



**UNIVERSITATEA DE VEST DIN TIMIȘOARA
FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ
PROGRAMUL DE STUDII DE LICENȚĂ : Informatică
Aplicată**

LUCRARE DE LICENȚĂ

COORDONATORI:

Prof. Dr. Daniela Zaharie
Dr. Felix Mic

ABSOLVENT:

Dosan Andrei Bogdan

TIMIȘOARA

2022

UNIVERSITATEA DE VEST DIN TIMIȘOARA
FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ
PROGRAMUL DE STUDII DE LICENȚĂ : Informatică
Aplicată

LUCRARE DE LICENȚĂ

COORDONATORI:

Prof. Dr. Daniela Zaharie
Dr. Felix Mic

ABSOLVENT:

Dosan Andrei Bogdan

TIMIȘOARA

2022

Abstract

The idea behind this bachelor's thesis was the necessity of sharing the medical history. This is important both from a patient perspective and from a medical staff perspective.

Nowadays, after an interaction with the doctor, the patient receives a physical document, called a medical letter, which contains a series of data relevant to the type of the medical event: diagnosis, prescription, recommendations, etc. The patient has the duty to keep all the documents in a personal archive, an archive he needs for the following interventions.

Using documents in physical format can cause difficulties for both the patient and the doctor. The physical documents are likely to be damaged or lost, then the patient's file becomes incomplete.

On the other hand, the data that the patient brings to a doctor's appointment could get seriously big. The medical staff has to analyze the information in a short period of time. The problems that come with this way of sharing are: data is often not organized, data can't be filtered (for example, by the type of it, by date, by doctor's name), data lacks visual representation.

Cuprins

Rezumat	5
1 Introducere	6
1.1 Descrierea problemei	6
1.2 Obiective	6
1.3 Abordare propusă	7
1.4 Structura lucrării	8
2 Abordări existente	10
10section.2.1	
11section.2.2	
13section.2.3	
3 Analiza cerințelor	15
3.1 Perspectiva pacientului	15
3.2 Perspectiva medicului	15
4 Aplicatia MyMedicalRecord	17
4.1 Arhitectura aplicației	17
4.1.1 Arhitectura componentei front-end	18
4.1.2 Arhitectura componentei back-end	21
4.1.3 Arhitectura bazei de date	22
4.2 Tehnologii folosite	24
4.2.1 React.js	24
4.2.2 Redux	24
4.2.3 Material UI	26
4.2.4 Recharts	26
4.2.5 React-router-dom	26
4.2.6 Firebase	27
4.3 Detalii de implementare	28
4.4 Exemple de utilizare	28
4.4.1 Diagrama cazurilor de utilizare	29
4.4.2 Automatul de stări	30
5 Concluzii și direcții viitoare de dezvoltare	32
5.1 Rezultate obținute	33
5.2 Dezvoltări ulterioare	39
Bibliografie	42

Rezumat

Ideea de la care a pornit această lucrare este necesitatea de a partaja istoricului medical între pacient și medic. În prezent, partajarea istoricului se realizează în format fizic, adică, după orice tip de eveniment medical, pacientul primește documente pe care are datoria să le păstreze și să le prezinte la următoarele consultații medicale. Această metodă poate genera dificultăți, atât pentru pacient, cât și pentru medic. Documentele fizice pot fi ușor deteriorate sau pierdute, astfel dosarul pacientului devine incomplet.

Din perspectiva medicului, analiza informației prezentată este dificilă, deoarece este, de cele mai multe ori, neordonată și nu poate fi filtrată.

Lucrarea are ca scop dezvoltarea unei aplicații web dedicată exclusiv pacienților și medicilor. Aplicația își propune să fie un mediu sigur, intuitiv și ușor de folosit, în cadrul căreia un pacient își creează un istoric medical digital. Platforma reprezintă, în același timp, o arhivă personală și un canal de partajare a datelor medicale, cu medicul.

Medicul își creează lista de pacienți, după care poate vizualiza date personale ale acestora, statistici și istoricul medical.

Capitolul 1

Introducere

1.1 Descrierea problemei

Odată cu dezvoltarea tehnologiei, distribuirea și stocarea informației a devenit mai eficientă și mai accesibilă. Metodele învechite de arhivare sunt înlocuite de practici îmbunătățite care impun un nou nivel de ușurință și comoditate. Oamenii încep să se bazeze pe tehnologie când vine vorba de instrumente care ușurează și îmbunătățesc aspectele din viața cotidiană. Din ce în ce mai multe domenii de activitate decid să îmbine mediul online cu cel fizic, astfel având parte de beneficiile ambelor părți. Învățământul își mută activitatea în mediul online, companiile, în special cele de tehnologie, normalizează telemunca, iar afacerile mici și medii nu mai au nevoie de un sediu propriu-zis, pot funcționa doar în online.

Din păcate, în domeniul medical, formatul cel mai des întâlnit este încă cel fizic. Documentele sunt emise, stocate și partajate în această manieră. Acest lucru poate genera dificultăți pentru fiecare parte din acest sistem, dar cei mai afectați sunt beneficiarii direcți, adică medicii și pacienții.

În cazul pacienților, disconfortul apare în momentul în care au de a face cu stocarea și partajarea documentelor medicale, deoarece sunt ușor deteriorabile și pot fi pierdute. După o consultație, o intervenție sau o internare în spital, pacienții primesc o scrisoare medicală care conține informații relevante, de exemplu: data, diagnostic, prescripție, recomandări, detalii etc. În continuare, ei sunt responsabili de stocarea acestor documente și păstrarea lor într-o condiție cât mai bună pentru a le prezenta la următoarele consultații medicale.

Din punctul de vedere al medicilor, problema apare în analizarea și filtrarea informației. La o consultație, medicului i se prezintă o arhivă de documente, din care acesta are nevoie să utilizeze doar unul sau două documente specifice. Acest proces necesită timp și devine ușor repetitiv.

1.2 Obiective

Lucrarea are ca scop dezvoltarea aplicației web, care poartă numele de `MyMedicalRecord`. Aceasta își propune să fie un mediu sigur pentru prelucrarea și stocarea datelor, intuitiv și ușor de folosit, în cadrul căreia pacienții și medicii pot partaja informații din dosarele medicale printr-o manieră sigură. Prelucrarea datelor necesită acordul utilizatorului și nu există opțiunea de a fi partajate în afara aplicației, tocmai pentru

a păstra integritatea datelor și confidențialitatea lor.

Sistemul prezintă o interfață utilizator prietenoasă și minimalistă. Astfel procesul de navigare și folosire a aplicației în scopul dorit este intuitiv. Animațiile și tranzițiile lipsesc de pe platformă pentru a păstra nivelul minim de resurse cât mai jos posibil. Prin urmare, utilizatorul se poate folosi de platformă folosind orice aparatură.

Aplicația furnizează, pentru pacient, un loc de stocare al datelor în cloud. Istoricul medical este completat de către pacient și reprezintă totalitatea evenimentelor medicale.

Partajarea istoricului între medic și pacient se desfășoară, în cadrul aplicației, cu acordul pacientului. Așadar acesta decide cu ce medici dorește să partajeze datele și are dreptul să anuleze dreptul de acces al medicului. Prin această procedură este solidificată confidențialitatea datelor.

Pentru medici beneficiile aplicației sunt: organizarea pacienților, organizarea datelor și filtrarea acestora. Medicii își aleg pacienții care fac parte din lista personală și pot vizualiza date cu privire la aceștia. Informația este organizată în coloane ce pot fi ordonate alfabetic (crescător sau descrescător) și sunt filtrabile după câmpuri precum nume, data sau valoare.

Nu există constrângeri asupra dimensiunii ecranului a device-ului folosit, browser-ului sau a locației folosite pentru accesarea platformei. Aceasta oferă aceleași beneficii pentru laptop, telefon, computer sau browser (Google Chrome sau Microsoft Edge), oriunde în lume. Componenta vizuală a aplicației este responsive, ceea ce înseamnă că poate fi accesată de pe orice mărime a ecranului.

1.3 Abordare propusă

Aplicația este divizată în două mari părți, perspectiva medicului și perspectiva pacientului. Rolul contului este selectat la înregistrarea pe platformă, astfel fiecare utilizator dispune de conținutul și serviciile adecvate fiecărui tip de cont.

Perspectiva pacientului conține două pagini, pagina de profil și pagina de acces. În cadrul paginii de profil se regăsesc datele personale ale pacientului și o secțiune dedicată pentru istoricul medical. Informațiile asociate fiecărui pacient ajung de la numele complet, date de contact (e-mail, număr de telefon), vârsta până la fotografie de profil, attribute personale (înălțime, greutate, gen) și listă de comorbidități. O parte mică din aceste date sunt completate la înregistrare, adică e-mail, numele complet și data de naștere, iar celelalte informații deservesc în mare parte pentru analiza medicului.

Componenta adresată istoricului medical este structurată dintr-o parte cu câmpuri de filtrare a istoricului medical în funcție de dată, tipul evenimentului și numele medicului, și o secțiune în care pacientul adaugă întâmplările medicale. Informațiile care trebuie completate pentru a adăuga un eveniment medical sunt: tipul evenimentului (vizită la medicul de familie, vizită la medicul specialist, internare în spital), diagnostic, numele medicului care a emis scrisoarea medicală, data sau perioada, prescripția și detalii. Istoricul medical reprezintă totalitatea evenimentelor care au fost încărcate de utilizator pe platformă.

Prin intermediul paginii de acces aferentă pacientului, se gestionează cererile de acces trimise de către medici. Aceste cereri pot fi acceptate sau respinse, astfel, dacă pacientul acceptă solicitarea, el va permite medicului să îi vizualizeze datele personale, datele de contact și istoricul medical. În cazul în care cererea este respinsă, medicul

nu poate vizualiza informațiile pacientului respectiv solicitarea va fi ștearsă din listă. Decizia de a accepta cererea nu este definitivă, pacientul având posibilitatea ca oricând dorește să restricționeze privilegiul medicului.

Perspectiva medicului prezintă trei pagini: pagina de profil, pagina de acces și pagina care conține lista de pacienți. Pagina de profil este asemănătoare cu cea a pacientului, diferențele fiind lipsa secțiunii dedicată istoricului medical și câmpurile pe care contul de medic le deține. Astfel sunt eliminate secțiunile cu attribute personale precum ziua de naștere și lista de comorbidități, fiind adăugate detalii legate de experiența profesională, specializări, funcția și instituția medicală, educație, orarul și adresa instituției; date care l-ar putea interesa pe pacient.

Pagina de acces găzduiește o listă cu toți pacienții înregistrați pe platformă, la care medicul le poate trimite o cerere de acces. Dacă solicitarea este acceptată, pacientul este adăugat în lista personală a medicului și este șters din lista paginii de acces. Dacă cererea este respinsă atunci medicul nu va avea acces la datele pacientului. Odată trimisă revendicarea de acces, acesta fiind în așteptare, ea poate fi anulată oricând de către medic.

Ultima pagină, cuprinde lista pacienților și statistici legate de attributele lor. Fiecare rând din listă reprezintă datele unui pacient care a acceptat cererea de acces. De aici medicul poate vizualiza istoricul medical, poate aplica filtre și, dacă acesta dorește, poate șterge pacientul din propria listă.

Secțiunea de statistici înglobează statistici privind attribute, de exemplu: repartizarea comorbidităților, categoria de greutate, ponderea genurilor și categoriile de vârstă. Datele de intrare pentru grafice sunt datele pacienților din lista de pacienți.

Pentru ca fiecare cont să își dovedească identitatea, la înregistrare îi este atribuit un identificator unic. Astfel, pacienții vor putea recunoaște medicul atunci când primesc cerere și vice-versa, medicii știu către cine să trimită cererea.

