

Programul de studii:

Informatică aplicată

Lucrare de licență

Titlul lucrării

|  |  |
| --- | --- |
| Autor: | **Diana Andreea Alexandru** |
| Coordonator ştiințific: | **Lect. dr. Vlad Monescu** |

Braşov, 2023

Cuprins

[1.Introducere 4](#_Toc136796839)

[1.1 Scurtă prezentare a temei 4](#_Toc136796840)

[1.2 Motivația alegerii temei 5](#_Toc136796841)

[1.3 Structura lucrării 6](#_Toc136796842)

[2.Tehnologii folosite 7](#_Toc136796843)

[2.1 Flutter 7](#_Toc136796844)

[2.1.1 Dart 8](#_Toc136796845)

[2.2 Firebase 8](#_Toc136796846)

[2.2.1 Firebase Authentication 9](#_Toc136796847)

[2.2.2 Firestore Database 10](#_Toc136796848)

[2.2.3 Storage 12](#_Toc136796849)

[2.3 Pachetul “flutter\_bloc” 13](#_Toc136796850)

[2.4 Biblioteca “GoRouter” 14](#_Toc136796851)

[2.5 Biblioteca “syncfusion\_flutter\_charts” 15](#_Toc136796852)

[3.Structura aplicației și pattern-ul de proiectare BLoC 18](#_Toc136796853)

[3.1 Despre BLoC 18](#_Toc136796854)

[3.2 Structura aplicației 19](#_Toc136796855)

[4. Baza de date 21](#_Toc136796856)

[4.1 Colecția “users” 22](#_Toc136796857)

[4.2 Colecția “doctor” 23](#_Toc136796858)

[4.3 Colecția “Category” 24](#_Toc136796859)

[4.4 Colecția “appointments” 25](#_Toc136796860)

[4.5 Colecția “Reviews” 25](#_Toc136796861)

[5.Prezentarea aplicației 27](#_Toc136796862)

[5.1 Aplicația din perspectiva rolului “User” 27](#_Toc136796863)

[5.1.1 Autetificarea în aplicație 28](#_Toc136796864)

[5.1.2 Creearea contului 29](#_Toc136796865)

[5.1.3 Resetarea parolei 30](#_Toc136796866)

[5.1.4 Pagina de home 31](#_Toc136796867)

[5.1.5 Meniul aplicației 33](#_Toc136796868)

[5.1.6 Pagina Doctors 33](#_Toc136796869)

[5.1.7 Pagina Doctor details 33](#_Toc136796870)

[5.1.8 Rezevarea unei progrmări 35](#_Toc136796871)

[5.1.9 Pagina “My appointments” 36](#_Toc136796872)

[5.1.10 Pagina My Profile 37](#_Toc136796873)

[5.2 Aplicația din pespectiva rolului “Doctor” 38](#_Toc136796874)

[5.2.1 Pagina de autentificare 39](#_Toc136796875)

[5.2.2 Crearea unui cont de doctor 39](#_Toc136796876)

[5.2.3 Resetarea parolei 40](#_Toc136796877)

[5.2.4 Pagina de home 40](#_Toc136796878)

[5.2.5 Meniu 41](#_Toc136796879)

[5.2.6 Pagina Appointments 42](#_Toc136796880)

[5.2.7 Pagina Program 43](#_Toc136796881)

[5.2.8 Pagina My Profile 44](#_Toc136796882)

[5.2.9 Pagina My Doctor Details 44](#_Toc136796883)

[5.2.10 Pagina Reviews 45](#_Toc136796884)

[5.3 Aplicația din perspectiva rolului “Administrator” 45](#_Toc136796885)

[5.3.1 Pagina de autentificare 46](#_Toc136796886)

[5.3.2 Pagina de resetare a parolei 46](#_Toc136796887)

[5.3.3 Meniul aplicației 46](#_Toc136796888)

[5.3.4 Pagina Dashboard- pagina home 47](#_Toc136796889)

[5.3.5 Pagina Category 50](#_Toc136796890)

[5.3.6 Pagina Doctors 51](#_Toc136796891)

[5.3.7 Pagina de adăugare doctor 52](#_Toc136796892)

[5.3.8 Pagina de profil a doctorului 53](#_Toc136796893)

[5.3.9 Pagina programărilor 54](#_Toc136796894)

[5.3.10 Pagina utilizatorilor pacienții 55](#_Toc136796895)

# 1.Introducere

## 1.1 Scurtă prezentare a temei

În această lucrare, am documentat realizare unei aplicații de telefon mobil si web pentru booking medical și administrarea acestuia, crearea aplicației fiind un răspuns la nevoia de digitalizare a sectorului medical. În prezent, programarea întâlnirilor medicale poate fi un proces consumator de timp pentru pacienți și medici, iar administrarea programărilor poate fi o sarcină ineficientă pentru clinici sau spitale.

Prin urmare, aplicația “MediConnect – the all-in-one medical booking and management solution” reprezintă o soluție completa când vorbim despre rezervarea de programari medicale si administrarea acestora. Aplicația are scopul de a simplifica și de a optimiza acest proces. Prin intermediul acesteia, pacienții pot să își facă programările medicale într-un mod rapid și convenabil, evitând astfel timpul petrecut în așteptare la telefon sau la recepție unei clinici medicale. De asemenea, aplicația permite medicilor să își gestioneze mai eficient programul.

În ceea ce privește pagina web de administrare a aplicației, aceasta oferă clinicii o solutie ușorară de a gestiona și de a urmări programările pacienților, dar si alte aspecte administrative inportante in domeniu, si anume statistici care ofera o imagine clara aspupra evulutiei clinici in legatura cu numarul de programari, review-uril ,doctori si categoriile medicale din are acestia fac parte.

## 1.2 Motivația alegerii temei

În ultimii ani, tehnologia a avansat foarte mult și a devenit o parte indispensabilă a vieții noastre, digitalizarea devenind din ce în ce mai importantă în domeniul medical. Sistemul de booking medical este unul dintre acele aspecte ale sănătății care ar putea beneficia în urma digitalizării.

Sistemul de booking medical tradițional este adesea unul ineficient și frustrant pentru pacienți și personalul medical sau administrativ. Pacienții trebuie să aștepte în linii lungi pentru rezervarea unei programări, iar personalul medical poate fi suprasolicitat și poate pierde un timp îndelungat gestionând programările telefonice sau prin intermediul agendelor de hârtie.

Digitalizarea sistemului de booking medical, printr-o aplicație care oferă o soluție completă ar putea aduce numeroase beneficii atât pacienților, cât și personalului unei clinici. În continuare voi prezenta câteva beneficii care m-au motivat in alegerea acestei teme.

O aplicație creată in acest scop ar putea reduce timpul de așteptare pentru a face o programare medicale. Pacienții ar putea să se programeze la medicul dorit direct de pe dispozitivele lor mobile, fără a fi nevoie să facă apeluri telefonice sau să se prezinte personal la clinică.

Digitalizarea unui astfel de sistem ar putea îmbunătăți eficiența și productivitatea personalului clinici. În loc să petreacă timp prețios preluând apeluri telefonice sau actualizând agende de hârtie, personalul ar putea utiliza un sistem de booking digital pentru a gestiona programările. Acest lucru ar putea ajuta la reducerea timpului petrecut în activități administrative și ar putea permite personalului medical să se concentreze mai mult pe furnizarea de îngrijiri de calitate pacienților.

Aplicația propusa ar putea îmbunătăți și experiența pacienților. Aceștia ar putea beneficia de o mai bună vizibilitate asupra disponibilității medicilor, a programelor lor și a review-urilor devenind un proces ușor și rapid, la îndemâna oricărei persoane.

Având în vedere acest context și toate beneficiile mai sus menționate, consider că tema aleasă pentru acestă lucrare este una actuală de interes și reprezintă o soluție sau o îmbunătățire a unei probleme din domeniul medical.

## 1.3 Structura lucrării

Lucrarea este structurată in X capitole care conțin următoarele informații:

Capitolul 1……(De completat)

# 2.Tehnologii folosite

Acest capitol cuprinde o scurtă descriere a tuturor tehnologiilor și instrumentrlor utilizate pentru realizarea acestei lucrări, cuprinzând de asemena și exemple de utilizarea.

## 2.1 Flutter

Flutter este un framework bazat pe limbajul de programare Dart și este construit pentru a permite dezvoltarea de aplicații mobile, web și desktop cu o singură bază de cod. Acest lucru este posibil datorită modului în care Flutter utilizează widget-uri pentru a crea interfețe de utilizator, care pot fi personalizate și reutilizate pe toate platformele.

Flutter oferă, de asemenea, suport pentru diferite platforme, cum ar fi iOS, Android, Windows, macOS și web, ceea ce face dezvoltarea multiplatformă mult mai ușoară și mai accesibilă.

Flutter utilizează un motor de randare personalizat numit Flutter Engine, care permite crearea de interfețe grafice rapide și reponsive. Acesta oferă un set bogat de widget-uri și biblioteci care pot fi utilizate pentru a construi interfețe complexe și a oferi o experiență vizuală plăcută utilizatorilor. Flutter facilitează, de asemenea, animațiile și efectele vizuale fluide, datorită arhitecturii sale reactive.

Dezvoltarea în Flutter este susținută de un set puternic de instrumente și biblioteci, inclusiv un editor de cod dedicat numit Flutter SDK, care oferă suport complet pentru dezvoltarea și depanarea aplicațiilor.

Un alt aspect important al Flutter este hot reload, care permite dezvoltatorilor să vadă imediat efectele modificărilor făcute în codul sursă. Aceasta facilitează procesul de dezvoltare, deoarece permite utilizatorilor să testeze rapid idei și să găsească soluții pentru problemele întâmpinate.

### 2.1.1 Dart

Dart este un limbaj de programare modern dezvoltat de către Google, care a fost lansat pentru prima dată în 2011. Acesta este un limbaj de programare orientat către obiecte, care a fost proiectat să fie ușor de învățat și de utilizat pentru dezvoltatorii de aplicații.

Dart poate fi utilizat pentru dezvoltarea de aplicații mobile, aplicații web și aplicații desktop. Acesta are o bibliotecă standard extinsă, care oferă multe funcții utile pentru dezvoltatori, cum ar fi funcții pentru lucrul cu fișiere, codificarea și decodificarea datelor și multe altele.

O altă caracteristică importantă a limbajului Dart este suportul pentru programarea asincronă. Acest lucru permite creearea de aplicații care pot gestiona mai multe sarcini în același timp, fără a bloca procesul principal de execuție.

Acest limbaj rulează pe propriul său motor de execuție (runtime) numit Dart VM, care oferă performanță excelentă.

Dart este folosit în special pentru dezvoltarea de aplicații mobile Flutter, acesta utilizând Dart ca limbaj de programare principal, ceea ce face integrarea foarte ușoară.

## 2.2 Firebase

Firebase este o platformă de dezvoltare a aplicațiilor mobile și web dezvoltată de Google. Aceasta oferă o serie de servicii și funcții bazate pe cloud, pentru a ajuta la construirea aplicaților, inclusiv servicii de autentificare, baze de date în timp real, stocare în cloud, mesagerie, analize și multe altele.

Firebase oferă o interfață de utilizare simplă, care face ca integrarea cu această platformă să fie ușoară și accesibilă pentru dezvoltatorii de toate nivelurile de experiență. De asemenea, aceasta oferă suport pentru o serie de limbaje de programare, inclusiv Java, Swift, Objective-C, JavaScript, Node.js și multe altele.

Firebase oferă un pachet de instrumente gratuite, fapt care permite folosirea platformei fără costuri inițiale. Serviciile sunt gratuite in limite foarte generoase, dar există și pachete plătite, care oferă funcții și capacitate suplimentare pentru aplicații mai mari.

Datorită facilităților sale de scalabilitate și serviciilor bazate pe cloud, Firebase este o platformă populară pentru cei care doresc să construiască aplicații puternice și scalabile într-un mod eficient și ușor de gestionat.

În contiunuare voi descrie principalele service din Firebase utilizate pentru realizarea acestei lucrari.

### 2.2.1 Firebase Authentication

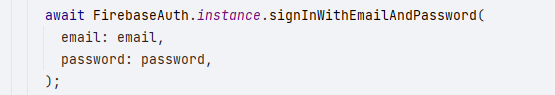
Firebase Authentication este un serviciu puternic care simplifică procesul de autentificare a utilizatorilor în aplicații. Cu opțiuni variate de autentificare, precum email, număr de telefon și rețele sociale (Google, Facebook, Twitter, etc.), oferă flexibilitate utilizatorilor. Integrarea ușoară cu alte servicii Firebase permite accesul la informațiile utilizatorilor și personalizarea experiențelor.

De asemenea, oferă funcționalități de gestionare a conturilor, inclusiv creare, ștergere și actualizare. Acest serviciu utilizează proceduri de verificare a autentificării cu OpenID Connect și OAuth 2.0 în ceea ce privește identificările la nivel de server.

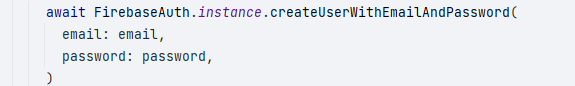
Firebase Authentication reprezintă o soluție sigură, flexibilă și ușor de utilizat pentru autentificarea utilizatorilor în aplicații.

În aplicație am ales să folosesc autentificare cu email și parola. În continuare voi enumera cu ajutorur unor exemple câteva metodele pe care le-am utilizat în aplicație din serviciul Firebase Authentication:

* **Autentificare:**

****

* **Înregistrare/Creare cont:**

****

* **Resetare parolă:**

****

* **Identificarea utilizatorului current:**



* **Actualizarea emailului:**

await user?.updateEmail(email);

* **Deconectare:**

****

### 2.2.2 Firestore Database

Firestore Database este o bază de date NoSQL document-based, scalabilă și puternică, care face parte din platforma Firebase dezvoltată de Google. Aceasta oferă utilizatorilor o soluție flexibilă pentru stocarea, interogarea și sincronizarea datelor în timp real pe platforme web, mobile și server.

Firestore organizează datele în colecții și documente, permițând stocarea de date structurate într-un format ierarhic. Fiecare document conține câmpuri și poate găzdui structuri de date complexe precum tablouri și obiecte imbricate. Această flexibilitate permite un model de date eficient și recuperarea ușoară a informațiilor.

Firestore oferă o consistență și fiabilitate puternice pentru date, asigurând stocarea securizată și redundantă a informațiilor. Acesta furnizează reguli de securitate încorporate pentru a defini controale de acces și permisiuni granulare, protejând datele sensibile de accesul neautorizat.

În continuare voi prezenta câteva exemple de utilizarea a metodelor disponibile din serviciul Firestore Database:

* Extrage din baza de date a documentelor unei colecții:

QuerySnapshot appointmentsSnapshot =

await FirebaseFirestore.instance.collection('appointments').get();

”QuerySnapshot” este o variabilă care stochează toate documentele extrase în urma operației de obținere a datelor, “FirebaseFirestore.instance” accesează instanța Firestore a serviciului Firebase, “.collection('appointments')” specifică colecția din baza de date Firestore pe care dorim să o accesăm, “.get()” execută operația de obținere a datelor din colecția specificată anterior. Prin urmare, cu ajutorul secvenței mai sus menționate, în “appointmentShnapshot” se stochează toate documentele din colecția appointment.

* Ștergerea unui document dintr-o colecție:

await FirebaseFirestore.instance.collection('appointments')

.doc(appointmentId)

.delete();

Cu ajutorul acesteia se realizează, in baza de date, ștergerea documentului cu id-ul/numele documentului ” appointmentId”din colecția ”appointments”.

* Adăugarea unui document într-o colecție:

final Map<String, dynamic> review = {

"doctor\_id": doctorId,

"patient\_id": userId,

"date\_and\_time": DateTime.now().toString(),

"comment": comment,

"stars": stars,

};

await FirebaseFirestore.instance.collection("reviews").add(review);

Această secvență de cod adaugă in colecția ”reviews” un nou document cu cămpurile declarate cu ajutorul map-ului review. Nu am specificat ce nume doresc să primească documentul, acesta va fi generat la persistarea datelor.

* Actualizarea unui document:

final data = {"confirmed": true};

await FirebaseFirestore.instance .collection("appointments") .doc(appointmentId)

.set(data, SetOptions(merge: true));

Cu ajutorul acetui cod documentul cu id “appointmentId” din colecția ”appointments” se va actualiza cu datele stocate in map-ul ”data”, astfel câmpul bulean ”confirmed” se va actualiza cu valoarea ”true”.

### 2.2.3 Storage

Firebase Storage este un serviciu de stocare în cloud oferit de platforma Firebase. Aceasta permite stocharea și gestionarea de fișiere, cum ar fi imagini, videoclipuri, fișiere audio și altele, în mod scalabil și securizat.

Cu Firebase Storage, se pot încărca și descărca fișiere cu ușurință folosind o gamă variată de metode, inclusiv SDK-uri pentru diverse platforme. Fișierele sunt stocate în cloud și pot fi accesate și partajate în mod eficient în aplicații web și mobile.

Firebase Storage oferă funcționalități puternice pentru gestionarea și manipularea fișierelor. Utilizatorii acestui serviciu pot crea reguli personalizate pentru controlul accesului, pot genera linkuri de partajare și pot efectua transformări asupra fișierelor, precum redimensionarea și comprimarea.

În plus, Firebase Storage asigură redundanță și durabilitate pentru date, astfel încât fișierele sunt protejate împotriva pierderii sau coruperii. Serviciul oferă, de asemenea, suport pentru transferuri de fișiere rapide și eficiente, asigurând o experiență rapidă și performantă pentru utilizatori.

În continoare voi evidenția și voi detalia un exemple de utilizarea a serviciului storage in aplicatia flutter.

