

# Mobilna aplikacija za prevenciju pretilosti prikupljanjem parametara o životnim navikama

---

**Radić, Svetozar**

**Undergraduate thesis / Završni 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:200:171418>

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

*Download date / Datum preuzimanja:* **2025-01-30**

*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**

**Mobilna aplikacija za prevenciju pretilosti  
prikupljanjem parametara o životnim navikama**

**Završni rad**

**Svetozar Radić**

**Osijek, 2018.**

## Sadržaj

1. UVOD .....	1
1.1. Zadatak završnog rada .....	2
2. PRETILOST I PROBLEMI KOJE DONOSI .....	3
2.1. Simptomi pretilosti .....	3
2.2. Prevencija pretilosti .....	4
2.3. Liječenje pretilosti .....	4
2.3.1. Prikaz postojećih računalnih rješenja za pomoć kod prevencije pretilosti .....	5
3. MODEL APLIKACIJE ZA PREVENCIJU PRETILOSTI .....	6
3.1. Registracija korisnika .....	6
3.2. Prijava korisnika .....	6
3.3. Unos visine, težine i broja godina korisnika.....	7
3.4. Povezivanje s narukvicom Fitbit Charge HR .....	7
3.5. Obrada i analiza podataka prikupljenih s Fitbit Charge HR narukvice.....	7
3.6. Odjava korisnika.....	7
4. PROGRAMSKO RJEŠENJE APLIKACIJE ZA PREVENCIJU PRETILOSTI .....	9
4.1. Opis alata i tehnologija.....	9
4.1.1. Android Studio.....	9
4.1.2. IntelliJ IDEA .....	11
4.1.3. Operacijski sustav Android.....	11
4.1.4. Narukvica Fitbit Charge HR .....	13
4.1.5. Standard OAuth 2.0 .....	14
4.2. Razvoj i način rada sustava za prevenciju pretilosti.....	14
4.2.1. Prikaz programskog rješenja po komponentama .....	15
4.2.1.1. Postupak registracije i prijave korisnika .....	15
4.2.1.2. Rad s bazom podataka SQLite .....	18
4.2.1.3. Postupak unosa visine, težine i broja godina.....	19

4.2.1.4. Postupak povezivanja s narukvicom Fitbit Charge HR .....	21
4.2.1.4. Postupak obrade podataka prikupljenih s narukvice Fitbit Charge HR .....	22
5. PRIKAZ KORIŠTENJA I ISPITIVANJE RADA APLIKACIJE .....	24
5.1. Prikaz registracije korisnika .....	25
5.2. Prikaz prijave korisnika .....	27
5.3. Prikaz unosa visine, težine i broja godina korisnika .....	28
5.4. Prikaz povezivanja s Fitbit Charge HR narukvicom .....	29
5.5. Prikaz obrade podataka prikupljenih s narukvice Fitbit Charge HR .....	30
5.6. Prikaz rada sustava na primjeru .....	35
6. ZAKLJUČAK .....	39
LITERATURA .....	40
SAŽETAK .....	43
ABSTRACT .....	44
ŽIVOTOPIS .....	45
PRILOZI .....	46

## 1. UVOD

Tema ovog završnog rada je izgradnja mobilne aplikacije kojom će se spriječiti nastajanje pretilosti kod korisnika prikupljanjem parametara koji opisuju način života korisnika, te omogućiti unos parametara bitnih za prevenciju pretilosti. Na temelju prikupljenih i unesenih parametara, korisniku se generiraju preporuke i alarmiraju ga u slučaju nedopuštenog ponašanja. U praktičnom dijelu rada ostvareno je programsko rješenje koje omogućuje primjenu mobilnih i komunikacijskih tehnologija pomoću kojih se ostvaruje jedno od rješenja pri prevenciji pretilosti. Razvijeni sustav namijenjen prevenciji pretilosti sastoji se od: (1) *fitness* narukvice Fitbit Charge HR za praćenje tjelesne aktivnosti korisnika, (2) Android mobilne aplikacije za dohvaćanje podataka s Fitbit-ovog web poslužitelja, analizu i obradu prikupljenih podataka, te pružanje pristupačnog korisničkog sučelja korisniku.

Pretilost je jedan od vodećih problema koji ugrožava opće zdravstveno stanje te dovodi do stvaranja ostalih bolesti. Kako u svijetu tako i u Hrvatskoj pretilost izaziva velike probleme za zdravlje. Prema podacima Hrvatskog zavoda za javno zdravstvo [HZZJZ], 60% populacije u Republici Hrvatskoj ima prekomjernu tjelesnu težinu. Incidencija pretile djece u populaciji u stalnom je porastu. Zabilježeno je da 12% djece koja pohađaju osnovnu školu ima povećanu tjelesnu težinu, a od navedenog postotka 5% djece je pretilo. Pretile osobe često obolijevaju od kardiovaskularnih bolesti, te imaju povišeni krvni tlak, dijabetes, otežano dišu i hodaju te imaju razne druge probleme. Osobe koje boluju od pretilosti često moraju aktivno pratiti i regulirati svoju prehranu i tjelesnu aktivnost. Razvojem tehnologije omogućeno je stvaranje i usavršavanje senzorskih uređaja koji uvelike pomažu prilikom praćenja i kontroliranja zdravlja. Također je olakšano i liječnicima praćenje bolesnika oboljelih od pretilosti, ali i potaknuta veća aktivnost oboljelih kako bi sami sebi pomogli i očuvali svoje zdravlje. Osobe koje boluju od pretilosti često ne znaju kako se nositi s tim problemom te pokušavaju razne tretmane i tablete koje su neučinkovite i skupe. Pomoću današnje tehnologije omogućeno je smanjenje broja pretilih osoba pomoću raznih alata i pametnih narukvica koje u skladu s programskom podrškom omogućuju korisniku lakše praćenje dnevnih aktivnosti i na taj način lakše sprječavanje nastanka pretilosti.

Na početku drugog poglavlja opisana je pretilost, način na koja ona nastaje i problemi koje ona stvara. Opisana je uloga pretilosti u životu osobe te na koji način se ona manifestira na psihičko i fizičko stanje osobe. Također su opisani simptomi pretilosti, kako je prepoznati, te na koji način se može spriječiti i liječiti. Treće poglavlje sadrži opis modela aplikacije za prevenciju pretilosti koja je bazirana na opisima pojedinih komponenata aplikacije. U četvrtom poglavlju opisano je programsko rješenje aplikacije za prevenciju pretilosti te alati, računalne platforme i tehnologije korištene za izradu cjelokupne programske podrške koji su popraćeni odgovarajućim kodovima za pojedina rješenja. U petom poglavlju opisuju se značajke i način rada cjelokupnog sustava, prikazani su pojedini dijelovi sučelja, te ispitan i analiziran način rada sustava. U zadnjem poglavlju se nalazi cjelokupni sažeti prikaz ovog rada te njegov zaključak .

### **1.1. Zadatak završnog rada**

U radu je potrebno osmisliti i izraditi mobilnu aplikaciju za Android platformu s odgovarajućom bazom podataka. Aplikacija treba s *bluetooth* uređaja Fitbit Charge HR prikupljati parametre koji opisuju način života korisnika, te omogućiti unos parametara bitnih za prevenciju pretilosti. Na temelju prikupljenih i unesenih parametara, korisniku treba generirati preporuke i alarmirati ga u slučaju nedopuštenog ponašanja. Rješenje treba ispitati na prikladnom skupu ulaznih podataka.

## **2. PRETILOST I PROBLEMI KOJE DONOSI**

Pretilost je kronična bolest koja nastaje prekomjernim nakupljanjem masti u organizmu i povećanjem tjelesne težine. Stupanj pretilosti se može dobiti računanjem indeksa tjelesne mase (engl. "*BMI*"). Taj broj se dobije kada se težina osobe u kilogramima podjeli s kvadratom visine osobe u metrima. Osoba čiji je indeks tjelesne mase veći od 27 smatra se pretilom. Vrijednosti preporučenog indeksa tjelesne mase iste su za oba spola, on iznosi od 18,5 – 24,9 kg/m<sup>2</sup> prema klasifikaciji Svjetske zdravstvene organizacije za europsko stanovništvo. Indeks tjelesne mase nije jedina i cjelovita mjera, jer će više mišićavije osobe općenito imati viši indeks tjelesne mase, iako će udio masnog tkiva biti relativno nizak. Pri procjeni rizika od razvoja bolesti povezanih s pretilošću važno je uzeti u obzir i neke druge mjere koje ukazuju na raspodjelu masnog tkiva u tijelu poput odnosa opsega struka i bokova ili samo opsega struka.

Visok indeks tjelesne mase ukazuje i na stupanj rizika od razvoja nekih bolesti poput visokog krvnog tlaka, šećerne bolesti, poremećaja razine masnoća u krvi, krvožilnih bolesti uključujući srčani i moždani udar, žučnih kamenaca, osteoartritisa, nekih vrsta raka i drugih. Što je indeks tjelesne mase viši, to je i veći rizik obolijevanja od navedenih bolesti. Osobe oboljele od pretilosti često obolijevaju i od depresije, često su nezadovoljne svojim izgledom pa se javljaju problemi u prehrani i smanjuje im se nivo samopoštovanja.

Važno je naglasiti da indeks tjelesne mase treba uzeti kao okvirnu metodu zato što stvarno zdravstveno stanje osobe treba ocijeniti širom medicinskom analizom.

U ovom poglavlju bit će opisani simptomi pretilosti, način na koji se pretilost može spriječiti, te načini i metode liječenja pretilosti, kao u [1] i [2].

### **2.1. Simptomi pretilosti**

Pretilost nastupa kada kroz duže vremensko razdoblje osoba unosi više kalorija nego što potroši. Javljaju se velike količine masnih naslaga po tijelu (salo). Kao što je navedeno pretila osoba imaju indeks tjelesne mase veći od 30. S pretilošću se znatno povećava rizik i za niz drugih zdravstvenih problema. No, pretilost ne nastaju samo nepravilnom prehranom, već na nju utječu i razni genetički, psihosocijalni, socio-ekonomski i okolišni čimbenici. Kod pretilih osoba javljaju se teškoće s disanjem, brzo umaranje, visok krvni tlak, dijabetes, bol u zglobovima i slično. Osim problema na medicinskom području, pretilost donosi i niz psihosocijalnih problema. Takve osobe imaju nisko samopouzdanje, mogu patiti od depresije, često ih se

diskriminira te im je generalno narušena kvaliteta života. Simptomi pretilosti prikazani su u tablici 2.1.

Tab. 2.1. Simptomi pretilosti

<b>SIMPTOMI PRETILOSTI</b>
Otežano disanje
Brzo umaranje
Visoki krvni tlak
Dijabetes
Povećana kilaža
Bol u zglobovima
Depresija

## **2.2. Prevencija pretilosti**

S obzirom na to da je pretilost sve češće stanje u svijetu važno je znati načine prevencije. Treba imati uravnotežene obroke te raznoliku prehranu. Treba uvijek imati na umu da, što je raznolikija prehrana, to je unos značajnih makronutrijenata i mikronutrijenata zadovoljavajući. Fizička aktivnost je vrlo bitan zaštitni faktor, ne samo u prevenciji pretilosti već i za smanjivanje rizika od kardiovaskularnih bolesti, osteoporoze i raka. Preporučuje se barem 30 minuta kretanja dnevno, na bilo koji način- hodanje, trčanje ili kućanski poslovi. To ne mora samo označavati naporno vježbanje u teretani, dovoljno je i zamijeniti dizalo stubama ili otići pješke umjesto autom.

## **2.3. Liječenje pretilosti**

Kada pretilost postane problem koji zahtjeva liječničku intervenciju, koriste se i lijekovi za smanjivanje apetita. Naravno, oni neće sami istopiti kilograme i masti već se koriste u kombinaciji s prepisanom dijetom i tjelovježbom. Zapravo, nema određenog lijeka osim promjene stila života. Potrebna je dosljednost u držanju dijetе i redovita tjelovježba. Pomoć pri liječenju pretilosti jeste korištenje raznih *online* stranica te raznih aplikacija koje će olakšati liječenje pretilosti korisnicima.



### 2.3.1. Prikaz postojećih računalnih rješenja za pomoć kod prevencije pretilosti

Postoje brojna postojeća rješenja za pomoć kod prevencije pretilosti. Budući da je danas tehnologija vrlo napredna olakšano je praćenje korisnika i njegova prevencija od pretilosti. Nekih od već gotovih rješenja su :

- Ontology-based Obesity Tracking System for Children and Adolescents – omogućava konzultacije sa stručnjacima za zdravlje putem mobilnog uređaja. Također sadržava vlastitu bazu podataka s uputama za prevenciju pretilosti i za liječenje pretilosti [3].
- Cloud and sensors based obesity monitoring system – omogućava praćenje korisnika u stvarnom vremenu, obrađivanje podataka i dijeljenje savjeta korisniku kako bi poboljšao korisnikovo zdravlje [4].
- FitSpace –web aplikacija napravljena za pomoć kod pretilosti kod djece [5].

