Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Klemen Susič

Zaznavanje človeških funkcij z uporabo senzorjev in mobilnih naprav

diplomsko delo

visokošolski strokovni študijski program
prve stopnje
računalništvo in informatika

Mentor: doc. dr. Mira Trebar

Ljubljana, 2017

Besedilo je oblikovano z urejevalnikom besedil LaTeX.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Kandidat naj v diplomskem delu razišče področje senzorskih tehnologij za izvedbo mobilne aplikacije, ki sprejema podatke o srčnem utripu uporabnika in izračunava njegovo lokacijo s senzorji pametnega telefona. Izdela naj algoritme za zaznavanje človeških funkcij v povezavi z razlicnimi načini gibanja. Na podlagi ugotovljenih odstopanj pri meritvah srčnega utripa ali detekciji padca naj posreduje SMS sporočilo z obvestilom o lokaciji osebam s shranjenimi kontaktnimi podatki.

Zahvalil bi se rad staršem in starim staršem za vso izkazano podporo in spodbudo v času študija. Posebej bi se rad zahvalil še mentorici doc. dr. Miri Trebar, za vodenje in strokovno pomoč pri pisanju diplomskega dela.

Kazalo

Povzetek

Abstract

1 Uvod **1**

2 Tehnologije in orodja **3**

2.1 Senzorske tehnologije 3

2.2 Razvojna orodja 8

2.3 Analiza sorodnih tehnologij 10

2.4 Podobne rešitve 12

3 Polar H7 **13**

3.1 Način uporabe in delovanja 13

3.2 Življenjski cikel senzorja 14

4 Implementacija algoritmov **19**

4.1 Algoritem za detekcijo padca 19

4.2 Algoritem za določanje gibanja 23

4.3 Algoritem za pridobivanje vrednosti preko GATT strežnika 27

5 Razvoj mobilne rešitve **31**

5.1 Ideja 31

5.2 Metodologija 31

5.3 Opis funkcionalnosti in zahtev 33
5.4 Delovanje mobilne rešitve meProtect 34
5.5 Strukturna zasnova aplikacije 36

6 Testiranje **55**
6.1 Mobilne naprave 55
6.2 Tipi senzorjev 55
6.3 Obravnava prekinitev 56
6.4 Ugotovitve 56
6.5 Primer uporabe 56

7 Sklepne ugotovitve **61**

Literatura **65**

Seznam uporabljenih kratic

kratica	angleško	slovensko
SMS	Short Message Service	storitev kratkih sporočil
GPS	Global Positioning System	globalni pozicijski sistem
BLE	Bluetooth Low Energy	nizko energijski Bluetooth
API	Application Programming Interface	vmesnik za programiranje aplikacij
NFC	Near Field Communication	komunikacija kratkega dosega
GAP	Generic Access Profile	splošen dostop do profila
GATT	Generic ATTribute	protokol splošnih značilnosti
ATT	Attribute protocol	protokol značilnosti
SIG	Special Interest Group	pravna oseba za nadzor standardov Bluetooth
PPG	Photoplethysmogram	zaznavanje na osnovi slikovnih sprememb
ISP	Internet Service Provider	ponudnik internetnih storitev
MAC	media access control address	naslov nadzora dostopa do medija
RSSI	Received Signal Strength Indicator	indikator moči prejetega signala
UNIT	unsigned integer type	nepredznačeno celo število
LSB	least significant bit	bit z najmanjšo težo
IPS	Indoor Positioning Systems	notranji pozicijski sistemi
USB	Universal Serial Bus	univerzalno serijsko vodilo
XML	eXtensible Markup Language	razširljiv označevalni jezik

Povzetek

Naslov: Zaznavanje človeških funkcij z uporabo senzorjev in mobilnih naprav

Avtor: Klemen Susič

"Wearable Health Technology"je eno od področij, ki za delovanje izkorišča različne senzorje in se zaradi napredka le-teh vse bolj uveljavlja. V diplomski nalogi je predstavljena mobilna rešitev za sisteme Android, ki posega na omenjeno področje in za delovanje uspešno vključuje uporabne senzorske tehnologije. Cilj obravnavane tematike je izvedba avtomatizirane mobilne rešitve na pametnem telefonu za obveščanje ob srčnih zastojih in padcih oseb v primerih, ko si te ne morejo same pomagati.

Za razvoj omenjene aplikacije so bile uporabljene napredne tehnologije, kot so senzorji srčnega utripa BLE (angl. Bluetooth Low Energy), pospeškometri in sistem GPS (angl. Global Positioning System), ter nekaj sorodnih tehnologij, ki so vključene kot predmet raziskave. V osrednjem delu sta predstavljena implementacija in opis načrtovanih algoritmov za ugotavljanje padca in načinov gibanja, ki skrbijo za pravilno integracijo in komunikacijo uporabljenih tehnologij.

Predstavitve ključnih aktivnosti, strukturne zasnove, delovanje mobilne rešitve in funkcionalnosti so podane v zadnjem delu, ki poleg omenjenega vsebuje še rezultate testiranj in analizo izboljšav na posameznih modulih.

Ključne besede: senzorji, GPS, BLE, pospeškometer, Android.

Abstract

Title: Detection of human functions with sensors and mobile devices

Author: Klemen Susič

Wearable Health Technology is one of the fields that uses different sensors for operation. Due to development of sensors technology this field is more and more popular. In the thesis is presented a mobile solution for Android system, which takes part of this field and uses different sensors technologies. The main goal is to achieve an automated solution for mobile device that can inform others in case of falling or heart attack of a person.

In the development of this solution were used technologies like heart rate BLE (Bluetooth Low Energy) sensors, accelerometers, GPS (Global Positioning System) and some others that are part of research. Focus in the second part is put on individual implementation of technologies and used algorithms for detect falling and different movements.

Presentation of main activities, structural composition, functionality and app working are described in third part, which additionally includes the results of testing and future improvements of individual modules.

Keywords: sensors, GPS, BLE, accelerometer, Android.

Poglavje 1

Uvod

Mobilne naprave so v zadnjih letih postale človekov nepogrešljiv sopotnik. Trenutno je na svetu skoraj 5 milijard aktivnih mobilnih naprav, ki se povezujejo v različna globalna, kot tudi lokalna omrežja [21]. Prav ta sposobnost različnih vrst povezav daje tovrstnim napravam veliko priljubljenost med ljudmi in zavoljo tega visoko funkcionalnost in učinkovitost. Danes mobilne naprave omogočajo bistveno več kot telefoniranje, pošiljanje sporočil SMS (angl. Short Message Service) in zajem fotografij. Lahko rečemo, da so pravi mali super računalniki, ki zmorejo obdelavo velikih količin podatkov in tako z nekaj kliki uporabniku postrežejo z želenimi informacijami.

Sodobne mobilne naprave so poleg zmogljivih komponent opremljene z različnimi senzorji, katerih se uporabniki malo kdaj zavedajo, saj jih mobilne aplikacije uporabljajo brez njihove vednosti. V to kategorijo sodijo senzorji za toploto, pospeškometri, giroskop, barometer, senzor za svetlobo in vlažnost. Podatki, ki jih ti senzorji nudijo, precej razširjajo samo funkcionalnost naprave in lahko ob pravi interakciji z ostalimi senzorji za povezljivost (GPS, Bluetooth, NFC, ...) predstavljajo široko uporabnost naprave.

Zaradi dejstva, da ima danes skoraj vsak posameznik v svojem žepu mobilno napravo, ki omogoča zgoraj naštete funkcionalnosti, si prizadevajo različne organizacije v sodelovanju z industrijo najbolje izkoristiti tovrstne možnosti. Eno izmed področij, ki si prizadeva za učinkovito izrabo mobilnih

naprav, je tudi medicina in z njo povezan pojem "Wearable Health Technology".

V diplomskem delu je predstavljena mobilna rešitev, ki posega na področje zgoraj omenjene tehnologije in uporabniku preko vgrajenih in zunanjih senzorjev omogoča nadzor nad psihofizičnim stanjem. Mobilna aplikacija meProtect spremlja in nadzoruje kardiovaskularni in gibalni del. V kardiovaskularnem delu se preko senzorja srčnega utripa nadzoruje delovanje srca, medtem ko pospeškometer skrbi za spremljanje gibanja. Namen opazovanja omenjenih funkcij je avtonomna detekcija srčnega zastoja oziroma padca osebe in ustrezno ukrepanje s pošiljanjem sporočila SMS vnaprej določenim osebam.

Poglavje 2

Tehnologije in orodja

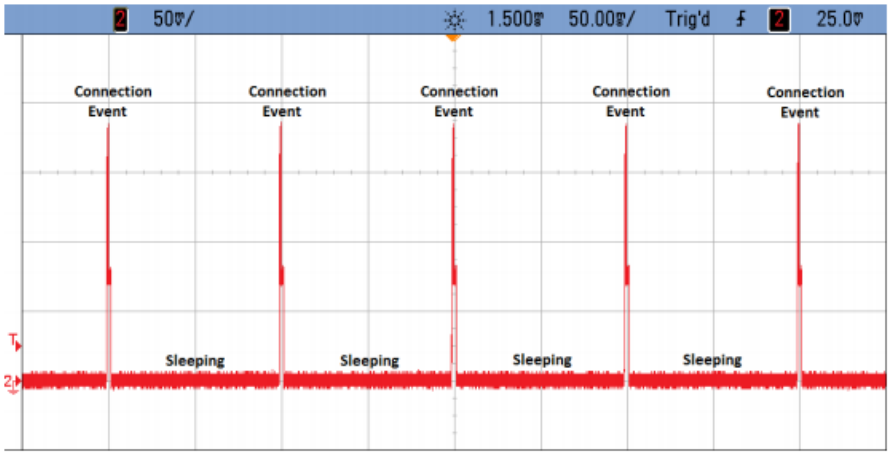
2.1 Senzorske tehnologije

Mobilna rešitev bo za svoje delovanje uporabljala senzor srčnega utripa, ki temelji na komunikaciji Bluetooth Low Energy, pospeškometer in GPS.

2.1.1 BLE (Bluetooth Low Energy)

Bluetooth je brezžična tehnologija, namenjena prenosu podatkov med napravami. Gre za varno povezavo, ki deluje na frekvenci 2,4GHz in v povprečju omogoča prenos podatkov do 10 metrov. Z leti so se pojavljale različne tehnologije Bluetooth, med njimi tudi Bluetooth Low Energy [10], ki je bila posebej razvita za povezavo mobilnih naprav s športnimi in fitnes napravami ter različnimi senzorji, ki zahtevajo prenos manjših količin podatkov. Bluetooth Low Energy spada med protokol klasičnega Bluetooth 4.0, ki je bil predstavljen junija 2010. V mobilne naprave z operacijskim sistemom Android je bil integriran z različico 4.3. Princip delovanja protokola je omejen na 1 Mbit prenosa podatkov na sekundo s pulznim načinom, imenovanim "Beacon", ki zmanjša porabo energije in omogoča napravam delovanje na majhne baterije, imenovane gumb. Prednost Beacon tehnologije [4] je v kratkih pulzih, pri katerih se prenese želen podatek in nato dovoli povezavi Bluetooth, da preide v spanje in na ta način varčuje z energijo (Slika 2.1).

