Mobilna aplikacija za potporu biomedicinskoj dijagnostici

Mikulić, Frane

Undergraduate thesis / Završni rad

2017

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:951380

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-29**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

zir.nsk.hr

Image not found or type unknown

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Sveučilišni preddiplomski studij računarstva

MOBILNA APLIKACIJA ZA POTPORU BIOMEDICINSKOJ DIJAGNOSTICI

Završni rad

Frane Mikulić

Osijek, 2017.

SADRŽAJ:

1. UVOD 1
   1.1. Zadatak završnog rada 1
2. RAZRADA IDEJE 2
   2.1. Medicinska dijagnostika 2
   2.2. Simptomatsko dijagnosticiranje 2
   2.3. Bolesti štitnjače i osnovno načelo razvoja aplikacije 2
   2.4. Model aplikacije 3
      2.4.1. Idejno rješenje aplikacije 3
   2.5. Operacijski sustav Android 5
   2.6. Android studio 6
      2.6.1. Android SDK 8
   2.7. Programski jezik Java 8
   2.8. Baza podataka SQLite 9
3. ANDROID APLIKACIJA THYROID'S APP 11
   3.1. Struktura aplikacije 11
   3.2. Programsko rješenje aplikacije 12
      3.2.1. Postupak dijagnosticiranja 19
4. TESTIRANJE I KORIŠTENJE APLIKACIJE 25
5. ZAKLJUČAK 27

LITERATURA 28

SAŽETAK 29

ABSTRACT 30

ŽIVOTOPIS 31

PRILOZI(na DVD-u) 32

1. UVOD

Tema ovog završnog rada je izrada mobilne aplikacije za potporu biomedicinskoj dijagnostici. Glavni cilj ovog projekta je omogućiti korisnicima aplikacije(oboljelim osobama) lakše dijagnosticiranje i analizu simptoma bolesti te ih samim time što bolje upoznati s mogućom bolesti. Aplikacija je u prvom planu namijenjena pacijentima koji bi se, odabirom određenih simptoma, usmjerili ka određenoj bolesti i po potrebi traženju stručne pomoći. S druge strane, aplikacija može poslužiti i stručnim osobama kako bi zajedno s pacijentom što lakše ustanovili o kojoj se bolesti radi. Ideja je da se korisnik (pacijent) prijavi u sustav, odabere određene simptome, a aplikacija mu daje moguću dijagnozu na osnovu odabranih simptoma.

Za uspješno ostvarenje projekta bilo je potrebno prikupiti znanja i potrebne informacije vezane za dijagnosticiranje u medicini s naglaskom na simptomatsko dijagnosticiranje. Kako je aplikacija namijenjena za što veću masu ljudi, izrađena je za mobilne uređaje s Android operacijskim sustavom, dok je za izradu baze podataka korišten sustav SQLite. Za kreiranje aplikacije korišteno je razvojno okruženje Android Studio. Budući da Android Studio ima već unaprijed ugrađenu podršku za SQLite, nije bilo potrebno instalirati dodatne programe za rad s bazom. Programiranje u Android Studiju temelji se na objektno orijentiranom programskom jeziku Java. Poglavlje 2 opisuje dijagnosticiranje u medicini općenito, nadalje je razvijena ideja o samom modelu aplikacije, nakon toga objašnjene su tehnologije koje će se koristiti pri izradi. U poglavlju 3 prikazana je programska realizacija aplikacije u Android studiju. Zaključno s programskom realizacijom obavljeno je prvo testiranje i korištenje o čemu će također biti riječi.

1.1. Zadatak završnog rada

Zadatak završnog rada je izraditi Android mobilnu aplikaciju za potporu biomedicinskoj dijagnostici koja će na osnovu odabranih simptoma korisniku dati što vjerodostojniju dijagnozu te je za korisnika pohraniti u bazu podataka. Aplikacija je izrađena u programskom okruženju Android Studio, pisana je programskim jezikom Java i sadrži bazu podataka razvijenu pomoću SQLite-a.

2. RAZRADA IDEJE

2.1. Medicinska dijagnostika

Dijagnoza je izraz koji se u medicini koristi za prepoznavanje određene bolesti. Dijagnozu postavlja liječnik na osnovu anamneze (dobivenih podataka o bolesniku i njegovoj bolesti), rezultata različitih morfoloških i funkcionalnih pretraga te na osnovu simptoma. Aplikacija će biti zasnovana na simptomatskom dijagnosticiranju koju će voditi sami korisnici s ciljem da što bolje razluče o kojoj je bolesti riječ.

2.2. Simptomatsko dijagnosticiranje

Simptomatsko dijagnosticiranje je vrsta dijagnosticiranja gdje se na osnovu izraženih simptoma na ljudskom tijelu i u ljudskom organizmu otkrivaju bolesti. Ovo načelo se u medicini koristi na samom početku liječenja gdje liječnik na osnovu nekakvih tjelesnih znakova pokušava odrediti o kojoj je vrsti bolesti riječ. Što više simptoma pacijent pokaže, to je liječniku lakše razlučiti i dati bolju dijagnozu. Ovakav tip liječenja ne daje pouzdanu dijagnozu, pogotovo ako pacijent skriva neke od simptoma, ali je izvrstan u samom razlučivanju u kojem smjeru treba ići s liječenjem [1].

2.3. Bolesti štitnjače i osnovno načelo razvoja aplikacije

Ideja za ovakvom aplikacijom dolazi iz slijedeće situacije. Kada pacijenti dolaze liječniku i trebaju mu dati informacije o tome kako se osjećaju, boli li ih što, primjećuju li kakvu promjenu na tijelu ili raspoloženju, mršave li ili pak gube tek. Postoji mogućnost da će pacijent ponekad i zatajiti određene simptome ili će ih jednostavno zaboraviti reći, a to bi liječnika moglo odvesti na krivi trag ili pak još više produžiti vrijeme dijagnosticiranja prave dijagnoze. Upravo zbog ovakvih situacija stvara se ovakva ideja o dijagnosticiranju. Pacijent je uvijek najiskreniji prema sebi i kada bi nasamo koristio aplikaciju i provjeravao određene simptome koje mu aplikaciju nudi, zasigurno bi išao ispravnim putem. Postoje i druge primjene i razlozi stvaranja ove aplikacije. Ponekad liječnici, zasuti svim informacijama o mnogobrojnim bolestima, ne mogu sasvim točno odrediti o kojoj je bolesti riječ dok ne pogledaju u svoje knjige ili zabilješke. Upravo zbog toga, ova aplikacija bi im mogla skratiti vrijeme u određivanju i dijagnosticiranju bolesti. Tako bi liječnik, zajedno s pacijentom, prošao kroz ovakav sustav simptomatskog dijagnosticiranja i odredio moguću dijagnozu ili samo olakšao put do pronalaska dijagnoze, što bi također bio iznimno značajan korak [2].

Kako bi navedeno bilo moguće, najprije treba dobro proučiti sastav svake bolesti, njene simptome, posljedice i uzroke kako bi sve te informacije mogli unijeti u aplikaciju. Također bi za sve to najprije trebala službena potvrda od strane ovlaštene bolnice ili liječnika da je aplikacija i sav njen sadržaj ispravan i da se može služiti široj populaciji. Budući da je izrada baze podataka, sa svim bolestima ljudskog organizma i svim mogućim informacijama vezanim uz pojedinu bolest, vrlo složen posao, ovaj projekt bit će posvećen hormonalnim poremećajima, konkretno poremećajima štitnjače. Odabrane su upravo ove bolesti jer su česte greške u njihovu razlučivanju, pogotovo kod pacijenata kada rade na vlastitu ruku [3].

Također, obavljeno je proučavanje postojećih aplikacija koje se bave različitim proučavanjem ove žlijezde. Prema [4] i [5], u Google Play trgovini postoje mobilne aplikacije za praćenje stanja bolesti štitnjače koje dijele savjete za eventualno liječenje te savjete za održavanje zdravog života. U odnosu na već postojeće aplikacije, ova mobilna aplikacija nudi posebne novosti u proučavanju štitnjače o kojima će se govoriti kroz rad.

2.4. Model aplikacije

2.4.1. Idejno rješenje aplikacije

U bolesti, odnosno poremećaje štitnjače, spadaju hipotireoza, hipertireoza, razni dobroćudni tumori (adenomi), te Grawes-Basedowljeva bolest. Uzmimo za primjer hipotireozu i hipertireozu. Hipertireoza je poremećaj metabolizma koji je posljedica prejakog izlučivanja hormona štitnjače odnosno pretjerane aktivnosti štitnjače. S druge strane hipotireoza predstavlja poremećaj metabolizma uzrokovan smanjenjem količine hormona štitnjače u tijelu, koji se očituje s brojnim simptomima. Simptomi su u oba slučaja mnogobrojni i često usko povezani. Sve su to simptomi s kojima se često srećemo u životu i koji vrlo lako mogu biti uzrokovani nečim drugim, no kada dolaze u određenim kombinacijama upućuju upravno na jednu od ove dvije bolesti. To su simptomi poput ubrzanog metabolizma, usporavanja metabolizma, česte stolice, nervoze, nemira, nepodnošenje topline odnosno hladnoće, gubitaka na težini, gubitaka teka i mnogi drugi. Dakle, cilj je ove aplikacije ponuditi korisniku sve moguće simptome koji se javljaju kod ovakvih i sličnih bolesti, korisnik bi odabrao one koji mu se javljaju, a aplikacija bi iščitala i ponudila određenu dijagnozu [6].