Također postoje razni *tutoriali* za prevenciju pretilosti ,kao i *online* treninzi i savjetovanja s liječnicima, stručnjacima za zdravlje, *fitness* trenerima itd.

Navedena postojeća računalna rješenja za pomoć kod prevencije pretilosti omogućuju konzultacije sa stručnjacima za zdravlje, praćenje korisnika u stvarnom vremenu, web *tutoriale*, razgovor s *fitness* trenerima itd. Za razliku od navedenih rješenja, aplikacija za prevenciju pretilosti prikupljanjem parametara o životnim navikama koristi *fitness* narukvicu Fitbit Charge HR koja pomoću moderne tehnologije prikuplja podatke o korisnikovim dnevnik aktivnostima te ih sprema na web poslužitelj. Sličnost koju ima s navedenim rješenjima je to što aplikacija prikuplja podatke u stvarnom vremenu te koristi web poslužitelj kao bazu podataka za spremanje prikupljenih podataka.

Nadalje, izrađena aplikacija obrađuje prikupljene podatke te pruža korisniku preporuke koji ukazuju korisniku na koji način treba poboljšati svoju dnevnu aktivnost. Aplikacije se može koristiti kod osoba neovisno o njihovim godinama.

### **3. MODEL APLIKACIJE ZA PREVENCIJU PRETILOSTI**

U ovoj aplikaciji zahtjeva se od korisnika da se prijavi tj. unese svoje podatke kao što su korisničko ime, *e-mail* i lozinka te se ti podaci spremaju u bazu podataka koja je ostvarena pomoću SQLite-a. Pomoću tih podataka se korisnika prijavljuje na svoj korisnički račun koji će sadržavati podatke o parametrima prikupljenih pomoću Fitbit Charge HR *fitness* narukvice. Tijekom nošenja narukvice, prikupljaju se podaci kao što je broj prijeđenih koraka, broj prijeđenih stepenica, broj kalorija koje je korisnik potrošio, otkucaji srca itd. Nakon što su takvi podaci prikupljeni oni se šalju na web poslužitelj na koji korisnik pristupa preko OAuth 2.0 standarda, ali prije toga se mora napraviti korisnički račun na službenoj Fitbit stranici te prijaviti sama aplikacija kako bi korisnik dobio ID pomoću kojeg može dobiti *token* kojim se određuje pravo pristupa podacima na web poslužitelju. Nakon toga slijedi obrada prikupljenih podataka te njihovo daljnje analiziranje i prilagođavanje za prevenciju pretilosti kroz alarmiranje korisnika.

#### **3.1. Registracija korisnika**

Nakon što korisnik preuzme i instalira aplikaciju najprije mora obaviti *sign up* odnosno registrirati se ako prvi put koristi aplikaciju, kako bi dalje mogao koristiti aplikaciju. Prilikom prijave od korisnika se zahtjeva korisničko ime, *e-mail* te lozinka. Nakon što obavi navedeno, korisnikovi podaci se spremaju u bazu podataka te korisnik nastavlja s daljnjom uporabom aplikacije.

#### **3.2. Prijava korisnika**

Prilikom svake novog korištenja aplikacije korisnik će se morati prijaviti odnosno obaviti *log in* pomoću podataka koje je naveo prvi put prilikom korištenja aplikacije. Prilikom prijave korisnik će imati opciju „zapamti moju prijavu“ kako ne bi morao svaki put unositi podatke te na tome izgubiti dosta vremena. Prilikom prijave povjerava se ispravnost unesenih podataka (*e-mail* i lozinka) te ako je sve ispravno uneseno korisnik se prosljeđuje na daljnje korištenje aplikacije.

### **3.3. Unos visine, težine i broja godina korisnika**

Nakon što se korisnik prijavi, aplikacije od njega zahtjeva unos visine, težine i broj godina kako bi kasnije koristila te podatke u izračunu parametara bitnih za prevenciju pretilosti kojih se korisnik treba pridržavati. Navedeni podaci se mogu kasnije mijenjati pomoću opcije koja se nalazi u daljnjem radu aplikacije

### **3.4. Povezivanje s narukvicom Fitbit Charge HR**

Nakon što se korisnik prijavi i stvori svoj korisnički račun te unese svoju visinu, težinu i broj godina, slijedi povezivanje same aplikacije s narukvicom Fitbit Charge HR. To je ostvareno pomoću OAuth 2.0. standarda koji je opisan u poglavlju 4.1.5.

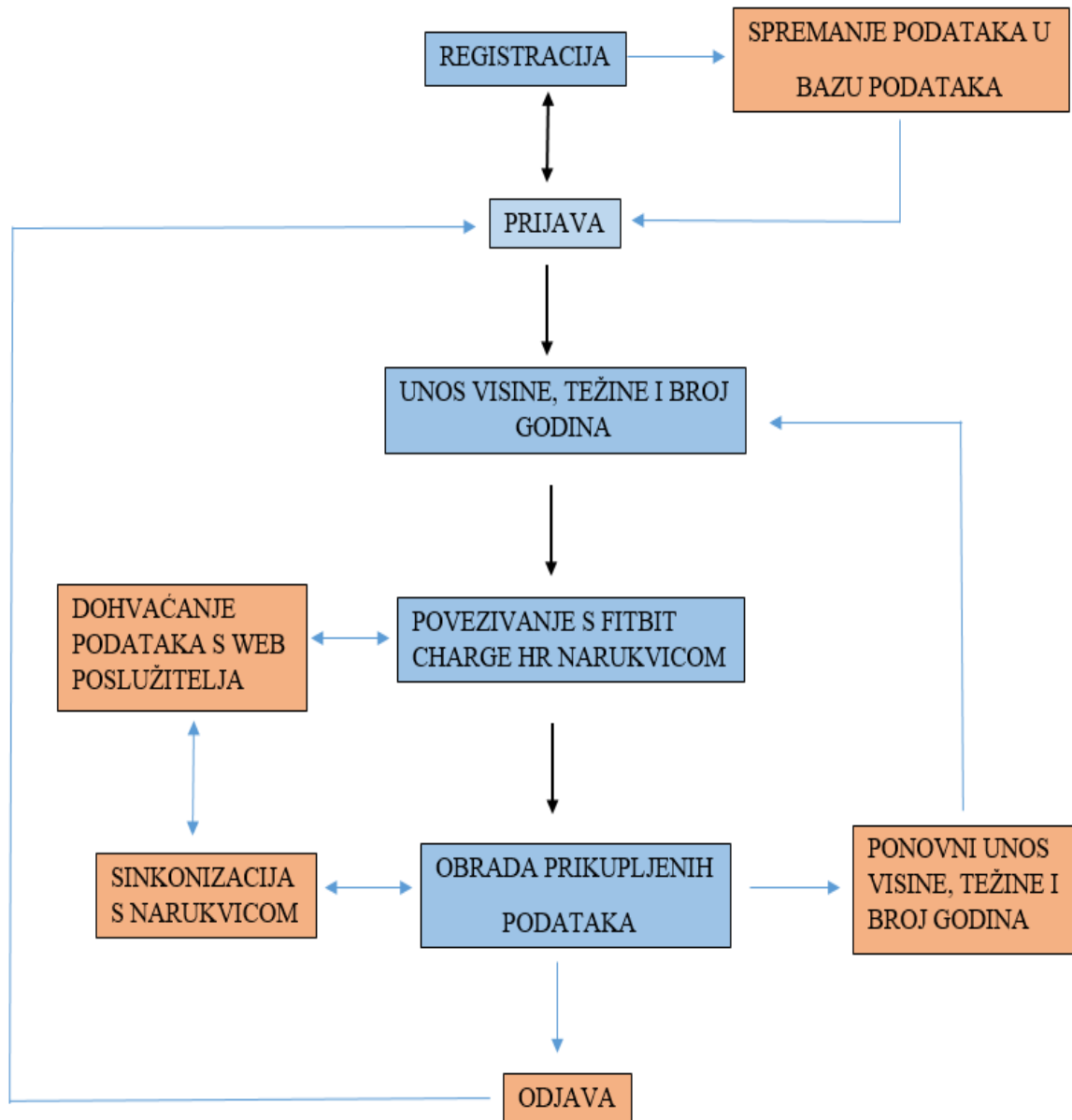
### **3.5. Obrada i analiza podataka prikupljenih s Fitbit Charge HR narukvice**

Nakon što se aplikacija poveže s Fitbit-ovim web poslužiteljem, dobiveni podaci se obrađuju te se korisniku pružaju preporučene vrijednosti za prevenciju pretilosti. Preporučeni se podaci kasnije uspoređuju s ostvarenim vrijednostima koje se prikupe uz pomoć narukvice Fitbit Charge HR te se na osnovu analize preporučenih i prikupljenih podataka alarmira korisnika o uspješnosti sprječavanja pretilosti.

### **3.6. Odjava korisnika**

Aplikacija sadrži i mogućnost odjave korisnika, pri čemu se zadržavaju svi uneseni podaci koji su vezani uz pojedini korisnički račun.

Navedeni model aplikacije za prevenciju pretilosti prikazan je na slici 3.1.



Sl. 3.1. Model aplikacije za prevenciju pretilosti

## **4. PROGRAMSKO RJEŠENJE APLIKACIJE ZA PREVENCIJU PRETILOSTI**

S obzirom na široku raširenost pretilosti u svijetu razvojem ove aplikacije omogućuje se kvalitetnije i bolje praćenje zdravstvenog stanja korisnika te poboljšava i olakšava prevencija pretilosti. Aplikacija, odnosno cjelokupni sustav se sastoji od Android mobilne aplikacije koja prikupljene podatke analizira i na temelju toga obavještava korisnika o radnjama koje treba poduzeti kako bi se spriječila pretilost, *fitness* narukvice Fitbit koja prikuplja podatke o otkucajima srca, broju prijeđenih koraka, broju prijeđenih kilometara i broju prijeđenih stepenica koje zatim prosljeđuje mobilnoj aplikaciji. Osim što aplikacija omogućuje bolje praćenje aktivnosti korisnika, također omogućuje smanjenje ručnog unosa podataka te se bazira na automatskom prikupljanju podataka pomoću Fitbit narukvice.

U ovom poglavlju prikazat će se cjelokupno programsko rješenje za izradu aplikacije za prevenciju pretilosti. Također su opisani korišteni alati i tehnologije potrebne za razvoj aplikacije, te je prikazano programsko rješenje po komponentama u kome su dani bitni kodovi za ostvarivanje razvoja aplikacije.

### **4.1. Opis alata i tehnologija**

Glavni dio aplikacije za prevenciju pretilosti razvijen je u integriranom razvojnom okruženju Android Studio koja se koristi za razvoj Android aplikacija. Prikupljanje podataka omogućeno je pomoću suvremene *fitness* narukvice koja s Android aplikacijom čini cjelokupni sustav za prevenciju pretilosti.

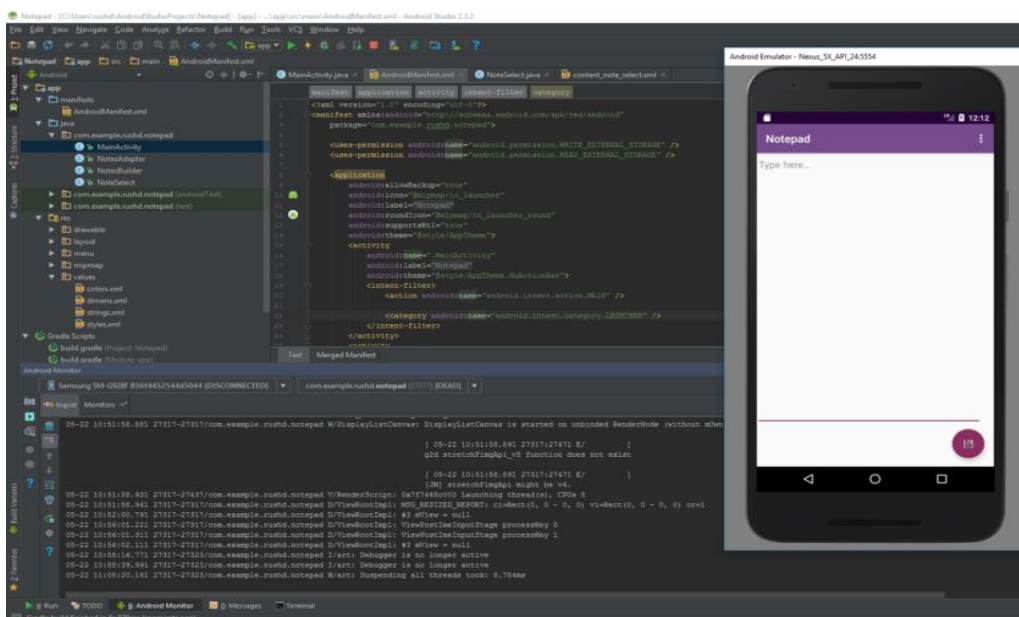
#### **4.1.1. Android Studio**

Android studio je službeno razvojno okruženje za Android operacijski sustav temeljenog na JetBrains IntelliJ IDEA programskoj podršci i namijenjen isključivo za razvoj Android aplikacija. Također se može koristiti i na ostalim operacijskim sustavima kao što su Linux i MacOS. Kako bi računalo moglo pokrenuti Android Studio mora imati karakteristike prikazane u tablici 4.1 [6].