UploadTask uploadTask;

Reference ref = FirebaseStorage.instance.ref().child('doctors').child('/$selctFile');

final metadata = SettableMetadata(contentType: 'image/png');

uploadTask = ref.putData(selectedImageInBytes!, metadata);

Cu ajutorul acestei secvențe de cod adăugăm o imagine în containerul Storage creat în serviciul Firebase Storage. Voi explica fiecare line:

* “UploadTask uploadTask;”: Această linie declară o variabilă de tip "UploadTask" pentru a stoca referința către sarcina de încărcare a fișierului în Firebase Storage. Obiectul “UploadTask” oferă metode și proprietăți pentru monitorizarea progresului încărcării, adăugarea de ascultători pentru evenimente specifice încărcării, obținerea informațiilor despre starea curentă a încărcării.
* “Reference ref = FirebaseStorage.instance.ref().child('doctors').child('/$selctFile');”: Aici se definește o referință către locația în care doresc să încărc fișierul. În acest exemplu, accesez containerul "doctors" și se specific numele fișierului utilizând variabila "selctFile".
* “final metadata = SettableMetadata(contentType: 'image/png');”: Se creează un obiect de metadate care specifică tipul de conținut al fișierului, "image/png".
* “uploadTask = ref.putData(selectedImageInBytes!, metadata);”: Această linie inițiază procesul de încărcare a datelor în locația specificată prin referința "ref". "selectedImageInBytes" reprezintă datele imaginii selectate și "metadata" sunt metadatele asociate cu imaginea. Rezultatul acestei operații este stocat în variabila "uploadTask".

## 2.3 Pachetul “flutter\_bloc”

Flutter\_bloc este un pachet Flutter care facilitează implementarea arhitecturii BLoC (Business Logic Component) în aplicațiile Flutter. Acest pachet oferă un set de clase și metode pentru a creea și gestiona blocuri, evenimente și stări într-un mod eficient și ușor de utilizat.

Principala clasă din pachetul flutter\_bloc este Bloc. Aceasta reprezintă un bloc care se ocupă de gestionarea stării și a evenimentelor în aplicație. Pentru a crea un bloc, se poate extinde clasa Bloc și se pot defini evenimente și stări personalizate. Blocul primește evenimente și produce stări în funcție de logica specifică implementată în metodele sale.

Un alt element important în pachetul flutter\_bloc este clasa BlocBuilder. Aceasta este utilizată pentru a construi interfața utilizatorului (UI) în funcție de stările emise de bloc. Prin utilizarea BlocBuilder, se poate asocia un bloc și o funcție de construire a UI-ului, care va fi apelată de fiecare dată când blocul emite o stare nouă. Astfel, interfața utilizatorului se actualizează automat în funcție de schimbările de stare.

Pachetul flutter\_bloc oferă și alte clase și funcții utile pentru a lucra cu arhitectura BLoC, cum ar fi BlocProvider, BlocListener și MultiBlocProvider. Acestea facilitează gestionarea și organizarea blocurilor în aplicație și asigură un flux unidirecțional coerent al datelor.

Unul dintre avantajele principale ale pachetului flutter\_bloc este ușurința cu care poate fi implementată arhitectura BLoC în aplicațiile Flutter. Pachetul oferă un set de abstracții și un flux clar de gestionare a stării, ceea ce facilitează separarea logicii de componentele UI și simplifică testarea și reutilizarea codului.

De asemenea, pachetul flutter\_bloc oferă o integrare facilă cu alte pachete și librării Flutter, cum ar fi rxdart pentru gestionarea fluxurilor de date și equatable pentru compararea obiectelor în mod eficient.

Am ales să folosesc în aplicație “flutter\_bloc” deoarece este un pachet puternic și ușor de utilizat pentru implementarea arhitecturii BLoC în aplicațiile Flutter. Prin utilizarea acestui pachet, se pot crea aplicații cu o arhitectură solidă, modulară și ușor de testat, care separă clar logica de componentele UI.

## 2.4 Biblioteca “GoRouter”

GoRouter este o bibliotecă pentru gestionarea navigării în aplicații Flutter. Aceasta oferă o abordare simplă și elegantă pentru a defini rute și a gestiona tranzițiile între ecranele aplicației. GoRouter este dezvoltată de echipa Gothinkster și este inspirată de biblioteca de navigare a framework-ului Angular din JavaScript.

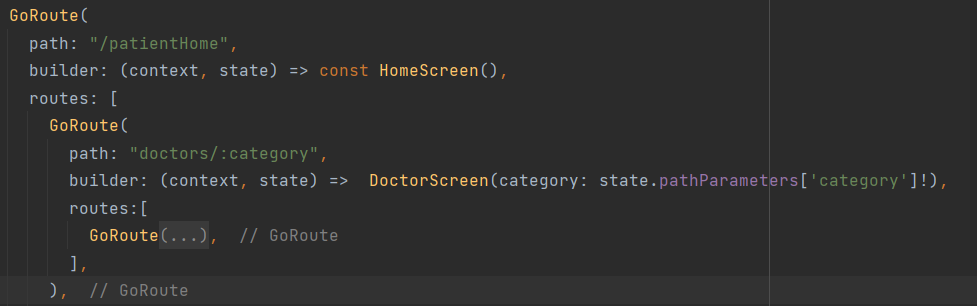
Principala caracteristică a GoRouter este definirea rutelor folosind obiecte de tip GoRoute. Un obiect GoRoute conține informații despre numele rutei, ecranul corespunzător și orice alți parametri necesari pentru navigare.

Utilizarea bibliotecii GoRouter în dezvoltarea aplicațiilor Flutter aduce mai multe avantaje:

* Gestionarea simplă a navigării: GoRouter oferă un mod simplu și declarativ de a defini rute și de a gestiona navigarea între ecrane. Aceasta facilitează dezvoltarea unei logici de navigare coerente și ușor de întreținut în aplicații Flutter.
* Flexibilitate în definirea rutelor: GoRouter permite definirea rutelor cu parametri variabili, care facilitează trimiterea și recepționarea datelor între ecrane. De asemenea, se pot defini ierarhii complexe de rute pentru a reflecta structura navigării aplicației.
* Compatibilitate cu navigator-ul Flutter: GoRouter este compatibil cu navigator-ul Flutter, ceea ce înseamnă că se pot utiliza în continuare funcționalitățile native ale navigator-ului atunci când este necesar. Se pot combina navigarea gestionată de GoRouter cu navigarea clasică a navigator-ului, oferind flexibilitatea de a adapta biblioteca la nevoile și preferințele utilizatorului.
* Gestionarea istoricului de navigare: GoRouter asigură o gestionare corectă a istoricului de navigare. Utilizatorii pot reveni la ecranele anterioare folosind butonul de revenire al dispozitivului sau gesturile de navigare specifice platformei. GoRouter se ocupă de actualizarea corectă a istoricului și a stării aplicației pentru a asigura o experiență de navigare fluidă.

În continuare voi prezenta un exemplu de utilizare în cadrul aplicației.

Definirea unor rute cu GoRouter:



În acest exemplu, ruta "/patientHome" este definită prin intermediul obiectului GoRoute. Acesta specifică faptul că ruta corespunde ecranului HomeScreen. În plus, GoRouter oferă flexibilitate în gestionarea parametrilor și opțiunilor de navigare pentru fiecare rută. Tot in imaginea 2.() este prezent un explemplu de definire a unei rute, copil pentru ruta principală “/patientHome”. Calea acestei rute copil este "doctors/:category", unde ":category" este un parametru variabil în cale. De exemplu, calea "doctors/cardiology" ar corespunde categoriei "cardiology". Atunci când această rută este accesată, se va construi un DoctorScreen, iar valoarea parametrului "category" va fi extrasă din state.pathParameters și transmisă la DoctorScreen.

Pentru a utiliza GoRouter în aplicația Flutter, trebuie definită o instanță a clasei GoRouter și înregistrate rutele pe care dorim să le utilizăm. În cadrul aplicației am definit două instanțe a clasei GoRoute, una pentru partea web- “\_routerForWeb” și una pentru partea de aplicație pentru telefonul mobil - “\_routerForMobile”.

## 2.5 Biblioteca “syncfusion\_flutter\_charts”

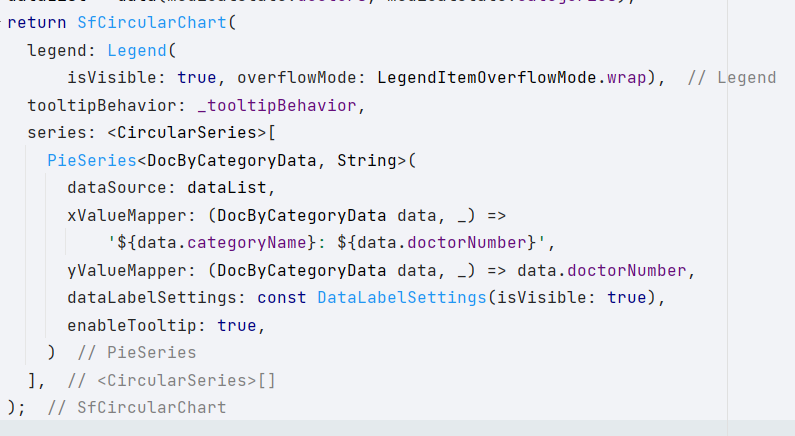
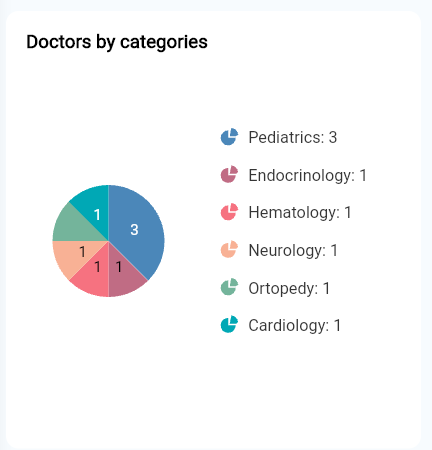
Syncfusion Flutter Charts este o bibliotecă puternică și versatilă pentru crearea de grafice interactive în aplicații Flutter. Această bibliotecă oferă o gamă largă de tipuri de grafice și funcționalități avansate, permițând utilizatorilor să creeze grafice cu aspect profesional și să le personalizeze în funcție de nevoile lor.

O caracteristică-cheie a Syncfusion Flutter Charts este gama sa diversă de tipuri de grafice. Aceasta include grafice de linii, grafice cu bare, grafice cu sectoare, grafice cu histograme și multe altele. Această varietate de tipuri de grafice permite selectarea cel uimai potrivit tip de grafic pentru datele lor și scopul lor specific.

Pe lângă tipurile de grafice, Syncfusion Flutter Charts oferă și funcționalități avansate de interactivitate și animație. Graficele create cu această bibliotecă pot fi interactivitate, permițând utilizatorilor să interacționeze cu graficele prin gesturi precum atingere, glisare sau zoom. Aceasta adaugă un nivel suplimentar de angajament și experiență utilizatorului.

Syncfusion Flutter Charts oferă, de asemenea, suport pentru interacțiunea cu datele. Dezvoltatorii pot adăuga marcaje, adnotări și indicatori pentru a evidenția anumite puncte de date sau pentru a furniza informații suplimentare.

În cadrul aplicațieiei, am folosit această bibliotecă pentru majoritata graficelor, aducănd un real beneficiu în realizarea acestora. În continuare voi descrie realizare unei digrame cu ajutorul acestei biblioteci.

Codul reprezintă crearea unui grafic circular (pie chart) utilizând biblioteca Syncfusion Flutter Charts.

* “SfCircularChart” reprezintă clasa principală pentru a crea un grafic circular folosind Syncfusion Flutter Charts. Este utilizată pentru a crea o instanță a graficului circular.
* “legend” este o proprietate a graficului care controlează afișarea legendei.
* “tooltipBehavior” este o proprietate a graficului care controlează comportamentul sugestiilor (tooltips) atunci când utilizatorul interacționează cu graficul. În exemplul dat, \_tooltipBehavior este o variabilă definită anterior.
* “series” reprezintă o listă de obiecte CircularSeries, care definesc seriile de date ale graficului circular.
* “PieSeries” reprezintă o clasă care definește o serie de date de tip pie (segmente circulare) în graficul circular. În exemplul dat, “dataSource” reprezintă sursa de date utilizată pentru a genera graficul. “xValueMapper” și “yValueMapper” sunt funcții de mapare care extrag valorile x și y din obiectele de date. În acest caz, se utilizează obiecte de tip “DocByCategoryData”. Funcția “xValueMapper” concatenează “categoryName” și “doctorNumber” pentru a forma etichetele x. Funcția “yValueMapper” extrage “doctorNumber” ca valoarea y. “dataLabelSettings” controlează configurarea etichetelor de date afișate pe grafic, iar “enableTooltip” permite afișarea sugestiilor atunci când utilizatorul interacționează cu segmentele graficului.

Astfel, codul prezentat definește crearea unui grafic circular personalizat cu legenda vizibilă, sugestii afișate și etichete de date configurate. Sursa de date este furnizată prin lista dataList, iar valorile x și y sunt extrase din obiectele de date folosind funcțiile de mapare specificate.

# 3.Structura aplicației și pattern-ul de proiectare BLoC

## 3.1 Despre BLoC

BLoC (Business Logic Component) este un pattern de proiectare utilizat în dezvoltarea aplicațiilor mobile și web, care facilitează gestionarea stării și fluxului de date într-o manieră reactivă. BLoC este un acronim pentru Business Logic Component, iar conceptul a devenit popular în comunitatea dezvoltatorilor de aplicații Flutter, dezvoltată de Google.

BLoC se bazează pe separarea clară a logicii de afaceri și a stării aplicației de interfața utilizator. Ideea principală este de a avea un component care gestionează logica de afaceri și comunică cu interfața utilizator prin intermediul unui sistem de flux de date bidirecțional.

Într-un design BLoC tipic, există trei componente principale:

* **Bloc**: Este componenta centrală a pattern-ului. Un BLoC conține logica de afaceri și gestionează starea aplicației. Acesta primește evenimente de la interfața utilizator și poate produce schimbări în starea aplicației. Blocul este izolat de interfața utilizator și poate fi testat independent.
* **Evenimente**: Evenimentele reprezintă acțiunile sau schimbările care au loc în interfața utilizator. Acestea pot fi, de exemplu, acțiuni ale utilizatorului sau evenimente declanșate de sistem. Evenimentele sunt trimise către BLoC pentru a iniția modificări în starea aplicației.
* **Stare**: Starea reprezintă starea curentă a aplicației. Este actualizată de către BLoC în funcție de evenimentele primite. Starea poate fi reprezentată de obiecte sau clase care conțin informații relevante despre starea aplicației. Interfața utilizator poate să se aboneze la modificările stării și să se actualizeze în consecință.

Comunicarea între componentele BLoC se realizează prin intermediul fluxului de date. Fluxul de date este o cale bidirecțională de transfer a evenimentelor și stării între interfața utilizator și BLoC. Astfel, interfața utilizator poate iniția evenimente, iar BLoC procesează aceste evenimente și actualizează starea aplicației. La rândul său, interfața utilizator se actualizează în funcție de modificările stării.

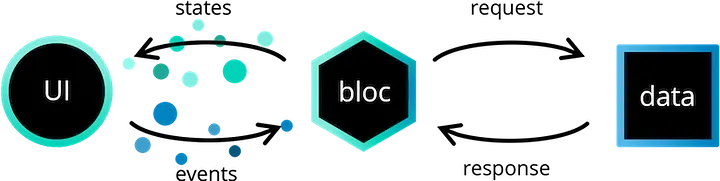


Figura 3.1 –

În figura 3.1 se poate observa modul de funcționare al unui proiect care folosește pattern-ul de proiectare BLoC. La acționare unor componente din interfața utilizatorului, spre exemplu apăsarea unor butoane sau navigarea între pagini, se declanșează un eveniment care informează blocul cu privire la informațiile cerute de interfață. Mai departe, blocul va solicita aceste date,în unele cazuri cu ajutorul unui repository care comunică cu baza de date a aplicației. La primirea datelor solicitate Blocul decide daca aceste date au fost găsite cu succes sau s-a întâmpinat o eroare și emite o stare. Interfața asculta starea primită de la bloc și se actualizează în funcție de informațiile primire.

Folosirea design pattern-ului BLoC în aplicații Flutter vine cu mai multe avantaje:

* Separarea logică a afacerii de interfața utilizator (UI): BLoC promovează separarea clară a logicii de afaceri de partea de interfață utilizator. Aceasta facilitează gestionarea și organizarea codului, făcându-l mai ușor de înțeles, testat și întreținut.
* Reutilizarea codului: BLoC permite reutilizarea logicilor de afaceri în mai multe locuri ale aplicației. Deoarece BLoC-urile sunt independente de interfața utilizator, pot fi utilizate în mai multe ecrane sau componente, ceea ce duce la un cod mai modular și mai ușor de gestionat.
* Testabilitate îmbunătățită: BLoC facilitează testarea unitară a logicii de afaceri. Prin izolarea BLoC-ului de interacțiunea cu interfața utilizator, este mai ușor să se creeze teste independente, care să verifice comportamentul BLoC-ului în diferite scenarii și evenimente.
* Reactivitate: BLoC folosește fluxuri de date pentru a comunica între componentele logicii de afaceri și interfața utilizator. Acest aspect reactiv permite actualizarea automată a interfeței utilizator în funcție de schimbările din starea aplicației. Astfel, interfața utilizator este mereu sincronizată cu logica.

## 3.2 Structura aplicației

Pentru structura aplicației am ales să folosesc următoarele componente pe care le voi detalia în acest subcapitol: BLoC (Business Logic Component), repository, models, entities și screens. Această arhitectură împarte responsabilitățile aplicației în diferite straturi, pentru a menține o separare clară între logica aplicației, interacțiunea cu baza de date sau alte servicii externe și prezentarea datelor către utilizator.

