**DOCUMENTAȚIA PROIECTULUI**

**DoctorHow**

STUDENȚI:

Radu Virgil-Andrei

Basarlici Narcis-Florin

**CUPRINS:**

[**I Prezentarea proiectului** 3](#_Toc43246905)

[**II Tehnologiile folosite(Android)** 5](#_Toc43246906)

[**III Baza de date** 7](#_Toc43246907)

[**iv Arhitectura Aplicatiei** 9](#_Toc43246908)

[**V Modul de Utilizare** 14](#_Toc43246909)

[**VI Concluzii** 21](#_Toc43246910)

[**VII Bibliografie** 23](#_Toc43246911)

# **I Prezentarea proiectului**

Într-un moment al umanitățîi în care aproape orice poate fi realizat de pe telefon, “DoctorHow” își propune să simplifice procesul prin care trece un pacient cu scopul de a-și face o programare la un doctor și totodată să ajute doctorii în a-și organiza mai bine progamul. Ultimele telefoane au făcut progrese dramatice în cee ace privește puterea de procesare, memoria RAM și împreună cu numărul de funcții suplimentare, cum ar fi accesul la internet, au deschis ușile către o gamă largă de aplicații.

          “DoctorHow” este o aplicație implementată de Radu Virgil-Andrei și Basarlici Narcis-Florin care oferă oricărui utilizator șansă de a caută și contacta doctori.Orice persoană care este bolnavă sau care dorește să viziteze un doctor pentru un control de rutină poate caută în aplicația DoctorHow doctorul potrivit și își poate face programare printr-un simplu apel către numărul afișat pe profilul doctorului. Dacă doctorul va anula programarea însă, pacientul va putea fi înștiințat doar printr-un apel din partea cabinetului sau prin prezența să la cabinet. Totodată, în aplicație este disponibil un profil complet a fiecărui medic ce conține detalii precum emailul, numărul de telefon și adresa spitalului la care activează.

          Prin urmare, această lucrare propune un mod foarte simplu, pentru orice om care are în posesia să un telefon, de a caută doctori. Aplicația Android “DoctorHow” conține două module.

Primul modul este alcătuit din partea de aplicație concepută pentru pacient. În acest modul există un ecran de “Login” și un buton care duce clientul la ecranul de “Register”.

Pentru a se putea conecta, clientul trebuie mai întâi să se înregistreze, introducandu-și în baza de date informațîi precum numele, email-ul și parolă dorită si numarul de telefon. După conectare, pacientului ii apare pe ecran o lista de doctori. El poate adaugă un doctor la favoriți sau poate să selecteze un doctor din lista, pe ecran apărându-I detaliile doctorului respectiv. Din ecranul principal, user-ul poate accesa un submeniu unde, prin apasarea unor butoane pot accesa lista doctorilor favoriți si profilul lor de utilizator. Tot în acest submeniu, user-ul poate vedea detaliile pe care acesta le-a completat la înregistrare. Din ecranul principal, userul poate filtra lista de anunturi, căutând anuntul dorit după titlu. Din profilul unui anunt utilizatorul are posibilitatea de a suna doctorul sau de a-i trimite un email.

          Al doilea modul, reprezintă partea de aplicație concepută pentru doctori. Inițial, doctorul trece prin aceleași procese că și userul, trebuind să se conecteze și dacă e cazul, să se înregistreze.Pe langa datele introduse de un utilizator client, doctorul beneficiaza si de un cod ce ii permite vizionarea unor elemente destinate lor din aplicatie Doctorul are acces la tot ce are acces un pacient, singură diferența fiind un buton in ecranul principal prin apăsarea cărui doctorului i se oferă ocazia de a adaugă un nou anunț. În acest anunț doctorul are posibilitatea de a-și adaugă un titlu, un conținut al anunțului și poze cu cabinetul în care lucrează. Din pagina anuntului, doctorul care a pus anuntul il poate sterge.

# **II Tehnologiile folosite(Android)**

   Android este un system de operare “open-source” bazat pe Linux conceput pentru dispozitive mobile cum ar fi telefoanele, tabletele sau chiar tevelizoarele. Android permite dezvoltatorilor să scrie cod gestionat în limbajul Java, controlând dispozitivul prin intermediul bibliotecilor Java dezvoltate de Google. Aplicațiile scrise în C și în alte limbaje pot fi compilate în cod mașînă ARM și executate, dar acest model de dezvoltare nu este sprijinit oficial de către Google.

Prima versiune beta de Android a fost lansată de Google în 2007, urmând că în septembrie 2008 să fie lansată prima versiune comercială, Android 1.0.  Începând cu 21 octombrie 2008, Android a fost disponibil că Open Source. Google a făcut disponibill întregul cod sursă (inclusiv suportul pentru rețea și telefonie), care urmă să devină indisponibil sub licență Apache.

Android este un sistem de operare puternic care concurează cu IOS al celor de la Apple. Printre multiplele avantaje oferite de către Android se numără următoarele :

* **“Beautiful UI”**
* **“Conectivitatea”**- GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE, NFC and WiMAX.
* **“Capacitatea de Stocare”**- SQLite, o bază de date relațională ușoară, este utilizata pentru stocarea datelor.
* **“Suport Media”**- H.263, H.264, MPEG-4 SP, AMR, AMR-WB, AAC, HE-AAC, AAC 5.1, MP3, MIDI, Ogg Vorbis, WAV, JPEG, PNG, GIF, and BMP.
* **“Sistemul de mesagerie”**- SMS and MMS
* **“Browser Web”**- Bazat pe WebKit, împreună cu motorul JavaScript Chrome V8 care acceptă HTML5 și CSS3.
* **“Multi-touch”**- Utilizatorul poate sări de la o sarcină la alta și, în același timp, diverse aplicații pot rula simultan.
* **“Wi-Fi Direct”**- O tehnologie care permite aplicațiilor să descopere și să se asocieze direct, printr-o conexiune peer-to-peer cu lățime mare de bandă.
* **“Android Beam”** -O tehnologie populară bazată pe NFC, care permite utilizatorilor să partajeze instantaneu, doar prin atingerea a două telefoane compatibile NFC.
* **“Uz multiplu”**-Utilizatorul poate sări de la o sarcină la alta și, în același timp, diverse aplicații pot rula simultan.
* **“Google Cloud Messaging”**- un serviciu care permite dezvoltatorilor să trimită date de mesaje scurte utilizatorilor lor pe dispozitivele Android, fără a avea nevoie de o soluție de sincronizare.

