

MINISTERUL EDUCAȚIEI ȘI CERCETĂRII ȘTIINȚIFICE



**UNIVERSITATEA
TEHNICĂ**
DIN CLUJ-NAPOCA

FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE
SPECIALIZAREA CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI

Proiectarea software

“Aplicație Web pentru donare de sânge”

Ciobănică Tadeuș
Grupa 30231
An academic 2022-2023

Cuprins

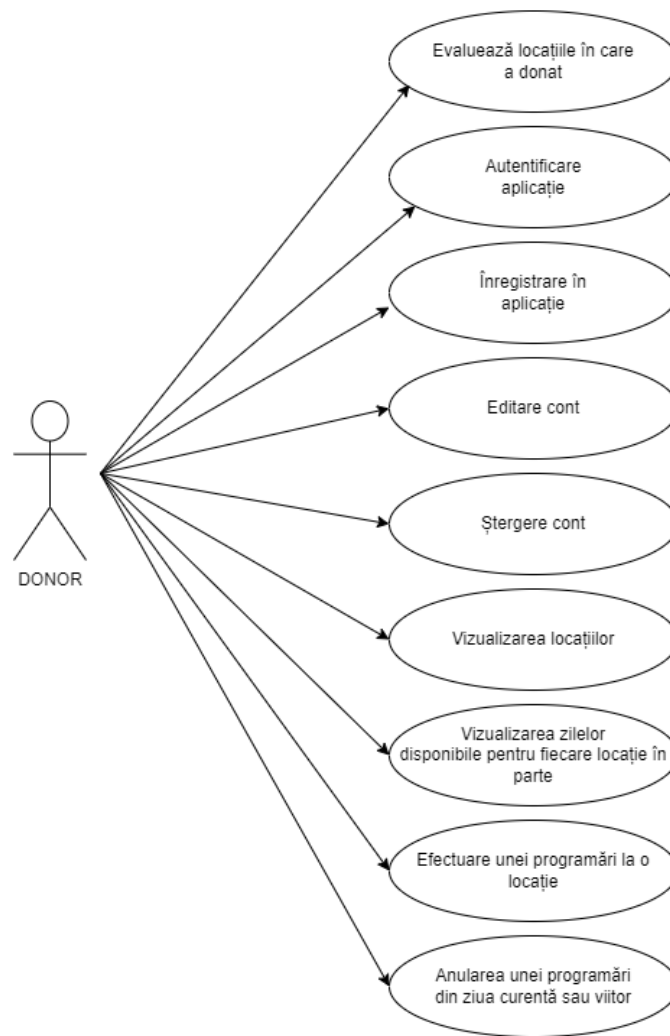
1. Introducere	3
2.Diagrama cazurilor de utilizare	3
3. Structură și tehnologii.....	5
4.Cazul de utilizare #1 – Crearea unei programări	6
5.Cazul de utilizare #2 – Operații CRUD pe Doctor.....	9
6.Cazul de utilizare #3 – Vizualizarea tuturor programarilor pentru o locație folosind paginare ca utilizator de tip Doctor.....	11
7.Cazul de utilizare #4 - Confirmarea unei programari.....	12
8.Cazul de utilizare #5 – Evaluarea unei locații.....	13
9.Concluzie	14

1. Introducere

Proiectul de față își propune să fie o Aplicație Web care să ajute în cadrul procesului de donare de sânge. Rolul principal al aplicației va fi de gestiune. Consider că aplicația va fi de mare ajutor, deoarece donarea de cele mai multe ori implică mult timp de stat la coadă, din lipsă de locuri și organizare.