Am decis ca aplicația să fie implementată preponderent cu ajutorul ReactJS și Firebase, datorită beneficiilor pe care acestea le au: performanță, securitate, robustețe și multe altele.

1.4 Structura lucrării

Primul capitol constituie introducerea în lucrarea de licență. Acest capitol descrie tema pe baza căreia este structurată lucrarea, prezintă o variantă de rezolvare a acesteia și descrie abordarea propusă pentru a combate problema.

Scopul celui de al doilea capitol este de analiză a sistemelor similare deja existente. În cadrul acestui capitol sunt enumerate și descrise trei dintre cele mai de succes sisteme EMR (Electronic Medical Record), la care se pune accent pe beneficiile și diferențele dintre ele. Beneficiile sunt preluate și implementate în MyMedicalRecord.

Cel de al treilea capitol evidențiază serviciile oferite de aplicație și utilitatea acestora, atât pentru medici cât și pentru pacienți. Cerințele aplicației sunt descompuse în două părți: cerințe pentru conturi de tip medic și cerințe pentru conturi de tip pacient. De aceea analiza acestora este efectuată individual, pentru fiecare dintre cele două perspective.

Capitolul patru, cel mai amplu, trece, pe rând, prin aspectele tehnice ale implementării aplicației. Descrie arhitectura componentelor (back-end, front-end), arhitectura sistemului final, tehnologiile folosite, detaliile de implementare și oferă exemple

de utilizare ale aplicației. În dreptul tuturor tehnologiilor folosite se regăsește o argumentare în alegerea făcută.

Ultimul capitol are rol de concluzionare. Aici sunt împărtășite rezultatele obținute, se pune în balanță viziunea inițială a aplicației cu aplicația finală, sunt prezentate vulnerabilitățile și punctele forte ale sistemului și sunt sugerate viitoare extinderi care ar putea îmbunătăți aplicația.

Capitolul 2

Abordări existente

În urma căutărilor și analizelor efectuate în legătură cu abordările similare existente, am descoperit numeroase sisteme asemănătoare din punct de vedere arhitectural și din punct de vedere al beneficiilor. Am decis să aleg trei dintre aceste sisteme; selecția a fost făcută în funcție de serviciile pe care fiecare aplicație le oferă, atât pentru medic, cât și pentru pacient. Abordările extrase fac parte din categoria sistemelor EMR (Electronic Medical Record).

Sunt sisteme digitalizate care includ diagnostice, alergii, istorice medicale, date de imunizare, rezultate de la laborator, medicamente și note ale medicilor. Sistemele EMR se pot ocupa de orice, de la documentarea datelor pacientului și programarea întâlnirilor, până la completarea prescripțiilor și verificarea asigurărilor. Platformele de acest tip sporesc productivitatea și eficiența, și fac mai ușoară comunicarea între cele două părți.

2.1 AdvancedMD ¹

AdvancedMD este conceput pentru organizațiile mici și mijlocii și se ocupă de: programarea pacienților, gestionarea sarcinilor, prescrierea electronică și oferă funcția de mesagerie între pacienți și medici. Designul său compact include facturare, dosare online ale pacienților și multe alte servicii. Planurile personalizate de sănătate îi ajută pe medici să gestioneze condițiile specifice ale pacientului pe baza unor factori precum vârsta, genul, rezultatele de la laborator și multe altele.

Platforma permite pacienților să programeze întâlniri față în față, precum și la telemedicină. Medicii pot face capturi de ecran în timpul apelurilor conferință și pot salva în siguranță imagini și alte informații sensibile în dosarul unui pacient. Solicitățile de prescripție medicală pot fi revizuite și semnate electronic de către medici, indiferent de locația lor fizică.

¹<https://www.advancedmd.com/>



Figura 2.1: Structura generală AdvancedMD

Figura 2.1 reprezintă pagina principală a aplicației, redată pentru contul de medic. Din imagine se pot distinge principalele funcționalități ale sistemului. În partea din stânga sunt programările medicilor din ziua curentă, iar în partea din dreapta se află lista de pacienți cu datele personale și datele de contact, în mijloc sunt butoanele care declanșează acțiuni sugestive, de exemplu: Notes (notițe), Rx (listă de medicamente pe care medicul le poate prescrie pacienților), Messages (funcționalitate de mesagerie între medic și pacient) etc.

AdvancedMD oferă următoarele facilități:

Tabloul de bord pentru activități: Gestionează și prioritizează sarcinile cu liste, analize, alerte și acces bazat pe roluri. Oferă acces la sarcinile zilnice.

Rapoarte: Captează automat informații precum vizitele medicale. Permite utilizatorilor să trimită rapoartele în format electronic către registrele de sănătate.

Rețetă electronică: Aplicația mobilă permite semnarea și vizualizarea electronică a rețetelor. Utilizează simboluri și fraze de acces securizate pentru autentificarea cu doi factori.

Tabloul de bord pentru medic: Medicii pot programa întâlniri și vizualiza mesajele pacientului și a personalului, rezultatele testelor, scrisorile de recomandare și listele de sarcini.

2.2 athenaOne ²

athenaOne permite gestionarea înregistrărilor medicale electronice, coordonarea îngrijirilor, vizualizarea sănătății pacienților și facturarea medicală.

²<https://www.athenahealth.com/solutions/athenaone>

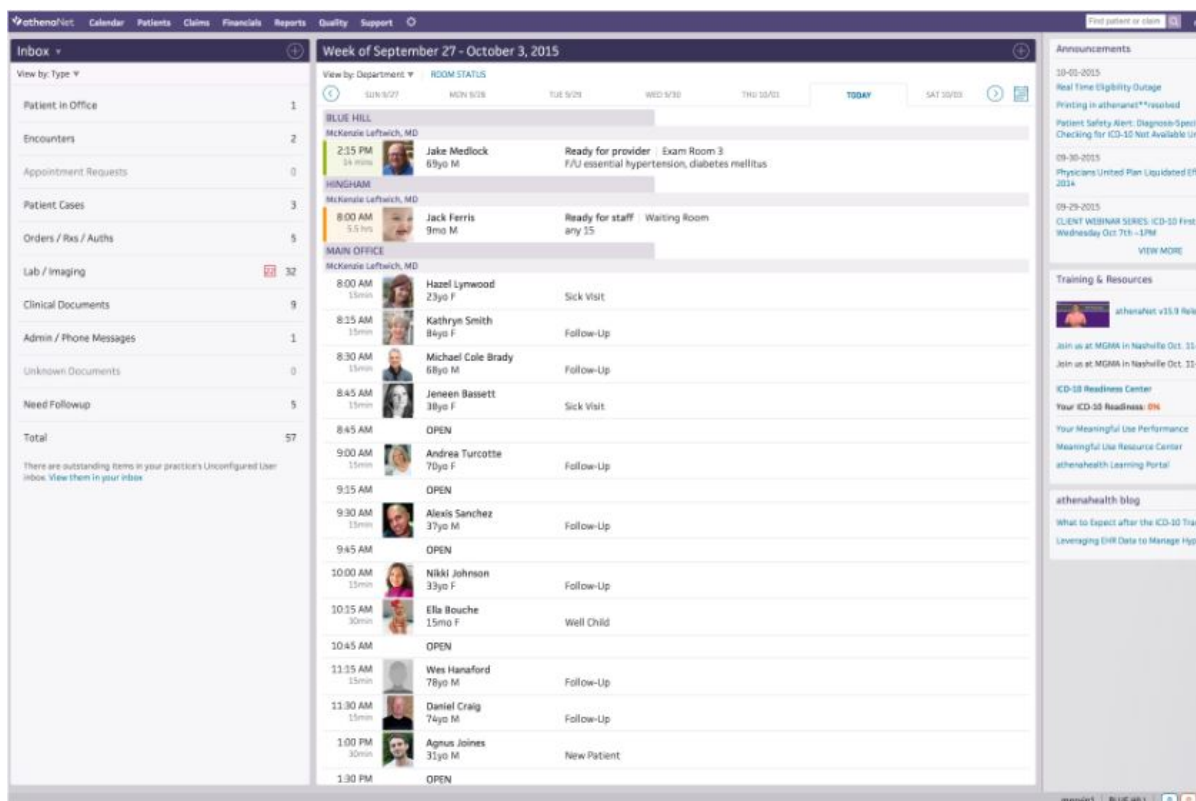


Figura 2.2: Platforma athenaOne

Este potrivit pentru organizațiile de îngrijire a sănătății de toate dimensiunile. Aplicația permite organizarea de vizite la pacienți, îngrijirea pacientului, operațiuni clinice și documentație, susținând în același timp facturarea și interoperabilitatea.

Figura 2.2 redă pagina principală a medicului, în care se află agenda din ziua curentă, un meniu din care se navighează spre pagina dorită și o secțiune de știri.

Printre facilitățile aplicației se numără:

Gestionarea listei de medicamente: permite medicilor să urmărească toate medicamentele prescrise unui pacient, permițându-le să înregistreze medicamente, atât din sistem, cât și din alte sisteme sau prescrise de alți medici.

Portalul Pacienților: Permite pacienților să programeze întâlniri, pe care medicii trebuie să le accepte și permite obținerea rezultatelor de la analizele efectuate de aceștia în format exclusiv digital.

Rezumate clinice: Oferă cabinetelor opțiunea de a prezenta rezumate clinice prin portalul pacientului sau în format fizic. Rezumatele întâlnirilor sunt pregătite după vizitele pacientului și postate pe portal sau trimise direct pacientului.

Înregistrări demografice: Păstrează înregistrări ale datelor demografice, cum ar fi: tipul de asigurare, genul, limba vorbită, etnia și rasa. Permite medicilor să contacteze anumiți pacienți pentru informare, folosind atributele de mai sus ca filtre sau criterii de excludere sau de includere.

2.3 Cure MD ³

The screenshot displays the CureMD web application interface. At the top, there's a header with the CureMD logo and a user profile for Julia J. Anthony. Below this, a navigation menu on the left lists various functions like Messaging, Registration, and Billing. The main content area is titled 'Evaluation and Management (E&M)' and contains a form for entering patient data. The form includes fields for Type, Primary Dx, and a table for Selected Codes (Procedures, Modifiers, Diagnoses, Unit, Charge, Amount). The table shows three rows of data: 88490 (100 insulin syringes), 74176 (Cl angle abdom w/o), and 99211 (Office/outpatient visit, est 5 mins). Below the table, there are sections for 'Add Procedures' and 'Add Diagnosis', each with a list of codes and descriptions. At the bottom, there's a 'Comments' field and a 'Show on Charges' checkbox.

Procedures	Modifiers	Diagnoses	Unit	Charge	Amount
88490 100 insulin syringes			1	PLJ	0.00
74176 Cl angle abdom w/o			1	PLJ	0.00
99211 Office/outpatient visit, est 5 mins			1	PLJ	25.00
				PLJ	0.00

Figura 2.3: cureMD

CureMD este un EMR (Electronic Medical Record) potrivit pentru sistemele de monitorizare a sănătății de toate dimensiunile. Oferă operațiuni clinice și administrative pentru specialități precum: ginecologie, oncologie, dermatologie, obstetrică, sănătate publică și multe altele. Caracteristicile sale importante includ automatizarea fluxului de lucru, prescrierea electronică, interoperabilitatea și laboratoarele electronice.

Pe pagina principală (figura 2.3) a aplicației se regăsesc alertele, tendințele și istoricul medical al pacienților.