U daljnjem dijelu ovog rada bit će opisan model aplikacije, način rada te postupak njenog korištenja. Najprije bi se korisnik prijavio u sustav (prijava u sustav realizirana je preko Google računa, o čemu će više riječi biti u nastavku). Nakon toga, korisnik bio dobio ponuđene

simptome i odabrao bi one koji ga se tiču, potvrdio zabilješku, a aplikacija bi izbacila dijagnozu, dakle moguću bolest, te usmjerila na upute za liječenje i prevenciju što bi bilo detaljno opisano u profilima svake od bolesti u samoj aplikaciji. Ukoliko se odabrani simptomi ne tiču samo jedne bolesti, nego su odabrani različiti simptomi uzrokovani različitim bolestima, korisniku će biti prikazane i ostale mogućnosti i to u određenom postotku. Svaki simptom imao bi određeno težinu kojom bi utjecao na konačan rezultat, o čemu će također biti rečeno u nastavku. Upravo ta dijagnoza koju korisnik dobije bit će pohranjena u bazu podataka pod određenim imenom i datumom kada je obavljena. Ista dijagnoza korisniku će biti dostupna kod svake prijave u sustav. Slika 2.1 prikazuje slijed kojim bi korisnik koristio aplikaciju.



Slika 2.1. Prikaz slijeda korištenja aplikacije

Način rada aplikacije bit će prikazan i objašnjen u poglavlju 4.4. rada kada bude objašnjeno i programsko ostvarenje, a u daljnjem tekstu bit će opisane programske tehnologije koje su se koristile pri izradi ove mobilne aplikacije. Budući da su tema ovog rada bolesti uzrokovane hormonalnim poremećajem štitnjače, aplikacija je dobila naziv „Thyroid's app“ prema engleskoj riječi za štitnjaču.

2.4.2. Parametri bolesti korišteni u modelu

U tablici 2.1, u drugom stupcu, za svaku od navedene četiri bolesti, ispisani su parametri, odnosno simptomi, koji će se koristiti pri dijagnosticiranju. Također su ispisani i koeficijenti (težine) koju svaki pojedini simptom ima pri računanju vjerojatnosti. Kod hipertireoze sedam parametara ima vrijednost 0.09, a tri parametra vrijednost 0.045 iz razloga što se tri parametra ponavljaju i kod ostalih bolest. O parametrima i računanju njihovih koeficijenata bit će više riječi u poglavlju 3.2.

Tablica 2.1. Prikaz parametara i njihovih koeficijenata pri računanju vjerojatnosti

Bolest	Parametri	Koeficijent pri računanju
Hipotireoza	Dobivanje na težini, usporenost, pospanost, bezvoljnost, učestala hladnoća, česte opstipacije, blijeda koža , edemi, poremećaj rada srca	0.11
Hipertireoza	Gubitak na težini, povećani apetit, nervoza, nemir, nepodnošenje topline, pojačano znojenje, umor, grčevi u mišićima, česte stolice, menstrualne nepravilnost, guša	7 x 0.09 i 3 x 0.045
Adenom	Destrukcija žljezdanog tkiva, nakupljanje ožiljkastog tkiva, ciste, čvorovi, tumori	0.2
Grawes-Basedowljeva bolest	Gubitak težine, uznemirenost, razdražljivost, nepodnošenje topline,pretjerano znojenje, drhtanje, ubrzan puls, mišićna slabost, uvećanje štitne žlijezde, topla i vlažna koža	6 x 0.1 i 4 x 0.05

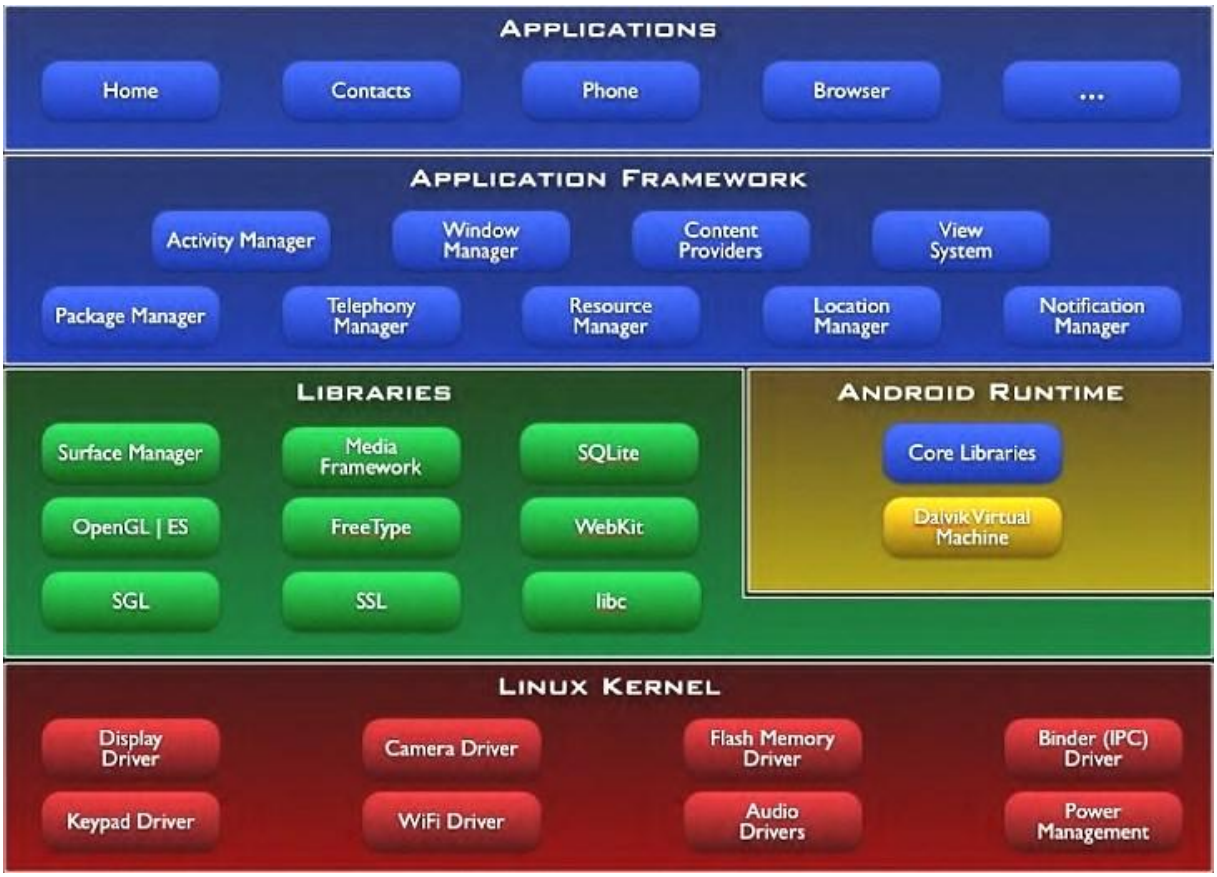
2.5. Operacijski sustav Android

Kao što je opisano u [7], Android je operacijski sustav za mobilne uređaje, otvorenog koda, koji je osnovan 2003. godine. Temelji se na Linux *kernelu* i napisan u programskom jeziku C/C++. Tvrtku Android Inc. koja je „udarila temelje“ ovom operacijskom sustavu, osnovali su Andy

Rubin, Rich Miner, Nick Sears i Chris White 2003. godine, a tvrtku je 2005. godine kupio Google i oformio ovaj operacijski sustav kakav možemo vidjeti i danas.. Zbog svoje lake prilagodbe, Android se u današnje vrijeme koristi na sve više vrsta uređaja poput televizora, satova, različitih ugradbenih uređaja pa čak i automobila. Korisničko sučelje Androida temelji se na izravnoj uspostavi kontakta koristeći dodire koji odgovaraju pokretima iz stvarnog vremena, kao što su povlačenje, dodirivanje, zakretanje zaslona i drugi. Ovaj operacijski sutav zasnovan je na Linux 2.6 jezgri koja je ujedno i prva razina operacijskog sustava a njezinu arhitekturu može se vidjeti na slici 2.2 preuzetoj iz [7]. Iznad jezgre nalaze se SGL, SQLite, WebKit i druge biblioteke. Slijedi *Android Runtime*, sloj koji se koristi pri otvaranju aplikacija, a temelji se na dva dijela. Prvi dio su *Core biblioteke* koje sadrže većinu jezgrenih knjižica ovog programskog jezika te *Dalvik Virtual Machine* (DVM) koji pokreće aplikacije kao samostalne procese, Slijedeći sloj je aplikacijski sloj (eng. *Application Framework*) koji je potreban za pisanje aplikacija te isto tako dozvoljava upotrebu API-ja (eng. *Application Programming Interface*). Na ovaj način omogućeno je upravljanje programskim paketima, pozivima, prozorima i resursima kao i dohvaćanje lokacije. Na vrhu se nalazi sloj koji je vidljiv korisniku i koji se sastoji od osnovnih i ugradbenih aplikacija te aplikacija koje se također nalaze u trgovini Google Play.

