SISTEME DISTRIBUITE

Energy Management System

Moşilă Luciana 30641/2

FACULTATEA DE AUTOMATICA
SI CALCULATOARE
2024

CONTENTS

DESCRIERE GENERALĂ

Aplicatia este o platformă web distribuită, destinată gestionării utilizatorilor și a dispozitivelor dintr-un sistem de măsurare inteligentă a consumului de energie. Există două tipuri de utilizatori: **administratorii**, care au acces complet pentru a gestiona utilizatorii și dispozitivele, și **utilizatorii standard**, care pot vizualiza și gestiona doar dispozitivele proprii.

FUNCȚIONALITĂȚI PRINCIPALE:

Autentificare și autorizare: utilizatorii pot accesa funcționalități conform rolului lor.
Operațiuni CRUD pentru administratori: aceștia pot crea, citi, actualiza și șterge utilizatori și dispozitive.
Vizualizarea dispozitivelor: utilizatorii standard pot vizualiza detalii despre dispozitivele lor și consumul energetic asociat.

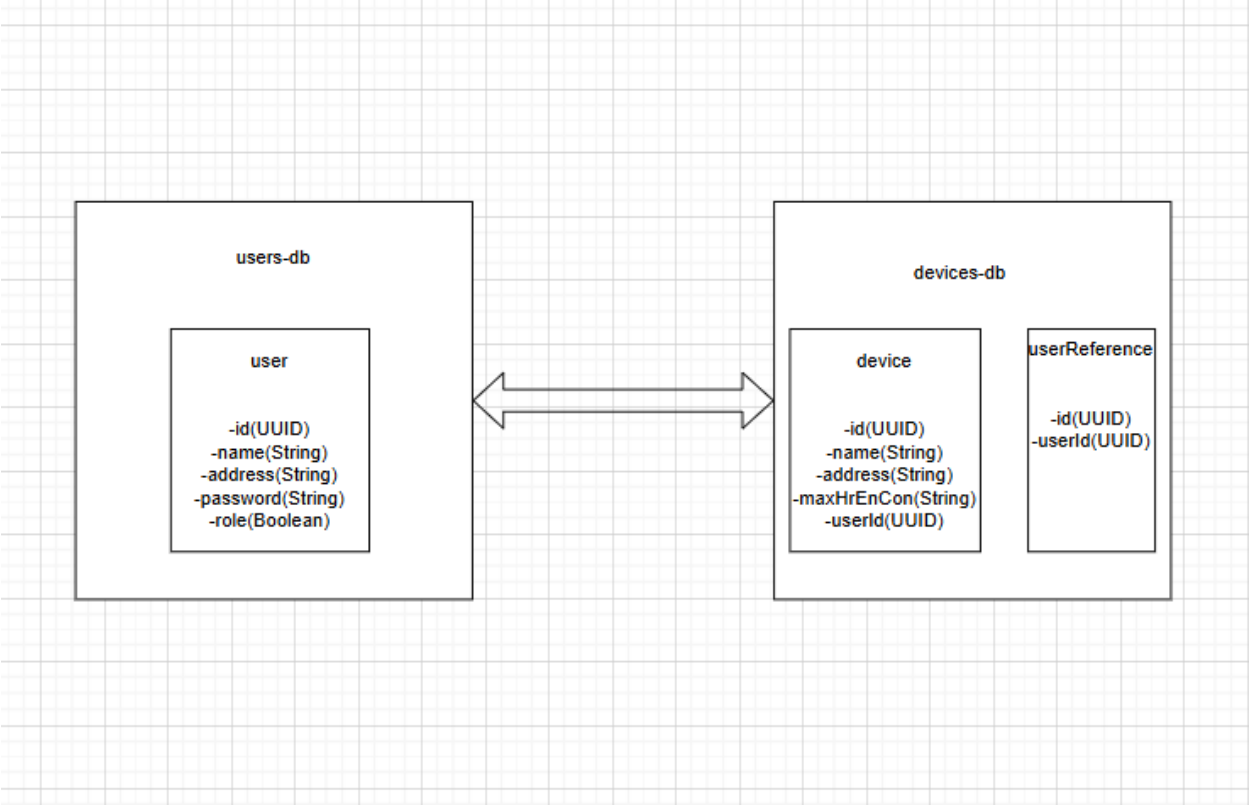
Administratorii pot efectua operațiuni CRUD (creare, citire, actualizare, ștergere) pe conturi Client, definit prin ID (UUID generat, nu poate fi modificat), nume, e-mail și parolă. Administratorii pot gestiona și dispozitivele înregistrate, care sunt indexate de Consumul de energie (mhec). Aceste dispozitive constau din ID, descriere, adresă (locație fizică).
Prin atașarea tabelului „UserReference" care conține ID-ul utilizatorului care deține dispozitivul. Administratorii pot mapa clienții pe dispozitive (fiecare client poate avea unul sau mai multe echipamente). Fiecare client are acces la o interfață prin care poate vizualiza echipamentele pe care le deține, consumul de energie În prezent, este, de asemenea, este posibil să adăugați dispozitive și să modificați dispozitivele existente.

