

MOBILNA ANDROID APLIKACIJA ZA POTPORU I PRAĆENJE TJELOVJEŽBE ZASNOVANA NA ANALIZI PODATAKA

Pavlović, Zvonimir

Undergraduate thesis / Završni rad

2019

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:200:308593>

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

Download date / Datum preuzimanja: **2025-02-02**

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

**MOBILNA ANDROID APLIKACIJA ZA POTPORU I
PRAĆENJE TJELOVJEŽBE ZASNOVANA NA ANALIZI
PODATAKA**

Završni rad

Zvonimir Pavlović

Osijek, 2019

Sadržaj

1. UVOD	1
1.1 Zadatak završnog rada	1
2. SVAKODNEVNA TJELESNA AKTIVNOST	2
2.1 Tjelesna aktivnost	2
2.2 Važnost svakodnevne tjelesne aktivnosti u životu čovjeka.....	3
2.3 Problemi koji nastaju pri manjku tjelesne aktivnosti.....	3
2.4 Prikaz postojećih rješenja za praćenje tjelesne aktivnosti	3
2.4.1 Samsung Health	3
2.4.2 Google Fit	4
2.4.3 BodBot: Fitness Trainer	5
2.5 Idejno rješenje vlastite aplikacije za potporu i praćenje tjelovježbe	6
3. MODEL APLIKACIJE ZA POTPORU I PRAĆENJE TJELOVJEŽBE	7
3.1 Korisnički zahtjevi na mobilnu aplikaciju za potporu i praćenje tjelovježbe.....	7
3.2 Registriranje korisnika.....	7
3.3 Prijava korisnika	7
3.4 Unos parametara pri prvoj prijavi.....	7
3.5 Obrada podataka	8
3.6 Dodjela vježbi zasnovana na unesenim podacima	9
3.7 Uređivanje korisničkog profila	11
3.8 Ažuriranje podataka o masi korisnika	11
3.9 Detaljan pregled svake pojedine vježbe	11
3.10 Odjava korisnika	11
4. PROGRAMSKO RJEŠENJE APLIKACIJE ZA POTPORU I PRAĆENJE TJELOVJEŽBE	13
4.1 Opis korištenih alata i tehnologija	13
4.1.1 Android Studio	13
4.1.2 Programski jezik Kotlin	14
4.1.3 Model-Pogled-Prezenter arhitekturni obrazac	14
4.1.4 Firebase	15
4.1.5 XML.....	16
4.2 Prikaz programskog rješenja po komponentama.....	17
4.2.1 Registriranje i prijava korisnika	17
4.2.2 Rad s bazom podataka Firebase	19

4.2.3 Postupak unosa parametara.....	20
4.2.4 Obrada unesenih podataka	21
4.2.5 Dohvaćanje prikladnog skupa vježbi	22
4.2.6 Ažuriranje podataka o masi korisnika.....	23
4.2.7 Detaljan prikaz pojedine vježbe	25
4.2.8 Dodavanje odrađene vježbe u povijest.....	26
4.2.9 Pregled povijesti odrađenih vježbi.....	27
5. PRIKAZ RADA APLIKACIJE I ANALIZE REZULTATA	29
5.1 Prikaz ostvarenih funkcionalnosti aplikacije.....	29
5.2 Prikaz korištenja aplikacije.....	29
5.2.1 Prikaz registriranja korisnika	29
5.2.2 Prikaz prijave korisnika	30
5.2.3 Prikaz unosa parametara	32
5.2.4 Prikaz pregleda korisničkog profila	33
5.2.5 Prikaz ažuriranja podataka o masi	34
5.2.6 Prikaz uređivanja profila.....	35
5.2.7 Prikaz detaljnog pregleda vježbe	35
5.2.8 Prikaz dodavanja vježbe u povijest.....	36
5.2.9 Prikaz pregleda povijesti vježbanja.....	37
5.3 Slučajevi ispitivanja funkcionalnosti aplikacije	38
5.3.1 Ektomorfni slučaj	38
5.3.2 Endomorfni slučaj	39
5.3.3 Mezomorfni slučaj	40
5.4 Usporedba slučajeva ispitivanja aplikacije.....	41
ZAKLJUČAK	42
LITERATURA.....	43
SAŽETAK.....	45
ABSTRACT	46
ŽIVOTOPIS	47
PRILOZI (na CD-u)	48

1. UVOD

Tjelesna aktivnost uvelike je bitna za održavanje zdravog života. Ona se provodi tijekom svakodnevnih radnji poput šetnje, odlaska na posao, spremanja kuće i drugih radnji. Razina aktivnosti može se povećati pomoću sporta ili neke druge rekreativne aktivnosti. Međutim, razvoj tehnologije populaciju sve više usmjerava sjedilačkom načinu života, pa se razina tjelesne aktivnosti smanjuje na minimum. Normalna razina tjelesne aktivnosti smanjuje rizik za pojavu bolesti kao što su visok krvni tlak i kolesterol, pretilost te depresija. Korištenje ove aplikacije olakšalo bi korisnicima da njihova tjelesna aktivnost bude barem na prosječnoj razini.

Tema ovog završnog rada je izrada mobilne aplikacije koja će olakšati korisniku praćenje tjelesne aktivnosti te pomoći pri održavanju iste na potrebnoj razini. Aplikacija omogućuje registriranje pri kojem se unose parametri koji fizički opisuju korisnika. Na temelju unesenih parametara stvara se prikladan skup vježbi čije se svakodnevno izvođenje korisniku preporučuje. U praktičnom dijelu rada ostvareno je programsko rješenje aplikacije razvijene za Android platformu te namijenjeno osobama koje žele poboljšati razinu tjelesne aktivnosti.

U drugom poglavlju objašnjava se važnost svakodnevne tjelesne aktivnosti, te problemi koji nastaju pri smanjenoj razini aktivnosti. Također, navedena su postojeća rješenja za praćenje tjelesne aktivnosti. Treće poglavlje prikazuje model aplikacije za potporu praćenja tjelesne aktivnosti. U njemu je opisana svaka pojedina komponenta i prikazano korisničko sučelje aplikacije. Četvrto poglavlje sadrži programsko rješenje aplikacije u kojem je predstavljena i programska okolina u kojem je ona izrađena. Opisane su sve tehnologije korištene pri izradi, a svi dijelovi programskog rješenja popraćeni odgovarajućim opisom programskog koda. U petom poglavlju može se vidjeti prikaz rada i postupak korištenja aplikacije. Aplikacija je ispitana na tri različite vrste ulaznih podataka te su objašnjeni rezultati ispitivanja.

1.1 Zadatak završnog rada

U radu je potrebno objasniti ulogu redovite tjelovježbe i kretanja u očuvanju zdravlja čovjeka, te probleme koji nastaju zbog nedovoljne tjelesne aktivnosti. Također, treba izraditi mobilnu aplikaciju koja bi olakšala praćenje tjelesne aktivnosti i ponudila prikladan skup vježbi na temelju unesenih parametara. Ostvarenu aplikaciju treba ispitati na prikladnom skupu ulaznih podataka.

2. SVAKODNEVNA TJELESNA AKTIVNOST

U ovom poglavlju analiziran je pojam tjelesne aktivnosti. Obradena je njena uloga u životu, te važnost za zdravlje čovjeka. Navedena je i problematika koja se pojavljuje kada tjelesna aktivnost nije na potrebnoj razini, što može uzrokovati mnoštvo loših iskustava po čovjeka. Na kraju poglavlja navedena su postojeća rješenja za praćenje tjelesne aktivnosti, većinom predvođena velikim tehnološkim tvrtkama.

2.1 Tjelesna aktivnost

Prema [1], tjelesna aktivnost definira se kao skup kretnji tijela proizvedenih koštano-mišićnim sustavom tijekom koje se troši veća količina energije od količine u stanju mirovanja. Tjelovježba je jedan od oblika tjelesne aktivnosti s ciljem poboljšanja tjelesne sposobnosti te održavanje iste [2]. Prema Svjetskoj zdravstvenoj organizaciji (WHO), pod tjelesnom aktivnošću podrazumijevaju se kretnje u svakodnevnom životu poput kućanskih poslova, transporta, sportskih aktivnosti, itd. [3]. Važno je naglasiti da su programi tjelesne aktivnosti različiti te ih treba pravilno prilagoditi pojedincu i njegovim mogućnostima, a prema razini intenziteta može biti niska, umjerena i visoka. U obzir se uzimaju dob, spol, zanimanje postojeće bolesti i poremećaji te razina edukacije [4]. Provođenjem redovite tjelesne aktivnosti smanjuje se rizik prerane smrti na način da: smanjuje razvoj koronarnih bolesti, dijabetesa, moždanog udara, drugog srčanog infarkta (kod onih osoba koje su već jednom bolovale od istog), karcinoma, snižava se razina kolesterola u krvi, pomaže u postizanju zdrave tjelesne težine, smanjuje depresiju i osjetljivost na stres te općenito unaprjeđuje psihičko zdravlje [5]. Redovitim vježbanjem osoba čuva svoje fizičko i emocionalno stanje, a dokazano je da ponavljajuća tjelesna aktivnost u svim aspektima života uvelike povećava ukupnu kvalitetu života [6]. Kako bi postigli i očuvali dobro zdravlje, preporučeno je prakticiranje bilo kojeg oblika tjelesne aktivnosti u periodu od trideset minuta na dan. Aktivnost kao takva ne mora nužno biti sportskog tipa. Svaka pa makar i najmanja tjelesna aktivnost čini osobu pokretljivijom i zadovoljnijom, stoga je srednja razina tjelesne aktivnosti dovoljna u smislu smanjenja zdravstvenih rizika i prevencije bolesti, dok je preveliki intenzitet u odnosu na pripremljenost organizma štetan za čovjeka i njegovo zdravlje. Većina fizioloških odgovora organizma uzrokovanih umjerenom tjelesnom aktivnošću utječu na poboljšanje zdravlja unaprjeđujući strukturu i funkcionalne sposobnosti organa, dok smanjena tjelesna aktivnost ili potpuna neaktivnost povećavaju rizik od pojave mnogih bolesti [7]. Tijekom provođenja tjelesne aktivnosti, u organizmu se događaju brojne promjene. Kontinuirano ponavljanje tjelesne aktivnosti potiče organizam na trajnu prilagodbu u smislu strukturalnih i metaboličkih promjena. Posljedica

toga jest snižena frekvencije srca, povećana prokrvljenost mišića, koordinacija pokreta te bolja opskrba aktivnih mišića krvlju [8].

2.2 Važnost svakodnevne tjelesne aktivnosti u životu čovjeka

Osim pozitivnog učinka vježbanja na fizičko zdravlje, tjelesna aktivnost, također, ima pozitivan učinak i na psihičko zdravlje pojedinca. Od davnina poznata je činjenica da su duh i tijelo međusobno povezani te da oni zapravo čine jednu cjelinu. Redovita tjelesna aktivnost ima preventivnu ulogu u nastajanju dugotrajnog bolovanja, psihičkih simptoma i lošeg općeg zdravstvenog stanja te je lijek za glavne javnozdravstvene probleme. Ne samo što uvelike doprinosi fizičkom aspektu pojedinca, učestalo provođenje tjelesnih aktivnosti ima značajnu ulogu u upravljanju psihičkog dijela, odnosno utječe na stres, raspoloženje, osjećaj općeg zadovoljstva i u povećanje samopouzdanja. [9]. Također, dokazano je da tjelesna aktivnost uzrokuje smanjenje simptoma depresije i anksioznosti.

2.3 Problemi koji nastaju pri manjku tjelesne aktivnosti

Tjelesna neaktivnost predstavlja značajan zdravstveni problem današnjice i jedan je od vodećih uzroka smrtnosti u svijetu [7]. Kronične nezarazne bolesti glavni su uzrok morbiditeta i mortaliteta u svijetu, a glavni čimbenik rizika za razvoj tih bolesti tjelesna je neaktivnost. [10.]. Ljudski organizam je predodređen za fizičku aktivnost, stoga u situacijama dugotrajne neaktivnosti pokazuje znakove opadanja funkcija. Nedostatak kretanja smatra se destruktivnim za organizam, a uzrokuje česte smetnje u regulaciji krvotoka, srčani infarkt, pretilost, česte greške i slabosti tijekom tjelesnog držanja, atrofiju, itd. [11]. Porast životne dobi zajedno sa smanjenjem tjelesne aktivnosti uzrokuje rast broja kroničnih bolesti poput moždanog udara, osteoporoze, srčanog infarkta, povišenog krvnog tlaka, dijabetesa, itd. [8]. Ubrzani tempo života i globalizacija dovele su do toga da se ljudi sve manje bave tjelesnim aktivnostima, stoga dolazi do niza negativnih posljedica na čovjekovo fizičko, ali i psihičko zdravlje.

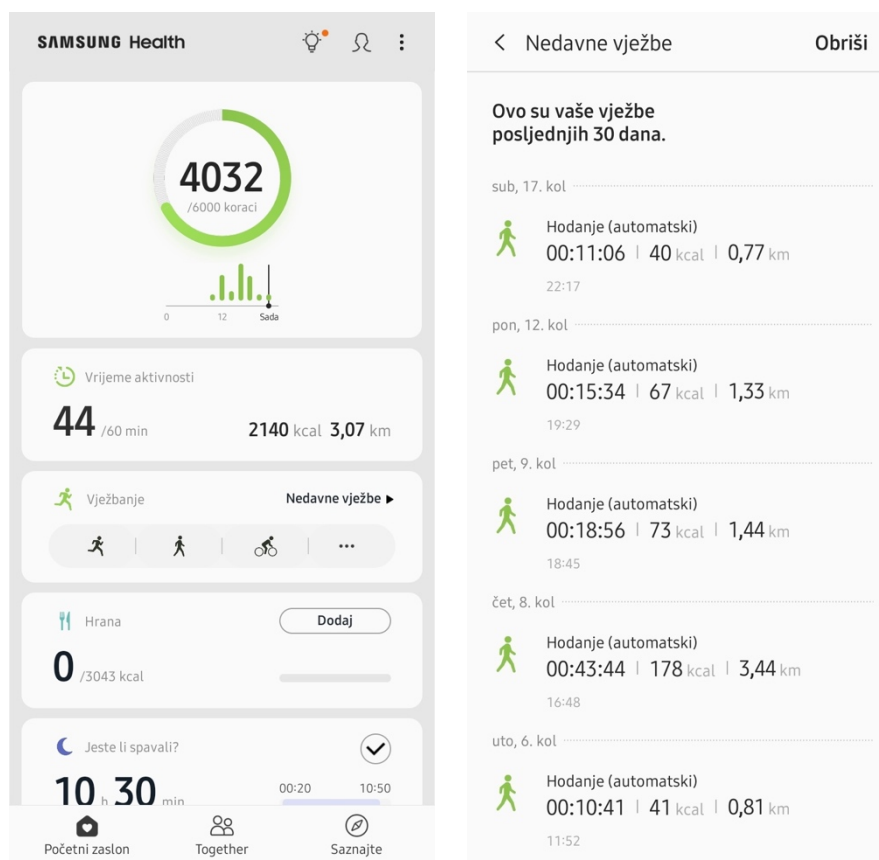
2.4 Prikaz postojećih rješenja za praćenje tjelesne aktivnosti

Na tržištu već postoji mnoštvo vrlo razvijenih rješenja za praćenje tjelesne aktivnosti. Rješenja su razvijena i u web i u mobilnim tehnologijama kako bi se proširila grupa korisnika. U nastavku će biti prikazana tri mobilna rješenja razvijena od strane velikih kompanija.