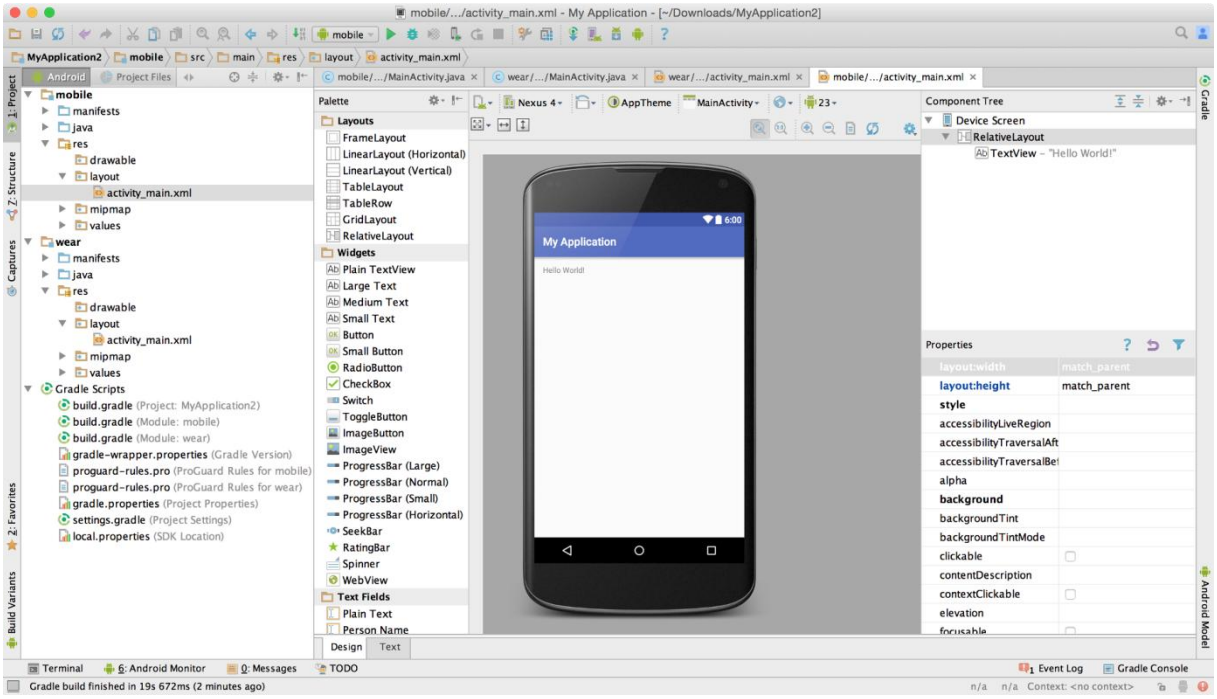
2.6. Android studio

Prema [8], Android Studio je službeno integrirano razvojno okruženje (*IDE*) namijenjeno za izradu Android mobilnih aplikacija. Android Studio prvotno je bio najavljen još u svibnju 2013. Dok je prva stabilna inačica izašla krajem 2014. Godine. Zasnivan na JetBrain's IntelliJ IDEA programu, Android Studio je dizajniran za razvoj aplikacija isključivo vezanih za Android operacijske sustave.



Slika 2.2. Arhitektura operacijskog sustava Android

Android Studio tako je zamijenio Eclipse koji je do taba bio rješenje za razvoj Android mobilnih aplikacija, a danas je dostupan za preuzimanje na Windows, Linux i Mac OS X operacijskim sutavima. Jedne od glavnih karakteristika ovog razvojnog okruženja svakako su podrška za *Gradle build* sustav, *Lint* alati za poboljšanje kvalitete koda, čarobnjak za kreiranje predložaka za uobičajne dizajne i komponente, korisničko sučelje koje omogućuje jednostavno uređivanje izgleda zaslona pomoću tzv. *drag and drop* načina rada, podrška za *Gradle build* sustav, ugrađena podrška za Google *Cloud* platformu, *Android Virutal Device* (emulator) koji služi za pokretanje aplikacije na virtualnom uređaju te ostale mnogobrojne značajke. Prvotna inačica bila je 1.0 , a ova aplikacija je rađena na inačici 2.3. Minimalne specifikacije računala potrebnog za pokretanje ove inačice Android Studija su: 2 GB RAM memorije, 500 MB prostora na disku za pohranu Android Studia te 1.5 GB za Android SDK i emulator, JDK (eng. *Java Development Kit*) 8 ili 6 te rezolucija zaslona 1280x800. Nadalje. Android Studio namijenjen je za sljedeće operacijeske sustave su Windows 7/8/10, Mac OS X 10.8.5 do 10.11.4, Gnome ili KDE.



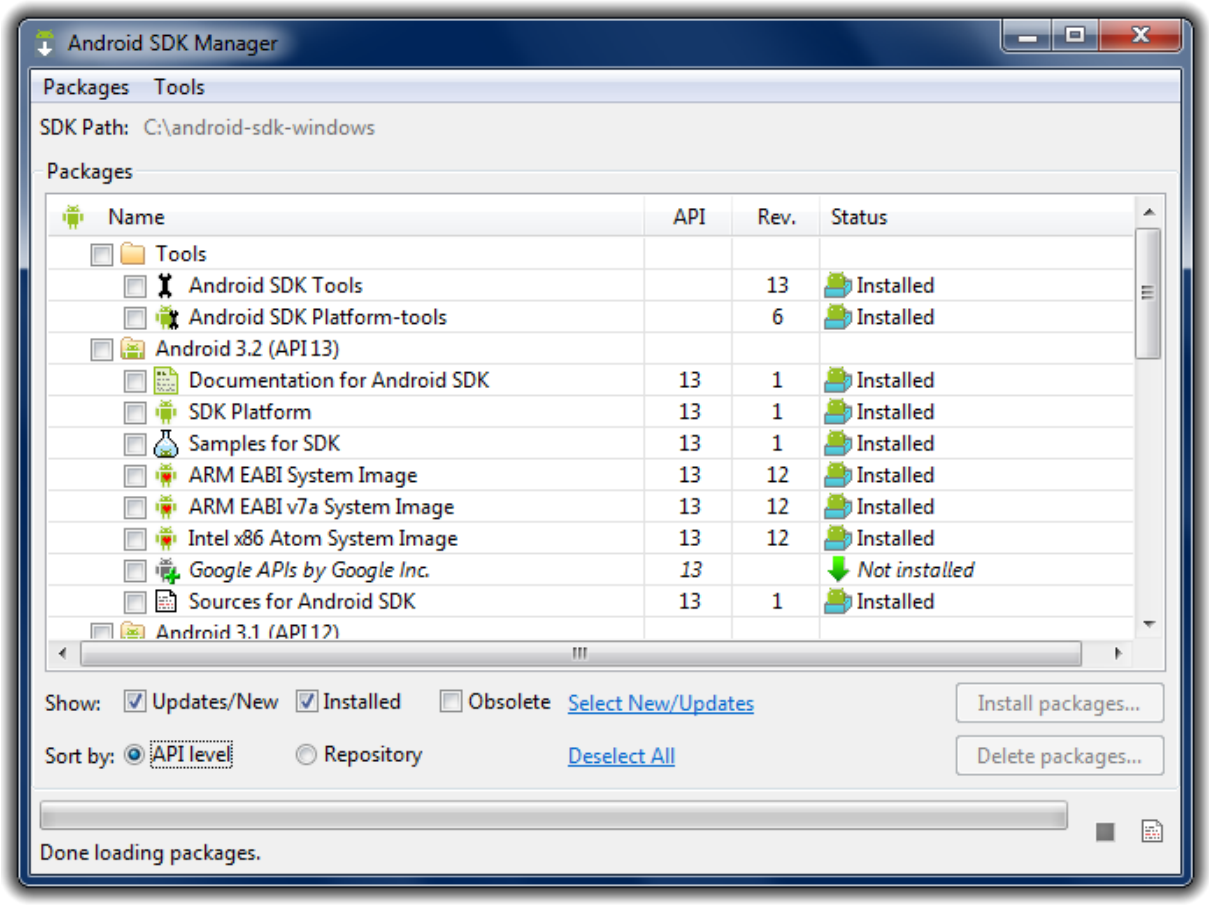
Slika 2.3. Izgled programskog sučelja Android Studia

2.6.1. Android SDK

Android SDK (eng. *Software Development Kit*) je skup alata koji su potrebni za razvoj aplikacije. Posjeduje program za otkrivanje pogrešaka u kodu (*debugging*), biblioteke, emulator, dokumentaciju i neke primjere koda. Razvojni alati dostupni su za preuzimanje za sve inačice operacijskog sustava, a osim za razvoj aplikacija, koriste se i za testiranje određenih kompatibilnosti. SDK podržava nove i starije inačice Android operacijskog sustava te osigurava razvoj aplikacija za sve vrste uređaja [9].

