

Android aplikacija za praćenje trudnoće

Angebrandt, Gabriela

Master's thesis / Diplomski rad

2018

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://um.nsk.hr/um:nbn:hr:200:230979>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom](#).

Download date / Datum preuzimanja: **2025-01-28**

Repository / Repozitorij:

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I
INFORMACIJSKIH TEHNOLOGIJA**

Sveučilišni studij računarstva

ANDROID APLIKACIJA ZA PRAĆENJE TRUDNOĆE

Diplomski rad

Gabriela Angebrandt

Osijek, 2018.

Sadržaj

1. UVOD	1
1.1. Zadatak diplomskog rada	2
2. RAZVOJ APLIKACIJE	3
2.1. Struktura projekta	3
2.2. MVP arhitektura	3
2.3. Implementacija programskog rješenja	4
2.4. Baza podataka	20
2.5. Kreiranje API-a	21
2.6. Prijevod aplikacije	25
3. ZAKLJUČAK	27
LITERATURA	29
SADRŽAJ	31
ABSTRACT	32
ŽIVOTOPIS	33

1. UVOD

Period trudnoće je najosjetljivije razdoblje u životu žene. Tijekom trudnoće žene prolaze kroz brojne fizičke i psihičke promjene te imaju snažne emocije. Svaka od njih pomno prati svoj razvoj tih 280 dana, koliko iznosi prosječno trajanje trudnoće. Za vrijeme trudničkog perioda, trudnica mora redovito posjećivati liječnika koji joj prati trudnoću, pa čak i onda kada ne primjećuje smetnje ili ne osjeti bolove. Trudnoća zahtijeva praćenje određenih zdravstvenih parametara radi očuvanja zdravlja buduće majke i djeteta. Neki od tih parametara su krvni tlak, tjedan trudnoće (tjedan + broj dana u narednom tjednu), razina proteina, šećera i leukocita majke u mokraći, prirast tjelesne mase, pojava edema ili oticanja dijelova tijela, pregled u spekulima¹, UZ pregled, bimanuelni pregled trbuha², a od kraja prvog tromjesečja se mjere i otkucaji djetetovog srca. Ranije navedeni parametri očitavaju se kod svakog ginekološkog pregleda, a svakako je važno, jednom tijekom trudnoće, odrediti i krvnu grupu i Rh faktor majke, PAPA test, bris na klamidiju, gonoreju i herpes te, po potrebi kod starijih trudnica, Tripple test (tri markera na Down sindrom). Osim medicinskih podataka, važna je i prehrana te svakodnevno kretanje buduće majke. U prošlosti smatralo se da su trudne žene bolesne i da se trebaju što manje kretati, no u današnje vrijeme većina trudnica sklonija je svakodnevnoj tjelovježbi – aktivnosti na otvorenom, programu kućnih treninga ili lakšem fitnessu. Prema istraživanju Hammer, Perkins i Parr (2000) redovitim vježbanjem žena efikasno održava zdravu tjelesnu masu, smanjuje mogućnost nastanka dodatnog masnog tkiva te se smanjuju bolovi u leđima. No, uz sve to, žena se mora pravilno hraniti, tj. mora u organizam unositi do 300 kalorija više od uobičajenog kako bi se fetus pravilno razvijao.

Ovaj diplomski rad prikazuje korake razvoja aplikacije za praćenje svih stadija trudnoće. Aplikaciji je dodijeljen naziv „Android aplikacija za praćenje trudnoće“. Namijenjena je za sve one buduće majke koje žele podatke o trudnoći pohranjivati na svom pametnom telefonu. U aplikaciju se uvode podaci sa svih termina pregleda kod ginekologa te podaci osobne anamneze. Korisnica aplikacije može pregledavati popis vježbi iz programa kućnih treninga za trudnice te može pregledavati popis jela koji ulaze u program zdrave prehrane. Budući da sve više žena u trudničkom razdoblju želi ukazati na razvoj bebe objavljivanjem slika na društvenim mrežama, ova aplikacija omogućava dodavanje slika u zajedničku galeriju koju mogu pregledavati samo korisnice aplikacije. Nadalje, kako bi se žena što više informirala o svom stanju te promjenama i mogućim rizicima u trudnoći, aplikacija sadrži oblik foruma u

¹ Vrsta ginekološkog pregleda – kontrolira se rodnica i vrat maternice

² Pregled procjene položaja djeteta

kojemu korisnici mogu postavljati pitanja anonimno te odgovarati na ista. Forum je osmišljen kako bi si korisnici međusobno pružale podršku te kako bi davale savjete jedne drugima u različitim situacijama.

Za razvoj ove aplikacije odabran je Android operacijski sustav američke tvrtke Google Inc. koji je poznat po svojoj jednostavnosti i učinkovitosti. Također, potrebno je napredno poznavanje programskog jezika Java i programa Android Studio. Kao dodatni dio diplomskog rada bit će korištena RxJava s Retrofitom te *Model-View-Presenter* arhitektura, a za razvoj API-a potrebnog za pohranjivanje programa vježbi i prehrane bit će korišten Node.js, Mongo baza podataka, *ngrock* za preusmjerenje lokalnog servera te programi Atom i Postman za pisanje koda.

Strukturu diplomskog rada čine uvod, glavni dio rada – razvoj aplikacije, zaključak, literatura, sažetak na hrvatskom jeziku, sažetak na engleskom jeziku, životopis te prilozi. Uz diplomski rad prilaže se i elektronička verzija rada na CD-u.

1.1. Zadatak diplomskog rada

Zadatak ovog rada je opisati problematiku praćenja trudnoće i svih relevantnih parametara. Izraditi Android aplikaciju koja će omogućiti trudnicama praćenje i pohranjivanje važnih parametara tijekom trudnoće. Aplikacija treba imati mogućnost pohrane slika. Aplikacija treba imati mogućnost razmjene iskustava trudnica u obliku nekakve društvene mrežne komponente.

2. RAZVOJ APLIKACIJE

U ovom poglavlju opisan je razvoj aplikacije za vođenje trudnoće.

2.1. Struktura projekta

Projekt se sastoji od klasa koje mogu biti pisane u Java ili Kotlin programskom jeziku te pripadajućih datoteka rasporeda koje su pisane u XML-u (engl. *EXtensible Markup Language*) – jeziku za označavanje. U datoteci rasporeda postavljaju se kontrole ili pogledi koji će biti prikazani na zaslonu uređaja (npr. *TextView*, *Button*, *ImageButton* itd.), a u klasama se zapisuju funkcionalnosti koje pogledi rasporeda izvode. Nadalje, za strukturu važne su još:

- a) datoteka koja sadrži ključne informacije o aplikaciji – *AndroidManifest.xml*
- b) mapa za pohranjivanje ikona i slika - *drawable*
- c) datoteka za definiranje stilova koji se koriste za poglede aplikacije – *style.xml*
- d) datoteka u koju se zapisuje sav tekst koji se prikazuje na zaslonima aplikacije – *strings.xml*
- e) datoteka u kojoj se definiraju sve boje koje se koriste u aplikaciji – *colors.xml*

Vanjske biblioteke koje nisu dio osnovnog skupa biblioteka u Android Studio programa mogu se uvesti implementiranjem u *gradle* datoteci.

2.2. MVP arhitektura

Budući da se u ovom radu koristi MVP arhitektura, svaka aktivnost imaće se svoj direktorij unutar stabla projekta u kojemu se nalazi direktorij za *view* klasu, direktorij za *presenter* klasu te po potrebi direktorij za različite adaptere i direktorij za fragmente. Modeli se nalaze u posebnom direktoriju koji se zove *models*. Osim što se koristi za lakše testiranje aplikacije, pravilna upotreba ove arhitekture omogućava razvojnom inženjeru lakše snalaženje u kodu i veću preglednost nad sintaksom.

Model je odgovoran za upravljanje podacima koji se mogu dobivati pomoću API-a ili se mogu spremati u različite vrste baza podataka. Također, model manipulira i predmemorijom. Zbog čiste arhitekture, u ovom radu model će se zvati *Interactor*.

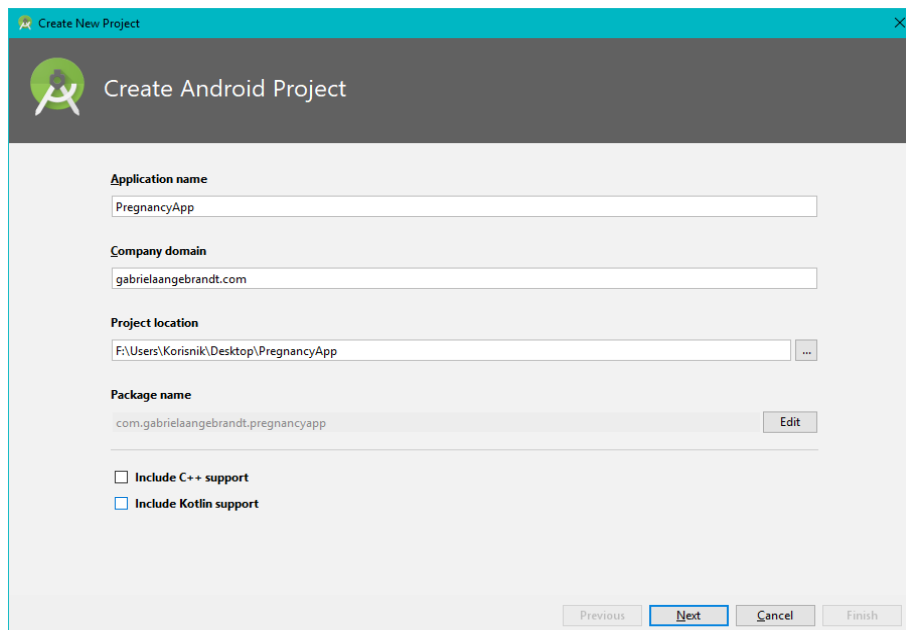
View je zadužen samo za prikazivanje podataka na zaslonu. Njega najčešće predstavljaju aktivnosti, fragment ili neki *widget-i* koji mogu ažurirati komponente iz datoteke rasporeda.