Tab.4.1. Karakteristike računala potrebnih za pokretanje Android Studio 3.1.2.

<b>Operacijski sustav</b>	Microsoft Windows 7/8/10 (32-bit ili 64-bit), Mac OS X 10.10 (Yosemite) ili 10.13 (MacOS High Sierra) GNOME ili KDE desktop Linux (64 bit )(GNU C Library (glibc) 2.19+)
<b>RAM</b>	Minimalno 3 GB RAM ,preporučeno 8 GB RAM plus 1 GB za Android Emulator
<b>Veličina diska</b>	Minimalno 2GB ,preporučeno 4 GB
<b>Java</b>	Java Development Kit (JDK) 8
<b>Rezolucija monitora</b>	1280 × 800

Android Studio pruža korisniku sučelje za lakše razvijanje Android aplikacija. Programski jezik koji se koristi prilikom razvijanja aplikacija u Android Studio je Java, no sve veća je upotreba programskog jezika Kotlin. Android Studio također omogućava korisniku pristup Android SDK odnosno programskom razvojnom paketu koji omogućuje lakše pokretanje aplikacije na Android uređaju. Prilikom pokretanja aplikacije, Android Studio omogućuje korisniku pokretanje aplikacija na emulatoru tj. virtualnom stroju koji oponaša zadaću te ima većinu karakteristika stvarnog Android uređaja. Android Studio, također, olakšava testiranje i otkrivanje i ispravljanje pogrešaka nastalih prilikom razvijanja aplikacije. Jedno od korisnih obilježja je što Android Studio dodaje svoje prijedloge, izmjene i automatsko nadopunjavanje prilikom pisanja koda kako bi kod bio što ispravniji i pregledniji te također lakši za razvoj aplikacije. Nadalje, na slici 3.1 može se vidjeti izgled razvojnog okruženja Android Studio.



Sl. 4.1. Prikaz razvojnog okruženja Android Studio

### 4.1.2. IntelliJ IDEA

IntelliJ IDEA je Java razvojno okruženje za razvoj računalnih aplikacija koju je razvio JetBrains u siječnju 2001. IntelliJ IDEA se upotrebljuje za pisanje koda u Android Studio, ali i u mnogo drugim razvojnim okruženjima. Omogućuje nadopunjavanje koda analiziranjem sadržaja, ispravak koda te pružanje vlastitih prijedloga. Karakteristike koje su potrebne računalu za pokretanje IntelliJ IDEA prikazane su u tablici 4.2.

Tab.4.2. Karakteristike računala za pokretanje IntelliJ IDEA

<b>Operacijski sustav</b>	Microsoft Windows 7/8/10 (32-bit ili 64-bit), Mac OS X 10.08. ili novije inačice, GNOME ili KDE desktop Linux (64 bit)
<b>RAM</b>	Minimalno 1 GB RAM ,preporučeno 4 GB
<b>Veličina diska</b>	300 MB
<b>JDK verzija</b>	JDK 1.8
<b>Rezolucija monitora</b>	1024 ×768

### 4.1.3. Operacijski sustav Android

Prema [8], Android je mobilni operacijski sustav koji je implementiran u mobilnim uređajima i tabletima. Razvio ga je Android Inc. te ga je kasnije preuzeo Google. Bazira se na verziji Linux-a i danas je u širokoj uporabi. Također Google je razvio Android TV za pametne televizore, Android Auto za automobile i Wear OS za pametne satove. Budući da je Android operacijski sustava otvorenog koda programeri ga lako koriste za razvijanje različitih aplikacija što je jedna od prednosti Androida. Osim što svatko može prepravljati, doraditi i mijenjati kod, Android se može pokrenuti na raznim uređajima koji koriste Android OS. Neke od značajki koje Android pruža su:

- Korisničko sučelje (UI, engl. *user interface* )
- Povezivanje – Bluetooth,Bluetooth Low Energy, Wi-fi...
- Pohrana podataka - baze podataka kao što su SQLite, Firebase...
- *Multi-touch* - mogućnost prepoznavanja dva ili više dodira istovremeno
- *Multi-tasking* - obavljanje više radnji i procesa istovremeno

Za razvoj aplikacija za Android operacijski sustav koriste se brojna razvojna okruženja. Jedan od najzastupljenijeg razvojnog okruženja je Android Studio. Prilikom razvoja aplikacije potrebno je razlikovati četiri glavne komponente:

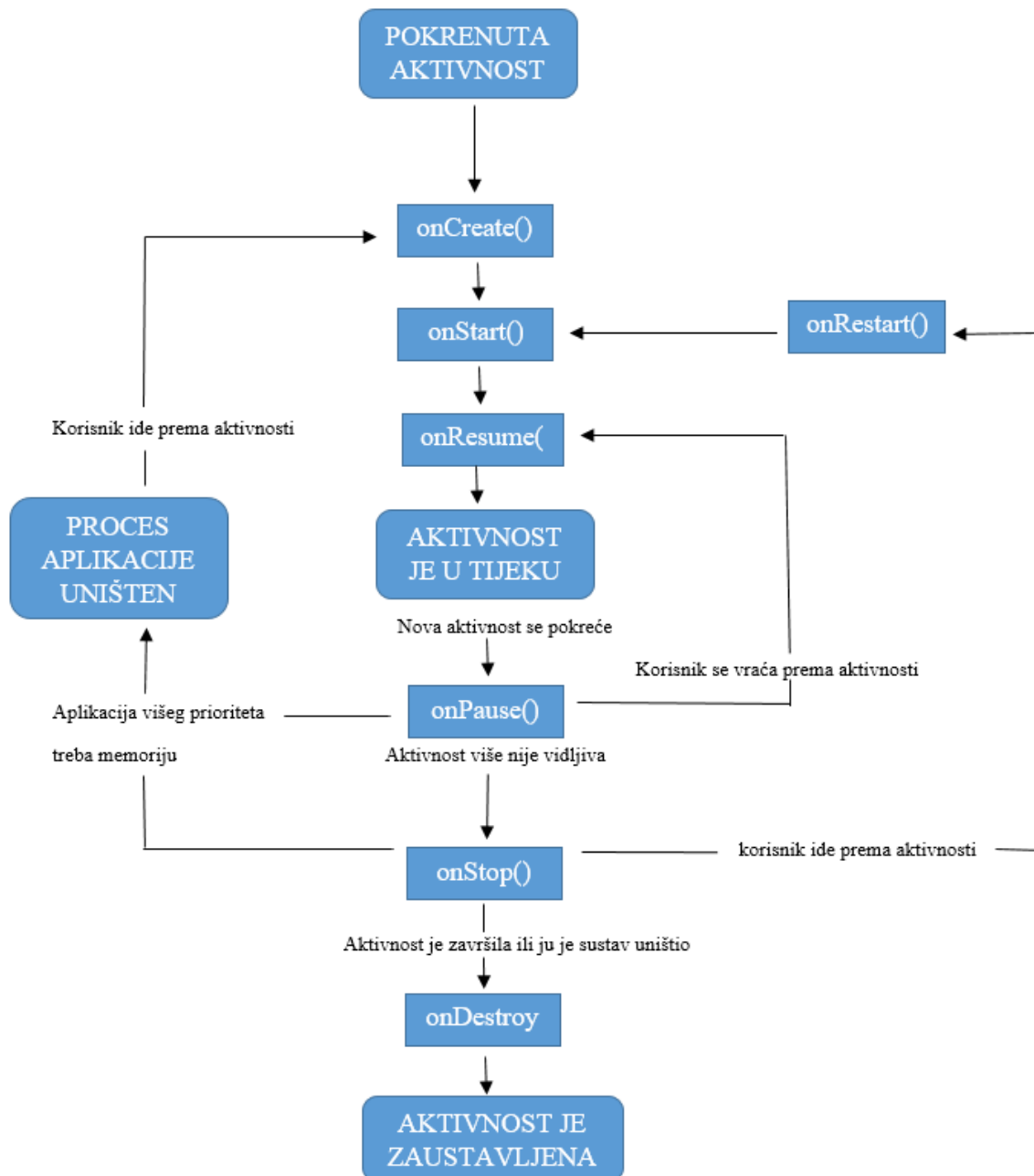
1. Aktivnost (engl. *Activity*) služi za interakciju s korisnicima. Predstavlja prozor (engl. *screen*) s korisničkim sučeljem. Aplikacija može sadržavati više aktivnosti koje nisu međusobno zavisne. Prikazuje ono što je korisniku trenutno važno, te brine o ispravnom radu aplikacije.
2. Usluge (engl. *Services*) služe za pokretanje aplikacije u pozadini zbog različitih slučajeva. Također služi za pokretanje glazbe u pozadini dok korisnik koristi neku drugu aplikaciju. Omogućuje slanje podataka bez prekidanja normalnog rada aplikacije itd.
3. Prijemnici emitiranja (engl. *Broadcast Receivers*) služe za odziv na odlazne poruke ili poruke primljene od strane nekih drugih aplikacija.
4. Pružatelji sadržaja (engl. *Content Provider*) služe za prijenos podataka između aplikacija te mogućnost spremanja tih podataka.

Svaka Android aplikacija ima svoj ciklus života koji čine sljedeće metode:

- *OnCreate()*
- *OnStart()*
- *OnResume()*
- *OnPause()*
- *OnStop()*
- *OnDestroy()*
- *OnRestart()*

Ove se metode pozivaju ovisno o stanju aplikacije te pružaju odgovarajući odziv. Životni ciklus aplikacije prikazan je slikom 4.2





Sl. 4.2. Prikaz životnog ciklusa aplikacije [6]

#### 4.1.4. Narukvica Fitbit Charge HR

Fitbit Charge HR [9] je prenosiva *fitness* narukvica koja pomoću senzora mjeri otkucaje srca, broj pređeni koraka i stepenica, izračunava potrošnju kalorija i prijeđenih kilometara. Služi za prikupljanje podataka o korisniku i njegovim dnevnim aktivnostima koje kasnije prosljeđuje mobilnoj aplikaciji pomoću OAuth 2.0 standarda. Preporučeno je nošenje narukvice što duže jer se samim time omogućuje dugotrajnije praćenje i prikupljanje parametara vezanih uz

korisnikov način života te se omogućava bolja prevencija korisnika od pretilosti. Na slici 4.3. prikazane su narukvice Fitbit Charge HR u različitim bojama.



Sl. 4.3. Prikaz narukvica Fitbit Charge HR

#### **4.1.5. Standard OAuth 2.0**

Cilj ovog standarda je sinkronizirati podatke između mobilne aplikacije i narukvice Fitbit Charge HR .

Prema [11], OAuth 2.0 standard danas je jedan od najzastupljenijih standarda koji se najčešće implementira u autorizacijskim okvirima. Može se koristiti na raznim sustavima i za različite svrhe. Omogućuje korisniku pristup s jednog mjesta podacima koji su spremljeni na nekom drugom mjestu. Postoje ključne stvari vezane za ovaj standard a to su zahtjevi, uloge i tzv. *tokeni*. Kako bi aplikacija pristupila podacima na web poslužitelju ona treba autorizaciju. Aplikacija šalje zahtjev web poslužitelju da želi pristupiti njegovim podacima. Poslužitelj može odobriti ili odbiti pristup. Također ako aplikacija želi pristupiti podacima mora imati određeni *token* koji određuje pravo pristupa nad podacima.

#### **4.2. Razvoj i način rada sustava za prevenciju pretilosti**

U ovom poglavlju je opisano cjelokupno programsko rješenje za razvoj sustava za prevenciju pretilosti. Opisano je povezivanje mobilne aplikacije s Fitbit narukvicom te izgled i funkcionalnost same Android mobilne aplikacije.

### 4.2.1. Prikaz programskog rješenja po komponentama

U ovom poglavlju prikazano je programsko rješenje razdvojeno po komponentama koje čine ovu aplikaciju. Također su prikazani kodovi koje se povezuju za pojedinu komponentu aplikacije.