2.7. Programski jezik Java

Prema [10] Java je objektno orijentirani programski jezik koji su razvili inženjeri James Gosling i i Patrick Naughton pod pokroviteljstvom tvrtke Sun Microsystems 1995. godine. Java je danas jedan od najkorištenijih programskih jezika bazirana na C++ , a kao osnovni koncepti koriste se klase i objekti, odnosno sav programski kod koji se piše napisan je unutar klasa ili razreda. Osnovni tipovi podataka koji se koriste u Javi su int, boolean, byte, string i drugi.



Slika 2.4. Android SDK Manager

Java je jednostavan, objektno orijentirani jezik, neovisan o arhitekturi i prenosiv na sva računala. Višenitnost i dinamikatakođer su neke od odlika. Programerima je omogućeno da napisani kod, bez dodatnih prevođenja pokrenu na bilo kojem računalu. To omogućuje JVM (eng. *Java Virtual Machine*) koja je razvijena za određeni operacijski sustav.

Do sada je objavljeno deset inačica Jave, počevši od JDK 1.0 (1996. god.), . Postoje i inačice Jave, Java SE (eng. *Standard Edition*), Java ME (eng. *Micro Edition*) izdanje dizajnirano za mobilne uređaje te Java EE (eng. *Enterprise Edition*) koje pruža alate namijenjene za razvoj programske podrške korištene u poslovne svrhe.

2.8. Baza podataka SQLite

SQLite je programska podrška otvorenog koda koja omogućuje samostalan, jednostavan, transakcijski i poslužiteljski neovisan sustav za upravljanje SQL bazama podataka. Za korištenje sustava SQLite na Android uređajima, potrebno je jedino definirati upite za kreiranje i uređivanje baze podataka. Nakon kreiranja upita, bazom je moguće upravljati pomoću Android

platforme. SQLite nije potrebno instalirati i podešavati, nema klijent-poslužitelj arhitekturu nego se kod izvršava po pozivu.

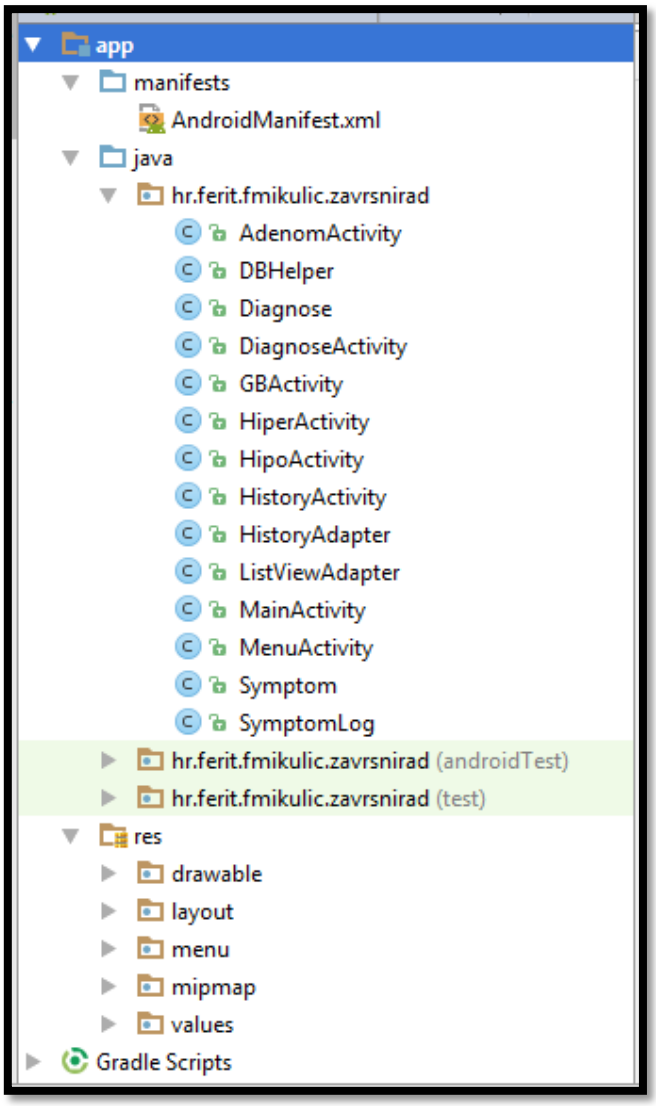
Unutar Android SDK nalazi se paket *Android database* koji sadrži sve potrebne i već korištene klase za rad s podacima. Tako se u paketu nalazi i „SQLiteHelper“, klasa koja nudi metode za korištenje baze podataka. Te metode su korištene pri izradi baze podataka u ovom projektu [11].

3. ANDROID APLIKACIJA THYROID'S APP

3.1. Struktura aplikacije

Mobilna aplikacija rađena je dakle u Android studiju u programskom jeziku Java. Kao što je navedeno ranije, aplikacija se bavi dijagnosticiranjem bolesti uzrokovanih hormonalnim poremećajem štitnjače. Radi se o četiri bolesti, a to su hipotireoza, hipertireoza, adenomi i Grawes-Basedowljeva bolest. U ovom dijelu rada najprije ćemo navesti sadržaj, odnosno koncept aplikacije te u nastavku za svaki pojedini dio prikazati i programsku realizaciju istoga. Nakon otvaranja aplikacije najprije dolazi početni zaslon gdje se korisniku nudi prijava svojim Google računom. Nakon toga prikazana je „home“ stranica sa raznim opcijama. Moguće je pristupiti svakoj od ove četiri bolesti te pročitati određene pojedinosti o njoj. Nakon toga u sadržaju „menu-a“ pristupa se samoj dijagnozi na način da se odaberu određeni simptomi te provjeri rezultat koji bi aplikacija dala. Na kraju postoji i opcija za spremanje rezultata, odnosno korisnikove dijagnoze koju je moguće zadržati i pohraniti u bazu podataka. Prilikom izrade same aplikacije, potrebno je znati neka osnovna obilježja, odnosno osnovne dijelove aplikacije te struktru projekta. Aplikacija se sastoji od tri osovne datoteke u kojima je sadržan sav kod prilikom izrade. To su datoteke „manifest“, „java“ i „res“ kao što je vidljivo na slici 3.1.
U mapi manifest nalazi se *AndroidManifest.xml* datoteka. Ovo je datoteka pisana XML jezikom u kojoj se nalazi gotovo svaki sastavni element aplikacije. Unutra su onda, primjerice, sadržani metapodaci o aplikaciji, definirana struktura aplikacije, dozvole, verzija aplikacije i drugo. U sljedećem primjeru na slici 3.2 definirano je ime, ikona i tema mobilne aplikacije.

U mapi *res* smješteni su resursi za aplikaciju. Resursi, odnosno izgled, slike i razni drugi podaci uglavnom se razdvajaju od samog koda aplikacije što iznimno olakšava izmjene, odnosno prilagodbe različitim uređajima. Tako se primjerice izgled aplikacije, odnosno elementi korisničkog sučelja definiraju u resursima koji se nazivaju *layout*. *Layouti* u principu predstavljaju svaki zaslon aplikacije posebno. U mapi *drawable* nalaze se slike korištene na svim zaslonima u aplikaciji dok se u mapi *mipmap* može pronaći ikonu koju koristi aplikacija. Mapa *values* sprema sve boje, stilove, tekst i slično. Posljednja mapa je *menu* gdje se nalazi kod za izgled izbornika koji je prisutan na svakom pojedinom zaslonu. Sve navedene datoteke pisane su XML jezikom.



Slika 3.1. Sadržaj projekta u Android Studiu

U mapi *java* nalaze se klase u kojima je pisan kod mobilne aplikacije. Prema slici 3.1. može se vidjeti da su definirane klase *DBHelper*, *Diagnose*, *Symptom*, *SympromLog*, *HistoryAdapter*, *ListViewAdapter* te gotovo sedam *Activitya*. *Activity* je klasa koja predstavlja jedan zaslon aplikacije i sadrži mnoštvo metoda koje održavaju životni ciklus aplikacije i koje će biti detaljnije objašnjene u nastavku.