Klasičen Bluetooth tega načina ne dovoljuje.



Slika 2.1: Način delovanja povezave BLE [4].

2.1.2 BLE komunikacijski protokol

Za komunikacijo med napravo in senzorjem skrbita protokola **GAP** (angl. Generic Access Profile) [6] in **GATT** (angl. Generic ATTribute) [7].

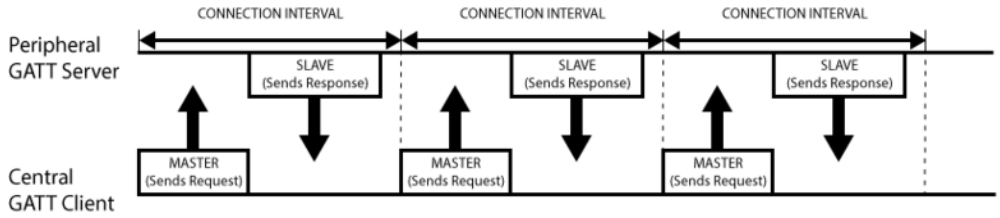
Protokol GAP

Protokol GAP skrbi za vzdrževanje in vzpostavitev povezave med napravama. Preko tega protokola se naprava oglašuje in je vidna ostalim. V protokolu GAP obstajata dve vlogi, poimenovani **periferna** in **centralna**. Pod periferno vlogo sodijo razni senzorji, ki omogočajo oddajanje podatkov in manj kompleksnejša procesiranja. Centralne naprave pa so zmogljivejše in dovoljujejo zahtevnejša procesiranja. Navadno gre za mobilne telefone, pametne ure, tablice, itd. V protokolu GAP lahko periferne naprave oddajajo majhno količino podatkov (do 31 bajtov), znano kot oglaševanje. S pomočjo teh podatkov lahko centralne naprave vidijo njihov obstoj v okolici in se z njimi povežejo. Ko je povezava med centralno in periferno napravo vzpostavljena,

protokol GAP poskrbi, da periferna naprava preneha z oglaševanjem in s tem prepreči povezovanje z drugimi centralnimi napravami.

Protokol GATT

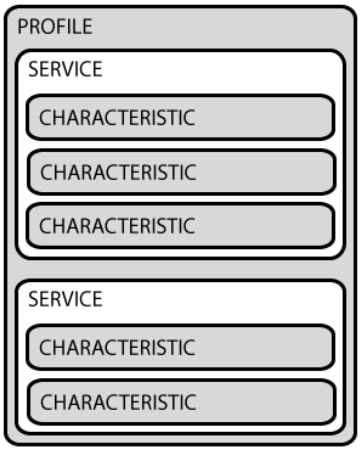
Po uspešni povezavi v ospredje stopi protokol GATT, ki je namenjen prenosu podatkov med napravama. Prenos poteka po konceptu storitve in karakteristike (angl. Services and Characteristics), ki sestoji iz protokola ATT (angl. Attribute protocol). Ta omogoča shranjevanje storitvenih in karakterističnih vrednosti v tabelo z uporabo 16 bitnih ID-jev. Periferna naprava se v tem protokolu predstavlja kot strežnik GATT, ki drži tabelo ATT z definiranimi storitvenimi in karakterističnimi vrednostmi. Na drugi strani centralna naprava oziroma odjemalec GATT preko zahtevkov od strežnika GATT prejema podatke, katere ta hrani v tabeli ATT (Slika 2.2).



Slika 2.2: Prenos podatkov med strežnikom in odjemalcem GATT [7].

Storitve in karakteristike

Storitve in karakteristike se v BLE komunikaciji uporabljajo za kategorizacijo podatkov. Vsaka periferna naprava ima lahko več storitev (Slika 2.3), ki vsebujejo različne karakteristike. Ti podatki se združujejo v profil, ki je v perifernih napravah vnaprej definiran s strani proizvajalca. Vrednosti teh podatkov določa združenje Bluetooth SIG (angl. Special Interest Group), ki skrbi za konsistentnost med proizvajalci [5].



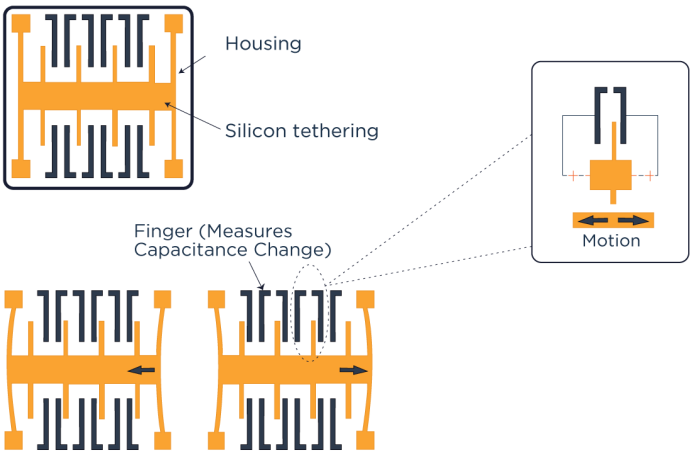
Slika 2.3: ATT tabela [7].

2.1.3 Pospeškometer

Pospeškometer je senzor, ki je vgrajen v skoraj vse mobilne naprave. Omogoča merjenje gravitacijske sile glede na napravo. Na ta način lahko določimo položaj telefona, oziroma silo premika naprave v prostoru. Senzor zaznava tri pozicije premika. Premik po x–osi, y–osi in z–osi. V mirovanju naprave je ena izmed osi v odvisnosti od položaja ekvivalentna sili gravitacije ($9,81 m/s^2$). V primeru, ko se telefon premika, lahko z opazovanjem razporejanja sil po oseh določimo smer gibanja naprave.

Zgradba pospeškometra

Pospeškometer je v mobilnih napravah integriran na osnovno vezje. Za delovanje izkorišča mehanske lastnosti silicija, ki je v obliki vzmeti pritrjen na samo ohišje čipa in se ob premikih naprave ustrezno giblje med kapacitivnimi senzorji oz. piezoelektričnimi kristali (Slika 2.4). Pri nihanju silicijeve vzmeti nastane določena napetost, ki se ustrezno preračuna v silo premika [18].



Slika 2.4: Shema delovanja pospeškometra [18].

2.1.4 GPS (Global Position System)

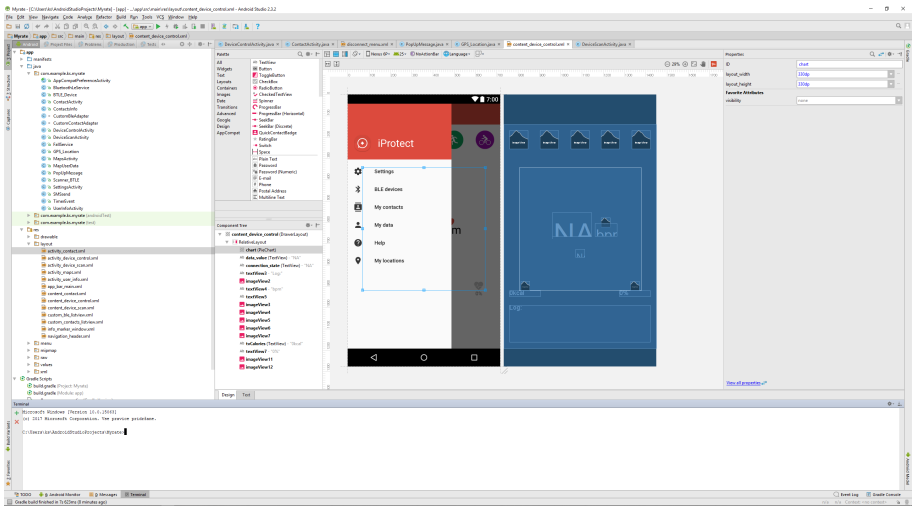
GPS [8] je ena izmed pogosteje uporabljenih tehnologij v sodobnem svetu. Omogoča določanje pozicije na zemlji z natančnostjo do petih metrov. Za njen izračun so potrebni štirje sateliti, pri katerih se preračuna razlika v prejeti časovni znački. Vsak satelit vsebuje atomsko uro, ki sporoča točen čas, in tako lahko mobilna naprava na podlagi zamika prejetih časov iz satelitov določi lokacijo na zemlji.

Sistem Android omogoča, da se frekvenca pridobivanja signala določi poljubno. Lahko temelji na oddaljenosti od prejšnje pozicije ali na časovnem intervalu. Na ta način lahko omejimo količino prejetih podatkov in zmanjšamo porabo energije.

2.2 Razvojna orodja

2.2.1 Android Studio

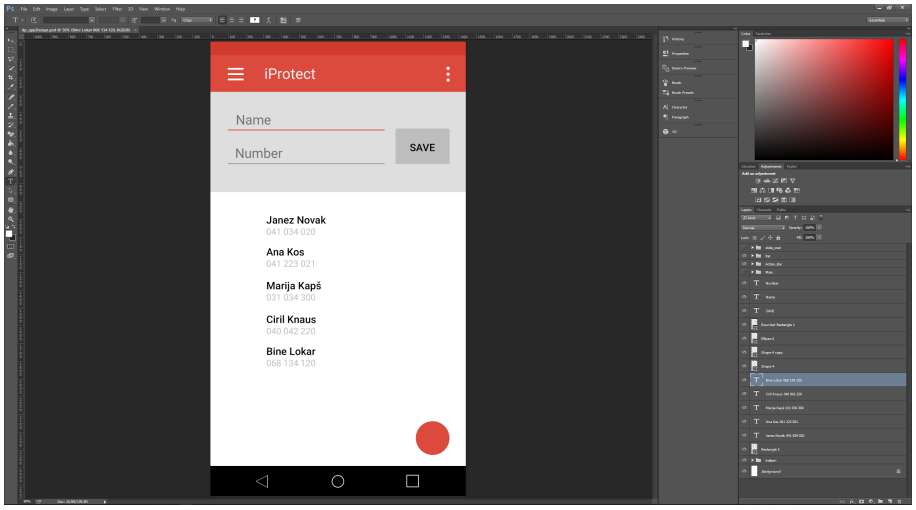
Je razvojno okolje, namenjeno programiranju mobilnih aplikacij za platformo Android v za to prirejenem programskem jeziku JAVA. Gre za brezplačno programsko opremo, razvito pri Googlu, ki deluje na vseh treh večjih operacijskih sistemih Linux, MAC OS in Windows. Ponuja ogromno funkcionalnosti: od *front–end* do *back–end* razvoja, ter implementacije prenastavljenih zaslonov. Poleg tekstovnega kodiranja ima možnost gradnje vmesnika z gradniki in na ta način pripomore k hitrejši in lažji implementaciji. Razvoj lahko poteka v ponujenih emulatorjih, ki jih Android Studio vključuje za razvijalce brez fizičnih naprav, kot tudi v samih fizičnih napravah preko povezave USB (angl. Universal Serial Bus). Veliko boljši in lažji način je programiranje neposredno v fizično napravo, saj je precej hitrejši in omogoča preizkus funkcij, ki jih z emulatorji ne moremo simulirati, npr. prenos preko povezave Bluetooth, zajem fotografij s fotoaparatom, predvajanje zvoka, itd. (Slika 2.5).