Presenter predstavlja posrednika između *viewa* i *modela*. On sadrži svu logiku za prezentiranje podataka na zaslonu. On je odgovoran za ažuriranje podataka u modelu i za osvježavanje prikaza na zaslonu

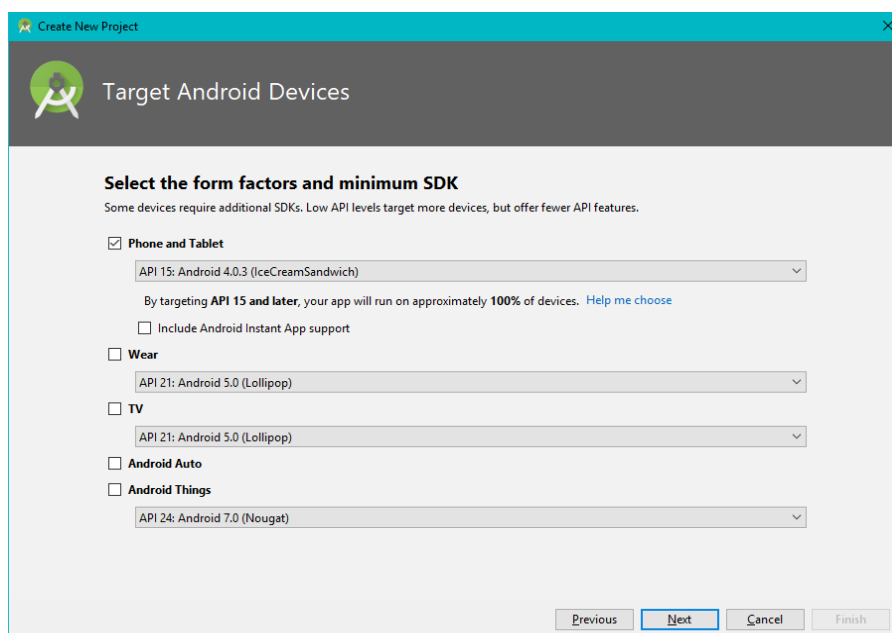
Dakle, ukoliko korisnik odabere na zaslonu određeni gumb kojim se treba prikazati nekakav sadržaj na zaslonu, *view* zahtijeva podatke od *presenter-a*, *presenter* zna kojom logikom se trebaju tražiti ti podaci od modela, model pronalazi podatke te ih šalje *presenter-u* i na kraju *presenter* prosljeđuje podatke u *view* koji ih prikazuje na zaslonu. Ove tri komponente međusobno komuniciraju preko sučelja, tzv. *interface-a*, koji su definirani u datoteci koja ima u drugom dijelu naziva riječ *Contract*, npr. *MainContract*, *FragmentContract* i slično.

2.3. Implementacija programskog rješenja

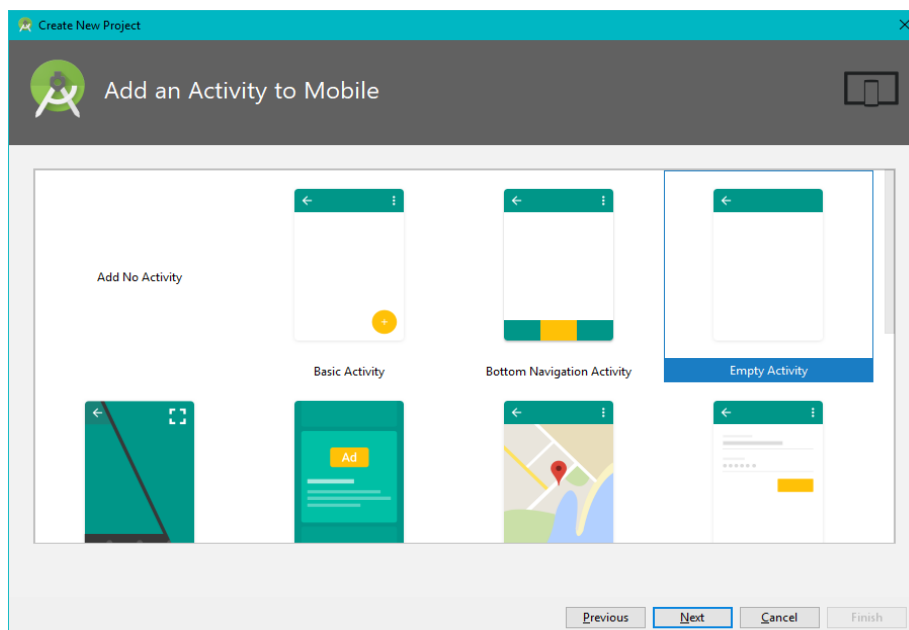
Na početku razvoja aplikacije potrebno je kreirati projekt. Nakon otvaranja programa Android Studio potrebno je prvo odabrati ime aplikacije, lokaciju gdje će se spremati projekt te domenu pod kojom će se projekt spremati (Sl. 2.1.). Zatim se odabire najniža inačica operacijskog sustava na kojemu će se moći koristiti aplikacija (Sl. 2.2.). U ovom slučaju, to je API 15: Android 4.0.3 (IceCreamSandwich) koju trenutno koristi oko 0.4% korisnika Android (Statista, 2018). Nakon inačice, potrebno je odabrati vrstu početnog zaslona aplikacije (Slika 3.).



Sl. 2.1. Odabir naziva aplikacije i lokacije projekta



Sl. 2.2. Odabir najniže verzije SDK



Sl. 2.3. Odabir izgleda početne aplikacije

Za ikonu aplikacije korištena je ikona prema slici 2.4. (lijevo). Ikona je preuzeta s web izvora (Design, n.d.). Uz ikonu napravljen je i logo za aplikaciju s istog izvora (Sl. 2.4. desno). Ove dvije datoteke spremljene su u *drawable* mapi. Ikona aplikacije postavlja se u *AndroidManifest* datoteci, a logo se koristi u aktivnosti za prijavu korisnika.

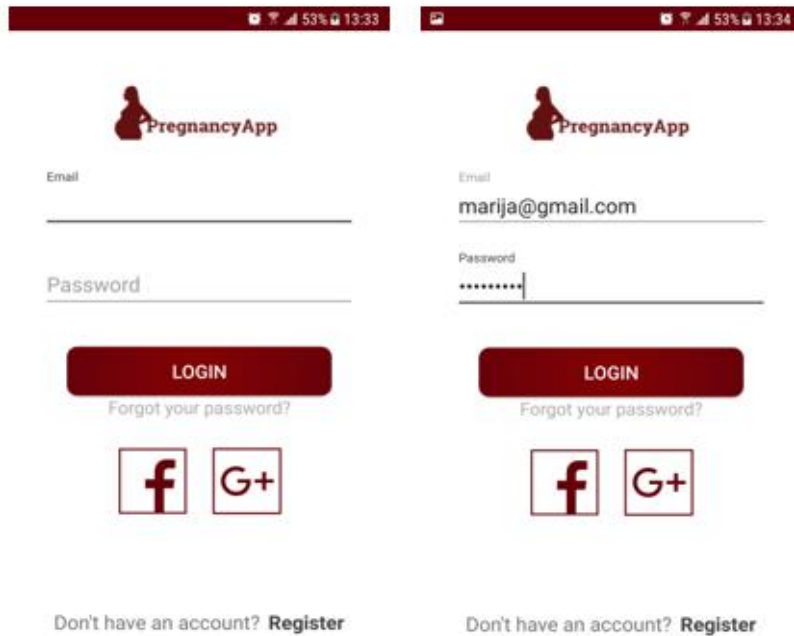


Sl. 2.4. Ikona i logo aplikacije

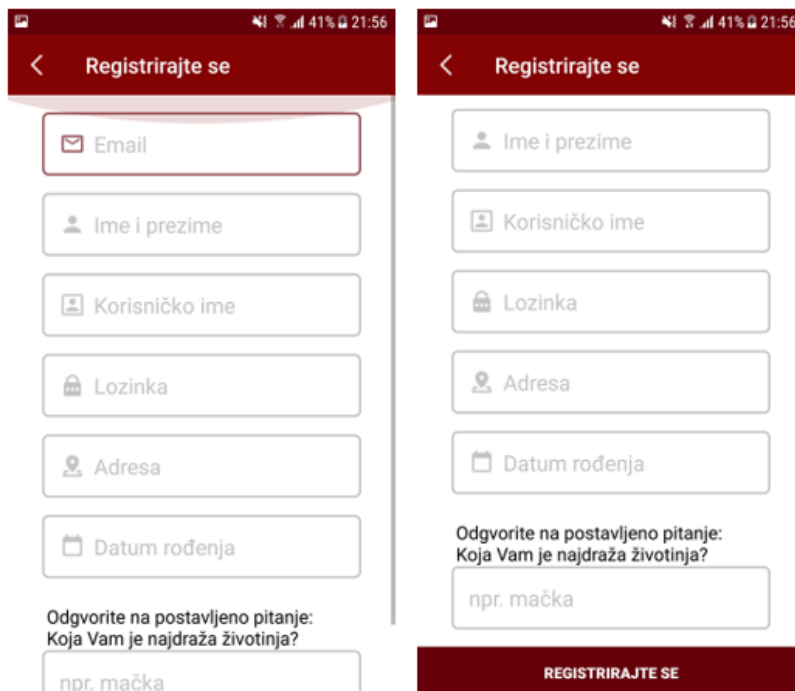
Jezik aplikacije ovisi o jeziku koji je postavljen na Android uređaju. U slučaju da je jezik uređaja hrvatski, sav tekst u aplikaciji bit će na hrvatskom jeziku. Za sve ostale jezike na uređaju bit će postavljen tekst na engleskom jeziku u aplikaciji.

Početni zaslon aplikacije sastoji se od obrasca za prijavu korisnika u sustav (Sl. 2.5.). Najprije su dodani zaštitni logo aplikacije i dva polja za unos teksta koji zahtijevaju unos elektroničke pošte i zaporke koje je korisnik definirao pri registraciji. Ukoliko korisnik nije registriran, na zaslonu za prijavu nalazi se i gumb za registraciju. Odabirom gumba za registraciju otvara se zaslon koji sadrži 7 tekstualnih polja u koje se unose email, ime i prezime,

željeno korisničko ime, zaporka, adresa, datum rođenja te odgovor na pitanje ugrađeno u aplikaciju koje predstavlja sigurnosnu riječ za oporavak zaporce (Sl. 2.6.).