Aplicațiile Android sunt de obicei dezvoltate în limbajul Java folosind Android Software Development Kit. Odată dezvoltate, aplicațiile Android pot fi vândute ușor prin intermediul magazinelor online precum Google Play, SlideME, Opera Mobile Store, Mobango, F-droid and the Amazon Appstore.

Sistemul Android alimentează sute de milioane de dispozitive mobile în peste 190 de țări din întreaga lume. Este cea mai mare bază instalată a oricărei platforme mobile și se dezvoltă rapid. În fiecare zi, peste 1 milion de dispozitive Android noi sunt activate în toată lumea.

În concluzie, dacă cumva competita nu decide să înceapă să vândă produse mai ieftine de o calitate bună, ar sună rezonabil să prezicem că Android va continuă să domine piața de sisteme de operare mobile. Sistemul de operare este instalat pe telefoane care sunt vândute cu mult mai puțîn de 100 de dolari, până la dispozitivele cele mai scumpe precum actualul: Samsung Galaxy S10 Plus. Această flexibilitate, combinată cu actualizări anuale, va asigura că Android va rămâne lider în această industrie în anii următori.

# **III Baza de date**

Pentru construierea bazei de date am folosit Realm, un sistem de gestionare a unei baze de date open-source, folosit de către aplicații Android, Apple și chiar și de către aplicații desktop.

Realm, care aparținea celor de la Apache, a apărut pentru prima dată pe piață în Ianuarie 2017. În aprilie 2019, Realm a devenit propietatea celor de la MongoDB.

Baza de date pe care am creat-o are că scop evidență datelor unei aplicații Android ce oferă utilizatorilor posibilitatea de a caută un doctor și a-l contacta. Pentru această baza de date vom aveam nevoie de informații despre useri și informații despre doctori regăsite în anunțuri.

Pentru useri vom avea nevoie de informații cum ar fi un id, numele, prenumele, numărul de telefon, adresa de email și parolă pe care aceștia și-o setează și tipul de user. Pentru anunțurile doctorilor, id ,posesor, detalii, numărul de telefon, poze, adresa de email, titlu și adresa cabinetului în care activează acesta.

*Identificarea atributelor pentru useri :*

-**id** care este de tipul Integer

-**firstName** care este de tipul String

-**lastName** care este de tipul String

-**email** care este de tipul String

-**phone** care este de tipul String

-**password** care este de tipul String

-**type** care este de tipul UserType

*Identificarea atributelor pentru anunturile doctorilor :*

-**id** care este de tipul Integer

-**owner** care este de tipul User

-**details** care este de tipul String

-**phone** care este de tipul String

-**title** care este de tipul String

-**imagePaths** care este de tipul RealmList<String>

*Identificarea atributelor pentru codurile doctorilor :*

-**id** care este de tipul Integer

-**code** care este de tipul String

*Identificarea atributelor pentru tipul userilor :*

-**id** care este de tipul Integer

-**type** care este de tipul String

**Diagrama EER:**

# 

# 

# **Iv Arhitectura Aplicatiei**

“DoctorHow” este structurată în două Activități și șapte fragmente.

Prima activitate este “MainActivity” care comunica direct cu fragmentul de “Login” cu ajutorul Fragment Managerului.

 În layout-ul fragmentului de “Login” există două căsuțe text, una pentru introducerea adresei de email și una pentru introducerea parolei, un checkbox care odată apăsat permite utilizatorului să vadă parolă, o imagine și două butoane. Unul dintre butoane, butonul “Login”, îi aduce utilizatorului pe ecran cea de a două activitate, “HomePageActivity”, doar în cazul în care datele introduce sunt corecte și există în baza de date. Al doilea buton, butonul “Register”, îi aduce utilizatorului pe ecran fragmentul “Register” unde aceste își poate crea un cont nou. Pe fragmentul “Register” există șase căsuțe de tip text, una pentru nume, una pentru prenume, una pentru adresa de email, una pentru numărul de telefon, una pentru codul de doctor și una pentru parolă pe care userul o va alege. Pe lângă aceste căsuțe de tip text, mai există și două butoane, un buton “Back” care returnează utilizatorul în fragmentul de “Login” și un buton “Register” care îi crează utilizatorului un cont nou doar dacă datele introduse de către acesta sunt valide.

Cea de-a două activitate este “HomePageActivity” în care există un toolbar, un buton de căutare un buton ce îi permite ușerului să adauge un anunț și un fragment, “HomeDefaultFragment” în care apar anunțurile încărcate de doctori. În toolbarul menționat mai devreme există un buton ce deschide un NavigationView. În partea de sus a NavigationView există datele utilizatorului, adresa de email și numele, iar în partea de jus există opțiuni, precum “Favorites”, “Home”, “Profile”. Butonul “Favorites” duce utilizatorul la pagină cu doctorii pe care acesta i-a selectat drept favoriți. Butonul “Home” duce utilizatorul către ecranul principal acolo unde se află lista tuturor anunțurilor. Butonul “Profile” duce utilizatorul la pagină de profil a acestuia unde își poate vedea datele introduse la înregistrare, își poate modifică aceste date și se poate deconecta din cont.