Slika 2.5: Razvojno orodje Android Studio.

2.2.2 Adobe Photoshop CC

Adobe Photoshop (Slika 2.6) je program, ki ga uporabljajo izključno oblikovalci in fotografi. Je eno izmed bolj razširjenih orodij, ki ponuja obdelavo nad rasterskimi slikami. Gre za profesionalno programsko opremo, ki je plačljiva in ima možnost tri mesečne brezplačne uporabe. Deluje samo na operacijskih sistemih Windows in MAC OS. Njena glavna lastnost in prednost je delo s plastmi v kombinacijah s slikovnimi efekti in filtri. Pri razvoju aplikacije smo z njim narisali celoten izgled z vizualnimi elementi, potrebnimi za pravilen prikaz.



Slika 2.6: Oblikovalsko orodje Adobe Photoshop CC.

2.2.3 MATLAB

MATLAB [14] je še eno izmed plačljivih orodij, ki so bolj namenjena raziskovalcem kot širšim uporabnikom. Je dokaj zahtevno za uporabo, saj zahteva določeno predznanje specifičnih ukazov in ne omogoča toliko vnaprej definiranih funkcionalnosti. MATLAB temelji na podajanju ukazov, preko katerih se lahko izvajajo razne matematične operacije: od enostavnih do zelo kompleksnih izrisov grafov in analiz signalov. S tem orodjem smo izvedli celotno

analizo signala, ki nastane pri padcu, in signala, ki nastaja v različnih fazah gibanja.

2.3 Analiza sorodnih tehnologij

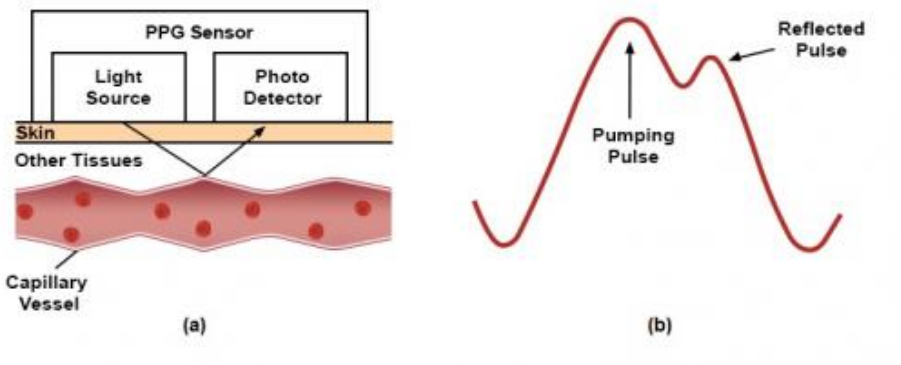
2.3.1 Tehnologija PPG (Photoplethysmography)

Senzorji so čedalje bolj izpopolnjeni in je raba le-teh precej enostavna. Razvijalcem uspeva integracija senzorjev, ki so za človeka nemoteči in precej intuitivni. Ena od zadnjih načinov merjenja srčnega utripa, ki se je pojavila prav z razvojem mobilnih tehnologij, je merjenje s pomočjo foto kamere mobilne naprave oziroma PPG (angl. photoplethysmography). Tehnologija deluje tako, da uporabnik položi prst na bliskavico in lečo kamere. Medtem algoritem skozi foto kamero zajema odstopanje in razliko v odbiti rdeči svetlobi, ki ustreza različnemu krvnemu pretoku. Pri različnih stopnjah utripa je gostota krvi drugačna, kar vpliva na odtenek rdeče barve. Algoritem tako lahko analizira povprečje rdečih okvirjev, zajetih skozi foto kamero. Graf povprečnih vrednosti rdečih komponent vsebuje ostre lokalne maksimume, za katere so značilni hitri premiki iz visokih pozitivnih na nizke negativne vrednosti (Slika 2.7b). Ti maksimumi ustrezajo enemu srčnemu utripu, pri čemer seštevek vseh daje končni srčni utrip [12].

Takšna meritev traja nekaj sekund in odčitek velja za čas odčitavanja. Tehnologija sicer omogoča spremljanje srčnega utripa skozi čas, vendar zaradi izvedbe s foto kamero ne dovoljuje monitoringa. Zato tudi ni bila uporabljena kot rešitev pri mobilni aplikaciji.

Enako tehnologijo PPG uporabljajo zapestni senzorji (pametne ure), kateri namesto foto kamere uporabljajo infrardečo svetlobo [23, 16]. Slika 2.7a prikazuje vgrajene led diode, ki proizvajajo zeleno svetlobo, katero kri zaradi rdeče barve absorbira. Pri različnih stopnjah utripa se absorbcija barve spreminja in tako lahko z detektiranjem količine absorbcije in odbite svetlobe

določimo število srčnih utripov.



Slika 2.7: Delovanje tehnologije PPG [19].

Uporabljena tehnologija velja za manj natančno predvsem zaradi nenadnih in ne ponavljajočih se premikov roke, ki lahko povzročijo napačno zaznavanje utripa. Je pa tovrstna tehnologija povsem kompatibilna s podano rešitvijo mobilne aplikacije meProtect, na katero ne vpliva tehnologija, ki jo senzor uporablja za detekcijo utripa, temveč tehnologija, ki jo senzor uporablja za komunikacijo z odjemalcem. Če ta uporablja za prenos podatkov BLE, je uporaba takšnega senzorja povsem sprejemljiva.

2.3.2 Zunanji BLE pospeškometri

Tovrstni senzorji za komunikacijo uporabljajo enako tehnologijo kot Polar H7, kar jim omogoča pošiljanje podatkov na mobilno napravo. Primer takšnega senzorja je "ASensor" [2], ki v rešitvi ni podprt. Razlog so težave pri rokovanju z več senzorji, ki za komunikacijski protokol uporabljajo BLE. Android namreč ne dovoljuje paralelne komunikacije, zato je potrebno vrednosti odčitavati zaporedno, kar je počasneje in manj natančno. V rešitvi se uporablja integriran pospeškometer, ki ne obremenjuje komunikacije Bluetooth. V kolikor bi obstajal senzor srčnega utripa in pospeškometra v eni napravi, bi bil način uporabe zaželen, saj ena BLE povezava lahko prenaša več različnih podatkov in bi bilo procesiranje izvedeno na paralelni način.

2.4 Podobne rešitve

Trenutno na trgu ni mobilne rešitve, ki bi ustrezala enaki funkcionalnosti, kot smo jo zasnovali v aplikaciji meProtect. Obstoječe rešitve so le delno podobne oziroma izkoriščajo enako tehnologijo za povezovanje.

Polar Beat

Polar Beat je mobilna aplikacija, ki jo proizvajalec senzorja Polar H7 nudi kot orodje za uporabo skupaj s senzorjem. Namenjena je spremljanju vadbe oziroma treninga. Omogoča pregled nad maksimalnim in povprečnim srčnim utripom, porabo kalorij in maščob, časom vadbe, prepotovano razdaljo in prikaz omenjenih funkcij na grafu. Prednosti takšne aplikacije so predvsem raznoliki podatki in analize nad njimi, ki jih meProtect ne ponuja. Dejstvo, da se tovrstnih funkcionalnosti ni vključilo v mobilno rešitev, je drugačna namembnost aplikacije. Pri Polar Beat je v ospredju spremljanje fizične aktivnosti, medtem ko meProtect pridobljene podatke izkorišča za varovanje osebe. Aplikacija Polar Beat je prosto dostopna v trgovini Google Play[1].

Safety One Touch

Safety One Touch je plačljiva mobilna aplikacija za detekcijo padca. Njeno delovanje je precej podobno mobilni rešitvi meProtect, saj omogoča enak način obveščanja in lociranja pomoči potrebne osebe. Slabost aplikacije v primerjavi z meProtect je zmožnost nastavljanja le treh oseb za obveščanje in slabša podpora za novejše Android platforme. Aplikacija je dostopna preko trgovine Google Play[2].

[1] https://play.google.com/store/apps/details?id=fi.polar.beat.

[2] https://play.google.com/store/apps/details?id=com.gc.android.safetyonetouch.

Poglavje 3

Polar H7

Podjetje Polar je eno izmed vodilnih proizvajalcev elektronskih športnih merilcev. Proizvaja visokotehnološke naprave, ki se uporabljajo pri profesionalnih športnikih in rekreativcih. Polar H7 BLE (Slika 3.1) je brezžični senzor srčnega utripa, ki za prenos podatkov uporablja tehnologijo Bluetooth Low Energy. Namenjen je spremljanju srčnega utripa na napravah, ki podpirajo komunikacijo Bluetooth. V to kategorijo sodijo tablice, pametni telefoni in fitnes naprave.

3.1 Način uporabe in delovanja

Naprava je sestavljena iz elektrode na pasu in elektronskega sprejemnika. Pas se pritrdi okrog prsnega koša v predelu srca, ter se s tem zagotovi detekcijo električne aktivnosti, ki jo proizvaja srce med delovanjem. Senzor za detekcijo uporablja elektrokardiograf, ki zaznava spremembe v električni aktivnosti in na ta način preračuna število utripov na minuto. Ti se preko komunikacije Bluetooth prenesejo na odjemalca.

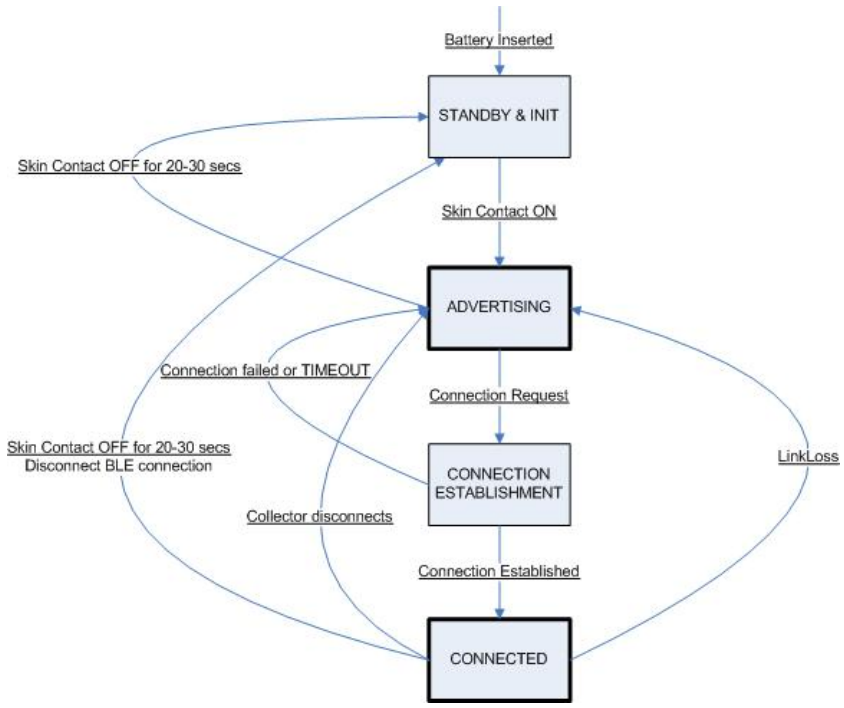
Slika 3.1: Senzor srčnega utripa Polar H7 [9].

