**MEDBUDDY**

**Candidat: Cotulbea Adrian-Ionuț**

**Coordonator științific: Asist.(SL/Lect./Conf./Prof.)dr.ing.(arh./ec./chim.) Prenume, Nume**

Sesiunea: Iunie/Iulie 2023

**REZUMAT**

Aplicația mobilă MedBuddy reprezintă o platformă revoluționară creată cu scopul de a simplifica modul prin care fiecare dintre noi poate obține o părere calificată oferită de unul dintre medicii specialiști, care s-au înscris și ei, la rândul lor, în această aplicație.

Majoritatea dintre noi ajungem să neglijăm, de cele mai multe ori, sănătatea personală până când aceasta se agravează din diferite motive din viața de zi cu zi, cum ar fi lipsa timpului, orarul haotic de la muncă, traficul din oraș sau chiar și mai rău, comoditatea.

MedBuddy vine cu o soluție rapidă și convenabilă, deoarece cu ajutorul acestei aplicații poți obține un diagnostic și un plan de tratament pentru simptomele pe care le manifesti în acel moment, indiferent de unde te afli, atât timp cât telefonul tău are conexiune la internet.

În cazul în care consideri că starea ta de sănătate necesită un consult față în față cu un medic specialist, aplicația îți oferă posibilitatea de a cere acest lucru comunității de doctori și să primești, ca răspuns, locul și numele medicului specialist care te va consulta, în momentul în care cererea ta a fost acceptată de acel medic.

**1. CUPRINS**

1. CUPRINS 3

2. INTRODUCERE 4

2.1 CONTEXT 4

2.2 DESCRIEREA PROIECTULUI 4

3. ANALIZA DOMENIULUI 6

3.1 APLICAȚII ASEMĂNĂTOARE 6

4. TEHNOLOGII FOLOSITE 7

4.1 KOTLIN 7

4.2 ANDROID STUDIO 7

4.3 C++ 7

4.4 VISUAL STUDIO CODE 7

4.5 DOCKER 7

4.6 MYSQL 7

4.7 VERSIONAREA CODULUI - GIT 7

4.8 [LIBRĂRII FOLOSITE](#_Toc105955609) 7

4.8.1 RETROFIT 7

4.8.2 ASIO 7

4.8.3 MySQL CONNECTOR/C++ 7

4.8 [LIBRĂRII FOLOSITE](#_Toc105955609) 7

4.8 [LIBRĂRII FOLOSITE](#_Toc105955609) 7

5. PROIECTAREA APLICATIEI 7

# 2. INTRODUCERE

Bine ați venit în aplicația Medbuddy, dezvoltată folosind Kotlin și C++.

Această aplicație este concepută pentru a oferi o platformă fără probleme și sigură pentru ca pacienții să se conecteze cu medicii lor și să-și administreze nevoile de îngrijire medicală. Cu această aplicație, pacienții pot adăuga și urmări ușor simptomele lor, pot alege un medic dintr-o listă de profesioniști calificați, pot accesa documentația medicală, pot primi un tratament, pot comunica prin chat în timp real cu medicul, pot primi feedback și pot seta memento-uri pentru medicamentele viitoare.

Motivația pentru alegerea acestei aplicații specifice este de a oferi o modalitate convenabilă și eficientă pentru pacienți de a comunica cu medicii lor și de a-și gestiona nevoile de îngrijire medicală. Medbuddy își propune să reducă decalajul dintre pacienți și medici prin furnizarea unei platforme în care pacienții pot adăuga și urmări ușor simptomele lor, pot alege un medic, pot accesa documentația med­­­­icală și pot seta mementouri pentru programări sau medicamente.

Aplicația Medbuddy este o combinație de funcționalități existente din diferite aplicații de îngrijire medicală și comunicare. Medbuddy are multe funcționalități pe care nu le avem într-o singură aplicație. Prin implementarea acestui proiect, aplicația își propune să ofere o modalitate mai convenabilă și eficientă pentru pacienți de a comunica cu medicii lor. Această aplicație poate ajuta pacienții să acceseze ușor informațiile lor medicale și să comunice cu medicii lor în mod prompt. În plus, Medbuddy poate contribui și la reducerea încărcării pe clinici și spitale, oferind pacienților o modalitate convenabilă și ușoară de a accesa informațiile de care au nevoie fără a fi necesară vizita la clinica sau spital. Ideea unui chat în timp real facilitează comunicarea între pacient și medic, după ce medicul a stabilit un tratament și pacientul are nevoie de mai multe informații.