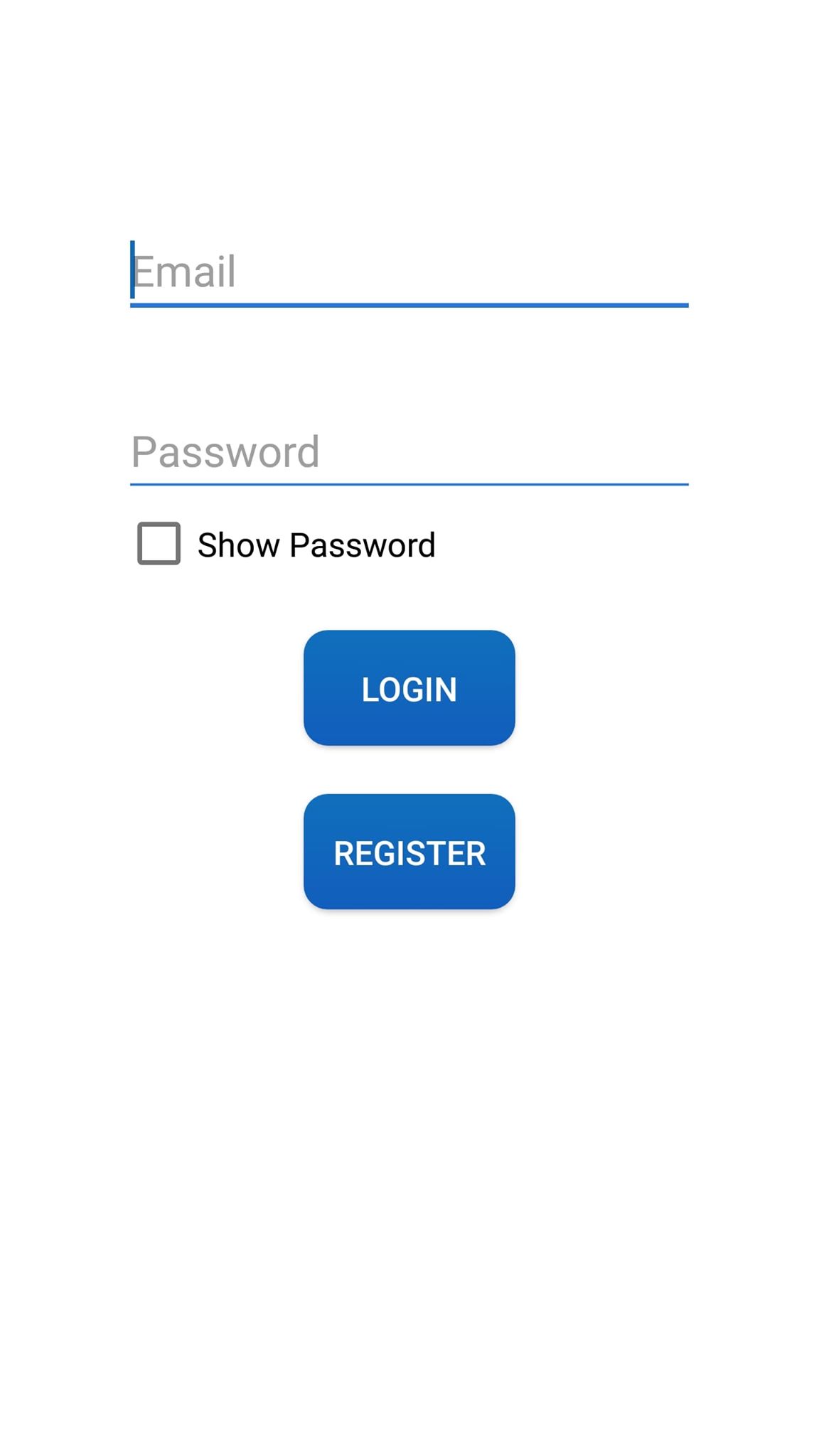
Butonul de search din toolbar permite utilizatorului să filtreze lista de anunțuri după titlu în timp ce butonul “+” din partea dreapta jos a aplicației, îi deschide doar utilizatorului care are cod de doctor fragmentul “NewListingFragment” în care acesta își poate crea un nou anunț. În acest fragment există 3 căsuțe de tip text, una pentru introducerea titlului anunțului, una pentru introducerea numărului de telefon și una pentru introducerea detaliilor. La baza ecranului există un buton “Pick Images” care oferă utilizatorului posibilitatea de a-și adaugă poze la anunț. În partea din dreapta jos a ecranului există un buton a cărui apăsare confirmă crearea anunțului și readuce utilizatorul la fragmentul anterior, doar dacă datele completate de utilizator sunt valide. La apăsarea unui anunț din lista de anunțuri, plasată în fragmentul “HomeDefaultFragment”, se deschide un nou fragment în care sunt afișate mai amănunțit informațiile acestui anunț. Din acest fragment utilizatorul are posibilitatea de a sună doctorul, prin apăsarea pe numărul de telefon deschizându-se aplicația de apeluri a telefonului, sau de a-i scrie un email acestuia, prin apăsarea pe email deschizându-se aplicația de email la pagină de creare a unui nou email. In aceeasi pagina, prin apasarea adresei, userul poate primi indicatii pentru a ajunge la cabinetul doctorului, deschizandu-se o pagina noua cu harta. In aceasta pagina cu harta, userul poate apasa pe butonul de maps, deschizandu-se astfel aplicatia Google Maps. Tot aici utilizatorul beneficiază de un buton la apăsarea căruia acesta își poate adaugă anunțul la favorite. Utilizatorii care dispun de codul de doctorul au în această pagină un buton în plus, butonul “Remove” prin care aceștia își pot șterge anunțul dacă doresc.

