



# **IMMPlah**

**Documentation**

**Assignment 1**

**Nume :** Emanuel Golban

**Grupa:** 30245

## 1. REQUIREMENTS

The first module of the system consists of an online platform designed to manage patients, caregivers and medication. The system can be accessed by three types of users after a login process: doctor, patient and caregiver.

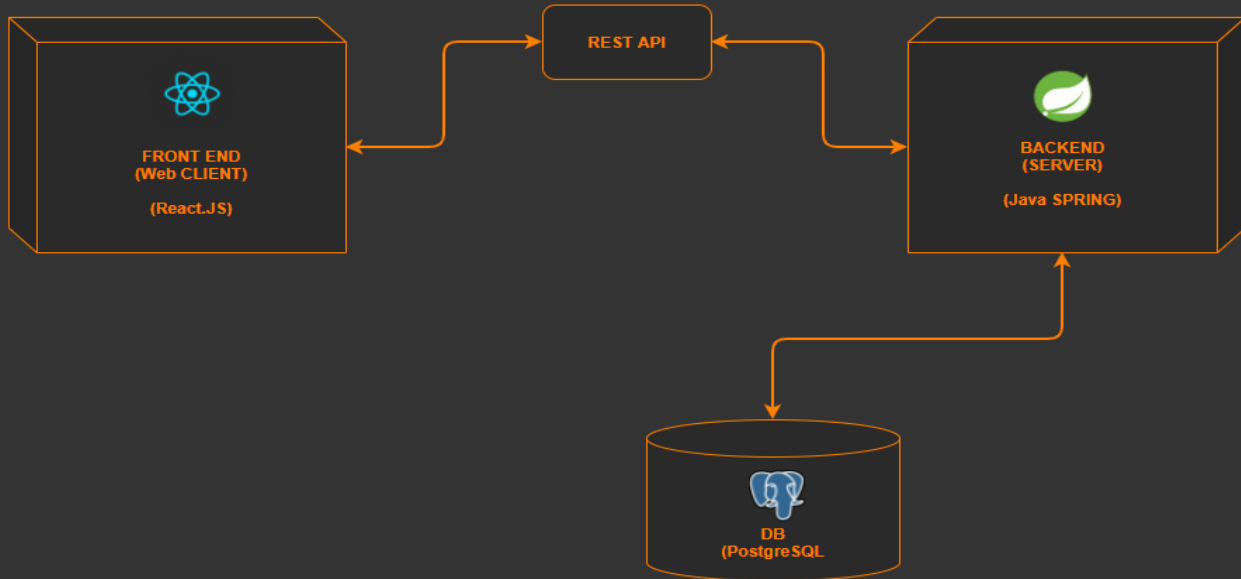
The doctor can perform CRUD operations on patient accounts (defined by ID, name, birth date, gender, address, medical record) caregiver accounts (defined by ID, name, birth date, gender, address, list of patients taken care of) and on the list of medication (defined by ID, name, list of side effects, dosage) available in the system.

The medical record of a patient must contain a description of the medical condition of the patient.

Furthermore, the doctor can create a medication plan for a patient, consisting of a list of medication and intake intervals needed to be taken daily, and the period of the treatment. The patients can view their accounts and their medication plans.

The caregivers can view their associated patients and the corresponding medication plans

## 2.SYSTEM ARCHITECTURE



## A. FRONT END

Aceasta este componenta cu care interacționează user-ul și a fost implementată folosind Node JS și React.JS ca librărie JavaScript.

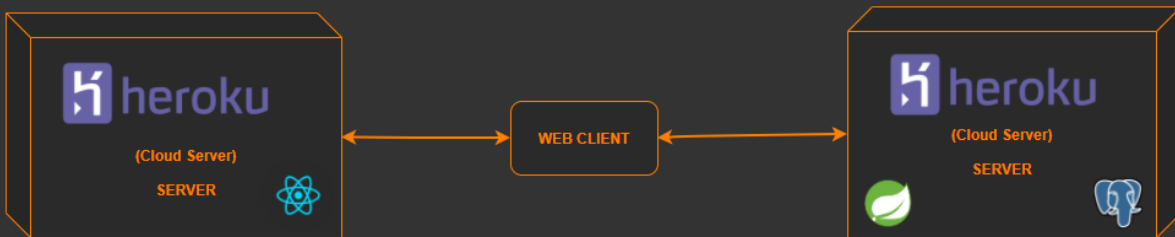
Componenta de Front end are rolul de a trimite request-uri către server (folosind un API) pentru a trimite/primi informații și a permite utilizatorului să acceseze și să modifice date de pe server.

## B. BACK END

Componenta de back-end include următoarele:

- Entități
  - Acestea formează model-ul aplicației și sunt folosite pentru a mapa relațional obiectele cu tabelele din baza de date folosind Hibernate.
- DTO-uri
  - Obiectele folosite pentru transferul de date
- Buildere
  - Acestea fac conversia din entități în DTO-uri și vice-versa.
- Repository-uri
  - Acestea sunt folosite pentru a accesa efectiv datele din baza de date.
- Controllere
  - Acestea gestionează request-urile primite de către server și apelează controllerele aferente fiecărui tip de REQUEST.
- Servicii
  - Acestea realizează operațiunile de CRUD și sunt accesate de către controllere.

### 3. DEPLOYMENT



Deployment-ul s-a realizat folosind Heroku Cloud, atat pentru partea de back-end cat si pentru front-end.

## 4.DATABASE