ARHITECTURA APLICATIEI

Se utilizeaza o arhitectură server-client, împărțita în două componente principale: Backend și Frontend.

Backend:
Partea de backend este construită în Java, folosind framework-ul Spring Boot, pentru a facilita dezvoltarea rapidă și eficientă a aplicațiilor. Aplicația include două servicii REST care comunică între ele, fiecare având componente pentru gestionarea bazelor de date într-o arhitectură stratificată. Acestea oferă endpoint-uri pentru autentificare, operațiuni CRUD și validare a rolurilor.

Frontend:
Partea de frontend este dezvoltată în React, permițând interfețe de utilizator dinamice și modulare. Frontend-ul include un sistem de autentificare și redirecționează utilizatorii către paginile /admin sau /persondevice în funcție de rol. Administratorii au acces la funcționalități CRUD pentru utilizatori și dispozitive, putând asocia dispozitive pe baza ID-urilor între cele două baze de date.



CONEXIUNEALA BAZA DE DATE

Conexiunea la baza de date este configurată în fișierul application.properties, folosind următorii parametri:

- database.ip: IP-ul serverului care găzduiește baza de date.
- database.port: Portul de ascultare al serverului PostgreSQL (implicit 5432).
- database.user: Utilizatorul bazei de date (implicit postgres).
- database.password: Parola utilizatorului.
- database.name: Numele bazei de date. În acest proiect, persons-db este utilizat pentru microserviciul de management al utilizatorilor, iar devices-db pentru gestionarea dispozitivelor.

SINCRONIZARE

Pentru a simplifica sincronizarea datelor, ID-urile utilizatorilor din tabela `Person` din microserviciul de utilizatori au fost duplicate într-o nouă entitate numită `PersonReference` în microserviciul de management al dispozitivelor. Aceasta stochează toate ID-urile utilizatorilor.

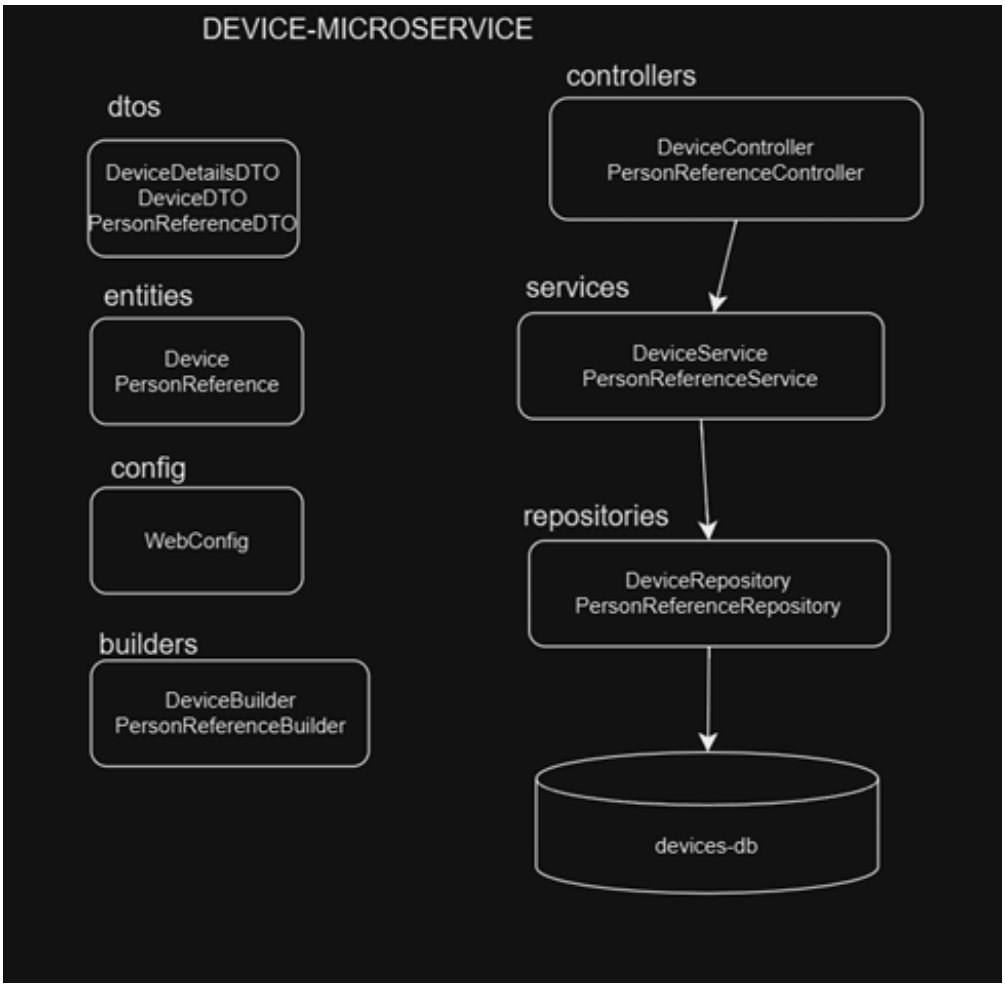
Când un administrator adaugă un utilizator nou prin frontend, o cerere POST este trimisă către `/person-references` din microserviciul de dispozitive pentru a înregistra ID-ul utilizatorului. Similar, la ștergerea unui utilizator, o cerere DELETE va elimina ID-ul din `PersonReference`.