3.2. Programsko rješenje aplikacije

Nakon ulaska u bilo kakvu aplikaciju , otvara se početni zaslon. Tako i u ovoj mobilnoj aplikaciji nakon ulaska prikazan je početni zaslon s imenom aplikacije, kratka uputa i informacije o aplikaciji i također gumb za prijavu. Izgled početnog *Activityja* na slici 3.3 realiziran je kodom koji se nalazi u *activity_main.xml* datoteci. Prema kodu sa slike 3.4 vidi se da se radi o više linearnih *layouta* povezanih u jednu cjelinu . Dio gdje s se nalazi logo aplikacije zajedno s

uputom ostvaren je linearnim *layoutom* s okomitom orijentacijom. Elementi se u ovakom *layoutu* slažu jedan ispod drugoga. Postoji i horizontalna orijentacija na kakav je način, primjerice, orijentiran donji dio zaslona gdje se nalazi gumb za prijavu. U ovom dijelu koda, prikazanom na slici 3.4, definiran je samo izgled zaslona pa su tako, primjerice, određene boje teksta, pozadine, stil i font slova te sve što se nalazi na početnom zaslonu. Primjerice *TextView* je klasa koja predstavlja UI kontrolu za prikaz tekstualnog sadržaja, a parametrima *layout_width* i *layout_height* definiraju se dimenzije širine i visine sadržaja.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="hr.ferit.fmikulic.zavrsnirad">

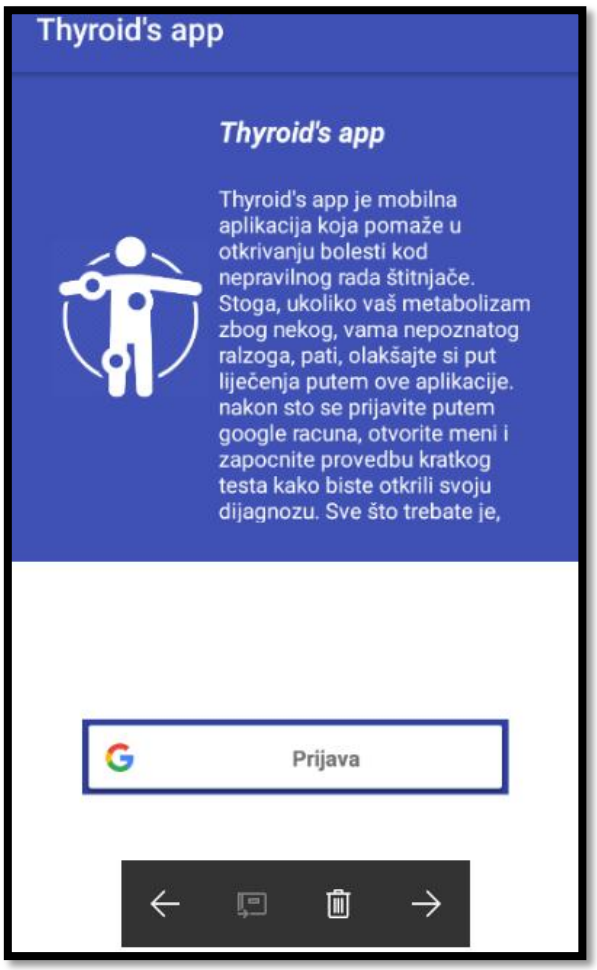
    <uses-permission android:name="android.permission.INTERNET" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/app_logo"
        android:label="Thyroid's app"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name="hr.ferit.fmikulic.zavrsnirad.MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name="hr.ferit.fmikulic.zavrsnirad.MenuActivity"
            android:label="Ime aplikacije"
            android:theme="@style/AppTheme.NoActionBar" />
        <activity android:name="hr.ferit.fmikulic.zavrsnirad.SymptomLog" />
        <activity android:name="hr.ferit.fmikulic.zavrsnirad.HipoActivity" />
        <activity android:name="hr.ferit.fmikulic.zavrsnirad.HiperActivity" />
        <activity android:name="hr.ferit.fmikulic.zavrsnirad.AdenomActivity" />
        <activity android:name="hr.ferit.fmikulic.zavrsnirad.GBActivity" />
```

Slika 3.2. XML kod datoteke AndroidManifest.xml

Sada aplikacija ima i konačan izgled početnog zaslona, no kako bi svi ti elementi bili funkcionalni potrebno je dodati određeni kod koji je zapisan u *MainActivity.java* datoteci. Ova datoteka sadrži određene metode kojima se, iz XML opisa, kreiraju određeni objekti i tako sve elemente čine funkcionalnima. Na slikama 3.5 i 3.6 prikazan je rad nekih ključnih metoda za funkcionalnost kompletnog zaslona. *OnCreate()* metoda spaja sve iz spomenutih *layouta* u jednu cjelinu. Dakle, *buttone, TextViewove* i drugo, dok se *u onClick()* metodi prosljeđuje aplikaciju na druge aktivnosti ovisno o tipki koja se pritisne.



Slika 3.3. Prikaz početnog zaslona aplikacije

```
        <LinearLayout
            android:layout_width="0dp"
            android:orientation="vertical"
            android:layout_marginLeft="5dp"
            android:layout_height="match_parent"
            android:layout_weight="2">

            <TextView
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:textStyle="bold|italic"
                android:textSize="18dp"
                android:textColor="@color/white"
                android:text="Thyroid's app"
                />
        <ScrollView
            android:layout_height="wrap_content"
            android:layout_width="match_parent">
            <TextView
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:textColor="@color/white"
                android:textSize="14dp"
                android:text="Thyroid's app je mobilna aplikacija koja pomaže u otkri..."/>
        </ScrollView>
        </LinearLayout>
```

Slika 3.4. XML kod activity_main.xml datoteke

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    Prof_Section = (LinearLayout) this.findViewById(R.id.prof_section);
    LoginSection= (LinearLayout) this.findViewById(R.id.LoginSection);
    SignOut= (Button) this.findViewById(R.id.bn_logout);
    SignIn = (SignInButton) this. findViewById(R.id.bn_login);
    Name = (TextView) this.findViewById(R.id.name);
    Email = (TextView) this.findViewById(R.id.email);
    Menu= (Button) this.findViewById(R.id.btnAppMenu);
    Menu.setOnClickListener(this);
    Prof_Pic= (ImageView) this.findViewById(R.id.prof_pic);
    SignIn.setOnClickListener(this);
    SignOut.setOnClickListener(this);
    Prof_Section.setVisibility(View.GONE);
    Menu.setVisibility(View.GONE);
    LinearLayout.LayoutParams param = new LinearLayout.LayoutParams(
            ViewGroup.LayoutParams.MATCH_PARENT, 0, 4f
    );
```

Slika 3.5. Java kod metode onCreate()

```
public void onClick(View v) {
switch(v.getId())
{
    case R.id.bn_login:
        signIn();
        break;
    case R.id.bn_logout:
        signOut();
        break;
    case R.id.btnAppMenu:
        Intent intent = new Intent(this, MenuActivity.class);
        intent.putExtra(REQUEST_NAME,mName);
        intent.putExtra(REQUEST_EMAIL,mEmail);
        intent.putExtra(REQUEST_URL,mPictureURL);
        startActivity(intent);
        break;
}
}
```

Slika 3.6. Java kod metode onClick()

Također korištena je *GoogleApiClient* klasa koja je uvijek prisutna kada se integriraju određene Google usluge. Potrebno je samo u argumentu napisati koju se usluga koristi. Metode *onConectionFailed*, *signIn*, *signOut*, *handleResult*, *updateUI* su sve metode koje se koriste pri Google prijavi. Kada se pritisne tipka, Google prijava, poziva se metoda *signIn()* koja kaže *GoogleApiBuilderu* da treba uslugu *Google Login* i pokreće *Activity* na temelju te usluge. Samim pokretanjem *Activitya*, automatski se pokreće i metoda *onActivityResult* koja radnju dalje prosljeđuje metodi *handleResult*. Ova metoda pak provjerava podatke za prijavu te ukoliko je sve u redu, poziva metodu *updateUI()*, što znači da se pojavljuje na zaslonu sekcija s korisnikovom slikom i podacima.

```
public void onConnectionFailed(@NonNull ConnectionResult connectionResult) {

}

private void signIn(){
    Intent intent = Auth.GoogleSignInApi.getSignInIntent(googleApiClient);
    startActivityForResult(intent,REQ_CODE);
}

private void signOut()
{
    Auth.GoogleSignInApi.signOut(googleApiClient).setResultCallback((ResultCallback) (status) -> {
            updateUI(false);
    });
}
```

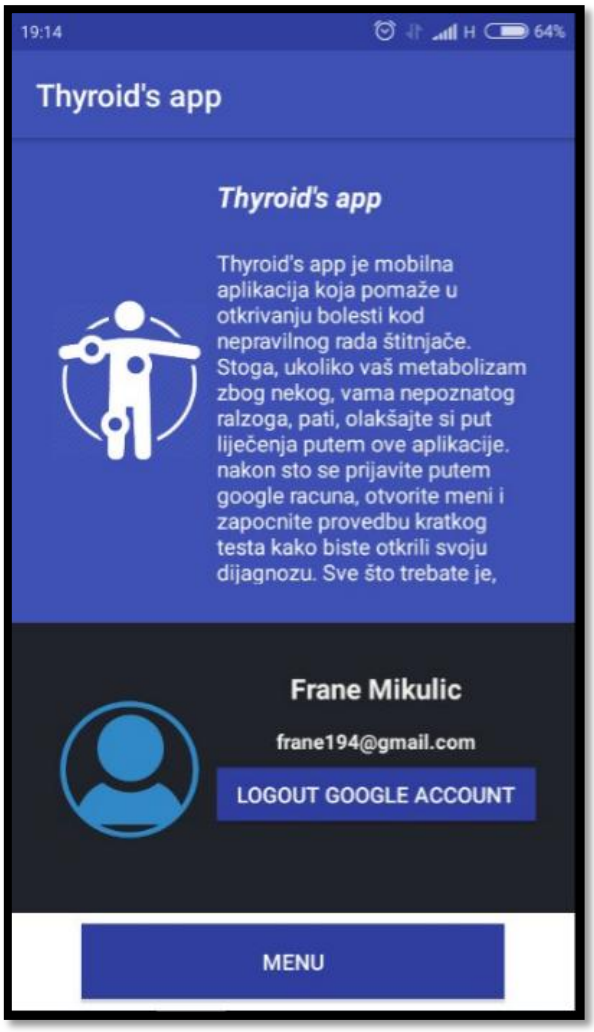
Slika 3.7. Java kod metode onConnectionFailed()