2.4.1 Samsung Health

Samsung, vodeći brand mobilnih uređaja i ostale tehnologije, razvio je aplikaciju [12] koja prati korisnikovu aktivnost, podatke o stresu, otkucaje srca i spavanje. Na temelju tih informacija daje korisne informacije i savjete kako održati zdrav život. Jedna od značajki aplikacije je brojač koraka

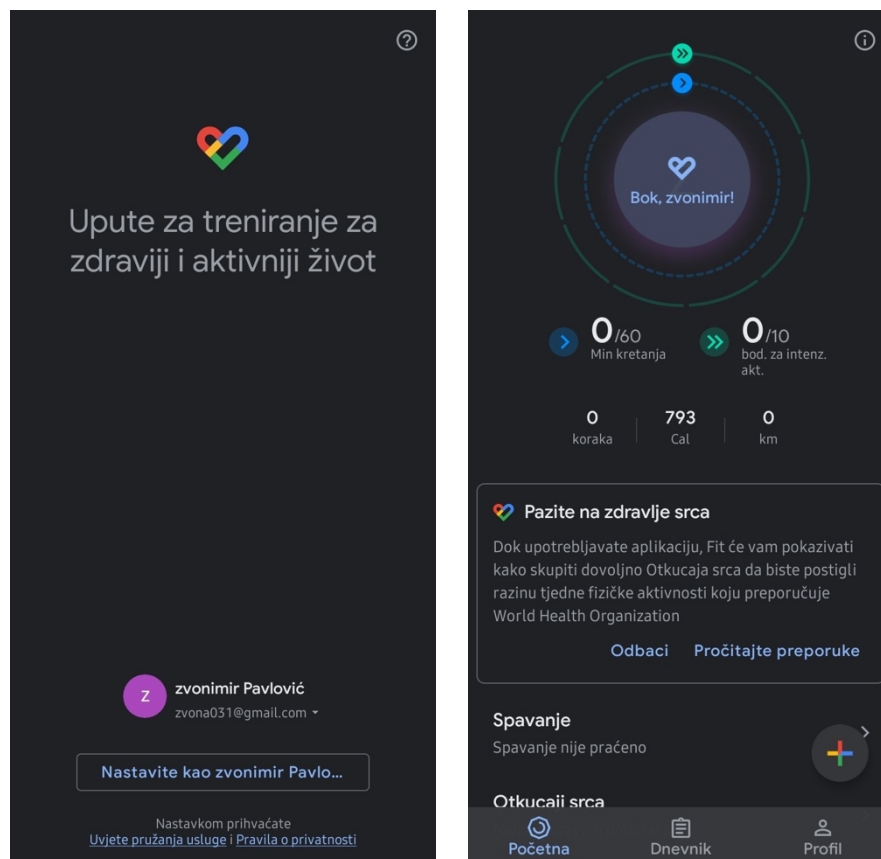
koji na temelju prijedehenih koraka računa potrošene kalorije. Time korisniku daje do znanje je li pojedini dan bio dovoljno aktivan. Korisničko sučelje same aplikacije može se vidjeti na slici 2.1.



Slika 2.1. Sučelje aplikacije Samsung Health

2.4.2 Google Fit

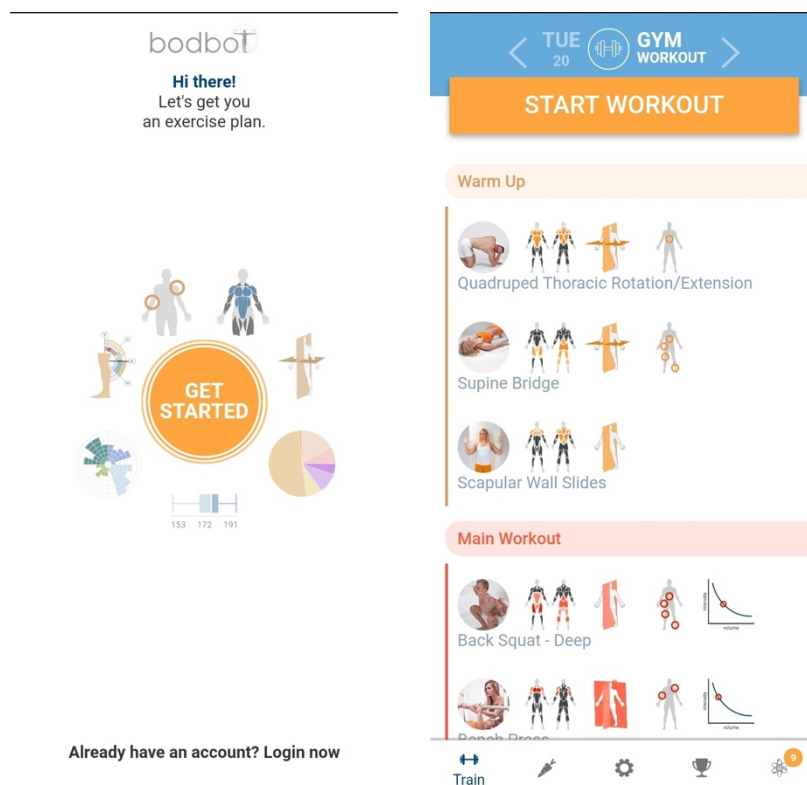
Prema [13], u suradnji sa Svjetskom zdravstvenom organizacijom (eng. WHO), Google je razvio aplikaciju koja prati dvije stvari. Prva je vrijeme u danu koje korisnik provodi u pokretu, a druga se temelji na minutama u kojima njegovo srce radi jače nego uobičajeno. Što je tih minuta više, korisnik više razvija srčani mišić, te se bolje osjeća. Aplikacija omogućuje lako kreiranje korisničkog računa koji se povezuje s već postojećim *gmail* računom. Slika 2.2 prikazuje izgled aplikacije Google Fit.



Slika 2.2. Sučelje aplikacije Google Fit

2.4.3 BodBot: Fitness Trainer

BodBot: Fitness Trainer [14], vidljiv na slici 2.3, je osobni digitalni trener. Na temelju korisnikovih želja, mogućnosti, fokusa, odabire prilagođen skup vježbi uz koje će napredak biti najizraženiji. Velika prednost ove aplikacije je to što se može prilagođavati na korisnikove nedostatke. Ukoliko korisnik ima bol u ramenima, aplikacija će odabrati vježbe u kojima ramena nisu toliko korištena, kako bi se olakšalo izvođenje vježbi. Plan treninga se također prilagođava obavezama korisnika, te korisnik sam odabire koliko puta tjedno i kojim danima želi vježbati.



Slika 2.3. Sučelje aplikacije BodBot: Personal Trainer

2.5 Idejno rješenje vlastite aplikacije za potporu i praćenje tjelovježbe

Izazovi opisani u dosadašnjem dijelu poglavlja te mogućnosti koje nude navedena mobilna rješenja, bile su motivacija za razvoj aplikacije za potporu i praćenje tjelovježbe. Uz vrlo jednostavno sučelje, korisnik ima mogućnost održavanja razine aktivnosti na zavidnom nivou. Po uzoru na dosadašnja rješenja, aplikacija će sadržavati postupak registriranja korisničkog profila, prijavu, te uređivanje istog. Zanimljiva značajka aplikacije, koja ju razlikuje od ostalih, jest da su izdvojene vježbe potpuno prilagođene fizičkoj spremi korisnika. Kod ostalih aplikacija, korisniku su vidljive sve moguće vježbe, te on samostalno odabire vježbe prikladne za njega. Detaljan opis modela aplikacije za potporu i praćenje tjelovježbe biti će opisan u poglavlju 3.

3. MODEL APLIKACIJE ZA POTPORU I PRAĆENJE TJELOVJEŽBE

Ovo poglavlje sadrži detaljan opis modela aplikacije za potporu i praćenje tjelovježbe. Navedene su i pojašnjene sve glavne zamisli o radu pojedinog dijela aplikacije, kao i zahtjevi aplikacije prema korisniku.

3.1 Korisnički zahtjevi na mobilnu aplikaciju za potporu i praćenje tjelovježbe

U ovoj aplikaciji od korisnika se zahtjeva da se registrira. Za registriranje su potrebni adresa e-pošte i lozinka. Nakon registracije, unose se parametri potrebni za kalkulaciju korisnikove fizičke spremnosti, te se svi ti podaci spremaju na bazu podataka Firebase. Nakon obrade parametara, posebnim programskim kodom se izdvaja prilagođen skup vježbi iz *JSON* datoteke. Korisnik naknadno može uređivati svoj profil ukoliko je pogrešno unesen neki parametar, a poželjno je ažuriranje podatka o masi, kako bi se mogao pratiti napredak u vježbanju.

3.2 Registriranje korisnika

Pri ulasku u aplikaciju, korisniku će biti vidljiv zaslon za registriranje na kojem će se nalaziti tri polja za unos adrese e-pošte te lozinke koju mora ponoviti dva puta kako bi se izbjegla slučajna pogreška pri njenom unosu. Ispod njih će se nalaziti gumb za registraciju, a pri dnu stranice tekst koji nas klikom odvodi na zaslon prijave. To je predviđeno za korisnike koji su već registrirani i nemaju potrebu činiti to opet. Registriranje se vrši pomoću Firebase Auth-a. Firebase Auth omogućava da se ograniči kreiranje samo jednog računa na jednoj adresi e-pošte, štiteći korisnikovu lozinku. Svaki korisnik nakon registracije dobije jedinstveni ID pomoću kojeg se kasnije može identificirati.

3.3 Prijava korisnika

Zaslon prijave sastoji se od dva polja gdje se unose email adresa i lozinka. Klikom na gumb ispod, ako je utipkana ispravna adresa e-pošte i lozinka, korisnik se prijavljuje na svoj račun te odlazi na zaslon koji ovisi o sljedećem. Ukoliko korisnik još nije unio informacije o svom računu, onda odlazi na zaslon za unos parametara nakon prve prijave. Ako je to već učinio, odlazi na zaslon gdje su prikazani njegovi podaci i popis vježbi koje su mu dodijeljene.

3.4 Unos parametara pri prvoj prijavi

Nakon prijave korisnika, aplikacija provjerava je li korisnik već uspješno unio parametre koje možemo vidjeti prikazane u tablici 3.1, no ukoliko nije, odvodi ga na stranicu za unos parametara. Korisnik unosi svoje osobne podatke kao ime, prezime, spol i starost. No ti podaci nisu toliko bitni za algoritam odabira prikladnih vježbi. Najvažniji parametri koji se unose su visina u

centimetrima i masa u kilogramima. Pomoću tih podataka računa se indeks tjelesne mase (BMI) [15]. On je najvažniji čimbenik pri odabiru vježbi koje bi korisniku bile najbolje za izvođenje. Ako je indeks tjelesne mase u prihvatljivim granicama, pretpostavlja se da je korisnik fizički spremniji za izvođenje nekih vježbi. Također, korisnik iz padajućeg izbornika odabire dnevnu razinu aktivnosti. Ukoliko korisnik većinu dana provodi u kući ili uredu i ne bavi se nikakvih sportom, znači da ima nisku razinu aktivnosti. Ako se bavi sportom, ide u teretanu ili se bavi bilo kojom drugom aktivnošću, odabrat će visoku razinu aktivnosti. Primjer srednje razine aktivnosti bio bi ako korisnik ide na posao pješice ili svakodnevno šeta psa. Profilna fotografija može biti učitana kako bi korisnički račun bio što osobniji. Svi podatci osim fotografije su obvezni za unos. Posljednja dva parametra u tablici nisu unesena od strane korisnika, već se računaju pomoću ostalih parametara.

Tablica 3.1. Prikaz parametara korisnika

Parametar	Tip varijable
Ime	String
Prezime	String
Visina	Double
Težina	Double
Starost	Integer
Spol	String
Razina dnevne aktivnosti	String
<i>Indeks tjelesne mase</i>	Double
<i>Intenzitet</i>	Integer

3.5 Obrada podataka

Najbitniji podatci za određivanje prikladnog skupa vježbi su visina, težina te razina dnevne tjelesne aktivnosti korisnika. Pomoću visine i težine određuje se čovjekov indeks tjelesne mase. Uzimaju se vrijednosti mase u kilogramima te visina u metrima. Budući da je to puno praktičniji pristup, aplikacija unosi visinu u centimetrima, no to bude korigirano u izračunu. Prema [15] izraz za indeks tjelesne mase čovjeka glasi :

$$BMI = \frac{masa (kg)}{visina (m)^2}$$

(3-1)

Kombinacijom indeksa tjelesne mase i razine intenziteta dnevne tjelesne aktivnosti, dolazi se do varijable intenziteta. Intenzitet je ograničen na cijele brojeve jedan, dva i tri. Jedan predstavlja najmanji intenzitet vježbe, dok tri najveći. Korisnik s niskom razinom aktivnosti i normalnim indeksom tjelesne mase imat će intenzitet vrijednosti jedan. S druge strane, osoba s visokom

razinom aktivnosti i indeksom tjelesne mase koji upućuje na pothranjenost, će imati intenzitet vrijednosti tri. Sredina, to jest intenzitet vrijednosti dva, bila bi osoba sa srednjom razinom aktivnosti i indeksom tjelesne mase koji upućuje na prekomjernu tjelesnu masu.

3.6 Dodjela vježbi zasnovana na unesenim podacima

Uneseni podatci ulaze u algoritam koji računa fizičku spremnost korisnika. Pseudo kod algoritma vidljiv je na slici 3.1.

```
funkcija CalculateIntensity (korisnik ) {  
  Kada je (razina aktivnosti) :  
    = NISKA { Kada je (indeks tjelesne mase) :  
      = između 0 i 19 vrati 1  
      = između 19 i 25 vrati 1  
      = između 25 i 30 vrati 1  
      = ostalo vrati 1  
    }  
    = SREDNJA { Kada je (indeks tjelesne mase) :  
      = između 0 i 19 vrati 2  
      = između 19 i 25 vrati 3  
      = između 25 i 30 vrati 2  
      = ostalo vrati 2  
    }  
    = VISOKA { Kada je (indeks tjelesne mase) :  
      = između 0 i 19 vrati 3  
      = između 19 i 25 vrati 3  
      = između 25 i 30 vrati 3  
      = ostalo vrati 2  
    }  
  }  
}
```

Slika 3.1. Pseudo kod algoritma za izračun fizičke spremnosti

Pomoću indeksa tjelesne mase korisnik je svrstan u jednu od četiri kategorije, a to su: pothranjenost, normalna težina, prekomjerna težina i pretilost. Usporedba te kategorije i razine dnevne aktivnosti koju je korisnik unio dovodi do rezultata njegove spremnosti na fizičke aktivnosti. Taj rezultat ima naziv intenzitet. Svaka vježba ima određen intenzitet, a korisniku se dodjeljuju one vježbe čiji se intenzitet poklapa s intenzitetom korisnika. Sve vježbe s njihovim intenzitetom su vidljive u tablici 3.2 [16].

Tablica 3.2. Prikaz svih vježbi i njihovih intenziteta

Naziv vježbe na engleskom	Naziv vježbe na hrvatskom	Intenzitet
Arnold shoulder press	Potisak ramena Arnold	2
Power partials	Horizontalno dizanje ruku	2
Single leg press	Potisak jednom nogom	2
Barbell bench press	Potisak šipke na klupi	1
Barbell full squat	Čučanj sa šipkom	2
Barbell hip thrust	Potisak kuka sa šipkom	2
Finger curls	Pregib prstima	1
Close grip barbell bench press	Uski potisak šipke na klupi	1
Wrist rotation with straight bar	Okretanje zgloba sa ravnom šipkom	1
Incline cable flye	Povlačenje kablova uz nagib	2
Bent over barbell row	Pognuto veslanje šipkom	2
Dips triceps version	Propadanje verzija triceps	2
Weighted pull ups	Zgibovi sa utegom	3
V bar pulldown	Povlačenje V šipke	2
Wide grip lat pulldown	Povlačenje šipke širokog hvata	3
Elevated cable rows	Veslanje kablovima uz povišenje	2
Straight arm pulldown	Povlačenje ravnom rukom	2
Rope jumping	Preskakanje užeta	1
Calf press	Potisak za list	2
Lying leg curls	Ležeći pregib nogu	2
Rowing stationary	Stacionarno veslanje	2
Cross body crunch	Trbušnjaci	2
T bar row with handle	Veslanje sa T ručkom	2
Reverse grip bent over rows	Pognuto veslanje obrnutog hvata	2
Hyperextensions	Leđnjaci na spravi	1
Decline EZ bar triceps extension	EZ šipka uz nagib verzija triceps	2
Reverse grip triceps pushdown	Triceps guranje obrnutog hvata	2
Triceps pushdown rope attachment	Triceps guranje užem	3
Dumbbell triceps extension	Triceps ekstenzija bučicama	2
Kneeling cable triceps extension	Triceps ekstenzija kablovima u čučnju	3
Leverage shrug	Podizanje ramena polugom	2
Concentration curls	Koncentracijski pregib	3
Hammer curls	Čekić pregib	1
Incline hammer curls	Čekić pregib uz nagib	2
Seated dumbbell curls	Sjedeći pregib bučicama	3
Sumo deadlift	Sumo mrtvo dizanje	3
Front squat with two kettlebells	Prednji čučanj sa dvije girje	2
Smith machine calf raise	Stoj na prstima na Smith mašini	2
Barbell walking lunge	Hod sa šipkom	1
Running treadmill	Trčanje na traci	1
Plank	Pozicija skleka na laktovima	1
Bottoms up	Podizanje stražnjice	1
Push ups	Sklekovi	1

3.7 Uređivanje korisničkog profila

Ako je neki od parametara krivo unesen, lako se može izmijeniti. Odlaskom na uređivanje profila, korisnik vidi svoje trenutne podatke koji mogu biti promijenjeni i nakon željenih promjena spremljeni. Ukoliko je korisnik zaboravio svoju lozinku, ona se može resetirati unosom adrese e-pošte koja je korištena pri registriranju profila.