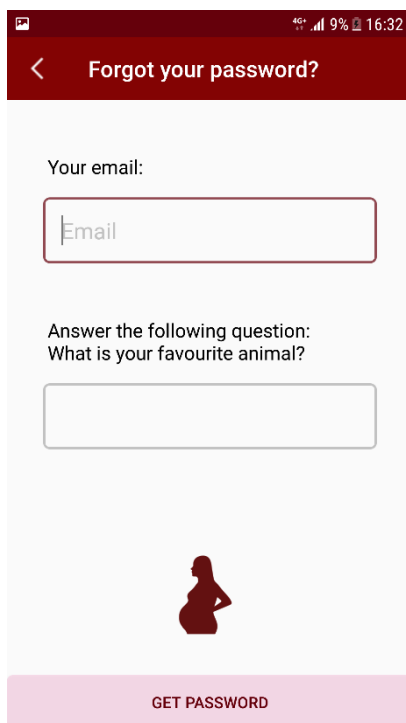
Sl. 2.5. Zaslona prijave



Sl. 2.6. Zaslona registracije

Tekstualno polje za elektroničku poštu ima ograničenje da unos mora biti u obliku koji je definiran standardom, dok polje za zaporku ima ograničenje na minimalno 8 znakova. Znakovi zaporke su skriveni pri unosu. Nakon unosa podataka potrebnih za registraciju, korisnik se registrira odabirom gumba „REGISTER“ te se, u slučaju uspješne registracije, zaslon preusmjerava na prikaz prijave u sustav. U slučaju greške, na zaslon se ispisuje poruka o greški. Podaci o korisniku spremaju se u bazu podataka Firebase platforme koja je u vlasništvu tvrtke Google Inc. Ova baza omogućava zapisivanje podataka u JSON formatu te interakciju u stvarnom vremenu.

U slučaju da je korisnik zaboravio svoju zaporku, na zaslonu za prijavu nalazi se gumb „Forgot your password?“ koji preusmjerava korisnika na zaslon u kojemu su polja za unos adrese elektroničke pošte i sigurnosnog odgovora na pitanje koje je postavljeno pri registraciji. Ukoliko korisnik točno unese tražene podatke te pritisne gumb „GET PASSWORD“ (podaci se uspoređuju s podacima u bazi koji su uneseni prilikom registracije), na adresu elektroničke pošte šalje se poruka s poveznicom na kojoj se može „oporaviti“ zaporka (Sl. 2.7.). U slučaju krivog upisa podataka, na donjem dijelu prikaza ispisuje se poruka o greški.



4G+ 9% 16:32

< Forgot your password?

Your email:

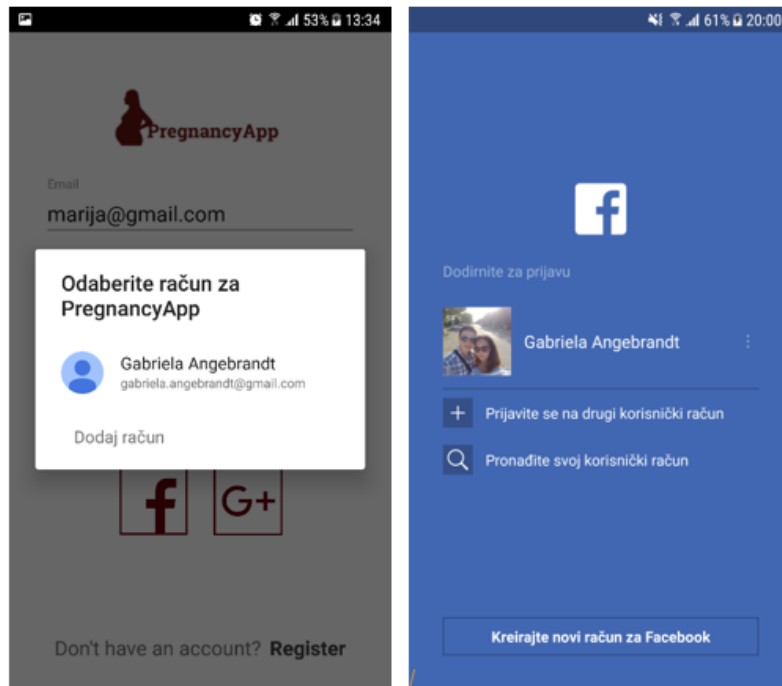
Email

Answer the following question:
What is your favourite animal?

GET PASSWORD

Sl. 2.7. Zaslon za prikaz lozinke

Osim lokalne prijave u sustav, korisnik se može prijaviti pomoću Gmail i Facebook usluge. Za te funkcionalnosti postoje dva pripadajuća gumba na zaslonu za prijavu i za ovakvu prijavu nije potrebna prethodna registracija u sustav aplikacije za praćenje trudnoće, već je potrebno da korisnik posjeduje korisničke račune na navedenim uslugama (Sl. 2.8.).



Sl. 2.8. Prijava pomoću Gmail (lijevo) i Facebook (desno) usluge

Odabirom bilo kojeg od ta dva gumba korisnik se preusmjerava na zaslone koje je predefinirao Gmail-a, odnosno Facebook-a te se slijede dane upute za proces prijave. Autentifikacija korisnika izvršava se pomoću Firebase autentifikacije.

Nakon uspješnog prijavljivanja u sustav, omogućen je pristup kliznom izborniku putem kojeg se vrši navigacija kroz aplikaciju. Klizni izbornik otvara se aktivacijom tzv. hamburger ikone u gornjem lijevom kutu alatne trake i omogućava pristup zaslonima na kojima se nalaze podaci o trudnoći. Osim naslovne slike na vrhu, navigacijski prozor aplikacije sadrži i sljedeće elemente (odozgo prema dolje):

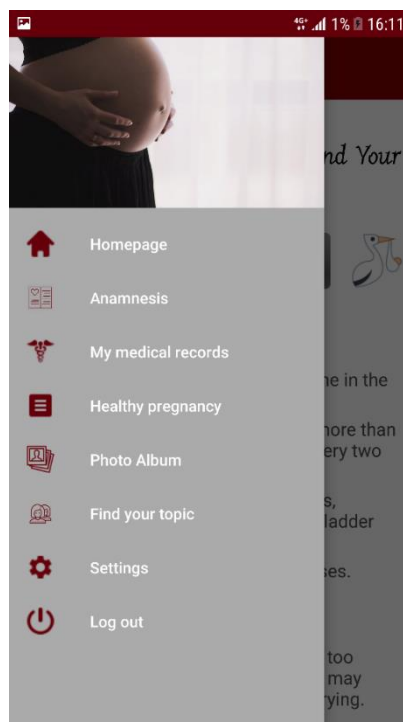
1. početni zaslon
2. zaslon za podatke o osobnoj i obiteljskoj anamnezi te ugroženosti trudnoće
3. zaslon za podatke o ginekološkim pregledima
4. zaslon koji prikazuje popis treninga i obroka za trudnice

5. zaslona galerije slika

6. zaslona foruma

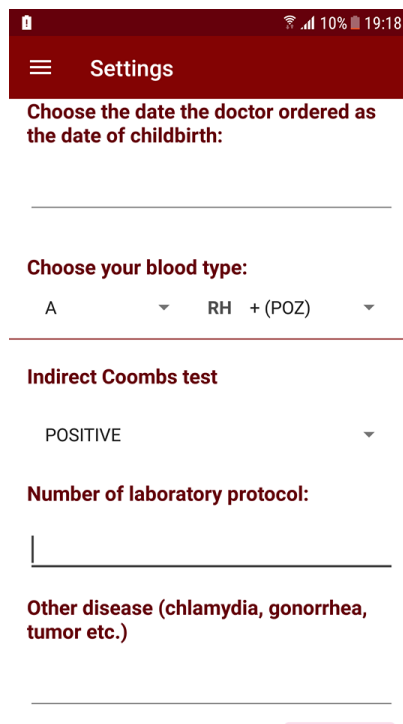
7. zaslona postavki

Navigacijski prozor izveden je implementacijom *NavigationDrawer-a* (Sl. 2.9.). On ima sposobnost otvaranja/zatvaranja i smješten je na lijevoj strani zaslona od vrha do dna. On sadrži prethodno nabrojane elemente odabirom kojih se otvara određeni zaslona. Zadnji element u listi odjavljuje korisnika iz aplikacije, odnosno briše podatke koji sadrže informacije o prijavljivanju određenog korisnika (brišu se podaci Firebase autentifikacije). Slijedom odozgo prema dolje, prvi, treći i sedmi element izvedeni su pomoću fragmenta, dok su drugi, četvrti i šesti izvedeni pomoću aktivnosti. Kod rada s fragmentima otvara se transakcija fragmenata, vrši se zamjena određenih fragmenata unutar *FrameLayouta* koji se nalazi u rasporedu početne aktivnosti te se na kraju zamjene transakcija mora zatvoriti. Kod aktivnosti nema transakcije, već se poziva otvaranje nove aktivnosti s metodom *startActivity()*. U svakoj aktivnosti, definirana je alatna traka na kojoj se nalazi navigacijska ikona, izuzev početne aktivnosti gdje se na alatnoj traci na mjestu navigacijske nalazi hamburger ikona. Odabirom standardne navigacijske ikone zaslona se preusmjerava na prikaz fragmenta/aktivnosti koja se reprezentirala prije trenutne ili na početni zaslona. Također, u svakoj aktivnosti definiran je i naslov koji se ispisuje na alatnoj traci.