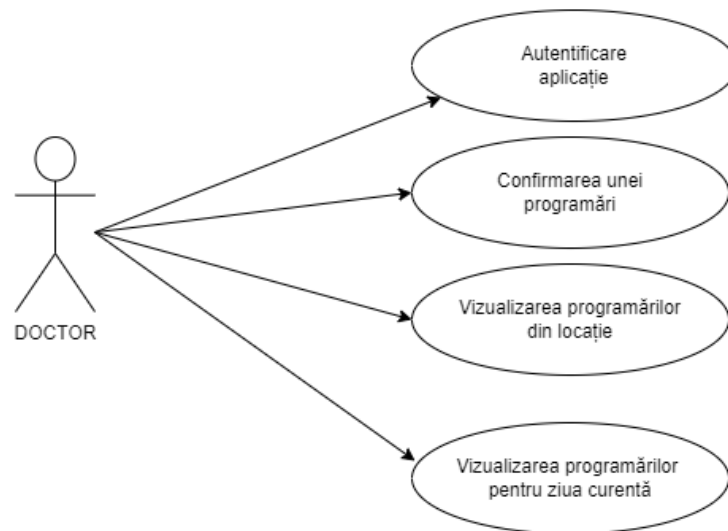
Aplicația va facilita gestionarea programărilor la diferite locații, donatorii putând să vizualizeze locațiile și datele disponibile în care își pot face programare. De asemenea, la fiecare locație există doctori care se ocupă de programări, iar în momentul în care donatorul se prezintă la donare, doctorul îi confirmă programarea.

2. Diagrama cazurilor de utilizare



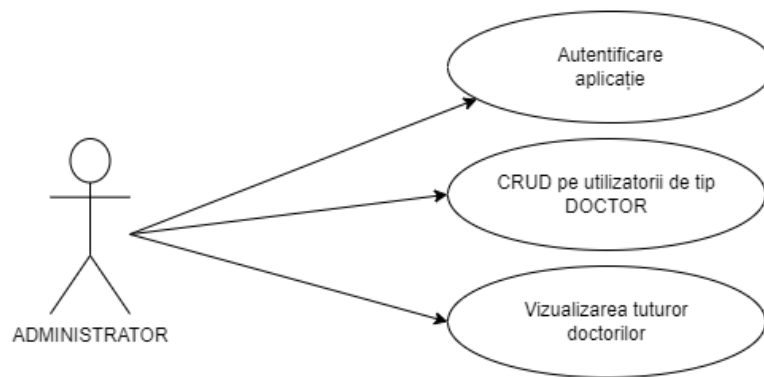
Cazuri de utilizare pentru actorul de tip Donator:

- Autentificare aplicație
- Înregistrarea în aplicație
- Editare cont
- Ștergere cont
- Vizualizarea locațiilor
- Vizualizarea zilelor disponibile pentru fiecare locație în parte
- Efectuarea unei programări la o locație
- Anularea unei programări din ziua curentă sau viitor
- Evaluează locațiile în care a donat
- Este notificat prin mail la crearea unei programări



Cazuri de utilizare pentru actorul de tip Doctor:

- Autentificare aplicație
- Poate confirma programarea unui donator după ce acesta a donat.
- Vizualizarea programărilor din locație
- Vizualizarea programărilor pentru ziua curentă



Cazurilor de utilizare pentru actorul de tip Administrator:

- Autentificare aplicație
- Realizarea de operații CRUD pe utilizatorii de tip Doctor
- Vizualizarea tuturor doctorilor

3. Structură și tehnologii

Am folosit o arhitectură de tip layered pentru realizarea proiectului, arhitectură ce presupune existența a mai multe „layere” sau straturi, începând cu cel de prezentare – interfața propriu-zisă, folosită de utilizator, și până la cel mai de jos nivel, cel de baze de date.