3.2 Življenjski cikel senzorja

Polar H7 senzor ima tri stanja delovanja (Slika 3.2), med katerimi prehaja (pripravljenost, oglaševanje in povezanost) [17]. V stanju pripravljenosti senzor miruje in ne oddaja signala Bluetooth v okolico, ter tako varčuje s porabo energije, ki znaša manj kot 1uA. Pogoj za ohranjanje tega stanja je prekinjen stik elektrod s kožo za več kot 20–30 sekund.

Stanje oglaševanja nastopi ob stiku elektrod s kožo. Takrat se aktivira oddajanje signala Bluetooth po protokolu GAP, ki poskrbi za identifikacijo naprave v prostoru.

Ob uspešni povezavi s centralno napravo senzor preide v stanje povezanosti, ki s sekundnim intervalom po protokolu GATT pošilja vrednosti srčnega utripa.

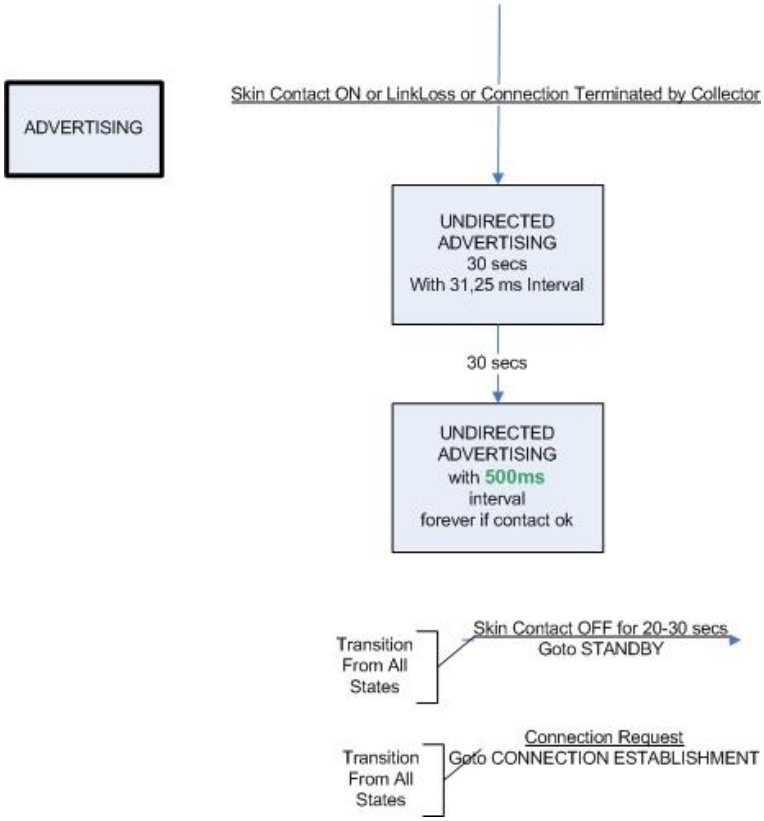


Slika 3.2: Diagram prehajanja stanj za Polar H7 [17].

Pravila delovanja stanj

Znotraj posameznega stanja imamo prav tako možne različne tipe delovanja, ki se lahko pojavijo v odvisnosti od zunanjih sprememb oziroma dejanj.

Stanje oglaševanja (Slika 3.3) ima štiri različne vloge delovanja. Prvih 30 sekund se naprava oglašuje z 31,25 ms intervalom. Interval je na začetku višji z namenom boljšega odkrivanja naprave. Po preteku 30 sekund se interval oglaševanja skrajša na 500 ms, kar zmanjšuje porabo energije. V primeru, da senzorski pas izgubi stik s kožo, se po 20–30 sekundah premakne v stanje pripravljenosti. Ob uspešni povezavi pa v stanje povezanosti.

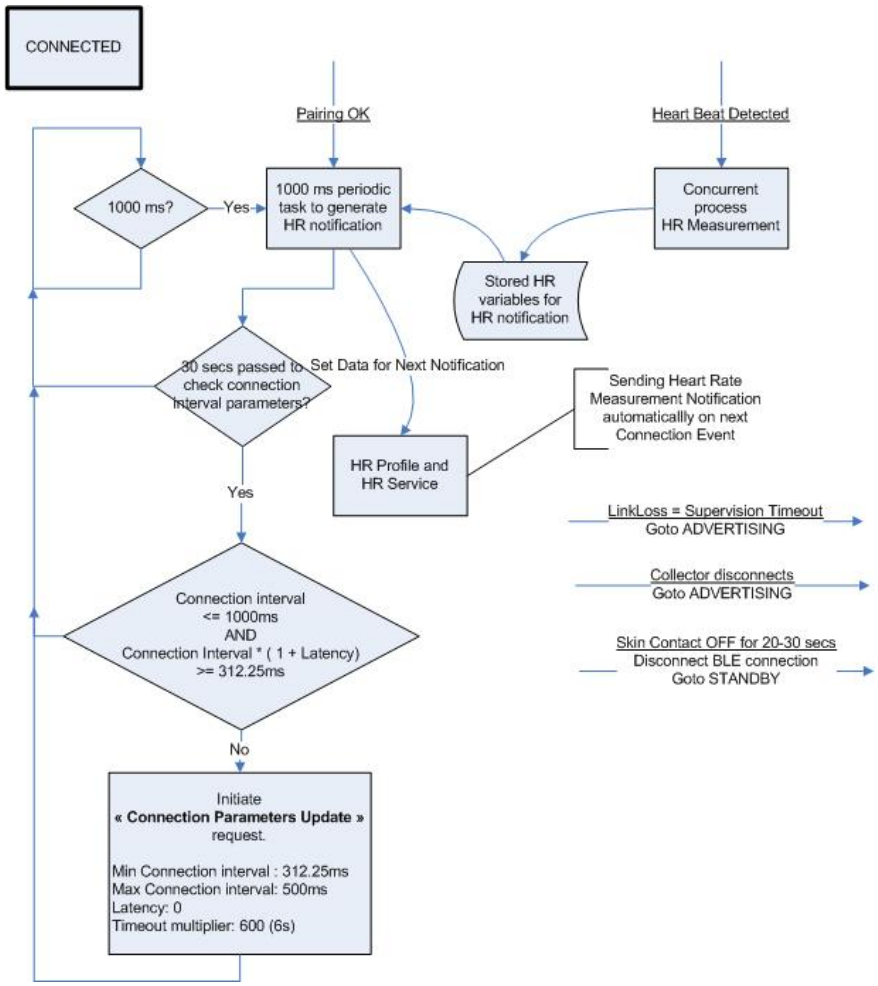


Slika 3.3: Preklopi znotraj stanja oglaševanja [17].

Stanje povezanosti je po načinu delovanja bistveno kompleksnejše od predhodnih dveh. Vsebuje namreč množico operacij, ki skrbijo za pravilno izvajanje podatkovnega toka znotraj stanja (Slika 3.4).

Ob uspešni povezavi se vsako sekundo generira nalog za nastavitev HEART RATE obvestila. V kolikor senzor zahtevane podatke pridobi, se v naslednjem intervalu ti posredujejo centralni napravi. Ker izmenjava podatkov med periferno in centralno napravo poteka preko povezave Bluetooth Low Energy, ki uporablja tehnologijo *Beacon*, se vsakih trideset sekund preverijo parametri povezovalnega intervala. Ti so pomembni za ohranjanje nizke porabe energije. Kontrolo nad njimi ima centralna naprava, ki lahko zahteva drugačen, daljši interval in s tem še dodatno zmanjša porabo. Posledično

se zaradi daljših intervalov nekateri paketi ne dostavijo centralni napravi, kar vpliva na natančnost podatkov. Maksimalni zakasnitveni interval je štiri sekunde, kar pomeni izgubo treh paketov, ki jih sicer senzor posreduje v sekundnem intervalu. Potrebno se je zavedati, da nepravilno nastavljeni parametri lahko vodijo v prekinitev povezave oziroma preklop senzorja v stanje oglaševanja. Med stanjem povezanosti se lahko pojavijo tri vrste prekinitev: izguba povezave, prekinjena povezava in izguba stika elektrod s kožo. Vrsta prekinitve določa preklopno stanje senzorja. Pri izgubi stika elektrod s kožo sledi stanje pripravljenosti, sicer stanje oglaševanja.



Slika 3.4: Preklopi znotraj stanja povezanosti [17].

Poglavje 4

Implementacija algoritmov

Postopkov in algoritmov, ki so vključeni v aplikacijo je veliko. V tem poglavju si bomo pogledali tri najpomembnejše algoritme, ki so bili razviti posebej za namen aplikacije meProtect in se nahajajo znotraj dveh procesov `BluetoothLeService.java` in `FallService.java`.

4.1 Algoritem za detekcijo padca

Algoritmi, ki temeljijo na presoji in odločanju, so pogosto nezanesljivi, ko gre za razlikovanje situacije nad podobnimi podatki. Takrat je določitev stopnje, pri kateri algoritem zavzame določeno stanje, precej težavno. Zato se nam velikokrat pojavijo tako imenovani False/Positive[1] odgovori, ki so posledica enakih podatkov, generiranih s strani različnih okoliščin. Da se takšnim primerom, ko je odgovor False/Positive[1], izognemo, uporabljamo večstopenjsko ugotavljanje. V to kategorijo sodi tudi algoritem za detekcijo padca pri človeku, ki uporablja tri stopnje ugotavljanja:

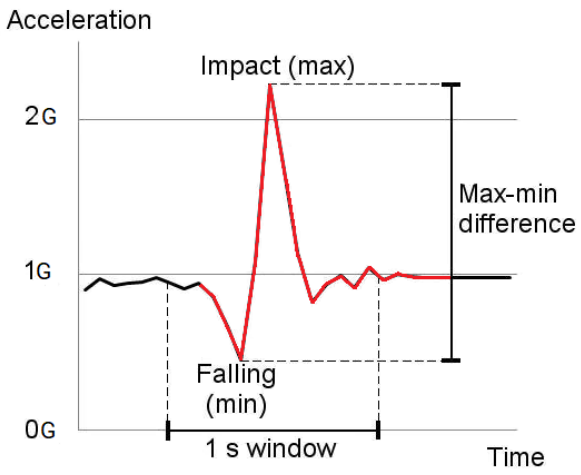
- pridobljene podatke pred padcem,
- pridobljene podatke med padcem in
- pridobljene podatke po padcu.

[1]False/Positive - Odgovor, ki je pravilen vendar ne velja za dano situacijo.

Na podlagi vrednosti, pridobljenih iz navedenih stopenj, se nato algoritem odloči ali je prišlo do padca.

