# Aplicație pentru gestionarea unei rețele de clinici de donare de sânge

## Proiectare Software – Proiect de semestru

## Andrei-Denis Alexandru

## Grupa 30231

## Profesor îndrumător: Iulia Maria Cornea

## An universitar 2022-2023

1. Introducere

Proiectul consta într-o aplicație web care are ca scop principal definirea unei platforme digitale ale unei rețele de clinici de donare de sânge. Aceasta facilitează atât doctorilor, cat si posibililor donatori, accesul la programări pentru donații, confirmarea si monitorizarea programărilor sau vizualizarea noutăților din domeniul medical. De asemenea, administratorii platformei sunt capabili sa efectueze operații asupra datelor doctorilor.

Din punct de vedere logistic, aplicația își rulează local atât componenta de back-end cat si cea de front-end.

1. Diagrama de use-case

O imagine care conține diagramă, desen, schiță, cerc

Descriere generată automat

Cerințele tehnice ale proiectului identifica cei 4 actori care utilizează platforma.

Donatorul este singurul utilizator ce își poate crea un cont pe cont propriu, in timp ce pentru utilizarea aplicației atât donatorii, doctorii cat si administratorii trebuie sa se autentifice. Donatorul are posibilitatea de a vizualiza centrele de donare si a-si programa o sesiune de donare la centrul preferat, in funcție de locurile disponibile.

Doctorul aparține unui centru de donare, pe care acesta il administrează prin confirmarea si monitorizarea programărilor. Acesta este capabil si sa creeze si editeze (propriile) articole medicale sub forma unui blog.

Administratorul are responsabilitatea de a crea si gestiona conturile doctorilor.

Actorul de tip „Guest” nu are nevoie de un cont in platforma, iar singurele capabilități ale sale sunt de a viziona articolele postate de către doctori, si, opțional, sa le aprecieze.

1. Structura si tehnologiile folosite

Pentru dezvoltarea aplicației am avut la baza conceptul de Layered Architecture.

O imagine care conține text, captură de ecran, Font, număr

Descriere generată automat

Layer-ul Presentation(front-end-ul) conține întreaga interfață a utilizatorului, fiind cel mai „de sus” strat al arhitecturii. Implementarea acestuia a fost realizata cu ajutorul React.js si in special al bibliotecii de componente Material UI.

Back-end-ul aplicației a fost construit cu ajutorul framework-ului Java Spring si consta in 2 straturi: cel de Business Logic si cel de Data Access.

Layer-ul Bussiness Logic se ocupa de preluarea cererilor din partea utilizatorilor(front-end), procesarea acestora sub diferite forme cerute de către layer-ele de „mai jos” si trimiterea către acestea, cat si de returnarea rezultatelor înapoi la front-end. Printre componentele ce formează acest layer se numără REST Controller-ele, Service-urile, DTO-urile sau Mapper-ele.

Data Access Layer, cunoscut si ca Persistence Layer, este responsabil cu comunicarea cu baza de date. Acesta este reprezentat in cadrul proiectului de obiecte de tip Repository, care apelează query-uri pe baza de date.

Data Store Layer este reprezentat de însăși baza de date, care in cazul aplicației este o baza de date MySQL ce conține tabele generate cu ajutorul componentelor de tip Entity din cadrul framework-ului Spring. Acesta comunica doar cu layer-ul Persistence, executând query-uri.

1. Use-case-uri particulare
   1. Programarea unei donații

O imagine care conține text, captură de ecran, Paralel, diagramă

Descriere generată automat

După ce se autentifica, își poate vedea datele contului (inclusiv modul in care dorește sa primească notificări legate de programări), poate vedea o lista cu toate centrele de donare existente, lângă care se afla un buton pentru crearea unei programări la centrul respectiv, si o lista cu toate programările sale create.

Pentru a se programa la un centru, donatorul trebuie sa selecteze data pe care o dorește, iar pentru centrele care data respectiva este indisponibila, butonul de programare nu va putea fi apăsat. După ce programarea este realizata, donatorul își poate reîncarcă lista cu programări, unde poate vedea statusul fiecăreia.

* 1. Confirmarea unei programări

O imagine care conține text, captură de ecran, Paralel, diagramă

Descriere generată automat

Pe pagina principala a doctorului sunt afișate 2 liste cu programări: una cu programările pentru ziua curenta si una cu toate programările înregistrate la centrul de care aparține doctorul (lista afișata fiind un fragment din lista totala, obținută prin paginare). Pentru fiecare element din cele 2 liste este atașat un buton de confirmare a programării, care este indisponibil pentru programările deja validate.

* 1. Publicarea articolelor, vizualizarea si editarea acestora

O imagine care conține text, captură de ecran, Paralel, diagramă

Descriere generată automat

Doctorul poate de asemenea sa posteze articole medicale sau sa își vadă si actualizeze articolele publicate de-a lungul timpului.

Articolele tuturor doctorilor pot fi vizualizate de către oricine, selectând opțiunea „News and Blogs” din cadrul barei de navigare.

1. Concluzie

Prin dezvoltarea acestui proiect am reușit sa aduc o aplicație web de scara mica într-un stagiu funcțional, am învățat despre structurarea unei astfel de aplicație in front-end si back-end, sa dezvolt cele 2 componente cu ajutorul bibliotecii React, respectiv framework-ul Spring, completând cu folosirea de Design Patern.