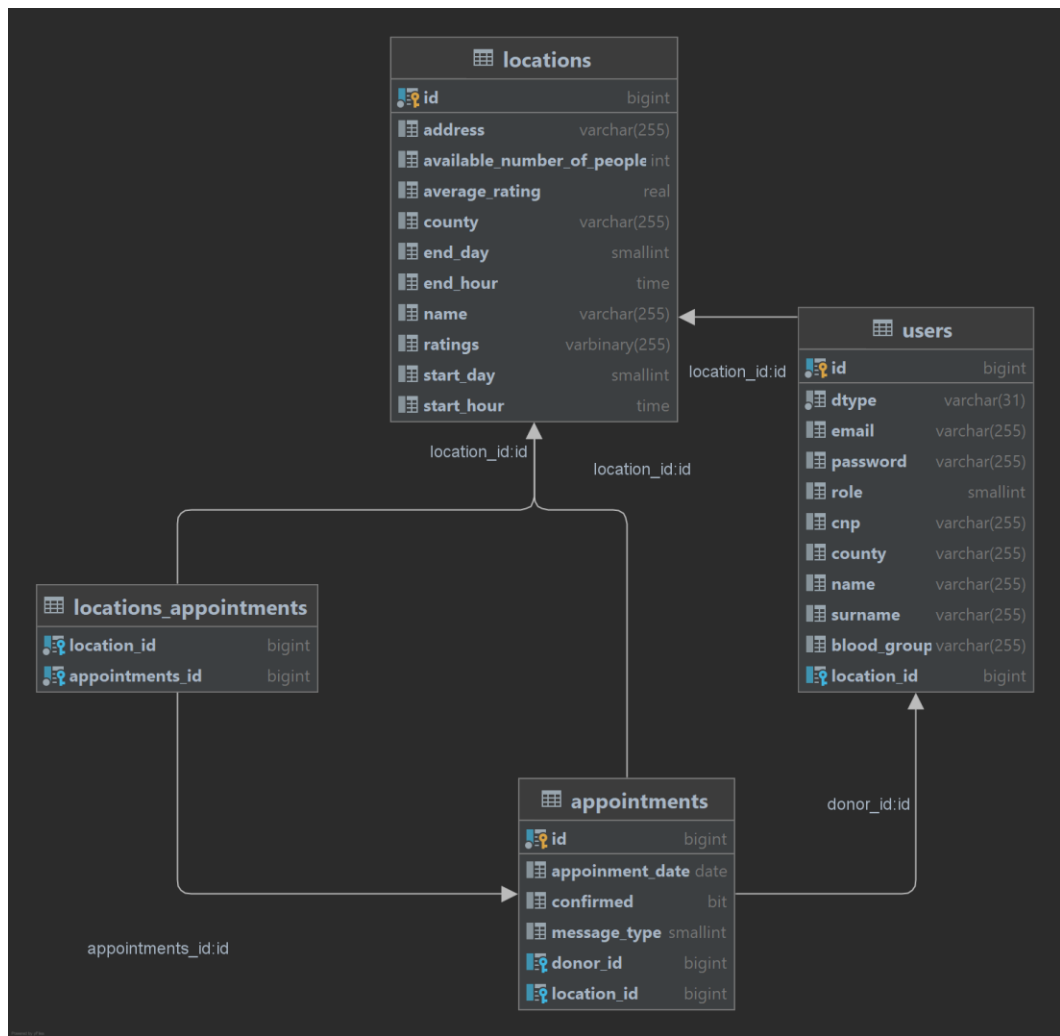
Începând cu cel mai înalt nivel, **layerul de prezentare**, acesta reprezintă interfața de utilizator, pe care o vor folosi cei 3 actori ai aplicației. Ea va pune la dispoziție paginile, butoanele și toate componentele necesare pentru utilizarea facilă a aplicației. Am folosit pentru layerul de prezentare JavaScript împreună cu biblioteca React.js, tot odata și alte framework-uri din biblioteca React cum ar fi Bootstrap și Materail-UI.

Urmează **layerul de control**, care reprezintă API-ul aplicației, și ne pune la dispoziție mai multe endpointuri folosite pentru procesare REST requesturilor primite din layerul de prezentare. Tipurile de requesturi folosite sunt PUT, POST, DELETE și GET.

Următorul este **layerul de servicii**, care conține metodele folosite în cel de control. Metodele din acest layer reprezintă logica aplicației, interogarea bazei de date respectiv prelucrarea datelor și trimiterea lor către layerul superior.

Repository layer este următorul nivel, și este folosit de cel superior, de servicii. Acesta este responsabil pentru accesul la baza de date, și se folosesc metode din JpaRepository.

Următorul și cel mai de jos layer este **layerul de baze de date**, care este responsabil pentru păstrarea datelor aplicației, și anume date despre utilizatori, locații, programări șamd. Acesta conține 4 tabele, Users – în care sunt salvați utilizatorii, Locations – în care sunt salvate locațiile pentru donare de sânge, Appointments – unde sunt salvate programările și o tabelă de legătură între Appointments și Locations.