BLoC (Business Logic Component): BLoC este responsabil de gestionarea logicii și de gestionarea stării. Este un strat intermediar între interfața utilizatorului și resursele de date. BLoC primește evenimente de la utilizator și produce stări corespunzătoare. În cadrul aplicației am folosit doua structuri bloc: “auth” care conține clasele “auth\_bloc.dart”, “auth\_event.dart”, “auth\_state.dart” și “medical” care conține clasele “medical\_bloc.dart”, “medical\_event.dart”, “medical\_state.dart”.

Blocul “auth” este responsabil pentru gestionare datelor și acțiunilor legate de autentificarea utilizatorulrului. Cu ajutorul acestui bloc gestionez în aplicație, autentificarea și înregistrarea unui utilizator, resetarea parolei, deconectarea din cont și obținerea datelor despre utilizatorul curent al aplicației, date indispensabile pentru determinearea rolului utilizatorurului și multe altele. Acest bloc folosește metode definite cu ajutorul repository-ului “auth\_reporitory.dart” metode care facilitează obținerea, adăugarea, stergerea și actualizarea datelor reținute în Firebase, atât în serviciul de autentificare Firebase Authentification cât și în baza de date nonSql Firestore Database.

Blocul “medical” este resonsabil pentru gestionarea acțiunilor desfășurate în aplicație cu excepția celor legate de serviciul de autentificare. Acest bloc se ocupă de manipularea informațiilor legate de utilizatori, doctori,program, programări, categorii și review-uri. Pentru a interacționa cu resursele de date, blocul "medical" folosește metode definite în repository-ul "medical\_repository.dart". Acest repository facilitează obținerea, adăugarea, ștergerea și actualizarea datelor stocate în bază de date din Firestore Database asociată aplicației.

Repository-ul servește ca un strat intermediar între BLoC și baza de date. Scopul principal al repository-ului este de a abstractiza date și de a oferi o interfață simplă pentru BLoC pentru a le accesa și a le modifica. Repository-ul definește metode pentru a realiza operațiile CRUD (Create, Read, Update, Delete) asupra datelor. Include metode care sunt apelate de către BLoC în funcție de necesități. Aceasta facilitează izolarea logicii de implementarea concretă a surselor de date, permițând schimbarea sau înlocuirea surselor fără a afecta logica aplicației. Astfel, dacă se dorește schimbare surselor de date cu altă platormă de servicii acest lucru nu influențează logica aplicației, trebuie modifcare doar repository-urile și enitătile. În cadrul aplicației există doua repository pe care le-am menționat anterior: “auth\_repository.dart” și “medical\_repository.dart”.

Entitățile reprezintă obiectele fundamentale, informațiile brute care sunt stocate în baza de date sau utilizate în serviciile externe. Acestea sunt transformate în modele pentru a fi utilizate în BLoC.

Modelele reprezintă structurile de date care sunt utilizate în aplicație. Acestea sunt create cu entitățile de bază ale aplicației și descriu datele și comportamentul acestora. Modelele abstractizează și encapsulează datele și comportamentul asociat acestora într-o formă mai simplă și mai ușor de utilizat. Acestea sunt de obicei structuri de date simple și independente de implementarea surselor de date reale. Modelele sunt utilizate în BLoC pentru a stoca și manipula date temporar, precum și pentru a facilita transferul de date între diferite componente ale aplicației. Acestea pot include doar anumite proprietăți relevante sau pot transforma și formata datele într-un format corespunzător. Modelele pot fi adaptate pentru a răspunde cerințelor logicii de afaceri și interfeței utilizatorului, fără a afecta structura sau implementarea entităților brute.

Utilizarea entităților și a modelelor în același timp permite o separare clară între datele brute și datele adaptate/logica de afaceri. Entitățile sunt utilizate pentru a interacționa cu datele reale și pentru a asigura integritatea și eficiența în interacțiunea cu sursa de date, în timp ce modelele abstractizează și encapsulează datele pentru a fi utilizate în logica de afaceri și în interfața utilizatorului.

Transformarea entităților în modele se ralizează, în acestă aplicație, în metodele implenentate în repository-uri.

În acestă aplicație interfața utilizatorului este implementată prin intermediul screenurilor sau paginilor. Screenurile reprezintă componentele vizuale cu care utilizatorul interacționează direct. Acestea afișează datele și stările furnizate de BLoC și permit utilizatorului să interacționeze cu aplicația prin intermediul acțiunilor precum apăsarea unui buton sau completarea unui formular. În acest contextul, screenurile pot fi implementate ca widget-uri care sunt afișate pe ecranul dispozitivului. Acestea pot fi componente complexe care conțin alte widget-uri încorporate și sunt responsabile de prezentarea datelor, gestionarea evenimentelor și comunicarea cu BLoC-ul pentru a actualiza starea aplicației.

# 4. Baza de date

Soluția folososită în aplicație pentru stocarea și administrarea datelor este Firestore Database. Acest serviciu este o bază de date NoSQL gestionată de Google, care oferă o soluție scalabilă și ușor de utilizat pentru stocarea datelor în timp real. Este parte a platformei Firebase și este construit pe arhitectura Google Cloud, oferind performanțe ridicate și scalabilitate în timp real. Această soluție este gratuită în limite foarte generoase fiind potrivită pentru o scară largă de aplicații web sau de mobil.

(\* figura cu imagine din aplicație cu legătura la costul detelor prezente)

Această baza de date cuprinde mai multe colecții fiecare dintre acestea având numeroase documente unice. Unicitatea documentului este asigurata de ID-ul documentului auto-generat.

Baza de date folosită cuprinde următoarele colecții și subcolecții:

* Users
* Doctor (conținând subcolecția Program)
* Category
* Appointments
* Reviews

**Detalierea colecțiilor**

## 4.1 Colecția “users”

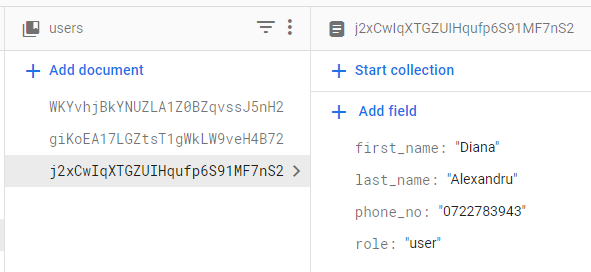
****

Figura - Baza de date: users

În această colecție se salvează date funizate de către utilizatori la momentul creări contului, dar și rolul acestuia în aplicație, ulterior toate acestea putând fi modificate de utilizatorlul cu rol user sau rol administrator după caz. Informațiile reținute cuprind următoarele câmpuri obligatorii:

* **Document-ID** (User UID-ul generat la crearea contului de seviciul Autentification din Firebase)
* **first\_name** (prenumele utilizatorului)
* **last\_name** (numele utilizatorului)
* **phone\_no** (numărul de telefon al utilizatorului)
* **role** (poate fi “user”- rolul implicit la crearea contului specific pentru utilizatorii pacienți, “doctor”- rolul primit de un utilizator doctor la crearea contului de către administrator sau la modificarea acestuia, “administator”- administratorul/persoana desemnată de clinica pentru utilizarea aplicației de administrare)

## 4.2 Colecția “doctor”

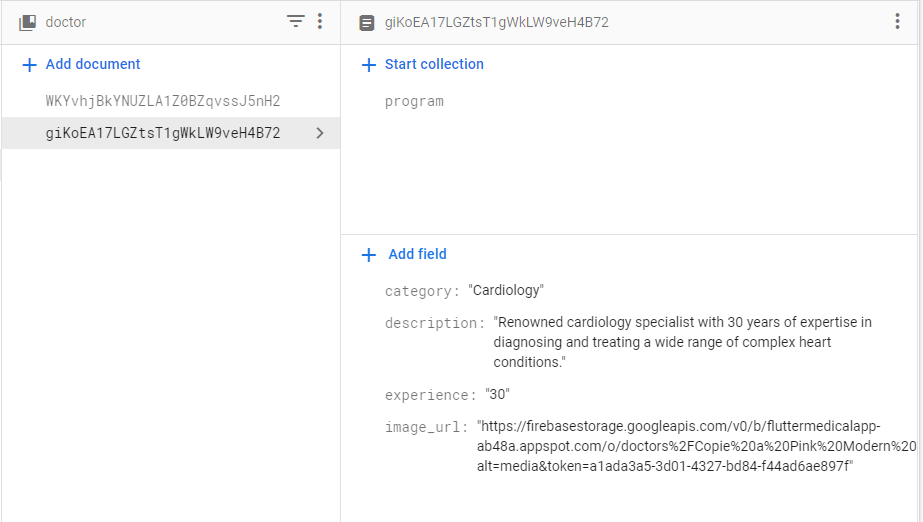
****

Figura -Baza de date: colecția doctor

Aceasta are rol de a reține detaliile specifice unui doctor, detalii suplimentare fața de informatiile unui utilizator pacient.

Un document de tip doctor conține urmatoarele informații:

* **Document-ID** (ID-ul documentului reprezentativ contului specific utilizatorului cu rol doctor din colecția user)
* **category** (numele categoriei din care face parte specializarea doctorului ex: “Cardiology”, nume care apartine unui document din colecția Category)
* **description** (descriera doctorului)
* **experience** (anii de experiența ai doctorului)
* **image\_url** (adresa URL a pozei doctorului, adresa stocată prin intermediul serviciului Storage din Firebase)
* **program – subcolecție a colecției doctor**

Aceasta contine documente care reprezinta programul de lucru al doctorului adica locurile pentru programarile care pot fi rezervate de pacienți ulterior. Fiecare document conține urmatoarele câmpuri:

* + **day\_of\_week** (numarul zilei din saptamana in care doctorul are program)
  + **start\_hour** (ora de inceput a unei programri)
  + **end\_hour** (ora de final a unei programari)

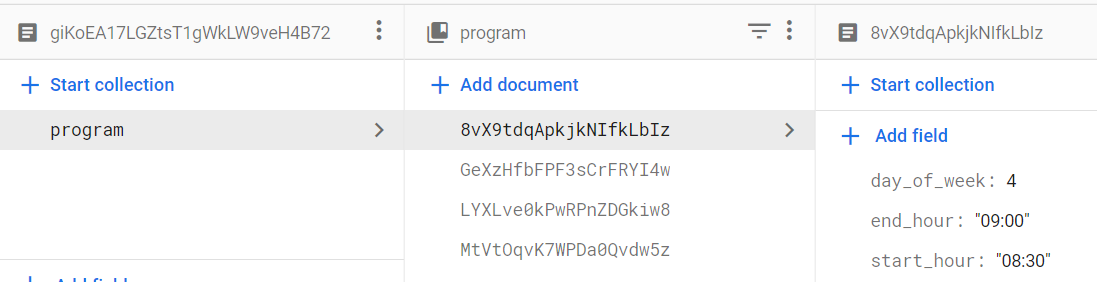


Figura – Baza de date : Subcolecția program

## 4.3 Colecția “Category”

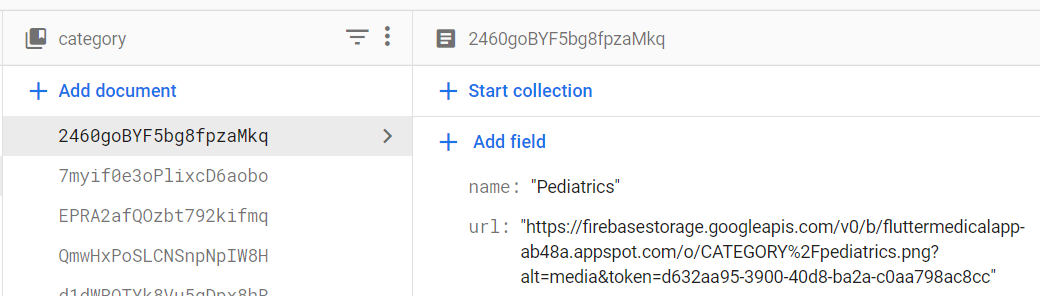


Figura – Baza de date: colecția Category

Colecția Category conține documente care reprezintă categoriile sau specializarile din care fac parte doctorii. Fiecarui doctor îi este atribuită o categorie, acesta ajuta atat la filtrare doctorilor in aplicație cât si la o imagine mai clara asupra detaliilor despre doctor.

Fiecare document conține un **Document ID** – unic și auto-generat și următoarele câmpuri:

* **name** (numele categoriei)
* **url** (adresa URL a imaginii stocate cu ajutorul serviciului Storage din Firebase, imagine reprezentativă pentru fiecare categorie.

## 4.4 Colecția “appointments”

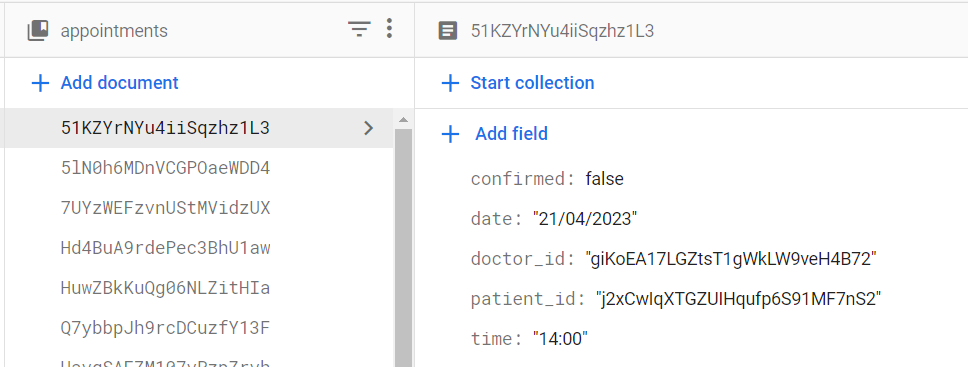
****

Figura – Baza de date: Colecția Appointments

Această colecție cuprinde totalitatea programărilor care sunt rezervate de catre pacienți. Fiecare document reprezintă o programare si conține un **Document ID** unic și urmatoarele câmpuri:

* **doctor\_id** (ID-ul doctorului va efectua progrmarea)
* **patient\_id** ( ID-ul utilizatorului pacient cu rol user care a rezervat programarea)
* **date** (data la care se efectueaza programarea)
* **time** (ora la care se efectueaza programarea)
* **confirmed** (câmp de tip bulean cu urmatoarele semnificați **true**- pacientul s-a prezentat la programarea solicitata, informatie confirmata de catre utilizatorul cu rol doctor sau cu rol administrator, **false** – o programare viitoare sau o programare trecuta neconfirmata de către utilizatorii menționați mai sus)

## 4.5 Colecția “Reviews”

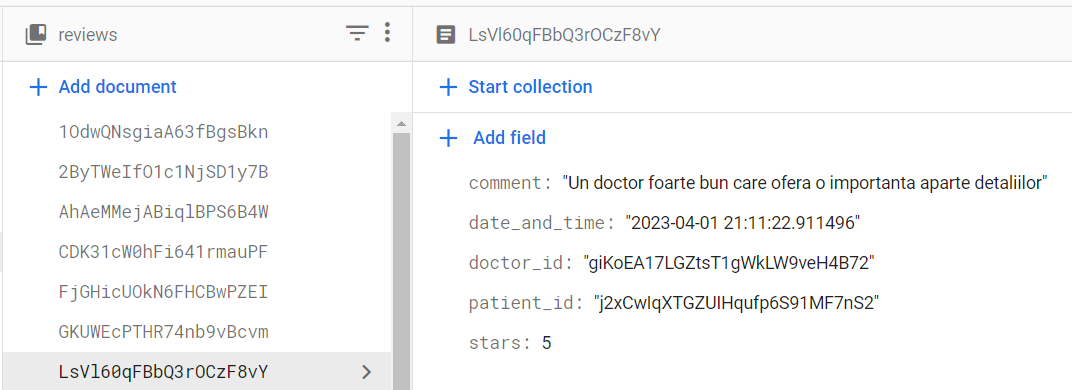
****

Figura – Baza de date: Colecția Review

Aceasta colecție cuprinde toate review-urile tuturor medicilor. Fiecare document reprezintă un review oferit de un utilizator pacient care a avut cel puțin o programare (un document in colecția Appointments ) confirmată. Fiecare document conține un **Document ID** unic și următoarele câmpuri:

* **doctor\_id** ( ID-ul doctorului căruia îi este oferit review-ul )
* **patient\_id** ( ID-ul utilizatorului pacient care care a oferit review-ul)
* **date\_and\_time** ( data si ora la momentul inregistrări review-ului)
* **stars** ( numarul de stele oferit de pacient)
* **comment** ( comentariul pacientului în legătură cu review-ul)

# 5.Prezentarea aplicației

Aplicația “MediConnect – the all-in-one medical booking and management solution” este o aplicație cross-platform care rulează atât pe browsere web pe windows cât și pe sistemele de operare Android și IOS.

Aplicația este de asemenea împărțită în trei tipuri de interfețe fiecare cu propriile funcționalități în concordanță cu necesitățile tipului de utilizator al acesteia.

Astfel, aplicația cross-platform dezvoltată tine locul a trei aplicații: un site web și două aplicații pentru telefonul mobil.

Exista trei roluri pentru utilizatorii acestei aplicații: ”user”(utilizatorul pacient ), ” doctor” și ” administrator”. Aceste trei roluri determină în continuare tipul de interfața permisă care poate fi accesată de utilizator:

* Utilizatorul cu rol “user”: acesta are acces la aplicația destinată telefonului mobil (Android/IOS), continuând cu interfața destinată pacienților și cu funcționalitățile specifice.
* Utilizatorul cu rol “doctor”: acesta are acces la aplicația destinată telefonului mobil (Android/IOS), continuând cu interfața destinată doctorilor și cu funcționalitățile specifice.
* Utilizatorul cu rol “administrator”: acesta are acces la site-ul web, deschis dintr-un browser, continuând cu interfața destinată adminstratorilor și cu funcționalitățile specifice.