Facilități:

³<https://www.curemd.com/>

Laboratoare electronice: Există posibilitatea de primire a rezultatelor de la analize și alertă în cazul în care rezultatele au valori anormale.

Scanare documente: Permite scanarea și atașarea imaginilor clinice, diagramelor, fișierelor audio, desenelor medicale și fotografiilor.

Analizând funcționalitățile sistemelor descrise mai sus, se observă că, în medie, serviciile cele mai vitale sunt: lista de pacienți a medicului, prelucrarea și partajarea datelor din istoricul medical. O comparație între sistemele analizate este prezentată în tabelul 2.3.

	Programări	Listă de pacienți	Istoric medical	Analize	Licență
AdvancedMD	X	X	X		X
cureMD		X	X	X	X
anthenaOne	X	X	X	X	X

Prin urmare, am ales ca aceste două funcționalități principale, lista de pacienți și istoricul medical, să constituie baza implementării și descrierii aplicației MyMedicalRecord. Aplicația rezultată conține un număr mai mic de facilități, de aceea aceasta va fi gratuită.

Capitolul 3

Analiza cerințelor

Motivația realizării acestei aplicații este nevoia de a se asigura partajarea istoricului medical. De la aceasta se dezvoltă cerințele aplicației: completarea istoricului medical, vizualizare, filtrare, partajare și stocare. Analiza cerințelor a fost efectuată împreună cu un medic, reprezentant al cadrelor medicale, care a împărtășit experiența profesională în privința datelor necesare pentru a analiza trecutul medical al pacientului. Aplicația oferă aceste facilități în funcție de tipul utilizatorului. De aici se dezvoltă cele două perspective: perspectiva pacientului și perspectiva medicului.

3.1 Perspectiva pacientului

Aplicația oferă, pentru pacient, un domeniu sigur care deservește ca un mediu de stocare a documentelor fizice în format digital și ca o modalitate de partajare a datelor pacienților cu medicul.

Istoricul medical reprezintă totalitatea evenimentelor medicale. Pacientul este cel care, prin intermediul unui formular, completează datele asociate evenimentului medical, informații care vor ajunge în baza de date a aplicației. Din perspectiva unui pacient, scopul istoricului medical este acela de vizualizare. Secțiunea dedicată acestuia cuprinde câmpuri de filtrare, care ordonează evenimentele medicale în funcție de dorința utilizatorului: dată (an, luna, zi), tipul evenimentului și numele medicului.

Datele personale ale pacientului sunt private. Acesta alege medicii care au permisiunea de a vizualiza și prelucra informațiile medicale personale.

3.2 Perspectiva medicului

MyMedicalRecord reprezintă pentru medic un instrument care permite vizualizarea datelor legate de pacienți, în format digital. După crearea unui cont de medic, meniul aplicației va conține link-uri către pagină de profil și către lista de pacienți.

Profilul medicului conține informații personale, precum: nume, prenume, email, număr de telefon, instituția în care acesta își desfășoară activitatea, funcția pe care acesta o ocupă, adresa instituției, orarul cabinetului și date legate de pregătirea și experiența profesională (educație, locuri de muncă anterioare).

Pentru a vizualiza datele unui pacient, mai întâi medicul trimite o cerere de acces către pacient. Răspunsul pacientului este unul pozitiv sau unul negativ, aprobare, respectiv respingere. În cazul în care răspunsul este unul pozitiv, pacientul este adăugat

în lista de pacienți a medicului. Din acel loc medicul are acces la informațiile personale relevante ale pacientului: înălțime, greutate, gen, comorbidități și istoricul medical. Pacientul are opțiunea să restricționeze accesul medicului în orice moment.

Sub lista de pacienți, există o secțiune, în care medicul poate vizualiza statistici legate de pacienții lui din care poate observa anumite tipare, de exemplu: procentajul pacienților care prezintă o anumită comorbiditate, comorbidități în funcție de vârstă, media de vârstă și altele.

Capitolul 4

Aplicatia MyMedicalRecord

În acest capitol este descrisă atât structura aplicației ca un sistem compact, cât și arhitectura componentelor individuale. Este descompus fluxul de date și este explicată și exemplificată direcția acestuia. Sunt descrise tehnologiile folosite și este justificată alegerea lor, accentul fiind pe beneficiile acestora. În final sunt expuse detaliile de implementare ale aplicației și parcursul utilizatorului pe platformă.

4.1 Arhitectura aplicației

În realizarea aplicației am optat pentru modelul MVVM datorită faptului că este un tip de arhitectură performant, relativ nou și potrivit pentru aplicații web.

Model-View-ViewModel (MVVM) este un model de proiectare software care permite separarea logicii programului și controlul interfeței cu utilizatorul. MVVM este cunoscut și ca model-view-binder. La fel ca multe alte modele de design, ajută la organizarea codului și la împărțirea programelor în module pentru a face dezvoltarea, actualizarea și reutilizarea codului mai simple și mai rapide. Separarea codului în MVVM este împărțită în View, ViewModel și Model.

View este o colecție de elemente vizibile, care primește intrările utilizatorului. Aceasta include interfețele utilizator (UI), animații și text. Prin interacțiunea cu conținutul componentei View (apăsarea unui buton, navigarea la altă pagină), nu este modificat direct ceea ce este prezentat. Acțiunile utilizatorilor declanșează evenimente care sunt trimise în componenta logică care decide cum și dacă se schimbă componenta de vizualizare.

ViewModel este situat între straturile View și Model. Aici sunt găzduite controalele pentru interacțiunea cu View, în timp ce se utilizează o legătură pentru a conecta elementele UI din View la controalele din ViewModel.

Modelul găzduiește logica pentru program, care este preluată de ViewModel la primirea propriei intrări de la utilizator, prin View.

4.1.1 Arhitectura componentei front-end

Figura 4.1 prezintă componentele din partea vizuală a aplicației.

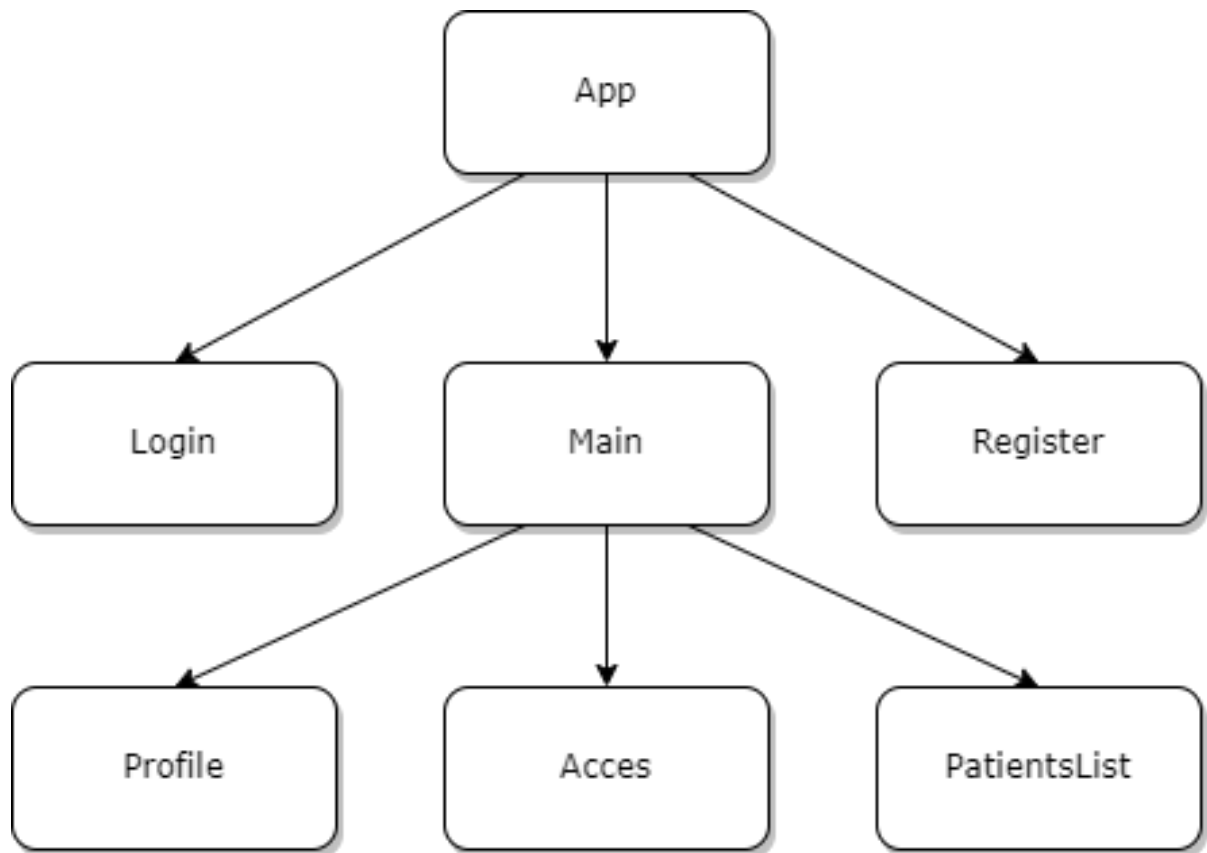


Figura 4.1: Arhitectura front-end

Componenta părinte a tuturor componentelor este **App**. În funcție de datele utilizatorului, **App** decide pe care componentă să o afișeze. În cazul în care utilizatorul nu este conectat pe platformă, aplicația va putea să afișeze doar componentele **Login** sau **Register**. În schimb, dacă utilizatorul este conectat cu succes pe platformă se va afișa conținutul componentei **Main**. La rândul ei, **Main**, decide pe care dintre componente să le afișeze în funcție de *tipul* utilizatorului conectat și în funcție de cale.

Componentele front-end-ului sunt alcătuite din sub-componente, printre care cele mai importante sunt:

- profileCard

Este componenta principală din cadrul paginii de profil. Ea afișează datele personale ale utilizatorului, cum ar fi date de contact, poza de profil, nume complet, email, număr de telefon și alte câmpuri care depind de tipul contului. Id-ul este afișat într-un buton pentru că, la apăsare, acest este copiat în clipboard și poate fi comunicat medicului/pacientului. Componenta profileCard conține butonul ”Modifică”, care la apăsare deschide un formular pentru actualizarea datelor.



Figura 4.2: Componenta profileCard

Figura 4.2 prezintă componenta afișată pentru un cont de tip medic. Aceasta are aceeași structură și pentru contul de pacient, singura diferență sunt câmpurile care sunt afișate.

- accesProfileCard

Captura de ecran 4.3 arată o cerere primită de la medic. Butonul cu valoarea id-ului are aceeași funcționalitate ca la componenta profileCard, celelalte butoane declanșează acțiunile sugerate de conținutul lor.