**3.ANALIZA DOMENIULUI**

**3.1 APLICAȚII ASEMĂNĂTOARE:**

Trei dintre aplicațiile similare existente sunt "Practo", "Doctor on Demand" și "Talkspace".

"Practo" este o aplicație populară de sănătate disponibilă atât pe platforma iOS, cât și pe cea Android. Permite pacienților să caute medici și să își facă programări, să vizualizeze înregistrările medicale și să comande medicamente online. Una dintre caracteristicile cheie ale Practo este posibilitatea ca pacienții să caute și să își facă programări cu medici, precum și să își vizualizeze înregistrările medicale. Aplicația oferă, de asemenea, pacienților posibilitatea de a comanda medicamente online, ceea ce este o facilitate convenabilă. Cu toate acestea, Practo nu are funcția de chat, cerere și feedback între pacienți și medici pe care o propune aplicația dvs., lipsind, de asemenea, funcționalitățile de notificare și editare a profilului pentru medici.

“Doctor on Demand” este o aplicație de telemedicină disponibilă atât pe platforma iOS, cât și pe cea Android. Permite pacienților să aibă consultații virtuale cu medici, să vizualizeze înregistrările medicale și să primească tratament la distanță. Cu toate acestea, nu are funcții precum monitorizarea simptomelor, funcționalitatea de notificare sau o funcție de cerere pentru pacienți de a crea o programare.

“Talkspace” este o aplicație de sănătate mentală care oferă pacienților acces la terapie virtuală cu terapeuți licențiați. Utilizatorii pot accesa aplicația pentru a programa și gestiona sesiunile de terapie, primi notificări și urmări progresul lor. Cu toate acestea, nu are funcții precum monitorizarea simptomelor, funcționalitatea de notificare, cerere, chat în timp real și funcții de feedback între pacienți și medici, precum și funcția de editare a profilului pentru medici.

Un avantaj potențial al aplicației MedBuddy este includerea funcțiilor de chat și de feedback între pacienți și medici, ceea ce ar putea permite o comunicare mai eficientă și o mai bună gestionare a nevoilor de sănătate.

MedBuddy își propune să ofere un punct unic de gestionare a nevoilor de sănătate pentru pacienți și medici, și prin valorificarea avantajelor oferite de celelalte aplicații existente, să creeze o experiență completă pentru utilizatori.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Characteristics | [Practo](https://play.google.com/store/apps/details?id=com.practo.fabric) | [Doctor On Demand](https://play.google.com/store/apps/details?id=com.doctorondemand.android.patient) | [Talkspace](https://play.google.com/store/apps/details?id=com.talkspace.talkspaceapp) | MedBuddy |
| *Adresa site-ului web* | [Practo.com](https://www.practo.com/) | [Doctorondemand.com](https://doctorondemand.com/) | [Talkspace.com](https://www.talkspace.com/) | - |
| *Link magazin* | [Practo](https://play.google.com/store/apps/details?id=com.practo.fabric) | [Doctor On Demand](https://play.google.com/store/apps/details?id=com.doctorondemand.android.patient) | [Talkspace](https://play.google.com/store/apps/details?id=com.talkspace.talkspaceapp) | - |
| *Rating Magazin* | 4.4 / 5 | 4 / 5 | 2 / 5 | - |
| *Număr de instalări* | 10M+ | 1M+ | 500k+ | - |
| *Număr de instalări* | 300k | 40k | 6K | - |
| *Anunțuri/achiziții în aplicație* | x | x | x | - |
| Login/user | x | x | x | x |
| *Chat* | - | - | - | x |
| *Acces la înregistrările medicale* | x | x | x | x |
| *Funcții de feedback* | - | x | - | x |
| *Editare profil* | - | - | - | x |
| *Programare de consultații* | x | x | - | x |
| *Gestionarea medicamentelor* | x | - | x | x |
| *Notificări* | - | x | x | x |
| *Cerere de virtuală* | - | - | x | x |

**4. TEHNOLOGII FOLOSITE**

**4.1. Kotlin**

MedBuddy fiind o aplicație mobile a trebuit sa aleg un limbaj de programare cât mai avantajos pentru dezvoltarea acesteia, așa ca am ales Kotlin.