#### 4.2.1.1. Postupak registracije i prijave korisnika

U ovom djelu je prikazano što se događa kada korisnik prvi put pristupi aplikaciji, koje korake mora učiniti i što sve mora zadovoljiti kako bi mogao dalje nastaviti koristiti aplikaciju. Ono što korisnik mora napraviti jeste prijaviti se (u ovom slučaju *Sign up*) pomoću korisničkog imena, *e-mail* adrese te lozinke. Prvo se provjerava da li je korisnik popunio sva polja s podacima pomoću funkcije *validate()* i ako je to zadovoljeno pritiskom na tipku „*Sign up*“ podaci korisnika spremaju se u bazu podataka pomoću funkcije *addUser()*. Kodovi koji su omogućili prijavu korisnika prikazani su na slici 4.4. i 4.5.

```
private Boolean validate() {
    Boolean result = false;

    username = etUsernameSignUpActivity.getText().toString();
    email = etEmailSignUpActivity.getText().toString();
    password = etPasswordSignUpActivity.getText().toString();
    confirmpassword = etConfirmPassword.getText().toString();

    if (username.isEmpty() || email.isEmpty() || password.isEmpty() || confirmpassword.isEmpty()) {
        Toast.makeText(context, this, text: "Please enter all the details", Toast.LENGTH_SHORT).show();
    }
    else if (!confirmpassword.equals(password)) {
        Toast.makeText(context, this, text: "Incorrect confirmed password", Toast.LENGTH_SHORT).show();
    } else {
        result = true;
    }
    return result;
}
```

Sl. 4.4. Prikaz koda funkcije *validate()* za registraciju korisnika

Slika prikazuje na koji način aplikacije provjerava jesu li svi podaci ispravno uneseni. Prvi uvjet je da korisnik mora unijeti sve podatke odnosno da ne smije niti jedno polje za unos podataka ostati prazno. Nadalje, provjerava da li je potvrda lozinke ispravna, odnosno da li je korisnik unio isto lozinku koju je unio prvi puta. Ukoliko je unos ispravan korisniku je omogućeno daljnje korištenje aplikacije, a ako nisu ispravno uneseni podaci korisniku se pruža određena

poruka ovisno o vrsti greške. Navedena funkcija `validate()` vraća `true` vrijednost ako je unos ispravan, odnosno ispisuje poruku ako unos nije ispravan.

Prilikom pokušaja registracije korisnika provjerava se postoji li već korisnik s unesenim *e-mail*-om i lozinkom. Ukoliko postoji korisnik dobiva poruku „*User already exists!*“. Ukoliko ne postoji unijeti *e-mail* i lozinka, podaci se spremaju u bazu podataka te korisnik dobiva poruku „*You are successfully signed up*“.

```
@Override
public void onClick(View v) {
    username = etUsernameSignUpActivity.getText().toString();
    email = etEmailSignUpActivity.getText().toString();
    password = etPasswordSignUpActivity.getText().toString();

    switch (v.getId()) {
        case R.id.button:
            if (validate()) {
                boolean adduser = false;
                int count = 0;
                User user = new User(username, email, password);
                Cursor cursor = userDbHelper.getUser();
                if (cursor.getCount() == 0) {
                    adduser = userDbHelper.addUser(user);
                } else {
                    while (cursor.moveToNext()) {
                        if (email.equals(cursor.getString( columnindex: 2)) || password.equals(cursor.getString( columnindex: 3))) {
                            count++;
                        }
                    }
                }
                if (count == 0) {
                    adduser = userDbHelper.addUser(user);
                } else {
                    Toast.makeText( context: this, text: "User already exists!", Toast.LENGTH_SHORT).show();
                    break;
                }

                if (adduser) {
                    Toast.makeText( context: this, text: "You are successfully signed up", Toast.LENGTH_SHORT).show();
                    PreferenceUtils.clearHeightPreference( context: SignUpActivity.this);
                    PreferenceUtils.clearCounterPreference( context: SignUpActivity.this);
                    startActivity(new Intent( packageContext: SignUpActivity.this, RegisterActivity.class));
                } else {
                    Toast.makeText( context: this, text: "Error", Toast.LENGTH_SHORT).show();
                }
            }
            break;
        case R.id.tvAlreadySignUp:
            startActivity(new Intent( packageContext: this, RegisterActivity.class));
            break;
    }
}
```

Sl. 4.5. Prikaz koda za registraciju korisnika

Nakon što je korisnik unio tražene podatke, može nastaviti koristiti aplikaciju.

Sljedeći put kad korisnik uđe u aplikaciju provjerava se da li je korisnik već prijavljen (u ovom slučaju *Log in*) i ako nije mora se prijaviti. Također je omogućena i opcija *sign up* ako korisnik prvi put koristi aplikaciju. U ovoj aktivnosti se također provjerava da li je korisnik popunio sva

polja s podacima pomoću te se povjerava jesu li ti podaci istovjetnima podacima s kojima se korisnik prvi put prijavio odnosno obavio *sign up* pomoću funkcije *validate()*. Nakon što je korisnik ispravno unio tražene podatke pomoću klika na tipku „Log in“ korisnik se uspješno prijavljuje te nastavlja daljnjim korištenjem aplikacije. Sve to je omogućeno kodom na slici 4.6. i na slici 4.7.

```
private boolean validate() {
    Boolean result = false;
    int size = 0;
    int countEmail = 0;
    int countPassword = 0;

    email = etEmail.getText().toString();
    password = etPassword.getText().toString();

    Cursor cu = userDbHelper.getUser();
    if (cu.getCount() == 0) {
        Toast.makeText(context, text: "You must first sign up!", Toast.LENGTH_SHORT).show();
    } else {
        while (cu.moveToNext()) {
            size++;
            if (email.equals(cu.getString(columnIndex: 2)) && password.equals(cu.getString(columnIndex: 3))) {
                result = true;
            }
            if (!email.equals(cu.getString(columnIndex: 2))) {
                countEmail++;
            }
            if (!password.equals(cu.getString(columnIndex: 3))) {
                countPassword++;
            }
        }
    }
    if (countEmail == size && countPassword == size) {
        Toast.makeText(context, text: "Incorrect email and password!", Toast.LENGTH_SHORT).show();
    } else if (countEmail == size || countEmail > countPassword) {
        Toast.makeText(context, text: "Incorrect email!", Toast.LENGTH_SHORT).show();
    } else if (countPassword == size || countEmail < countPassword) {
        Toast.makeText(context, text: "Incorrect password!", Toast.LENGTH_SHORT).show();
    }
    cu.close();
    return result;
}
```

Sl. 4.6. Prikaz koda funkcije *validate()* za prijavu korisnika

```

@Override
public void onClick(View v) {
    switch (v.getId()) {
        case R.id.btnSignUp:
            startActivity(new Intent( packageContext RegisterActivity.this, SignUpActivity.class));
            break;
        case R.id.btnLogIn:
            if (validate()) {
                if (checkBox.isChecked()) {
                    PreferenceUtils.saveEmail(email, context: this);
                    PreferenceUtils.savePassword(password, context: this);
                }
                startActivity(new Intent( packageContext RegisterActivity.this, UserBodyParameters.class));
            }
            break;
    }
}
}

```

Sl. 4.7. Prikaz koda za registraciju korisnika

#### 4.2.1.2. Rad s bazom podataka SQLite

Nakon što se korisnik uspješno prijavio njegovi podaci su spremljeni u bazu podataka na samom uređaju. Za rad s bazom podataka korištena je baza podataka SQLite. Implementiranje baze podataka SQLite unutar razvojnog okruženja Android Studio prikazan je kodom na slici 4.8.

```

public UserDbHelper(Context context) { super(context, DATABASE_NAME, factory: null, DATABASE_VERSION); }

@Override
public void onCreate(SQLiteDatabase db) {
    this.db = db;

    final String SQL_CREATE_USER_TABLE = "CREATE TABLE " +
        UserTable.TABLE_NAME + " ( " +
        UserTable._ID + " INTEGER PRIMARY KEY AUTOINCREMENT, " +
        UserTable.COLUMN_USERNAME + " TEXT, " +
        UserTable.COLUMN_EMAIL + " TEXT, " +
        UserTable.COLUMN_PASSWORD + " TEXT" +
        ")";

    db.execSQL(SQL_CREATE_USER_TABLE);
}

@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    db.execSQL("DROP TABLE IF EXISTS " + UserTable.TABLE_NAME);
    onCreate(db);
}

public boolean addUser(User user) {
    db = this.getWritableDatabase();

    ContentValues cv = new ContentValues();
    cv.put(UserTable.COLUMN_USERNAME, user.getUsername());
    cv.put(UserTable.COLUMN_EMAIL, user.getEmail());
    cv.put(UserTable.COLUMN_PASSWORD, user.getPassword());

    long result = db.insert(UserTable.TABLE_NAME, nullColumnHack: null, cv);
    if (result == -1) {
        return false;
    } else {
        return true;
    }
}
}

```

Sl. 4.8. Prikaz koda korištenog za implementiranje SQLite

Pomoću naredbe „CREATE TABLE“ SQLite stvara novu tablicu sa zadanim atributima koji su definirani u daljnjim naredbama. Također je implementirano i obnavljanje tablice u metodi onUpgrade(), kao i dodavanje potrebnih podataka u tablicu pomoću metode addUser() .

#### 4.2.1.3. Postupak unosa visine, težine i broja godina

Nakon što se korisnik prijavio, korisnik se prosljeđuje na aktivnost za unos visine, težine i broj godina čiji se podaci kasnije koriste za izračun potrebnih parametara za prevenciju pretilosti. Kod korišten za ostvarivanje unos korisnika prikazan je na slici 4.9.

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_user_body_parameters);
    ButterKnife.bind(target: this);

    if (PreferenceUtils.getHeight(context: this) != 0) {
        startActivity(new Intent(packageContext: this, CalculationActivity.class));
    }

    btnBodyParameters.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            if (etHeight.getText().toString().isEmpty() || etWeight.getText().toString().isEmpty() || etYears.getText().toString().isEmpty()) {
                Toast.makeText(context: UserBodyParameters.this, text: "Please enter all the details", Toast.LENGTH_SHORT).show();
            } else if (etHeight.getText().toString().equals("0") || etWeight.getText().toString().equals("0") || etYears.getText().toString().equals("0")) {
                Toast.makeText(context: UserBodyParameters.this, text: "Wrong entered value", Toast.LENGTH_SHORT).show();
            } else {
                weight = etWeight.getText().toString();
                float finalWeight = Float.parseFloat(weight);

                height = etHeight.getText().toString();
                float finalHeight = Float.parseFloat(height);

                years = etYears.getText().toString();
                int finalYears = Integer.parseInt(years);

                PreferenceUtils.saveWeight(finalWeight, context: UserBodyParameters.this);
                PreferenceUtils.saveHeight(finalHeight, context: UserBodyParameters.this);
                PreferenceUtils.saveYears(finalYears, context: UserBodyParameters.this);
                if (PreferenceUtils.getCounter(context: UserBodyParameters.this) == 0) {
                    startActivity(new Intent(packageContext: UserBodyParameters.this, ConnectFitbitActivity.class));
                } else {
                    startActivity(new Intent(packageContext: UserBodyParameters.this, CalculationActivity.class));
                }
                counter++;
                PreferenceUtils.saveCounter(counter, context: UserBodyParameters.this);
            }
        }
    });
}

```

#### Sl. 4.9. Prikaz koda za unos visine, težine i broj godina korisnika

Prilikom korisnikovog unosa navedenih podataka provjerava se ispravnost navedenih podataka te jesu li svi podaci uopće uneseni. Nakon što je sve ispravno uneseno podaci se spremaju u memoriju uređaja pomoću naredbe `PreferenceUtils.saveWeight` kojom se sprema težina korisnika dok se naredbom `PreferenceUtils.saveHeight` sprema visina korisnika i naredbom `PreferenceUtils.saveYears` sprema broj godina korisnika.



#### 4.2.1.4. Postupak povezivanja s narukvicom Fitbit Charge HR

Prilikom nošenja narukvice Fitbit Charge HR, prikupljaju se podaci vezani uz korisnikovu aktivnost tijekom dana. Ti podaci spremaju se na vanjsku bazu podataka koja se nalazi na Fitbit poslužitelju. Kako bi se ostvarila povezanost između aplikacije i baze podataka na Fitbit poslužitelju odnosno slanje HTTP zahtjeva prema Fitbit poslužitelju korišten je Google Volley biblioteka te je uz pomoć API razvojnog okruženja Postman-a kreiran *URL* na koji šaljemo zahtjev. Postupak se obavio pomoću OAuth 2.0 standarda. Sinkronizacija aplikacija s Fitbit Charge HR narukvicom omogućeno je kodom na slici 4.10.