3.8 Ažuriranje podataka o masi korisnika

Kako bi postojao uvid u napredak, podatak o masi može biti ažuriran. Podatak o razlici mase koju je korisnik unio pri prvom unosu parametara i masi koju je unio pri zadnjem ažuriranju, bit će istaknut na profilu. Tako će se uvijek znati napreduje li ili nazaduje korisnik.

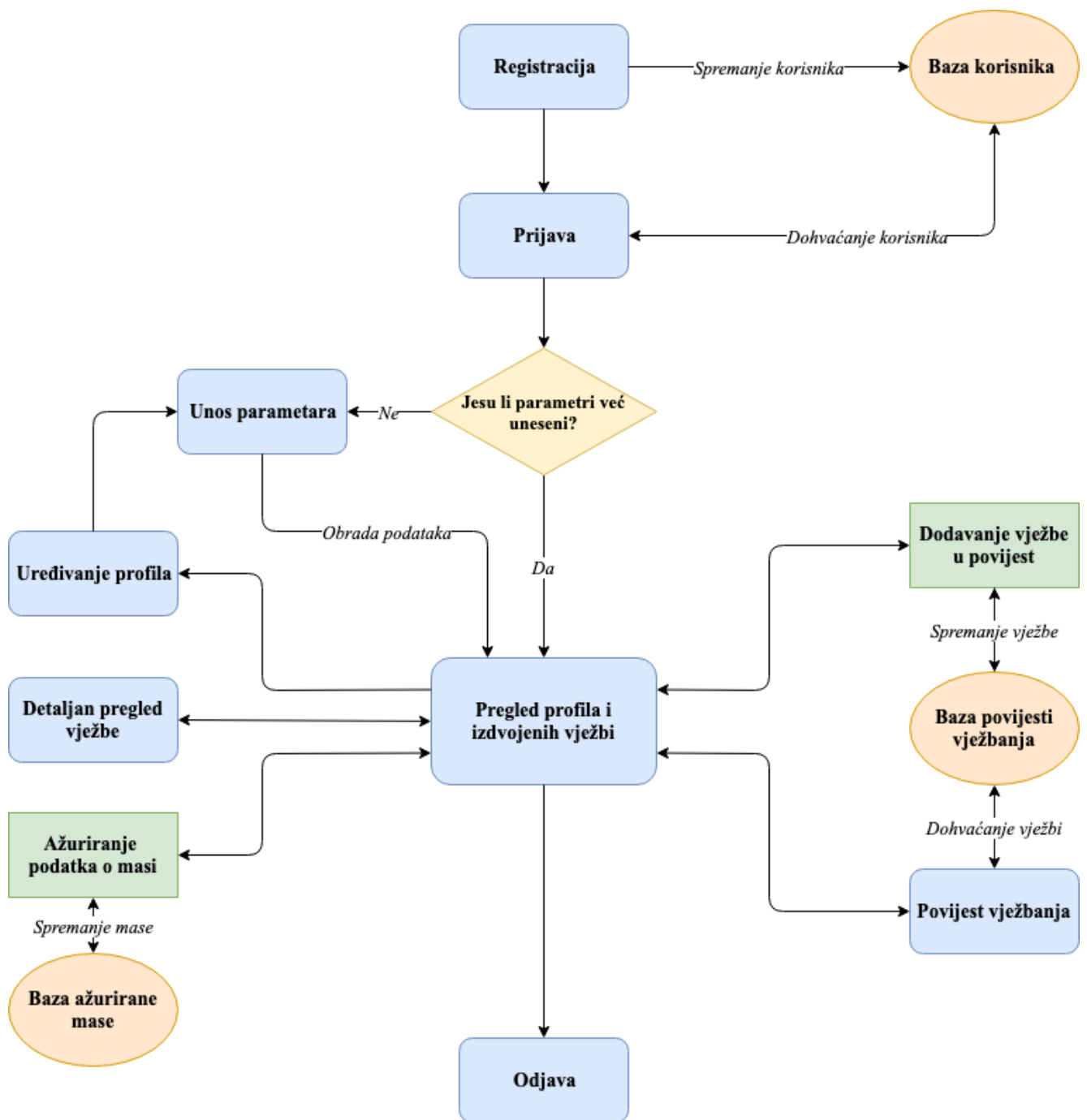
3.9 Detaljan pregled svake pojedine vježbe

Nakon unosa podataka, odlazi se na stranicu gdje postoji pregled korisnikovih osnovnih podataka te popis vježbi koju su mu namijenjene za izvođenje. Ukoliko korisnik ne zna izvesti neku od vježbi ili se do sada s njom nije susreo, klikom na nju ulazi u njen detaljan prikaz. Tamo se nalazi ime vježbe, kategorija mišića koju najviše zahvaća, te detaljan opis izvođenja vježbe. Svaka vježba je popraćena s dvije slike početnog i krajnjeg položaja tijela.

3.10 Odjava korisnika

Aplikacija sadrži mogućnost odjave korisnika. Svi podatci o pojedinom profilu ostaju zaštićeni i spremljeni na poslužitelju. Nakon odjave aplikacija odlazi na zaslon prijave.

Model aplikacije za potporu i praćenje tjelesne aktivnosti prema prethodnim opisima se može vidjeti na slici 3.2.



Slika 3.2. Model aplikacije za potporu i praćenje tjelesne aktivnosti

4. PROGRAMSKO RJEŠENJE APLIKACIJE ZA POTPORU I PRAĆENJE TJELOVJEŽBE

4.1 Opis korištenih alata i tehnologija

Cijela aplikacija za praćenje tjelesne aktivnosti napravljena je u integriranom razvojnom okruženju Android Studio. Korišten je moderan i relativno nov programski jezik Kotlin, kombiniran s bazom podataka i sustavom autentifikacije Firebase. Sva korisnička sučelja napisana su koristeći XML. Kako bi aplikacija bila što čitljivija, modernija, te lakša za održavanje, rađena je po uzoru na Model-Pogled-Prezenter arhitekturu.

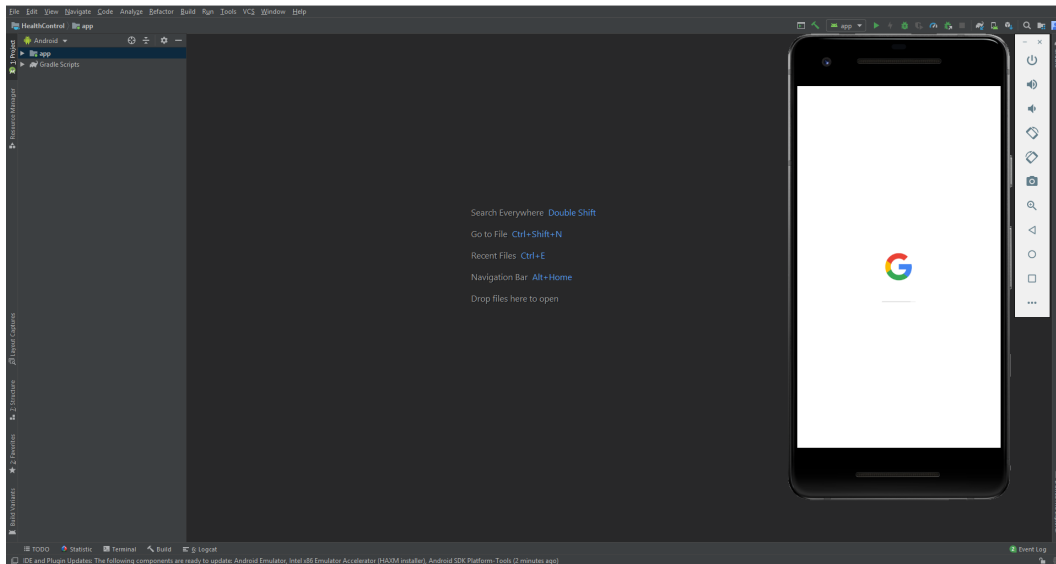
4.1.1 Android Studio

Android Studio [17] je službeno integrirano razvojno okruženje za Android operacijski sustav. Izgrađen je na JetBrains - ovom IntelliJ IDEA programskoj podršci i dizajniran posebno za razvoj Android aplikacija. Može se koristiti na mnoštvu operacijskih sustava kao što su Windows, Linux, MacOS i tako dalje. Kako bi Android Studio mogao raditi u povoljnim uvjetima i između ostalog, biti pokrenut od strane računala, mora zadovoljiti minimalne karakteristike prikazane u tablici 4.1.

Tablica 4.1. Minimalne karakteristike računala za Android Studio

Operacijski sustav	Microsoft Windows 7/8/10 (32-bit ili 64-bit), MacOS X 10.10 ili viši do 10.14 (macOS Mojave), GNOME ili KDE Ubuntu
RAM	Minimum 4GB, preporučeno 8 GB
Mjesto na disku	Minimalno 2GB, preporučeno 4 GB
Minimalna rezolucija	1280 x 800

Android Studio omogućuje pokretanje aplikacija na virtualnom stroju tj. emulatoru koji ima ulogu oponašati stvarni uređaj radi lakšeg testiranja, a može se vidjeti na slici 4.1. Jezik koji se većim dijelom upotrebljava pri stvaranju Android aplikacije je Java, no u posljednje vrijeme, sve veća je uporaba Kotlin. Android Studio ima vrlo razvijen sustav otkrivanja i ispravljanja pogrešaka, što uvelike olakšava samo pisanje i održavanje koda [18]. Jedna od najrazvijenijih karakteristika je automatsko nadopunjavanje koda, gdje program automatski prepoznaje što bi korisnik sljedeće mogao napisati te mu to ponudi u padajućem izborniku.



Slika 4.1. Integrirano razvojno okruženje Android Studio i emulator

Zbog jednostavnosti i rasprostranjenosti ovoga programa, aplikacija za potporu i praćenje tjelovježbe izrađena je baš u njemu.

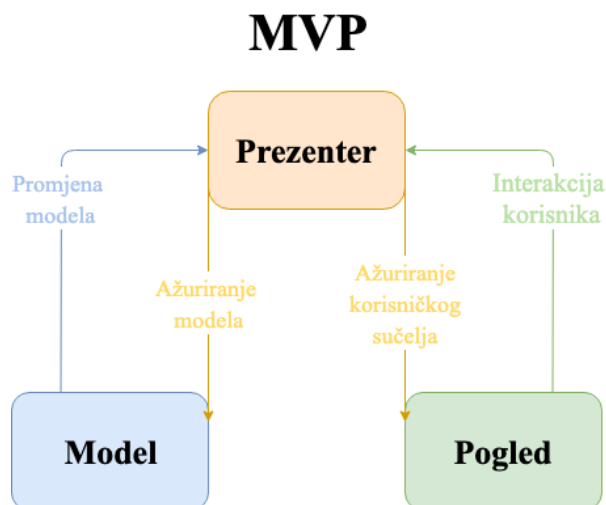
4.1.2 Programski jezik Kotlin

Prema [19], Kotlin je statički programski jezik za razvoj multiplatformskih aplikacija. Razvijen je od strane JetBrains- a te je službeno prihvaćen od strane Google- a 2017. godine. Za razliku od Jave, puno je moderniji, apstraktniji i napredniji programski jezik. Unatoč razlici, Kotlin je ostao interoperabilan s Javom, što znači da se oba jezika mogu kombinirati i koristiti jedan u drugom. Sintaksa Kotlina je oku ugodna i kompaktna, što korisnicima omogućuje pisanje istih komponenata u puno manje redaka, koji su uvelike pregledniji od na primjer Jave. Kako bi aplikacija bila u korak s vremenom, napisana je u gore navedenom programskom jeziku.

4.1.3 Model-Pogled-Prezenter arhitekturni obrazac

Model-Pogled-Prezenter [20] trenutno je jedan od najzastupljenijih arhitekturnih obrazaca u Androidu, zbog svoje jednostavnosti, prilagodljivosti i brzine postavljanja. Aplikacija se odvaja na tri sloja, model sloj, sloj pogleda i sloj poslovne logike tj. prezentacijski sloj čiji prikaz možete vidjeti na slici 4.2. Sloj model je najudaljeniji od korisnika. Predstavlja entitete koji nam pomažu pri grupiranju podataka u neke nama razumljive cjeline. Sloj prezentacije bavi se povezivanjem slojeva modela i pogleda. On povezuje korisnikovu interakciju s logikom koja se događa nakon iste. Nakon odrađene logike prosljeđuje upute sloju pogleda radi prilagodbe njegova izgleda. Model pogleda služi isključivo za prikaz sučelja i podataka na sučelju. On ovisi samo o platformi i njezinim pojedinostima te ne zna ništa o logici same aplikacije i dohvaćanju podataka. Ovaj arhitekturni obrazac omogućava brzu nadogradnju, bolju čitljivost, te ono najvažnije, vrlo lako

testiranje aplikacije. Zbog toga se većina kompanija koje razvijaju Android aplikacije odlučuje baš za ovaj obrazac.

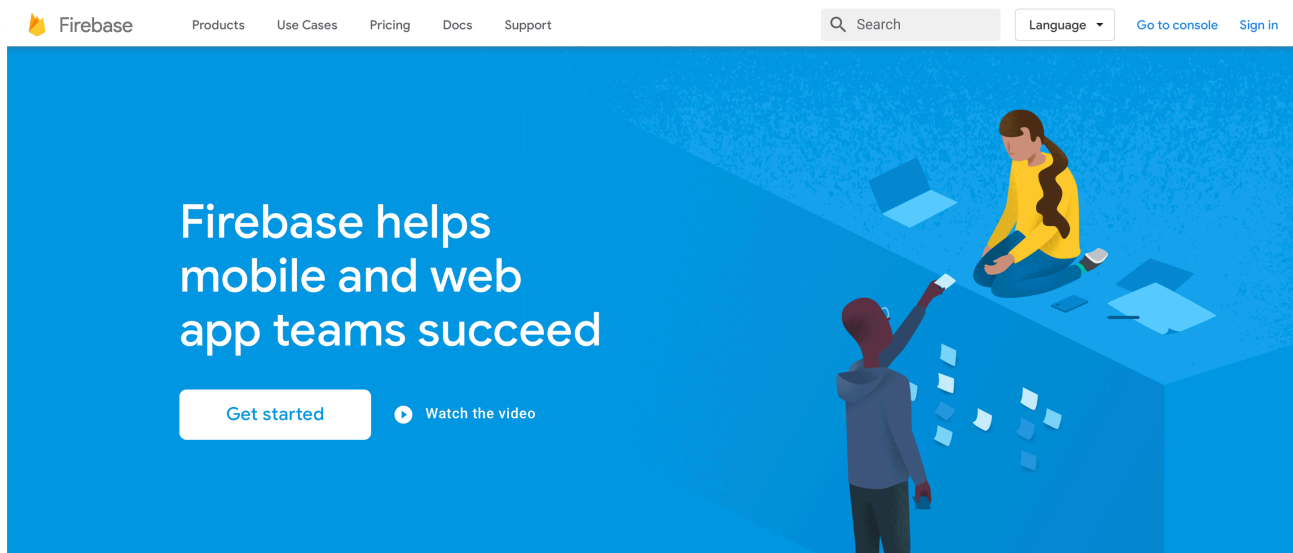


Slika 4.2. Model-Pogled-Prezenter arhitekturni obrazac

Kako bi aplikaciju bilo što jednostavnije i preglednije napisati te kasnije poboljšavati i nadograđivati, pri izradi je korišten Model-Pogled-Prezenter arhitekturni obrazac. Također, korištenje tog obrasca aplikaciju čini konkurentnom na tržištu zbog njene stabilnosti.

4.1.4 Firebase

Firebase [21] je razvojna platforma za mobilni i web razvoj, prihvaćena od Google-a. Sadrži čak 18 proizvoda koji se koriste od strane preko 1.5 milijuna aplikacija u svijetu. Najpopularniji proizvod je Firebase Auth koji se koristi za registraciju korisničkih računa. Postoji mnoštvo mogućnosti registracije, no najčešće korištena je registriranje pomoću adrese e-pošte i lozinke. Da bi se uz stvorenog korisnika spremili željeni podatci koji se povezujemo s njim, implementira se baza podataka RealTime. Ona se vrlo jednostavno koristi te omogućava praćenje promjena na bazi u stvarnom vremenu. Firebase omogućava i spremanje dokumenata kao što su fotografije i video dokumenti što je jako korisno pri izradi aplikacija. Spremanje i dohvaćanje podataka odvija se iznimno brzo, što pogoduje izradi aplikacija koje često i u velikim količinama moraju komunicirati s bazom podataka. Izgled početne stranice sustava baze podataka Firebase može se vidjeti na slici 4.3.