4.Cazul de utilizare #1 – Crearea unei programări

În interfață, donatorul după logare are posibilitatea de creare a unei programări. Astfel da click pe “View all locations” și după pe “See locations” pentru a afișa toate locațiile disponibile din zona lui, precum și pentru fiecare locație se face un request pentru a afla dacă pentru fiecare locație utilizatorul a mai donat sau nu. Iar în cazul în care acesta a mai donat vom afișa în căsuța cu locația și un field pentru a putea da rating. Crearea unei programări se face folosind un date picker care este dezactivat pentru zilele de weekend și pentru zilele cu număr de locuri indisponibile.

Hello, Donor!

Edit account

Delete account

View my locations

View my appointments

Delete an appointment

Cancel

See locations

Cancel

See locations

Regina Maria

ID: 1
str.Louis Pasteur nr.41
MONDAY - FRIDAY
09:00:00 - 18:00:00
Rating : 8

MAKE AN APPOINTMENT

RATE

MedStar

ID: 3
str.Marasti nr.1
MONDAY - FRIDAY
09:00:00 - 18:00:00
Rating : 0

MAKE AN APPOINTMENT

Create new appointment

06/01/2023

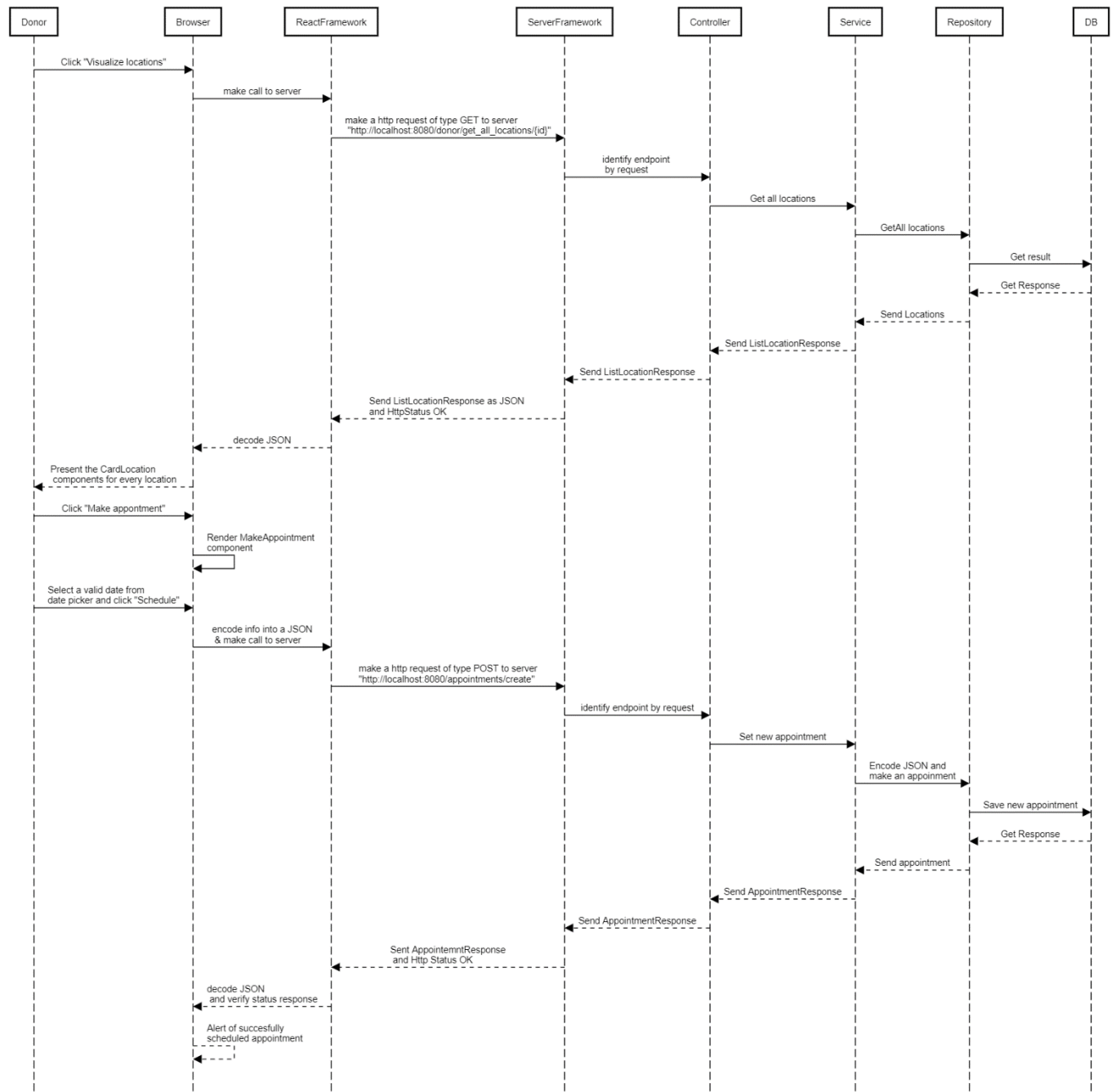
CANCEL

SCHEDULE

June 2023

Su	Mo	Tu	We	Th	Fr	Sa
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	1

Diagrama de secvență pentru realizarea unei programări:



5.Cazul de utilizare #2 – Operații CRUD pe Doctor

Dupa autentificarea propriu-zisă în aplicație administratorul are posibilitatea operațiilor CRUD (Create, Read, Update și Delete) pe utilizatorii de tip Doctor. Acestea sunt disponibile prin butoane cu titlu specific, iar pentru fiecare se va face render la cate o componentă de tip Modal în care se poate introduce date în field-uri. Aceste date sunt organizate sub forma de obiecte JSON și trimise prin HTTP requests către Server. În cazul unei editări se face întâi un request pentru a completa field-urile cu informațiile deja existente despre Doctor.

Hello, Admin!

Get all doctors

Create Doctor

Edit Doctor

Delete Doctor

ID:

Print information

Email:

Password:

Name:

Surname:

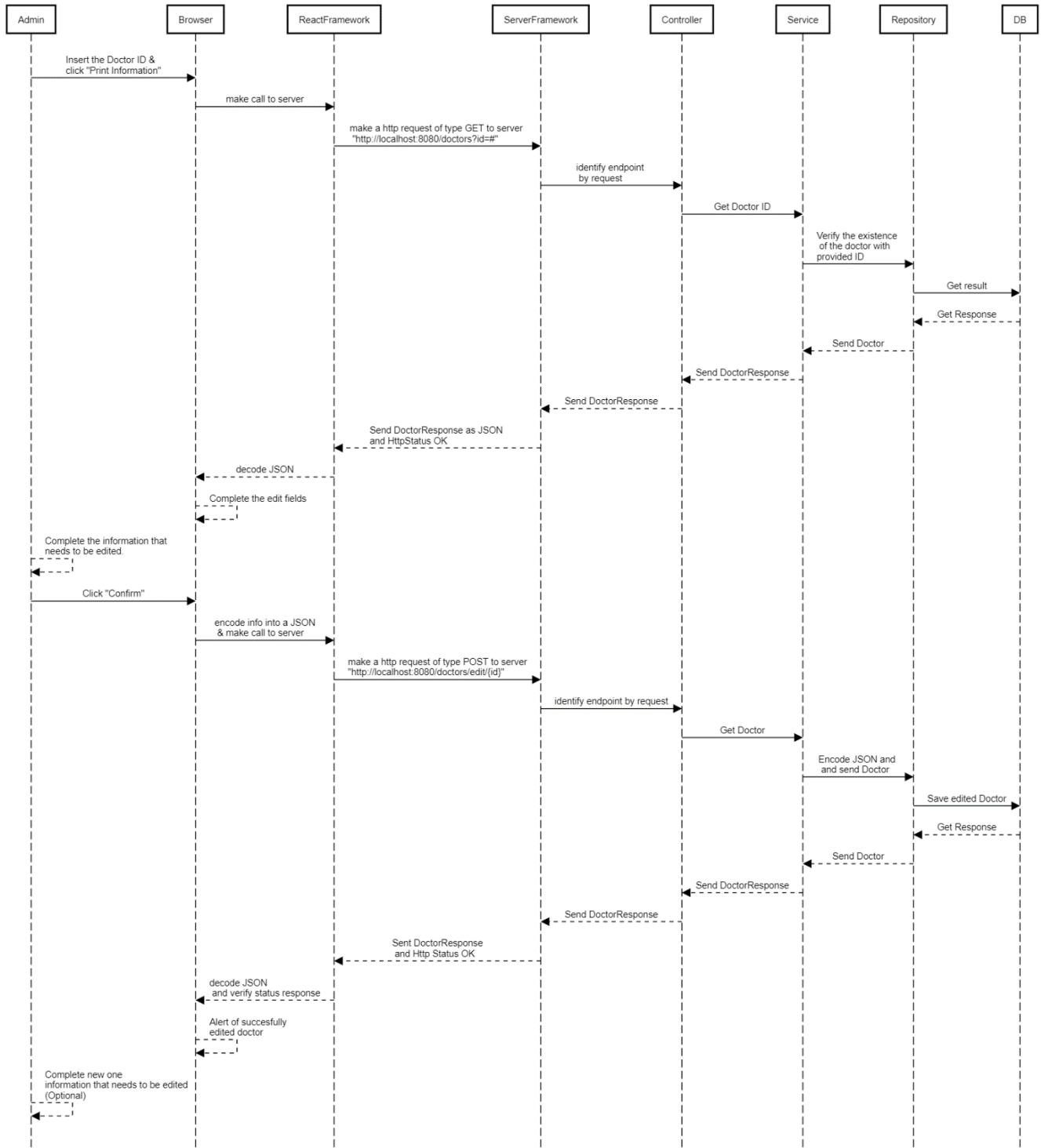
CNP:

County:

Cancel

Confirm

Diagrama de secvență pentru editarea informațiilor despre doctor:



6.Cazul de utilizare #3 – Vizualizarea tuturor programarilor pentru o locație folosind paginare ca utilizator de tip Doctor

În urma logării, un doctor poate vizualiza toate programările pentru locația la care este asignat acesta. Pentru aceasta va apăsa butonul “All appointments” după care se face render la o component de tip Modal cu un tabel care initial este gol, se apasă butonul “View appointments” pentru afisarea primei pagini. Fiecare pagina este de dimensiunea trei. Se pot folosi butoanele “Previous” “Next” pentru a interschimba între pagini.

Hello, DOCTOR!

Today appointments

All appointments

Confirm appointment

ID	Donor ID	NAME	Location ID	Appointment Date	Confirmed
1	5	Costea Gabriel	1	2023-06-01	Pending
2	5	Costea Gabriel	1	2023-06-01	Pending
3	5	Costea Gabriel	1	2023-06-01	Confirmed

Page 1

View appointments

Previous

Next

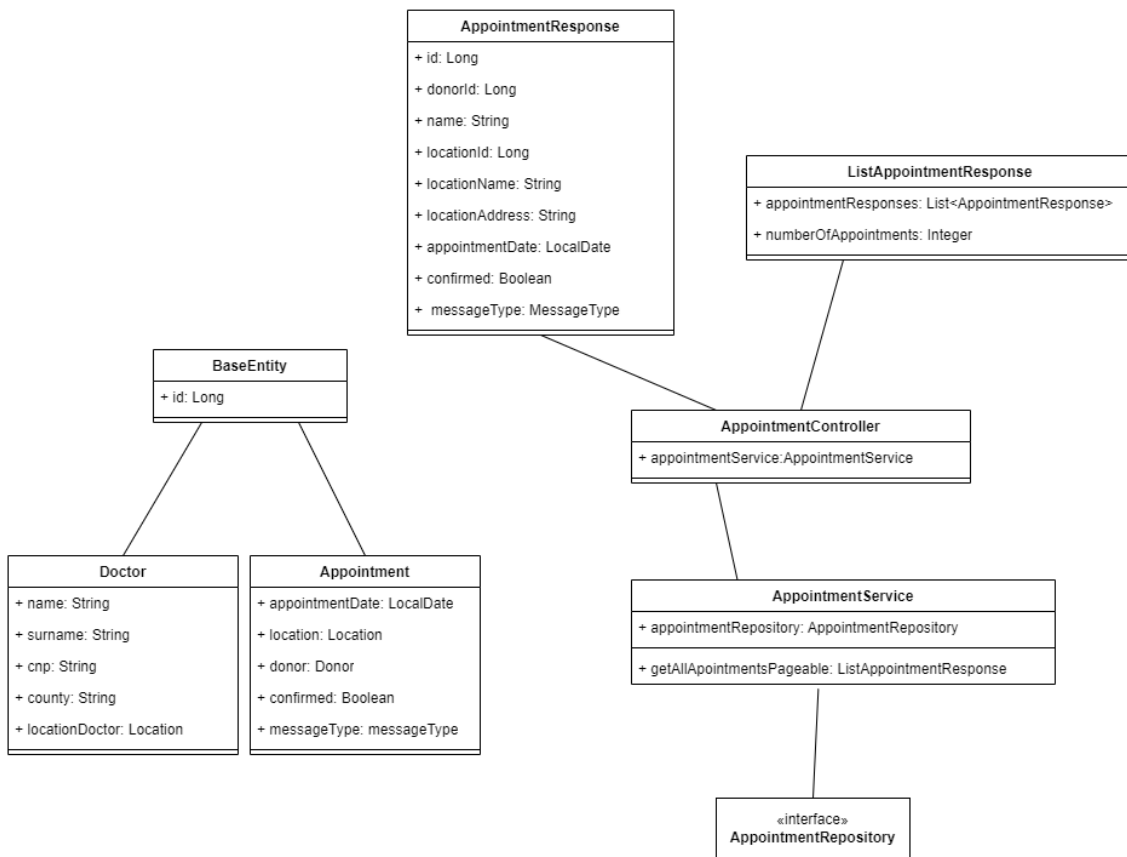
ID	Donor ID	NAME	Location ID	Appointment Date	Confirmed
4	5	Costea Gabriel	1	2023-06-01	Pending
5	6	Narita Diana Nicoleta	1	2023-06-01	Pending
6	5	Costea Gabriel	1	2023-06-07	Pending