Kotlin este un limbaj de programare modern, concis și static, care rulează pe JVM (Java Virtual Machine) și este complet interoperabil cu Java. A fost dezvoltat de JetBrains și a câștigat popularitate rapidă în comunitatea dezvoltatorilor datorită sintaxei clare și expresive, siguranței la tip și caracteristicilor moderne. Kotlin oferă o sintaxă concisă și expresivă, reducând cantitatea de cod necesar pentru a realiza aceeași funcționalitate comparativ cu Java. De asemenea, aduce îmbunătățiri în siguranța la tip, eliminând erorile de tip la timpul de execuție și gestionând mai eficient valorile null prin conceptul de tipuri nullable și non-nullable. Un alt avantaj major al lui Kotlin este interoperabilitatea completă cu Java, permițând integrarea ușoară a codului Kotlin în proiectele Java existente. De asemenea, oferă suport pentru funcții moderne, precum funcții de ordin superior și lambda expressions, facilitând dezvoltarea de cod clar și concis. <https://kotlinlang.org/>

**4.2. Android Studio**

Mediul de dezvoltare ales pentru dezvoltarea aplicației mobile este Android studio, deoarece acesta ofera foarte multe utilitati incorporate în interfata lui, cum ar fi dezvoltarea fisierelor XML care reprezintă scenele și design-ul aplicației sau emulatoarele de dispozitive mobile virtuale, unde ai de ales dintr-o varietate de modele de telefoane sau tablete, pe care poți rula și testa aplicația.

<https://developer.android.com/studio>  
 Android Studio este un mediu de dezvoltare integrat (IDE) creat special pentru dezvoltarea aplicațiilor Android. Dezvoltat de Google, Android Studio oferă un set complet de instrumente și funcționalități necesare pentru a crea, testa și distribui aplicații Android de înaltă calitate. IDE-ul dispune de o interfață prietenoasă și ușor de utilizat, precum și de funcționalități avansate, cum ar fi un editor de cod, depanator și emulator Android integrat. Unul dintre avantajele sale majore este suportul nativ pentru limbajul Kotlin, permițând dezvoltatorilor să utilizeze Kotlin pentru dezvoltarea aplicațiilor Android într-un mod mai concis și mai sigur. Android Studio integrează, de asemenea, serviciile Google Play, facilitând implementarea funcționalităților precum autentificarea cu contul Google și integrarea cu serviciile Google.

**4.3. C++**

Partea de server a lucrarii mele de licenta a fost realizata în C++, deoarece acest limbaj de programare este renumit ca fiind eficient din punct de vedere al costului timpului și al resurselor, folosit deseori în dezvoltarea de servere.

C++ este un limbaj de programare general, puternic și versatil, bazat pe limbajul C. A fost dezvoltat în anii 1980 și este utilizat într-o varietate largă de aplicații, de la dezvoltarea de software de sistem până la jocuri și aplicații de înaltă performanță. C++ combină caracteristicile limbajului C cu adăugiri semnificative, cum ar fi programarea orientată pe obiecte, gestionarea dinamică a memoriei și suport pentru șabloane. Acesta oferă un control fin asupra resurselor și performanță înaltă, fiind preferat în domenii care necesită eficiență și control detaliat asupra hardware-ului.

C++ este un limbaj puternic de programare de nivel scăzut și oferă acces la operațiuni de nivel jos, cum ar fi manipularea directă a memoriei și controlul precis al resurselor. De asemenea, oferă suport pentru programarea orientată pe obiecte, permițând dezvoltatorilor să structureze și să organizeze codul în clase și obiecte. Limbajul C++ este cunoscut pentru performanța sa ridicată și este adesea utilizat în dezvoltarea de aplicații care necesită eficiență maximă și control granular.

<https://en.cppreference.com/w/cpp>

**4.4. Visual Studio Code**

Serverul a fost dezvoltat în editorul Visual Studio Code, deoarece am avut la dispoziție diferite unelte care au ajutat la livrarea unui progres considerabil și corect, din punct de vedere functional.