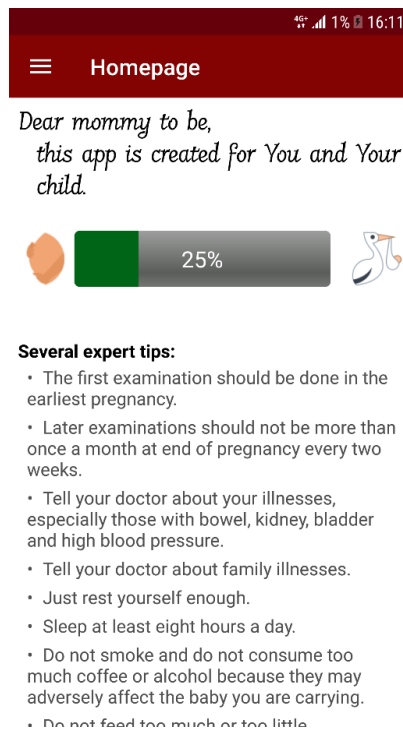
Sl. 2.9. Navigacijski meni

Nakon prve prijave korisnika u sustav otvara se zaslon s postavkama (Sl. 2.10.). Na njemu se nalazi polje za unos datuma koji je odredio doktor kao datum poroda te dva *spinnera* za odabir krvne grupe (vrijednosti: A, AB, B, 0) i Rh faktora (vrijednosti: + (POZ), - (NEG)). Ove vrijednosti definirane su u *strings.xml* datoteci. Nadalje tu je i polje za unos broja laboratorijskog protokola, *spinner* za Coombs-ov test (test aglutinacije, vrijednosti: (POZ), (NEG)) te polje za unos niza drugih bolesti ako postoje. Na dnu zaslona nalazi se gumb čijim se odabirom uneseni podaci spremaju u bazu podataka pod korisnika koji je trenutno prijavljen. Prijavljenog korisnika može se provjeriti pozivanjem metode *getCurrentUser()* na instanci Firebase autentifikacije. U slučaju greške ili nedostatka unesenih podataka ispisuje se poruka o greški. U slučaju uspješnog pohranjivanja podataka, podaci se također spremaju i korištenjem tzv. *SharedPreferences* za jednostavniji izračun trajanja trudnoće te se korisnik preusmjerava na glavni zaslon aplikacije.



Sl. 2.10. Zaslon postavki

Naslovnica ili glavni zaslon u aplikaciji sadrži pozdravnu poruku korisniku koja je ispisana koristeći *TextView* i napisana je modificiranim kosim Garibaldi fontom (Sl. 2.11.).

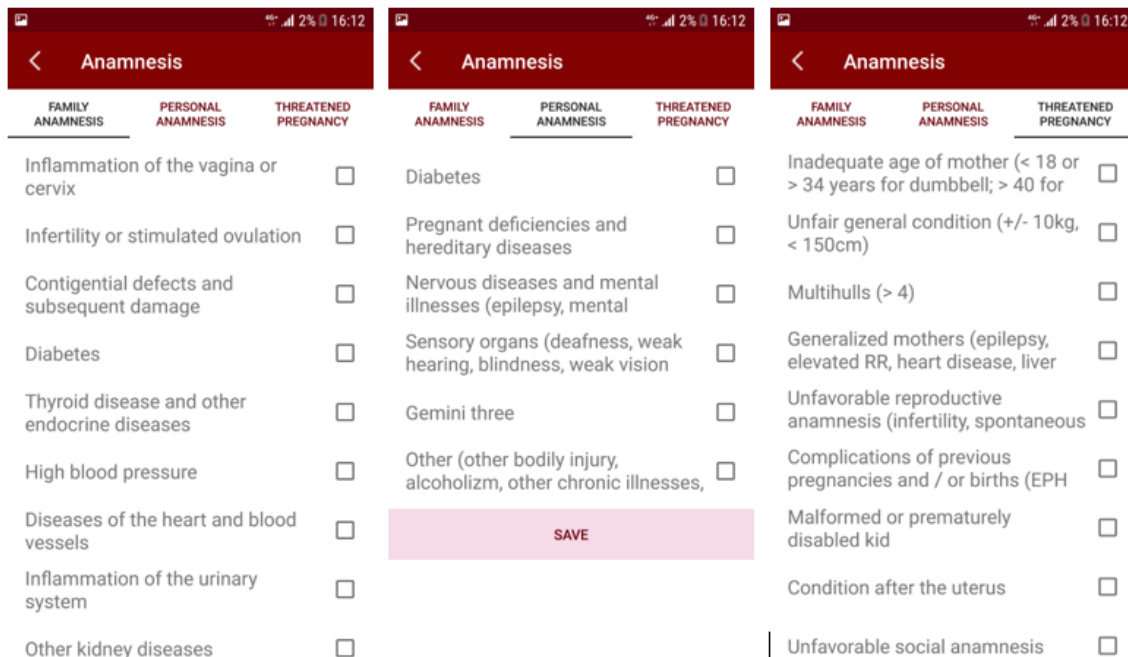


Sl. 2.11. Glavni zaslon aplikacije

Ispod pozdravne poruke prikazana je traka za napredak trudnoće. Postotak koji je iscrtan na traci predstavlja vrijeme koje je prošlo otkada je započela trudnoća. Ovaj postotak računa se na temelju unesenog datuma poroda koje je odredio ginekolog (upisuje se u zaslonu postavki) te trenutnog datuma. Broj dana između unesenog i trenutnog datuma oduzima se od 280, zatim se dijeli s 280 te se dobiveni rezultat množi sa 100 kako bi se dobio postotak. Postotak se ispisuje zaokružen na jednu decimalu. Osim spomenute trake, tu su i dvije slike s desne i lijeve strane trake koje predstavljaju razvoj trudničkog perioda. Ispod trake za napredak trudnoće zapisani su neki općeniti savjeti stručnjaka u ovom području. Za ispisivanje savjeta korišten je *RecyclerView* s linearnim rasporedom. Elementi su jednostavni, sadrže samo jedan *TextView*. U ovoj aktivnosti nema interakcije s korisnikom već služi za informiranje korisnika.

Odabirom drugog elementa u navigacijskom prozoru otvara se zaslon koji prikazuje podatke o anamnezi (Sl. 2.11.). U ovoj aktivnosti korišteni su *TabLayout* za prikazivanje tzv. *tab*-ova ili kartica te *ViewPager* za izmjenu prikaza podataka klizeći prstima s lijeva na desno i obrnuto. Definirane su tri kartice od kojih prva predstavlja podatke o obiteljskoj anamnezi, druga predstavlja podatke o osobnoj anamnezi te zadnja predstavlja podatke o ugroženosti trudnoće, tj. podaci koji opisuju trenutni period i način života majke, a utječu nepovoljno na fetus. Unutar kartica implementiran je *ListView* čiji elementi sadrže *TextView* i *CheckBox* za označavanje elemenata. Korisnik odabire elemente koji su istiniti za njegov način života te na

kraju liste odabire gumb za spremanje podataka. Podaci se spremaju na sličan način kao i podaci iz postavki, uzimanjem podatka elektroničke pošte trenutnog korisnika pomoću Firebase autentifikacije te zapisivanjem podataka u *Realtime Database* pod ključem kojeg predstavlja spomenuta elektronička pošta. Korisnik naknadno može izmjenjivati odabrane elemente, tj. označavati ili odznačavati elemente te ih može ponovno spremiti u bazu.



Sl. 2.11. *TabLayout i ViewPager - zaslon anamneze*

Treći element u navigacijskom zaslonu predstavlja gumb koji vodi na prikaz popisa svih unesenih zdravstvenih pregleda kod ginekologa tijekom trudnoće. Za prikaz liste koristi se *RecyclerView*. Na dnu zaslona nalazi se tzv. *FloatingActionButton* čijim se odabirom otvara obrazac za unos podataka novog pregleda. Nakon svakog ginekološkog pregleda potrebno je upisati sve podatke s pregleda kako bi popis bio ažuran. Obrazac sadrži polja za upis sljedeći parametara:

1. Datum pregleda
2. Tjedan trudnoće – tjedan + broj dana u narednom tjednu
3. Prirast mase – najčešće u decimalnom obliku
4. Krvni tlak – sistolički i dijastolički

5. Edemi – opisuje otekline na bilo kojem dijelu tijela majke, bez obzira o kojem dijelu tijela je riječ; koristi se podatak „DA“ ili „NE“
6. Urin – ispituje se količina proteina, glukoze i leukocita u urinu majke. Rezultantna vrijednost se izražava s „DA“ ili „NE“, odnosno važno je postojanje li ove veličine u urinu.
7. KCS ili KDS – definira se postojanje otkucaja djetetovog srca. Neki ginekolozi izražavaju vrijednost s „+“ ili „-“, dok neki uz to dodaju i izmjereni broj otkucaja u minuti.
8. Dodatni nalazi – dodatni nalazi ili savjeti i napomene ginekologa tijekom pregleda
9. Naredni pregled – datum sljedećeg kontrolnog pregleda

Nakon unosa svih podataka, podatke je potrebno spremi odabirom gumba na kojemu piše „SAVE“.

Slika 2.12. prikazuje unos podataka pregleda, dok slika 2.13. prikazuje listu unesenih pregleda.

The image displays two side-by-side screenshots of a mobile application interface for entering examination data. Both screens have a dark red header with a back arrow and the text "New examination".

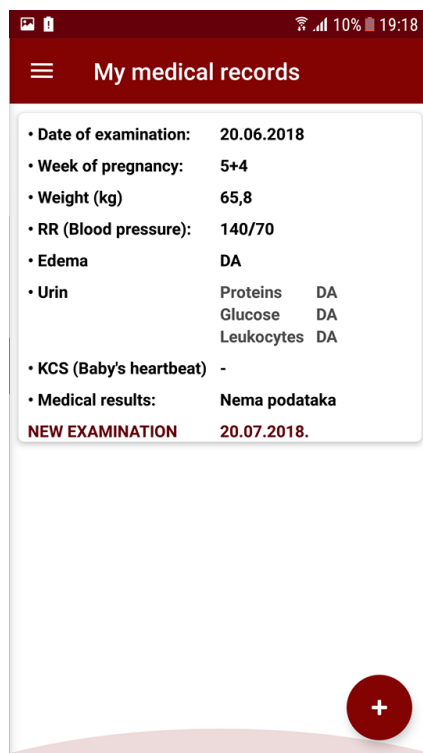
The left screenshot shows the following fields:

- Date of examination:** A text input field.
- Week of pregnancy:** Two dropdown menus, both set to "1", separated by a "+" sign.
- Weight (kg):** A text input field.
- RR (Blood pressure):** Two dropdown menus, both set to "1", separated by a "/" sign.
- Edema:** A dropdown menu set to "YES".
- Urin:** A dropdown menu set to "YES".