Metoda *handleResult* pak provjerava podatke za prijavu te ukoliko je sve u redu, poziva metodu *updateUI()*, što znači da se pojavljuje na zaslonu sekcija s korisnikovom slikom i podacima kao na slici 3.9.

```
private void handleResult(GoogleSignInResult result)
{
    if(result.isSuccess()){
        GoogleSignInAccount account = result.getSignInAccount();
        String name = account.getDisplayName();
        mName=name;
        String email = account.getEmail();
        mEmail=email;
        if(account.getPhotoUrl() != null) {
            String img_url = account.getPhotoUrl().toString();
            mPictureURL=img_url;
            Glide.with(this).load(img_url).into(Prof_Pic);
        }
        else
        {
            Glide.with(this).load(R.drawable.user_image).into(Prof_Pic);
        }
        Name.setText(name);
        Email.setText(email);
        updateUI(true);
    }
    else{
        updateUI(false);
    }
}
```

Slika 3.8. Java kod metode handleResult()

Nakon obavljene prijave klikom na gumb *menu* dolazi se do sljedećeg zaslona kojeg se ujedno može nazvati i *home page*. Na sljedećoj slici može se vidjeti da su na tom zaslonu ponuđene informacije o sve četiri bolesti do kojih se dolazi klikom na svaku od njih. XML kod za izgled ovog *Activityja* pisan je gotovo po identičnom principu kao i izgled početne stranice.



Slika 3.9. Prikaz početnog zaslona nakon uspješne prijave



Slika 3.10. Zaslon s bolestima

Na slici 3.11 prikazano je kako se poziva metoda *onClick()*. U toj metodi određuje se gdje će se aplikacija usmjeriti nakon što se dogodi klik na gumb na zaslonu. Ukoliko korisnik dotakne sliku hipotireoze, aplikacija se prosljeđuje na slijedeći *Activity* na kojem su informacije o navedenoj bolesti te tako za svaku bolest, pojavljuje se njen vlastiti zaslon.

3.2.1. Postupak dijagnosticiranja

Nakon što se potencijalni korisnik informira o određenim bolestima, u izborniku koji se nalazi u lijevom kutu zaslona može se pronaći gumb s kojim se pristupa samom postupku dijagnosticiranja. Tako se dolazi do glavnog dijela ove mobilne aplikacije, a to je izračun dijagnoze prema simptomima koje smo prethodno odabrali. Svaka od ove četiri bolesti ima simptome koji ju karakteriziraju. Ukoliko korisnik odabere simptome koji se tiču samo jedne bolesti, vrlo je jasno o kojoj se bolesti radi, no situacija se komplicira kada korisnik označi

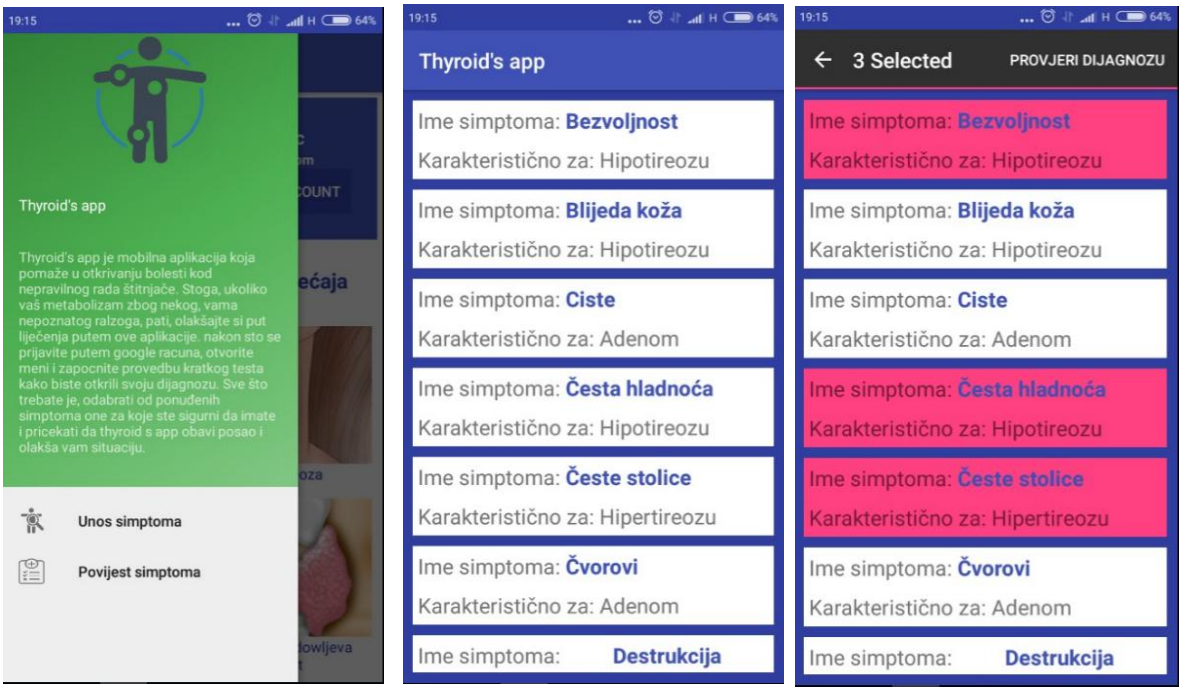
simptome sve četiri bolesti. Za to je kreirana posebna metoda koja računa vjerojatnost određene bolesti. Na slici 3.12 prikazan je daljnji slijed korištenja aplikacije gdje su nasumično odabrana tri simptoma.

```
public void onClick(View v) {
    Intent intent;
    switch (v.getId()){
        case R.id.ibhipo:
            intent = new Intent(this,HipoActivity.class);
            startActivity(intent);
            break;
        case R.id.ibhiper:
             intent = new Intent(this,HiperActivity.class);
             startActivity(intent);
            break;
        case R.id.ibadenomi:
             intent = new Intent(this,AdenomActivity.class);
             startActivity(intent);
            break;
        case R.id.ibgbbolest:
             intent = new Intent(this,GBActivity.class);
             startActivity(intent);
            break;

        case R.id.bn_logout:
            intent = new Intent(this, MainActivity.class);
            intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
            startActivity(intent);
            finish();
            break;
    }
}
```

Slika 3.11. Java kod metode onClick()

Nakon odabira simptoma, provjerava se korisnikova dijagnoza. Dijagnozu obavlja metoda *calculateDiagnose()* čiji je kod prikazan na slici broj 3.13 Što se tiče metode *calculateDiagnose()* , kreirano je polje imena svih simptoma. Za svaku su bolest kreirana dva polja, jedno se sastoji od indeksa onih simptoma za tu bolest u polju *names[]*, a drugo je težina tog indeksa u ukupnom postotku. Primjerice, simptomi 2., 5., 6., 15. i 25. u cijelom polju simptoma su za adenome, a njihove težine su sve 0.2 zato što ih ima pet pa svaki od njih pridonosi sa 20%.