Pentru realizarea aplicatiei “DoctorHow” am folosit componente si servicii Android, precum :

* **LayoutInflater**- Instantiaza un fisier XML de layout in obiectele de tipul View corespunzatoare. Este folosit Activity.getLayoutInflater() pentru a prelua o instanta LayoutInflater care este deja conectata la contextual curent si configurata corect pentru dispozitivul pe care se executa.
* **RecycleView**- Este o versiune mai avansata si mai flexibila a ListView. In RecycleView mai multe componente diferite colaboreaza pentru a afisa datele dorite. RecycleView este completatat cu „views” primite de la „Layout Manager”.
* **LinearLayoutManager**- O implementare a “RecycleView.LayoutManager” care ofera functionalitate similara ca un “ListView”.
* **Floating Action Button**- Un buton circular care se plaseaza peste continut pentru a promova o actiune principala in aplicatie.
* **Toolbar-**Este un ViewGroup care poate fi plasat oriunde în aspectele XML.
* **DrawerLayout**-Actioneaza ca un container pentru continutul ferestrei, care permite extragerea unui “view” din una sau ambele margini verticale ale ferestrei.
* **NavigationView**-Este un mod user de a afisa un meniu de navigare. Este cel mai frecvent utilizat impreuna cu “DrawerLayout” .Sertarele de navigare sunt dialoguri ce pornesc din partea stanga a aplicatiei.
* **BitmapFactory**-Folosind-ul se pot crea imagini de tip bitmap in trei moduri: dintr-o resursa, dintr-un fisier sau un InputStream
* **FragmentActivity**- clasa de baza pentru activitati care doresc sa utilizeze Fragmente.
* **ImageView**- Clasa folosita pentru afisarea unei imagini in aplicatie
* **TextView**- este un control al interfetei utilizatorului utilizat pentru a seta si afisa textul utilizatorului
* **ViewGroup**- Este un “view” special ce contine alte “views” numite copii. ViewGroup este clasa de baza pentru “layouts” si “view containers”
* **DialogInterface**- Interfata care defineste o clasa de tip dialog ce poate fi afisata, respinsa, sau anulata si poate avea butoane.
* **AlertDialog**-Creaza o alerta de tip “alert dialog”
* **ActivityCompat**-Ajuta la accesarea functiilor din activitate intr-o maniera compatibila
* **ContextCompat**-Folosita atunci cand se doreste stabilirea de resurse cum ar fi “Drawable” sau “Color”
* **Environment**- ofera accesul la variabile de tip environment
* **MediaStore**-Contractul dintre furnizorul media si aplicatie. Contine definitii pentru URI
* **FileOutputStream**- Menit pentru scrierea fluxurilor “raw bytes” cum ar fi “image data”
* **SimpleDataFormat**-Clasa ce permite formatarea si analizarea datelor intr-o maniera sensibila la nivel local
* **Toast**-Un mesaj mic afisat pe ecran similar cu o notificare. Continutul mesajului ramane vizibil doar pentru putin timp.
* **Serializable**- Interfata standard Java.
* **ViewPager**- Este widgetul care permite utilizatorului să gliseze spre stânga sau spre dreapta pentru a vedea un ecran cu totul nou.
* **WindowManager**- Interfața pe care aplicațiile o folosesc pentru a discuta cu managerul de ferestre. Fiecare instanță a managerului de ferestre este legată de un anumit afișaj