4.1.1 Postopek za detektiranje padca

Padec sam posebej sodi v kategorijo gibanja, ki se lahko ovrednoti s senzorjem pospeška. V tem primeru se pospeškometer nahaja znotraj mobilne naprave. Če si podrobneje ogledamo, kaj se dogaja pri padcu z vrednostmi, pridobljenimi iz pospeškometra, lahko ugotovimo posebno stanje, ki ni značilno za ostala gibanja. Pri padcu namreč pride do prostega pada, ki se skozi senzor odraža kot ničelni pospešek. Takrat so vrednosti signala zelo blizu ničle, saj na senzor ne deluje nobena sila. Drugo, zelo pomembno stanje, ki velja za padec, je ustavitveno stanje ob stiku s podlago, ki nastopi po prostem padu. Takrat so vrednosti signala precej visoke zavoljo pojemka, ki nastane ob stiku s podlago. Sila pojemka je sicer odvisna od hitrosti, teže in zaustavitvene razdalje, ki je pri padcu ni. Sile, ki se sprostijo pri padcu osebe, ustrezajo vrednostim med 2,5G do 5G ($1G = 9,81 m/s^2$), kar je ob prestrezni vrednosti v tem razponu že dober pokazatelj, da gre za padec (Slika 4.1). V zadnjem stanju se spremlja še signal po stiku s podlago, ki razjasni predhodni dve stanji. V kolikor se vrednosti sile gibljejo v razponu 1G, kar ustreza stanju osebe v mirovanju, lahko sklepamo, da je najverjetneje prišlo do padca in ne gre za False/Positive[1] odgovor. V kolikor vrednosti signala po stiku s podlago še naprej nihajo, lahko predvidevamo, da gre za skok in ne padec, saj je skok v obeh dveh stanjih zelo podoben padcu. Razlikuje se le v zadnjem stanju, ko so vrednosti sile večje od 1G [3, 20].



Slika 4.1: Prikaz vrednosti signala ob padcu [15].

4.1.2 Implementacija algoritma

Za implementacijo zgoraj opisanega postopka v algoritem je potrebno pridobljene senzorske vrednosti pospeškometra pretvoriti v vektor pospeška in v gravitacijsko silo. Pospeškometer, kot vemo, posreduje zgolj podatke o sili pospeška v x, y in z smeri, kar se lahko s pomočjo formul (4.1) in (4.2) preračuna v želeni vrednosti [24].

Formula za izračun vektorja pospeška:

$$A_t = \sqrt{A_x^2 + A_y^2 + A_z^2}, \tag{4.1}$$

Kjer so:

Ax: vrednost pospeška na x–osi,

Ay: vrednost pospeška na y–osi,

Az: vrednost pospeška na z–osi.

Formula za izračun gravitacijske sile:

$$G = \frac{A_t}{g} \qquad g = 9.81 \qquad (4.2)$$

Algoritem za analizo signala deluje v trosekundnih intervalih, pri katerih se zajete vrednosti začasno shranjujejo v ArrayList. Po izteku intervala se izvede izračun nad zajetimi vrednostmi in nadaljna odločitev. Trosekundni interval se uporablja zaradi zajema daljše slike signala, ki nam omogoča natančnejšo analizo le-tega.

```
public void fallDetection(){
 if(aT<1.7 && !detected_free_fall){
  detected_free_fall=true;
  timer_fall_detection.start();
 }

 if(timer_fall_detection.timerRunning())
  detected_values.add(aT);
 if(timer_fall_detection.isRunOut() &&
   !detected_values.isEmpty()){
  if((Collections.max(detected_values)/9.98)>3*weight){
   detected_impact=true;
   alert_timer.start();
  }else
   detected_free_fall=false;
  detected_values.clear();
 }
 if(detected_impact && alert_timer.isRunOut()){
  if(state.equals("Mirovanje"))
   broadcastUpdate("FALL", "HELP");
  else
   broadcastUpdate("FALL","CANCLE");
  detected_impact=false;
  detected_free_fall=false;
 }
}
```

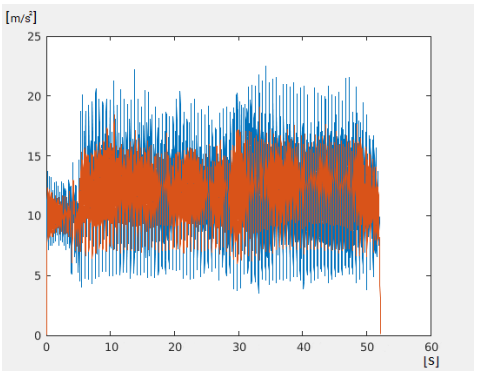
Izpis 4.1: Izsek kode za detekcijo padca.

4.2 Algoritem za določanje gibanja

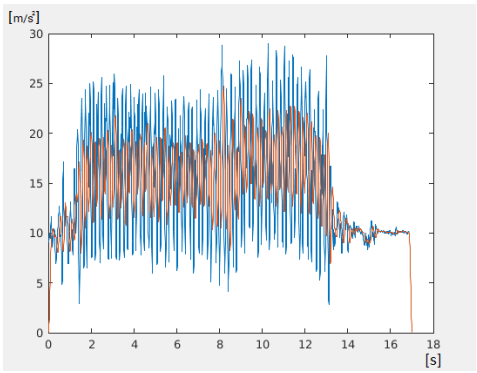
Vrst gibanj je več, npr. hoja, tek, mirovanje, poskakovanje ... in vsako od njih ima svoj specifični vzorec, ki je ponavljajoč (Slika 4.2). Za določitev vzorca je vsako gibanje potrebno spremljati in beležiti signal, ki pri tem nastane. Šele po pridobitvi vzorcev gibanj se lahko te implementira v algoritem, ki razpozna razliko med njimi in določi ustrezen tip gibanja.

4.2.1 Razvoj algoritma

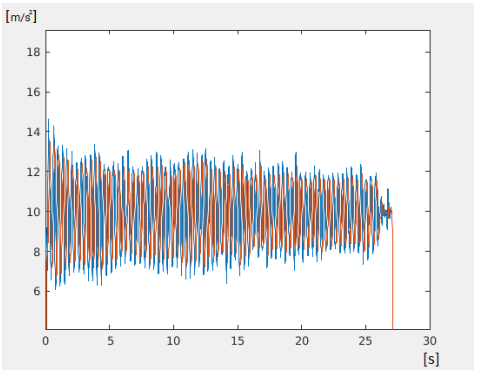
Za definiranje vzorcev je bil narejen poseben program, ki beleži silo pospeška pri različnem gibanju. S pomočjo programa MATLAB smo pridobljene podatke transformirali v signal, ki ustreza spremljajočemu se gibanju.



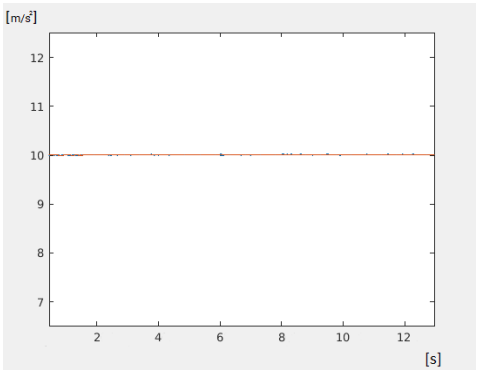
(a) Hoja



(b) Tek



(c) Gibanje



(d) Mirovanje

Slika 4.2: Vzorci signalov za različne aktivnosti.

Vsak signal smo v programu sprocesirali z dvema filtroma. Najprej je bil uporabljen filter tekočega povprečenja, ki zgladi motnje v signalu. Zatem je bila uporabljena Fourierjeva transformacija, ki iz signala izlušči frekvence in omogoča konstrukcijo nizkoprepustnega filtra. Uporaba nizkoprepustnega filtra se odraža skozi čistejši signal brez visokih in nizkih lokalnih maksimumov, ki je na sliki 4.2 predstavljen kot rezultat v rdeči barvi.

```
fid=fopen('signal', 'r');
y4=fscanf(fid, '%f');
Ts=0.06; %simplirni interval
Fs=1/Ts; %simplirna frekvenca 16,67Hz
y4=smooth(y4, 'moving'); %glajenje s tekocim povprecenje
nfft = 2^nextpow2(length(y4));
fftY=fft(y4, nfft);
fftY=fftY(1:nfft/2);
yfft=Fs.*(0:nfft/2-1)/nfft;
%konstrukcija low-pass filtra
cut_freq=1/Fs/2;
bh = fir1(16,cut_freq,'low',hamming(16+1));
sig=conv(y4,bh); %apliciranje low-pass filtra na signal
plot(y4) %originalni signal
hold on
plot(sig) %sprocesiran signa
```

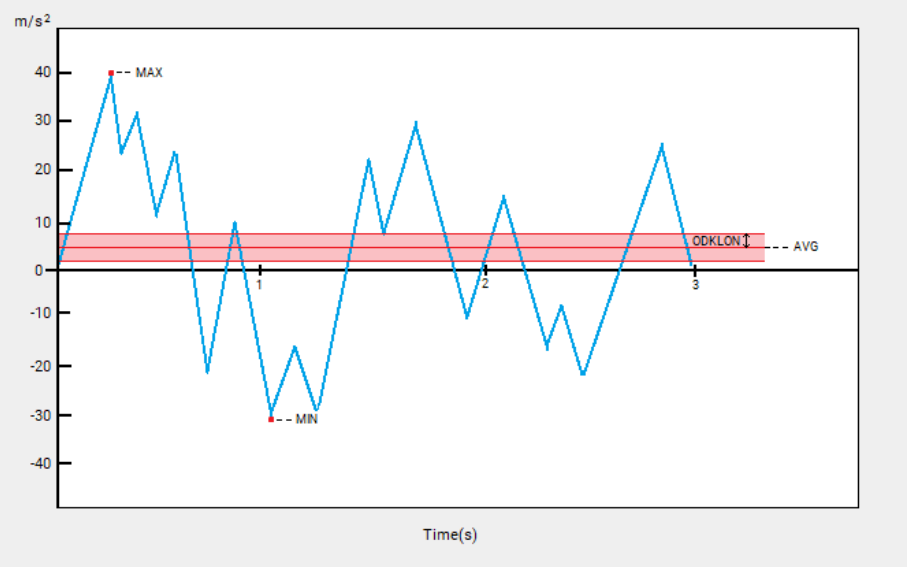
Izpis 4.2: Koda za procesiranje signala.

Iz pridobljenih vzorcev signala smo definirali gravitacijske sile (Tabela 4.1), ki ustrezajo posameznemu gibanju in so osnova za delovanje algoritma.

Tip gibanja	MIN	MAX	AVG	AVG (ODKLON)
Mirovanje	>8,5	<11,6	10	$\pm 0,4$
Gibanje	<8,5	>11,6	10	$\pm 0,4$
Hoja	>1,5	<40	11,65	$\pm 1,25$
Tek	>0,35	<55	14,95	$\pm 2,05$

Tabela 4.1: Klasifikacija gravitacijskih sil za gibanje.

Vrednosti MIN in MAX določata minimalni in maksimalni ekstrem v signalu, ki je možen za posamezno aktivnost. Vrednost AVG velja za povprečje zajetega signala s trosekundnim intervalom. ODKLON določa območje odstopanja od povprečne vrednost zajetega signala (Slika 4.3).



Slika 4.3: Prikaz definiranih vrednosti iz tabele 4.1 na signalu.

