ce au rolul de a monitoriza consumul de energie electrica. Lucrăm cu device-uri care sunt definite prin descriere, adresă, consum maxim per oră, consum mediu, utilizator, senzor.

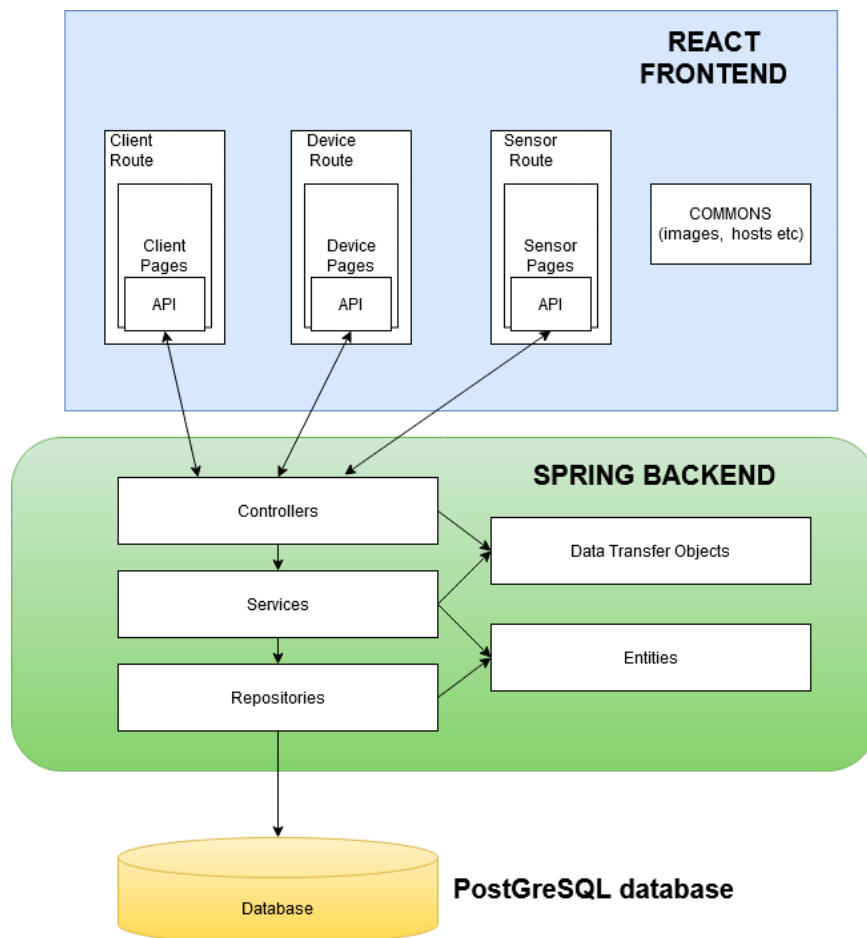
Tipuri de utilizatori:

- User al dispozitivelor (definit prin nume, adresă, rol)
- Administrator (aceleași câmpurii definitorii ca și userii)

Utilizatorul de tip administrator are posibilitatea de a efectua operatiile de baza (CRUD, create, read, update, delete) pe conturile userilor. În plus, acesta poate asocia device-urile unui utilizator.

Utilizatorul de tip user poate să își vizualizeze dispozitivele asociate și să genereze un grafic pe ore al consumului unui dispozitiv într-o zi.

## 2. Concept architectural



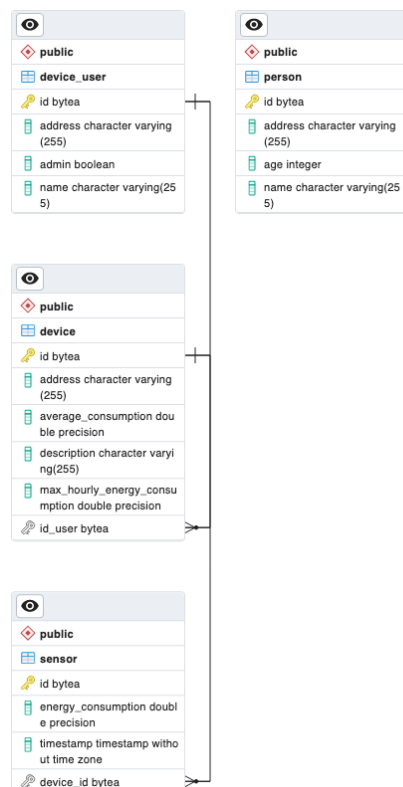
Pentru implementarea acestei prime componente din cadrul sistemului am ales a utiliza o arhitectura de tip client-server. Astfel am structurat proiectul in doua componente majore si anume backend si frontend.