The right screenshot shows the continuation of the form:

- A dropdown menu set to "YES".
- Urin:** Three dropdown menus, all set to "YES": Proteins, Glucose, and Leukocytes.
- KCS (Baby's heartbeat):** A text input field.
- Medical results:** A text input field.
- NEXT EXAMINATION:** A text input field.
- A pink "SAVE" button at the bottom.

Sl. 2.12. Unos novog pregleda



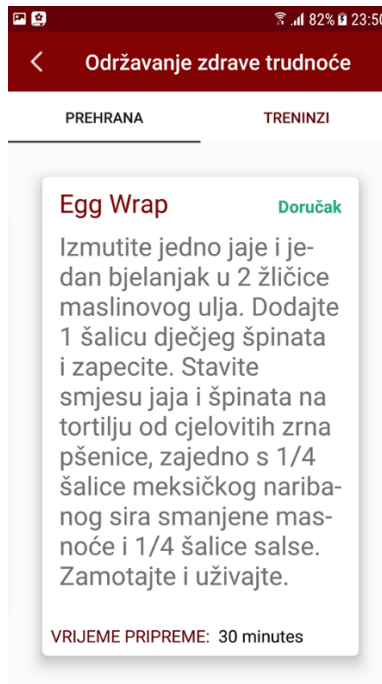
Sl. 2.13. Popis pregleda

Sve više žena se ne želi prestati baviti sportom ili drugom vrstom tjelovježbe kada uđu u trudnički period stoga aplikacija ima i za to prikladan sadržaj. Odabirom četvrtog elementa u navigacijskom prozoru otvara se aktivnost pod nazivom „Održavanje zdrave trudnoće“. U njoj je implementiran *TabLayout* bez *ViewPagera*. *TabLayout* sadrži dvije kartice od kojih jedan predstavlja odjel za prehranu, a druga predstavlja odjel za tjelovježbu. U svakoj kartici implementiran je *RecyclerView* koji sadrži elemente koji se mogu horizontalno listati što je omogućeno horizontalnom komponentom linearnog rasporeda.

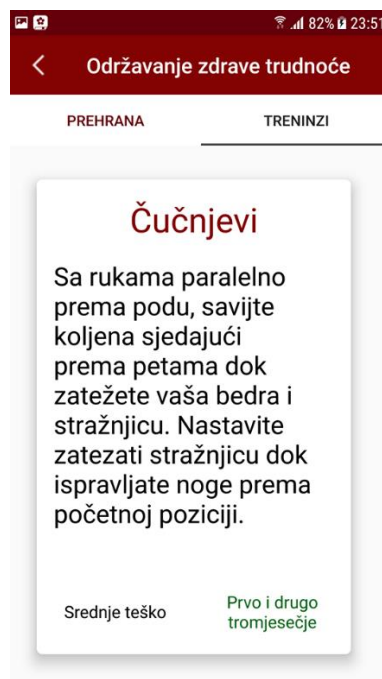
Element popisa jela u kartici za prehranu sadrži 4 tekstualna polja - naziv za jelo, kategoriju u koju se ubraja (doručak, ručak, večera ili međuobrok), opis načina pripreme te vrijeme potrebno za pripremu. Korisnik može listati po popisu te odabrati jelo koje želi, a da je prikladno za određeno vrijeme dana. Jela su definirana na način da se normalnom obroku dodalo do 300 kalorija više. Osim što je kategorija izražena tekстом, izražena je i bojama pa je tako doručak definiran zeleno, ručak plavo, večera smeđe i međuobrok žuto (Sl. 2.14.).

Elementi koji predstavljaju vježbe prikladne za trudnice nalaze se u kartici za treninge (Sl.2.15.). Jedan element sastoji se od 4 tekstualna polja koja predstavljaju naziv vježbe, opis izvođenja vježbe, težinu vježbe te podatak u kojem periodu trudnoće se može izvoditi ova vježba. Lakše vježbe definirane su za sva tromjesečja, teže za prvo i drugo tromjesečje i jako

teške samo za prvo tromjesečje. Ovo kategoriziranje vježbi prilagođeno je trudnicama koje su bile umjereno fizički aktivne u periodu života prije trudnoće. Za one trudnice koje su sportašice ili koje su se aktivno bavile fizičkom aktivnosti ova kategorizacija nije toliko važna. Preporučuje se da trudnice vježbaju 3-5 dana u tjednu. Za trudnice koje su u komplikacijama trudnoće korištenje programa treninga se ne preporučuje.

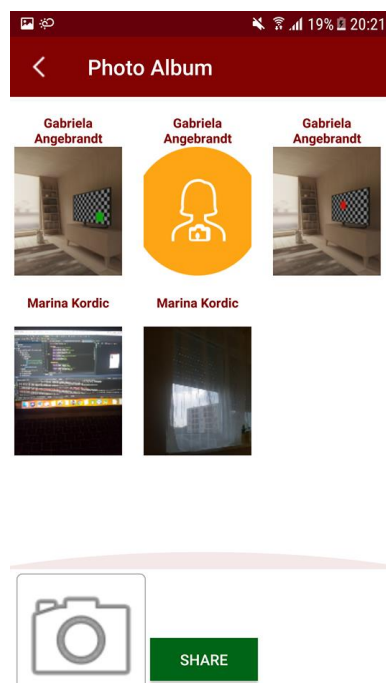


Sl.2.14. Odjel za prehranu



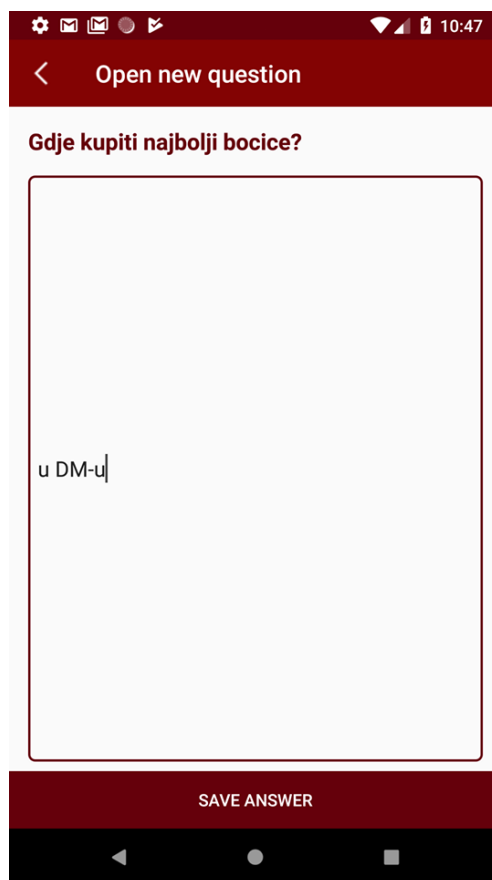
Sl. 2.15. Odjel za treninge

Odabirom petog elementa iz navigacijskog prozora otvara se zaslon koji predstavlja foto album svih korisnika aplikacije (Sl.2.16.). Na dnu zaslona nalazi se *ImageButton* čijim se odabirom otvara dijaloški okvir u kojemu se može odabrati fotografiranje ili izbor slike iz galerije. U slučaju fotografiranja poziva se temeljna aplikacija uređaja koja se koristi za fotografiranje, te nakon što je slika napravljena i potvrđena, vraća se zaslon s foto albumom i odabranom slikom prikazanom na pozadini *ImageButton-a*. U slučaju izbora slike iz galerije, postupak je sličan. Odabirom elementa iz dijaloškog okvira otvara se galerija, slika se odabire te se potvrđuje i korisnik se vraća na zaslon s foto albumom i slika se prikazuje na *ImageButton-u*. Pored ovog gumba nalazi se drugi gumb sa zelenom pozadinom i tekstom „SHARE“ koji služi za pohranjivanje odabrane slike u Firebase. Izvorna slika je u obliku Bitmap-a, a sprema se u tipu podatka String u bazu. Kodiranje i naknadno dekodiranje slike vrši se pomoću *ByteArrayOutputStream* i *Base64*. Osim slike, u bazu se sprema i ime korisnika kojem pripada fotografija. Kako bi sve pohranjene slike bile podijeljene s ostalim korisnicima, koristi se *RecyclerView* s mrežastim rasporedom koji se lista okomito. Elementi *RecyclerView-a* raspoređeni su u tri stupca i sadrže jedan *ImageView* za prikaz fotografije i jedno tekstualno polje za prikaz imena vlasnika fotografije. Nakon što su podaci dohvaćeni iz baze, potrebno je fotografiju dekodirati iz oblika String u oblik Bitmap te ju postaviti u okvir.