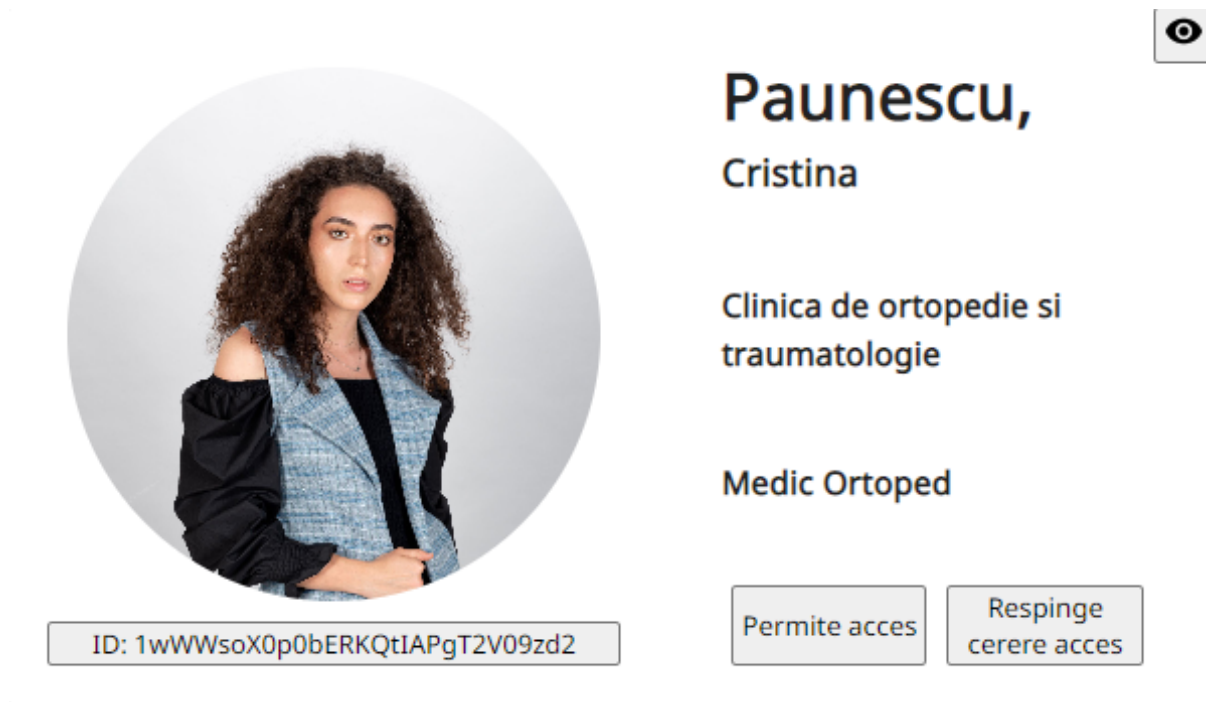


Figura 4.3: componenta accesProfileCard

Componenta apare pe pagina de acces atât a pacientului cât și a medicului. Diferențele între tipurile de cont sunt butoanele și acțiunile pe care le declanșează.

- patientsList

Figura 4.4 arată, din perspectiva medicului, lista de pacienți. Prin intermediul listei, medicul poate vizualiza datele pacienților săi, le poate filtra și ordona.

ID	Șterge	Avatar	Prenume	Nume	E-mail	Comorbidități	Gen	Înălțime	Greutate	Telefon
77IEHQNipQWHDazx1fkfqEZCX...			Dorian Alexan...	Popovici	dorian@yahoo.com	Listă de comorbidități	Masculin	168 cm	80 kg	40257291300
McaShSThepSvYtII4ExsqLtd8yD2			Tomas	Benitez	benitezthomas@gmail.com	Listă de comorbidități	Masculin	181 cm	95 kg	
TZcSpaOhybMgjoNDyCIWvYam...			Paul	Bechet	paul.bechet76@gmail.com	Listă de comorbidități	Masculin	164 cm	55 kg	40743929800
VnTxOpH4MIF96kow5d7YMuPsg...			Silvia	Manulescu	silviamanulescu62@hotmail...	Listă de comorbidități	Feminin	164 cm	96 kg	40762440806
iBQO4041WaeTbcbSTzEILgFcSk1			Darius David	Popovici	popovicidd@gmail.com	Listă de comorbidități	Masculin	186 cm	64 kg	40749898131
IsnUEao8ufOwBjIHruHQx4zQzm...			Adela Maria	Rosu	adela.maria84@gmail.com	Listă de comorbidități	Feminin	165 cm	56 kg	40757786443

Figura 4.4: Componenta patientsList

La începutul fiecărui rând se găsește butonul care elimină pacientul din lista medicului.

- reactModal

Este componenta care se declanșează la apăsarea butonului ”Modifică” din cadrul paginii de profil.

The image shows a web form titled 'reactModal' with two buttons at the top: 'Abandoneaza' and 'Trimite'. The form contains several input fields: 'First name' with the value 'Gabriel Marius', 'Last name' with the value 'Cazac', 'Telephone number' with a flag icon and the value '+40 765 434 101', 'Greutate' (Weight) with the value '70' and unit 'kg', 'Înălțime' (Height) with the value '187' and unit 'cm', a dropdown menu for 'Comorbidități' (Comorbidities), and another dropdown menu for 'Gen' (Gender) with the value 'Masculin'. At the bottom, there is a 'Choose File' button and the text 'No file chosen'.

Figura 4.5: Componenta reactModal

Ea conține un formular prin intermediul căruia, utilizatorul poate adăuga date sau le poate actualiza pe cele vechi.

Figura 4.5 reprezintă formularul pentru contul de pacient. Componenta este asemănătoare cu cea a medicului, diferența o fac câmpurile care trebuie completate.

S

4.1.2 Arhitectura componentei back-end

Componenta back-end se rezumă la logica din spatele aplicației, direcția fluxului de date și la comunicarea cu baza de date.

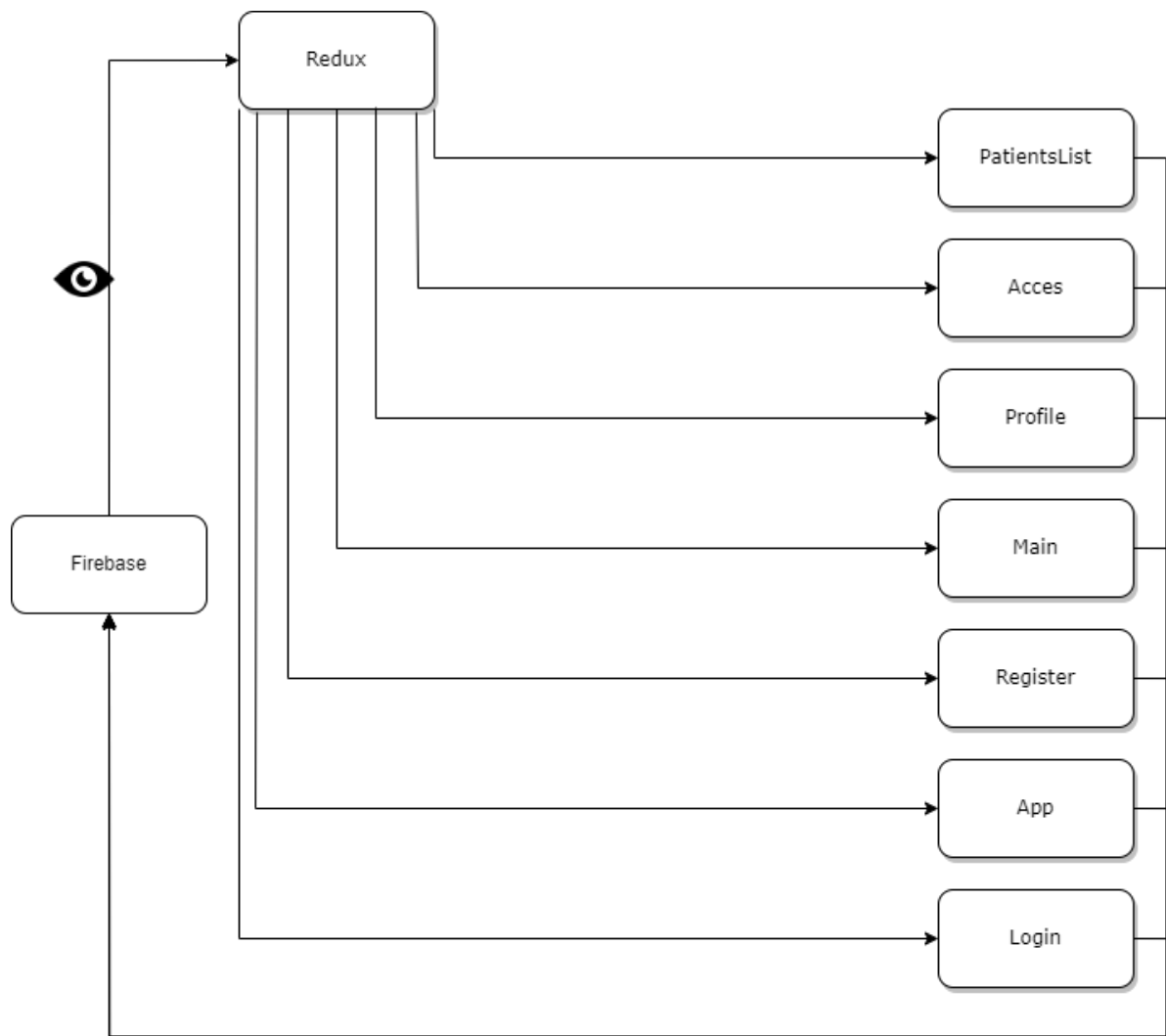


Figura 4.6: Ciclul fluxului de date

Se observă, din figura 4.6, că, pentru toate canalele de comunicare între componente, fluxul de date este unidirecțional. Ciclul de comunicare, de la început până la final, funcționează în felul următor:

Redux: Este componenta care deține *state-ul* global și care trimite această informație către componentele care aparțin front-end-ului. Tot ea monitorizează schimbările din baza de date și le actualizează în interfața aplicației.

Componenetele view: Acestea primesc *state-ul* aplicației de la Redux și, la interacțiunea utilizatorului cu componenta vizuală (de exemplu completarea informațiilor în formular), declanșează o schimbare în atributele contului, pe care mai apoi le trimit către baza de date.

4.1.3 Arhitectura bazei de date

Baza de date a aplicației cuprinde două tabele și un spațiu de stocare în cloud. Primul tabel conține datele de conectare a utilizatorilor, respectiv email-ul și parola asociată

contului. Aceste câmpuri sunt completate în formularul de înregistrare, la crearea contului.

În cel de al doilea tabel, sunt stocate datele care aparțin utilizatorilor. O parte din aceste informații sunt completate la înregistrarea contului, iar cealaltă parte este dedicată completării după conectarea utilizatorului pe platformă. În funcție de tipul contului, medic sau pacient, documentul utilizatorului din baza de date conține următoarele câmpuri:

- **Pacient:**

Camp	Tip	Descriere
email	string	email-ul compeltat la crearea contului
firstName	string	prenumele
lastName	string	numele
role	string	tipul contului (Medic, Pacient)
patientBirthDate	string	data de nastere a pacientului
createdAt	timestamp	data la care a fost creat contul (auto-generat)
myMedicsPending	Array<string>	lista medicilor care asteapta raspuns pentru cerere
myMedicsAllowed	Array<string>	lista medicilor carora li s-a permis accesul
profilePictureURL	string	link catre poza de profil stocata in cloud
comorbidities	Array<string>	lista de comorbiditati
gender	string	gen
height	string	inaltime (cm)
weight	string	greutate (kg)
medicalRecord	Array<JSON>	sir de evenimente medicale
tel	string	numar de telefon

- **Medic:**

Camp	Tip	Descriere
email	string	email-ul folosit la crearea contului
firstName	string	prenumele
lastName	string	numele
role	string	tipul contului (Medic, Pacient)
medicFunction	string	functia medicala
createdAt	timestamp	data la care a fost creat contul (auto-generat)
myPatientsPending	Array<string>	lista pacientilor care nu au raspuns cererii
myPatientsAllowed	Array<string>	lista pacientilor la care este permis accesul
profilePictureURL	string	link catre poza de profil stocata in cloud
medicInsitution	string	institutia medicala
experience	Array<string>[3]	lista cu experientele profesionale
specialities	Array<string>[3]	lista cu specializarile medicale
location	string	locatia institutiei medicale
timetable	JSON	orar de activitate
faculty	string	facultatea absolvita
absolvationYear	string	anul absolvirii
tel	string	numar de telefon

Intrările bazei de date sunt actualizate în funcție de tipul variabilei care le conțin. Astfel, intrările de tipul string sunt înlocuite direct de informațiile noi, pentru intrările de tipul Array<string>, în cazul în care se dorește a fi adăugat un obiect, se efectuează o reuniune între mulțimea deja existentă și mulțimea nouă, iar dacă se dorește o ștergere se realizează o scădere între mulțimea obiectelor existente și mulțimea obiectelor care doresc a fi șterse. Aceleași principii se aplică și pentru variabilele de tip Array<JSON>, singura diferență fiind modul intrărilor de tip frunză.