/////diagrama////

În cazul în care, unul dintre tipurile de utilizatori descrise mai sus încearca autentificare pe platforma care nu îi este destinată, v-a fi îndrumat printr-un mesaj aparut în interfață către plafrorma potrivită rolului său.

În contiunare voi prezenta aplicația din pespecitiva fiecărui utilizator menționat, detaliind funcționalitățile create special pentru fiecare tip de utilizator.

## 5.1 Aplicația din perspectiva rolului “User”

Partea aplicației destinată utilizatorului pacient este creată atent pentru nevoile acestuia având un design intuitiv, ușor de folosit pentru persoane de toate vârstele și funcționalități special create pentru o utilizare cu succes atât din punctul de vedere al unei rezervări rapide a unei programări medicale, cât și din perspectiva informării despre detaliile doctorului sau a review-urilor acestuia.

Utilizatorul pacient (rol “user”) beneficiază de următoarele funcționalități în cadrul aplicației:

* Poate creare un cont.
* Se poate autentifica în aplicație cu contul creat.
* Poate să reseteze parola contului.
* Poate vedea lista copletă de doctori și poate filtra această listă în funcție de categorie sau printr-o bara de cautare după numele doctorului
* Poate vedea lista cu toate categoriile medicale din care fac parte doctorii și poate filtra categoriile cu ajutorul unei bare de căutare.
* Poate să vadă delatiile fiecărui doctor și toate review-urile acestuia.
* Poate adauga un review.
* Poate să își faca o promgramare la doctorul ales.
* Poate vedea toate programările vitoare și anterioare.
* Poate anula o programare
* Poate să își vadă și să își editeze detaliile contului.
* Poate să se delogheze din aplicație

### 5.1.1 Autetificarea în aplicație

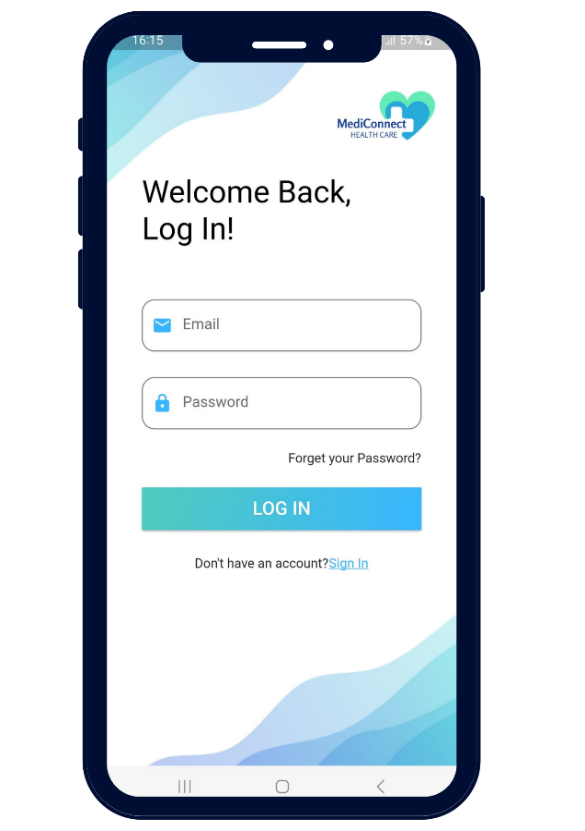
****

Figura 4.1- Login Screen

La intrarea în aplicație de pe telefonul mobil (Android/IOS) utilizatorul este întâmpinat cu pagina de autentificare (login\_screen) prezentată în figura 4.1. Utilizatorul se poate autentifica în aplicație prin intermediul celor doua căsuțe text, destinate email-ului și parolei. După introducerea acestor informații, la apăsarea butonului din partea de jos există doua posibile scenarii:

* autentificarea este cu succes: în funcție de rolul utilizatorului găsit în baza de date, aplicația va deschide pagina principală destinată în mod specific fiecărui rol.
* autentificare nu se poate realiza: în acest caz există mai multe motive care pot duce la incapacitatea de autentificare, pe care le voi detalia în continuare.

Lispa unui cont de utilizator cu emailul introdus v-a fi semnalat conform figurii (\*) printr-un mesaj de atenționare prezent în pagina de autentificare.

Parola introdusă nu coincide cu parola corespunzătoare contului, fapt semnalat conform figurii (\*).

Unul sau mai multe câmpuri nu respectă următoarele cerințele:

* emailul introdus trebuie să aibă format de email și minim 10 caractere. Nerespectarea acestei cerințe va fi semnalaă conform figurii(\*).
* parola trebuie să aibă minim 8 caractere, litere mari și mici, cel puțin un caracter număr și cel puțin un caracter special. Nerespectarea acestei cerințe va fi semnalată conform figurii(\*).

Prin intermediul ecranului de autentificare se poate face accesul către alte două pagini. Apăsând butonul cu text subliniat “Sign in” utilizatorul este direcționat către pagina de creare de cont, fiura(\*). Apăsând pe textul “Forget your Password?” utilizatorul este redirecționat către pagina de resetare a parolei, figura (\*).

### 5.1.2 Creearea contului

După apăsarea butonului de “Sign in”, utilizatorul este redirecționat în pagina de creare de cont (Resgister\_screen) prezentă în figura (\*).

Pentru crearea unui cont de urilizator/pacient, utilizatorul trebuie să introducă prin intermediul căsuțelor text numele, prenumele, numărul de telefon, emailul, parola și confirmarea parolei. După introducerea acestor informații există două posibilități:

* înregistrarea s-a realizat cu succes și utilizatorul este redirecționat către pagina de autentificare.
* Înregistrarea nu s-a putut realiza. Acest fapt se poate întâmpla din mai multe, motive toate fiind semnalate prin intermediul interfeței. Motivele pot fi următoarele:
* email-ul introdus corespunde deja unui cont din baza de date, sau nu are format de email.
* parola și confirmarea parolei nu coincid sau nu întrunesc cerințele de complexitate: minim 8 caractere, litere mari și mici, cel puțin un caracter cifra și cel puțin un caracter special.
* numele sau prenumele nu se află în limitele numărului de caractere, minim 2 caractere, maxim 30 de caractere.
* Numărul de telefon nu se încadrează în limitele numărului de caractere.

Pagina destinată creări contului, conține de asemenea și un buton “Log in” care redirecționează utilizatorul către pagina de autentificare.

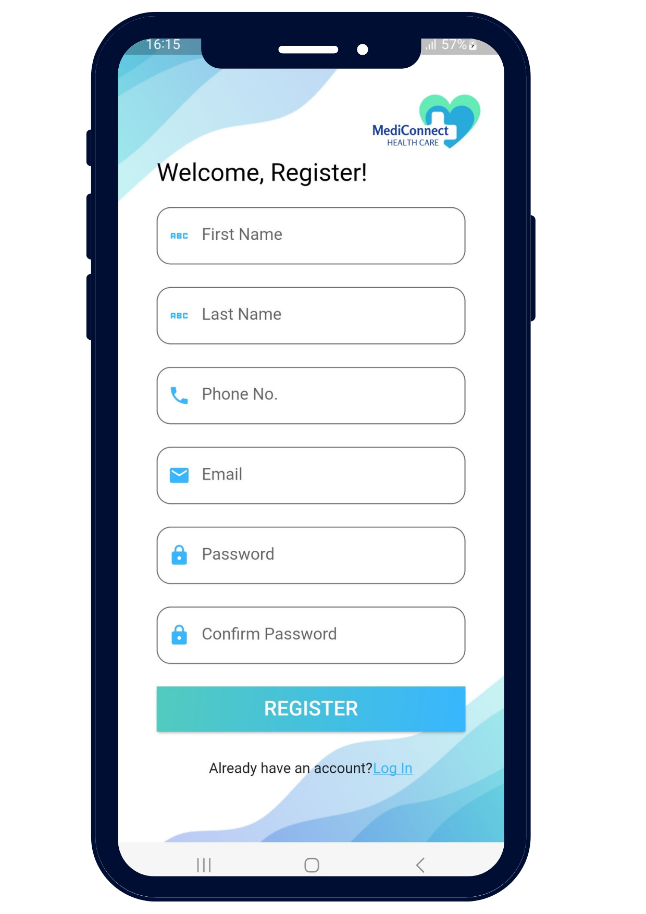


Figura 4. – Register Screen

### 5.1.3 Resetarea parolei

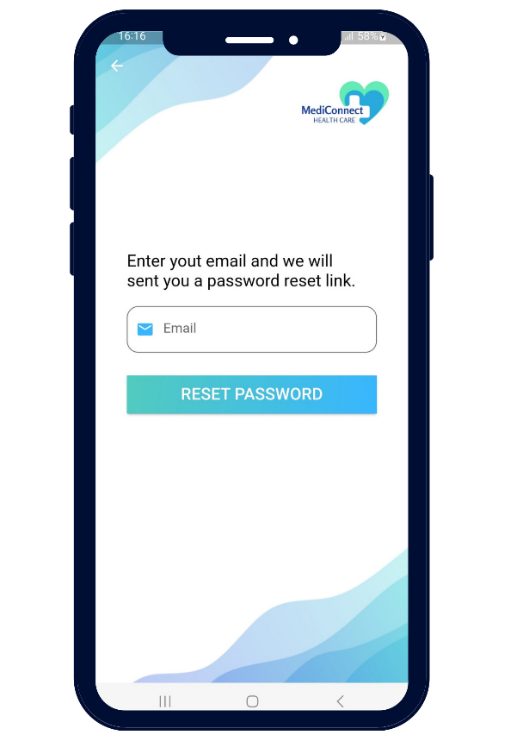
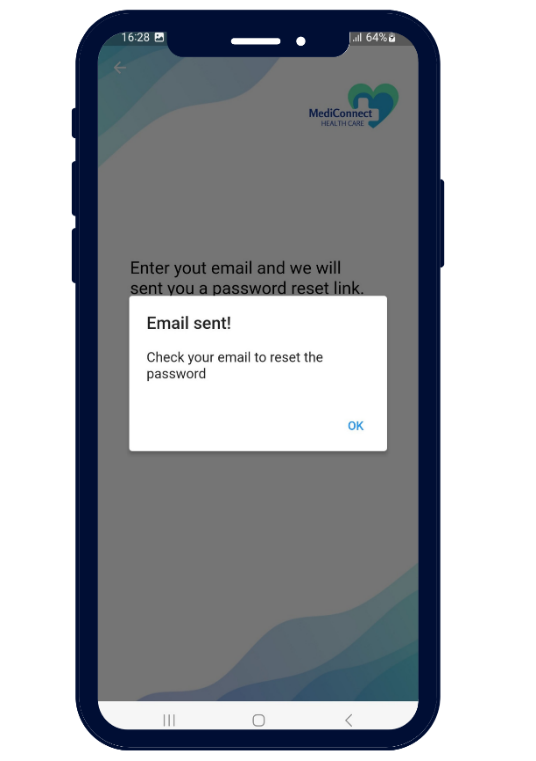
** **

Figura 4. – Forget Password Screen Figura 4. - Mesaj de confirmare

După apăsarea butonului text “Forget your password?” din pagina de autentificare, aplicația redirecționează utilizatorul către pagina de resetare a parolei (forget\_password\_screen) prezentată în figura (\*).

Utilizatorul este îndrumat prin intermediul unui text să introducă în căsuța text emailul contului pentru care acesta nu își mai reține parola. La apăsarea butonului “Reset Password”, se trimite un email către emailul introdus (acesta aparținând contului) prin intermediul căruia se poate face resetarea parolei. Toate informațiile necesare sunt prezente în emailul trimis. Pe ecran apare un mesaj care anunță utilizatorul că emailul a fost trmis ca în figura (\*).

De asemenea pagina aceasta cuprinde în partea de sus a ecranului un buton care face redirecționarea către pagina precedentă.

### 5.1.4 Pagina de home

După autentificarea cu succes în aplicație și identificarea rolului de utilizator, pe ecranul telefonului de deschide pagina principală, pagina Home prezentată în figura 4. (\*).

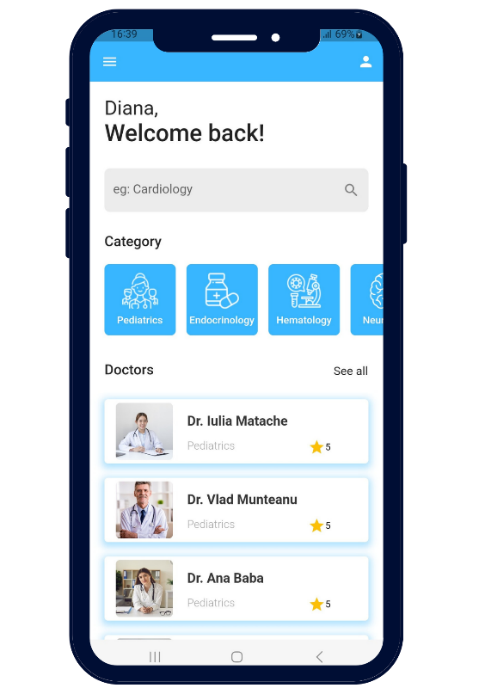
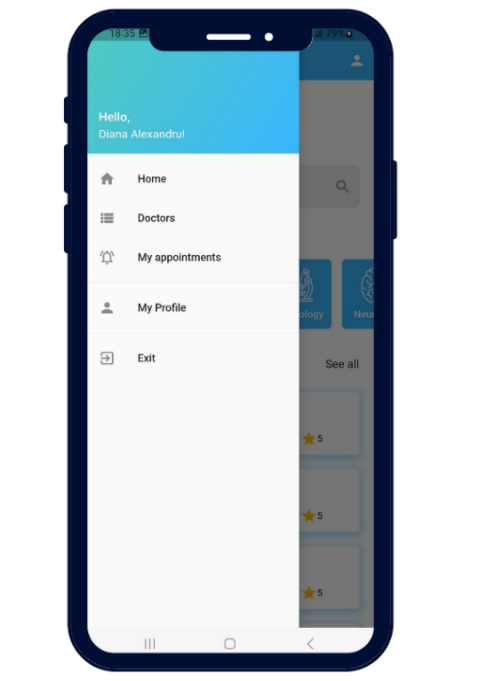
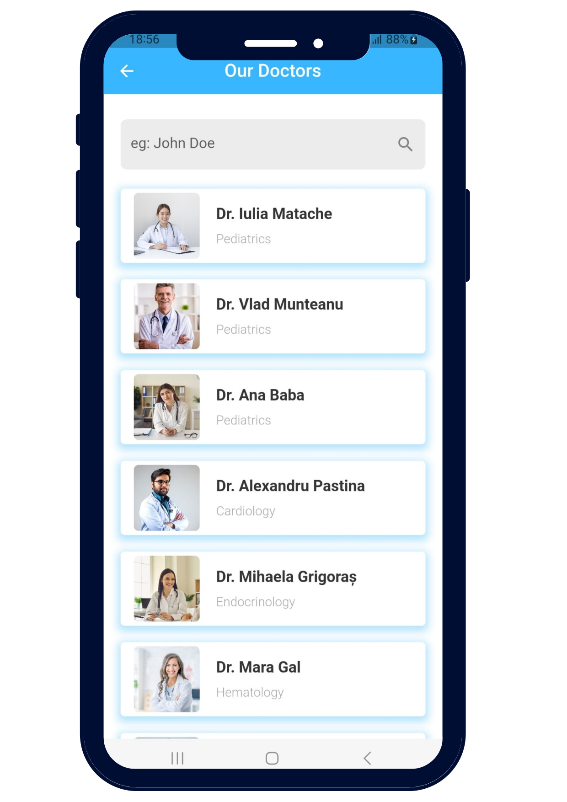
  

Figura 4. – Pagina Home Figura 4. - Meniul aplicației Figura 4. – Pagina doctorilor

În acest ecran sunt prezente numeroase funcționalități ale aplicației pe care le voi descrie în continuare.

În partea de sus a ecranului este prezent un buton cu pictogramă reprezentat prin trei linii care la acțiunea de apăsare deschide în pagină meniul aplicației prezentat prin figura 4. (\*).

Tot în partea de sus a ecranului exista un buton în partea stângă care la acțiunea de apăsare duce la ecranul cu detaliile de utilizator ale contului, figura 4. (\*) .

Utilizatorul poate vedea în a doua parte a ecranului lista completă a doctorilor care se poate derula pe verticală. La apăsarea pe un chenar al unui doctor, se aprinde ecranul cu detaliile despre doctorul respectiv conform figurii 4. (\*). La apăsarea butonului text “See all” se deschide un ecran care conține lista tuturor doctorilor conform figurii 4. (\*).

În prima parte a ecranului este prezentă o listă cu toate categoriile medicale din care doctorii fac parte, lista care se poate derula pe orizontală. Cu ajutorul barei de căutare se poate filtra lista de categorii după numele categoriei căutate. Daca se reușește găsirea categoriei, în listă va mai apărea doar aceasta, pana la ștergerea textului din bară. Dacă nu se găsește categoria căutată, în locul listei va apărea mesajul “No results found!”. La apăsarea unei categorii din listă se va desghide ecranul generic destinat listei de doctori, menționat și mai sus, dar asupra listei prezente în acel ecran se va aplica un filtru în funcție de categorie.

### 5.1.5 Meniul aplicației

Meniul aplicației se deschide la apăsarea butonului special destinat din pagina de Home, și arată conform figurii 4. (\*).

Cu ajutorul acestuia se face atât accesul către celelalte pagini principale din aplicație, dar și delogarea utilizatorului.