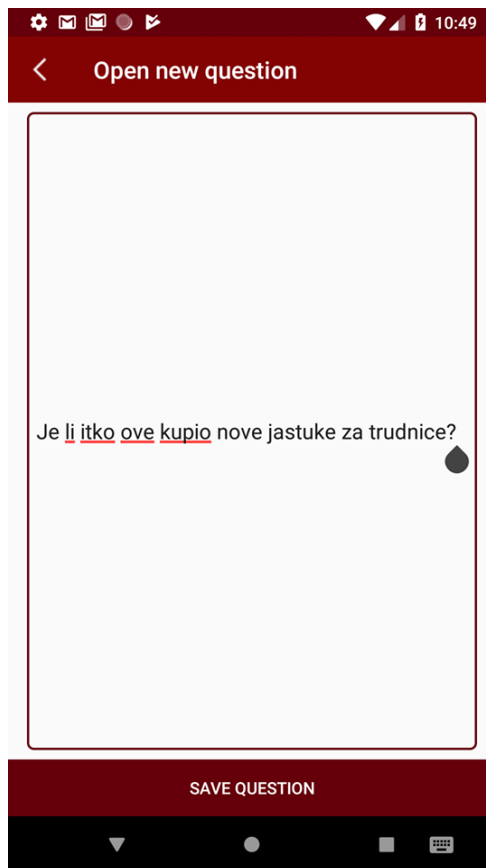


Sl. 2.16. Foto album s elementima

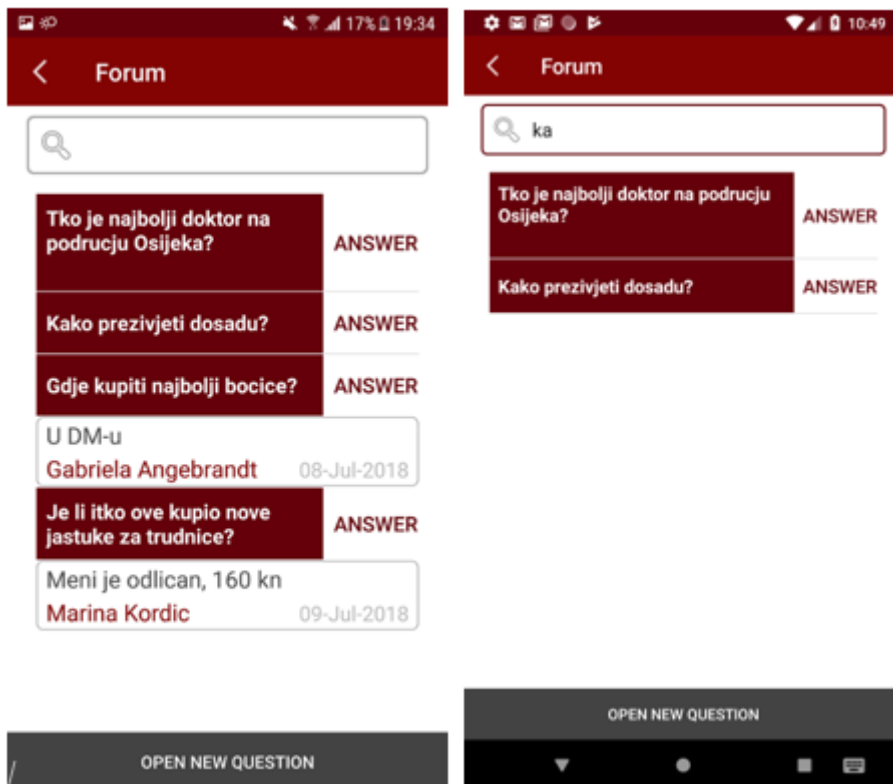
Naposljetku, šesti element u navigacijskom prozoru vodi na forum aplikacije na kojemu se mogu postavljati pitanja anonimno te se može i odgovarati na ista. U slučaju odgovora, anonimnost je isključena. U ovom rasporedu korišten je *ExpandableListView* za koji je potrebna lista pitanja ili tzv. *headers* i lista grupe odgovora odnosno *childList*. Za ovaj način prikaza potreban je poseban adapter koji će na odabir odgovora proširiti istu listu, ali elementima koji imaju drugačiji prikaza (odgovori). Kod pitanja korišteno je jedno tekstualno polje koje prikazuje tekst pitanja i jedan gumb s tekстом „ANSWER“. Ovaj gumb otvara aktivnost u kojoj se može unijeti odgovor na odabrano pitanje (Sl.2.17). Na dnu ove aktivnosti nalazi se gumb za pohranjivanje odgovora u bazu. Element odgovora sastoji se od tekstualnog polja za naziv korisnika koji je napisao odgovor, tekstualnog polja za tekst odgovora te tekstualnog polja za datum kada je odgovor stvoren. Ispod *ExpandableListView*-a nalazi se gumb za otvaranje nove teme, odnosno za upis novog pitanja. Aktivnost na koju vodi je slična kao i aktivnost za upis odgovora (Sl.2.18). Pitanja se mogu pretraživati samo po nazivu pitanja tako što se iznad *ExpandableListView*-a nalazi tekstualno polje za unos upita. Prema slici 2.19. može se prikazati rad foruma.



Sl. 2.17. Unos odgovora



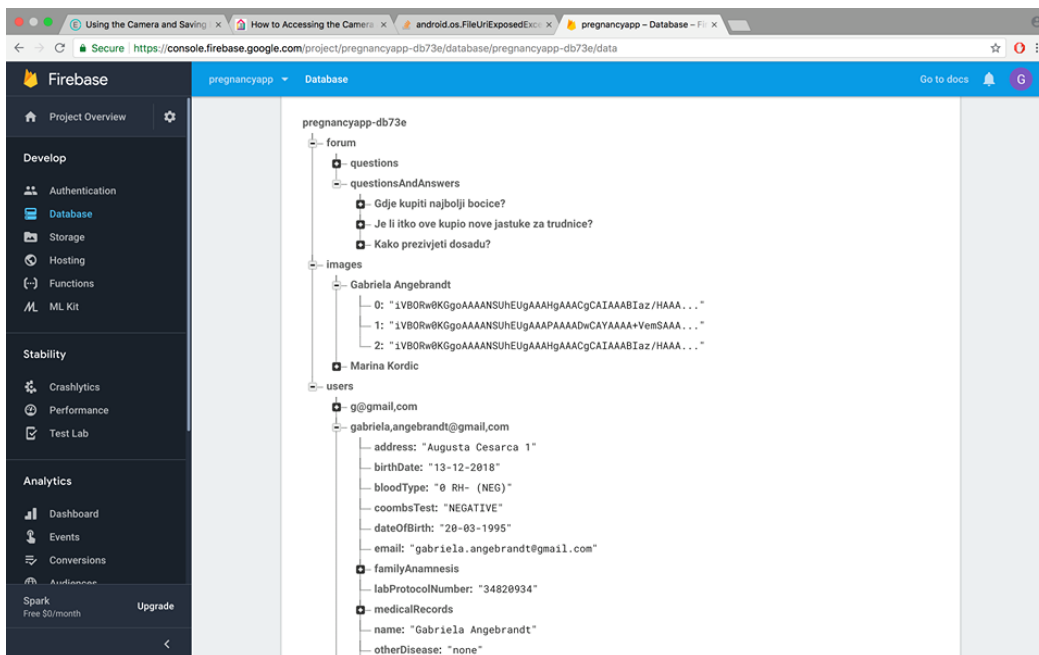
Sl. 32.18. Unos novog pitanja



Sl. 2.18. Forum

2.4. Baza podataka

Za pohranjivanje podataka korištena je baza podataka Firebase. Kod ove baze podaci se spremaju u JSON obliku, odnosno u obliku ključ-vrijednost (Sl. 2.19). Ova baza je baza stvarnog vremena i radi asinkrono, odnosno kada se pozove metoda iz baze izvođenje programa ne čeka na odgovor baze već nastavlja s izvršenjem koda. Budući da se većina podataka iz baze prikazuje u različitim listama, potrebno je nakon dobivenih podataka obavijestiti adapter liste o promjenama. Primjer je prikazan na slici 2.20.



Sl. 2.19. Hijerarhija baze podataka

```
FirebaseUser user = App.getFirebaseAuth().getCurrentUser();
if (user != null) {
    App.getFirebaseDb().getReference("forum").child("questions").addListenerForSingleValueEvent(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
            if (dataSnapshot.exists()) {
                for (DataSnapshot dataSnapshot1 : dataSnapshot.getChildren()) {
                    questionList.add(dataSnapshot1.getValue(String.class));
                }
            }
            filterList.addAll(questionList);
            adapter.notifyDataSetChanged();
            progressBar.setVisibility(View.GONE);
        }
    });

    @Override
    public void onCancelled(@NonNull DatabaseError error) {
        // Failed to read value
        Log.w(tag: "error", msg: "Failed to read value.", error.toException());
        Toast.makeText(context: ForumActivity.this, "Failed to read from database.", Toast.LENGTH_SHORT).show();
    }
};
```

Sl. 2.20. Dohvaćanje podataka iz baze

Važno je napomenuti da se u ovoj aplikaciji u bazu podataka najčešće spremaju mape (engl. *Hashmap*) ili liste (engl. *ArrayList*) jer se tako spremi najjednostavnija putanja. Najprije se pozove instanca baze podataka, zatim se definira referenca u bazi i na nju se navode čvorovi koji vode do traženog podataka. Početni čvorovi u ovoj aplikaciji su *forum*, *images* i *users*. Čvor *forum* sadrži 3 podčvorova: podčvor liste pitanja te podčvor mape pitanja i liste odgovora. *Images* sadrži podčvorove koji sadrže adresu korisnikove elektroničke pošte koja za svakog korisnika mora biti jedinstvena. Budući da čvorovi u Firebasu-u ne mogu sadržavati točku unutar naziva, točka je zamijenjena sa zarezom kako bi naziv čvora odgovarao standardu. Ovi podčvorovi sadrže liste fotografija svakog korisnika. Čvor *users* također sadrži podčvorove adrese elektroničke pošte, ali on služi za pohranjivanje osobnih podataka, podataka o anamnezi, ginekološkim pregledima itd. Podaci se u bazu podataka zapisuju tako da se koristi metoda *setValue()* na određenoj putanji (Sl.2.21).

```

FirebaseUser user = App.getFirebaseAuth().getCurrentUser();
if (user != null) {
    App.getFirebaseDb().getReference().child("users").child(user.getEmail()).replaceAll(regex: "\\.", replacement: ","), child("bloodType").setValue(bloodtype + " RH" + rh);
    App.getFirebaseDb().getReference().child("users").child(user.getEmail()).replaceAll(regex: "\\.", replacement: ","), child("coombsTest").setValue(coombsTest);
    App.getFirebaseDb().getReference().child("users").child(user.getEmail()).replaceAll(regex: "\\.", replacement: ","), child("labProtocolNumber").setValue(labProtocolNumber);
    App.getFirebaseDb().getReference().child("users").child(user.getEmail()).replaceAll(regex: "\\.", replacement: ","), child("birthDate").setValue(birthDate);
    App.getFirebaseDb().getReference().child("users").child(user.getEmail()).replaceAll(regex: "\\.", replacement: ","), child("otherDisease").setValue(other);
    Toast.makeText(getApplicationContext(), "Data saved.", Toast.LENGTH_SHORT).show();
}