```
public static void sync(final Context context, final TextViewSyncCallback TextViewSyncCallback) {
    DateFormat df = new SimpleDateFormat( pattern: "yyyy-MM-dd", Locale.getDefault());
    String date = df.format(Calendar.getInstance().getTime());

    String url = "https://api.fitbit.com/1/user/" + USER_ID + "/activities/date/" + date + ".json";

    JSONObjectRequest jsonObjectRequest = new JSONObjectRequest(Request.Method.GET, url, jsonObjectRequest: null, new Response.Listener<JSONObject>() {
        @Override
        public void onResponse(JSONObject response) {

            try {
                Log.e( tag: "Json object", String.valueOf(response));
                JSONObject obj = response.getJSONObject("summary");

                FitbitParameters.STEPS = obj.getInt( name: "steps");
                FitbitParameters.CALORIES = obj.getInt( name: "caloriesOut");

                TextViewSyncCallback.textViewCallback();

            } catch (JSONException e) {
                e.printStackTrace();
            }
        }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            Toast.makeText(context, text: "Something went wrong!", Toast.LENGTH_SHORT).show();
        }
    });

    @Override
    public Map<String, String> getHeaders() throws AuthFailureError {
        HashMap<String, String> headers = new HashMap<>();

        headers.put("Authorization", "Bearer " + Config.TOKEN);
        return headers;
    }
};

Singleton.getInstance(context).addToRequestQueue(jsonObjectRequest);
}
```

Sl. 4.10. Prikaz koda za sinkronizaciju aplikacije s Fitbit Charge HR narukvicom

Na početku metode postavlja se današnji datum koji se dohvaća pomoću naredbe `Calendar.getInstance().getTime()`. Taj podatak se koristi prilikom kreiranja *URL*-a na koji će aplikacija slati zahtjev za dohvaćanje podataka s Fitbit-ovog web poslužitelja. Nadalje, traženi podaci (broj koraka i broj kalorija) dohvaćaju se u obliku JSON objekta koji se pomoću naredbe `getIntent` pretvara u integer oblik kako bi se kasnije mogao obrađivati.

#### 4.2.1.4. Postupak obrade podataka prikupljenih s narukvice Fitbit Charge HR

Nakon što se podaci prikupe i ostvari se sinkronizacija s narukvicom Fitbit Charge HR počinje obrada prikupljenih podataka. Cilj obrade podataka je uvidjeti nedozvoljenu promjenu podataka unutar jednog dana kao što je npr. naglo smanjenje broja potrošenih kalorija te smanjenje broja prijedjenih koraka. Uočena promjena parametara bitnih za prevenciju pretilosti prosljeđuje se korisniku u obliku upozoravajućih poruka. Prvo se izračuna indeks tjelesne mase pomoću prije unesenih parametara visine i težine, te se zavisno o njemu računaju preporučene vrijednosti koraka, kilometara i kalorija koje korisnik treba ostvariti kako bi spriječio pretilost.

Obrada podataka ostvarena je pomoću koda na slici 4.11.

```
public void calculation() {

    AlertDialog.Builder alert = new AlertDialog.Builder( context: CalculationActivity.this);
    alert.setTitle("Pažnja!");
    alert.setPositiveButton( text: "OK", listener: null);

    float BMI = weight / (((height) / 100) * ((height) / 100));

    if (BMI > 25) {

        alert.setMessage(R.string.pazi_na_tezinu);
        alert.show();

        recommended_calories = 10 * weight + 6.25f * height - 5 * years + 5 + 700;
        recommended_kilometers = 0.414f * (height / 100000) * 15000;
        tvRecommendedCalories.setText(String.format(Locale.getDefault(), format: "%.0f", recommended_calories));
        tvRecommendedSteps.setText("15000");
        tvRecommendedKilometers.setText(String.format(Locale.getDefault(), format: "%.3f", recommended_kilometers));
    } else if (BMI < 18.5) {
        alert.setMessage(R.string.pazi_na_tezinu);
        alert.show();

        recommended_calories = 10 * weight + 6.25f * height - 5 * years + 5-400;
        recommended_kilometers = 0.414f * (height / 100000) * 2000;
        tvRecommendedCalories.setText(String.format(Locale.getDefault(), format: "%.0f", recommended_calories));
        tvRecommendedSteps.setText("2000");
        tvRecommendedKilometers.setText(String.format(Locale.getDefault(), format: "%.3f", recommended_kilometers));
    } else {
        alert.setMessage(R.string.tezine_je_dobra);
        alert.show();

        recommended_calories = 10 * weight + 6.25f * height - 5 * years + 5 + 400;
        recommended_kilometers = 0.414f * (height / 100000) * 10000;

        tvRecommendedCalories.setText(String.format(Locale.getDefault(), format: "%.0f", recommended_calories));
        tvRecommendedSteps.setText("10000");
        tvRecommendedKilometers.setText(String.format(Locale.getDefault(), format: "%.3f", recommended_kilometers));
    }
}
```

Sl. 4.11. Prikaz koda za obradu podataka

Nakon što se izračuna BMI, zavisno u kojoj skupinu korisnik pripada računaju se preporučene vrijednosti kalorija, kilometara i broj koraka koje korisnik mora ostvariti kako bi regulirao svoju tjelesnu težinu. Preporučena vrijednost kalorija računa se pomoću Mifflin-St Jeor formule:

$$10 \times \text{težina (kg)} + 6.25 \times \text{visina (cm)} - 5 \times \text{broj godina (god.)} + 5 \quad (4-1)$$

Broj koraka i kilometara se računa na osnovu optimalne vrijednosti broja koraka koja iznosi 10000 te ako je korisnik iznad granice normale (gleda se *BMI*) preporučeni broj koraka se povećava. Uz poznatu vrijednost broja koraka računa se broj kilometara prema sljedećoj formuli:

$$0.414 \times \left( \frac{\text{visina(cm)}}{100000} \right) \times \text{broj koraka} \quad (4-2)$$

Kasnije, korisnik može osvježiti podatke u aplikaciji tako da se ponovo poveže s Fitbit narukvicom te tako kontrolira preporučene i ostvarene vrijednosti parametara prikupljenih s Fitbit narukvice. Nakon što korisnik osvježi podatke i ponovo se poveže s Fitbit narukvicom, aplikacija ga alarmira o trenutnim vrijednostima te ako preporučene vrijednosti nisu ostvarene aplikacije pruža određenu poruku prema korisniku. Cijeli postupak ostvaren je kodom na slici 4.12.

```

public void syncFitbit() {
    FitbitUtils.sync( context this, new TextViewSyncCallback() {
        @Override
        public void textViewCallback() {
            float height = PreferenceUtils.getHeight( context CalculationActivity.this);

            tvCurrentCalories.setText(String.valueOf(FitbitParameters.CALORIES));
            tvCurrentSteps.setText(String.valueOf(FitbitParameters.STEPS));
            tvCurrentKilometers.setText(String.format(Locale.getDefault(), format "%.3f", ...args: 0.414f * (height / 100000) * FitbitParameters.STEPS));