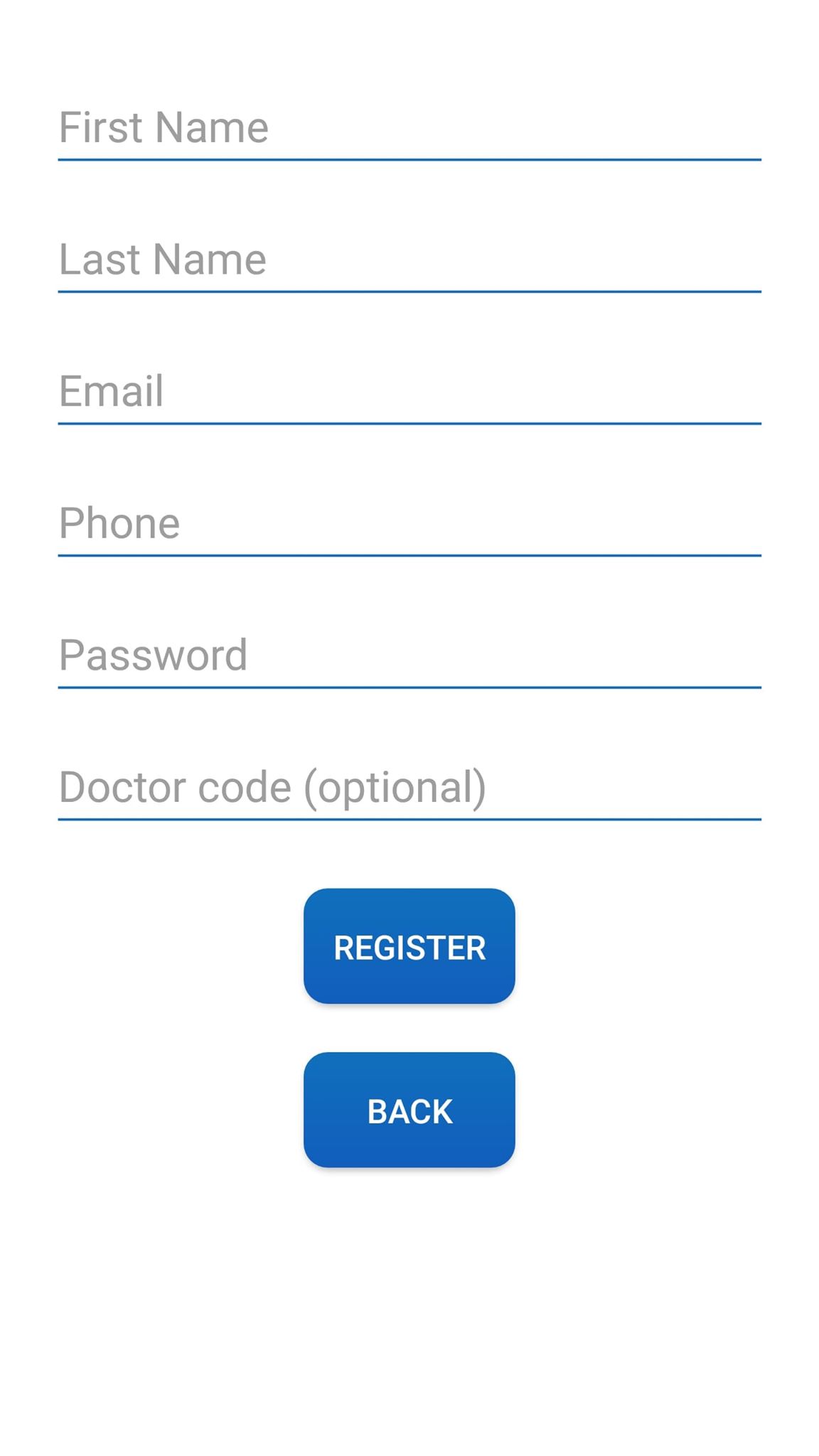
**V Modul de Utilizare**

**Login View:**

****

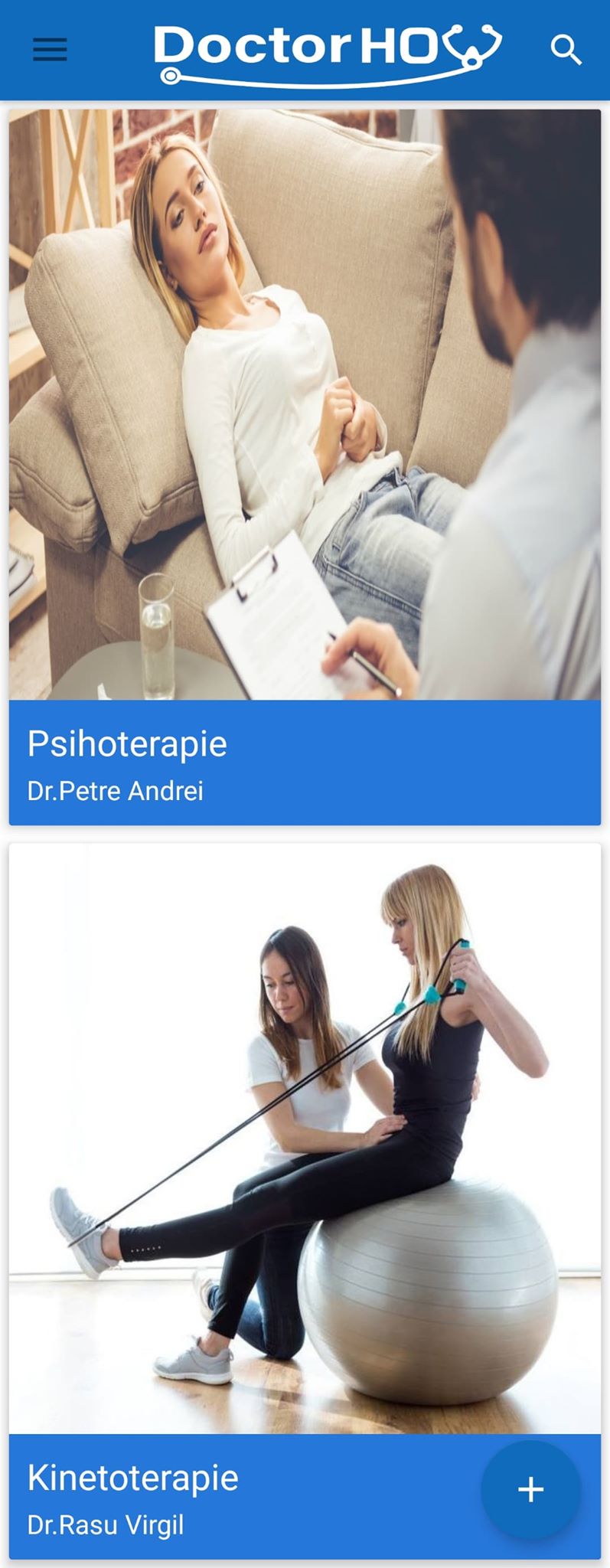
        Userul își poate introduce parolă și emailul și dacă ele sunt valide acesta va avea acces la pagină de start a aplicației. Tot de aici, userul poate să acceseze pagină de înregistrare prin apăsarea butonului “Register”.