Slika 4.3. Početna stranica sustava baze podataka Firebase

4.1.5 XML

EXtensible Markup Language [22] jezik je za označavanje podataka, koji sadrži određena pravila prevođenja instrukcija u format koji je lako razumljiv i čovjeku i računalu. Sadržaj je potrebno uokviriti oznakama koje ga opisuju te su lake za shvatiti. Neke od oznaka možete vidjeti na slici 4.4. Najkorištenije pozicioniranje pri izradi mobilnih aplikacija je pozicioniranje preko ograničenja (eng. constraints). Pomoću njih se najlakše može izraditi korisničko sučelje koje će se moći prilagoditi na velik broj mobilnih uređaja različitih dimenzija.

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:tools="http://schemas.android.com/tools"
5     xmlns:app="http://schemas.android.com/apk/res-auto"
6     android:layout_width="match_parent"
7     android:layout_height="match_parent"
8     tools:context=".ui.main.MainActivity">
9
10     <FrameLayout
11         android:id="@+id/fragmentContainer"
12         android:layout_width="match_parent"
13         android:layout_height="match_parent"
14     />
15 </androidx.constraintlayout.widget.ConstraintLayout>
```

Slika 4.4. Primjer XML dokumenta

XML je u ovoj aplikaciji korišten u svrhu izrade izgleda korisničkog sučelja. Za svaki dio sučelja izrađen je zaseban dokument nastavka .xml, koji definira izgled svakog teksta, polja za unos, slike, gumba i tako dalje.

4.2 Prikaz programskog rješenja po komponentama

U ovom poglavlju prikazuje se svaka komponenta koja čini ovu mobilnu aplikaciju. Uz svaku cjelinu bit će prikazan i kod koji je sačinjava.

4.2.1 Registriranje i prijava korisnika

Za registriranje korisnika koristi se jednostavna *metoda* po imenu „createUserWithEmailAndPassword“, koja se može vidjeti na slici 4.5. Kao što samo ime kaže, metodi se predaje adresa e-pošte i lozinka pomoću kojih se kreira korisnički račun s jedinstvenim ID-em na Firebase- u. Prvo što aplikacija provjerava je jesu li sva polja za unos popunjena. Ukoliko nisu prikazuje se „toast“ poruka „Check your inputs“. Ako su popunjena sva polja, slijedi druga provjera, a to je provjera unesenih lozinki. Lozinke u oba polja moraju se podudarati kako bi bili sigurni da korisnik svjesno unosi željenu lozinku. Ukoliko se lozinke razlikuju, korisniku iskače poruka „Passwords are not matching“. Treća provjera je provjera od strane Firebase sustava. On provjerava valjanost unesene adrese e-pošte. Ako je adresa valjana sustav će uspješno registrirati novi profil.

```
private fun handleRegister() {
    firebase.createUserWithEmailAndPassword(email, password).addOnCompleteListener { task : Task<AuthResult!> ->
        if(task.isSuccessful) {
            view.onRegisterSuccessful()
        }
        else view.onRegisterFailed()
    }
}
```

Slika 4.5. Primjer *metode* createUserWithEmailAndPassword

Za prijavu se koristi *metoda* „signInWithEmailAndPassword“ kojoj se također predaju adresa e-pošte i lozinka te ako postoji korisnički račun s istima slijedi provjera je li korisnik već unosio dodatne parametre. Za tu provjeru postoji varijabla isInfoSet tipa Boolean, te *metoda* „isInfoSet“ koju se može vidjeti na slici 4.6. Ta metoda pri prijavi provjerava stanje varijable, te ukoliko je varijabla *false*, to znači da korisnik nije popunio dodatne parametre i odlazi na zaslon za popunjavanje dodatnih parametara. Ako je varijabla *true*, odlazi direktno na zaslon pregleda profila.

```

private fun checkIsInfoSet() {
    database.addListenerForSingleValueEvent(object: ValueEventListener{
        override fun onCancelled(p0: DatabaseError) {
            view.onLoginFailed()
        }

        override fun onDataChange(p0: DataSnapshot) {
            val userID = firebase.currentUser?.uid
            if (userID != null && p0.child(userID).child( path: "infoSet").value != null ) {
                isInfoSet = p0.child(userID).child( path: "infoSet").value as Boolean
            }
            view.onLoginSuccessful(isInfoSet)
        }
    })
}
}

```

Slika 4.6. Programski kod za provjeru popunjenosti dodatnih parametara

Firestore također nudi jednostavnu metodu „sendPasswordResetMail“, ukoliko korisnik zaboravi svoju lozinku. Klikom na tekst „Forgot your password“ otvara se dijalog čiji se XML kod može vidjeti na slici 4.7. U kodu su vidljiva četiri elementa. Prvi element je vrsta rasporeda koji se koristi u slaganju ostalih elemenata koji se nalaze unutar njega. Ostala tri elementa su tekst, polje za unos teksta, te gumb. Svaki element ima atribute koji opisuju njegov smještaj u odnosu na zaslon, te u odnosu na druge elemente unutar zaslona. Postoje još atributi boje i veličine teksta, boje pozadine te mnogi drugi. U dijalog korisnik upisuje svoju adresu e-pošte te klikom na gumb šalje e-poštu pomoću koje resetira svoju lozinku. Izgled zaslona može se vidjeti u poglavlju 5 na slici 5.3.

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="250dp" xmlns:app="http://schemas.android.com/apk/res-auto"
    tools:context=".ui.login.fragments.ForgotPasswordFragment"
    android:padding="30dp">

    <TextView
        android:id="@+id/tvForgotPasswordTitle"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintBottom_toTopOf="@id/etEmail"
        android:layout_marginBottom="30dp"
        android:text="Enter your email : "
        android:textSize="30dp"
        android:textStyle="bold"
        android:textColor="@color/black"/>

    <EditText
        android:id="@+id/etEmail"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintBottom_toBottomOf="parent"
        android:layout_marginTop="30dp"
        android:textColor="@color/black"
        android:hint="Email"
        android:inputType="textEmailAddress"
        />

    <Button
        android:id="@+id/btnResetPassword"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:layout_constraintTop_toBottomOf="@id/etEmail"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        android:background="@drawable/button"
        android:padding="10dp"
        android:text="Reset your password"/>

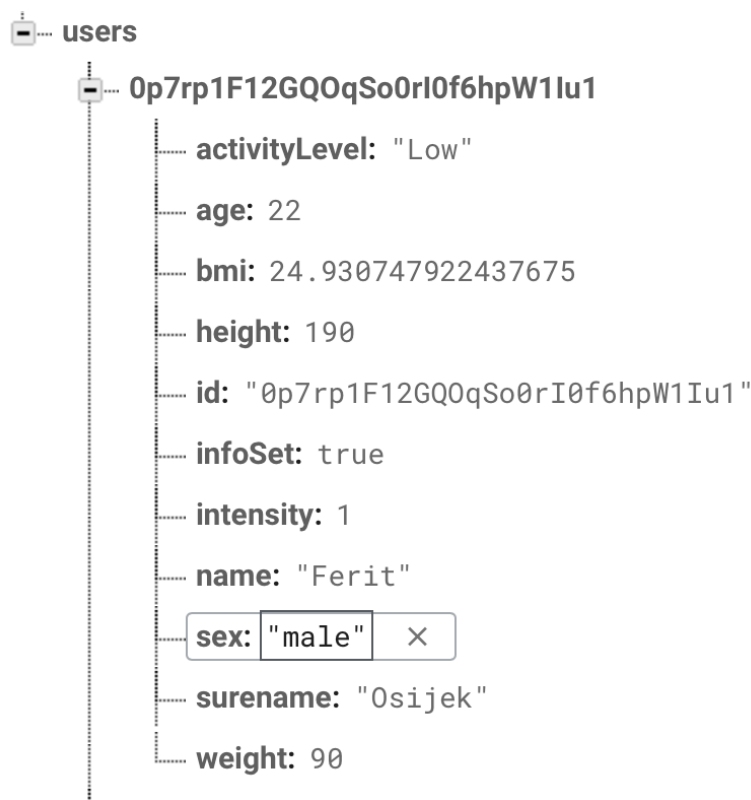
</androidx.constraintlayout.widget.ConstraintLayout>

```

Slika 4.7. XML kod za izgled dijaloga za resetiranje lozinke

4.2.2 Rad s bazom podataka Firebase

Kako bi se podatci o korisniku mogli negdje spremiti i kasnije biti dohvaćani, koristi se baza podataka Firebase RealTime. Svaki korisnik predstavlja jedan čvor koji je zapravo njegov ID. Iza tog ID- a spremaju se podatci koje korisnik unosi nakon prve prijave. Radi jednostavnosti, ne sprema se jedan po jedan podataka, već se kreira *objekt* User te se on sprema u cijelosti. Izgled jednog korisnika spremljenog u bazi, skupa sa njegovim parametrima, vidljiv je na slici 4.7.



Slika 4.7. Izgled korisnika spremljenog u bazu podataka

Kreiranje čvora radi se *metodom* „child“, a spremanje *objekta* s *metodom* „setValues“. Dohvaćanje podataka može se izvršiti na više načina. Podatak se može dohvatiti samo jednom, a može se dohvatiti i na svaku promjenu u bazi. To omogućuju metode „addListenerForSingleValueEvent“, koju se može vidjeti na slici 4.8, i „addValueEventListener“. Ukratko, ta metoda omogućuje jednokratni pregled i dohvaćanje svih podataka spremljenih u bazu.

```

override fun getUserData() {
    var user: User? = null
    database.addListenerForSingleValueEvent(object: ValueEventListener{
        override fun onCancelled(p0: DatabaseError) {
            view.onUserFailed()
        }
        override fun onDataChange(p0: DataSnapshot) {
            val userId = firebase.currentUser?.uid
            p0.children.forEach { it: DataSnapshot!
                if(it.child( path: "id").value == userId){
                    user = it.getValue(User::class.java)}
            }
            user?.let { view.onUserSuccessful(it) }
        }
    })
}

```

Slika 4.8. Primjer metode `addListenerForSingleValueEvent`

Kako bi se spremila profilna fotografija, kombiniraju se baze podataka RealTime i Storage. Baza podataka Storage ima mogućnost spremanja fotografija i video zapisa. Nakon što korisnik odabere fotografiju sa svog uređaja ili ju direktno fotografira u tom trenutku, ona se pohranjuje u bazu podataka Storage. Prvo se provjerava postoji li prijavljen korisnik. Ukoliko postoji, fotografija se sprema pod čvorom ID-a prijavljenog korisnika. Ako se fotografija uspješno spremi, aplikacija pronalazi destinaciju fotografije, te nju sprema u varijablu imena „imagePath“. Ta varijabla se kasnije sprema u bazu podataka RealTime, kako bi fotografija mogla biti pronađena i prikazana. Nakon pronalaska destinacije, poziva se *metoda* „onImageSuccess“ i predaje joj se odabrana fotografija, te se ista prikazuje na mjestu za profilnu fotografiju. Ako je došlo do pogreške pri spremanju fotografije, poziva se *metoda* „onImageFailed“ koja obavještava korisnika o neuspjelom spremanju fotografije prikladnom porukom. Na taj način su ove dvije baze povezane. Postupak spremanja fotografije prebačen u Kotlin kod prikazan je na slici 4.9.

```

override fun onSetImage(uri: Uri) {
    if (userID != null) {
        storage.child(userID).putFile(uri).addOnCompleteListener { task ->
            if(task.isSuccessful){
                storage.child(userID).child(uri.toString()).downloadUrl.addOnSuccessListener { it: Uri!
                    imagePath = it.toString()
                }
                view.onImageSuccess(uri)
            }
            else {
                view.onImageFailed()
            }
        }
    }
}

```

Slika 4.9. Prikaz postupka spremanja fotografije u Storage i dohvaćanje njenog odredišta

4.2.3 Postupak unosa parametara

Osim osnovnih podataka koji su se unosili pri registriranju, unose se i parametre koji će upotpuniti korisnikov profil, te mu pružiti sve mogućnosti koje ova aplikacija nudi. Popunjavaju se polja ime,

prezime, visina u centimetrima, masa u kilogramima, starost, spol i razina dnevne tjelesne aktivnosti. Zatim se klikom na gumb spremaju podatci. No prije toga metodom „inputsNotEmpty“, provjerava se jesu li sva polja popunjena. Ukoliko nisu, poziva se metoda „onInputsEmpty“, koja ispisuje prikladnu poruku korisniku. Ako su svi unosi ispravni, mogu se pohraniti u bazu podataka. Aplikacija prvo odrađuje provjeru postoji li prijavljen korisnik. Ukoliko postoji, šalje se zahtjev za spremanjem podataka. Podatci se spremaju na bazu podataka, a podatke različitih korisnika razlikujemo po jedinstvenom ID-u koji svaki korisnik dobije pri registriranju. Radi lakšeg baratanja sa spremanjem korisničkih podataka, prije slanja, podatci se upakiraju u *objekt klase* User. Budući da se u bazu mogu spremiti samo podaci tipa String, mora se pobrinuti za preoblikovanje spola koji je u aplikaciji tipa kružni gumb i razine dnevne aktivnosti koja je tipa padajućeg izbornika. Za kružni gumb provjerava se ID, te ako ID odgovara kružnom gumbu koji označava muškarca, onda se bazi predaje riječ „male“. U suprotnom predaje se riječ „female“. Na isti način funkcioniše i pretvorba padajućeg izbornika u riječ. Ukoliko spremanje korisnika u bazu bude uspješno izvršeno, poziva se *metoda* „onUserSaved“, koja odvodi korisnika na sljedeći zaslon. Postoji mogućnost da pri spremanju dođe do pogreške, bilo od strane poslužitelja ili iz nekog drugog razloga, tada se poziva *metoda* „onUserSavedFailed“ koja obavještava korisnika da pohrana nije uspjela. Kod zadužen za pohranu parametara može se pogledati na slici 4.10.

```

override fun onSaveUserInfo(
    name: String,
    surname: String,
    height: Double,
    weight: Double,
    radioGroup: RadioGroup,
    activityLevel: String,
    age: Int
) {
    if(inputsNotEmpty(name,surname,height,weight,age)){
        val sex = if(radioGroup.checkedRadioButtonId == R.id.radioMale) "male" else "female"
        val userData = User(firebase.currentUser?.uid,name,surname,height, weight,sex,imagePath,age,activityLevel,isInfoSet = true)
        if (userID != null) {
            database.child(userID).setValue(userData).addOnCompleteListener { task->
                if(task.isSuccessful) view.onUserSaved()
                else view.onUserSavedFailed()
            }
        }
        else{
            view.onUserSavedFailed()
        }
    }
    else
    {
        view.onInputsEmpty()
    }
}

```

Slika 4.10. Programski kod za spremanje podataka korisnika

4.2.4 Obrada unesenih podataka

Kako bi logika obrade unesenih podataka bila odvojena od ostatka koda, stvoren je *objekt* imena IntensityCalculator. U njemu postoji jedna *metoda* koja služi izračunu prikladnog intenziteta vježbe koji korisnik može podnijeti. Toj *metodi* se predaje *objekt klase* User, te se iz njega koristi dva parametra. Indeks tjelesne mase (BMI) i razina dnevne aktivnosti služe kako bi se što bolje

mogla odrediti varijabla intenziteta. *Metoda* koristi ugrađenu *funkciju* „when“ te pomoću nje kreira dvanaest mogućih ishoda. Prvo se provjerava razina dnevne aktivnosti, te u svakom od tih slučajeva slijedi ponovna provjera indeksa tjelesne mase. Programski kod za izračun intenziteta vidljiv je na slici 4.11.

```
object IntensityCalculator{
  fun calculateIntensity(user: User): Int {
    when(user.activityLevel){
      "Low" -> {return when(user.bmi?.toInt()){
        in 0..19 -> 1
        in 19..25 -> 1
        in 25..30 -> 1
        else -> 1
      }}
      "Medium" -> {return when(user.bmi?.toInt()){
        in 0..19 -> 2
        in 19..25 -> 3
        in 25..30 -> 2
        else -> 2
      }}
      "High" -> {return when(user.bmi?.toInt()){
        in 0..19 -> 3
        in 19..25 -> 3
        in 25..30 -> 3
        else -> 2
      }}
    }
  }
  return 1
}
```

Slika 4.11. Programski kod za izračun intenziteta

4.2.5 Dohvaćanje prikladnog skupa vježbi

Nakon obrade svih podataka, iz mnoštva vježbi koje postoje u bazi [21], potrebno je izdvojiti one koje odgovaraju trenutnom korisniku. Vježbe se nalaze u JSON dokumentu na poslužitelju. Jedna vježba u JSON formatu prikazana je na slici 4.12.

```
{
  "id":44,
  "description":"Lie on the floor face down and place your hands about 36 inches apart while holding your torso up at arms length. Next, lower yourself downward until your chest almost touches the floor as you inhale. Now breathe out and press your upper body back up to the starting position while squeezing your chest. After a brief pause at the top contracted position, you can begin to lower yourself downward again for as many repetitions as needed.",
  "name":"Push Ups",
  "creation_date":"2017-02-27",
  "muscles":"Chest",
  "muscles_secondary":" ",
  "image":"https://artifacts.bbcomcdn.com/@bbcom/exercises-app/2.0.0/img/guide-1.gif",
  "image2":"https://www.bodybuilding.com/exercises/exerciseImages/sequences/70/Male/l/70_2.jpg",
  "image3":"https://www.bodybuilding.com/exercises/exerciseImages/sequences/70/Male/l/70_1.jpg",
  "category":"push",
  "intensity":"1"
},
```

Slika 4.12. Izgled vježbe u JSON formatu

Svaka od vježbi ima svoj intenzitet. Intenzitet varira između jedan, dva i tri gdje je jedan najmanje intenzivna vježba, a tri najviše intenzivna. Manje intenzivne vježbe su one koje može izvoditi svaki početnik bez imalo iskustva. Najintenzivnije vježbe zahtijevaju puno veću fizičku spremu te iskustvo u vježbanju. Aplikacija šalje zahtjev za dohvaćanje svih vježbi iz JSON dokumenta.