Meniul permite următoarele acțiuni:

* direcționare către pagina Home
* direcționare către pagina Doctors, destintă listei de doctori
* direcționare către pagina My Appointments, destinată listelor cu programările viitoare și cele trecute
* direcționarea către pagina My Profile, detinată detaliilor contului deținut de utilizator
* ieșirea din contul conectat prin delogare

### 5.1.6 Pagina Doctors

Acesta pagina, prezentată prin figura 4.1 (\*), este detinată vizualizări listei de doctori, continând mai multe acțiuni posibile.

În funcție de modul în care s-a ajuns în pagina lista de doctori difera, deoarece d aca accesul s-a facut printr-o categorie prezentă în pagina Home , lista de doctori nu este cea complete aceștia fiind doar doctorii care apartin acelei categorii. Dacă accesul s-a făcut prin intermediul secțiunii din meniu, sau a butonului text “See all” prezent în pagina Home, atunci lista de doctori este cea completă.

În partea de sus a paginii există o bară de căutare cu ajutorul căreia se poate face filtrarea în funcție de nume, acesta filtrare fiind deosebit de utila dacă lungimea listei de doctori este relativ mare.

Prin apăsarea unui chenar care ilustrează un doctor, se aprinde pagina destinată profilului doctorului, pagina Doctor details.

### 5.1.7 Pagina Doctor details

Pagina destinată detaliilor doctorului, prezentată in figurile 4.(\*) si 4.(\*), este structurată cu ajutorul a doua tab-uri: primul dintre aceste intitulat “About”, Figura 4. și cel de al doilea intitulat,“Reviews” , figura 4. .

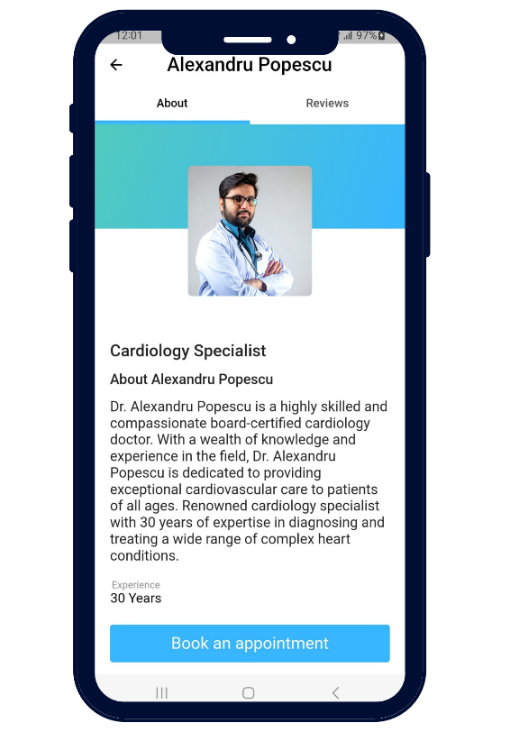
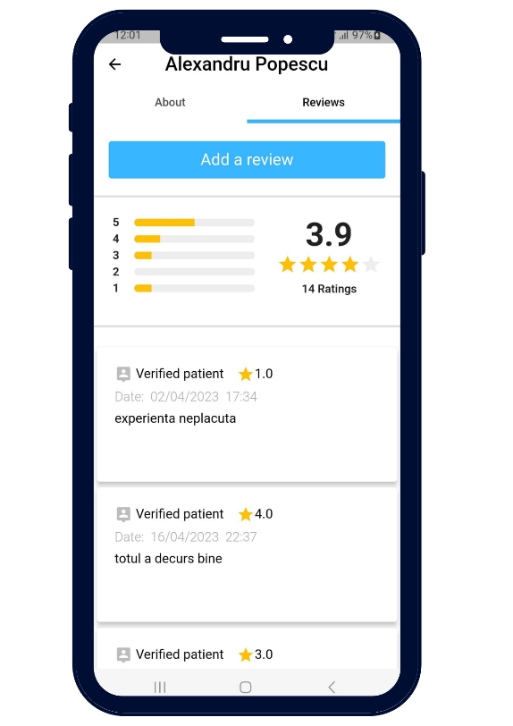
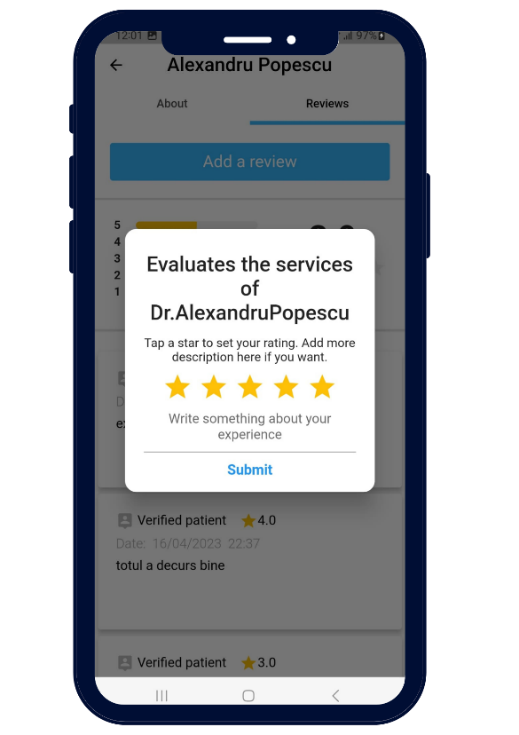
  

Figura 4. – Doctor details Figura 4. – Tab Reviews Figura 4. – Adaugare review

Această pagină este creată special pentru a simplifica procesul de informare a pacientului cu privire la detaliile doctorului și a review-urilor acestuia. Pacientul dechide o pagină care conține toate informațiile importante pe care le poate analiza în vederea deciziei de a efectua o porgramare, informații care reprezintă o imagine clară asupra profesionalismului doctorului.

În primul tab al paginii, se regasesc informații generale despre doctor ca numele complet, categoria medicala din care face parte, o scurta descriere dar și anii de experiența ai acestuia. De asemenea în partea de jos, se afla un buton text intitulat “Book an appointment”, care la acțiune de apăsare direcționează mai departe utilizatorul către pagina unde se poate efectua o progamare, figura 4. .

În al doilea tab al paginii, se regasesc mai multe componente, pe care le voi detalia în continuare. În partea de sus a tab-ului este prezent un buton text “Add a review”, iar prin apasare acestuia sunt posibile doua scenarii. În primul caz, cand pacientul are cel puțin o programare efectuata, se permite utilizatorului adăugarea unui nou review, prin deschiderea unei casuțe dialog special create pentru această actiune, figura 4. . În casuța de dialog sunt prezente mesaje instructive cu privire la adăugarea review-lui. Pentru a adăuga un review trebuie selectat numarul de stele dorit din cele cinci selectate implicit, si deasemenea trebuie adaugat un comentariu În căsuța text. La apăsarea butonului text intitulat “Submit” se efectuează adăugarea review-ului în baza de date.

În cel de-al doilea caz, cand pacientul nu are nicio programare efectuala la doctorul pentru care dorește să faca review, acțiunea nu este posibilă si este semnalatp printr-un mesaj care apare pe ecran sub forma de casuța dialuc cu un text sugestiv “If you don't have a confirmed appointment for this doctor in the past you can't add a review.”.

Următorul widget al paginii este o statistică bazată pe review-urile doctorului. În partea din stanga a acesteia sunt reprezentate prin 5 bare cât la sută din totalitatea review-urilor reprezintă fiecare categorie din cele 5 variante de răspuns. În partea din treapta este prezent media stelelor oferite de pacienții prin raspunsurile acestora, dar și numarul total de review-uri adăugate.

În continuare exista o listă a tuturor review-urilor adăugate de pacienții fiecare fiind reprezentat de un chenar unde se gasesc data, ora, numarul de stele oferit și comentatiul scris de fiecare pacient. Numele acestora nu apare, pentru a proteja confidențialitatea fiecarui utilizator, nefiind o informație absolut necesară pentru creeare unei opinii despre calitatea serviciului oferit de doctori.

### 5.1.8 Rezevarea unei progrmări

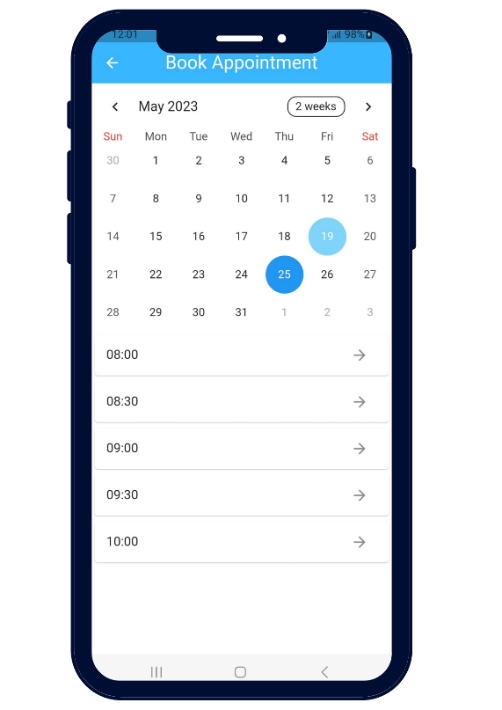
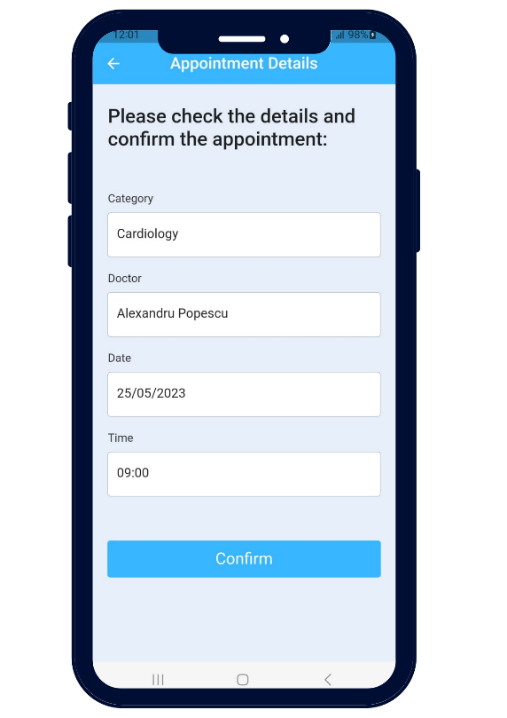
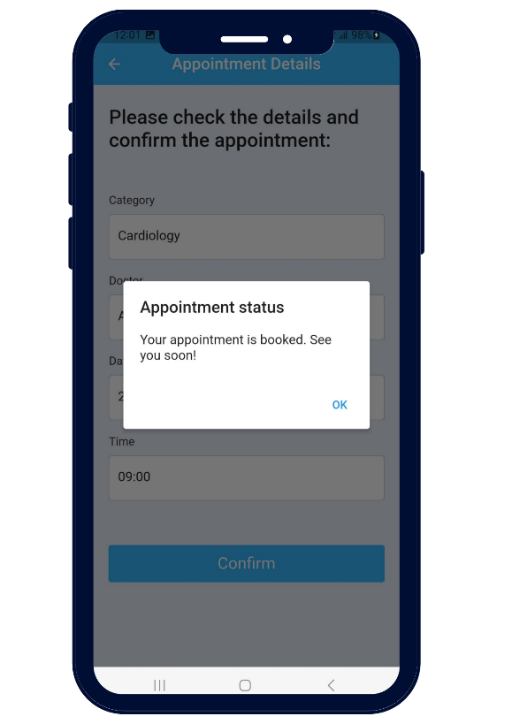
  

Figura 4. - Figura 4. - Figura 4. -

În acest subcapitol, cum reiese și din nume, voi prezenta în continuare modul de efectuare al unei programări. După ce utilizatorul alege un doctor și apasă butonul text “Book an appointment” din pagina cu detaliile doctorului, acesta este direcționat catre ecranul de rezervarea al unei progrmări, conform figurii 4. (\*). Acest ecran are o organizare intuitiva, fapt care ajuta pacientul să își efectueze în mod rapid o progrmare, fiind un real avantaj.

În prima parte a ecranui este prezent un calendar de unde utilizatorul poate selecta data dorită pentru programare. La deschiderea ecranului este selectată în mod implicit data zilei curente. Calendarul se deschide implicit, in mod obișnuit, cu toate saptămănile din luna curentă, dar afișarea acestuia poate fi modifică de către utilizator prin butonul text din partea dreaptă care sugereza celelalte moduri de afisare, în funcție de modul current. Calendarul permite următoarele trei moduri: două săptămâni (2 weeks), o săptămână (week), o lună (month). De asemenea, prin butoanele “<” și “>”, utilizatorul poate naviga prin lunile din anul calendaristic.

În a doua partea a ecranului se afișează în funcție de data selectată locurile libere pentru progrmări din care utilizatorul poate alege. Locurile libere sunt prezentate prin intermediul unei liste de chenare, fiacre dintre acestea avand ora de inceput a programări si un buton reprezentat printr-o săgeata care la atingere deschide pagina pentru confirmarea programării conform figurii 4. (\*). În cazul în care doctorul nu mai are niciun loc liber in programul din data selectată, în locul listei apare un mesaj care sugerează pacientului acest lucru, “We are sorry, there is no available time on this day.”.

După alegerea datei și orei dorite pentru programare, și apăsarea butonului mai sus menționat, pe ecranul telefonului se deschide pagina cu detaliile programării. Aici, utilizatorul își poate citii cu atenție detaliile despre progrmarea dorită: numele doctorului, categoia acestuia, data și ora aleasă. După verificarea acestor informații, la apasarea butonului text “Confirm”, se efectueaza confirmarea programării, prin introducerea acesteia în baza de date. În plus, dupa introducerea progrmarii în baza de date, pe ecran apare un mesaj care informaează pacientul că progrmarea a fost inregistrată, conform figurii 4. (\*). La apăsarea butonului “Ok” utilizatorul este redirecționat către pagina Home.

### 5.1.9 Pagina “My appointments”

Această pagină este destinată în principal vizualizări programărilor viitoare, dar și trecute a pacientului înregistrat. Structura acesteia consta în două tab-uri: “Upcoming” pentru vizualizare programarilor viitoare, figura 4.() și ”Past” pentru vizualizarea celor efectuate în trecut, figura 4. (\*).

Ambele tab-uri conțin câte o listă a programărilor, fircare dintre programări constând întrun chenar cu urmatoare infromații: data, ora și numele. Listele sunt ordonate în mod cronologic, dinîndu-se cont atât de dată, cât și de oră.

În plus fiecare chenar prezintă câte un buton intitulat “Cancel”. În cazul listei cu programări viitoare buton permite acțiunea de anulare a progamării. La apăsarea acstuia se deschide o fereastră de dialog care intraba utilizatorul dacă dorește anularea progrmări. La apăsarea butonului “Yes” din fereastra dialog, se efectueaza stergerea progrmării din baza de date , în caz contrar, la apăsarea butonului “No”, se închide fereastra fara a se efecua ștergerea. În cazul listei cu programări trecute, butonul “Cancel” nu permite nicio acțiune.

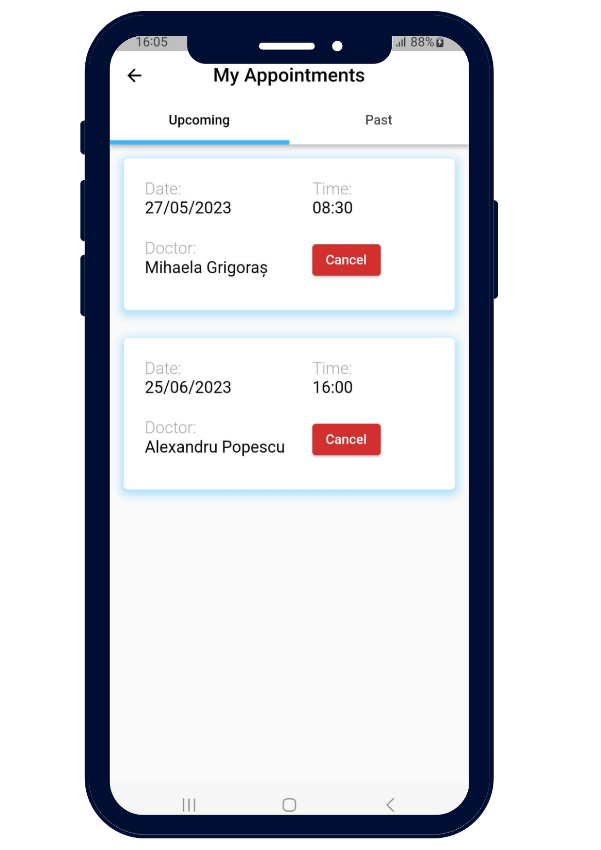
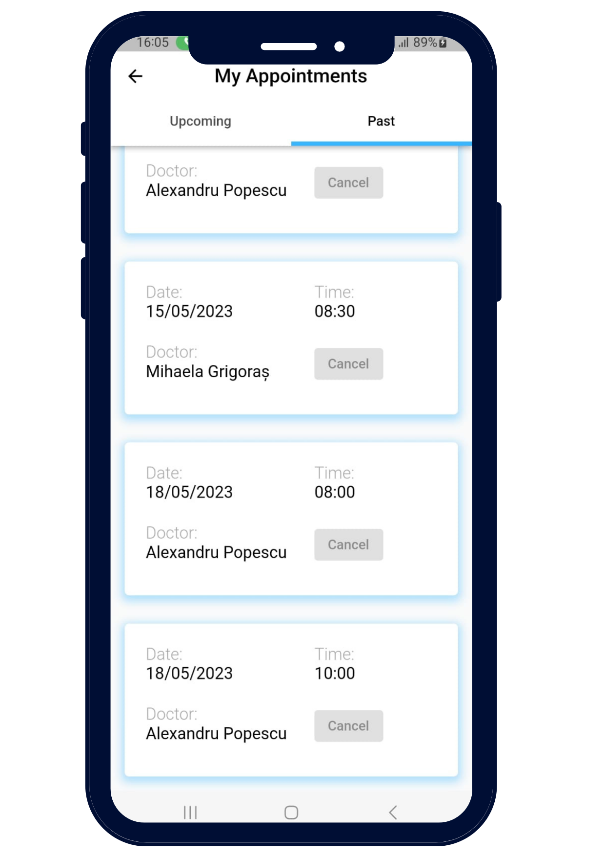
 

Figura 4. () Figura 4.()

### 5.1.10 Pagina My Profile

În această pagină, așa cum îi sugrează și numele, utilizatorul își poate vizualiza și edita detaliile contului cu care este intregistrat în aplicație.