**Register View:**

****

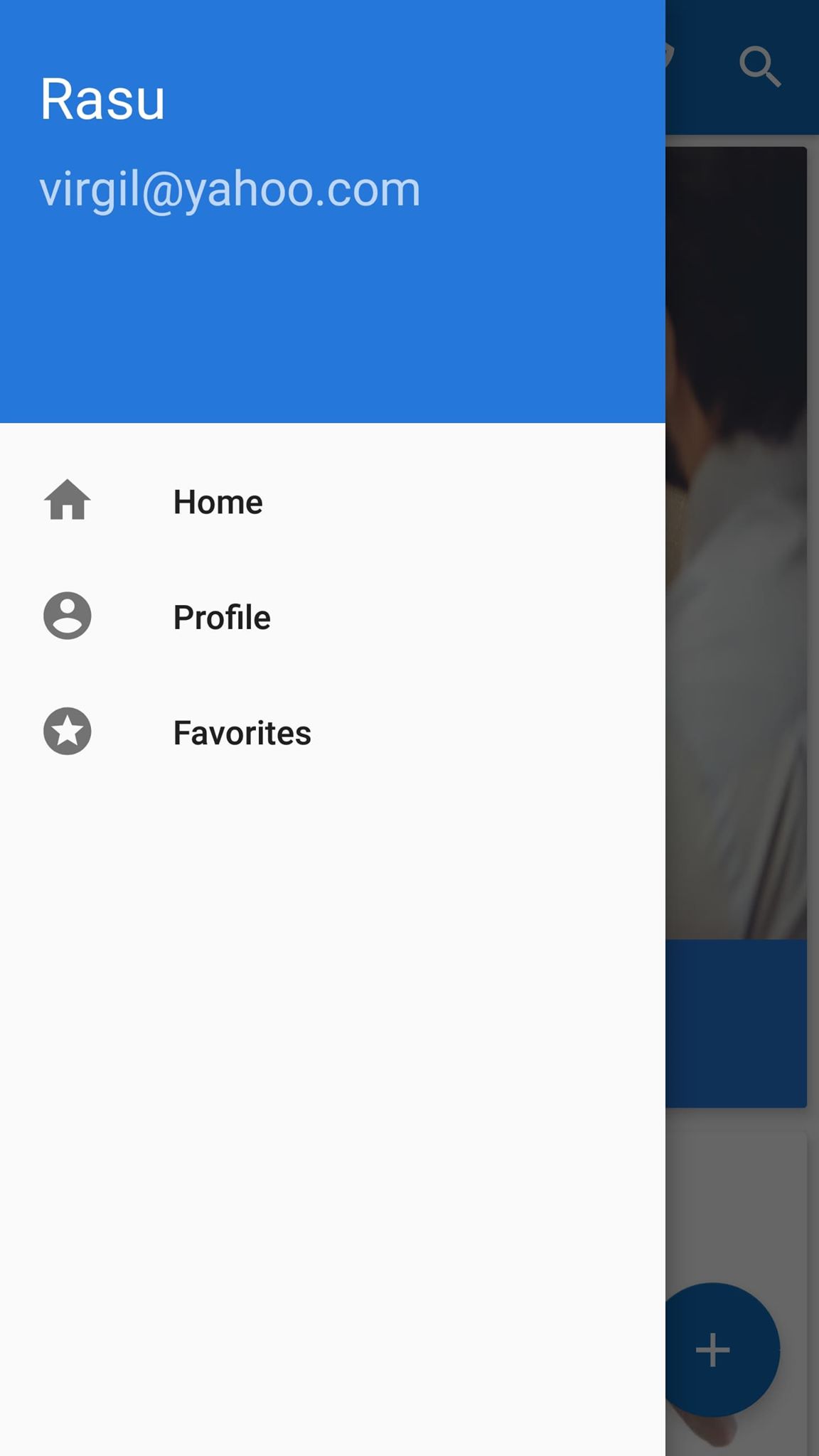
Aici userul își poate crea cont introducându-și datele și apăsând pe butonul “Register”, contul urmând să fie creat dacă datele introduce sunt valide.

**Home Page View:**

****

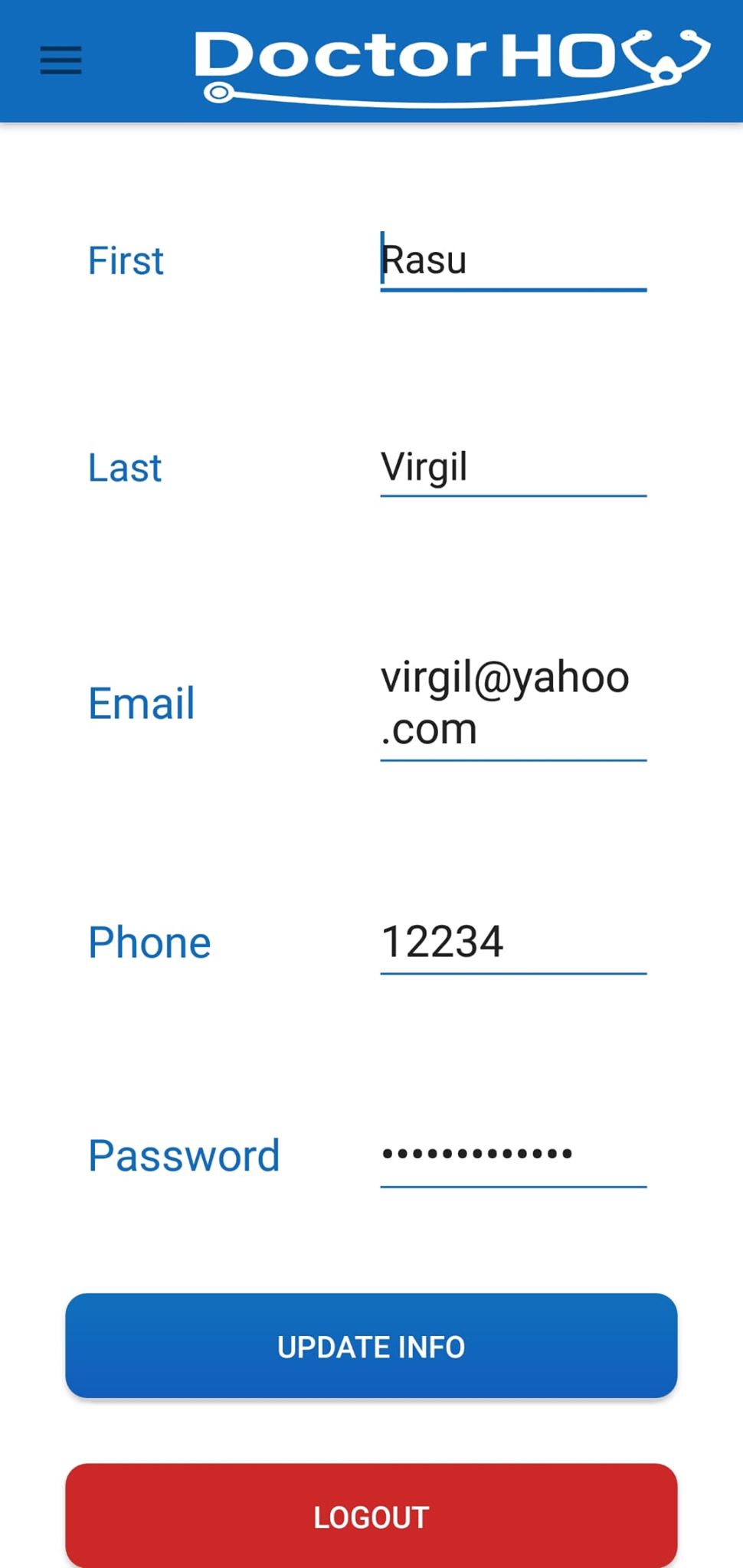
Aici userul are ocazia de a selecta doctorul dorit, de a caută un doctor după nume prin apăsarea pictograma “search”. Dacă userul care se loghează are cod de doctor va beneficia de butonul “+” prin care poate adaugă un nou anunț. Tot aici userul poate accesa un meniu prin apărea butonului din stânga sus al ecranului.