Ukoliko odgovor na zahtjev bude uspješan, to jest, kod odgovora bude „200“, to znači da su vježbe uspješno dohvaćene u obliku liste *objekata* tipa „Exercise“. No kako bi korisniku bile vidljive samo vježbe koje odgovaraju njegovom intenzitetu, dobivena lista prolazi kroz filter koji odabire samo vježbe čiji se intenzitet poklapa sa intenzitetom korisnika. Nakon toga, lista je prosljeđena *metodi* „getExerciseSuccessful“, te je prikazana na zaslonu. Ako odgovor nije uspješan, tada se poziva *metoda* „getExerciseFailed“ ili „getExerciseError“, ovisno o vrsti pogreške koja se dogodila tijekom zahtjeva. Kod za dohvaćanje vježbi je vidljiv na slici 4.13. Ukoliko korisnik ažurira masu, postoji mogućnost da se, razina intenziteta na koju je korisnik spreman, promjeni. Tada se automatski sve vježbe ponovno filtriraju kako bi se odabrale one pravog intenziteta.

```
private fun getExercisesCallback(): Callback<ExercisesResponse> = object: Callback<ExercisesResponse> {
    override fun onFailure(call: Call<ExercisesResponse>, t: Throwable) {
        view.getExercisesFailed()
    }

    override fun onResponse(call: Call<ExercisesResponse>, response: Response<ExercisesResponse>) {
        if (response.isSuccessful) {
            when (response.code()) {
                200 -> response.body()?.results?.let { it: List<Exercise> } {
                    var list = it.filter { it.intensity == intensity }
                    view.getExercisesSuccessful(list) }
            }
        } else {
            view.getExercisesError(response.message())
        }
    }
}
```

Slika 4.13. Programski kod za dohvaćanje vježbi te njihovo filtriranje

4.2.6 Ažuriranje podataka o masi korisnika

Dođe li do potrebe da korisnik ažurira svoju masu zbog promjene iste, to se može učiniti pritiskom na „Update weight“ koji se nalazi u padajućem izborniku iza tri točkice koje su postavljene u gornjem desnom dijelu zaslona. Nakon klika otvara se dijalog u koji korisnik unosi svoju ažuriranu masu. Kod dijaloga napisanog u XML kodu vidljiv je na slici 4.14, a izgled sučelja moguće je pogledati u sljedećem poglavlju na slici 5.6.

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="250dp"
    android:padding="30dp">

    <TextView
        android:id="@+id/tvWeightUpdateTitle"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        android:textColor="@color/black"
        android:textSize="30dp"
        android:text="Enter your current weight: "/>

    <EditText
        android:id="@+id/etWeight"
        android:layout_width="100dp"
        android:layout_height="50dp"
        app:layout_constraintTop_toBottomOf="@id/tvWeightUpdateTitle"
        app:layout_constraintBottom_toTopOf="@id/btnUpdate"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        android:inputType="numberDecimal"
        android:textColor="@color/black"
        android:textColorLink="@color/colorPrimary"
        android:outlineAmbientShadowColor="@color/colorPrimary"
        android:outlineSpotShadowColor="@color/colorPrimary"/>

    <Button
        android:id="@+id/btnUpdate"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:layout_constraintTop_toBottomOf="@id/etWeight"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        android:background="@drawable/button"
        android:padding="10dp"
        android:text="UPDATE"/>

</androidx.constraintlayout.widget.ConstraintLayout>

```

Slika 4.14. XML kod za izgled dijaloga za ažuriranje mase

Slijedi provjera postoji li uopće uneseni podatak. Ukoliko ne postoji korisnik je obavješten porukom na zaslonu. U suprotnom, masa se prosljeđuje kroz *metodu* „onUpdateWeight“ te se ondje prvo upakira u *objekt* „Weight“. Ako postoji prijavljen korisnik, *objekt* se sprema na novo mjesto u bazi, kako početna masa nikada ne bi bila izbrisana. Nakon uspješnog spremanja, poziva se *metoda* „onWeightUpdatedSuccessfully“ koja zatvara dijalog te ažurira prikazanu razliku u masi. Tako uvijek postoji uvid u početnu i trenutnu masu korisnika te on može biti obaviješten o kretanju njegove mase. Ako korisnik do sada nije ažurirao podatak o svojoj masi, njegova trenutna masa smatra se jednakom masi koju je naveo pri registriranju, te je razlika u masi jednaka nula. Kod ovoga rješenja vidljiv je na slici 4.15.

```

override fun onUpdateWeight(weight: Double) {
    val id = firebase.currentUser?.uid
    val weightData = Weight(weight)
    if(id != null)
    {
        database.child(id).setValue(weightData)
        .addOnCompleteListener { task ->
            if(task.isSuccessful)
                view.onWeightUpdatedSuccessfully()
        }
    }
}
}

```

Slika 4.15. Programski kod za spremanja ažurirane mase korisnika

4.2.7 Detaljan prikaz pojedine vježbe

Klikom na bilo koju vježbu otvara se njen detaljan prikaz uz slikovni i tekstualni opis čiji XML kod je vidljiv na slici 4.17. Kliknuta vježba zajedno sa spolom trenutno prijavljenog korisnika se kroz *intent* šalje u novi *activity*. Tamo se prvo na temelju spola postavlja boja pozadine zaslona *metodom* „*setBackgroundColor*“. Ukoliko se radi o muškom korisniku, postavlja se boja „male“, dok se u suprotnom postavlja boja „female“. Nakon toga dohvaća se poslana vježba pomoću *metode* „*getParcelableExtra*“. Uz pomoć dodatne biblioteke zvane *Picasso*, postavljaju se dvije slike izvođenja vježbe. Također, postavlja se tekst imena, glavnog mišića i opisa vježbe. Kod postavljanja izgleda zaslona detaljnog prikaza vidljiv je na slici 4.16, dok je izgled zaslona vidljiv na slici 5.7.

```

private fun setBackgroundColor() {
    val sex = intent.getSerializableExtra( name: "sex")
    val view : View = this.window.decorView
    if(sex == "male")
    {
        view.setBackgroundColor(ContextCompat.getColor( context: this,R.color.male))
    }
    else{
        view.setBackgroundColor(ContextCompat.getColor( context: this,R.color.female))
    }
}

private fun setText() {
    val exercise = intent.getParcelableExtra( name: "exercise") as Exercise
    Picasso.with( context: this).load(exercise.image2).fit().centerCrop().into(ivFirst)
    Picasso.with( context: this).load(exercise.image3).fit().centerCrop().into(ivSecond)
    tvDetailedTitle.text = exercise.name
    tvDetailedMuscle.text = exercise.muscles
    tvDetailedDescription.text = exercise.description
}
}

```

Slika 4.16. Programski kod za detaljan prikaz pojedine vježbe

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".ui.details.DetailsActivity"
    android:padding="10dp">
    <ImageView
        android:id="@+id/ivFirst"
        android:layout_width="0dp"
        android:layout_height="180dp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toStartOf="@id/ivSecond"
        app:layout_constraintTop_toTopOf="parent"
        android:layout_margin="5dp"/>
    <ImageView
        android:id="@+id/ivSecond"
        android:layout_width="0dp"
        android:layout_height="180dp"
        app:layout_constraintStart_toEndOf="@id/ivFirst"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        android:layout_margin="5dp"/>
    <TextView
        android:id="@+id/tvDetailedTitle"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@id/ivFirst"
        android:textSize="25sp"
        android:textStyle="bold"
        android:maxLines="2"/>
    <TextView
        android:id="@+id/tvDetailedMuscle"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@id/tvDetailedTitle"
        android:textSize="20sp"/>
    <TextView
        android:id="@+id/tvDetailedDescription"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@id/tvDetailedMuscle"
        app:layout_constraintBottom_toBottomOf="parent"
        android:textSize="15sp"/>
</androidx.constraintlayout.widget.ConstraintLayout>

```

Slika 4.17. XML kod za izgled sučelja detaljnog prikaza pojedine vježbe

4.2.8 Dodavanje odradene vježbe u povijest

Klikom na znak plusa pored vježbe, otvara se dijalog u kojem korisnik upisuje broj serija i ponavljanja. Ta dva broja, skupa sa *objektom* vježbe se prosljeđuju u *metodu* „onExerciseAddClicked“ koja je vidljiva na slici 4.18. Ona kreira novi *objekt* „DoneExercise“ koji sadrži ime vježbe, te broj serija i ponavljanja. Datum se formatira u obliku mjesec, dan, godina, te je mjesec napisan u tekstualnom obliku skraćenom na tri slova. Npr., datum 07.10.2019. bit će napisan u obliku Oct, 10.2019. Slijedi provjera postoji li prijavljen korisnik, te ukoliko postoji, vježba se sprema pod danim datumom. Kako bi svaka vježba imala poseban čvor prepoznavanja, dodaje se čvor točnog vremena u milisekundama, trenutka spremanja vježbe. Tako nikada dvije vježbe neće biti spremljene u isti čvor. Nakon uspješnog spremanja vježbe poziva se funkcija „onExerciseAddSuccessful“ koja zatvara dijalog i obavještava korisnika da je vježba spremljena.

Ako dođe do pogreške poziva se funkcija „onExerciseAddFailed“ koja obavještava korisnika da spremanje vježbe nije uspjelo.

```
override fun onExerciseAddClicked(exercise: Exercise, series: Int, repetition: Int) {
    val id = firebase.currentUser?.uid
    val doneExercise = DoneExercise(exercise.name, series, repetition)
    val dateTime = getDateInstance()
    val date = dateTime.format(Date())
    if(id!= null){
        database.child(id).child(date)
            .child(System.currentTimeMillis().toString())
            .setValue(doneExercise).addOnCompleteListener { task ->
                if (task.isSuccessful){
                    view.onExerciseAddSuccessful()
                }
                else{
                    view.onExerciseAddFailed()
                }
            }
    }
}
```

Slika 4.18. Programski kod za dodavanje vježbe u povijest

4.2.9 Pregled povijesti odrađenih vježbi

Kako bi pregledao odrađene vježbe, korisnik odabire datum koji želi pregledati. Nakon odabira datuma, on se prosljeđuje *metodi* „onGetHistory“ vidljivoj na slici 4.19. U njoj se stvara prazna lista *objekata* tipa „DoneExercise“ što predstavlja odrađene vježbe. Ukoliko postoji prijavljen korisnik šalje se zahtjev na bazu podataka. Prvo je odrađena provjera postoji li čvor u bazi koji odgovara traženom datumu. Ukoliko postoji, svako dijete (eng. child) tog čvora, formatira se u *objekt* „DoneExercise“ te ako formatiranje prođe uspješno, dodaje u listu odrađenih vježbi. Nakon što su sva djeca dodana u listu, a lista nije prazna, ona se predaje *metodi* „onGetHistorySuccessful“ koja ju prikazuje na zaslonu. Ukoliko ne postoji nijedna vježba spremljena uz dani datum i lista ostane prazna, poziva se metoda „onHistoryEmpty“ te se na zaslonu ispisuje prikladna poruka objašnjenja da je lista prazna. Ako korisnik nikada nije spremio niti jednu vježbu na dani datum, čvor s tim datumom neće ni postojati, pa će se pozvati *metoda* „onGetHistoryFailed“, koja ispisuje poruku vidljivu korisniku.

```

override fun onGetHistory(date: String) {
    val id = firebase.currentUser?.uid
    val list = mutableListOf<DoneExercise>()
    if(id != null){
        database.addListenerForSingleValueEvent(object: ValueEventListener{
            override fun onCancelled(p0: DatabaseError) {
            }
            override fun onDataChange(p0: DataSnapshot) {
                if(p0.child(id).child(date).exists()) {
                    p0.child(id).child(date).children.forEach { it: DataSnapshot! }
                    val exercise = it.getValue(DoneExercise::class.java)
                    if (exercise != null) {
                        list.add(exercise)
                    }
                }
                if(list.isNotEmpty()){
                    view.onGetHistorySuccessful(list)
                }
                if(list.isEmpty()){
                    view.onHistoryEmpty()
                }
            }else
            {
                view.onGetHistoryFailed()
            }
        })
    }
}

```

Slika 4.19. Programski kod za dohvaćanje liste odrađenih vježbi

5. PRIKAZ RADA APLIKACIJE I ANALIZE REZULTATA

U ovom poglavlju prikazat će se korištenje aplikacije, svi koraci koje korisnik mora proći, te sve funkcionalnosti koje ova aplikacija nudi.

5.1 Prikaz ostvarenih funkcionalnosti aplikacije

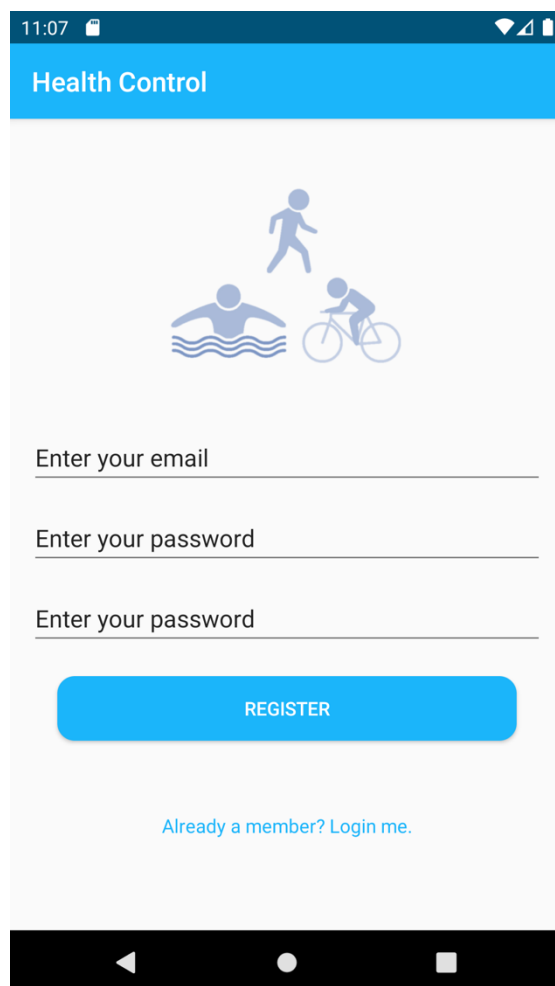
Korisnik, naravno, najprije mora instalirati aplikaciju. Nakon instalacije ima mogućnost registriranja novog profila. Ukoliko je uspješno registriran novi profil, aplikacija odlazi na zaslon prijave. Koristeći iste podatke kao i kod registracije, korisnik je prijavljen u sustav te nakon prve prijave ispunjava obrazac s osobnim podacima. Unosi se ime, prezime, visina, masa, starost, te razina dnevne aktivnosti. Po želji, korisnik ima mogućnost učitavanja svoje profilne fotografije, no taj dio nije obavezan, za razliku od ostatka. Nakon unosa svih parametara, korisnik se odvodi na zaslon njegova profila. Na njemu se vidi njegovo ime i prezime, starost, razina tjelesne aktivnosti, profilna fotografija i trenutna masa. Klikom na trenutnu masu, korisnik može istu ažurirati ukoliko je došlo do promjene. Odmah kraj trenutne mase nalazi se i podatak o razlici početne korisnikove mase pri registriranju i zadnje ažurirane mase. Ispod tih podataka nalazi se lista posebno izdvojenih vježbi koje su odabrane na temelju korisnikove visine, težine te tjelesne aktivnosti. Korisnik ima mogućnost detaljnog pregleda svake pojedine vježbe. Svaka vježba ima opsežan opis izvođenja popraćen s dvije slike i navedenim glavnim mišićem koji izvodi vježbu. Ako je pri unosu nekih od parametara došlo do greške, korisnik ima mogućnost ispravka. Odlaskom na zaslon uređivanja profila, može se promijeniti svaki od parametara i ponovno spremi u bazu podataka. Također, ukoliko nakon izvjesnog vremena korisnik zaboravi lozinku svog profila, postoji opcija resetiranja lozinke. Ovo poglavlje pokriti će i dio testiranja aplikacije na različitim tipovima unosa. Postojat će tri vrste unosa, a to su ektomorfni, endomorfni i mezomorfni. Svaki od njih posjeduje različite karakteristike tijela kako bismo uvidjeli da će svaki od njih imati skup vježbi koji je različito zahtjevan. Fizički najspremniji tip tijela imat će najzahtjevnije tipove vježbi, dok će onaj najmanje spreman imati vježbe prilagođene početnicima.