4.2 Tehnologii folosite

Scopul acestui capitol este să descrie, în parte, tehnologiile folosite în implementarea aplicației. Accentul este pus pe beneficiile lor, printre care cele mai importante aspecte sunt securitatea, siguranța oferită, performanța și claritatea în implementare.

4.2.1 React.js

React.js este o bibliotecă JavaScript gratuită și open-source pentru construirea de interfețe utilizator bazate pe componente UI. Aceasta poate fi folosită ca bază în dezvoltarea de aplicații cu o singură pagină, mobile sau randate pe server cu cadre precum Next.js. Cu toate acestea, React se preocupă doar de gestionarea stării și de redarea acelei stări către DOM(Document Object Model), astfel încât crearea aplicațiilor React necesită de obicei utilizarea de biblioteci suplimentare pentru rutare, precum și anumite funcționalități pe partea clientului.

Front-end-ul aplicației este implementat cu ajutorul ReactJS, așadar conținutul este bine organizat, implicit actualizat, aerisit și timpul schimbărilor în cadrul interfeței utilizator este neobservabil. React încurajează dezvoltatorii să scrie cod într-o manieră bine organizată, sugerând o structură pentru directoarele și fișierele aplicației.

Principiul de funcționare este simplu și eficient: o componentă poate deține un obiect de tip JSON, numit state, care dictează cum se comportă interfața. Atunci când o intrare din state se schimbă, React va ști să actualizeze doar sub-componentele care se folosesc de această intrare, spre deosebire de metoda clasică, de a reactualiza toată pagina. State-ul unei componente poate fi trimis, mai departe sub-componentelor, care pot consuma date din el, dar nu îl pot modifica.

4.2.2 Redux

Redux a apărut ca rezolvare a unei probleme pe care React o are. Pe măsură ce aplicația crește, devine dificilă menținerea organizată a codului și a fluxului de date. Redux rezolvă această problemă gestionând starea aplicației cu un singur obiect global numit Store. Principiile fundamentale Redux ajută la menținerea coerenței în întreaga aplicație, ceea ce face depanarea și testarea mai ușoară.

Redux este, de fapt, rădăcina aplicației. Componentele frunză pot comunica cu aceasta printr-un abonament, în funcție de citire sau scriere. Astfel componentele din interfață au acces direct la state, fără ca obiectul să parcurgă un număr mare de elemente până să ajungă la aceasta. Tot ele pot să schimbe starea state-ului global, lucru nepermis în React.

În arhitectura Redux sunt cinci componente logice:

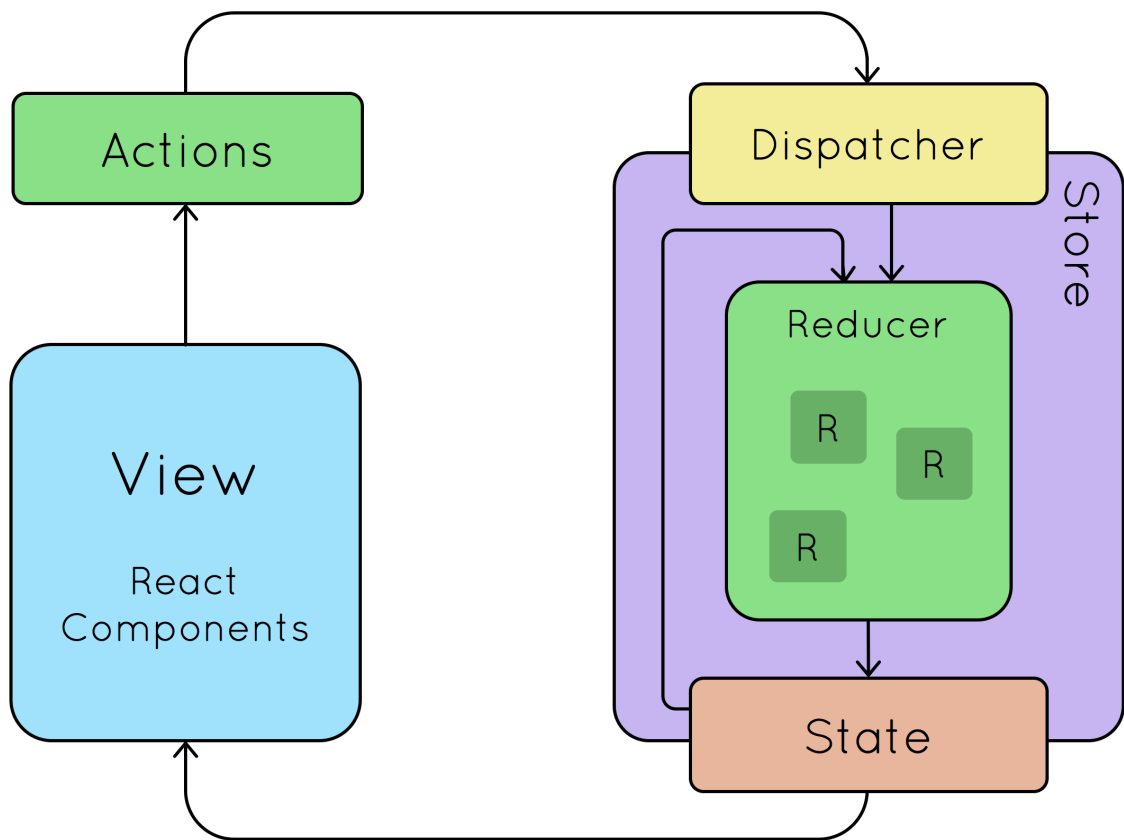


Figura 4.7: Arhitectura componentei Redux

Actions: Sunt obiecte JavaScript simple care sunt folosite pentru a trimite informații de la componenta React către alte blocuri de construcție. Acțiunea trebuie să aibă proprietatea „tip” și date opționale ale încărcăturii utile care trebuie comunicate altor blocuri.

Dispatcher: Este folosit pentru a executa cod asincron între Actions și Reducer. Interceptează toate acțiunile trimise de la componente, înainte ca acestea să fie trecute la Reducer.

Reducer: Este o funcție pur sincronă, care primește ca argumente starea anterioară și un obiect Action și returnează starea următoare. Aceste stări sunt stocate ca obiecte și tot ele specifică modul în care starea unei aplicații se schimbă ca răspuns la o acțiune trimisă în Store. Aplicația poate avea mai multe componente Reducer în funcție de utilitate și poate fi combinată într-un singur Reducer folosind o metodă predefinită.

Store: Este un loc centralizat pentru a menține starea aplicației. Este recomandată păstrarea unui singur magazin în orice aplicație Redux. Dezvoltatorul poate accesa, actualiza și se poate abona la Store folosind diverse metode de ajutor.

State: State reprezintă produsul final care, în continuare, va fi trimis componentelor abonate din React.

Rolul Redux în aplicația MyMedicalRecord este crucial, el menține starea utilizatorului și comunicarea dintre aceasta și componentele care au nevoie de ea. Redux privește tot timpul baza de date și, dacă informațiile se schimbă, va actualiza sesiunea utilizatorului în cadrul sistemului.

4.2.3 Material UI

Material UI este un framework front-end open-source pentru componentele React. Material UI se concentrează pe furnizarea de diferite modele și creează texturi concentrându-se pe modul în care componentele dețin umbre și reflectă lumina. Cuprinde un volum enorm de componente HTML asupra cărora sunt aplicate stiluri în CSS, precum butoane, liste, iconițe, formulare, containere și lista poate continua. Modelele furnizate acceptă parametri care decid aspectul elementului: culoare, poziționare, dimensiune, precum și atribute de localizare (componente de selecție a datei, a orei sau a numărului de telefon).

Material UI reprezintă o scurtătură în implementarea front-end-ului. Astfel dezvoltatorul se poate concentra pe aspecte precum fluxul de date, securitate și performanță, în timp ce interfața utilizator este alcătuită din componente din această bibliotecă.

4.2.4 Recharts

Este o bibliotecă care servește pentru afișarea de reprezentări grafice în cadrul aplicațiilor implementate cu ajutorul React. Conține o listă generoasă de tipuri de diagrame: diagramă cu bare, diagramă linie, diagramă radială, diagramă prin puncte, diagramă de canal și diagramă Sankey. Este cea mai populară librărie din tipul său datorită aspectului îngrijit și posibilității de personalizare.

Componentele din această bibliotecă se regăsesc în cadrul aplicației, în perspectiva medicului, sub lista de pacienți. Rolul graficelor este de a crea o imagine de ansamblu pentru cel care vizualizează, ca apoi acesta să poată lua decizii bazate pe datele din diagramă.

4.2.5 React-router-dom

React Router DOM este un pachet care permite implementarea rutării dinamice într-o aplicație web. Permite afișarea de pagini și permite utilizatorilor să le navigheze. Este o bibliotecă de rutare completă pentru client și server pentru React. React Router Dom este folosit pentru a construi aplicații cu o singură pagină, adică aplicații care au multe pagini sau componente, dar pagina nu este niciodată reîmprospătată, în schimb, conținutul este preluat dinamic pe baza adresei URL, acest proces se numește routing.

Avantajul major este că pagina nu trebuie să fie reîmprospătată atunci când se face clic pe un link către o altă pagină. În plus, este foarte rapid în comparație cu navigarea tradițională în pagină. Astfel experiența utilizatorului este mai bună, iar aplicația are performanțe generale mult mai bune.

React Router Dom are multe componente utile și pentru a crea o rutare complet funcțională, este nevoie de cele mai multe dintre acestea.

Router (importat de obicei ca BrowserRouter): Componenta părinte care este folosită pentru a stoca toate celelalte elemente. Totul din aceasta va face parte din funcționalitatea de rutare.

Comutare: Componenta Comutare este utilizată pentru a înapoia numai primul traseu care se potrivește cu locația, mai degrabă decât pentru a reda toate rutele care se potrivesc.

Rută: Aceasta verifică adresa URL curentă și afișează componenta asociată cu acea cale exactă. Toate rutele sunt plasate în elementele comutatorului. Componenta Rută preia 2 parametri. Prima este calea care va fi în URL și a doua este componenta care va fi afișată dacă URL-ul curent se potrivește cu calea din primul parametru.

Link: Componenta Link este folosită pentru a crea link-uri către diferite rute.

4.2.6 Firebase

Google Firebase este un software pentru dezvoltare de aplicații susținut de Google care le permite dezvoltatorilor să creeze aplicații iOS, Android și web. Firebase oferă instrumente pentru urmărirea analizelor, raportarea și remedierea blocărilor aplicației, crearea de marketing și experimente de produs.

Firebase oferă o serie de servicii, inclusiv:

Autentificare: Firebase Authentication facilitează dezvoltatorilor să construiască sisteme de autentificare sigure și îmbunătățește experiența de conectare și de integrare pentru utilizatori. Această caracteristică oferă o soluție completă de identitate, care acceptă conturi de e-mail și parole, autentificare telefonică, precum și autentificare Google, Facebook, GitHub, Twitter și multe altele.