**Navigation Menu View:**

****

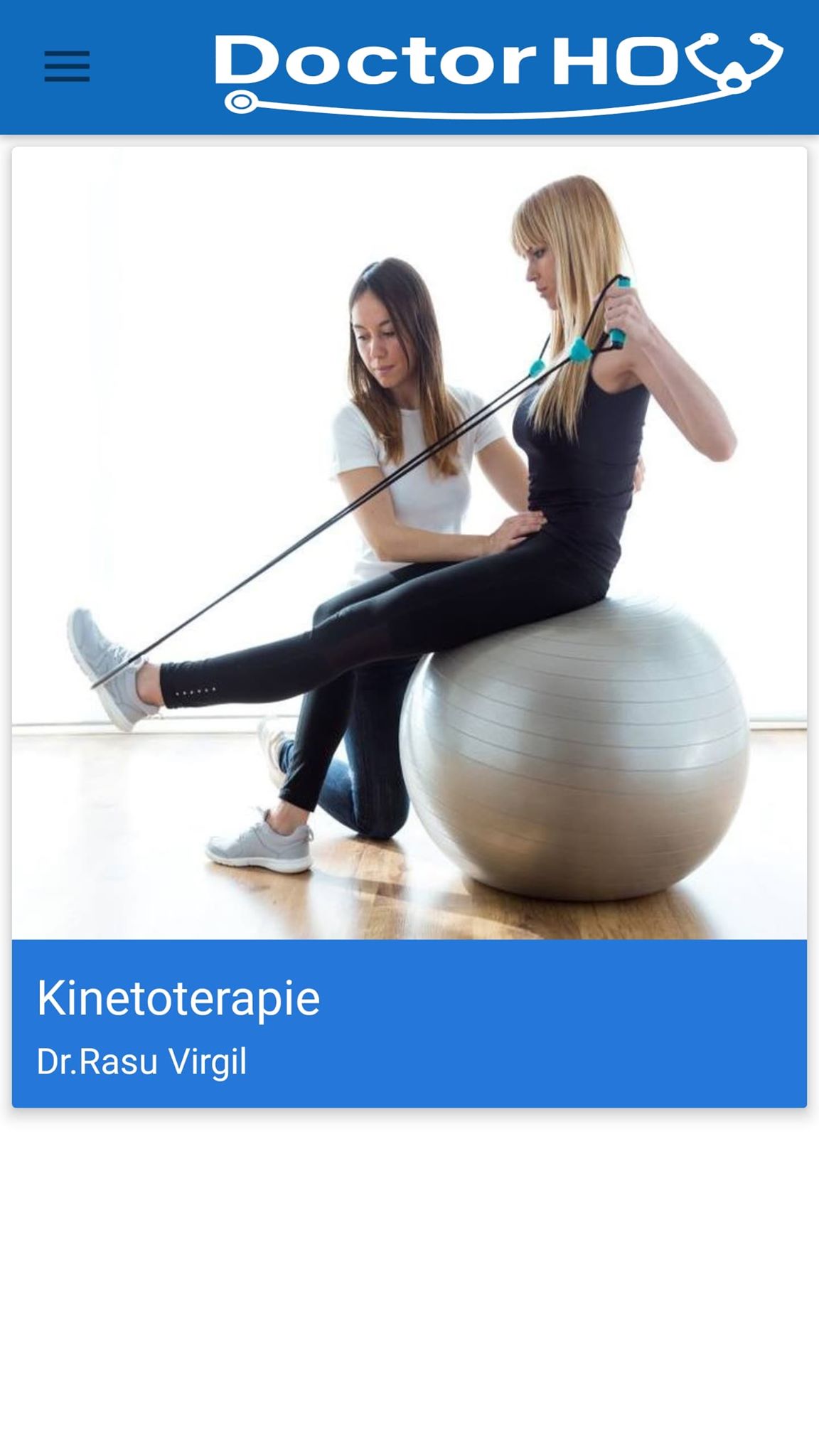
În acest meniu userul își poate vedea numele și adresa de email. Tot aici userul beneficiază de trei butoane. Butonul “Home” aduce userului “Home Page”, butonul “Profile” îl duce pe user la profil, iar butonul “Favorites” îl duce pe acesta la lista de favorite.