5.2 Prikaz korištenja aplikacije

5.2.1 Prikaz registriranja korisnika

Pri prvom pokretanju aplikacije, otvara se zaslon registracije korisnika. Za uspješnu registraciju novog profila, korisniku je potrebna adresa e-pošte i lozinka koja sadržava najmanje šest znakova, od kojih barem jedan mora biti broj. Lozinku je potrebno unijeti dva puta kako bi se spriječila slučajna pogreška pri unosu, te bi se tako onemogućila daljnja prijava. Klikom na tipku „REGISTER“, šalje se zahtjev za kreiranjem profila s unesenim podacima. Ukoliko su podatci ispravni, te već ne postoji adresa s istom adresom e-pošte, profil se kreira i korisnik je automatski

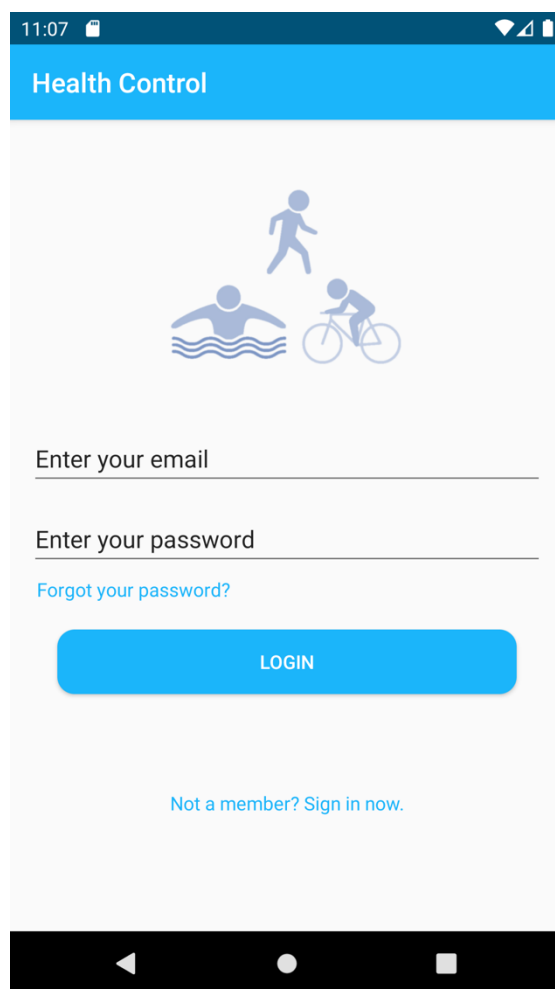
prosljeđen na zaslon prijave. Ako već postoji profil kreiran s unesenom adresom e-pošte ili je došlo do nekakve pogreške u sustavu, korisnik bude obaviješten „toast“ porukom da registriranje nije uspjele. Slika 5.1 prikazuje registracijsko sučelje aplikacije. Za slučaj kada korisnik već ima profil, postoji gumb za prelazak na zaslon prijave.



Slika 5.1. Registracijsko sučelje aplikacije

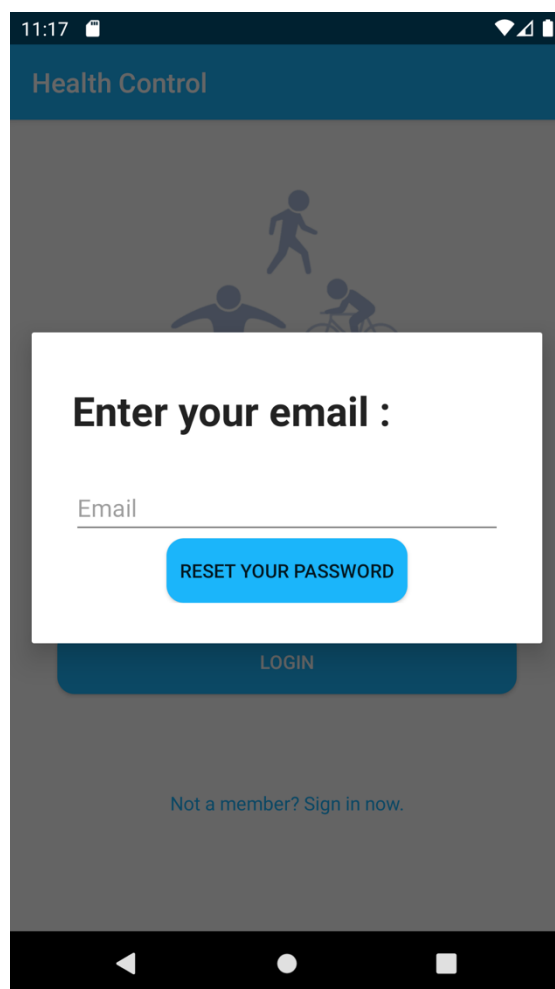
5.2.2 Prikaz prijave korisnika

Nakon uspješnog registriranja u sustav, korisnik je spreman prijaviti se. Prijava se izvršava s istim podacima kao i registriranje. Korisnik unosi adresu e-pošte i lozinku, te pritiskom na gumb „LOGIN“ šalje zahtjev za prijavu na bazu podataka. Ukoliko postoji profil s danim podacima, korisnik se šalje na zaslon unosa parametara. No, ako se korisnik ne prijavljuje prvi puta na svoj profil te je već unio svoje parametre, aplikacija ga vodi direktno na zaslon pregleda korisničkog profila. Slično kao i u primjeru registracije, ako korisnik nema izrađen profil, postoji mogućnost prelaska na zaslon registracije pomoću gumba pri dnu zaslona. Sučelje prijave je vidljivo na fotografiji 5.2.



Slika 5.2. Sučelje prijave aplikacije

Aplikacija ima mogućnost resetiranja lozinke korisničkog računa. Ukoliko korisnik zaboravi svoju lozinku, pritiskom na tipku “Forgot your password” otvara se skočni dijalog prikazan na slici 5.3, čiji je XML kod prikazan u poglavlju 4 na slici 4.7. U njega se unosi adresa e-pošte koju je korisnik koristio pri izradi korisničkog računa, te se pritisne “RESET YOUR PASSWORD”. Na tu adresu e-pošte stiže automatski sastavljena poruka od strane poslužitelja Firebase. Ona vodi na internetsku stranicu gdje korisnik upisuje novu lozinku koja se kasnije koristi u aplikaciji.



Slika 5.3. Dijalog za ponovno postavljanje lozinke

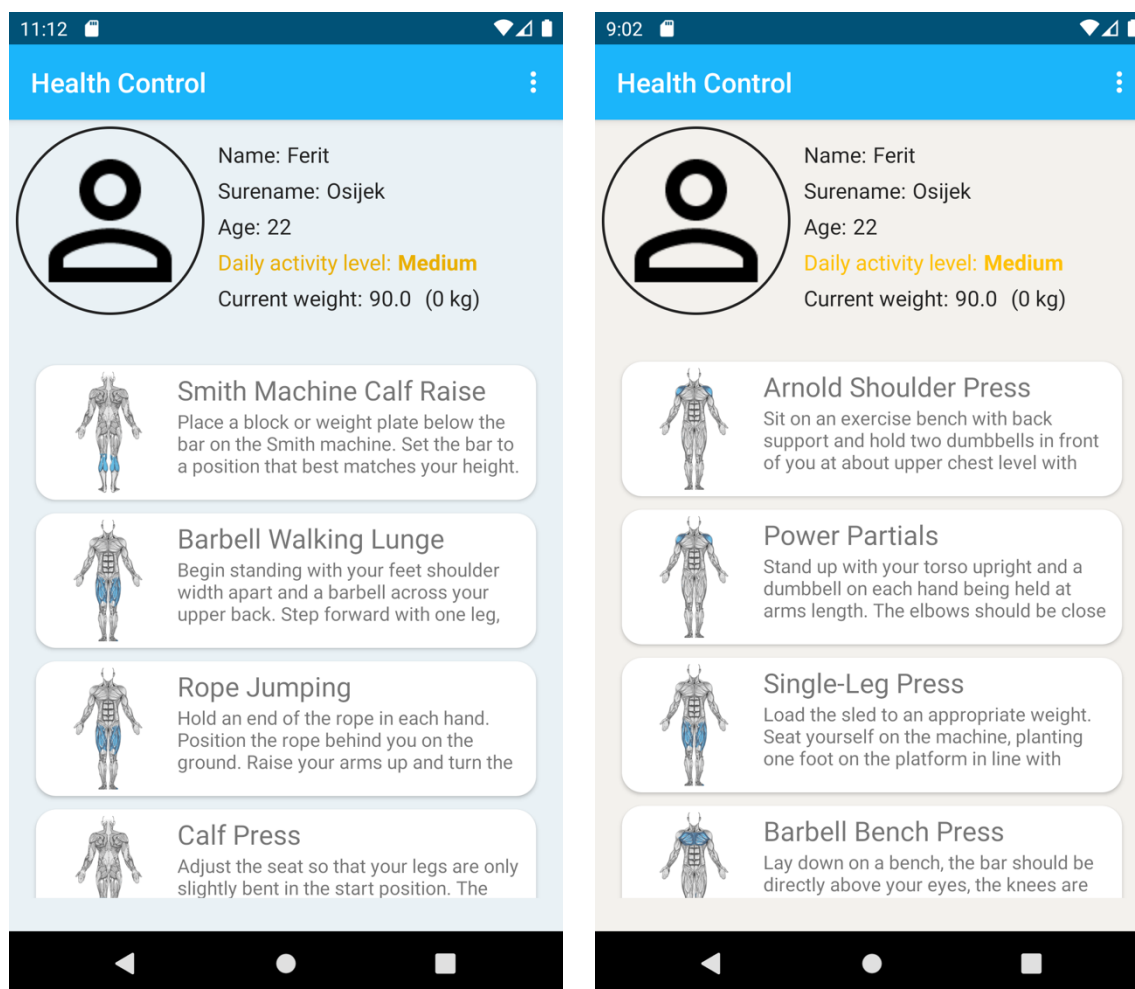
5.2.3 Prikaz unosa parametara

Zaslon sa slike 5.4 korisniku je vidljiv samo jednom. Nakon što se jednom ispuni obrazac sa podacima i uspješno ih se spremi, korisnik nije više u mogućnosti vraćati se na ovaj zaslon. Korisnik na njemu unosi svoje ime, prezime, visinu u centimetrima, masu u kilogramima, starost u godinama, te razinu dnevne tjelesne aktivnosti koja može varirati između „LOW“ kao niska razina, „MEDIUM“ kao srednja razina i „HIGH“ kao visoka razina tjelesne aktivnosti u danu. Postoji mogućnost koja nije obavezna, a to je učitavanje fotografije profila korisnika. Ukoliko korisnik ne učita fotografiju, imat će prethodno zadanu. Svi ovi podatci se spremaju na bazu podataka te su vezani jedinstvenim „ID“ ključem, koji ne može biti isti kod dva korisnika, tako da nema brige o gubitku podataka.

Slika 5.4. Unos dodatnih parametara

5.2.4 Prikaz pregleda korisničkog profila

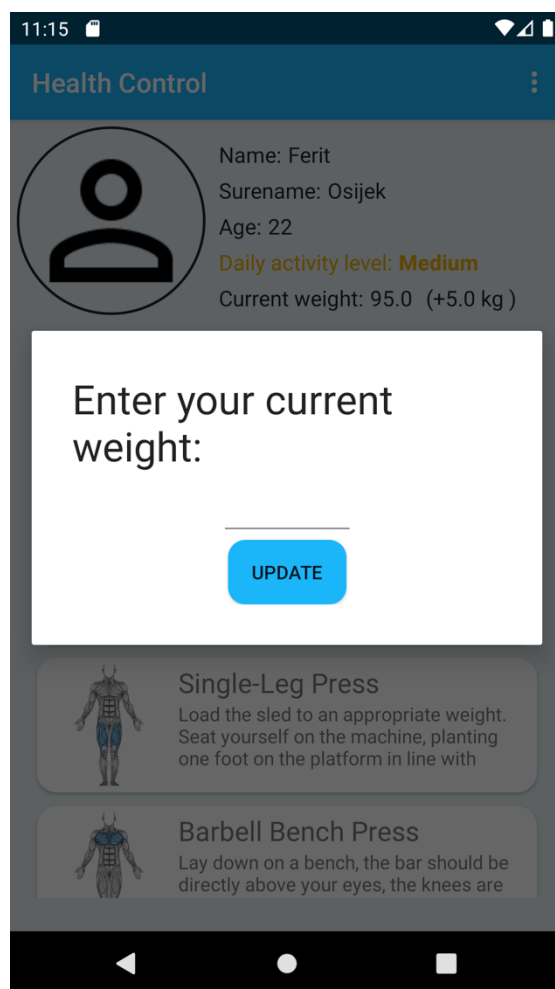
Osvrt na svoje registriranje i unos parametara korisniku je vidljiv na zaslonu pregleda računa. Taj zaslon se sastoji od njegove profilne fotografije, imena, prezimena, starosti, razine dnevne aktivnosti te trenutnoj masi. Iza trenutne mase stoji podatak o tome koliko se razlikuje masa unesena pri registriranju, te zadnje ažurirana masa. Ukoliko je korisnik smršavio 5 kilograma, masa u zagradi će imati predznak minus (-), u suprotnom bit će predznak plus (+). Boja pozadine varira u ovisnosti o spolu korisnika što se može vidjeti na slici 5.5. Ako se radi o muškarcu boja će biti svijetlo plava, dok je kod žena boja pastel žuta. Ispod svih navedenih podataka nalazi se ono najvažnije, a to je lista posebno izdvojenih vježbi, prilagođenih korisnikovim fizičkim predispozicijama i mogućnostima.



Slika 5.5. Zaslona pregleda profila sa različitim bojama pozadine

5.2.5 Prikaz ažuriranja podataka o masi

Pritiskom na tri točkice koje se nalaze u gornjem desnom dijelu zaslona otvara se padajući izbornik. Odabirom „Update weight“, otvara se skočni dijalog koji vidimo na slici 5.6 u koji se zatim unosi trenutna korisnikova masa. Nakon unosa se sprema i vraća na zaslon pregleda profila, a razlika u masama se automatski korigira. Sustav ne pamti sva moguća ažuriranja mase, nego samo početnu masu korisnika, te posljednju unesenu masu. XML kod dijaloga vidljiv je u poglavlju 4 na slici 4.14.