Slika 3.12. Slijed korištenja glavnog dijela aplikacije

Dakle, ako korisnik odabere tih pet simptoma, vjerojatnost da pati od adenoma bit će 100%. Za svaku bolest težina indeksa se računa tako da se 1 podijeli sa brojem simptoma za tu bolest. No, ako se simptomi pojavljuju za više bolesti taj broj se još podijeli s brojem ponavljanja. Primjerice simptom usporenost javlja se i kod Grawes-Basedowljeve bolesti i kod hipotireoze pa je njegova težina jednaka 1/(10*2).

```
private void calculateDiagnose() {
    String[] names = new String[] { "Bezvoljnost", "Blijeda koža", "Ciste", "Česta hladnoća", "Česte stolice"
            "Dobivanje na težini", "Drhtanje","Edemi", "Grčevi u mišićima", "Gubitak na težini", "Guša", "Men
            "Nepodnošenje topline", "Nervoza", "Opstipacije", "Pojačano znojenje", "Poremećaj rada srca", "Po
            "Razdražljivost", "Topla i vlažna koža", "Tumori", "Ubrzan puls", "Umor", "Usporenost", "Uvećanje
    int[] selectedSymptoms = new int[0];
    int [] hipoIndexes= new int[]{0,1,3,7,9,18,20,21,28};
    double [] hipoWeights = new double[]{0.11,0.11,0.11,0.11,0.11,0.11,0.11,0.11,0.11};

    int[] hiperIndexes = new int[]{4,10,11,12,13,16,17,19,22,27,30};
    double[] hiperWeights = new double[]{0.09,0.09,0.045,0.09,0.09,0.09,0.045,0.09,0.045,0.09,0.09};

    int[] adIndexes = new int[]{2,5,6,15,25};
    double[] adWeights = new double[]{0.2,0.2,0.2,0.2,0.2};

    int [] gbIndexes = new int[]{8,11,14,16,19,23,24,26,29,30};
    double [] gbWeights=new double[]{0.1,0.05,0.1,0.05,0.05,0.1,0.1,0.1,0.1,0.05};
    Intent startingIntent = this.getIntent();
    if(startingIntent.hasExtra(SymptomLog.RESULT_CODE)) {
        selectedSymptoms= startingIntent.getIntArrayExtra(SymptomLog.RESULT_CODE);
```

Slika 3.13. Java kod metode calculateDiagnose()

Nadalje se for petljom prolazi kroz sve simptome, provjerava se za svaku bolest posebno sadrži li ona određeni simptom te ako sadrži, pridodaje se njegova težina u varijablu postotka svake bolesti posebno. Na kraju su prisutne četiri varijable *hipoD*, *hiperD*, *adD* i *gbD* koje sadrže vjerojatnosti dijagnoze za svaku bolest.

```
StringBuilder stringBuilderHipo = new StringBuilder();
StringBuilder stringBuilderHiper = new StringBuilder();
StringBuilder stringBuilderAd = new StringBuilder();
StringBuilder stringBuilderGb = new StringBuilder();
for(i=0;i<selectedSymptoms.length;i++){

    for(j=0;j<hipoIndexes.length;j++){
        if(selectedSymptoms[i]==hipoIndexes[j]){
            hipoD=hipoWeights[j]+hipoD;
            stringBuilderHipo.append(names[selectedSymptoms[i]]).append("\n");
        }
    }

    for(j=0;j<hiperIndexes.length;j++){
        if(selectedSymptoms[i]==hiperIndexes[j]){
            hiperD=hiperWeights[j]+hiperD;
            stringBuilderHiper.append(names[selectedSymptoms[i]]).append("\n");
        }
    }

    for(j=0;j<adIndexes.length;j++){
        if(selectedSymptoms[i]==adIndexes[j]){
            adD=adWeights[j]+adD;
            stringBuilderAd.append(names[selectedSymptoms[i]]).append("\n");
        }
    }

    for(j=0;j<gbIndexes.length;j++){
        if(selectedSymptoms[i]==gbIndexes[j]){
            gbD=gbWeights[j]+gbD;
            stringBuilderGb.append(names[selectedSymptoms[i]]).append("\n");
```

Slika 3.14. Java kod za računanje postotka

Na slici 3.15 slijedi prikaz rezultata odrađene dijagnoze koji su poredani od bolesti s najvećom vjerojatnošću prema bolesti s najnižom vjerojatnošću. Na hipotireozu otpada 22%, dok je vjerojatnost za hipertireozu 9%. Ostatak postotka koji nije registriran ni kod ostale dvije bolesti, odnosi se na mogućnost nekih drugih uzroka poteškoća kod pacijenata.



Slika 3.15. Rezultati dijagnoze

Nakon što se prikaže zaslon s rezultatima računanja dijagnoze, postoji i mogućnost spremanja dijagnoze u bazu podataka. Kod za korištenje lokalne baze podataka na mobilnom uređaju nalazi se u klasi *DBHelper*. Ova klasa nasljeđuje klasu *SQLiteHelper*, klasu koja nudi metode za korištenje baze podataka. Tablica u bazi napisana je pomoću SQL naredbi te su u klasi *DBHelper* kreirane funkcije za ubacivanje podataka u bazu, brisanje podataka iz baze te izlistavanje iz baze. Na slici 3.16. prikazan je rad metode za ubacivanje dijagnoze u bazu podataka. Metoda prima klasu dijagnoze, te njene podatke (ime, datum, postotak) sprema u tablicu. Slika 3.17 predstavlja i posljednji zaslon ove mobilne aplikacije gdje se nalazi uvid u sve dosadašnje spremljene dijagnoze.

```
public void insertDiagnose(Diagnose diagnose){
    ContentValues contentValues = new ContentValues();
    contentValues.put(Schema.DIAGNOSE_NAME,diagnose.getName());
    contentValues.put(Schema.DIAGNOSE_SYMPTOMS,diagnose.getSymptoms());
    contentValues.put(Schema.DIAGNOSE_PERCENTAGE,diagnose.getPercentage());
    contentValues.put(Schema.DATE,diagnose.getDate());
    SQLiteDatabase writableDatabase = this.getWritableDatabase();
    writableDatabase.insert(Schema.TABLE_SYMPTOMS_DIAGNOSE,Schema.DIAGNOSE_NAME,contentValues);
    writableDatabase.close();
}
```

Slika 3.16. Java kod metode insertDiagnose()



Slika 3.17. Spremljene dijagnoze

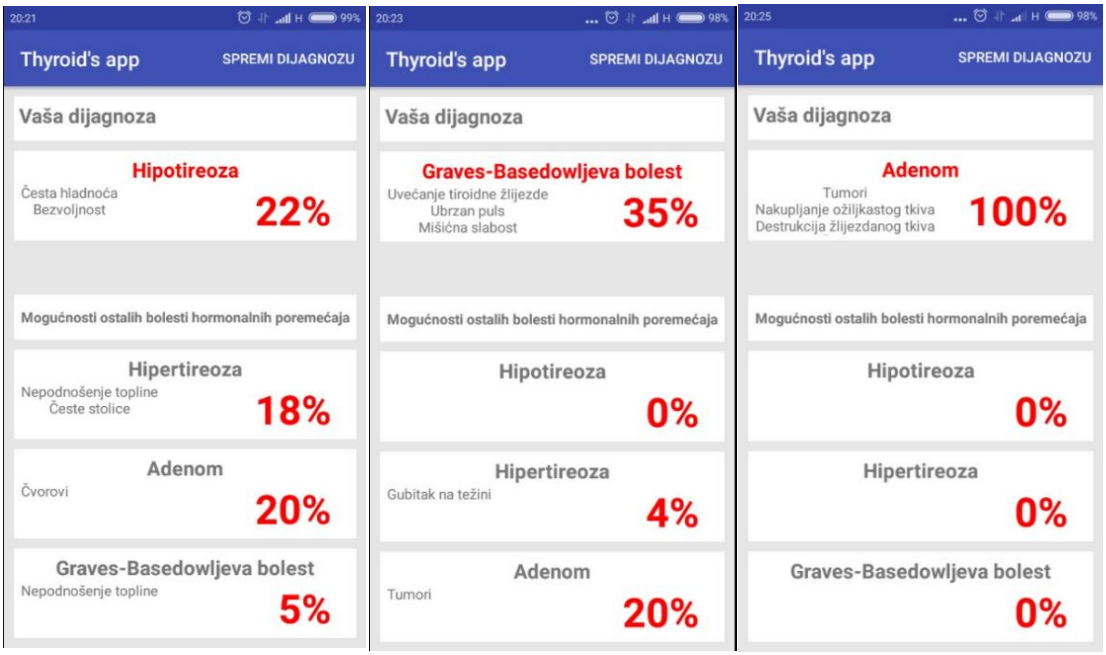
Aplikacija ne sadrži tipku koja korisnika vraća na početnu stranicu, već se po završetku spremanja dijagnoze, gumbom za vraćanje unazad na mobilnom uređaju, korisnika vraća na početni zaslon.

4. TESTIRANJE I KORIŠTENJE APLIKACIJE

Iako je testiranje aplikacije moguće i u Android Studiju na virtualnom uređaju, testiranje je obavljeno na mobilnom uređaju Xiaomi Redmi 3. Aplikacija nije zahtjevna pa je testiranje odrađeno vrlo lako i jednostavno. Thyroid's app je mobilna aplikacija koja uistinu može poslužiti bilo kojem korisniku (pacijentu ili liječniku) u što bržem snalaženju s bolesti i dijagnosticiranju bolesti. Jednostavnog je dizajna , ali također ima vrlo pregledan i sažet sadržaj. Neki od nedostataka ove mobilne aplikacije su simptomi korišteni u dijagnosticiranju bolesti. Simptomi nisu čvrsto „vezani“ i karakteristični za svaku od ove četiri bolesti, nego su to simptomi koji su prisutni i u mnogim drugim bolestima pa stoga mogu zavarati i odvesti korisnika na krivi trag. Baš zbog njihove varljive prirode, simptomi poput bezvoljnosti mogu često biti i proizvod psihe ili jednostavno lošeg dana, a biti ne povezani sa hipotireozom.