Accesul în pagina se face prin apăsarea butonului text “My profile” din meniu sau prin apăsarea butonului din dreapta sus a paginii principale.

Pagina se deschide automat în modul de vizualizare, conform figurii 4., unde se pot observa casetele text dezactivate care conțin numele, prenumele și numărul de telefon asociate contului utilizatorului. Prin apăsarea butonului cu pictogramă de creion situat în partea dreaptă sus a ecranului, pagina trece în modul de editare, conform figurii 4., ceea ce activează casetele text, făcându-le editabile. În acest mod, utilizatorul are posibilitatea să modifice câmpurile dorite și să salveze aceste modificări prin intermediul butonului text "Save" amplasat în partea de jos a ecranului.

În colțul sus dreapta, este prezent un buton cu săgeata orientată către stânga, care la acțiunea de apăsare direcționează utilizatorul către pagina principală.

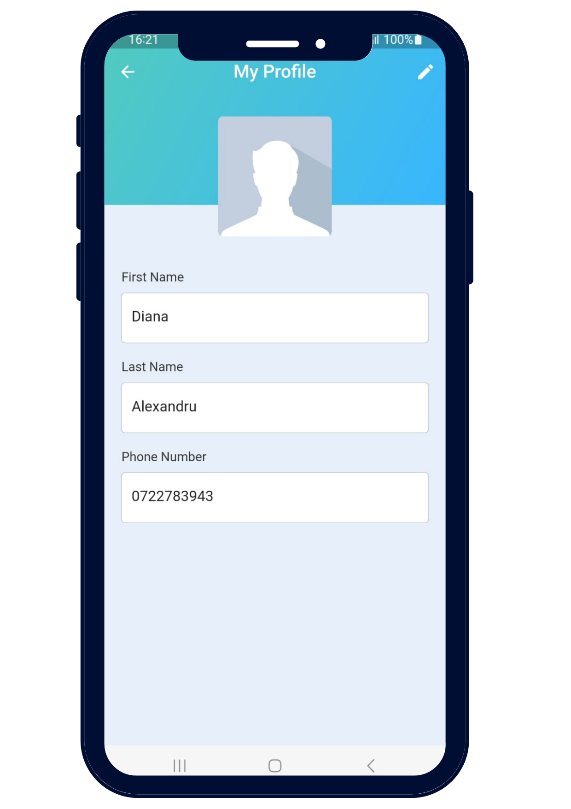
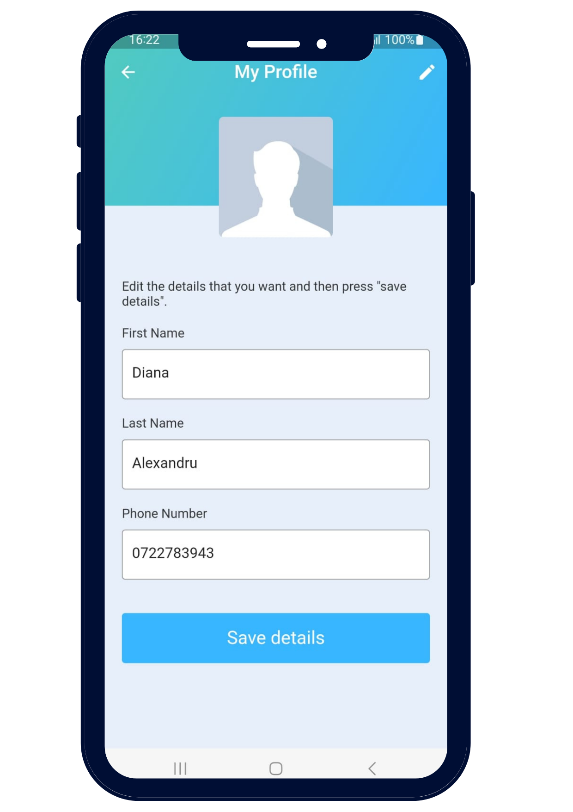
 

Figura 4. – My Profile -Modul vizualizare Figura 4. – My Profile modul editare

## 5.2 Aplicația din pespectiva rolului “Doctor”

Așa cum reiese și din titlul, în acest capitol voi detalia aplicația din perspectiva rolului “Doctor”.

Această partea a aplicației este creată punctual pentru nevoile doctorului angajat într-o clinică medicală. Desingn-ul este unul placut, care face utilizarea aplicației simplă și intuitivă.

Funcționalitătile acestei parții a aplicației sunt simple, predominant cu rol informativ, menite să tină locul unei agende a medicului.

Utilizatorul doctor (rol “doctor”) beneficiază de următoarele funcționalități în cadrul aplicației:

* Poate creare un cont de utilizator care se va modifica ulterior, prin intermediul aplicației web destinată administratorilor, cu rol de doctor si toate informatiile special destinate acestui tip de cont.
* Se poate autentificarea în aplicație.
* Poate să reseteze parola contului.
* Poate vedea lista copletă de doctori și poate filtra această listă în funcție de categorie sau printr-o bara de cautare după numele doctorului
* Poate sa vada programarile viitoare si cele trecute ordonate in functie de data si ora.
* Poata sa vada toate review-urile care ii sunt destinate inclusive statistica bazată pe acestea.
* Poate confirma venirea unui pacinet la programarea rezervată.
* Poate sa își vadă si sa își editeze detaliile generale ale contului (Nume, Prenume, numar de telefon).
* Poate să vadă detaliile contului, specific utilizatorului doctor (Categoie, descriere, experienta, imagine)
* Poate să se delogheze din aplicație.

### 5.2.1 Pagina de autentificare

Pagina de autentificare este comună pentru ulizatorul cu rol “user” și pentru cel cu rol “doctor”, pagină descrisă detaliat în subcaptitolul 4.1.1.

Diferențierea aplicatiei pentru cele două roluri începe după momentul autentificării cu scucces, când se verfică rolul contului. Astfel, după apăsarea butonului text “LOG IN”, În cazul autentificării cu success și recunoașterea rolului asociat contului, utilizatorul doctor este direcționat cătrea pagina sa de home, conform figurii 4. (\*).

### 5.2.2 Crearea unui cont de doctor

În cazul creări unui cont pentru un doctor, este nevoie de mai mulți pași fața de crearea unui simplu cont de utilizator pacient, pași pe care îi voi descrie succint în continuare, urmând să fie prezenți in mod detaliat in subcapitolul 4.2 destinat functionalităților administratorului.

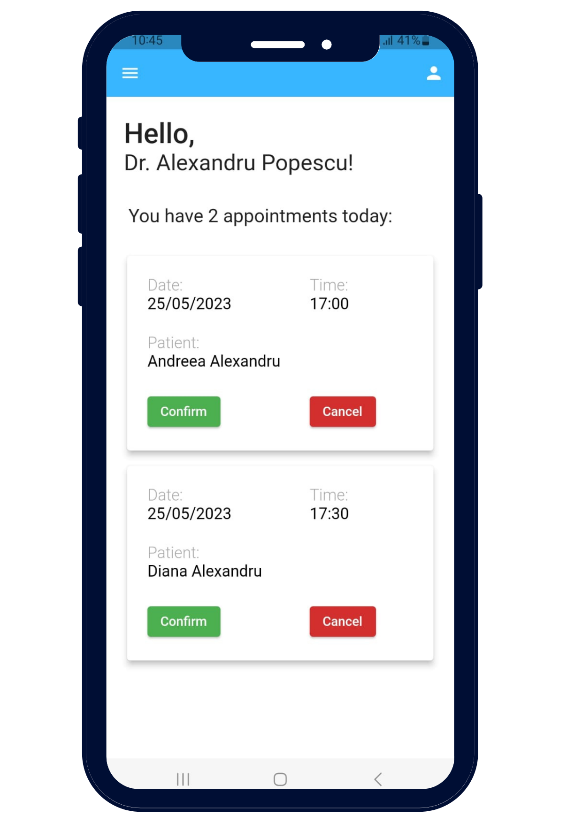
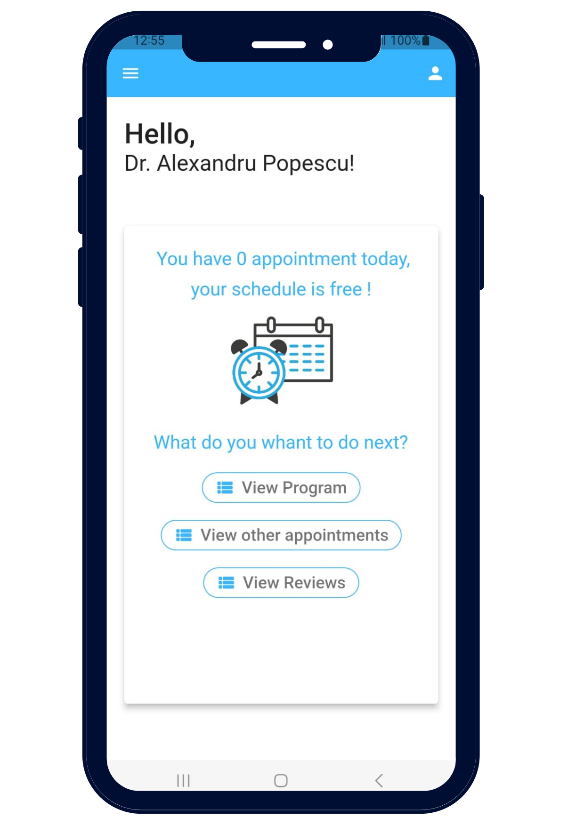
Primul pas ar fi crearea unui cont de utilizator cu rol implicit user de către doctor prin intermediul aplicației destinată telefonului mobil, acțiune prezentată în sectiunea 4.1.2, care asigură înregistrarea informațiilor persoanle de bază a doctorului și anume nume, prenume, numar de telefon, email.

În continuare, cel de al doilea pas presupune acțiunile permise administratorului, pentru crearea unui cont de doctor. Acesta va căuta și va alege, prin intermediului paginii web, contul asociat utilizatorului cu rol user pe care dorește să il transforme în cont doctor. Căutarea unui cont de user se poate face cu ajutorul unui tabel cu utilizatori sau prin intermediul unei bare de căutare și a email-ului. Contul găsit prin una dintre cele două metode menționate anterior poate fi transformat în cont asociat doctorului prin adăugarea și salvarea unor informații prin intermediul unei paginii special crate. Este necesară adaugarea urmatoarelor detalii: categorie, ani de experiență, descriere, imagine de profil, program. La salvarea acestor informații, rolul utilizatorului este schimbat din “user” în “doctor”.

### 5.2.3 Resetarea parolei

Funționaliatea de resetare a parolei este comuna pentru ambele tipuri de utilizatori ai aplicatiei mobile, astfel si in cazul unei personae cu cont avănd rol “doctor”, resetarea parolei se face în aceeași manieră și prin inermediul aceleiași paginii ca modul descris în secțiunea 4.1.3.

### 5.2.4 Pagina de home

** **

**Figura 4. – Pagina principala Figura 4.1- Pagina Pirnicipala**

Scopul acestei paginii este acela de a informa doctorul cu privire la programările din ziua curentă, astfel încă de la deschiderea aplicației doctorului poate vizualiza dacă la data curentă are de efectuat programări sau are programul liber. În continuare voi descrie pagina și modul de utilizare al acesteia în mod detaliat.

După ce utilizatorul s-a autentificat cu succes și i s-a identificat rolul de doctor acesta este direcționat către pagina principală a aplicației, pagina Home prezentată în figura 4. (\*).

La fel ca în pagina principală destinată pacienților și aceasta conține doua butoane cu pictograma în partea de sus, în stânga și în dreapta. Primul buton, cel din partea stânga, la acțiunea de apăsare deschide meniul aplicației, conform figurii 4. (\*). La apăsarea butonului din partea dreaptă a ecranului, se deschide pagina care conține informațiile persoanele ale contului de utilizator.

În partea de sus a ecranului este prezentă o forma de salut alături de numele doctorului, pentru a crea o experiență personalizată a folosirii aplicației.

În continuare, există doua cazuri diferite de afișaj și funcționalități în această pagină. Dacă doctorul are programări de efectuat în ziua curentă, atunci pe ecran apare un mesaj care specifică numărul de porigrogrmari, iar în contiunarea acestui mesaj pare o lista de chenare cu scrol vertical ficare chenar fiind asociat unei progrmări și prezentănd informatiile acestora precum numele și prenumele pacientului , data și ora progrmarii. Modul de utilizarea a chenarelor va fi descris in secțiunea 4.2.6. În cazul în care doctorul nu are nicio programare rezervată în ziua curentă, pe ecran apare nu mesaj care specifică acest lucru. În continuare, pe ecran vor fi sugerate alte trei posibile acțiuni în aplicație cu ajutorul butoanelor text: “View program”, “View other appointments”, “View reviews”. La apăsarea unuia dintre aceste butoane doctorul v-a fi redirecționat către pagina cu programul său de lucru sau către pagina cu toate programările sau către pagina de review-uri. Acest mod de prezentare a funcționalităților face utilizatorul să navigheze mai eficient în aplicație în cazul în care nu este prezentă nicio programare în pagina principală.

### 5.2.5 Meniu

Accesul la meniul aplicației se face prin intermediul butonului cu pictogramă plasat în pagina principala, pagina home, în colțul stânga sus, conform figurii 4.(\*).

Cu ajutorul acestuia se face atât accesul către celelalte pagini principale din aplicație, da și delogarea utilizatorului.

Meniul cuprinde două părți. În prima parte a acestuia este prezentă imaginea de profil a doctorului și numele acestuia pentru a oferii o experiența plăcută și personalizată a utilizării aplicației. În a doua parte, se află o lista de widget-uri, care cuprind o pictogramă desciptivă și un buton text care specifică numele paginii, către care utilizatorul este direcționat prin apăsarea acestuia.

Meniul permite următoarele acțiuni:

* direcționare către pagina Program
* direcționare către pagina Appointments, destintă listei de programari trecute si viitoare
* direcționare către pagina Reviews, destinată listei cu review-rile doctorului și statistica bazată pe acestea.
* direcționarea către pagina My Profile, detinată detaliilor personale ale utilizatorului contului
* direcționarea către pagina My Doctor Details, detinată detaliilor doctorului care vor fi de asemenea vizibile in aplicație pentru pacienții și adiministratori
* ieșirea din cont prin delogare

### 5.2.6 Pagina Appointments

Această pagină oferă o prezentare intuitivă și detaliată a tuturor programărilor medicului, inclusiv celor viitoare și celor efectuate în trecut. Funcționează ca o agendă digitală a medicului, oferindu-i un avantaj semnificativ atunci când dorește să vizualizeze în mod exact programările pe care le are de efectuat în diferite zile.

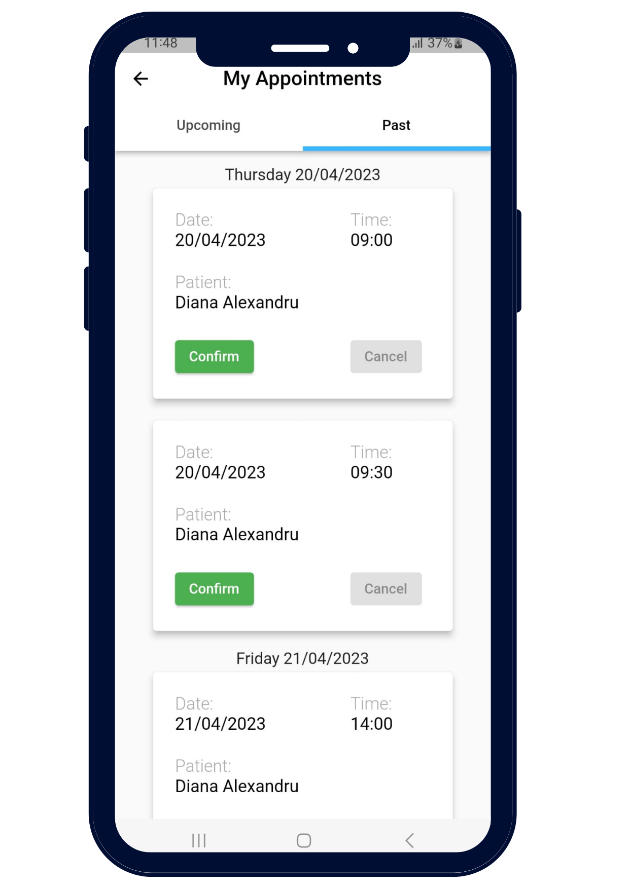
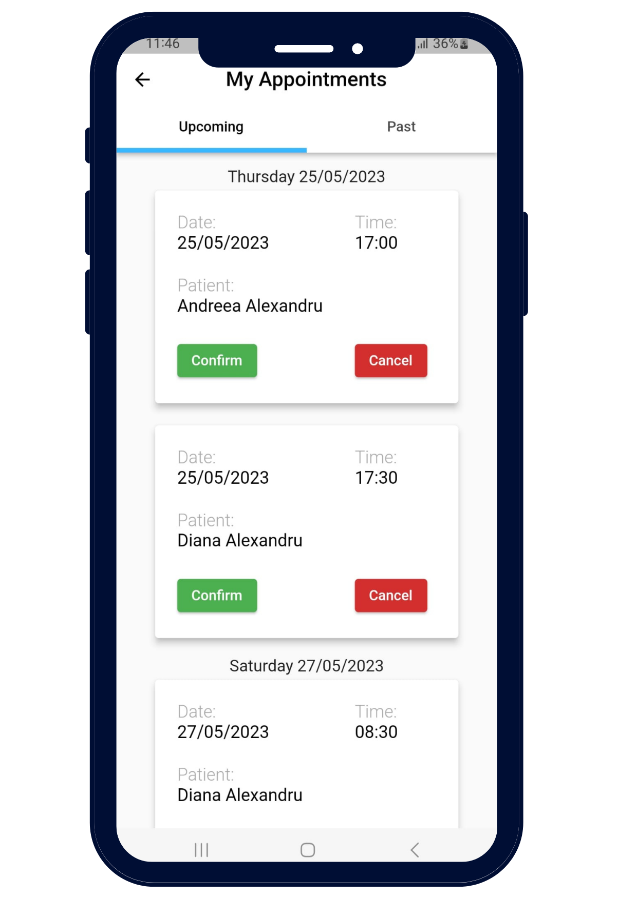
Prin intermediul acestei pagini, medicul poate obține informații complete despre programările sale. Aceasta include datele și orele programărilor viitoare, precum și detaliile despre programările efectuate anterior. Astfel, medicul are o imagine clară și concisă a tuturor programărilor sale.