            AlertDialog.Builder alert = new AlertDialog.Builder( context CalculationActivity.this);
            alert.setTitle("Upozorenje!");
            alert.setPositiveButton( text "OK", listener null);
            if (recommended_calories != FitbitParameters.CALORIES && (recommended_kilometers != 0.414f * (height / 100000) * FitbitParameters.STEPS)) {
                alert.setMessage("Preporučeni broj potrošenih kalorija i pređenih kilometara nije ostvaren!");
                alert.show();
            } else if (recommended_kilometers != 0.414f * (height / 100000) * FitbitParameters.STEPS) {
                alert.setMessage("Preporučeni broj kilometara i koraka nije ostvaren!");
                alert.show();
            } else {
                alert.setMessage("Preporučeni broj potrošenih kalorija nije ostvaren");
                alert.show();
            }
        }
    });
}

```

Sl. 4.12. Prikaz koda za sinkronizaciju s Fitbit narukvicom

## 5. PRIKAZ KORIŠTENJA I ISPITIVANJE RADA APLIKACIJE

U ovom djelu će se prikazati rad same aplikacije, njeno korištenje i funkcionalnost. Također će se prikazati sveukupni koraci koje korisnik prolazi prilikom rada aplikacije te mogućnosti koje aplikacija nudi. Nakon što korisnik instalira aplikaciju prvo što treba uraditi je obaviti registraciju, nakon toga se treba prijaviti te tek onda može početi koristiti ostale dijelove aplikacije. Nakon što se korisnik prijavi aplikacije od korisnika traži da unese svoju visinu, težinu i broj godina pomoću čega izračunava potrebne parametre za prevenciju pretilosti. Nakon toga slijedi sučelje unutar kojeg se korisnik povezuje s Fitbit Charge HR narukvicom odnosno povezuje se s Fitbit-ovim web poslužiteljem na kojem se nalaze podaci koje narukvica prikupi. Kada se ostvari komunikacije s web poslužiteljem korisnik se prosljeđuje na sučelje u kojem su prikazane preporučene i trenutne vrijednosti koraka, kalorija i kilometara koje je korisnik ostvario odnosno koje korisnik treba ostvariti kako bi spriječio pretilost, odnosno regulirao svoju tjelesnu masu. Također unutar tog sučelja korisnik ima omogućeno ponovo postavljanje svoje visine, težine i broj godina te se također može ponovo sinkronizirati s Fitbit narukvicom radi osvježavanja prikupljenih podataka nakon čega aplikacije alarmira korisnika o uspješnosti prevencije pretilosti. Također je omogućena i odjava korisnika kako bi se drugi korisnik prijavio na svoj korisnički račun.

Prilikom pokretanje aplikacije pojavljuje se tzv. Zaslone dobrodošlice (engl. *Welcome screen*) koji služi za osnovne informacije o nastanku aplikacije. Zaslone dobrodošlice prikazan je na slici 5.1.

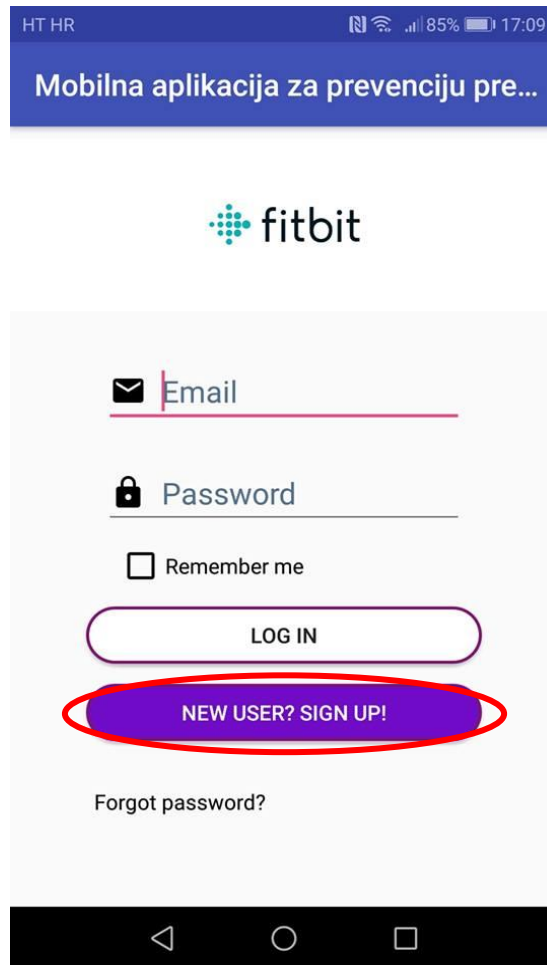


Sl. 5.1. Zaslون dobrodošlice

Na slici 5.1 vidi se FERIT-ova oznaka, izgled Fitbit Charge HR narukvice te tekst „Dobrodošli“ što korisniku pruža osnovne informacije o vrsti i nastanku aplikacije.

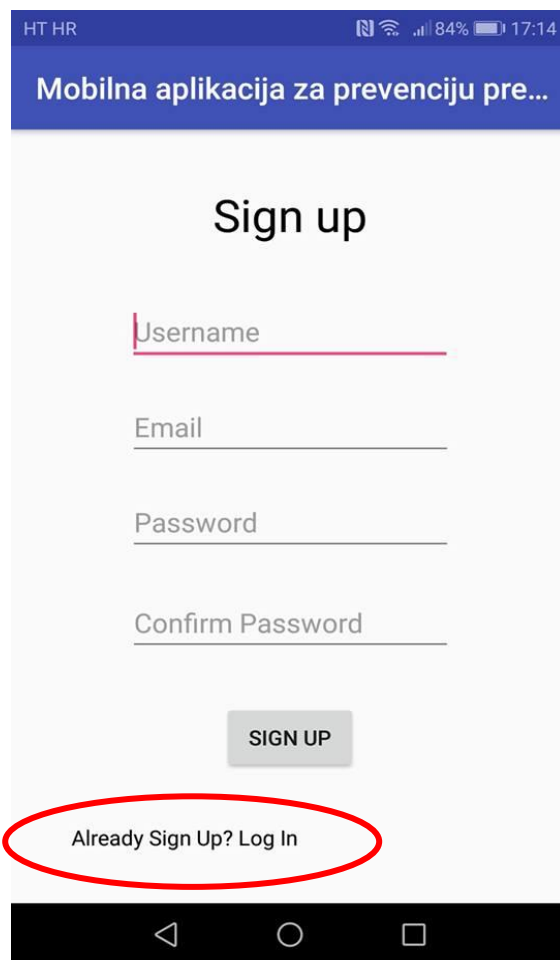
### **5.1. Prikaz registracije korisnika**

Kao što je već navedeno, prilikom prvog pokretanja aplikacije korisnik mora obaviti registraciju kako bi se njegovi podaci spremili u bazu podataka te je tako omogućeno lakše praćenje korisnikovih dnevnih aktivnosti. Korisnik se iz sučelja za prijavu pomoću pritiska na tipku prosljeđuje u sučelje za registraciju korisnika kako je i prikazano na slikama 5.2. i 5.3.



Sl. 5.2. Prikaz proces registraciju korisnika

Pritiskom na tipku označenog crvenom bojom na slici 5.2, aplikacije prosljeđuje korisnika na sučelje za unos podataka za registraciju koje je prikazano na slici 5.3.

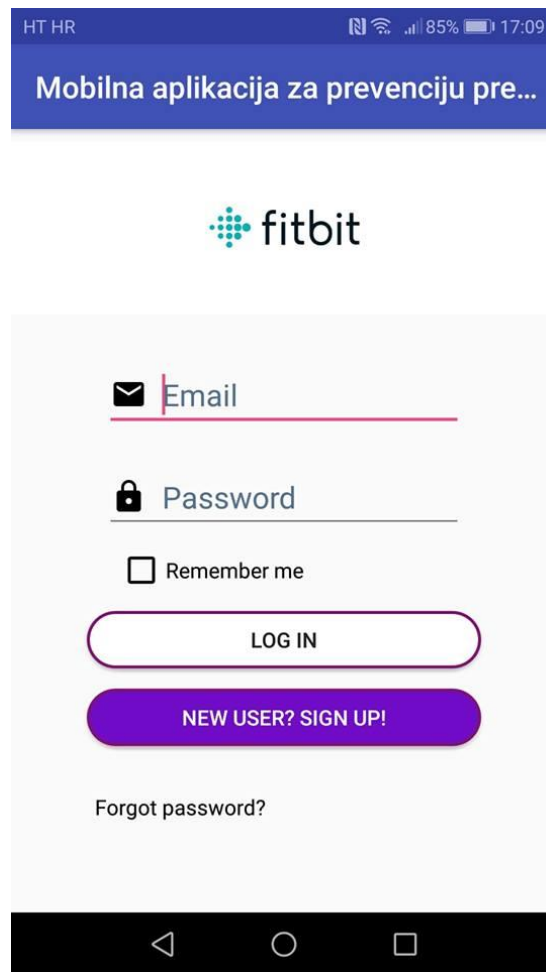


Sl. 5.3. Registracija korisnika

Prilikom registracije korisnik treba unijeti podatke kao što su korisničko ime, *e-mail* i lozinka te potvrditi zadanu lozinku. Nakon pritiske tipke „SIGN UP“ podaci o korisniku se spremaju u bazu podataka te se korisnik prosljeđuje nazad na sučelje za prijavu kako bi se prijavio u aplikaciju i nastavio daljnje korištenje. Nadalje, ako je korisnik već obavio registraciju može se vratiti na sučelje za prijavu klikom na dio označen crvenom bojom na slici 5.3.

## 5.2. Prikaz prijave korisnika

Nakon što korisnik obavi registraciju, treba se prijaviti odnosno obaviti *Log in*. To se izvršava tako da korisnik unese podatke o e-mail-u i lozinki pomoću kojih je obavio registraciju. Nakon toga aplikacija povjerava ispravnost unesenih podataka te ako je sve ispravno uneseno korisniku je dopušteno daljnje korištenje aplikacije. Također je omogućeno korisniku da aplikacija zapamti njegovu prijavu kako se korisnik ne bi morao prijavljivati svaki put kad ulazi u aplikaciju. Na ovom sučelju je prikazan i službeni Fitbit-ov logo te je također omogućen oporavak lozinke ako ju je korisnik zaboravio. Prijava korisnika prikazano je na slici 5.4.

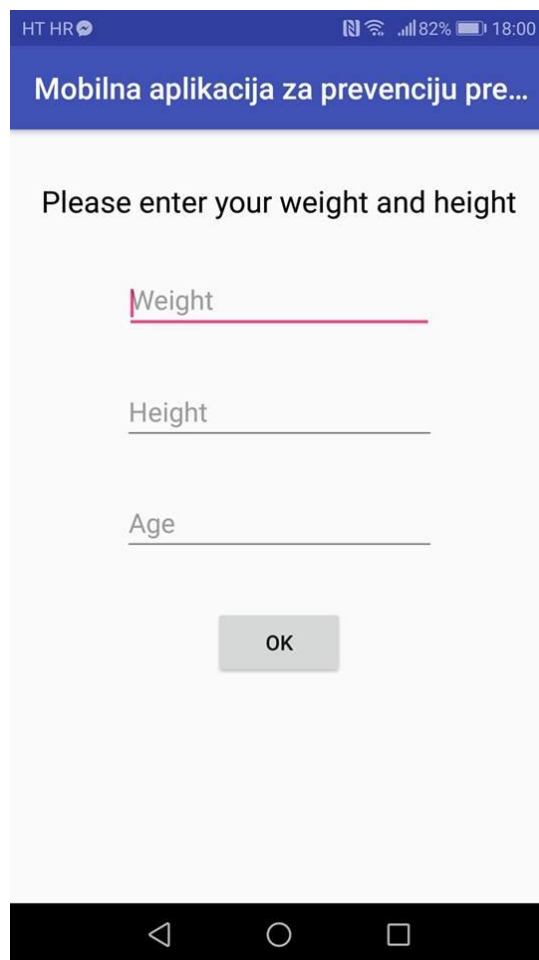


Sl. 5.4. Prikaz prijave korisnika

### 5.3. Prikaz unosa visine, težine i broja godina korisnika

Nakon što se korisnik uspješno prijavio slijedi unos visine, težine i broj godina korisnika. Navedeni podaci služe za kasniju obradu podataka i izračunavanje potrebnih parametara za prevenciju pretilosti. Korisnik prilikom unosa mora unijeti sve podatke kako bi mogao nastaviti s daljnjim korištenjem aplikacije. Prilikom unosa korisniku se pružaju smjernice koju vrijednost treba unijeti. Navedeno sučelje prikazano je na slici 5.5.



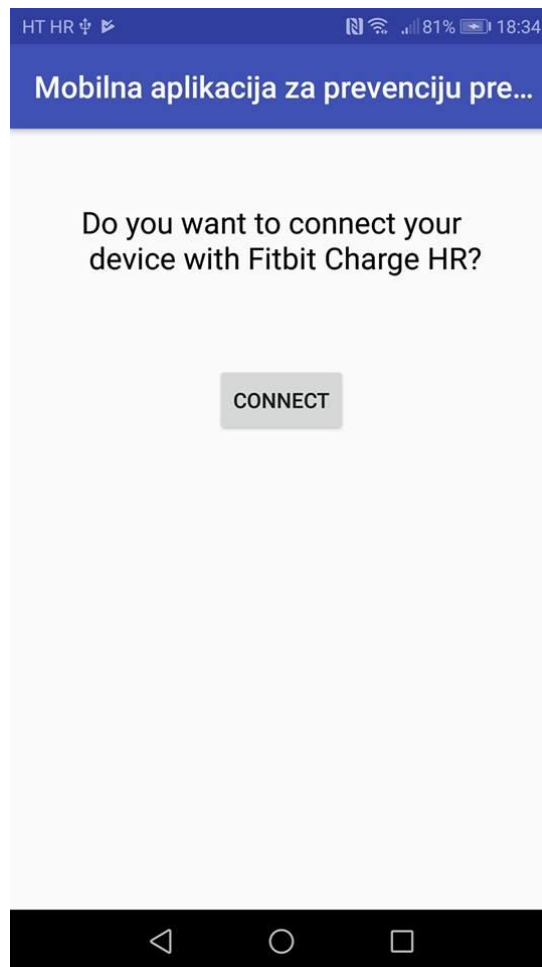


Sl.5.5. Prikaz unosa visine, težine i broja godina korisnika

Nakon što su uneseni svi podaci, provjerava se ispravnost unesenih podataka te pritiskom na tipku „OK“ korisnika se prosljeđuje u sljedeće sučelje.

#### **5.4. Prikaz povezivanja s Fitbit Charge HR narukvicom**

Nakon što je korisnik unio visinu, težinu i broj godina korisnik se prosljeđuje u sučelje za povezivanje s Fitbit narukvicom. Pri ulasku u navedeno sučelje korisnika se pita želi li se povezati s narukvicom Fitbit Charge HR te je korisniku ponuđena tipka „CONNECT“ pomoću koje se može povezati s Fitbit Charge HR narukvicom. Navedeno sučelje prikazano je na slici 5.6.



Sl. 5.6. Prikaz sučelja za povezivanje s Fitbit Charge HR narukvicom

Klikom na tipku „CONNECT“ aplikacije se povezuje odnosno traži zahtjev za pristup na Fitbit-ov web poslužitelj koji pomoću *tokena* koju aplikacija sadrži određuje pravo pristupa odnosno određuje koje podatke može pružiti aplikaciji na korištenje. *Token* se dobije preko aplikacije Postman. Nakon što utvrdi pravo pristupa, web poslužitelj pruža aplikaciji tražene podatke koji se onda dalje koriste.

## 5.5. Prikaz obrade podataka prikupljenih s narukvice Fitbit Charge HR

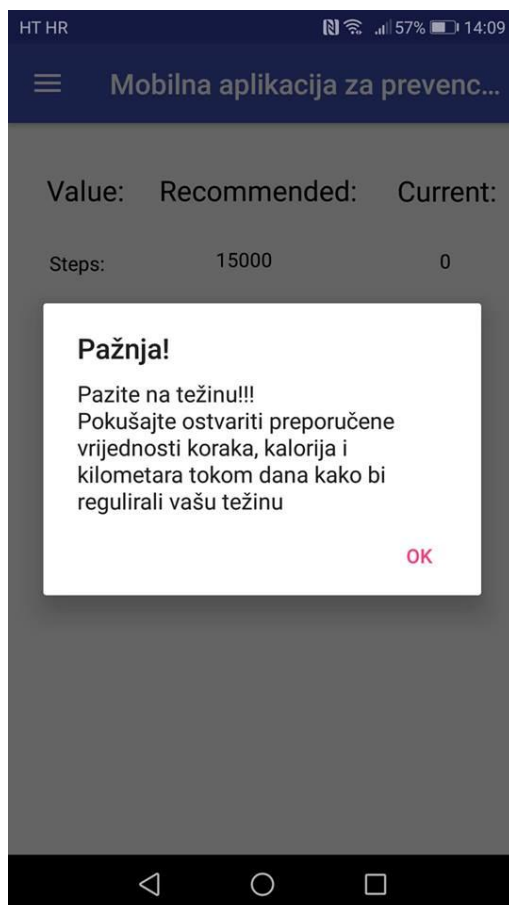
Nakon što se aplikacije poveže s Fitbit Charge HR narukvicom, pokreće se sučelje za obradu prikupljenih podataka s narukvice Fitbit Charge HR. Na tom sučelju prikazuju se trenutne i preporučene vrijednosti koje korisnik mora ostvariti kako bi spriječio pretilost. Prilikom ulaska u navedeno sučelje korisniku se pruža poruka ovisno o njegovom indeksu tjelesne mase. Ukoliko korisnik ima indeks tjelesne mase u granici normale ( $18,5 < \text{BMI} < 25$ ) onda mu se

pruža poruka koja je prikazana na slici 5.7. No, ako korisnik ima indeks tjelesne mase iznad ili ispod granice normale na sučelju se prikazuje poruka prikazana na slici 5.8.



Sl. 5.7. Prikaz poruke kada je indeks tjelesne mase unutar granice normale

Na slici je prikazana tzv. „pop up“ poruka koja obavještava korisnika da ima težinu u granici normale. Poruka ga savjetuje da se nastavi pridržavati preporučenih vrijednosti koraka, kalorija i kilometara kako bi regulirao svoju težinu i spriječio pretilost. Pritiskom na tipku „OK“ korisnik vraća povratnu informaciju aplikaciji potvrđujući da je pročitao poruku te mu je omogućeno daljnje korištenje aplikacije.



Sl. 5.8. Prikaz poruke kada je indeks tjelesne mase iznad ili ispod granice normale

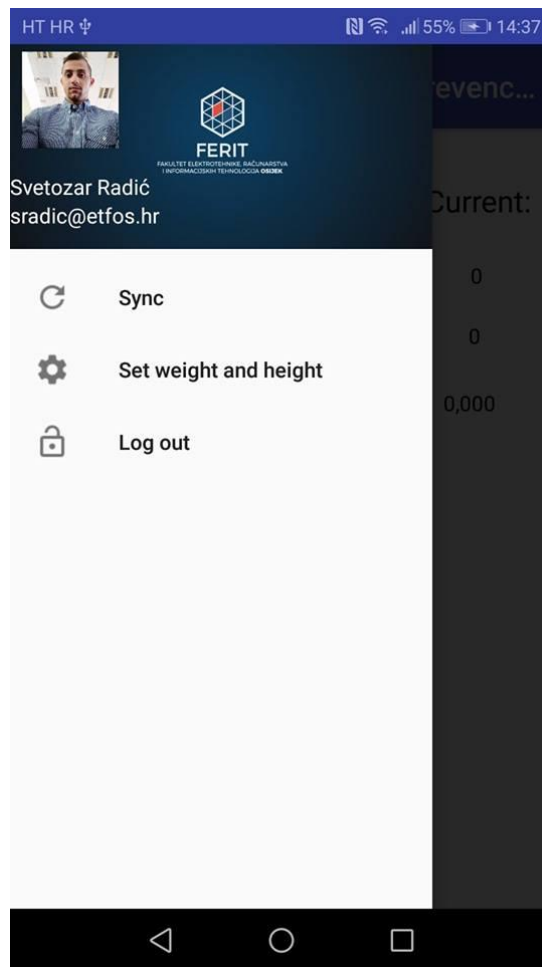
Na slici je prikazana tzv. „pop up“ poruka koja obavještava korisnika da pripazi na težinu. Poruka ga savjetuje da se pridržava preporučenih vrijednosti koraka, kalorija i kilometara kako bi regulirao svoju težinu i spriječio pretilost. Pritiskom na tipku „OK“ korisnik vraća povratnu informaciju aplikaciji potvrđujući da je pročitao poruku te mu je omogućeno daljnje korištenje aplikacije.