Baza de date în timp real: Firebase Realtime Database este o bază de date NoSQL găzduită în cloud, care permite stocarea și sincronizarea datelor între utilizatori în timp real. Datele sunt sincronizate între toți clienții în același timp și sunt încă disponibile atunci când o aplicație este offline.

Performanță: Serviciul de monitorizare a performanței Firebase oferă dezvoltatorilor o perspectivă asupra caracteristicilor de performanță ale aplicațiilor lor iOS și Android pentru a-i ajuta să determine unde și când performanța aplicațiilor lor poate fi îmbunătățită.

Laborator de testare: Firebase Test Lab este o infrastructură de testare a aplicațiilor bazată pe cloud. Cu o singură operație, dezvoltatorii își pot testa aplicațiile iOS sau Android pe o varietate de dispozitive și configurații de dispozitiv. Ei pot vedea rezultatele, inclusiv videoclipuri, capturi de ecran și jurnale, în consola Firebase.

Am ales să folosesc Firebase ca baza de date aplicației datorită simplității de folosire și datorită compatibilității cu ReactJS. Firebase necesită un fișier de configurare în cadrul aplicației, în care se află identificatorul bazei de date și protocoale de protecție, precum publickey și privatekey. Conectarea cu baza de date și funcțiile care trimit cereri către aceasta se găsesc în fișierul `/src/utils/firebase.js`.

4.3 Detalii de implementare

Acest subcapitol descrie procesul implementării, pașii care au fost urmați și provocări întâlnite în acest parcurs.

Procesul începe cu implementarea funcționalității de înregistrare, mai exact crearea formularului care preia datele de la utilizator, realizarea funcției care trimite aceste informații către baza de date și verificarea lor. Am realizat componenta numită Register și am descris un formular asupra căruia am aplicat attribute CSS. Câmpurile formularelor diferă în funcție de tipul contului selectat.

Pentru funcția de autentificare am creat componenta Login, asemănătoare aspectului componentei Register, după care am testat autentificarea utilizatorului deja existent în baza de date. Logica pentru autentificare este implementată de firebase.

În continuare, am implementat Redux și listener-ul pentru baza de date, am definit sensul fluxului de date în cadrul platformei și am creat dependențe pentru rute, în funcție de tipul contului.

Prin urmare, am creat meniul pentru navigare și am decis că spațiul rămas după afisarea meniului reprezintă componenta Main. Această componentă, în funcție de tipul contului și de ruta la care se află utilizatorul, afișează unul din celelalte elemente. Am implementat componenta ProfileCard, atât pentru medic cât și pentru pacient, care conține datele utilizatorului și în care este afișată pe pagina de profil. Prin urmare, funcționalitatea de modificare a datelor a venit ca necesitate a dorinței de a actualiza informațiile din cadrul paginii de profil. Următorul pas a fost implementarea serviciului de adăugare a evenimentului medical, afișarea istoricului medical și filtrele care pot fi aplicate pe acesta.

Utilitatea de partajare a datelor a fost o provocare. Mai întâi, am definit două liste în baza de date pentru fiecare cont. O listă este pentru conturile care au trimis/primit cererea, cealaltă listă este pentru conturile permise. Apoi, am creat subcomponenta prin care medicul poate trimite o cerere către pacient și elementul prin cadrul căruia pacientul acceptă această solicitare. Atunci când medicul trimite o cerere către pacient, id-ul acestuia este adăugat în lista pacientului dedicată pentru cereri la care nu a răspuns. În același timp, id-ul pacientului către care s-a trimis solicitarea este adăugat în lista medicului pentru pacienții care încă nu au răspuns. Dacă pacientul acceptă, atunci în listele pentru cereri permise ale ambilor utilizatori se adaugă id-ul celeilalte părți. În funcție de aceste liste am construit logica afișării solicitărilor pe pagina de acces și butoanelor dinăuntrul lor.

Mai apoi, am folosit o componentă din MaterialUI care definește lista de pacienți. Am parametrizat și adaptat lista la condițiile proprii și am adăugat iconița istoricului medical, care atunci când este apăsată deschide o fereastră în care se află istoricul medical al pacientului și filtre de căutare.

În cele din urmă, am folosit datele introduse de pacienții din lista medicului, pentru a afișa diagramele. Aceste grafice descriu vizual repartizarea atributelor pacienților în procentaje.

4.4 Exemple de utilizare

Acest subcapitol descrie parcursul utilizatorului pe platforma, în funcție de tip, medic sau pacient. Viitorul utilizator începe interacțiunea cu platforma prin înregistrare.

Apoi, acesta este autentificat pe platforma, iar aplicația decide, în funcție de tipul contului, ce va afișa pentru sesiunea curentă.

Rolul pacientului pe platforma este de a-și adăuga propriile informații în cadrul paginii de profil, după care începe să își completeze istoricul medical. Totalitatea evenimentelor medicale reprezintă istoricul medical. Pacientul adaugă evenimente prin intermediul unui formular, apoi acestea sunt afișate sub forma unui element care conține date personale. În continuare, pacientul decide scopul pentru care folosește aplicația. Astfel, el poate alege să accepte cererile medicilor și să partajeze datele personale împreună cu istoricul medical sau poate alege să păstreze datele pentru sine și să folosească aplicația ca un spațiu de stocare a arhivei digitale.

Sistemul oferă, pentru medic, un loc sigur în care acesta își poate coordona pacienții și poate vizualiza datele acestora. Medicul începe parcursul pe platforma cu crearea unui cont de medic, apoi urmând autentificarea. Medicul începe să își completeze propriul profil cu datele personale și datele legate de instituția în care activează, locația instituției, funcția pe care o deține, experiența profesională, specializări și educație.

Medicul, apoi, caută pacienții înregistrați pe platformă. Aceștia se găsesc în cadrul paginii de acces și pot fi filtrați după nume sau id. În continuare, medicul trimite o solicitare către pacient și, dacă acesta o acceptă, va fi adăugat în propria listă de pacienți.

Cu ajutorul listei de pacienți, medicul poate să vizualizeze date personale, attribute, precum și istoricul medical al acestuia. Pe acest set de date se pot aplica filtre sau pot fi ordonate, pentru a fi găsite mai ușor.

Pe pagina cu lista, medicul are la dispoziție o grupare de diagrame care descriu repartizare, în procentaj, a atributelor pacienților din cadrul listei.

4.4.1 Diagrama cazurilor de utilizare

Diagrama cazurilor de utilizare este o modalitate bună de a enumera facilitățile aplicație și de a descrie relațiile dintre acestea.

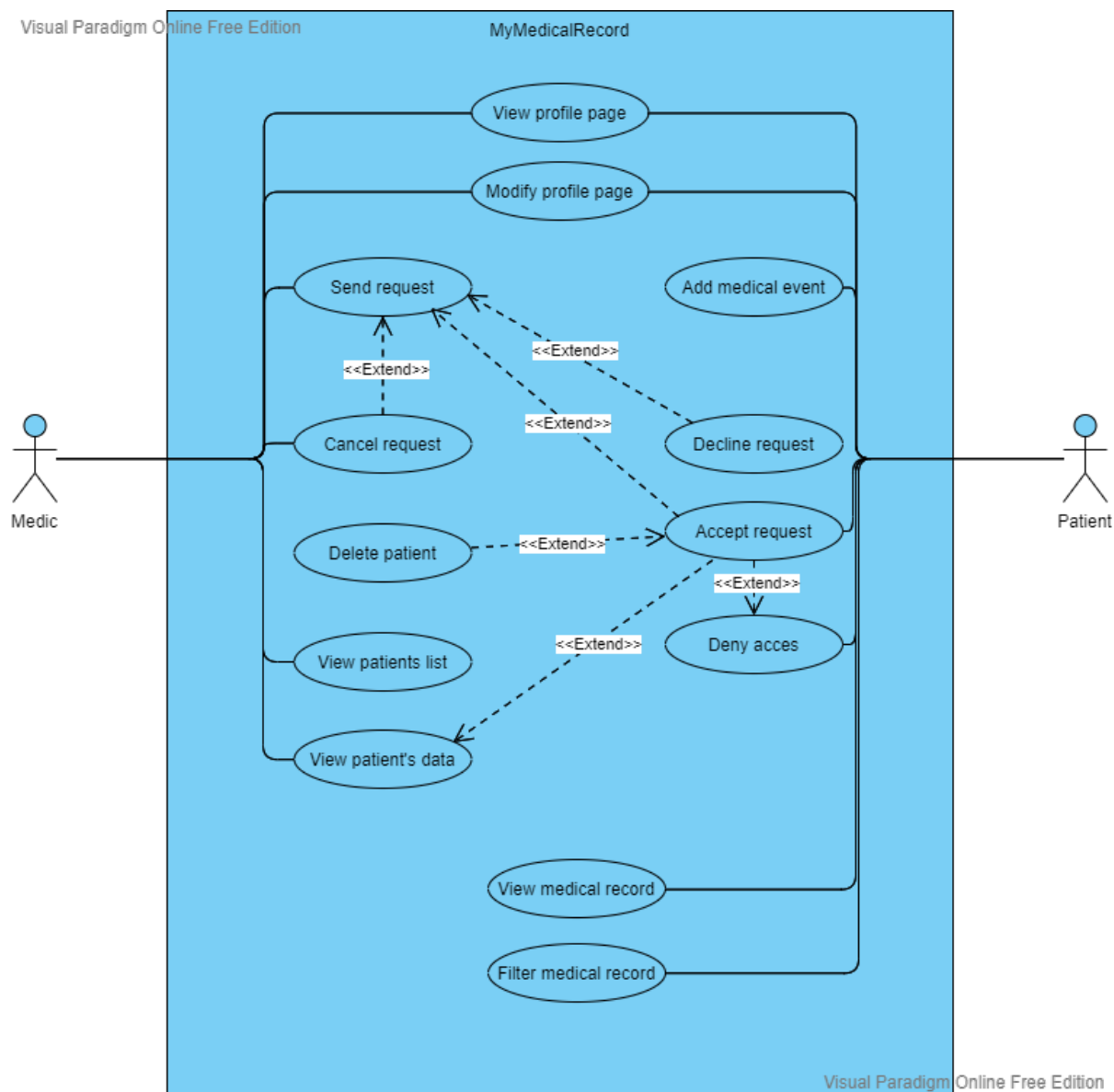


Figura 4.8: Diagrama cazurilor de utilizare

În figura 4.8 sunt prezente următoarele elemente: actorii sistemului, adică medic și pacient, sistemul, cazurile de utilizare ale sistemului și relațiile de dependență dintre acestea.

4.4.2 Automatul de stări

Automatul de stări al aplicației (vezi figura 4.9) descrie parcursul utilizatorului pe aplicație, în funcție de tipul contului autentificat.

Capitolul 5

Concluzii și direcții viitoare de dezvoltare

Acest ultim capitol are rolul de concluzie. El compară obiectivele inițiale cu sistemul final și expune dezvoltări ulterioare posibile pentru aplicație. Propunerea inițială descrie implementarea unei aplicații web, pe care medicii și pacienții o pot folosi pentru prelucrarea datelor. Astfel, aplicația oferă pacienților beneficii precum:

- Stocare
- Partajare

Utilizatorul de tip pacient poate folosi aplicația în scopul de a depozita, în format digital, propria arhivă medicală. El adaugă, prin intermediul unui formular, evenimente medicale precum vizite la medic sau internări în spital. Totalitatea evenimentelor medicale constituie istoricul medical propriu, asupra căruia se aplică filtre de căutare pentru a fi găsite în timp mult mai scurt în comparație cu căutarea manuală.