Pagina, conform figurii 4.(\*), este formată din doua tab-uri intitulate “Upcoming”, conținand lista de programări viitoare și “Past”, conținând lista de programări trecute.

Ambele liste sunt compuse din chenare, fiecare reprezentând o programare. Programările pentru zile diferite sunt despărțite prin intermediul unui separator care indică ziua săptămânii și data respectivă. Această structură de afișare în pagină oferă o imagine clară asupra numărului de programări pe care le are medicul în fiecare zi, facilitând astfel identificarea și urmărirea programărilor în mod efficient, printr-o perspectivă rapidă și intuitivă.

Fiecare chenar asociat unei programări cuprinde următoarele informații: data, ora și numele pacientului. Chenarele au de asemenea două butoane text “Confirm” și “Cancel”. Butonul Confirm permite doctorului să confirme venirea pacientului la progrmare și onarea acestei rezervări. Această acțiune este permisă atăt în tabul cu progamarile vitoare căt și în cazul programaărilor trecute deoarece spre exemplu o progrmare poate avea întarziere trecănd automat în categoria de progrmări trecute, insă pacientul poate onara acestă rezervare cu intarziere fiind confirmată ulterior. Cazul descris anterior este in mod cert un caz destul de rar deoarece confirmarea venirii la programare se poate face de catre administrator din pagina web la sosirea pacientului în clinică, acestă abordare fiind recomandtă în cazul utilizării aplicației. Prin aprsarea butonul Cancel care sugereaza anularea unei programări, in aplicatie utilizatorul doctor primeste un mesaj prin intermediul unei casete de dialog care sugerează ca aceasta actiune poate fi performată doar de catre un administrator, din motive de siguranță.

### 5.2.7 Pagina Program

****

Așa cum reiese și din nume, această pagină este destinată programului doctorului, pagina care arată conform figurii 4.(\*). La fel ca în cazul paginii descrise anterior și aceasta are rol de agendă digitală a doctorului.

Pagina cuprinde fiecare zi a săptămânii în care doctorul are program. Fiecare dintre aceste este urmată de câte o listă formată din chenare reprezentând intervalul orar al fiecărei programări pe care doctorul o poate efectua in ziua respectivă. Dacă in cadrul unei zile doctorul are locuri pentru programări care nu sunt una în continuarea celeilalte (ex: 8:30 – 9:00 și 17:00- 17:30) între cele două programări distanțate ca timp v-a apărea un separator cu un mesaj care sugerează faptul ca intre aceste perioade de timp, doctorul are timp liber. Această abordare a afișări permite utilizatorului să își vizualizeze programul într-un mod eficient fapt care îl poate ajuta în planificare timpului destinat activităților personale.

În colțul stânga sus al paginii este prezent un buton cu pictogramă sugestivă care la acțiunea de apăsare direcționează utilizatorul înapoi la pagina principală a aplicației.

În colțul dreapta sus se afla un buton cu pictogramă sugestiva care la apasare deschide un chenar cu un text care specifica utlilizatorului ca pentu modificarea programului este nevoie de intervenția administratorului.

### 5.2.8 Pagina My Profile

Această pagină oferă utilizatorului posibilitatea de a vizualiza și edita detaliile contului său înregistrat în aplicație, așa cum sugerează și numele său.

Această pagină este exact pagina folosită și pentru partea aplicației destinată utilizatorului pacient descrisă anterior în subcapitlul 4.1.9. Am ales reutilizarea acestei paginii pentru ambele roluri destinate aplicației mobile deoarece acesta are același scop indiferent de rol și anume vizualizarea și editare informațiilor personale ale contului.

Accesul în pagina se face prin apăsarea butonului text “My profile” din meniu sau prin apăsarea butonului din dreapta sus a paginii principale.

### 5.2.9 Pagina My Doctor Details

Această pagina are ca scop, afișarea informațiilor suplimentare specifice contului unui doctor. Acesta este similara ca aspect cu pagina destinata informatiilor personale, însă nu există posibilitatea de editare a acestor informații, acestă funcționalitate fiind atribuită exclusiv utilizatorului administrator, din motive de siguranța.

Pagina My Doctor Details conține câmpuri text destinate următoarelor informații: categoria din care face parte doctorul, mai exact specializarea acestuia, anii de experiență, descriea de prezentare a acestuia și imaginea de profil.

În colțul stanga sus există un buton cu pictograma sugestiva care la apăsare direcționează utilizatorul către pagina principală a aplicației.

### 5.2.10 Pagina Reviews

Această pagină, așa cum reiese și din nume, este destinată review-rilor doctorului. Pagina conține o lista de review-uri și statistica cu privire la acestea, prezentate în mod detaliat in subcapitlul 4.1.5, in cazul acela reprezentânt un tab al paginii. În aplicația destinată doctorilor, aceste informații sunt prezentate în cadrul unei paginii special destinate, în mod exclusiv doar review-urilor.

Pagina descrisă are un rol important deoarece asigură doctorului o părere constant despre calitatea serviciilor furnizate de acesta, având constant o imagine clara a ce ar mai putea îmbunatății în actul său medical, desigur dacă este cazul.

## 5.3 Aplicația din perspectiva rolului “Administrator”

În acest capitol, vom explora în detaliu aplicația din perspectiva utilizatorului administrator și modul său de utilizare. Aplicația web este destinată exclusiv administratorului și poate fi accesată prin intermediul unui browser de pe dispozitivele care rulează sisteme de operare precum Windows, Android sau iOS.

Scopul principal al aplicației este de a oferi un mediu ușor și eficient pentru îndeplinirea sarcinilor administrative specifice unei clinici medicale. Fiecare funcționalitate este proiectată pentru a răspunde nevoilor specifice ale unui administrator într-un mod adecvat și practic.

Utilizatorul administrator beneficiază de următoarele funcționalități în cadrul aplicației:

* Se poate înregistra cu contul special destinate administratiei,
* Poate reseta parola contului,
* Poate vizualiza: lista de doctori, detaliile, reviewurile si programul fiecărui doctor, lista de categorii, lista de programări trecute si viitoare, lista de utilizatori și statistici bazate pe informațiile înregistrate cu ajutorul aplicației,
* Poate edita: detailiile doctorilor, programul doctorilor, programarile și categoriile.
* Poate șterge: doctori ,programe , categorii și programări.
* Poate adauga: doctori (inclusiv programul acestora), categorii, imagini.
* Poate filtra listele mai sus menționate cu ajutorul unor bare de căutare sau a unor butoane.
* Se poate deloga

În continuare voi prezenta fiecare pagina a site-ului web, dar și meniul, modul de utlizare și funcționalitătile menționate mai sus.

### 5.3.1 Pagina de autentificare

Această pagină este implicit cea cu care se deschide site-ul web și prin intermediul careia utlilizatorul se poate conecta.

Autentificare în site, la fel ca în cazul aplicației mobile pentru doctori și pacienți, se face cu ajotorul emailului și a parolei. Utilizatorul trebuie să introduca în castele textn destinate autentificării prezente în pagină, emailul și parola contului de administrator. După ce aceste informații au fost furnizate, la apasarea butonului text Log in prezent in partea drepată a ecranului, se încearcă autentificarea cu contul introdus. Aici pot exista doua cazuri: autentificare are succes și administrator este direcționat către pagina principala Dashbord sau autentificarea nu are succes iar pe ecran va apărea un mesaj explicativ despre motivul refuzării conectării. La fel ca la aplicația de telefon mobil pot exista mai multe motive pentru care nu s-a reușit autentificare în site : datele introduce sunt greșite, nu aparțin unui cont înregistrat sau nu aparțin unui cont de administrator.

Acestă pagină, cuprinde în mod similar cu aplicațiile descrise în subcapitolele 4.1 și 4.2 un buton text “Forgot your password?”, care face accesul către pagina de resetare a parolei.

### 5.3.2 Pagina de resetare a parolei

Această pagină, așa cum reiese și din nume este destinată resetării parolei contului de administrator. Această funcționalitate este la fel de utiliă pentru orice tip de utilizator, indiferent de rolul său pacient, doctor sau administrator funcționând în același mod pentru toate cele trei roluri. Interfața acestei paginii este diferită, însă detaliarea modului de utilizare din sectiunea 4.1.3 este valabilă și în cazul acestei pagini web.

### 5.3.3 Meniul aplicației

Meniul aplicației este o componeantă, mai specific un widget, prezent in fiecare pagină a aplicației,în partea stăngă pentru o navigare ușoară și eficientă in site. În acest fel utilizatorul poate vedea în permanența butoanele text cu cuvintele expresive care îi sugerea ce acțiuni poate performa, navigâng către pagina respectivă.

Astfel, meniul face accesul către toate paginile principale prezente în site. Cu ajutorul acestuia utilizatorul poate naviga către:

* Dashboard- pagina de home, pagina pe care se deschide situl dupa autentificarea cu scucces;
* Category- pagina care contine lista de categorii medicale din care doctorii fac parte și funcționalitătile specifice;
* Doctors- pagina care prezintă lista de doctori și care conține funcționalitătile dedicate acestei entităti;
* Programs- pagina care face accesul către programul de lucru a fiecărui doctor;
* Appointments- pagina care conține lista toturor programarilor trecute si vitoare, care permite de asemenea și diferite acțiuni asupra acestora;
* Reviews – pagina care face accesul catre reviewurile și statisticile bazate pe reviewurilor fiecărui doctor;
* Users – pagina care prezintă lista toturor utilizatorilor pacienți ai aplicatiei

### 5.3.4 Pagina Dashboard- pagina home

Pagina Dashboard reprezintă nucleul aplicației, oferind o vedere consolidată și intuitivă asupra informațiilor și activităților relevante în cadrul clinicii medicale. Această pagină centrală furnizează o prezentare vizuală și ușor de înțeles a datelor și metricilor cheie, permitând administratorului să analizeze și să urmărească activitățile cu eficiență. Pagina Dashboard pune la dispoziție diverse statistici și metrici, cum ar fi numărul de pacienți, programări, personal medical, și multe altele, oferind astfel o imagine de ansamblu asupra performanței clinicii. Este un instrument indispensabil pentru administratori, oferindu-le informațiile necesare pentru luarea deciziilor strategice și optimizarea activităților clinice.

Accesul în aceeasă pagină se poate face în doua moduri: la autentificare cu success a unui utilizator cu rol administrator, acesta este redirecționat către pagina principală Dashboard deoarece aceasta este pagina de home a apicației, sau la apasare butonului text Dashboard din meniul aplicației.

În partea de sus a paginii, sunt prezente 4 chenare care conțin următoarele date metrice: numărul doctorilor din clinică, numărul pacienților care au conturi în aplicație, numărul de programaări efctuate prin intermediul apicației și numarul de categorii/specializări medicale din care fac parte doctorii clinicii. Aceste patru chenare conțin date pe baza înregistrărilor din baza de date care se actualizează odată cu acesarea paginii.

****

Figura

Pagina cuprinde de asemenea și șase grafice cu statistici bazate pe informațiile înregistrate în baza de date. În continuare voi detalia fiecare grafic.

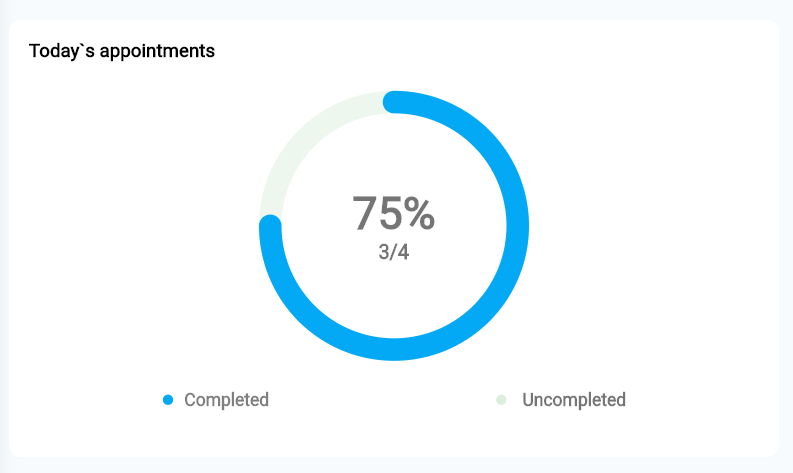
 

Figura Figura

În figura 5.() este un grafic care reprezintă progresul programărilor din ziua curentă. În interiorul graficului circular este specificat cât la sută din programările planificate in ziua curenta s-au realizat, bara circulară colorându-se cu albastru în funcție de acest procent. De asemenea este prezent în figură și raportul dintre câte programări au fost efectuate și totalul celor planificate. În partea de jos, există o legendă care specifica coloarea graficului pentru programările realizare și culoarea programărilor care încă nu s-au efectuat la momentul curent. În cazul în care nu exisă programări planificate în ziua curentă, în locul procentului și raportului se v-a afișa mesajul text “No appointments”.

Figura 5.() cuprinde graficul care prezintă programările efectuate în ultimele șapte zile, incluzând și ziua curentă. În partea de jos a graficului este specificat pentru fiecare vara verticală data, iar în partea de sus este specificat numărul de programări. Valoarea maximă de pe axa Oy, înălțimea barei verticale, este calculată în funcție de numărul maxim de programări al celor șapte zile. La apăsarea pe una dintre bare, textul de sus și cel de jos se vor colora în negru pentru a evidenția ziua aleasă.

Graficul fin figura 5.() este o diagramă circulară care reprezintă numărul de doctori în funcție de categoria sau specializarea medicală din care fac parte. În prima parte este prezent graficul circular, unde fiecare secțiune diferit colorată reprezintă o categorie fiind prezent pe fiecare dintre acestea numărul de doctori. În partea dreaptă se află legenda, care specifică aceleași detalii anterior menționate sub alt format. La apăsarea unei categorii din legendă, aceasta se deselectează și graficul se actualizează fără acea categorie.

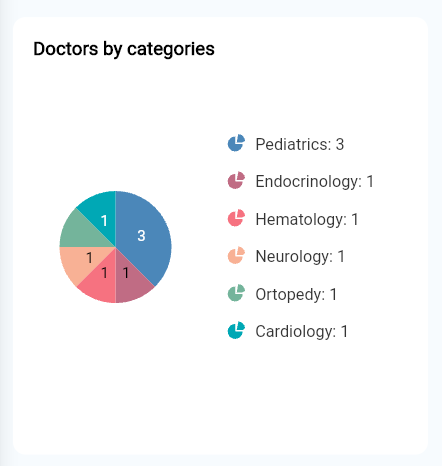
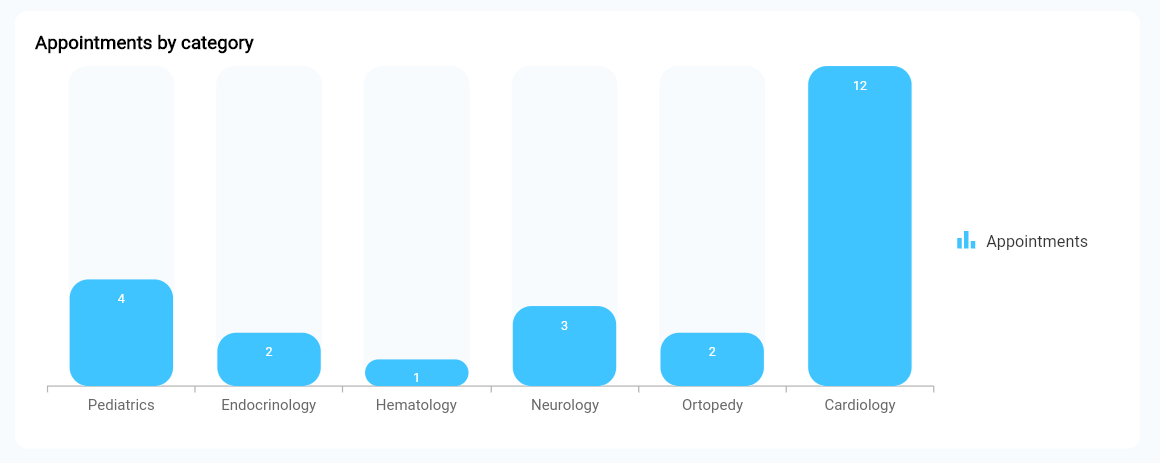
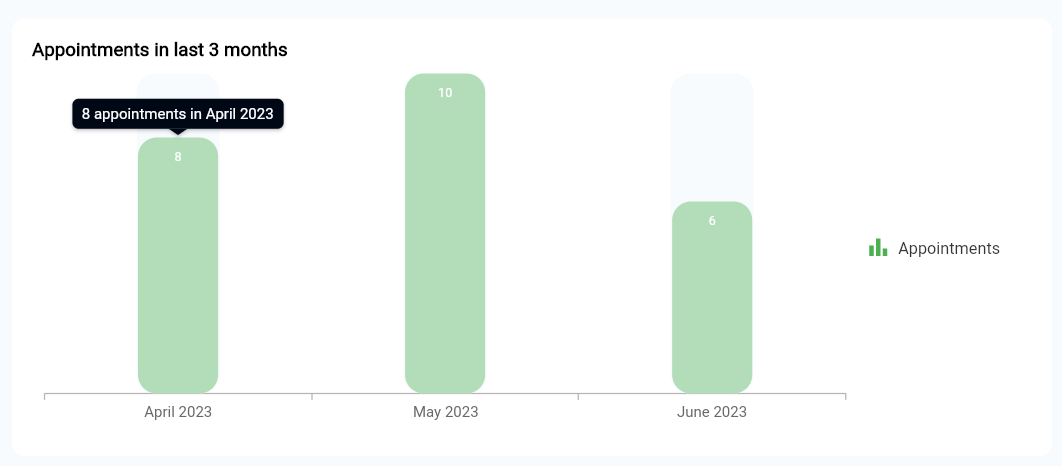
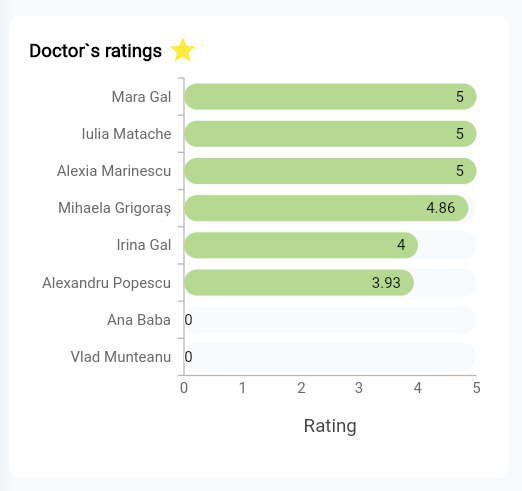
 