Nakon što korisnik potvrdi navedenu poruku prikazuje se sučelje s trenutnim i preporučenim vrijednostima koraka, kalorija i kilometara koji su bitni za korisnika. Navedeno sučelje prikazano je na slici 5.9.

Value:	Recommended:	Current:
Steps:	15000	0
Calories:	2538	1004
Kilometers:	9,315	0,000

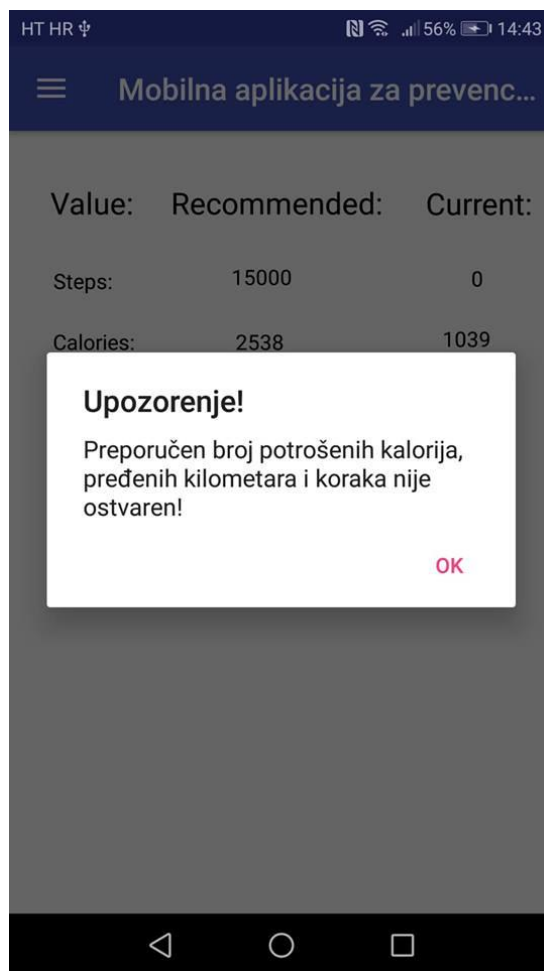
Sl. 5.9. Prikaz sučelja za obradu podataka

U ovom sučelju nalazi se izbornik (engl. *Navigator Drawer*) pomoću kojega se korisnik može ponovo sinkronizirati s Fitbit Charge HR narukvicom, ponovo postaviti svoju visinu, težinu i broj godina te se može odjaviti sa svog korisničkog računa. Budući da se visina, težina i broj godina korisnika svakodnevno mijenjaju ovaj izbornik olakšava praćenje korisnikovih tjelesnih promjena. Unutar izbornika nalazi se i profilna slika osobe koja je napravila aplikaciju te njegovi kontakt podaci. Navedeni izbornik prikazan je na slici 5.10.



Sl. 5.10. Prikaz izbornika

Nakon što korisnik stisne tipku „sync“ aplikacija se ponovo sinkronizira s Fitbit narukvicom te dohvati trenutne podatke s web poslužitelja prikupljenih korištenjem narukvice Fitbit Charge HR. Podaci prikupljeni s web poslužitelja uspoređuju se s preporučenim vrijednostima te se na osnovu toga alarmira korisnika o dosadašnjem rezultatu ostvarenih vrijednosti, kao što je prikazano na slici 5.11.



Sl. 5.11. Prikaz alarmiranja korisnika

Pritiskom na tipku „OK“ korisnik može nastaviti dalje koristiti aplikaciju, tj. vidjeti trenutne vrijednosti koje je ostvario te pokušati ostvariti preporučene vrijednosti.

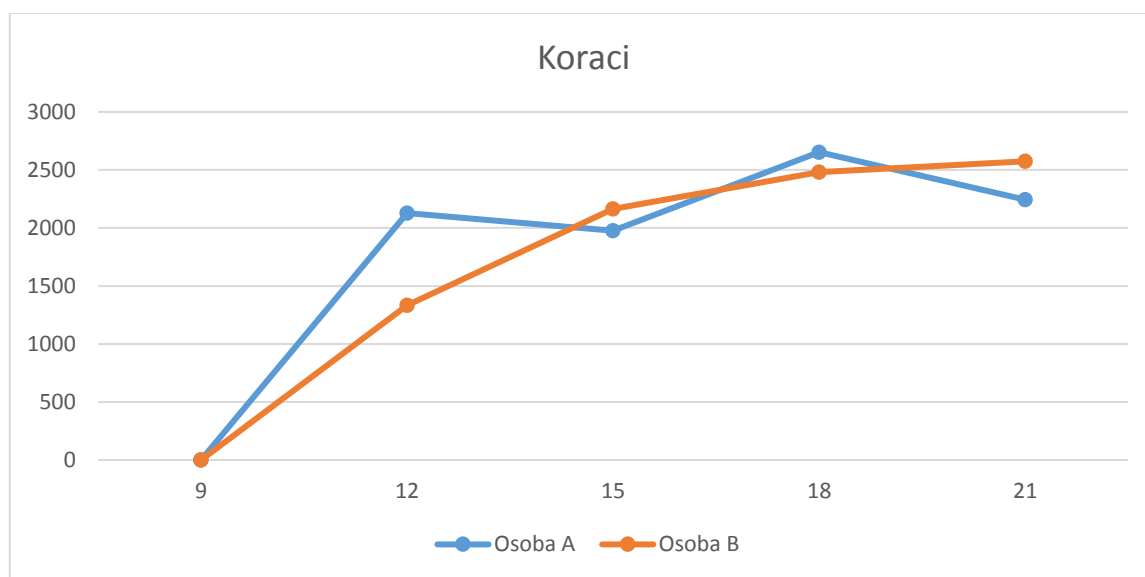
## 5.6. Prikaz rada sustava na primjeru

Rad sustava za prevenciju pretilosti prikupljanjem parametara o životnim navikama testirana je na dvije osobe (osoba A i osoba B). Osoba A je osoba s 172 cm visine, 72 kg i ima 21 godinu što znači da ima normalan indeks tjelesne mase, dok je osoba B osoba 180 cm visine, 89 kg i ima 22 godine što znači da ima povišeni indeks tjelesne mase. Obje osobe su nosile narukvice Fitbit Charge HR kroz 12 sati uzimajući u obzira da su osobe počele nositi narukvicu u 9 sati prijepodne, te su rezultati bili bilježeni svakih 3 sata. Prema izrazima (4-1) i (4-2) preporučene vrijednosti koraka, kalorija i kilometara za osobu A i osobu B prikazane su u tablici 5.1

Tab.5.1. Prikaz preporučenih vrijednosti koraka, kalorija i kilometara za osobe A i B

Vrijednost	Koraci	Kalorije	Kilometri
Osoba A	10000	2095	7.121
Osoba B	13000	2410	9.178

Tijekom praćenja rezultata unutar 12 sati osobe su postizala različite vrijednosti koraka, kalorija i kilometara koji su analitički prikazani na slikama 5.12, 5.13 i 5.14.

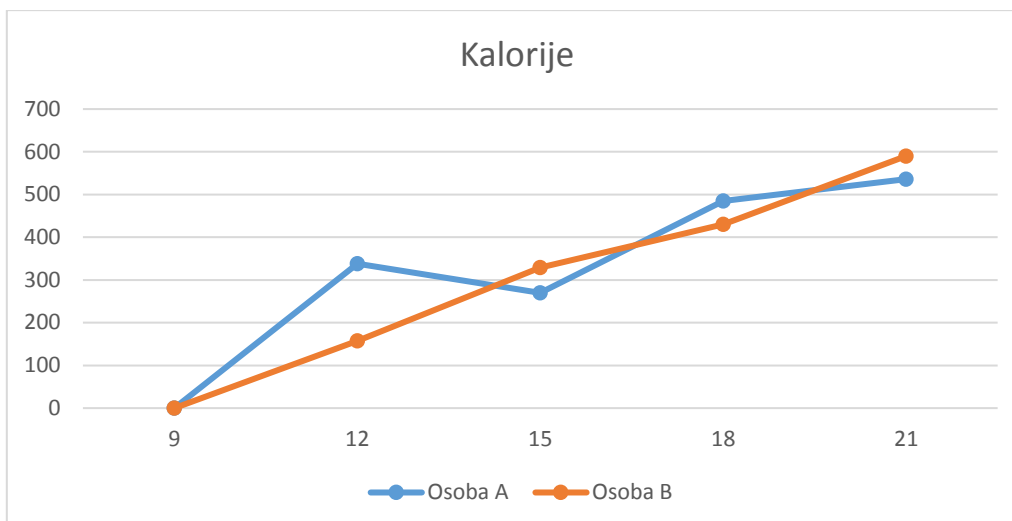


Sl. 5.12. Prikaz vrijednosti prijeđenih koraka tijekom 12 sati osobe A i osobe B

Prema slici 5.12, vidljivo je da na početku mjerenja osoba A, tj. osoba s normalnim indeksom tjelesne mase napravi veći broj koraka od osobe B, tj. osobe s povišenim indeksom tjelesne mase. No, tijekom dana korištenjem aplikacije za prevenciju pretilosti preporukama se uspjelo motivirati i potaknuti osobu B za povećanjem tjelesne aktivnosti. Samim se time osoba B uspjela približiti preporučenoj vrijednosti koraka čime je bitno utjecala na regulaciju svoje tjelesne mase i prevenciji pretilosti. Osoba A je korištenjem aplikacije uspjela ostvariti ciljni broj koraka čime je očuvala normalni indeks tjelesne mase.

Nadalje, mjerena je i vrijednost kalorija koje su osobe potrošile tijekom 12 sati te su rezultati bilježeni svakih 3 sata. Prikupljeni rezultati analitički su prikazani na slici 5.13.

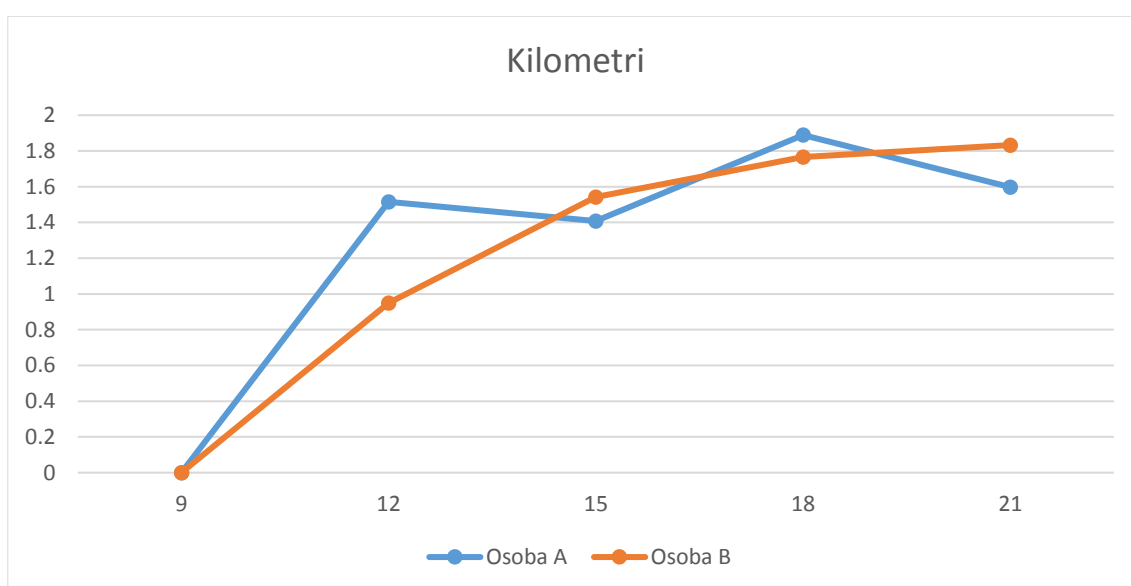




Sl. 5.13. Prikaz vrijednosti potrošenih kalorija tijekom 12 sati osobe A i osobe B

Prema slici 5.13, osoba A je u prvih tri sata od trenutka mjerenja potrošila znatno više kalorije od osobe B. Posljedica toga je povišenje indeksa tjelesne mase kod osobe B i mogući uzrok pretilosti. Nadalje, u sljedeća tri sata osoba B je potaknuta putem preporuka u aplikaciji na veću tjelesnu aktivnost uz povećanu potrošnju kalorija, dok je osoba A smanjila potrošnju kalorija. Osoba B je u narednih šest sati povećavala potrošnju kalorija što dovodi do regulacije indeksa tjelesne mase odnosno smanjenje rizika od pretilosti. Zadnjih par sati osoba A i osoba B su imali podjednak utrošak kalorija čime je pokazana učinkovitost aplikacije.

Posljednja vrijednost koja je mjerena je broj prijeđenih kilometara koje su ostvarile osoba A i osoba B tokom 12 sati. Prikupljeni rezultati analitički su prikazani na slici 5.14.



Sl. 5.14. Prikaz vrijednosti prijeđenih kilometara tijekom 12 sati osobe A i osobe B

Prema slici 5.14, osoba A je u početnih par sati bila tjelesni aktivnija od osobe B i prešla je veći broj kilometara od osobe B. U narednih par sati osoba B povećava svoju aktivnost uzrokovanu preporukama aplikacije koje su potaknule osobu B na veću aktivnost. Posljedica povećane tjelesne aktivnosti dovodi do regulacije indeksa tjelesne mase što pridonosi pri prevenciji pretilosti.

Analizom slika 5.12, 5.13 i 5.14 prikazano je kako aplikacija utječe na sprječavanje pretilosti. Osoba B u početku imala je puno manje vrijednosti prijeđenih koraka, potrošenih kalorija i prijeđenih kilometara nego osoba A. Uz korištenje aplikacije i preporuka unutar aplikacije osoba B je potaknuta na veću tjelesnu aktivnost što pridonosi sprječavanju pretilosti. Osoba A je korištenjem aplikacije uspjela ostvariti preporučene vrijednosti čime je održala svoj indeks tjelesne mase i pridonijela prevenciji pretilosti.

## 6. ZAKLJUČAK

U ovom radu razvijen je sustav za prevenciju pretilosti koji se sastoji od mobilne aplikacije te Fitbit narukvice. Sustav je razvijen u svrhu lakšeg i boljeg praćenje dnevnih aktivnosti korisnike s ciljem alarmiranja i upozoravanja korisnika u situacijama u kojima se ponaša potičući nastanak pretilosti, odnosno u situacijama u kojima bi njegova tjelesna težina krenula prelaziti granice normale. Tijekom dana, korisnik nosi narukvicu Fitbit Charge HR koja cijelo vrijeme prikuplja podatke o korisnikovim aktivnostima te ih šalje na web poslužitelj. Nakon što se aplikacija pokrene kreće povezivanje s Fitbit narukvicom te dohvaćanje prikupljenih podataka s web poslužitelja. Nakon toga aplikacija analizira i obrađuje podatke na osnovu čega prosljeđuje odgovarajuću poruku korisniku. Jednostavno korištenje aplikacije omogućuje korisniku prijavu, unos vlastitih dimenzija, stalno praćenje dnevnih aktivnosti te pomoć aplikacije pri ostvarivanju potrebnih vrijednosti za prevenciju pretilosti kroz alarmiranje korisnika na obraćanje pažnje na ostvarene rezultate. Sve to olakšava korisniku brigu o vlastitom zdravlju i praćenju vlastitih dnevnih aktivnosti. Ipak, ova aplikacija nije jedino i cjelovito rješenje za prevenciju pretilosti, već su tu ključni i brojni drugi čimbenici kao što je regulirana i ispravna prehrana, psihička ravnoteža i okoliš. Jedan od značajnih čimbenika pri prevenciji pretilosti je i fizička aktivnost, koja se manifestira kroz svakodnevne aktivnosti koja ova aplikacija prati. U raznim istraživanjima pretilost je prisutna kod raznih dobni skupina, stoga se ova aplikacija može koristiti od adolescentske dobi u kojoj se najčešće pojavljuje pretilost pa sve do zrele dobi kod ljudi koji se jednostavno nemaju vremena posvetiti praćenju svojih aktivnosti bilo zbog posla ili nečeg drugog. Kao rezultat navedenog aplikacija je pogodna za sve dobne skupine koje žele brinuti o svom zdravlju.

Aplikacija je razvijena u razvojnom okruženju Android Studio. Tijekom razvoja aplikacije korišten je programski jezik Java. Aplikacije je testirana na inačici Androida 8.0.0.

Ispitivanje rada aplikacije pokazuje učinkovitost i ispravnost u njenom radu što je vidljivo prema poglavlju 5.6. Iz navedenih rezultata zaključeno je da se osoba s višim indeksom tjelesne mase uspjela tjelesno aktivirati čime je potaknula prevenciju pretilosti što je bio i cilj ovog rada. Osoba koja je imala indeks tjelesne mase unutar normalnih vrijednosti je uspjela ostvariti ciljni broj parametar bilježenih putem narukvice Fitbit Charge HR čime je pridonijela prevenciji pretilosti. Nadalje, tijekom testiranja aplikacije uočene su mogućnosti za njeno poboljšanje u radu i alarmiranju korisnika čime bi se poboljšale usluge i način korištenja aplikacije.