**Profile View:**



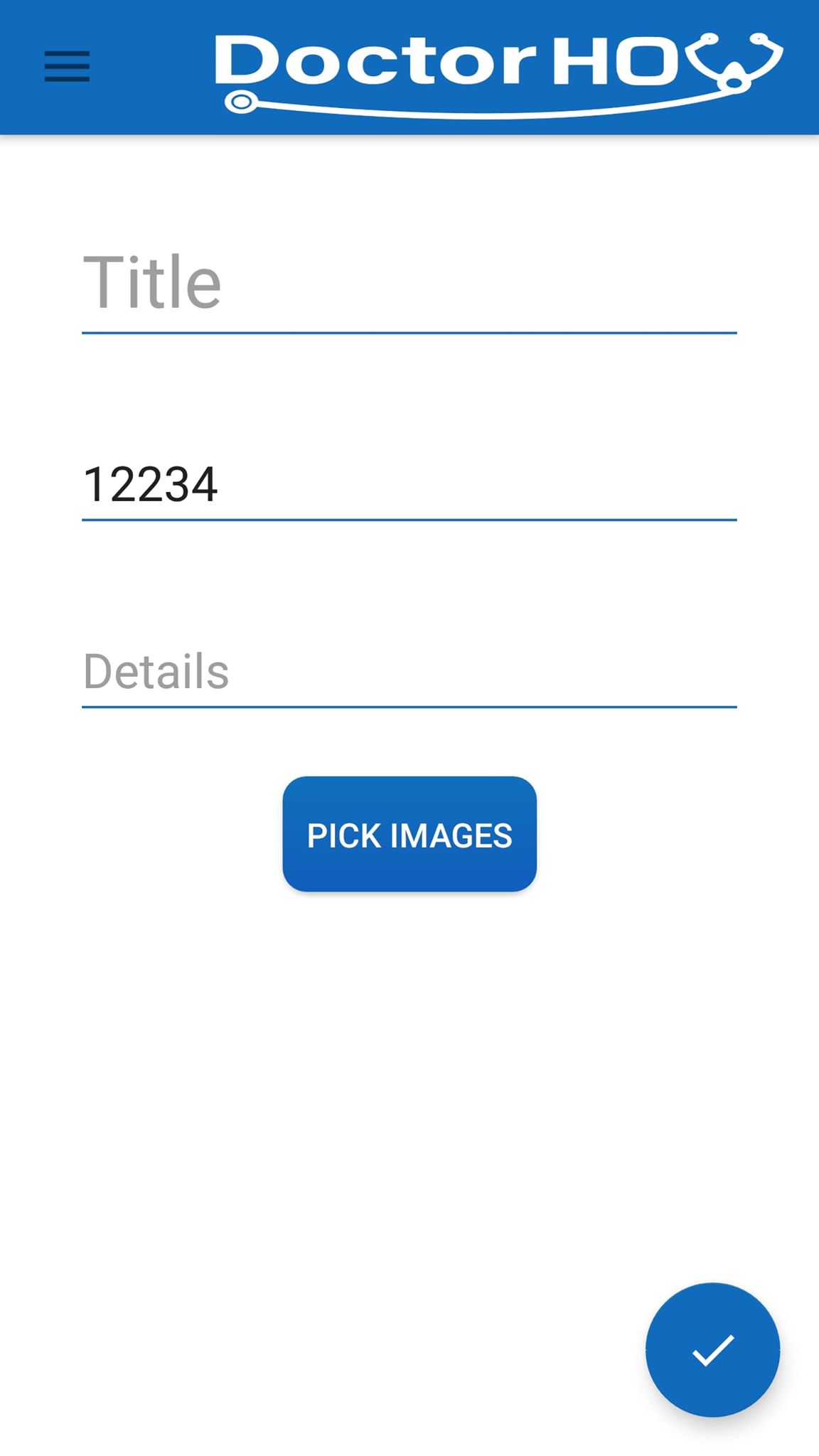
Aici userul își poate vedea datele introduse la înregistrare. El poate modifică aceste date dacă își introduce parolă corectă. Tot de aici userul se poate deloga.

**Favorites View:**

****

Aici userul poate vedea doctorii pe care i-a selectat că favoriți.

**Add Listing View:**

****

Această pagină este disponibilă doar userilor care au cod de doctor. De aici doctorii își pot adaugă titlu, număr de telefon, detalii și poze pentru anunțul lor.

**Listing View:**

****

În această pagină utilizatorul poate vedea detalii mai amănunțite despre anutul dorit. Tot aici au șansă de a adaugă la favorite un anunț prin apăsarea butonului “ADD TO FAVORITES”. Utilizatorii care au cod de doctor beneficiază de un buton în plus, butonul “REMOVE” prin care își pot șterge anunțul.

**VI Concluzii**

În concluzie, această aplicație implementată în Java și XAML ne-a oferit posibilitatea de a ne dezvoltă că și programatori și ocazia de a lucra și a jongla cu nenumăratele posibilități oferite de programarea android.

 În plus, suntem de părere că aplicația dezvoltată, cu puține îmbunătățiri și adăugări ar putea fi de folos atât pacienților ce nu ar mai fi nevoiți să își irosească timpul căutând un doctor, cât și doctorilor care și-ar putea organiza timpul și programul de lucru într-o manieră mai bună. Totodată, această aplicație este de folos străinilor într-un oraș pe care nu îl cunosc.

Un scop pentru viitorul acestei aplicații ar fi de a adaugă un modul de navigație în aplicație, astfel că utilizatorilor să li se prezinte drumul către cabinetul doctorilor. Un alt tel al acestei aplicații este de a adaugă un procedeu prin care utilizatorii să se poată programa online, fără a trebui să ia direct legătură cu doctoral sau un representant al acestuia.

# **VII Bibliografie**

1. <https://www.androidauthority.com/history-android-os-name-789433/>
2. <https://en.wikipedia.org/wiki/Realm_(database)#:~:text=Realm%20is%20an%20open%20source,licensed%20under%20the%20Apache%20License.>
3. <https://www.computerworld.com/article/3235946/android-versions-a-living-history-from-1-0-to-today.html>
4. <https://developer.android.com/>
5. <https://android-developers.googleblog.com/>
6. <https://www.scribd.com/document/337018850/Documentatie-Android-Project>