```

Sl. 2.21. Spremanje podataka iz postavki u bazu

2.5. Kreiranje API-a

Podaci koji se koriste u prikazu programa treninga i zdrave prehrane su podaci dobiveni pomoću API-a. API (engl. *Application Programming Interface*) je set protokola, definicija i raznih alata koji se koriste u razvoju aplikacije. Za izradu vlastitog API-a korišten je programski jezik Node.js te programi Atom i Postman. Za početak kreiran je projekt i otvoren u Atom-u programu koji sadrži direktorije *controllers*, *models*, *node_modules* i *routes* te datoteke *indeks.js* i *package.json*. U *models* direktorij potrebno je kreirati dva modela – *Meals* i *Workouts* prema slici 2.22..


```

1 var mongoose = require('mongoose');
2 var Schema = mongoose.Schema;
3
4 var MealSchema = new Schema({
5   ingredients: {
6     type: String,
7   },
8   category: {
9     type: String,
10  }
11 });
12
13 module.exports = mongoose.model('Meals', MealSchema);
14
var mongoose = require('mongoose');
var Schema = mongoose.Schema;

var WorkoutSchema = new Schema({
  title: {
    type: String,
  },
  time: {
    type: String,
  },
  moves: {
    type: String,
  }
});

module.exports = mongoose.model('Workouts', WorkoutSchema);

```

Sl. 2.22. Modeli

Nakon modela, napisane su rute za dohvaćanje popisa jela ili treninga ili samo jednog zasebnog elementa iz tih popisa. One se nalaze u mapi *routes*. Rute su definirani nastavak URL-a na kojem se nalaze određeni podaci (Sl. 2.23). U rutama se pomoću metoda GET i POST pozivaju funkcije definirane u *controlleru* u mapi *controllers* (Sl. 2.24.). Funkcije su jednostavne i kratke te služe za ispisivanje podataka u JSON formatu. Korištenjem GET metode nije potrebno slati i dodatne parametre za uspješan pristup podacima, dok je kod metode POST najčešće to potrebno.

```

exports.list_all_workouts = function(req, res) {
  Workout.find({}, function(err, workout) {
    if (err)
      res.send(err);
    res.json(workout);
  });
};

exports.create_a_meal = function(req, res) {
  var new_meal = new Meal(req.body);
  new_meal.save(function(err, meal) {
    if (err)
      res.send(err);
    res.json(meal);
  });
};

```

Sl. 2.23. Dio kontrolera

```

app.route('/meals')
  .get(controller.list_all_meals)
  .post(controller.create_a_meal);

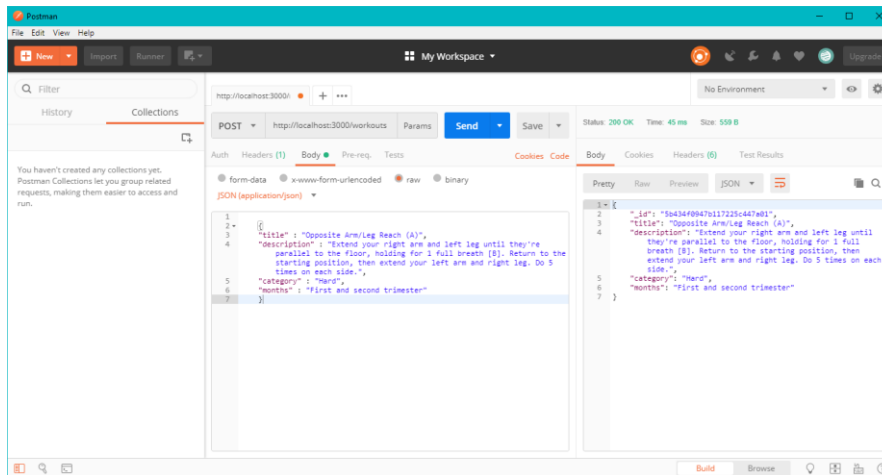
app.route('/workouts')
  .get(controller.list_all_workouts)
  .post(controller.create_a_workout);

app.route('/meals/:mealId')
  .get(controller.read_a_meal);

app.route('/workouts/:workoutId')
  .get(controller.read_a_workout);

```

Sl. 2.24. Rute



Sl. 2.25. Postman

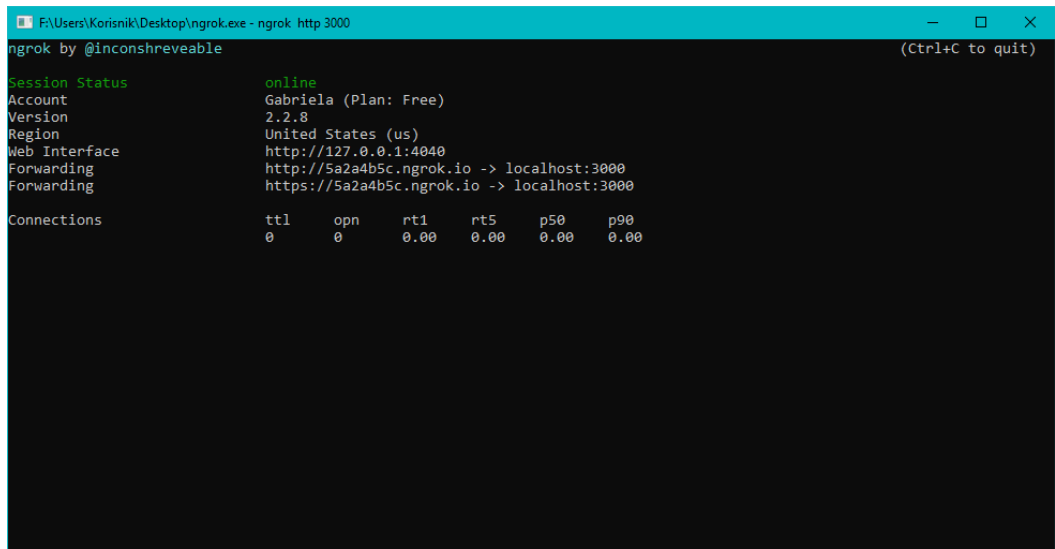
Datoteka *index.js* sadrži definicije za povezivanje *mongoose* i *body-parser* modula, za definiranje Mongo baze podataka, za definiranje datoteke koja sadrži rute te za instance ranije kreiranih modela.

Datoteka *package.json* sadrži osnovne informacije o projektu kao što su inačica, glavna .js datoteka, autor itd.

Prema slici 2.25., nakon izrade projekta, korišten je program Postman u kojemu se može provjeriti valjanost rute te se može podacima popuniti JSON. Prije rada u Postmanu potrebno je u naredbenom retku pokrenuti *nodemon* – jednostavnu nadzornu skriptu kod razvoja softvera te Mongo bazu naredbom *mongod.exe*. Kako bi mobilna aplikacija imala pristup podacima koji su uneseni u Postmanu korišten je *ngrok.exe*. To je virtualni tunel koji omogućuje povezivanje domene s lokalnom Mongo bazom (Sl.2.26), ali samo dok je tunel otvoren, tj. dok je *ngrok.exe* pokrenut. Svaki puta kada se ponovno pokrene, poveznica na domenu se mijenja. Pokreće se naredbom „ngrok http 3000“, s tim da 3000 predstavlja port koji koristi *localhost* (Sl.2.27).



Sl. 2.26. Izgled API-a



Sl. 2.27. ngrok.exe

Poslije izrade API-a, u Android aplikaciji definira se interaktor koji će sadržavati RxJava implementaciju. RxJava je biblioteka koja se koristi kod usklađivanja asinkronih i programa vođenih događajima. Ona koristi *Observable* tip podatka koji je specifičan po tome da može dohvatiti podatke s nekog izvora i da se na slušanje promjena podataka u njemu može pretplatiti/registirirati bilo koji objekt koji to želi. S druge strane tu je *Observer* objekt koji se registrirao za slušanje događaja promjene, odnosno koji želi biti obaviješten kada se neki *Observable* objekt promijeni.

U direktoriju modeli kreiran je direktorij *interactors* i u njemu datoteka *MainInteractorImpl* (Sl.2.28). *MainInteractor* sadržava funkcije koja dohvaćaju podatke s

prethodno definiranih ruta. Nakon što se podaci dohvate s vanjskog izvora, šalju se u *presenter* koji ih je zatražio te ih *presenter* prosljeđuje u *view*.

```
public class MainInteractorImpl extends BaseInteractorImpl implements MainInteractor {
    @Override
    public void getMeals(final MealListener listener) {
        addObserver(getMeals(), subscribeOn(Schedulers.io()).observeOn(AndroidSchedulers.mainThread()).subscribeWith(new DisposableObserver<List<Meal>>() {
            @Override
            public void onNext(List<Meal> meals) { listener.onSuccess(meals); }

            @Override
            public void onError(Throwable e) { listener.onError(); }

            @Override
            public void onComplete() {
            }
        }));
    }
}
```

Sl. 2.28. MainInteractor za dohvaćanje popisa za odjel prehrane

2.6. Prijevod aplikacije

Aplikacija je prevedena na hrvatski i engleski jezik. Jezik koji će se koristiti ovisi o tome koji je postavljeni glavni jezik na uređaju. Prijevod aplikacija vrši se na način da se svi tekstovi koji će se ispisivati na zaslonima spremaju u datoteku koja se zove *strings.xml*. Jedna ovakva datoteka dobije se s kreiranjem projekta. Ukoliko je potrebno dodati druge jezike, potrebno je u *res* direktoriju napraviti direktorij pod nazivom *values-hr*, gdje „hr“ predstavlja ISO 639-1 jezični kod. Nakon što je postavljen ovaj direktorij u njega se dodaje nova datoteka *strings.xml* koja ima isto ime kao i datoteka dobivena kreiranjem projekta, ali drugačiju putanju. Na taj način aplikacija zna koju datoteku treba koristiti. Tekstovi se zapisuju unutar elementa *string* gdje je potrebno navesti ključno ime određenog teksta i vrijednost tekst nakon toga. Važno je da sve *strings.xml* datoteke u aplikaciji imaju ista ključna imena elemenata, ali različite vrijednosti teksta.