## LITERATURA

- [1] Obesity, data and statistics, Svjetska zdravstvena organizacija (WHO, World Health Organization), <http://www.euro.who.int/en/health-topics/noncommunicable-diseases/obesity/data-and-statistics>, pristupljeno: 15.06.2018.
- [2] Pretilost, PLIVAzdravlje, <https://www.plivazdravlje.hr/bolest-clanak/bolest/51/Pretilost.html>, pristupljeno: 15.06.2018.
- [3] Ö. Taçyıldız, D. Ç. Ertuğrul, Y. Bitirim, N. Akcan, A. Elçi, Ontology-Based Obesity Tracking System for Children and Adolescents, 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), 22 June 2018, Tokyo, Japan, <https://ieeexplore.ieee.org/document/8377880/>, pristupljeno: 15.06.2018.
- [4] S. Chinchole, S. Patel, Cloud and sensors based obesity monitoring system, 2017 International Conference on Intelligent Sustainable Systems (ICISS), 21 June 2018, Palladam, India, <https://ieeexplore.ieee.org/document/8389388/>, pristupljeno:15.06.2018.
- [5] K. Feldman, M. Duarte, W. Mikels-Carrasco, N. V. Chawla, Leveraging health and wellness platforms to understand childhood obesity: A usability pilot of FitSpace, 2018 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI), 09 April 2018, Las Vegas, NV, USA, <https://ieeexplore.ieee.org/document/8333457/>, pristupljeno: 15.06.2108.
- [6] Meet Android Studio, Android Developers, <https://developer.android.com/studio/intro/>, pristupljeno: 16.06.2018.
- [7] N. Dorosh, H. Kuchmiy, O. Boyko, O. Dorosh, O. Stepanjuk, N.Maritz, Development the Software Applications for Mobile Medical System Based on OS Android, 13th International Conference on Modern Problems of Radio Engineering, Telecommunications and Computer Science (TCSET), Lviv, Ukraine, 23 Feb. 2016, pp. 808–810.
- [8] Application Fundamentals, Android Developers, <https://developer.android.com/guide/components/fundamentals>, pristupljeno: 16.06.2018.

- [9] J. Dahmen, A. La Fleur, G. Sprint, D. Cook, D.L. Weeks, Using Wrist-Worn Sensors to Measure and Compare Physical Activity Changes for Patients Undergoing Rehabilitation, The Third IEEE International Workshop on Sensing Systems and Applications Using Wrist Worn Smart Devices, Kona, Hawaii, USA, 13 March 2017, pp. 667–672.
- [10] Product Manual, Fitbit,  
[https://staticcs.fitbit.com/content/assets/help/manuals/manual\\_charge\\_hr\\_en\\_US.pdf](https://staticcs.fitbit.com/content/assets/help/manuals/manual_charge_hr_en_US.pdf),  
pristupljeno: 20.06.2018.
- [11] Marwan Darwish, Abdelkader Ouda, Evaluation of an OAuth 2.0 Protocol Implementation for Web Server Applications, Department of Electrical and Computer Engineering, 2015 International Conference and Workshop on Computing and Communication (IEMCON), The University of Western Ontario, London, Kanada, 15-17 Oct. 2015, pp. 1-4.
- [12] Additional Fitbit OAuth 2.0 Features,  
<https://dev.fitbit.com/build/reference/web-api/oauth2-additions/>, pristupljeno 21.06.2018.
- [13] Design Patterns, Sourcemaking,  
[https://sourcemaking.com/design\\_patterns](https://sourcemaking.com/design_patterns), pristupljeno:22.06.2018.
- [14] Uvod u butterknife, <http://jakewharton.github.io/butterknife/>, pristupljeno:25.06.2018.
- [15] Android SQLite Database Tutorial, Android Hive,  
<https://www.androidhive.info/2011/11/android-sqlite-database-tutorial/>,  
pristupljeno: 27.06.2017.
- [16] Activity and Exercise Logs, Fitbit dev,  
<https://dev.fitbit.com/build/reference/web-api/activity/>, pristupljeno: 10.07.2018.
- [17] Fitbit Community, <https://community.fitbit.com/t5/Web-API-Development/Fitbit-device-Integration-with-Android-app/td-p/1007656>, pristupljeno: 10.07.2018.
- [18] Android application integration with Fitbit ( Fitbit Tracker ),  
<https://stackoverflow.com/questions/28364202/android-application-integration-with-fitbit-fitbit-tracker>, pristupljeno: 10.07.2018.
- [19] Volley overview, <https://developer.android.com/training/volley/>, pristupljeno:12.07.2018.

[20] Using the Android Toolbar, Vogella,

<http://www.vogella.com/tutorials/AndroidActionBar/article.html>, pristupljeno: 20.07.2018.

[21] Android Sliding Menu using Navigation Drawer, Android Hive,

<https://www.androidhive.info/2013/11/android-sliding-menu-using-navigation-drawer/>,

pristupljeno: 21.07.2018.

[22] Android-Progress Circle, Tutorials Point,

[https://www.tutorialspoint.com/android/android\\_progress\\_circle.htm](https://www.tutorialspoint.com/android/android_progress_circle.htm),

pristupljeno: 21.07.2018.

[23] How to define callbacks in Android?, Stack Overflow,

<https://stackoverflow.com/questions/3398363/how-to-define-callbacks-in-android>,

pristupljeno: 25.07.2018.

[24] A. Mishra, A. Nieto, S. Kitsiou, Systematic review of mHealth interventions involving Fitbit activity tracking devices, IEEE International Conference on Healthcare Informatics (ICHI), Park City, UT, USA, 23 Aug. 2017, pp. 455-455.

## SAŽETAK

U ovom radu razvijen je sustav za prevenciju pretilosti koji se sastoji od mobilne aplikacije te Fitbit narukvice. Sustav je razvijen u svrhu lakšeg i boljeg praćenje dnevnih aktivnosti korisnika s ciljem alarmiranja i upozoravanja korisnika u situacijama u kojima se nedozvoljeno ponaša, odnosno u situacijama u kojima bi njegova tjelesna težina krenula rasti. Tijekom dana korisnik nosi Fitbit Charge HR koja cijelo vrijeme prikuplja podatke o korisnikovim aktivnostima te ih šalje na web poslužitelj. Nakon što se aplikacija pokrene kreće sinkronizacija s Fitbit narukvicom te dohvaćanje prikupljenih podataka s web poslužitelja. Nakon toga, aplikacija analizira i obrađuje podatke na osnovu čega prosljeđuje odgovarajuću poruku korisniku. Analizom rada aplikacije utvrđena je učinkovitost aplikacije na poticanje veće tjelesne aktivnosti korisnika čime se postiže prevencija pretilosti.

**Ključne riječi:** Android, aplikacija, Fitbit Charge HR, mobilna aplikacija, pretilost, zdravlje.

## **ABSTRACT**

**Title:** A mobile application for the prevention of obesity by collecting parameters of living habits

This paper elaborates on a system for the obesity prevention developed for the purpose of this final paper. The system consists of a mobile application and Fitbit bracelet. The system was developed in order to facilitate and monitor a user's daily activities with a purpose of alerting and warning a user in situations where illegal activities are to be undertaken, i.e. in situations where a user's weight starts to grow. During a day, a user carries Fitbit Charge HR which constantly collects information about a user's activity and sends them to the Web server. Once the application is run, the synchronization with the Fitbit bracelet and retrieval of collected data from the Web server are simultaneously initiated. The application analyzes and processes data upon which, it forwards an appropriate message to a user. Application performance analysis determined the effectiveness of the application to encourage greater body activity of the user and achieving obesity prevention.

**Key words:** Android, Fitbit Charge HR, mobile application, health, obesity.



## **ŽIVOTOPIS**

Svetozar Radić rođen je 05. siječnja 1997. u Novom Sadu. Pohađao je osnovnu školu u Silašu te nakon četvrtog razred kreće u osnovnu školu Tenja. III. gimnaziju u Osijeku upisuje 2011. godine gdje razrede prolazi s vrlo dobrim uspjehom. Nakon završetka srednjoškolskog obrazovanja, godine 2015. upisuje Fakultet elektrotehnike, računarstva i informacijskih tehnologija u Osijeku, preddiplomski studij računarstva na kojem trenutno studira.

---

## **PRILOZI**

Na CD-u:

Prilog 1: Završni rad „Mobilna aplikacija za prevenciju pretilosti prikupljanjem parametara o životnim navikama“ u *.docx* formatu.

Prilog 2: Završni rad „Mobilna aplikacija za prevenciju pretilosti prikupljanjem parametara o životnim navikama“ u *.pdf* formatu.

Prilog 3: Programski kod mobilne aplikacije za prevenciju pretilosti prikupljanjem parametara o životnim navikama.