Visual Studio Code (VS Code) este un editor de cod sursă deschis, dezvoltat de Microsoft. Este unul dintre cele mai populare și versatile instrumente de dezvoltare disponibile, potrivit pentru dezvoltarea aplicațiilor într-o gamă largă de limbaje de programare. VS Code se remarcă prin simplitatea și flexibilitatea sa, oferind o interfață intuitivă și o gamă largă de extensii și plugin-uri care pot fi personalizate pentru a se potrivi nevoilor fiecărui dezvoltator.

<https://code.visualstudio.com/>

**4.5. Docker**

Am ales sa folosesc docker pentru a definii un container(un Dockerfile) pentru serverul C++, deoarece acesta este inclus alături de o imagine docker mysql server într-un fisier yml (docker-compose), cu ajutorul caruia pot sa pornesc toată functionalitatea server

cu o singura comanda.

Docker este o platformă de virtualizare la nivel de sistem de operare, care permite împachetarea și distribuirea aplicațiilor în containere portabile și ușoare. Un container Docker conține toate componentele necesare pentru rularea unei aplicații, inclusiv codul sursă, bibliotecile și dependențele, într-un mod izolat de sistemul gazdă. Aceasta facilitează reproducerea și distribuirea consistentă a aplicațiilor pe diferite medii, indiferent de configurația sistemului gazdă.

<https://www.docker.com/what-docker>

**4.6. MySQL**

Server-ul C++ se conecteaza la o baza de date MySQL, unde sunt definite mai multe tabele care servesc nevoile aplicației client MedBuddy. Acest server MySQL este configurat și pornit cu ajutorul lui Docker și o imagine de MySQL, cum am menționat și mai sus.

MySQL este un sistem de gestiune a bazelor de date relaționale, open-source și popular, care oferă o platformă robustă pentru stocarea și gestionarea datelor. Acesta utilizează modelul relațional, organizând datele în tabele cu rânduri și coloane. MySQL oferă performanță ridicată, scalabilitate și funcționalități avansate de securitate. Este susținut de o comunitate puternică și oferă acces la resurse și suport pentru dezvoltatori.

<https://www.mysql.com/>

**4.7. Versionarea codului – GIT**

Codul aplicației și al serverului au fost versionate cu ajutorul platformei GitHub, unde am făcut issue tracking și am folost ca mod de dezvoltare GitHub Flow, fiecare issue se rezolva/implementeaza pe un branch separat fata de branch-ul main.

GitHub este o platformă de gestionare a dezvoltării software bazată pe sistemul de control al versiunilor Git. Aceasta permite dezvoltatorilor să colaboreze la proiecte, să urmărească modificările de cod și să gestioneze fluxul de lucru al dezvoltării software.

<https://github.com/about>

**4.8.** [**LIBRĂRII FOLOSITE**](#_Toc105955609)

* + - 1. **Retrofit**

Retrofit a fost folosit pentru a consuma API-ul construit pentru a face schimbul de date dintre server și aplicație, apeland astfel request-urile HTTP.

Retrofit este o bibliotecă populară pentru dezvoltarea de aplicații Android în limbajul de programare Kotlin. Aceasta oferă o modalitate simplă și eficientă de a realiza cereri HTTP și de a consuma API-uri web în aplicații.

https://square.github.io/retrofit/

* + - 1. **Asio**

Librăria Asio standalone (non-boost) a fost folosită pentru a rezolva request handle-urile pe partea de server, astfel implementând logica server-ului.

Asio Standalone este o bibliotecă C++ open-source, care oferă un suport eficient pentru programarea asincronă și gestionarea evenimentelor de rețea. Aceasta este proiectată pentru a fi ușor de utilizat și scalabilă, oferind performanțe ridicate și o abordare modernă în dezvoltarea aplicațiilor de rețea.

<https://think-async.com/Asio/asio-1.28.0/doc/>

* + - 1. **MySQL Connector/C++**

Librăria MySQL Connector/C++ a fost folosită în implementarea server-ului. Cu ajutorul ei am conectat server-ul la baza de date SQL și am efectuat operațiile necesare pentru a insera, actualiza sau a prelua date.

MySQL Connector/C++ este o bibliotecă C++ care oferă suport pentru conectarea și interacțiunea cu bazele de date MySQL. Acesta facilitează dezvoltarea aplicațiilor C++ care utilizează MySQL ca sistem de gestiune a bazelor de date.

https://dev.mysql.com/doc/connector-cpp/

**5. PROIECTAREA APLICAȚIEI**