Page 2

View appointments

Previous

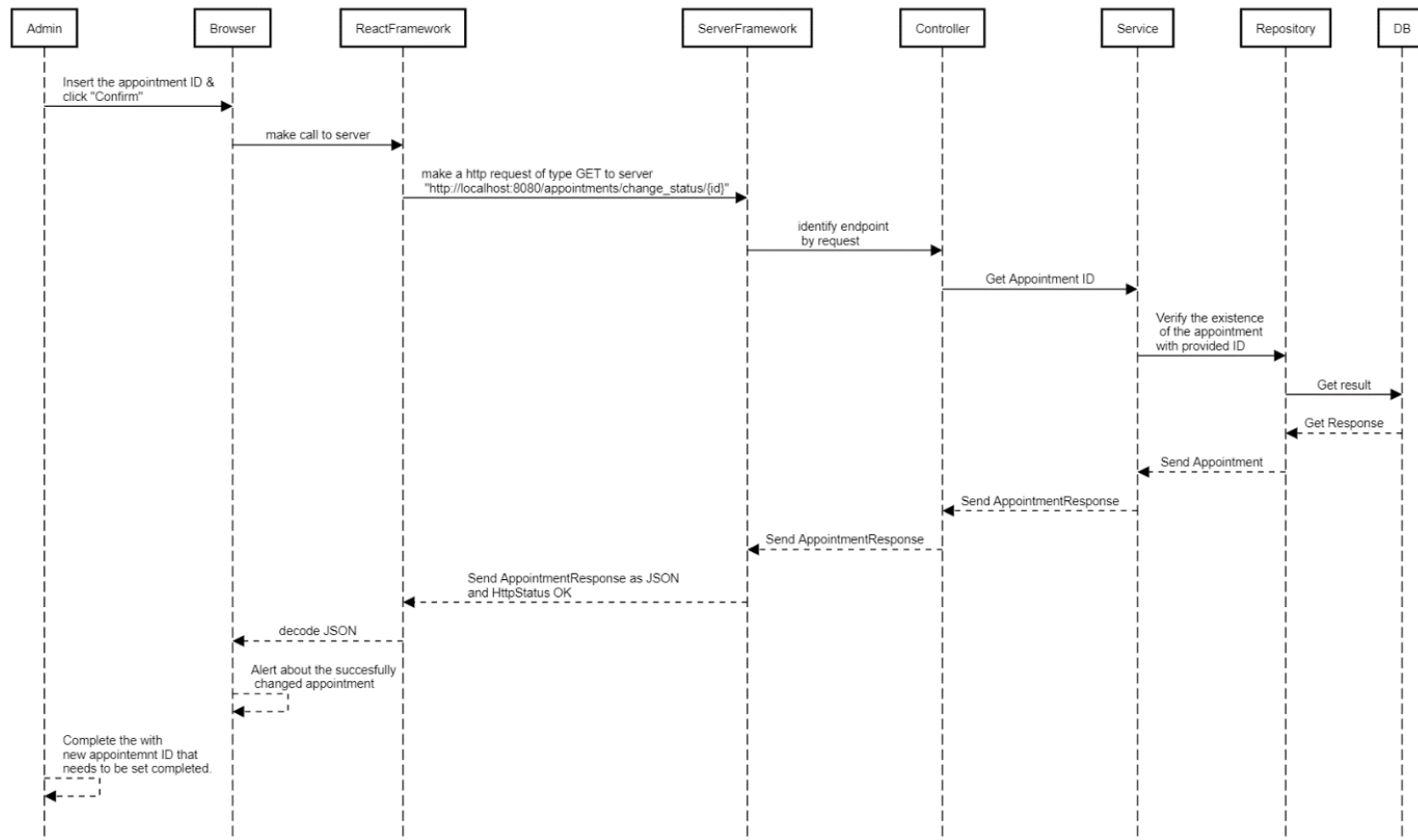
Next



7.Cazul de utilizare #4 - Confirmarea unei programari

Utilizatorul accesează pagina de logare, introduce datele de autentificare și apoi are loc logarea ca fiind Doctor în acest caz. Pe pagina principală se găsește un buton “Confirm appointment” care dacă este apasat va fi creată componenta respectivă din React. Conținutul acestei componente este un input field de tip text în care se introduce id-ul programării care se dorește a fi confirmată. La apăsarea butonului “Confirm” are loc schimbarea statusului pentru acea programare. Acum la secțiunea “All appointments” se poate vedea schimbarea statusului din Pending în Confirmed.

Diagrama de secvență pentru confirmarea unei programări:



8.Cazul de utilizare #5 – Evaluarea unei locații

Utilizatorul de tip Donator după cel puțin o donare la un centru de donare, are posibilitatea de adaugare/evaluare a experienței lui la respectiva locație. Se permite evaluarea doar celor care au donat pentru a putea aproxima cat mai bine rating-ul locației. Astfel în unele cazuri de rating mai mic se pot crea anumite cereri de feedback și a îmbunătăți centrul de donat și toate calitățile acestuia.

Cancel

See locations

Regina Maria

ID: 1
str.Louis Pasteur nr.41
MONDAY - FRIDAY
09:00:00 - 18:00:00
Rating : 8

MAKE AN APPOINTMENT

RATE

MedStar

ID: 3
str.Marasti nr.1
MONDAY - FRIDAY
09:00:00 - 18:00:00
Rating : 0

MAKE AN APPOINTMENT

9.Concluzie

Cred că o aplicație de genul celei de față ar fi foarte utilă în viața reală, deoarece rezolvă într-adevăr o problemă reală. La rândul meu am petrecut mult timp stând la coadă pentru donare, și un număr limitat de programări per zi per locație ar reduce reticența donatorilor, provocată și de timpul mare de așteptare.

Prin realizarea acestui proiect am folosit pentru prima dată Spring Framework, alături de Java, pentru a construi backendul. Am folosit Postman pentru a testa requesturile de tip REST, iar ulterior am folosit biblioteca React.js pentru a realiza frontendul.