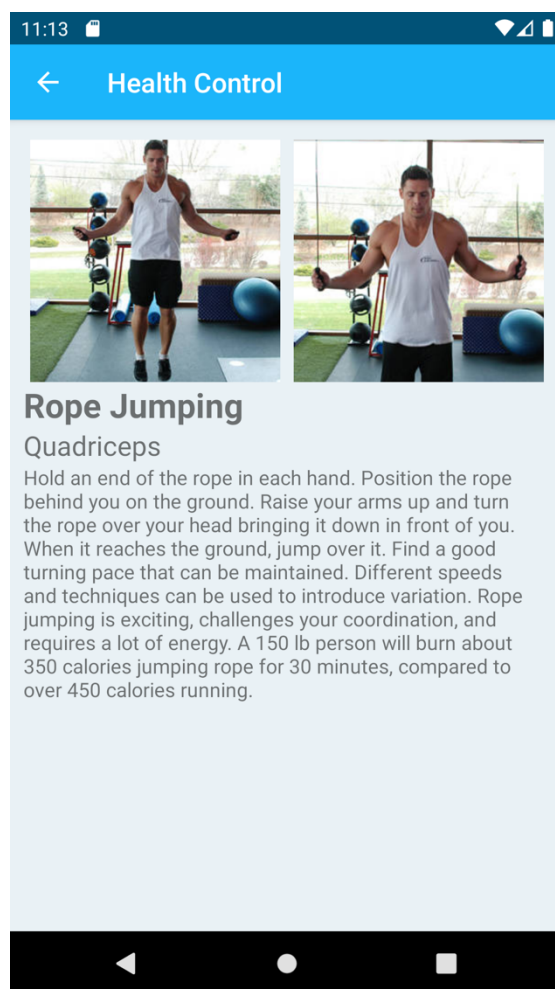
Slika 5.6. Prikaz skočnog prozora za ažuriranje mase

5.2.6 Prikaz uređivanja profila

Ukoliko je uočeno da je krivo unesen neki od parametara, može ga se izmijeniti odlaskom na zaslon uređivanja profila. Na njega se odlazi klikom na tri točkice koje se nalaze u gornjem desnom dijelu zaslona. Nakon klika spušta se padajući izbornik gdje se odabire „EDIT PROFILE“. Taj zaslon izgleda jednako kao i zaslon unosa parametara, osim što na zaslonu uređivanja već pišu korisnikovi parametri koji se mogu mijenjati.

5.2.7 Prikaz detaljnog pregleda vježbe

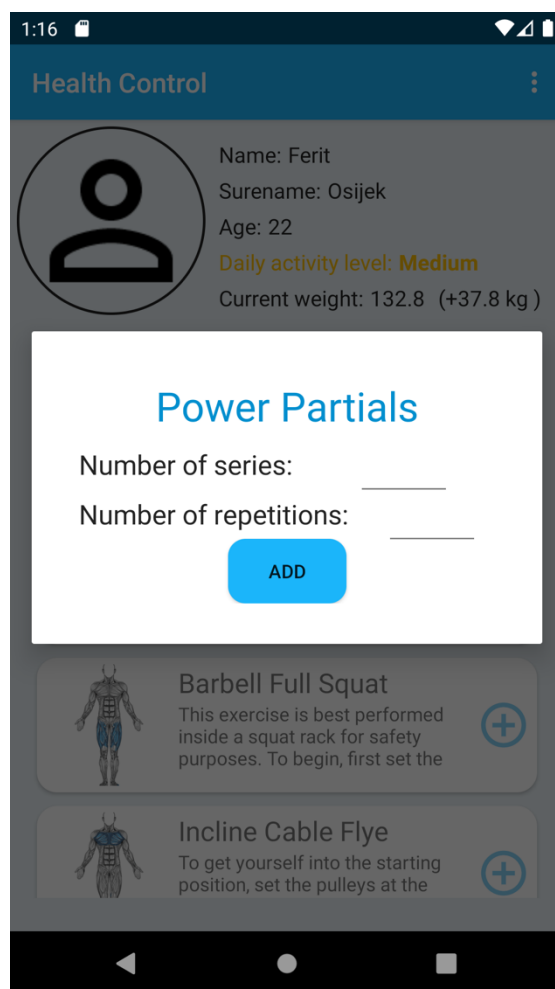
Kako bi korisnik u konačnici održavao zavidnu razinu tjelesne aktivnosti, mora znati pravilno izvoditi vježbe koje su mu zadane. Da se to što više olakša korisniku, aplikacija nudi detaljan pregled svake vježbe koji je vidljiv na slici 5.7. Upute za izvođenje sadrže se od dvije slike, imena vježbe, najopterećenijeg mišić, te kratkog ali preciznog opis izvođenja. XML kod koji daje izgled zaslona može se vidjeti na slici 4.17 u 4. poglavlju. Kao i u zaslonu pregleda profila, pozadina ovisi o spolu korisnika.



Slika 5.7. Detaljni pregled vježbe

5.2.8 Prikaz dodavanja vježbe u povijest

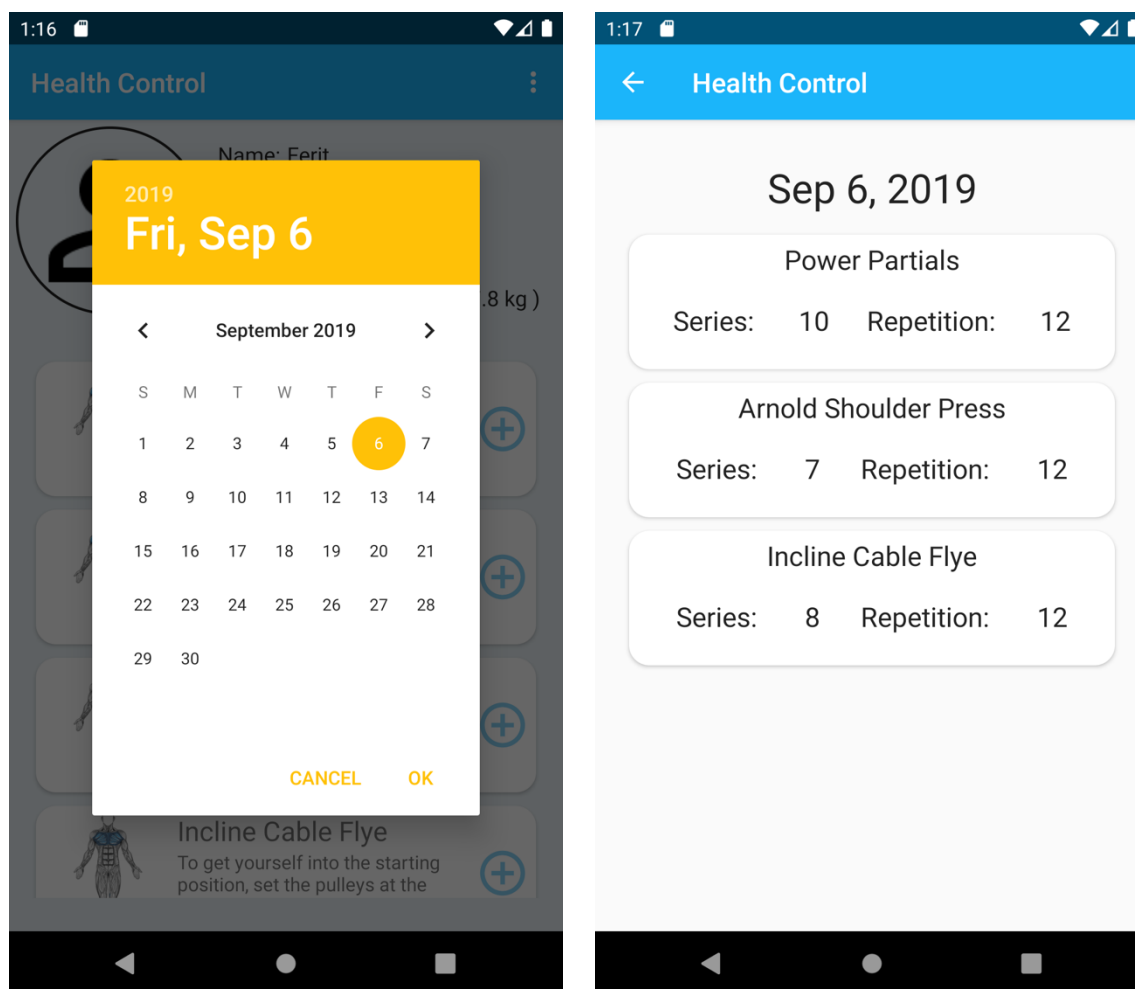
Radi lakšeg praćenja razine tjelesne aktivnosti tokom korištenja aplikacije, postoji opcija dodavanja vježbe u povijest odrađenih vježbi. Uz svaku ponuđenu vježbu postavljen je znak plusa „+“, na čiji klik se otvara dijalog za dodavanje vježbi u povijest vidljiv na slici 5.8. U dijalogu se nudi upis broja odrađenih serija te broja odrađenih ponavljanja. Klikom na gumb „ADD“, vježba je dodana u bazu, koja se kasnije može pregledati. Uz ime vježbe, broj serija i ponavljanja, spremi se i datum kada je vježba spremljena.



Slika 5.8. Dijalog dodavanja vježbe u povijest

5.2.9 Prikaz pregleda povijesti vježbanja

Radi uvida u odrađene vježbe, postoji opcija pregleda odrađenih vježbi po datumu odrađivanja. Klikom na tri točkice u gornjem desnom dijelu zaslona, otvara se padajući izbornik. Odabirom opcije „history“, otvara se kalendar, u kojem postoji mogućnost odabira određenog datuma. Nakon odabira datuma, otvara se zaslon na kojem su prikazane vježbe odrađene tog dana. Kalendar i prikaz vježbi mogu se vidjeti na slici 5.9.



Slika 5.9. Kalendar i zaslon povijesti vježbanja

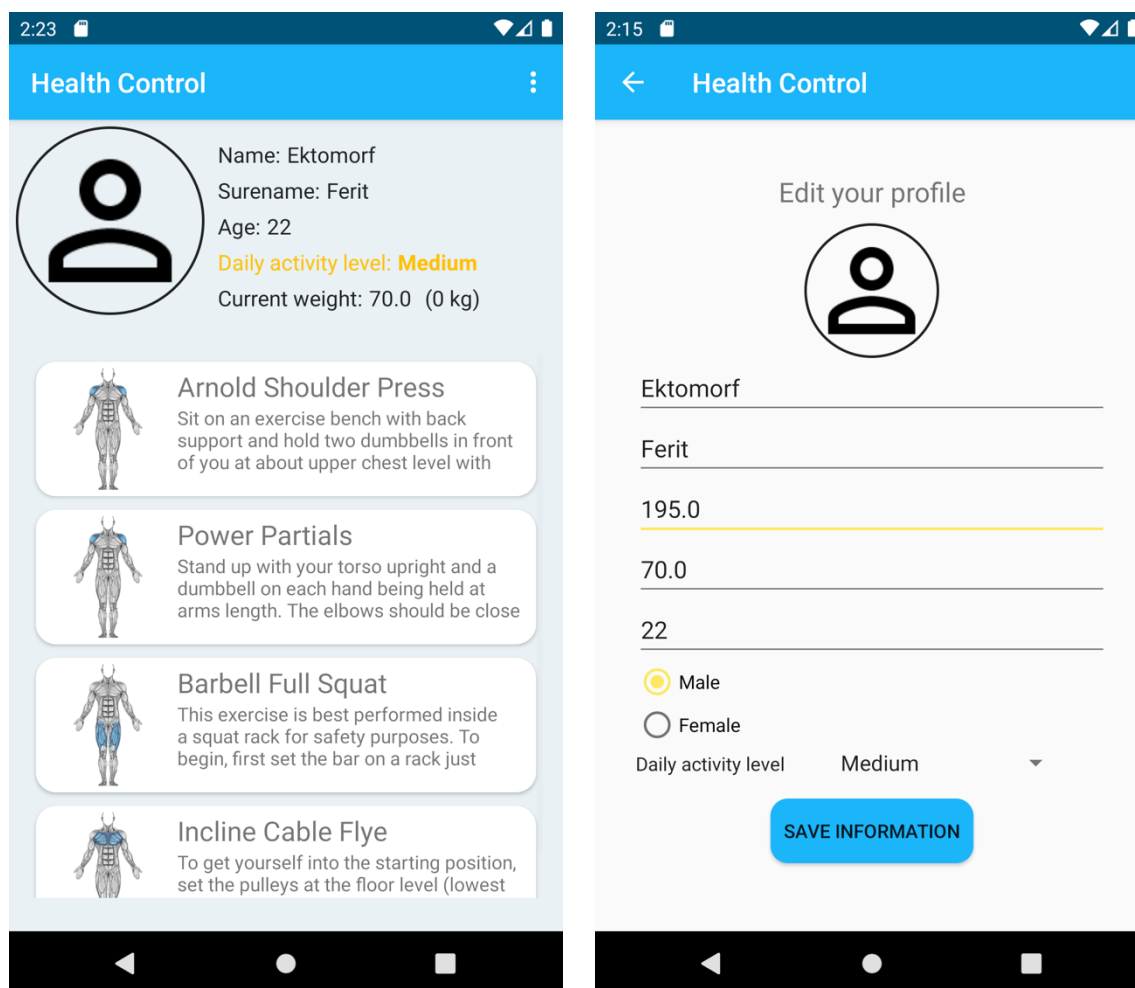
5.3 Slučajevi ispitivanja funkcionalnosti aplikacije

U nastavku su prikazana tri ključna slučaja ispitivanja aplikacije. Radi se o ektomorfnom, endomorfnom i mezomorfnom slučaju. Svaki od njih ima posebne fizičke karakteristike, te će dati različite izlazne rezultate. Svaki slučaj popraćen je slikama na kojima je prikazan pregled profila i vježbi te svi parametri korisnika.

5.3.1 Ektomorfnu slučaj

Prema [21] ektomorfnu osoba je visoka i mršava. Vrlo lako održava nizak postotak masnog tkiva bez velike aktivnosti, a o pretilosti ne mora brinuti. Jede samo kada ima potrebu za hranom, a može jesti bilo što bez brige o tome hoće li se udebljati ili dobiti masne naslage. Ostale karakteristike ektomorfa su uzak struk, tanki zglobovi, ravne grudi te ramena pognuta prema naprijed. U ovom testnom slučaju ektomorfnu korisnik ima srednju razinu tjelesne aktivnosti, što mu daje prednost nad niskim indeksom tjelesne mase. Intenzitet dodijeljen ovom korisniku je vrijednosti dva. To znači da osoba može izvoditi kompleksnije vježbe. Kao što je vidljivo na slici

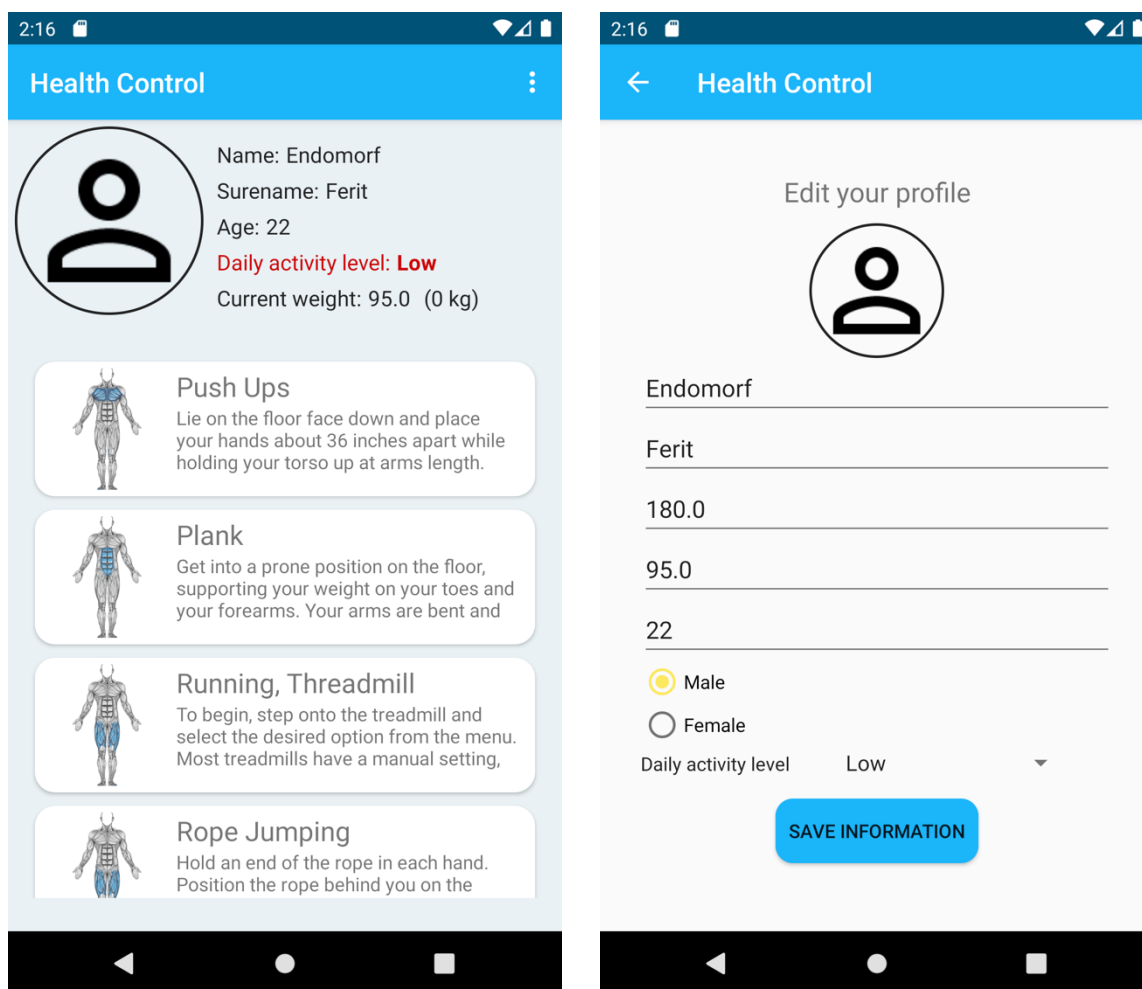
5.10, u njih već spadaju vježbe s utezima, kablovima te vježbe koje iziskuju više snage nego što ju ima osoba koja nema nikakvu tjelesnu aktivnost.