Exista o relație Many-to-One între `Device` și `PersonReference`, permițând fiecărui utilizator să fie asociat cu mai multe dispozitive.

ARHITECTURA CONCEPTUALA

Aplicația este construită pe o arhitectură stratificată, incluzând layerele Controller, Service și Repository. Fiecare microserviciu — persons-microservice pentru utilizatori și devices-microservice pentru dispozitive — implementează această structură și dispune de propria bază de date.

- **Controller**: Gestionează cererile HTTP și definește endpoint-urile.
- **Service**: Conține logica de business și sincronizează datele între microservicii.
- **Repository**: Asigură accesul la baza de date pentru operațiuni CRUD.
  Această arhitectură modulară permite scalabilitate și independența fiecărui microserviciu.



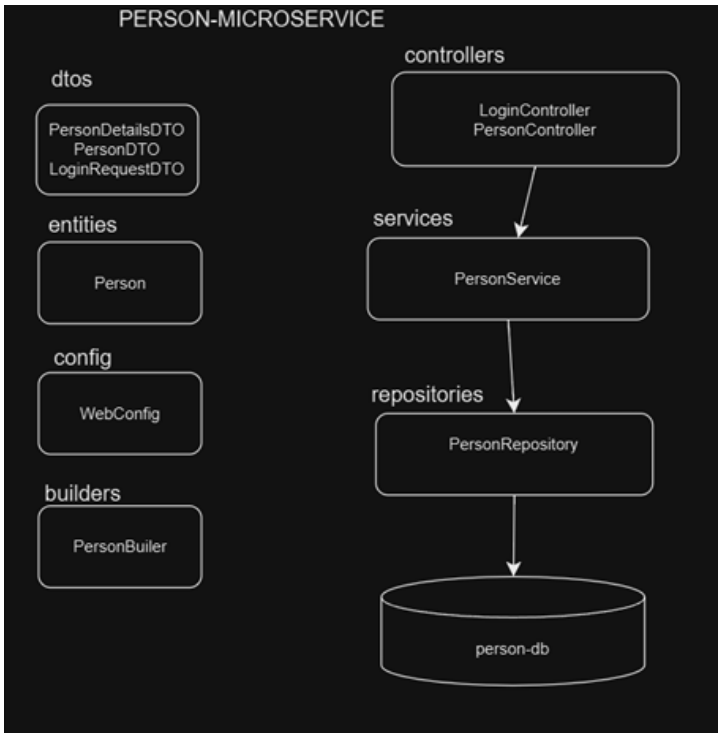


DIAGRAMA UML DEPLOYMENT

- **Frontend (React - sd-frontend)**: Rulează într-un container Docker separat și este accesibil utilizatorilor la adresa localhost:3000 prin browser.
- **Backend - Microservicii**:
  - **persons-microservice**: Rulează pe portul 8080, accesibil la localhost:8080.
  - **devices-microservice**: Rulează pe portul 8081, accesibil la localhost:8081.
- **Baza de Date (PostgreSQL)**: Rulează într-un container Docker dedicat, pe portul 5432, și conține două baze de date, persons-db și devices-db.