4.2.2 Implementacija algoritma

S pomočjo analize signala različnih gibanj je bil sestavljen algoritem, ki upošteva vzorce sil, podanih v tabeli 4.1. Njegovo delovanje temelji na ekstremih signala in odstopanju povprečnih vrednosti. Vsako gibaje ustvari določen nivo ekstremov in določeno odstopanje povprečne vrednosti od gravitacijske sile 1G. Naloga algoritma je analiza zajetega signala v trosekundnem intervalu, ter pravilna klasifikacija analiziranih vrednosti v štiri razrede: mirovanje, gibanje, hoja in tek. Za definiranje klasifikacijskih vrednosti smo uporabili podatke iz analize signala za določen tip gibanja (Slika 4.2).

```
public String detectActivity(){
        sum=0;
        for (double v:tabela)
            sum+=v;

        sum = sum/tabela.size();
        min = Collections.min(tabela);
        max = Collections.max(tabela);
        tabela.clear();

        if(sum>9.6 && sum<=10.4){
            if(min>8.5 && max<11.6)
                return "Mirovanje";
            return "Gibanje";
        }else if(sum>10.4 && sum<=12.9){
            if(min>1.5 && max<40.0)
                return "Hoja";
            return "Hoja"; //hitra hoja
        }else if(sum>12.9 && sum<=17.0){
            if(min>0.35 && max<55.0)
                return "Tek";
            return "Tek"; //hiter tek
        }
        return "NA";
    }
```

Izpis 4.3: Izsek kode za razpoznavo gibanja.

4.3 Algoritem za pridobivanje vrednosti preko GATT strežnika

O strukturi komunikacije in načinu delovanja strežnika GATT smo si lahko podrobneje prebrali v podpoglavju 2.1.2, kjer je zasnova predstavljena v teoriji. Več o dejanski implementaciji bomo opisali v nadaljevanju.

4.3.1 Definiranje algoritma

S storitvami in karakteristikami v protokolu GATT smo se že spoznali, zato vemo, da je karakteristika namenjena spremljanju določenega tipa meritve. V tem primeru imamo Heart Rate Mesurement, ki je definiran z edinstvenim ključem `00002a37-0000-1000-8000-00805f9b34fb`. Na podlagi ključa se določi spremljanje karakteristike in njenih vrednosti, ki so predstavljene s šestnajstiško obliko v posebnem strukturnem formatu. Tako imamo na prvem mestu po pravilu debelega konca zastavice. Te nam za Heart Rate Mesurement določajo format zapisa UNIT8 ali UNIT16. Če je zastavica nastavljena na 0 se uporabi format UINT8, sicer UINT16.

Primer heksadecimalnega zapisa vrednosti za Heart Rate Measurements.

```
Format UNIT8: Zastavica:Utripa:RR-interval:RR-interval
Realne vrednosti: 16:3E:19:04
```

Kako odčitamo vrednost zastavice?

Ker je vrednost zastavice predstavljena v heksadecimalni obliki, to pretvorimo v binarno (4.3) in pogledamo bit LSB (angl. least significant bit).

$$16_{hex} = 00010110_{bin} \tag{4.3}$$

Vrednost zastavice po LSB je 0, kar definira format zapisa UNIT8.

Kako določimo vrednost utripa?

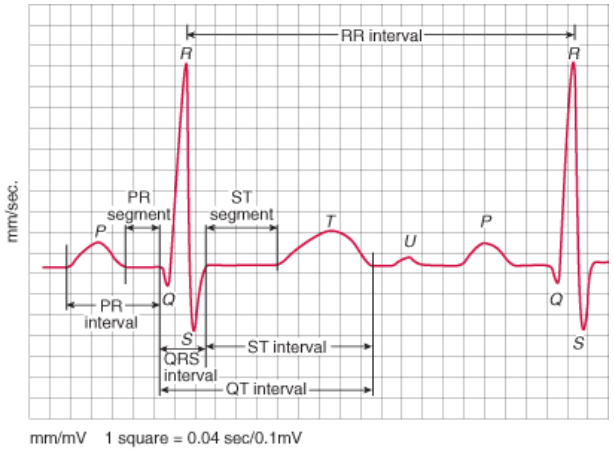
Vrednost za zastavicami določa srčni utrip. Pri danem primeru je ta vrednost 3E zapisana v heksadecimalni obliki, ki jo samo pretvorimo v desetiško vrednost (4.4) in dobimo število utripov na minuto.

$$3E_{hex} = 62_{dec} \tag{4.4}$$

Sledijo še vrednosti 19 in 04, ki predstavljajo RR interval, zapisan v obliki LSB (4.5). RR interval definira čas med dvema R valovoma, ki pripadata kompleksnosti QRS, ki nastaneta med srčnim utripom (Slika 4.4). Vsak srčni utrip sestoji iz treh valov: Q, R in S, ki določajo obliko amplitud. RR interval se v aplikaciji ne uporablja, je pa lahko uporaben za druge analize srčnega signala [22].

$$0419_{hex} = 10000011001_{bin} = 1049_{dec} \tag{4.5}$$

RR interval je predstavljen v milisekundah, kar v podanem primeru ustreza približno eni sekundi.



Slika 4.4: Prikaz RR intervala na signalu srčnega utripa [22].

4.3.2 Implementacija algoritma

Glede na opisan postopek smo definirali algoritem, ki razpozna obliko formata zapisa in zna heksadecimalne vrednosti pravilno interpretirati.

```
1  private void broadcastUpdate(final String action, final
       BluetoothGattCharacteristic characteristic) {
2
3   final Intent intent = new Intent(action);
4   int heartRate;
5   String HR_char=characteristic.getUuid().toString();
6
7   if (HR_char.equals("00002a37-0000-1000-8000-00805f9b34fb")) {
8    int flag = characteristic.getProperties();
9
10   if (flag != 16)
11    heartRate = characteristic.getIntValue(
       BluetoothGattCharacteristic.FORMAT_UINT16, 1);
12   else
13    heartRate = characteristic.getIntValue(
       BluetoothGattCharacteristic.FORMAT_UINT8, 1);
14
15   intent.putExtra("HEART_RATE",Integer.toString(heartRate));
16  }
17  sendBroadcast(intent);
18 }
```

Izpis 4.4: Izsek kode za razpoznavo gibanja.

Poglavje 5

Razvoj mobilne rešitve

5.1 Ideja

Ideja za razvoj opisane rešitve je nastala iz povsem človeških situacij. Ljudje se velikokrat znajdejo v položaju, kjer zaradi narave primera niso sposobni poklicati pomoč. V ta namen je bila razvita mobilna aplikacija meProtect, ki lahko ob padcih in srčnih zastojih samodejno obvesti svojce. Na trgu sicer obstajajo podobne aplikacije, ki pa ne zagotavljajo samodejnega obveščanja in so zato v primeru srčnega zastoja neuporabne.

5.2 Metodologija

Pred samim razvojem omenjene rešitve smo pregledali in analizirali trg s podobnimi primeri uporabe. Glede na izsledke so bile definirane ustrezne funkcionalnosti in okolje razvoja, ki ga obstoječe rešitve ne pokrivajo. Aplikacija meProtect je razvita za sistem Android 6.0 (Marshmallow), zaradi širše pokritosti naprav (Slika 5.1). Omenjeni sistem sicer pokriva 32.3 % vseh naprav, kar je relativno malo, a trenutno največ izmed vseh različic platforme [1]. Določena kompatibilnost je sicer mogoča tudi z novejšimi in starejšimi različicami sistema, ki procent pokritosti dvignejo na polovico.

Version	Codename	API	Distribution
2.3.3 - 2.3.7	Gingerbread	10	0.7%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	0.7%
4.1.x	Jelly Bean	16	2.7%
4.2.x		17	3.8%
4.3		18	1.1%
4.4	KitKat	19	16.0%
5.0	Lollipop	21	7.4%
5.1		22	21.8%
6.0	Marshmallow	23	32.3%
7.0	Nougat	24	12.3%
7.1		25	1.2%

Slika 5.1: Pokritost android različic [1].

Za integracijo zunanjih tehnologij (senzorjev) smo upoštevali več naprav. Temeljito smo preučili možnosti implementacije in dostopno dokumentacijo s strani proizvajalcev. Tovrstne tehnologije zahtevajo določene protokole za pravilno delovanje, ki v primerih slabe dokumentacije otežujejo razvoj. Zato je bila dobra izbira ključnega pomena za uspešno integracijo.

Po analizi vseh tehničnih ovir je sledil razvoj aplikacije v Android okolju. Najprej smo na podlagi grafičnih izrisov definirali vizualno podobo rešitve, ter zatem še vse operacije v ozadju. Za razvoj smo uporabili agilni pristop v kombinaciji z inkrementalnim in prototipnim modelom, ki sta glede na naravo primera najbolj ustrezna. Ponujata namreč način razvoja podsistemov, ki so lahko samo prototipno zasnovani in gredo skozi več faz, preden se integrirajo v končno rešitev. Agilni pristop nam daje možnost sprotnega planiranja in večjo fleksibilnost pri spreminjanju razvoja opravil, ki se jih vnaprej težko definira.

5.3 Opis funkcionalnosti in zahtev

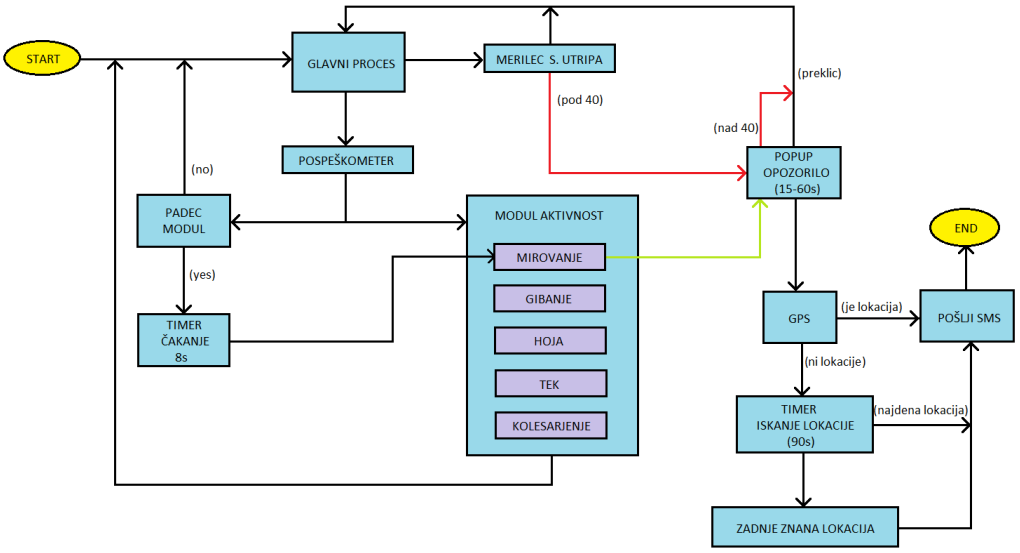
Za željen primer uporabe je bilo potrebno razviti aplikacijo, ki ima visoko stopnjo funkcionalnosti. Ena od glavnih funkcionalnosti, ki igra ključno vlogo pri uporabnosti aplikacije, je avtonomno pošiljanje obvestil v sodelovanju s senzorji. Ta funkcionalnost odpravlja pomanjkljivost podobnih aplikacij, ki omogočajo samo okolje s hitrejšim in enostavnejšim obveščanjem od navadnega telefonskega klica. V aplikacijo so implementirane še funkcionalnosti spremljanja srčnega utripa, pregled nad prepotovano potjo, zaznavanje padca, določanje gibajočega se stanja osebe in poraba kalorij. Te funkcionalnosti razširjajo uporabnost in omogočajo aplikaciji poleg nadziranje osebe tudi prikaz podatkov uporabniku v realnem času. Za tovrstne funkcionalnosti aplikacija zahteva uporabo senzorja Bluetooth Low Energy, vgrajen pospeškometer in modul GPS, ki jih načeloma vsebujejo vse novejše mobilne naprave. Bluetooth Low Energy je vgrajen v vse naprave z operacijskimi sistemi Android 4.3 naprej, ki je pogoj za uporabo aplikacije. Za pravilno in natančno delovanje se priporoča uporaba senzorja Polar H7, na katerem je temeljilo načrtovanje modula Heart Rate. Aplikacija dovoljuje uporabo tudi brez tovrstnega senzorja, vendar z zmanjšano funkcionalnostjo. Dobra lastnost in ena od zahtev je delovanje aplikacije brez podatkovnega prometa, saj ta ni dostopen povsod in bi morebitna zahteva po uporabi interneta bistveno zmanjšala uporabnost rešitve. Internetna povezava se tako uporablja zgolj za nalaganje zemljevida, ki pa ni primarna naloga aplikacije in ne vpliva na pravilno delovanje. Glavni cilj je zagotoviti pošiljanje sporočila SMS z vzrokom in lokacijo osebe, potrebne pomoči. Za to pa potrebujemo povezavo z mobilnim omrežjem in satelitom GPS, sicer se sporočilo ne more poslati oziroma je njegova vsebina pomanjkljiva.