3. ZAKLJUČAK

Android aplikacija za praćenje trudnoće je aplikacija koja omogućava pohranjivanje podataka vezanih za trudnoću. Za razvoj aplikacije korišten je razvojni alat Android Studio koji se ističe po svojoj jednostavnosti i fleksibilnosti. Za korištenje aplikacije potrebno je izvršiti prijavu u sustav kojoj prethodi registracija u sustav. Osim ovog načina, omogućeno je da se korisnik može prijaviti s Facebook ili Gmail uslugom ukoliko već posjeduje korisnički račun na jednoj od ove dvije usluge. Za prijavu se koristi Firebase autentifikacija. Nakon uspješne prijave, otvara se mogućnost pohranjivanja podataka o različitim zdravstvenim parametrima koji su važni kako bi majka i dijete bili što zdraviji. Neki od tih parametara su prirast mase, krvni pritisak, broj otkucaja djetetovog srca itd. Podaci se spremaju u Firebase bazu podataka stvarnog vremena. Aplikacija sadrži i popis obroka i vježbi koje su prilagođene trudnicama. Podaci su preuzeti s izvora [9] i [10], te su prevedeni i na hrvatski. Ovi podaci dobivaju se pomoću API-a. Za posluživanje odabran je *ngrok.exe* koji stvara virtualni kanal između lokalnog i udaljenog poslužitelja. Ovo nije najuspješnije rješenje zadatka jer zahtjeva lokalno pokretanje *ngrok.exe*-a na računalu te podaci nisu dostupni kada *ngrok.exe* nije pokrenut. Na početku ovog dijela zadatka za poslužitelj je odabran Firebase, no kroz zadatak nije se mogla uspostaviti pravilna veza Firebase-a s Mongo bazom podataka.

Kako aplikacija ne bi bila orijentirana samo na zdravlje, ona sadrži i oblik društvene mreže, odnosno forum i foto album. Na forumu mogu se postavljati različita pitanja vezana za trudnoću te se na njih može i odgovarati. Ovaj forum služi za informiranje budućih majki i za dijeljenje savjeta na određene probleme. Foto album omogućava dijeljenje fotografija među korisnicima. Korisnici odaberu fotografiju, odabirom iz galerije uređaja ili trenutnim fotografiranjem koristeći primarnu aplikaciju uređaja za pokretanje kamere, te ju mogu podijeliti s drugima odabirom samo jednog gumba. Za svaku fotografiju definirano je ime vlasnika iznad nje. Pitanja i odgovori iz foruma te podijeljene fotografije također se spremaju u Firebase bazu podataka stvarnog vremena.

Nakon završenog rada u aplikaciji, korisnik ima mogućnost odjave iz sustava. Korisnik se ne mora prijavljivati u aplikaciju svaki put kad ju otvori, već su podaci o prijavi pohranjeni sve dok se korisnik ne odjavi. U slučaju da je korisnik zaboravio zaporku, Firebase-om je omogućeno da se na adresu elektroničke pošte pošalje lozinka korisničkog računa u ovoj

aplikaciji, no uvjet je da korisnik ispuni sve tražene podatke koji se tada uspoređuju s podacima koji su uneseni pri registraciji.

Aplikacija za praćenje trudnoće izrađena je tako da podržava tekstove na hrvatskom i engleskom jeziku.

Svrha ove aplikacije je pohraniti sve podatke o trudnoći u bazu podataka kojoj se može pristupiti uvijek te ne postoji mogućnost da se svi podaci obrišu brisanjem aplikacije s mobilnog uređaja ili drugim postupcima. Uređivanju baze ima pristup samo razvojni inženjer. Također, ova aplikacija je digitalna zamjena za običnu papirnatu trudničku knjižicu koju trudnice znaju zagubiti. U budućnosti, ova aplikacija ima mogućnost daljnjeg razvijanja, npr. ostvariti da korisnik može pohranjivati podatke o više trudnoća, implementirati podatke o trudničkim tečajevima, omogućiti korisnicima interakciju s fotografijama (npr. funkcija „Sviđa mi se“), razviti web aplikaciju za doktore koji će odmah pri pregledu unositi podatke a zatim će se podaci sinkronizirati s mobilnom aplikacijom itd.

LITERATURA

- [1] Sve za bebe, Uvog u vježbanje u trudnoći, <http://www.svezabebe.ba/beba/trudnoca/277-uvod-vjezbanje-trudnoca> (02.07.2018)
- [2] pptrns, <https://ptrns.com/?scid=17/>, (03.02.2018)
- [3] Google, <https://material.io/guidelines/components/>, (03.02.2018.)
- [4] Design Mantic, <https://www.designmantic.com/logo-design/create?design=5497>, (03.02.2018)
- [5] Grover, Ankush, Basics of MVP—The Android Way, Hackernoon, <https://hackernoon.com/basics-of-mvp-the-android-way-f75da407019d> (27.06.2018.)
- [6] Uncle, Bob, The Clean Architecture, 8thlight, <https://8thlight.com/blog/uncle-bob/2012/08/13/the-clean-architecture.html> (27.06.2018)
- [7] Cervone, Francesco, Model-View-Presenter: Android guidelines, Medium, <https://medium.com/@cervonefrancesco/model-view-presenter-android-guidelines-94970b430ddf> (27.06.2018.)
- [8] Arriola, Christopher, Meet RxJava: The Missing Reactive Programming Library for Android, toptal, <https://www.toptal.com/android/functional-reactive-android-rxjava>, (27.06.2018.)
- [9] Baeldung, Introduction to RxJava , Baeldung, <http://www.baeldung.com/rx-java> (27.06.2018)
- [10] Parents, A Week of Delicious Pregnancy Meals and Snacks, <https://www.parents.com/pregnancy/my-body/nutrition/prenatal-meal-plan/> (08.07.2018)
- [11] fitPregnancy, Excercise, <https://www.fitpregnancy.com/exercise/prenatal-workouts/30-minute-total-body-workout> (08.07.2018)
- [12] ngRok, What is ngrok?, <https://ngrok.com/product>, (08.07.2018)
- [13] Firebase, Add Firebase to Your Android Project, <https://firebase.google.com/docs/android/setup?authuser=0> (05.07.2018)

[14] Firebase, Get Started with Firebase Hosting,

<https://firebase.google.com/docs/hosting/quickstart?authuser=0> (07.08.2018)

[15] mLab, express-mongodb-setup, <https://github.com/mlab/express-mongodb-setup>,
(09.07.2018)

[16] mongoDB, What is mongoDb?, <https://www.mongodb.com/what-is-mongodb>
(08.70.20178)

[17] Developers, Localize Your App,

<https://developer.android.com/guide/topics/resources/localization> (01.07.2018)

SADRŽAJ

Android aplikacija za praćenje trudnoće izrađena u ovom diplomskom radu namijenjena je za sve buduće majke koje žele podatke o svojoj trudnoći pohranjivati na svom pametnom telefonu. Za početka korištenja aplikacije potrebno je izvršiti registraciju te prijavu u sustav. Prijava može biti lokalna te putem Facebook ili Gmail usluge. Nakon prve prijave, potrebno je upisati važne informacije kao što su osobna anamneza, obiteljska anamneza, krvna grupa, datum koji je određen kao datum poroda i dr. Budući da je u periodu trudnoće potrebno pratiti određene parametre koji ukazuju na zdravlje majke i djeteta (krvni tlak, prirast mase, edemi itd.), a koji se očitavaju na redovitim zdravstvenim pregledima, korisnice aplikacije imaju mogućnost pohranjivanja vrijednosti spomenutih parametara u obliku kartica. Kako bi trudnice imale normalnu trudnoću, potrebna je i pravilna prehrana te svakodnevna fizička aktivnost pa se, prema tome, u aplikaciji prikazuju programi prehrane i treninga namijenjenih isključivo za trudnice. Nadalje, otvoren je i forum na kojemu korisnice mogu postavljati pitanja anonimno te odgovarati na ista. Korisnice također mogu koristiti i oblik društvene mreže na kojoj mogu javno postavljati slike vezane za svoju trudnoću.

Ključne riječi: Android, aplikacija, trudnoća, prijava, registracija, prehrana, fizička aktivnost, zdravstveni pregledi, parametri, forum, pitanja, odgovori, društvena mreža, slike

ABSTRACT

Android application for pregnancy monitoring

Android application for pregnancy monitoring is developed for all future mothers who want to store data about their pregnancy on their smartphones. Registration and login is required to use all application features. Login can be achieved using standard local credentials or by using Facebook or Gmail services. After the user has registered for the first time it is required to enter some personal information like personal anamnesis, family anamnesis, birth date, blood type etc. Since it is necessary to monitor specific parameters during pregnancy which affect the health of mother and child (blood pressure, weight gain etc.) and which are monitored by regular medical check-up, users are able to store mentioned parameters in form of cards. To ensure that pregnant women have normal pregnancy process it is important for them to have proper nutrition and everyday exercises. There are nutrition and exercise programs implemented in app to help them with that. Furthermore, there is a forum that users can use to post anonymous questions and to answer them. Users can also use kind of a social network to share their favorite pregnancy images.

Key words: Android, application, pregnancy, login, registration, proper nutrition, exercises, medical check-up, parameters, forum, questions, answers, social network, images

ŽIVOTOPIS

Gabriela Angebrandt rođena je 20. ožujka 1995. godine u Đakovu. Pohađala je osnovnu školu Ivana Gorana Kovačića u Đakovu. Nakon završenog osnovnoškolskog školovanja, upisala je opću gimnaziju Antuna Gustava Matoša u Đakovu gdje je maturirala 2013. Iste godine upisala je Fakultet elektrotehnike, računarstva i informacijskih tehnologija u Osijeku na kojem se školuje i danas. U 2015. godini bila je potpredsjednica Computer Society-a IEEE studentskog ogranka FERIT-a. U 8.mjesecu 2016. godine radila je kao student u tvrtki INEL, Kralja Stjepana Držislava 23, Đakovo te je sudjelovala u raznim projektima javne rasvjete i energetskog certificiranja. U ljeto 2017. godine završila je program stručne prakse tvrtke Plava Tvornica, Lorenza Jägera 2, Osijek u trajanju od 6 tjedana. Stručna praksa temeljila se na razvoju Android mobilne aplikacije uz korištenje RxJave i Dagger2 *frameworka*. Gabriela radi kao student za tvrtku Kod savjetovanje d.o.o, Hrvatske nezavisnosti 15, Vukovar na razvoju mobilnih aplikacija od veljače 2018.godine.

Gabriela ima iskustva u korištenju programskih alata Microsoft Office, Android Studio, Visual Studio, MATLAB, Photoshop, AutoCad, Atom, Xcode itd. Gabriela se služi programskim jezicima C, C++, Java, Swift, a poznaje i Kotlin, Ruby, JavaScript i PHP.

Vlastoručni potpis:

_____.

Angebrandt Gabriela