Tu se ostavlja dovoljno prostora za daljnji napredak oko aplikacije dodavanjem dodatnih simptoma koji bi se također mogli i stupnjevati, odnosno skalirati nekakvom tablicom kako bi rezultat bio što točniji. Također kao proširenje ove aplikacije u budućnosti moguće je izraditi dodatni poslužitelj preko kojeg bi se vršila registracija, a od mobilne aplikacije bi se dobivali korisnički podaci za prijavu. To bi dodatno obogatilo ovu mobilnu aplikaciju te bi joj dalo veće šanse i na tržištu.

Na slici 4.1. prikazani su rezultati dijagnoza temeljeni na različitim kombinacijama parametara. U prvom primjeru označeno je ukupno šest simptoma. Najveća vjerojatnost, 22%, odlazi na hipotireozu , 18% na hipertireozu, 20% na adenom, te 5% na Grawes-Basedowljevu bolest. Simptomi hipotireoze imaju težinu 0.11 jer ih je ukupno devet, a u ovom slučaju označena su dva pa samim time najveća vjerojatnost otpada upravo na hipotireozu. Praznina od 35% u ukupnom postotku odnosi se na nekakav drugi uzrok koji nije nužno vezan za hormonalni poremećaj štitnjače, ali ipak uzrokuje zdravstvene poteškoće korisnika. U trećem primjeru svih pet označenih simptoma odnosi se na adenom čija je težina 0.2 pa je stoga ukupna vjerojatnost 100%. Za adenom je karakteristično da ima najmanje simptoma, ali najveće težine. Primjerice, kada korisnik označi simptom čvora, velika je vjerojatnost da će konačna dijagnoza biti adenom. U tablici 4.1 prikazan je tablični prikaz rezultata sa zaslona sa slike 4.1.



Slika 4.1. Kombinacije parametara i rezultati dijagnoza

Tablica 4.1. Tablični prikaz rezultata različitih dijagnoza.

Dijagnoza	Simptomi	Rezultat
1.	Česta hladnoća, bezvoljnost, nepodnošenje topline, česte stolice, čvorovi	Hipotireoza - 22% Adenom - 20% Hipertireoza - 18% GB-bolest - 5%
2.	Uvećanje štitne žlijezde, ubrzan puls, mišićna slabost, gubitak na težini, tumori	GB-bolest – 35% Adenom – 20% Hipertireoza – 4% Hipotireoza – 0%
3.	Destrukcija žljezdanog tkiva, nakupljanje ožiljkastog tkiva, ciste. Čvorovi. tumori	Adenom – 100% Hipotireoza – 0% Hipertireoza – 0% GB-bolest – 0%

Budući da aplikacije sama po sebi nije zahtjevna, ne zahtjeva instalacije dodatnih programa, kao ni posebne specifikacije mobilnih uređaja. Jedini uvjet koji mora biti ostvaren je da se korisnik mora povezati na Internet kako bi mogao obaviti prijavu u aplikaciju.

5. ZAKLJUČAK

Zadatak ovog završnog rad bio je izrada mobilne aplikacije za potporu biomedicinskoj dijagnostici. Budući da se većina ljudi danas služi pametnim telefonima, cilj je bio pomoći da korisnik procijeni vjerojatnost da ima simptome određene bolesti, po potrebi posjeti liječnika, koji nakon kliničke obrade donosi dijagnozu. Rad je posebno posvećen hormonalnim poremećajima štitnjače. Tako se ovom aplikacijom pokušava ukazati na moguću dijagnozu bolesti kao što su hipotireoza, hipertireoza, adenomi i Grawes-Basedowljeva bolest. Aplikacija je rađena za Android, a kreirana u Android Studiu u programskom jeziku Java. Ima sedam zaslona na kojima se odvija čitav proces dijagnosticiranja bolesti. Glavna problematika ove aplikacije je, prema odabranim simptomima, izračunati s kolikom vjerojatnošću pacijent pati od određene bolesti. Odabirom različitih simptoma dobiveni su različiti rezultati te međusobno uspoređeni. Testiranjem aplikacije utvrđeno je da je postupak dijagnosticiranja ispravan te da je pretpostavka dijagnoze razmjerno pouzdana.

Nakon dobivenih rezultata, dijagnoza se može pohraniti i u bazu podataka izrađenu pomoću programske podrške SQLite. Također je ostavljeno dovoljno prostora za daljnji napredak u razvoju aplikacije, prije svega u pogledu uvrštavanja i dodatnih simptoma koji bi na težini dobivali korisnikovim skaliranjem od najniže ka najvišoj vrijednosti te mnoge druge pojedinosti.

LITERATURA

[1] Msd-priručnici, Poremećaji štitnjače, http://www.msd-prirucnici.placebo.hr/msd-prirucnik (stranica posjećena 06. lipnja 2017.)

[2] Hormonal imbalance,

http://symptoms.webmd.com/default.htm#introView (stranica posjećena 06. Lipnja 2017.)

[3] ACM Digital Library, Tracing digital thyroid culture,

http://dl.acm.org/citation.cfm?id=2524262&CFID=811291340&CFTOKEN=81946324 (stranica posjećena 06. lipnja 2017.)

[4] Thyroid Spot, https://scientificphoneapps.springeropen.com/articles/10.1186/s41070-017-0016-y (stranica posjećena 06. lipnja 2017.)

[5] Appadvice, U & Thyroid, https://appadvice.com/app/u-thyroid/1238183089 (stranica posjećena 08. lipnja 2017.)

[6] Plivazdravlje, priručnik bolesti, https://www.plivazdravlje.hr/prirucnik-bolesti (stranica posjećena 10. lipnja 2017.)

[7] Wikipedia, Android operacijski sustav

https://hr.wikipedia.org/wiki/Android_(operacijski_sustav)

[8] Andorid OS, https://www.android.com/ (stranica posjećena 14. lipnja 2017.)

[9] Android Developers, Android Studio https://developer.android.com/studio/index.html (stranica posjećena 14. lipnja 2017.)

[10] Wikipedia, Java https://en.wikipedia.org/wiki/Java_(programming_language) (stranica posjećena 14. lipnja 2017.)

[11] W3School, SQLite, https://www.w3schools.com/sql/ (stranica posjećena 14. lipnja 2017.)

SAŽETAK

Cilj ovog završnog rada bio je izrada Android mobilne aplikacije, za potporu biomedicinskoj dijagnostici bolesti štitnjače. Aplikacija na temelju odabranih simptoma korisniku daje procjenu vjerojatnosti obolijevanja od jedne od četiri bolesti uzrokovane hormonalnim poremećajem štitnjače. To su bolesti hipotireoza, hipertireoza, adenomi te Grawes-Basedowljeva bolest. U radu su prikazane bolesti štitnjače te razrađena ideja i model ove mobilne aplikacije. Nadalje dan je osvrt na programske okoline, jezike i tehnologije koje se koriste pri izradi ove aplikacije, a to su Android Studio te SQLite podrška za izradu baze podataka. Također, objašnjen je postupak razvoja, ključni programski dijelovi aplikacije, a provedeno je i testiranje ispravnosti aplikacije na odgovarajućem skupu ulaznih podataka. Pokazalo se da aplikacija daje ispravne rezultate moguće dijagnoze.

Ključne riječi: Android, mobilna aplikacija, simptomi, štitnjača

ABSTRACT

The main goal of this final work was to make an Android application to supplement biomedicinal diagnostics of thyroid disease. The application based on chosen symptoms gives the user an assessment of probability of one of four illnesses caused by hormonal disorders of the thyroid. Those illnesses are hypothyroidism, hyperthyroidism, adenoma and Graves-Basedow disease. In this final work illnesses of the thyroid are shown and the idea and model of this mobile application are elaborated. Furthermore, a review of IDEs, programming languages and technologies used to make this app ia given and those are Android Studio and SQLite database support. Also, the development process and key parts of the application are explained and also the application was tested on corresponding input data to ensure accuracy. The results were accurate when compared to known diagnostics data.

Keywords: Android, mobile application, symptoms, thyroid

ŽIVOTOPIS

Frane Mikulić rođen je 10.12.1994. godine u Đakovu, Hrvatska. Stanuje u Đakovu na adresi Franje Račkog 75. Godine 2001. započinje osnovnoškolsko obrazovanje u OŠ Ivan Goran Kovačić gdje se iskazao na brojnim županijskim natjecanjima, s istaknutim rezultatima iz matematike i geografije. Nakon završene osnovne škole, godine 2009. upisuje Gimnaziju Antuna Gustava Matoša u Đakovu. Nakon gimnazije, godine 2014. upisuje Preddiplomski studij računarstva na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija u Osijeku. Posjeduje određeno znanje engleskog jezika te kao vrline navodi marljivost i upornost. Od stečenih vještina i kompetencija posebno ističe rad u timu te iznimno brzo uklapanje u novu radnu okolinu.

PRILOZI(na DVD-u)

Prilog 1: Završni rad u docx i pdf formatu

Prilog 2: Projekt mobilne aplikacije u Android Studiu