Pentru implementarea componentei de backend am decis a utiliza limbajul de programare Java alaturi de framewor-ul Spring, datorita faptului ca acest framework reprezinta o sursa deschisa ce are ca scop simplificarea scrierii aplicatiilor in limbajul Java. Acest framework ofera multe functionalitati prin care se poate reduce timpul de dezvoltare al unui proiect datorita anotatiilor ce pot prelucra datele, pot crea constructori, metode de tipul getters si setters. De asemenea utilizarea interfetelor de tip JPA scurteaza durata de dezvoltare a unui proiect deoarece numai este nevoie sa se scrie metodele din repositories, acestea fiind preimplementate.

Pentru implementarea partii de frontend am ales a utiliza limbajul de programare JavaScript alaturi de biblioteca React. Javascript este un limbaj de programare care este una dintre tehnologiile de bază ale World Wide Web, alături de HTML și CSS. React (cunoscut și ca React.js sau ReactJS) este o bibliotecă JavaScript gratuită și open-source pentru construirea de interfețe cu utilizatorul bazate pe componente UI. Este întreținut de Meta și de o comunitate de dezvoltatori individuali și companii. React poate fi folosit ca bază în dezvoltarea de aplicații cu o singură pagină, mobile sau redate pe server cu cadre precum Next.js.

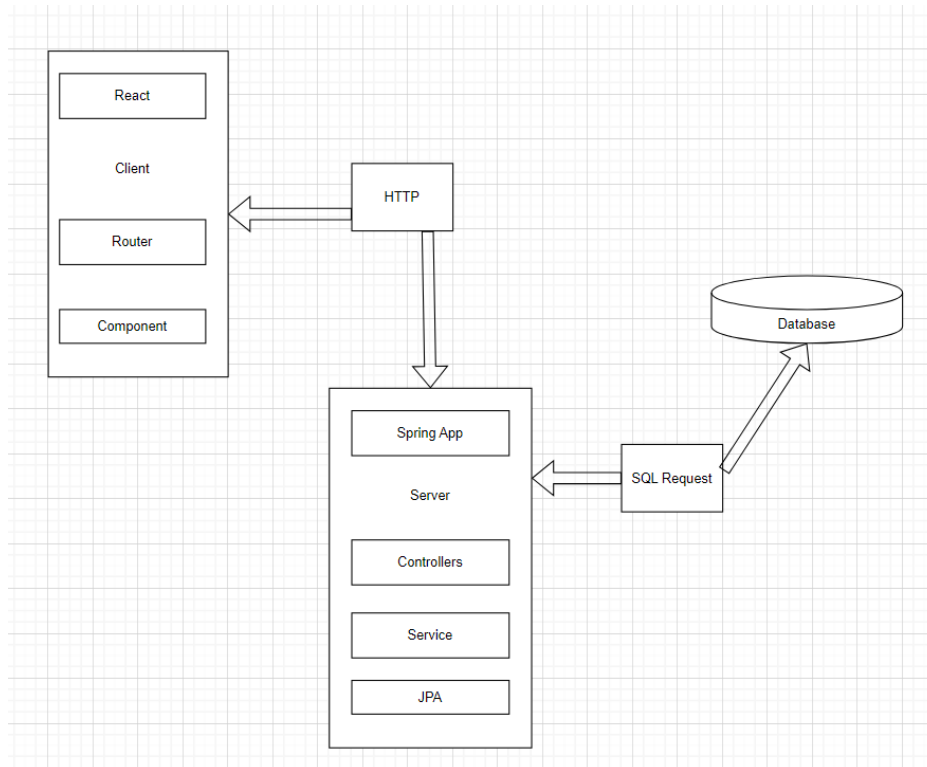
### 3. Diagrama bazei de date

Datele aplicatiei au fost stocate folosind utilitarul PgAdmin. Pentru fiecare componenta majora din cadrul proiectului in partea de backend exista cate un fisier de tip controller, service, dto, entity si repository. Intre tabelul de user si cel de device exista o relatie de many to one un client poate avea mai multe device-uri asociate, iar intre tabelul device si sensor exista o relatie de many to one, adica un device poate avea mai mulți senzori.



## 1. Diagrama UML pentru deployment

Acesta diagrama reprezinta comunicarea intre componenta client si server din cadrul arhitecturii client server. Componenta de backend comunica cu partea de frontend prin data transfer objects (DTO). Serverul de baze de date cu care se comunica prin intermediul repositoryurilor, acesta preia informatiile de la aceste interogari. Clientul comunica cu serverul prin intermediul protocolului HTTP. Partea de deployment a fost realizata utilizand *Docker desktop*.



## 1. Readme

Pentru rularea aplicatiei local, este necesar a se crea 2 foldere, unul pentru backend si unul pentru frontend. Se cloneaza repositoryul git aferent fiecăruia. Se deschide proiectul de backend si se ruleaza din IDE, iar apoi se deschide proiectul de frontend si se ruleaza in terminalul din IDE comanda: `npm start`. Se accesează `localhost:3000`.