5.4 Delovanje mobilne rešitve meProtect

Mobilna rešitev meProtect ima dva povsem ločena modula (Slika 5.2), ki nadzorujeta psihofizično stanje osebe. Prvi modul se povezuje z zunanjim senzorjem srčnega utripa Bluetooth Low Energy (Polar H7). Drugi pa na pospeškometer, ki je vgrajen v mobilni napravi. Takšna zasnova dovoljuje delovanje aplikacije tudi v primeru, ko kakšen od senzorjev ni na voljo. Glavna naloga obeh modulov je pridobivanje in dostava podatkov aplikaciji v določenem časovnem intervalu. Na ta način je zagotovljena konsistentnost podatkov, poslanih v obdelavo algoritmom za odločanje in prikaz človeku razumljivih vrednosti. Podatki, prejeti iz modula, ki se povezuje na zunanji senzor, predstavljajo srčni utip. Ti podatki se osvežujejo in nadzirajo z algoritmom, ki skrbi, da vrednost ne pade pod 40 udarcev, sicer se sproži alarmno sporočilo z zvočnim signalom. Alarmno sporočilo je aktivno 15–60 sekund in je odvisno od izbire uporabnika. V kolikor se utrip v tem časovnem intervalu zviša nad 40 udarcev, se alarmno sporočilo samodejno prekliče. Uporabnik ima tudi možnost ročnega preklica, v kolikor se izkaže, da gre za napačno presojo sistema. V primeru, da se alarmnega sporočila ne prekliče, algoritem začne izvajati postopek pošiljanja sporočila SMS. Za izvedbo pošiljanja morata biti izpolnjena dva pogoja:

1. pogoj: Uporabnik mora imeti izbrane stike, katerim se posreduje sporočilo,
2. pogoj: Znana oz. pridobljena trenutna lokacija.

V kolikor trenutna lokacija osebe ni znana, algoritem skuša to pridobiti, sicer se po izteku 90-sekundnega intervala uporabi zadnja znana lokacija. Lokacija je v sporočilu podana kot povezava na Googlov zemljevid, da prejemnik sporočila lahko nemudoma locira osebo.



Slika 5.2: Graf prehodov in delovanja aplikacije meProtect.

Pri vgrajenem sistemu je postopek pošiljanja enak, razlika je le v algoritmu, ki pridobiva podatke iz pospeškometra in na podlagi teh določa gibalni položaj osebe. V primeru padca, algoritem še osem sekund spremlja stanje osebe. Če je ta v mirovanju, se izvede prikaz alarmnega sporočila, in v kolikor ne pride do ročnega preklica, se izvede postopek pošiljanja sporočila SMS.

Ko aplikacija enkrat pošlje sporočilo SMS, teh znotraj iste seje ne pošilja več (Slika 5.2). Takrat je potrebno aplikacijo ponovno zagnati.

5.4.1 Omejitve mobilne rešitve

Omejitve, ki so vezane na delovanje aplikacije, se delijo v dve različni skupini. Na eni strani imamo komunikacijske sisteme, s katerimi se povezuje aplikacija (GPS, BLE, ISP), na drugi pa okolje oziroma platforme, na katerih se aplikacija izvaja. Komunikacijski sistemi niso 100 % zanesljivi. Pokritost s telefonskim omrežjem in signalom GPS ni na voljo povsod, zato se lahko ob takšnih situacijah funkcionalnost aplikacije zmanjša. Tovrstne napake

so v aplikaciji sicer predvidene in na delovanje ne vplivajo. Vplivajo pa na uporabnost aplikacije, saj so podatki lahko netočni oziroma se ne posredujejo. Takšne omejitve lahko le predvidimo in nanje ustrezno reagiramo, ne moremo pa na njih vplivati, kot lahko vplivamo na omejitve platform. Te predstavljajo manjši problem, saj se točno ve, za katere platforme se aplikacija gradi in je delovanje aplikacije na isti platformi za različne naprave enako.

Znane omejitve za razvito aplikacijo so:

- uporaba mobilne naprave v načinu "Varčevanje z energijo",
- uporaba v notranjih prostorih,
- izvajanje aplikacije na platformi Android 7.0.

5.5 Strukturna zasnova aplikacije

To poglavje bo namenjeno podrobnejši predstavitvi tehničnega delovanja in zasnove mobilne rešitve meProtect. Opisane bodo posamezne povezave med javanskimi razredi, strukturna sestava in prakse, ki so bile uporabljene za razvoj. Predstavitev bo temeljila na delujoči končni različici z vsemi pripadajočimi funkcionalnostmi, opisanimi v podpoglavju 5.3.

5.5.1 Struktura razredov

Aplikacijska struktura vsebuje šest aktivnosti in trinajst pomožnih razredov, ki skupaj tvorijo zgradbo aplikacije. Vsaki aktivnosti pripada najmanj ena datoteka XML (angl. eXtensible Markup Language) z izgledom. Skupaj je aplikacija sestavljena iz šestih osnovnih in trinajstih pomožnih datotek za izgled (Tabela 5.1).

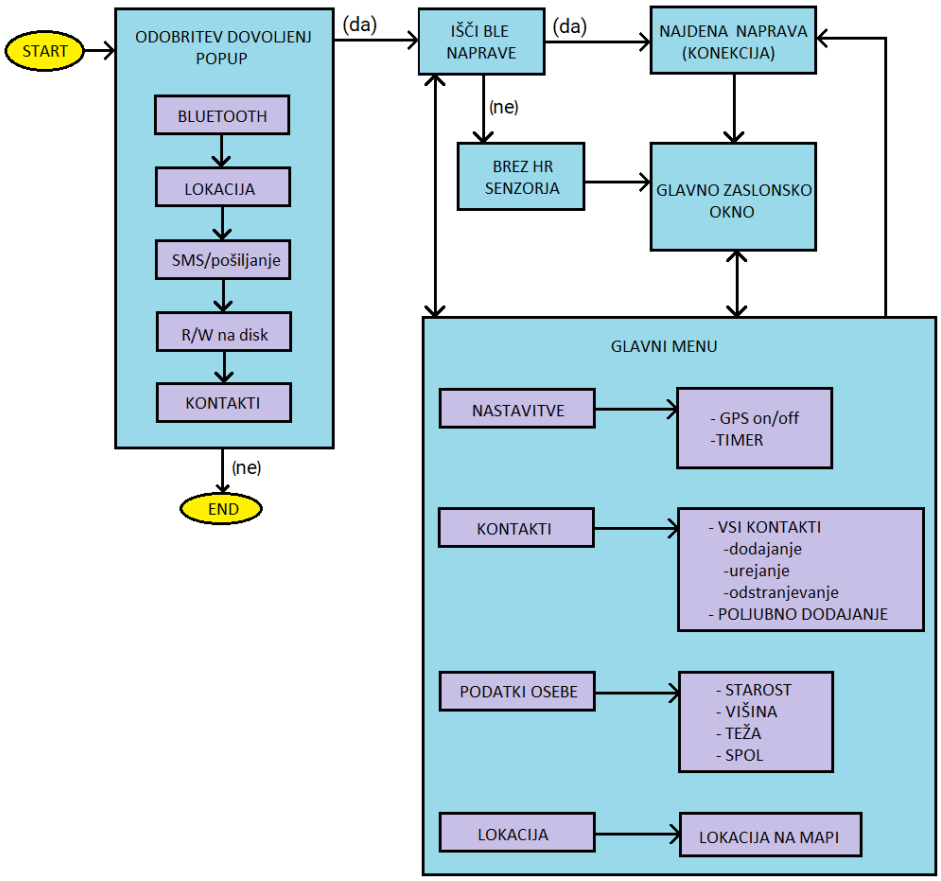
Razredi	**Pripadajoč izgled**
AppCompatPreferenceActivity.java	/
BTLE_Devices.java	/
BluetoothLeService.java	/
ContactActivity.java	activity_contact.xml content_contact.xml contact_delete_menu.xml contacts_option_menu.xml contact_delete.xml
ContactsInfo.java	/
CustomBleAdapter.java	custom_ble_listview.xml
CustomContactAdapter.java	custom_contacts_listview.xml
DeviceControlActivity.java	activity_device_control.xml content_device_control.xml disconnect_menu.xml navigation_headers.xml app_bar_main.xml app_menu.xml
DeviceScanActivity.java	activity_device_scan.xml settings_menu.xml
FallService.java	/
GPS_Location.java	/
MapsActivity.java	activity_maps.xml info_marker_window.xml
MapUserData.java	/
PopUpMessage.java	/
Scanner_BTLE.java	/
SettingsActivity.java	pref_headers.xml
SMSsend.java	/
TimerEvent.java	/
UserInfoActivityes.java	activity_user_info.xml

Tabela 5.1: Seznam razredov in pripadajočih izgledov aplikacije.

Razreda `FallService.java` in `BluetoothLeService.java` smo spoznali že v poglavju 4. Zasnovana sta kot procesa, ki se izvajata v ozadju, ločeno od glavnega procesa. Takšen tip zasnove razbremenjuje glavni proces, saj se vse zahtevnejše operacije izvajajo ločeno.