Figura Figura

În figura 5.() este prezentat graficul care ilustrează numărul de programări confirmate în funcție de fiecare categorie medicală. În partea de jos pe axa Ox sunt prezente numele tuturor categoriilor. Fiecăreia dintre aceste îi corespunde o bara verticală colorată cu albastru până la o înălțime calculată în funcție de numărul de programări din categorie și numărul maxim de programări înregistrate la una dintre categorii. De asemenea pe fiecare bara verticală este prezent și numărul de programări corespunzător. La trecerea cursorului peste o bara apare un mesaj care specifica catetegoria și numărul progrmărilor cu un mesaj text asemănător cu acesta: “2 appointments for Endocrinology”.



Figura

Pe ultimul rând al paginii principale, se afla cele două grafice corespunzătoare figurii 5.() și figurii 5.(). Primul dintre acestea reprezintă o diagramă cu bare orizontale destinate ratingului fiecărui doctor. Ratingul reprezintă media stelelor primite de fiecare doctor cu ajutorul review-urilor oferite de pacienții. Valoarea maximă a unui rating poate fi 5, fapt evidențiat pe axa Ox a graficului. Pe axa verticală sunt prezente numele fiecărui doctor, urmate de bara verticala colorată în funcție de rating. Datele doctorilor sunt afișate în mod descrescător în funcție de rating. În cazul în care un doctor nu are niciun review, ratingul acestuia va fi 0. La trecerea cu cursorul peste una dintre bare se va specifica cu ajutorul unui chenar numele doctorul și ratingul acestuia.

Cel de-al doilea grafic prezintă în mod asemănător graficului 5.(), numărul de programări efectuate în ultimele trei luni, inclusiv luna curentă.

### 5.3.5 Pagina Category

Această pagină cuprinde toate funționalitățile pe care administratorul le poate efectua asupra entității categorie.

La apăsarea butonului text Category din meniul aplicației, se deschide pagina prezentă in figura 4.(\*), cuprinzănd doar bara de căutare și tabelul din partea stăngă a imaginii. În această stare utilizatorul poate vizualiza sub forma de tabel toate categoriile medicale preznete în baza de date. Aceste categorii sunt afișate sub formă de tabel paginat, care poate conține câte zece categorii pe pagină. În cazul în care tabelul are mai mult de zece seturi de date, se poate naviga către celelalte cu ajutorul botoanelor “<” “>” de la finalul tabelului. Tabelul conține pe fiecare rând numărul de oridne, specific oricărui tabel, imaginea caracteristică categoriei, numele categoriei, un buton cu pictogramă pentru editarea categoriei și un buton cu pictogramă pentru ștergerea categoriei.

În partea de sus a paginii, este prezentă o bară de căutare cu ajutoul cărei, se poate filtra dupa numele lista de categorii din tabel. Astfel, dacă utilizatorul vrea să caute o categorie anume, cautând numele complet, dacă acesta există va rămâne singură în tabel, în caz contrar tabelul v-a rămâne gol. Bara de căutare suportă și căutarea parțială după nume, în tabel rămânând doar rezultatele care conțin textul căutat.

Pentru a adăuga o nouă categorie, utilizatorul trebuie să apse pe butonul text “Add a new category”. La apăsare pagina v-a intra în modul de adăugare și în partea deapta a ecranului va apărea formularul de adăugare ca în figura 4.(\*). În acest formular utilizatorul trebuie să completeze numele, în chenarul special destinat. In continuare utilizatorul trebuie să adauge o poza reprezentativă pentru acea categorie. Adugarea pozei se face cu ajutorul butonului text “Select Image”, care la apăsare va deschide fereastra specifică sitemului de operare care face acces la fișierele calculatorului. De aici utilizatorul poate cauta în mod obișnuit prin fișiere sale din computer imaginea dorită încărcarea acesteia se face cu ajutorul butonului text ”Open”. După apăsarea acestui buton fereastra se închide, și în locul imaginii generice din formular v-a apărea imaginea selectă. Pentru adăugarea categoriei și salvarea acesteia, ulterior completării informațiilor necesare, este necesară acționarea butonului text “Save”. Pentru adăugarea cu succes a unei categorii, este necesară completarea numelui, care trebuie să aiba intre trei și 25 de caractere. Dacă utilizatorul nu selecteză o imagine, implicit imaginea de referintă prezentă în pagină v-a fi adăugată. Dacă utilizatorul nu respectă linitele de caractere ale numelui acest fapt v-a fi semnalat cu ajutorul unui mesaj rosu sugestiv sub butonul de salvare, iar categoria nu se va adăuga. La abăsarea butonului text “Cancel”, pagina va revenii la modul de vizualizare și formularul v-a dispărea.

La apăsarea butonului cu pictogramă creion prezent în tabel, pagina v-a intra în modul de editare, formularul va reapărea în pagină de data aceasta precompletat cu datele categoriei pe care vrem să o edităm. Aici utilizatorul poate schimba numele sau imaginea și poate salva modificările la fel ca în cazul adăugarii descries mai sus. Prin apăsarea butonului se salvare, categoria moficicată se va actualiza în baza de date. Butonul text “Cancel” aduce înapoi pagina în modul de vizualizare.

La apăsarea butonului cu pictogramă roșu din tabel, se performează acțiunea de stergere a categoriei. În cazul în care categoria mai are doctori care fac partea din ea, în baza de date, pe ecran apare un chenar de dialog care specifică numărul de doctori care aparțin acelei categorii și faptul ca nu se poate realiza ștergerea dacă mai există doctori. În caz contrar daca în acea categorie nu se află niciun doctor ștergerea se poate realiza.

### 5.3.6 Pagina Doctors

În această pagină se face accesul către toate funcționalitățile de vizualizare, adăugare, editare și ștergere a doctorilor mai exact a detaliilor acestora, a programului și a review-rilor.

Această pagină cuprinde lista completă de doctori sub formă de tabel paginat, cu câte 10 doctori pe pagină. Fiecare rând al tabelului conține următoarele informații: nume, prenume, număr de telefon și categoria din care doctorul face parte. De asemenea pe fiecare rând este prezent un buton cu pictogramă, intitulat “View more” care prin păsare duce la pagina completa cu toate detaliile doctorului, pagina de profil acestuia.

Pagina conține în partea de sus o bară de căutare, cu ajutorul căreia administratorul poate căuta după nume și prenume, un doctor din tabel. În antetul tabelului este prezent un meniu ”Drop-down button” care cuprinde butoane text asociate fiecărei categorii inclusiv un buton care reprezintă toate categoriile. Prin apăsarea unui buton din meniu se aplica asupra conținutului tabelului filtru în funcție de categoria aleasă. Tot în antetul tabelului, în partea dreaptă există butonul text “Add new doctor”, care la apăsare direcționeză utilizatorul către pagina de adăugare a unui doctor.

### 5.3.7 Pagina de adăugare doctor

Aceasă pagină este creată pentru adăugarea unui nou doctor, în trei pași cu ajutorul unui widget “Stepper”. În partea de sus a paginii sunt mentionații cei trei pași, cu numarul fiecăruia și denumirea care are rol informativ. Odată ce un pas a fost realizat și utilizatorul trece la următurul, numărul de sus al pasului este înlocuit cu o bifă pentru a urmării progresul.

Așa cum am menționat anterior, în subcapitlul 4.2, este necesar mai întâi ca doctorul să își creeze un cont obișnuit de utilizator cu ajutorul aplicației de telefon, pentru înregistrarea datelor personale.

În continuare administratorul trebuie să parcurgă umătoarele etapele pentru adăugare a unui doctor, etape care reprezintă un pas din “Stepper-ul” mai sus menționat:

* Pasul 1: Alegerea unui utilizator – în această etapă pe ecran este prezent un tabelul paginat cu toate conturile de utilizator obișnuit, tabel care conține informații persoanele (nume, prenume, număr de telefon, email), prin apăsarea butonului cu pictogramă “Select”, prezent pe fiecare rând al tabelului, administratorul selectează utilizatorul pe care dorește să il facă medic în cadrul aplicației. Prin apăsarea butonului text “Next”, se face trecerea la următorul pas. Dacă nu se selecteză niciun utilizator, aplicația nu permite trecerea la pasul 2.
* Pasul 2: Adăugarea detaliilor doctorului – în această etapă pe pagina aplicației web apare un formular de adăugare a detaliilor doctorului. Cu ajutorul unor casete text administratorul trebuie să completeze următoarele informații: anii de experiență, categoria și descrierea. Aceste cămpurii sunt obiligarorii, iar necompletarea acestora împiedică utilizatorul să trecă la pasul umător. În partea dreaptă a paginii, utilizatorul are posibilitatea să adauge o imagine de profil a doctorului, în aceeași manieră descrisă în cazul categoriilor. Acesta nu este un câmp obligatoriu, în cazul în care o imagine nu este adăugată se vă selecta în mod implicit imaginea generica prezentă pe ecran.
* Pasul 3: Adăugarea programului- acest pas nu este obligaroriu, se poate complete și ulterior adăugări doctorului. În aceasă etapa, în partea dreptă avem un tabel paginat în care va apărea programul doctorului pe care il adăugăm, îar in partea stăngă formularul de adăugare. Fiecare rând din tabel, informație pe care o adăugăm cu ajutorul formularului, reprezintă o programare pe care o poate efecua doctorul în ziua respectivă care conține următoarele informații: ziua săptămânii, ora de începere a programării și ora de încheiere. Astfel în partea deaptă cu ajutorul unui meniu cu butoane “Drop-down” utlizatorul alege ziua, în continuare cu ajutorul a două widgeturi utilizatorul alege ora de început și ora de final. Pentru adăugarea acestor informatii in program este necesară acționarea butonului text “Add program”. Dacă utilizatorul dorește stergerea unui loc de programare din tabel poate acționa butonul cu pictograma “Delete” din dreptul acestuai. După ce utilizatorul a terminat de adăugat toate informațiile necesare, prin apăsarea butonului text “Add doctor” se deschide pe ecran un chenar de dialog care intreba utilizatorul dacă dorește adăugare acestui doctor. La apăsare butonului Yes se realizaează adugarea doctorului în baza de date și revenirea la pagina cu lista de doctori. În caz contrar, nu se realizează adăugarea.

### 5.3.8 Pagina de profil a doctorului

Această pagină cuprinde toate infromarțile despre doctorul ales din pagina cu lista de doctori, informații organizate cu ajutorul a trei tab-uri prezente în pagină, intitulate “Doctor details”, “Reviews” și “Program”. În continuare voi descrie fiecare tab în parte și funcționalitățile pe care le prezintă.

Tab-ul “Doctor details” cuprinde următoarele informații afișate cu ajutorul unor chenare text:

* Informații personale: nume, prenume, email, număr de telefon
* Detaliile doctorului: anii de experință, categoria și descierea acestuia
* Imaginea de profil

Informațiile personaele au doar caracter informativ și nu pot fi modificate de către administrator, moficarea acestora poate fi realizată doar de catre doctor prin intermediul aplicației pentru telefonul mobil.

Detaliile doctorului se pot edita prin apăsarea butonului edit din dreptul acestora, care va trece chenarele cu detalii în modul editabil. Aici utilizatorul poate modifca cămpurile și poate salva moficiările cu ajutorul butonului text “Save” apărut în locul butonui de editare.

Imaginea de profil a doctorului apare în partea dreptă unde este deasemenea prezent un buton “Edit” care permite selectarea și adăugarea altei poze. La acțiunea butonului “Save”, poza anterioară este ștearsă din baza de date și se adaugă noua poză selectată.

Tab-ul “Reviews” cuprinde o listă cu toate review-rile pe care doctorul respectiv le-a primit de la pacinenți, fiecare review fiind reprezentat de un chenar care cuprinde numărul de stele oferite, comentariul și data și ora la care acesta a fost adăugat. În această pagina este prezent și un widget pentru statistica cu privire la review-urile primite la fel ca cel prezentat în subcapitlolul 4.1 și 4.3. Această pagină are doar rol de vizualizare, fără a se putea efectua acțiunea de editare, adăugare sau ștergere.

Tab-ul “Program” conține programul complet al doctorului sub formă de tabel paginat cu 10 rănduri pe pagină, fiecare rând reprezentând un loc pentru o progrmare cu următoarele cămpuri: ziua săptămânii, ora de începere și ora de finalizare. La finalul fiecărui rând este prezent un buton cu pictogrmă, intitulat “Delete” care permite utilizatorului să șteargă un locul de programare din tabel. La apăsarea acestui buton, se deschide un chenar de dialog care intreabă utlilizatorul dacă dorește stergerea locului, avand posibilitatea să apese butonul “Yes” pentru ștergere și butonul “No” pentru anularea acțiuni.

Acestă pagină cuprinde și funcționalitatea de adăugare a locurilor în program. Prin apăsarea butonului text “Add” în pagina intra în modul de adăugare și în partea dreaptă a ecranului de deschide un formular pentru adăugarea unui loc nou. Acest folumular artă și functionază în mod asemănător cu cel descris anterilor la secțiunea despre adăugarea unui doctor, însă in acest caz locurile de programare se adugă unul căte unul în baza de date la acțiunea butonului “Save”,fără a se mai creea o listă intermediară.

### 5.3.9 Pagina programărilor

Acestă pagină conține lista completă a tuturor progrmarilor din clinică, programări trecute și vitoare, confirmate și neconfirmate. Pagina prezintă această listă sub formă de tabel paginat cu câte 10 programări pe pagină, fiecare programare din tabel având următoarele câmpuri: numele complet al doctorului, categoria doctorului, data și ora programării, numele complet al pacientului și numărul de telefon al acestuia.

De asemenea fiecare rând al tabelului conține două butoane, unul de confirmare și unul de ștergere. La acționarea butonului de confirmare, administraotorul confirmă venirea pacientului la progrmare și onorarea aceteia, deoarece pot exista și cazuti in care un pacient face o programare dar nu se prezintă la acesta. Aceasă confirmare a rol în numarătoarea exactă a cător programări au fost efectuate în clinica, prin intermediul aplicației. La apăsarea acstui buton care initial este gri, prin confirmare acesta devine verde pentru a oferi un feedback imediat. La apăsarea butonului de ștergere, butonul cu pictogramă “Delete”, în pagina web apare o casetă de dialog care intreabă utilizatorul daca dorește să performeze stergerea. Pentru conftimare, utilizatorul trebuie să avese pe buton text “Yes”, în caz contrar “No”.

În partea de sus a paginii sunt prezente câteva elemente care ajută la filtrarea listei de programări în funcție de necesitățile utilizatorului, elemente pe care le voi prezenta în continuare. Bara de căutare permite căutarea unei programări dupa numele și prenumele doctorului, complet sau partial. Este prezent un meniu “Drop-down button”, pentru filtrarea în funcție de categorie și unul pentru a alege daca lista afișează toate programările, sau doar cele trecute sau cele viitoare. Lista programărilor este ordonată în mod cronologic după dată și ora.

### 5.3.10 Pagina utilizatorilor pacienții

Pe această pagină administratorul poate vizualiza o listă a utilizatorilor pacienți, prezentată sub formă de tabel paginat, în care sunt afișate câte zece înregistrări pe pagină. Fiecare înregistrare conține informații ide baza despre pacient, precum numele complet, adresa de email și numărul de telefon.

Pagina conține de asemenea o bară de căutare care permite căutarea după nume, email sau număr de telefon.

Scopul principal al acestei liste este de a oferi administratorului acces rapid și ușor la datele de contact ale pacienților. Acest lucru se poate dovedi util în diferite situații, cum ar fi confirmarea programărilor sau reprogramarea unei rezervări. Având acces la aceste informații, administratorul poate comunica eficient cu pacienții în caz de nevoie.