După completarea atributelor personale, pacientul are opțiunea de a partaja informațiile cu medicul ales de el. Astfel, datele sunt confidențiale și medicul poate să efectueze o analiză pe aceste informații. În cazul în care valorile datelor sunt anormale, medicul are opțiunea să îl contacteze pe pacient via email sau via telefon.

Facilitățile aplicației care îl avantajează pe medic sunt:

- Filtrare
- Ordonare
- Vizualizare

Medicul are acces la datele minimalistice ale pacienților înregistrați pe platformă, adică nume, prenume, vârsta și id. În funcție de acești factori, medicul poate să identifice pacientul către care dorește să trimită cerere de acces. După ce pacientul acceptă această solicitare, este adăugat în lista personală a medicului și permite accesul deplin la datele sale.

Prin cadrul listei de pacienți, medicul poate vizualiza informații care aparțin pacienților, cum ar fi atribute (înălțime, greutate și listă de comorbidități) și istoricul medical. Datele pot fi ordonate și filtrate, astfel vizualizarea este accesibilă și necesită mai puțin timp.

5.1 Rezultate obținute

Facilitățile propuse pentru începutul aplicației se regăsesc și în varianta finală a acesteia. Interfața și experiența cu utilizatorul sunt minimaliste, tocmai pentru ca procesul de navigare și folosire generală a aplicației să fie unul intuitiv și simplu. Securitatea este realizată, în mare parte, de direcția fluxul de date bine definit, de către Firebase, în legătură cu autentificarea, înregistrarea și stocarea datelor în format digital, și de către modul de decizie al comportamentului elementului vizual în funcție de tipul utilizatorului autentificat pe platformă.

După autentificarea cu un cont de pacient, aplicația prezintă două pagini:

- Profil


Figura 5.1 este rezultată prin concatenarea mai multor capturi de ecran, pentru a fi vizibile toate componentele paginii de profil.

Salut, Gabriel Marius ! Deconectare

Profil

Acces

Modifica



Gabriel Marius Cazac

gab8989@yahoo.com

40765434101

ID: XuGOUcOLeherHPotNRa64XZnJ43

Vârsta:

27 ani

Gen:

Masculin

Înălțime:

187 cm

Greutate:

70 kg

Comorbidități:

Insuficiența cardiacă

Astm cronic

Boli digestive cronice

Hipertensiune arterială

Adaugă eveniment medical

Tipul evenimentului

Vizită medic familie

Instituție

Diagnostic

Prescripție

Data

mm/dd/yyyy

Detalii

Numele medicului

Anulează

Trimite

Tipul evenimentului

Toate

Numele medicului

Anul

Luna

Ziua

numele medicului:

-

diagnostic:

054 - Hepatita virala acuta A

detalii:

-

perioada:

2022/2/25 - 2022/3/2

tipul evenimentului:

Internare in spital

instituție:

Spitalul Clinic de Urgenta Timisoara

prescripție:

tratament cu Interferon tratament cu analogi nucleotidici/nucleozidici

instituție:

Medicina Generala nr. 87, Timisoara

detalii:

Infectia cu meningococi descoperita in stadiu avansat. Prin urmare, tratamentul recomandat este mai concentrat.

numele medicului:

Popovici Cristian

prescripție:

Moldamin 1/sapt Paracetamol 2/zi

data:

2022/1/5

tipul evenimentului:

Vizită medic familie

diagnostic:

029 - Infectia cu meningococi

detalii:

Ochi stang: -0.25 Ochi drept: +0.25

diagnostic:

000 - Nu este cazul

prescripție:

Ochelari

numele medicului:

Cosmin Dieci

data:

2022/6/17

tipul evenimentului:

Vizită medic familie

instituție:

Optica Veszi Arad

Figura 5.1: Pagina de profil a pacientului

Pagina de profil a pacientului conține un meniu, o secțiune de profil, o subdiviziune pentru adăugarea unui eveniment medical, iar ultima secțiune conține istoricul medical și câmpuri de filtrare. În meniu se găsesc butoane care, la apăsare, navighează spre pagina dorită sau fac posibilă deconectarea utilizatorului de pe platformă.

Secțiunea de profil conține informațiile personale ale pacientului, de exemplu

34

vârsta, genul, greutatea, înălțimea și lista ce comorbidități. Tot aici se află poza de profil, numele complet, id-ul și datele de contact. Informații precum numele, fotografia și id-ul sunt vizibile pentru toți medicii înregistrați pe platformă și ajută la căutarea și identificarea pacientului dorit. Celălalt set de date, adică atributele personale, sunt partajate cu medicul doar cu acordul pacientului.

În continuare, se găsește formularul pentru adăugarea unui eveniment medical pe care pacientul îl poate deschide apăsând pe butonul ”Adaugă eveniment medical”. Aici, pacientul alege tipul de problemă medicală după care trebuie să completeze câmpurile aferente tipului de eveniment ales. Există trei tipuri de eveniment medical: vizită la medic de familie, vizită la medic specialist și internare în spital. După completarea tuturor câmpurilor, utilizatorul apasă pe butonul ”Trimite” care va salva informațiile în baza de date și le va afișa în secțiunea dedicată istoricului medical.

Ultima secțiune este dedicată istoricului medical. Aici este afișată totalitatea evenimentelor medicale introduse de către utilizator în sistem. Ele pot fi filtrate după dată, tipul evenimentului, numele medicului sau pot fi afișate fără niciun filtru aplicat.

• Acces

Captura de ecran (vezi figura 5.2) prezintă pagina de acces a pacientului. Aici sunt desfășurate toate cazurile posibile și sunt prezentate toate acțiunile pe care pacientul le poate declanșa pentru cererile de acces.

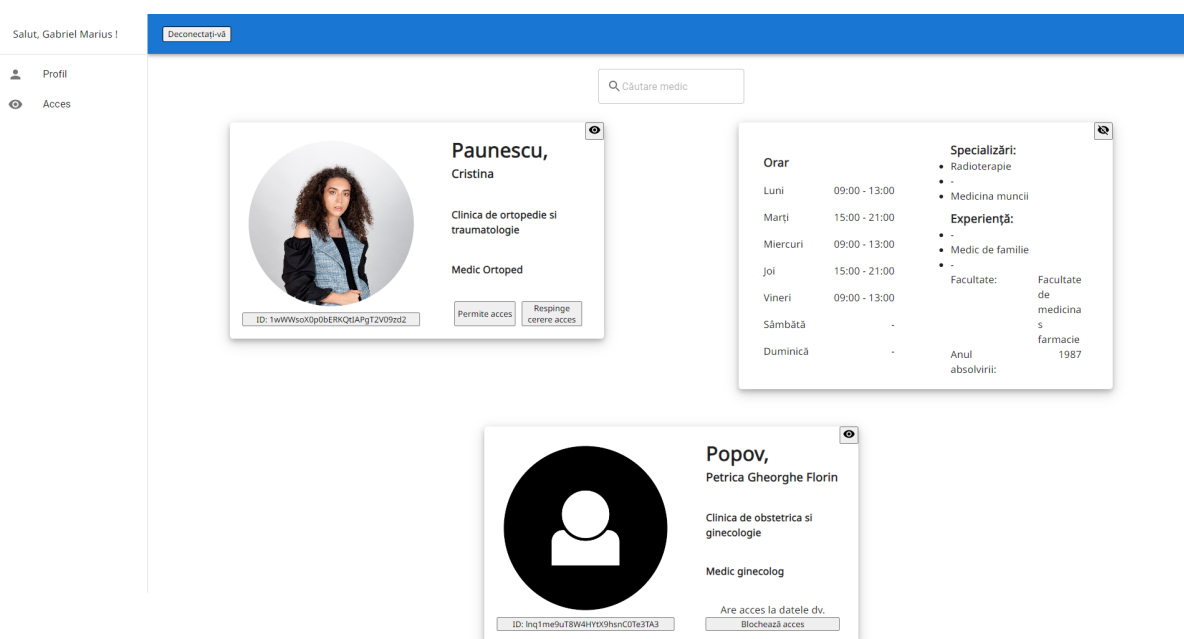


Figura 5.2: Pagina de acces a pacientului

Cererile trimise de către medic apar în cadrul acestei pagini sub forma unei structuri care conține poza de profil a medicului, id-ul, numele complet, instituția și

funcția medicală, și butoanele "Acceptă cererea" și "Respinge cererea". Pacientul poate căuta cererea după numele medicului sau după id-ul acestuia.

În colțul structurii se observă un buton care, odată apăsător, va schimba datele din aceasta și va afișa detalii legate de medic. Acest lucru se observă în a doua structură din imaginea 5.2. Vizualizarea detaliilor, precum specializări, experienți, educație și orar, are scopul de a informa pacientul în legătură cu medicul care a trimis cerere și joacă un rol important în alegerea, respingere sau acceptare pacientului.

În scenariul în care pacientul acceptă partajarea datelor, butoanele care reprezintă acțiunile de structură vor fi înlocuite de informarea "Are acces la datele dumneavoastră" și de butonul "Blochează acces". În cazul deciziei de a respinge accesul sau de a bloca accesul medicului, cererea este eliminată din baza de date și dispare din cadrul paginii de acces a pacientului.

Pentru medicii înregistrați pe platformă, aceasta oferă trei pagini:

- Profil

În figura 5.3 este prezentată pagina de profil a medicului. Aceasta conține o singură secțiune, în care sunt afișate datele contului. Apăsarea pe butonul "Modifică" din colțul structurii va deschide un formular prin intermediul căruia medicul poate să adauge, să șteargă sau să modifice date din profilul său.



Figura 5.3: Pagina de profil a medicului

Contul de medic deține un număr mai mare de informații decât cel al pacientului. Acestea deservesc pentru vizualizarea pacientului, atunci când o cerere este trimisă. Tot acestea oferă autenticitate contului de medic prin id, poză de profil, nume și date precum experiență, specializări și educație. În secțiunea de profil există și orarul, care informează pacientul despre intervalul orar la care se poate prezenta la cabinet.

- Acces

Pagina de acces a medicului cuprinde totalitatea pacienților înregistrați în cadrul platformei, care nu fac parte din lista personală a acestuia.