Slika 5.10. Skup vježbi intenziteta vrijednosti dva i informacije o ektomorfnom korisniku

5.3.2 Endomorfni slučaj

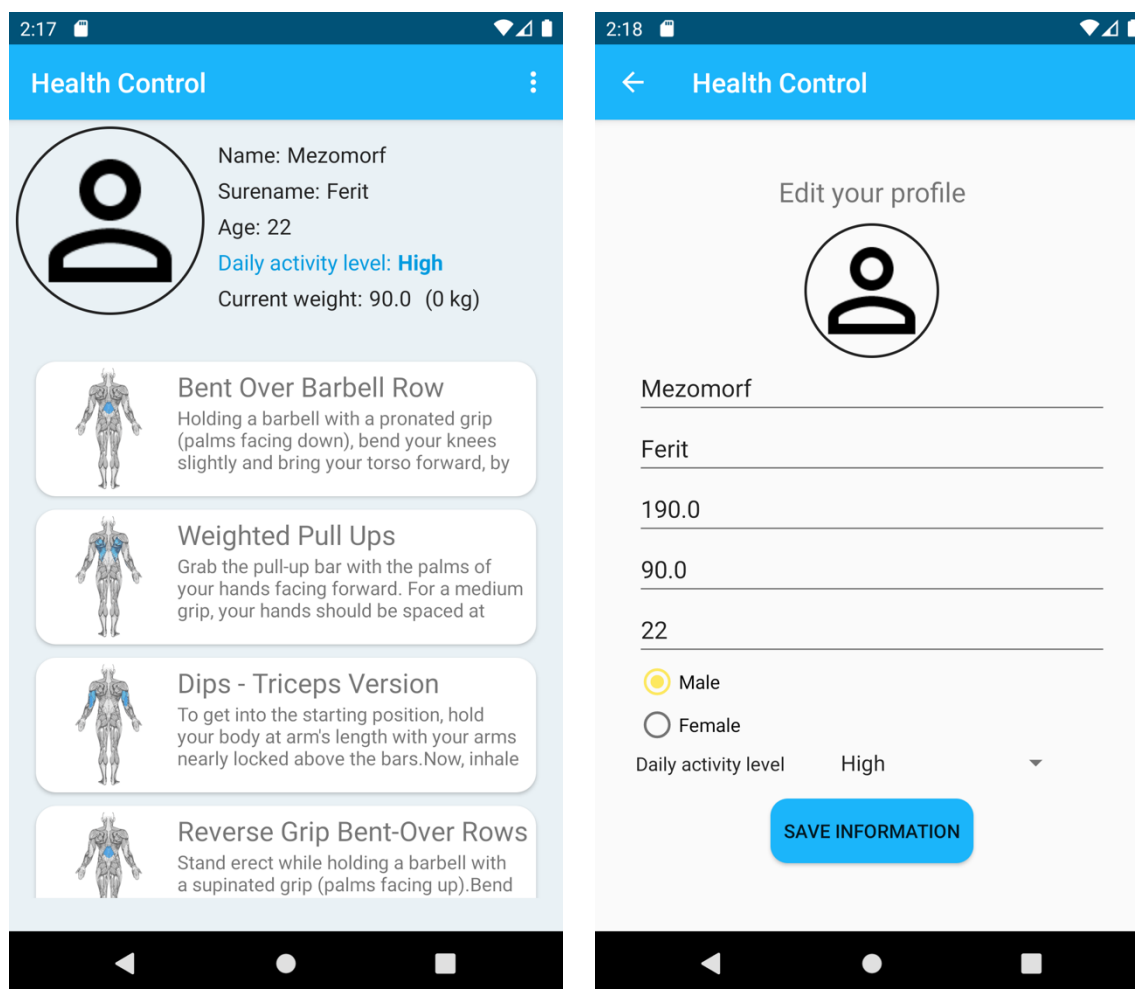
Endomorfni slučaj je osoba koja je potpuna suprotnost ektomorfnom korisniku. Prema [21] endomorf ima kruškolik oblik tijela i okruglije je građen. Ima široke zglobove, kratke ruke, te mekano tijelo puno masnih naslaga. Prati ga spor metabolizam što rezultira lakim dobivanjem masnog tkiva. Uz prekomjernu težinu, ovaj testni slučaj ima iznimno nisku razinu tjelesne aktivnost, što nas dovodi do vježbi intenziteta vrijednosti jedan. To su najjednostavnije vježbe. Većinom se izvode bez utega, a uključuju mnoštvo kardio vježbi kao što su trčanje i preskakanje užeta, što je vidljivo na slici 5.11.



Slika 5.11. Skup vježbi intenziteta vrijednosti jedan i informacije o endomorfnoj osobi

5.3.3 Mezomorfni slučaj

Za razliku od prva dva testna slučaja, mezomorf ima čvrsto tijelo, mišićave je građe uz vrlo mal udio masnog tkiva. Prema [21], ima vrlo muževan, zreo izgled koji je popraćen uspravnim držanjem. Ovakav tip osobe se u većini slučajeva bavi sportom i svi mu zavide na lakom dodavanju čiste mišićne mase. U ovom testnom slučaju razina tjelesne aktivnosti korisnika je visoka što ukazuje na najveći stupanj intenziteta, a to je vrijednost tri. U toj grupi nalaze se najkompleksnije vježbe koje zahtijevaju velik izvor snage, izdržljivosti te spretnosti. Naizgled jednostavne vježbe su ovdje zakomplicirane dodavanjem utega kao na primjer vježba propadanja s dodatkom utega, što je vidljivo na slici 5.12.



Slika 5.12. Skup vježbi intenziteta vrijednosti tri i informacije o mezomorfnoj osobi

5.4 Usporedba slučajeva ispitivanja aplikacije

Svaki od slučajeva ispitivanja aplikacije dao je različit skup izlaznih podataka. Endomorfni slučaj, koji predstavlja osobu okruglije građe i viškom kilograma, imao je intenzitet vrijednosti jedan. To znači da je ta osoba najmanje spremna na fizičku aktivnost. Ektomorfni slučaj predstavljao je visoku i mršavu osobu. Obje osobe su imale nepovoljan indeks tjelesne mase, no veliku razliku između ova dva slučaja činila je razina dnevne aktivnosti. Endomorfni slučaj je imao nisku razinu dnevne tjelesne aktivnosti, dok je ektomorfni imao srednju, što mu je dalo intenzitet vrijednosti dva. Najpovoljniji slučaj bio je mezomorfni. Ova osoba imala je visoku razinu dnevne tjelesne aktivnosti te povoljan indeks tjelesne mase. Takva osoba dobila je intenzitet vrijednosti tri, što govori da je spremna izvoditi vrlo kompleksne vježbe.

ZAKLJUČAK

U ovom radu razvijena je mobilna aplikacija za potporu i praćenje tjelovježbe. Aplikacija je razvijena u svrhu lakšeg praćenja tjelesne aktivnosti korisnika s ciljem poticanja na veću tjelesnu aktivnost i vježbanje. Jednostavno korištenje aplikacije korisniku omogućuje registriranje, prijavu, unos vlastitih parametara, te praćenje svog napretka kroz bilježenje odrađenih vježbi i ažuriranje mase, što uvelike može motivirati korisnika na daljnji rad i razvoj svojih fizičkih predispozicija. Korištenje ove aplikacije olakšava korisniku brigu o svome tijelu, što se odražava na njegovo fizičko i psihičko zdravlje. Osim ovog rješenja, veliku ulogu u održavanju pristojne razine tjelesne aktivnosti ima društvo u kojem se korisnik nalazi. Sve dobne skupine sve češće nailaze na problem smanjene tjelesne aktivnosti, što dovodi do toga da ova aplikacija ima širok spektar mogućih korisnika. Prikladna je kako za adolescente koji sve više vremena provode neaktivno, tako i za zrelije ljude koji zbog prirode posla možda ne mogu biti dovoljno aktivni.

Razvoj aplikacije obavljen je u razvojnoj okolini Android Studio, te je korišten sve popularniji programski jezik Kotlin. Aplikacija je ispitana na trenutno najnovijem Android sustavu inačice 10.0. Ispitivanje funkcionalnosti aplikacije pokazuje ispravnost u njenom radu. Vidljivo je da za različite ulazne podatke, aplikacija nudi različit spektar prikladnih vježbi koje odgovaraju ulaznim podacima. Ispitivanje aplikacije dovelo je do zaključka da bi se ona mogla poboljšati unosom veće baze vježbi, što bi dovelo do većeg izbora za korisnika. Također, veći broj vježbi poboljšao bi odabir najprikladnijih vježbi za određenog korisnika.

LITERATURA

- [1] N. Fluetsch, C. Levy, L. Tallon, (2015). The relationship of physical activity to mental health. *Journal of Affective Disorders*, 253, 96-101, pristupljeno 03.08.2019.
- [2] S. Carbone, M. G. Del Buono, C. Ozemek, C. J. Lavie, (2019). Obesity, risk of diabetes and role of physical activity, exercise training and cardiorespiratory fitness, 20 August 2019., <https://www.sciencedirect.com/science/article/abs/pii/S0033062019301070>, pristupljeno 04.08.2019.
- [3] WHO-World Health Organization. (2010). *Global Recommendations on Physical Activity for Health*. Geneva, pristupljeno 07.08.2019.
- [4] D. Jurakić, S. Heimer, (2012). Prevalencija nedovoljne tjelesne aktivnosti u Hrvatskoj i u svijetu: pregled istraživanja. *Arhiv za higijenu rada i toksikologiju*, 63 (Supplement 3), 3-11., pristupljeno 07.08.2019.
- [5] A. L. Rebar, A. Taylor, (2017). Physical activity and mental health, it is more than just a prescription, <https://www.sciencedirect.com/science/article/abs/pii/S1755296617301539>, pristupljeno 08.08.2019.
- [6] S. Janković, (2003). *50 savjeta za zdravlje*. Zagreb: Hrvatski zavod za javno zdravstvo, pristupljeno 08.08.2019.
- [7] A. Matković, A. Nedić, M. Meštrov, J. Ivković, (2010). *Uobičajena tjelesna aktivnost studenata Medicinskog fakulteta Sveučilišta u Zagrebu*. *Hrvatski športskomedicinski vjesnik*, 25(2), 87-91., pristupljeno 08.08.2019.
- [8] A. Močnik, M. Neuberg, I. Canjuga, (2015). *Tjelesna aktivnost starijih osoba smještenih u stacionarnim ustanovama*. *Tehnički glasnik*, 9(1), 112-119., pristupljeno 08.08.2019.
- [9] V. Šimunić, R. Barić, (2011). *Motivacija za vježbanje povremenih rekreativnih vježbača: spolne razlike*. *Hrvatski športskomedicinski vjesnik*, 26(1), 19-25., pristupljeno 09.08.2019.

- [10] M. Rakovac, (2012). *Edukacija zdravstvenih kadrova o zdravstveno usmjerenom tjelesnoj aktivnosti*. Arhiv za higijenu rada i toksikologiju, 63, 67., pristupljeno 09.08.2019.
- [11] N. Foretić, S. Rodek, D. Mihaljević, (2009). *Utjecaj medija na fizičku inaktivnost djece*. Školski vjesnik, 58 (4.), 381-397., pristupljeno 09.08.2019.
- [12] Samsung Health,
<https://www.samsung.com/hr/apps/samsung-health>, pristupljeno: 11.08.2019.
- [13] Google Fit, <https://www.google.com/fit/>, pristupljeno: 11.08.2019.
- [14] BodBot: Personal Trainer,
https://play.google.com/store/apps/details?id=com.bodbot.trainer&hl=en_US, pristupljeno:
11.08.2019
- [15] Nivea, <https://www.nivea.hr/nivea-noviteti/indeks-tjelesne-mase>, pristupljeno 26.7.2019.
- [16] Exercise guides, <https://www.bodybuilding.com/exercises/>, pristupljeno 27.7.2019.
- [17] Android Studio, <https://developer.android.com/studio>, pristupljeno 27.7.2019.
- [18] Documentation, Android Developers,
<https://developer.android.com/guide/topics/text/autofill-optimize>, pristupljeno 03.08.2019.
- [19] Kotlin, <https://kotlinlang.org>, pristupljeno 27.7.2019.
- [20] Rakshit Soral, Architectural Guidelines to follow for MVP pattern in Android, 6 March 2018, <https://android.jlelse.eu/architectural-guidelines-to-follow-for-mvp-pattern-in-android-2374848a0157>, pristupljeno 28.7.2019.
- [21] Firebase Docs, <https://firebase.google.com/docs>, pristupljeno 28.7.2019.
- [22] EXtensible Markup Language Wikipedija,
<https://hr.wikipedia.org/wiki/XML>, pristupljeno 4.9.2019.

SAŽETAK

U ovom radu razvijena je aplikacija za potporu i praćenje tjelovježbe. Aplikacija je razvijena u svrhu lakšeg održavanja potrebne razine aktivnosti čovjeka s ciljem preporuke prikladnog skupa vježbi koje bi korisniku bile najbolje za izvođenje. Korisnik nakon registriranja i unosa parametara dobije listu vježbi, te uz svaku vježbu ima detaljan opis i slikovni prikaz izvođenja iste. Nakon nekog vremena provođenja vježbi korisnik može izmijeniti svoju težinu, a vježbe će se po potrebi automatski korigirati ukoliko je to potrebno. Analiza rada aplikacije pokazuje da aplikacija uspješno potiče korisnika na bavljenje redovitom tjelovježbom.

Ključne riječi: Android, mobilna aplikacija, zdravlje, tjelovježba.

ABSTRACT

This final work evolves an application that monitors body exercise. The application is developed in order to easily maintain the required physical activity of an individual, recommending a suitable set of exercises that would be the best for the user to perform. After registering and entering parameters, the user is provided with a list of body exercises and with each one of it stands a detailed description that comes with the picture of a performance of an exercise. After a while, user can change its weight data after performing specific exercises. Due to the change of weight data, if necessary, the list of body exercises will adjust automatically. The application analysis indicates that the mobile application successfully encourages its users to exercise regularly.

Key words: Android, Mobile application, health, body exercise

ŽIVOTOPIS

Zvonimir Pavlović rođen je 07.11.1997. godine u Našicama. Pohađao je Osnovnu školu Ivane Brlić Mažuranić u Koški. Upisuje III. Gimnaziju u Osijeku 2012. godine koju uspješno završava 2016. Iste godine upisuje Fakultet elektrotehnike, računarstva i informacijskih tehnologija u Osijeku, preddiplomski studij računarstva.

PRILOZI (na CD-u)

Prilog 1: Završni rad „Mobilna aplikacija za potporu i praćenje tjelovježbe“ u *.docx* formatu.

Prilog 2: Završni rad „Mobilna aplikacija za potporu i praćenje tjelovježbe“ u *.pdf* formatu.

Prilog 3: Programski kod mobilne aplikacije za potporu i praćenje tjelovježbe.