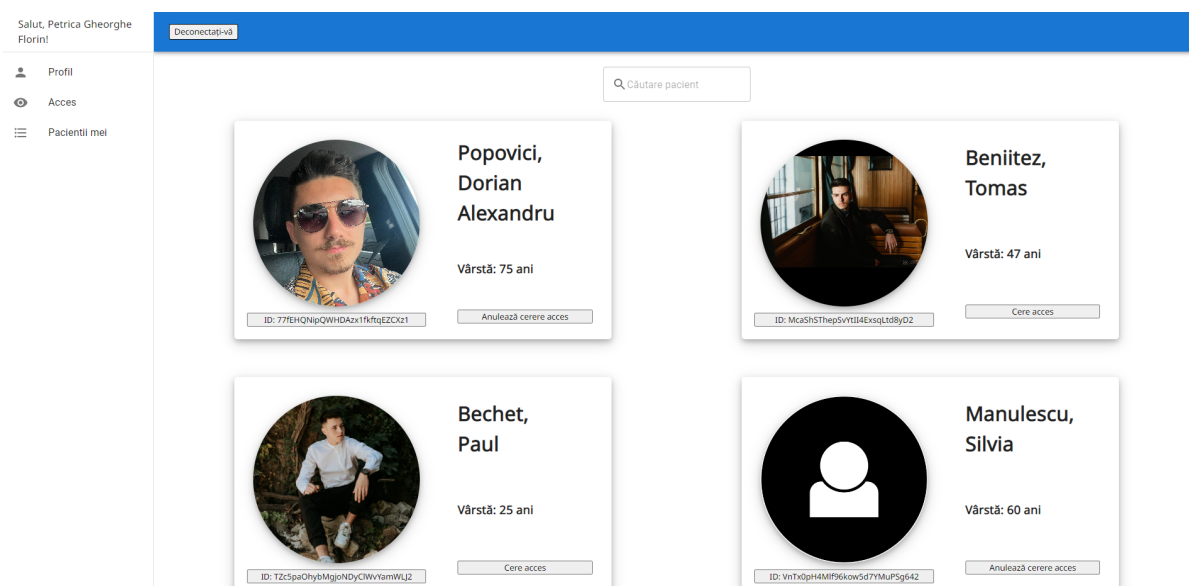


Figura 5.4: Pagina de acces a medicului

Figura 5.4 este o captură de ecran a paginii de acces a medicului. În cadrul acesteia se observă un câmp de căutare și câte o structură pentru fiecare pacient. Datele acestuia, pe care medicul le vede, sunt numele complet, poză de profil, id-ul și vârsta. Astfel el poate recunoaște pacientul către care dorește să trimită cererea de acces. Apăsând pe butonul "Cere acces" solicitarea este emisă, iar aceasta devine vizibilă în pagina de acces a pacientului. Odată trimisă cererea, ea poate fi anulată apăsând butonul "Anulează cerere acces". În cadrul acestei pagini apar toți pacienții înregistrați pe platformă, mai puțin cei care sunt deja în lista de pacienți.

- Pacienții mei

Această pagină conține o listă de pacienți și o secțiune în care se găsesc grafice. Structura listei este vizibilă în figura 5.5.

Salut, Ion Laurentiu ! Deconectati-vă

- Profil
- Acces
- Pacientii mei

Șterge	Avatar	Prenume	Nume	E-mail	Comorbidități	Gen	Înălțime	Greutate	Telefon	Vârstă	Istoric me
		Dorian Alexan...	Popovici	dorian@yahoo.com	Listă de comorbidități	Masculin	168 cm	80 kg	40257291300	75 ani	
		Tomas	Benitez	beniteztommas@gmail.com	Listă de comorbidități	Masculin	181 cm	95 kg		47 ani	
		Paul	Bechet	paul.bechet76@gmail.com	Listă de comorbidități	Masculin	164 cm	55 kg	40743929800	25 ani	
		Silvia	Manulescu	silviamanulescu62@hotmail.com	Listă de comorbidități	Feminin	164 cm	96 kg	40762440808	60 ani	
		Darius David	Popovici	popovicidd@gmail.com	Listă de comorbidități	Masculin	186 cm	64 kg	40749898131	21 ani	
		Adela Maria	Rosu	adela.maria84@gmail.com	Listă de comorbidități	Feminin	165 cm	56 kg	40757786443	40 ani	

Rows per page: 100 1-6 of 6

Figura 5.5: Listă de pacienți

Pacienții care fac parte din această listă sunt cei care au acceptat cererea de acces trimisă de către medic. Fiecare rând din listă reprezintă intrările unui pacient. Aici se poate observa butonul de ștergere, poza de profil, date personale, attribute, date de contact, iar în ultima coloană o iconiță pentru istoricul medical. Apăsare pe butonul de ștergere duce la eliminarea pacientului din listă și la revenirea în starea inițială, în care medicul nu are acces la datele acestuia. Coloanele pot fi ordonate alfabetic (crescător, descrescător), după valoare sau pot fi aplicate filtre prin câmpurile de căutare.

Istoricul medical devine vizibil atunci când medicul apasă pe iconița din ultima coloană. Apăsând iconița se deschide o fereastră în care se află evenimentele medicale ale pacientului și câmpuri de filtrare după tipul evenimentului, data sau numele medicului.

Sub lista de pacienți este lista de statistici. Datele de intrare ale statisticilor sunt attributele personale ale pacienților. Se observă în figura 5.6 o diagramă radială care arată procentajul din totalul de pacienți care se încadrează într-o anumită categorie.

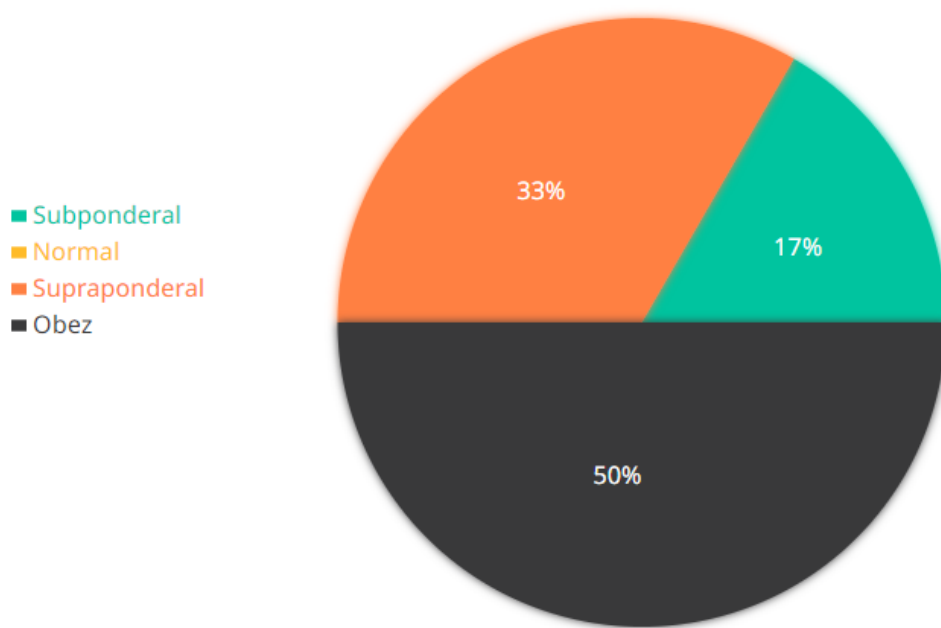


Figura 5.6: Graficul proporțiilor în funcție de indexul de masă corporală

În cadrul platformei există încă trei astfel de diagrame care afișează: repartizarea în funcție de gen, repartizarea în funcție de comorbidități și repartizarea în funcție de vârstă. Am ales să folosesc diagrama radială deoarece este un tip de reprezentare ușor de interpretat. Rolul acestor grafice este pentru vizualizarea medicului, pentru a crea o imagine de ansamblu asupra pacienților săi.

5.2 Dezvoltări ulterioare

Aplicația implementată reprezintă o structură solidă pentru un software medical. Conține facilități de bază, pe care majoritatea sistemelor EMR le implementează, cum ar fi gestionarea pacienților, stocarea istoricului medical și partajarea acestuia. Principiile care stau la baza aplicației sunt confidențialitatea, siguranța, ordonarea și eficiența.

Comparând aplicația MyMedicalRecord cu alte abordări asemănătoare existente, se observă că sistemul implementat oferă mai puține facilități decât acestea. Am identificat câteva funcționalități care sunt posibil de adăugat aplicației, care decid direcția viitoare a acesteia. Am decis să le împart în trei categorii, în funcție de necesitatea lor. Astfel, presupunând că resursele de timp și cost sunt infinite, avem următoarele facilități:

- Facilități de adăugat

Sunt facilități neimplementate în MyMedicalRecord, de care aceasta are nevoie pentru a se ridica la nivelul abordărilor existente.

- Analize
Implementarea unui laborator virtual, prin cadrul căruia medicul transmite rezultatele pacienților. Acesta conține un sistem de alertă, în cazul în care valorile rezultatelor sunt anormale.
- Scanare document
Adăugarea de evenimente prin scanarea cu camera device-ului. Astfel, pacientul nu mai trebuie să completeze manual toate câmpurile pentru un eveniment medical.
- Autenticitate
Dezvoltarea unui sistem de verificare a autenticității. În prezent, orice persoană, fie ca e de specialitate sau nu, poate să declare, la crearea contului că este medic. Sistemul de verificare determină dacă contul creat este autentic. În schimb, pe pagina de profil a medicului va apărea un însemn care să ateste verificarea identității.
- Programări
Crearea unei agende medicale, în care se află programările medicului. Tot ei au posibilitatea de a programa o vizită la cabinet, în armonie cu decizia pacientului.
- Telemedicină
Un sistem de conferințe. Acest sistem va aduce beneficii pacienților care au nevoie de o părere medicală care nu necesită prezența fizică la cabinet.
- Mesagerie
Implementarea funcționalității de mesagerie între medic și pacient, în cadrul căreia cele două părți pot comunica în scris, pot partaja documente și recomandări medicale.
- Facilități de îmbunătățit
 - Grafice
Diversificarea tipurilor de diagrame. În prezent diagramele afișează date din atributele pacienților. Direcția viitoare este de a descrie grafice complexe care arată evoluția pacientului în timp, condițiile medicale la care acesta este predispus și diagrame pentru vizualizarea istoricului medical.
 - Acces pacient
Dezvoltarea sistemului de acces, pentru a permite pacientului să trimită cerere către medic. Această cerere reprezintă dorința pacientului de a face parte din lista medicului și acordul de a partaja datele cu acesta.
- Facilități care aduc un plus în estetică și în experiența cu utilizatorul
- Știri
Secțiune dedicată știrilor. Această parte conține știri și noutăți din lumea medicală, folositoare atât medicului cât și pacientului.
- Sfaturi

Notificări care conțin sfaturi. Prespune implementare funcționalității de sfaturi, care sunt prezentate pacientului, sub formă de notificări. Aplicația privește evoluția pacientului în timp și decide sfaturi relevante pe care să le împărtășească cu pacientul.

- Interfața telefon

Interfață specializată pentru telefon. Odată cu creșterea în funcționalități și în dimensiune, platforma se aglomerează cu informație, aglomerare care nu este prietenoasă cu un device de dimensiuni mici. Astfel, implementarea unei versiuni minimaliste dedicată telefonului, ajută pacienții care doresc să acceseze platforma de pe un device cu ecran mic.

- Postări

Postări în tipul rețelelor de socializare. Presupune funcționalitatea dedicată tuturor utilizatorilor, prin care aceștia pot crea o postare care apare pe ecranele celorlalți utilizatori.

Bibliografie

- [1] <https://redux.js.org/tutorials/essentials/part-1-overview-concepts>
- [2] <https://reactjs.org/docs/thinking-in-react.html>
- [3] <https://reactjs.org/docs/state-and-lifecycle.html>
- [4] <https://reactjs.org/docs/lists-and-keys.html>
- [5] <https://reactjs.org/docs/lifting-state-up.html>
- [6] <https://reactjs.org/docs/getting-started.html>
- [7] <https://reactjs.org/docs/composition-vs-inheritance.html>
- [8] <https://firebase.google.com/docs/build>
- [9] <https://firebase.google.com/docs/release>
- [10] <https://firebase.google.com/docs/engage>
- [11] <https://docs.microsoft.com/en-us/xamarin/xamarin-forms/enterprise-application-patterns/mvvm>
- [12] <https://docs.couchbase.com/tutorials/mobile-travel-tutorial/csharp/develop/mvvm-architecture.html>
- [13] <https://nodejs.org/en/docs/>
- [14] Learning React: Functional Web Development with React and Redux, Alex Banks
Eve Porcello, Mai 2017, O'Reilly Media, Inc.
- [15] The Road to Learn React: Your Journey to Master Plain Yet Pragmatic React.
Js, Robin Wieruch, 1 Iunie 2020, Publicat independent
- [16] Redux in Action, Marc Garreau, Will Faurot, Iunie 2018, O'Reilly Media, Inc.