5.5.2 Predstavitev aktivnosti

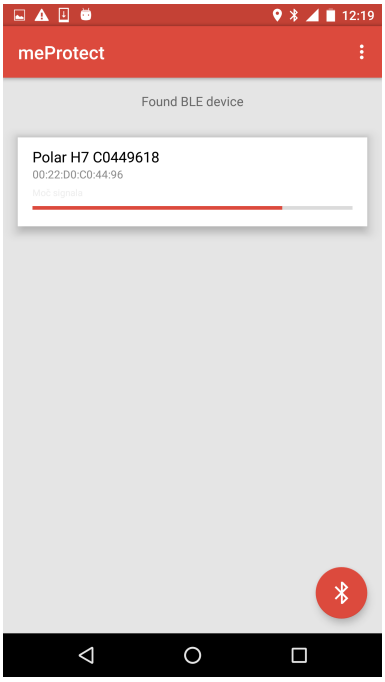
Izraz aktivnost se uporablja pri načrtovanju Android aplikacij za pojavno okno na zaslonu naprave. Aplikacija meProtect sestoji iz šestih aktivnosti, ki skupaj s programsko logiko definirajo izgled in funkcionalnost aplikacije (Slika 5.3).



Slika 5.3: Prehajanje med aktivnostmi v aplikaciji.

DeviceScanActivity

Gre za prvo aktivnost, ki se pojavi ob zagonu aplikacije. S pomočjo te aktivnosti se mobilna naprava lahko poveže z zunanjimi senzorji Bluetooth Low Energy. V tem primeru je povezava omejena na senzorje srčnega utripa (Slika 5.4).



Slika 5.4: Izgled aktivnosti DeviceScanActivity.

Naloge aktivnosti:

- iskanje razpoložljivih naprav BLE,
- prikaz vseh naprav BLE v dosegu,
- povezovanje z želeno napravo BLE.

Zgradba aktivnosti

Na `DeviceScanActivity.java` se povezujejo trije pomožni razredi:

- `Scanner_BTLE.java`,
- `CustomBleAdapter.java`,
- `BTLE_Device.java`.

Razred `Scanner_BTLE.java` od naprave zahteva uporabo BLE za iskanje razpoložljivih naprav v okolici. Vse najdene naprave se shranjujejo v ArrayAdapter, ki je modificiran v `CustomBleAdapter.java`. Prednost takšne modifikacije je lažje upravljanje z objekti v adapterju. Omogoča nam enostaven dostop in izpis atributov objekta v gradnike na zaslonu. ArrayAdapter, ki ga CustomBleAdapter razširja, dovoljuje namreč hranjenje objektov tipa `BTLE_Device`, ki jih definira razred `BTLE_Device.java`. Tako se vsaka najdena naprava preoblikuje v objekt `BTLE_Device`.

Atributi razreda `BTLE_Device.java`:

- ime naprave,
- MAC naslov,
- RSSI moč signala.

Težava, ki se pojavi pri iskanju razpoložljivih naprav, je, da se ena naprava v časovnem intervalu iskanja najde večkrat. To privede do duplikatov v ArrayAdapterju, kar je za uporabnika moteče. Za preprečevanje dupliciranja aplikacija uporablja algoritem, ki vsem že najdenim napravam znotraj ArrayAdapterja pregleda MAC naslove in primerja z najdenim. V kolikor najde ujemanje se ta naprava ne doda na seznam. Takšen način preverjanja nam dovoljuje razred `BTLE_Device.java`, ki za vsak nov objekt hrani MAC naslov.

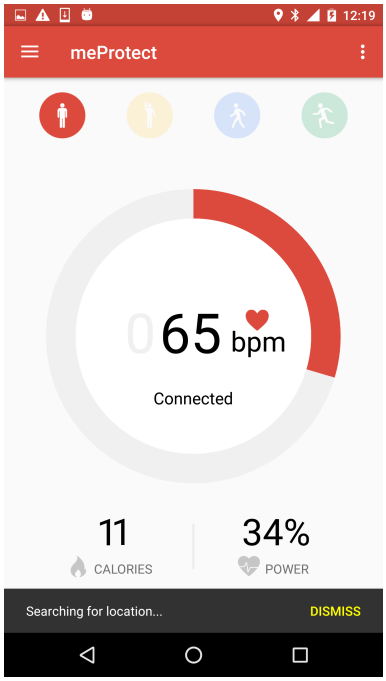
```
public void addDevice(BluetoothDevice device, int new_rssi){
        boolean contains=false;
        BTLE_Device btle_device;

        for(BTLE_Device d : mBTDeviceArrayList){
            if(d.getAddress().equals(device.getAddress())){
                d.setRSSI(new_rssi);
                contains=true;
                break;
            }
        }
        if(!contains){
            btle_device = new BTLE_Device(device);
            btle_device.setRSSI(new_rssi);
            mBTDeviceArrayList.add(btle_device);
        }

        contains=false;
        adapter.notifyDataSetChanged();
        if(mBTDeviceArrayList.size()>0)
            action.setText("Found BLE device");

}
```

Izpis 5.1: Izsek kode za dodajanje nedupliciranih naprav v ArrayAdapter.

DeviceControlActivity

Je glavna aktivnost, na katero so vezane vse ostale. Zaradi višje kompleksnosti so nekateri postopki računanja umaknjeni iz te aktivnosti in se izvajajo preko zunanjih storitev. S tem se izognemo preveliki dejavnosti na eni niti in izboljšamo uporabniško izkušnjo. Prevelika obremenitev glavne niti namreč lahko vpliva na odzivnost aplikacije, kar na manj zmogljivih napravah lahko povzroči sesutje aplikacije. Ta aktivnost se uporablja predvsem za prikaz vseh merljivih podatkov senzorjev, ki so preko algoritmov preračunani in podani v uporabniku razumljivi obliki (Slika 5.5). V aktivnost je imple-

mentiran tudi stranski meni, ki ga Google priporoča pri gradnji aplikacije v "Material" dizajnu zavoljo ohranjanja konsistentnosti med aplikacijami.



Slika 5.5: Izgled aktivnosti DeviceControlActivity.

Naloge aktivnosti:

- Prikaz aktivnosti osebe.
- Prikaz srčnega utripa.
- Prikaz porabe kalorij glede na srčni utrip, starost, težo in spol.
- Prikaz obremenitve srca.
- Proženje opozoril ob zaznavi psihofizičnih težav osebe.

Zgradba aktivnosti

Na `DeviceControlActivity.java` se povezujejo štirje pomožni razredi in dve storitvi:

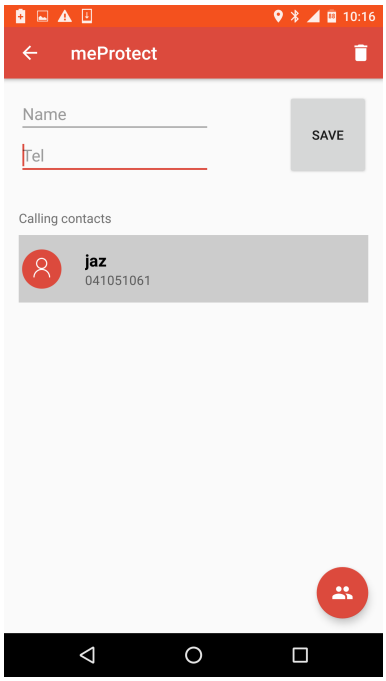
- `BluetoothLeService.java`,
- `FallService.java`,
- `TimerEvent.java`,
- `GPS_Location.java`,
- `PopUpMessage.java`,
- `SMSsend.java`.

`BluetoothLeService.java` in `FallService.java` sta storitvi, ki glavni aktivnosti pošiljata preračunane podatke iz senzorjev (vrednost srčnega utripa in gibalnega pospeška). Vsem prejetim podatkom razred `TimerEvent.java` določi časovne intervale, ki aplikaciji omogočajo lažje proženje obvestil v primerih odstopanj vrednosti od normale. Ena od posledic odstopanja vrednosti je javljanje preko razreda `PopUpMessage.java`, ki z glasovnim signalom in opozorilom na zaslonu sporoča uporabniku, da je prišlo do prevelikega odstopanja, bodisi zaradi napake na senzorju ali resnega stanja osebe.

Razred `GPS_Location.java` ima dva načina delovanja. Lahko spremlja lokacijo osebe od zagona aplikacije oziroma od zahteve razreda `SMSsend.java` po pridobitvi lokacije. Razred `SMSsend.java` to zahtevo sproži samo v kolikor se oseba ne odzove na opozorila aplikacije. S tem zagotovi polno vsebino sporočila SMS, ki se posreduje nastavljenim osebam.

ContactActivity

ContactActivity je aktivnost, namenjena nastavitvi kontaktov za obveščanje v primeru nujne pomoči. Aktivnost neposredno ne sodeluje z ostalimi, temveč zgolj posredno preko datoteke za shranjevanje kontaktov contactList.txt, kjer pripravi podatke v obliki, ki je primerna za nadaljno uporabo v drugih aktivnostih (Slika 5.6).



Slika 5.6: Izgled aktivnosti ContactActivity.

Naloge aktivnosti:

- Prikaz in urejanje nastavljenih kontaktov za obveščanje.
- Prikaz in dodajanje kontaktov iz telefonskega imenika.
- Dodajanje poljubnega kontakta.

Zgradba aktivnosti

Na `ContactActivity.java` se povezujeta dva pomožna razreda:

- `ContactsInfo.java`.
- `CustomContactAdapter.java`.

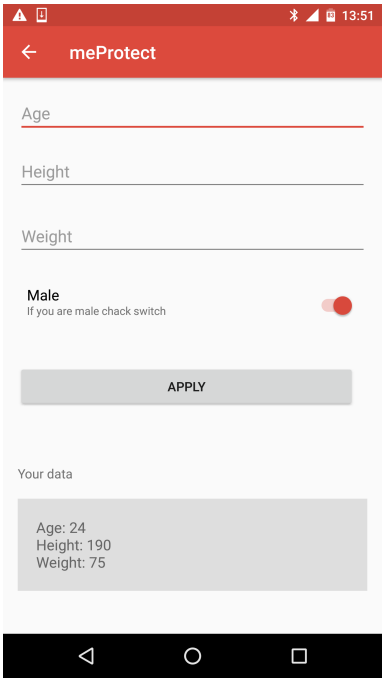
Za formiranje podatkov o kontaktih skrbi razred `ContactsInfo.java`, ki definira objekt z atributi o telefonski številki, imenu in profilni sliki kontakta. `ContactsInfo` objekti se shranjujejo v ArrayAdapter, preko katerega za lažji in preglednejši izpis podatkov skrbi razred `CustomContactAdapter.java`, ki razširja osnovni ArrayAdapter. Na `CustomContactAdapter.java` je implementirana funkcija `OnItemLongClickListener()`, ki omogoča novejšo obliko urejanja in dodajanja objektov v ArrayAdapter z dolgim klikom na želen objekt.

```
contacts.setOnItemLongClickListener(new AdapterView.
    OnItemLongClickListener() {
 @Override
 public boolean onItemLongClick(AdapterView<?> adapterView, View
     view, int pos, long l) {
  Object o = (Object) pos;

  if(contactInfoArrayList.size()>0)
    ActivityCompat.invalidateOptionsMenu(ContactActivity.this);
  else if(allContact.size()>0)
    ActivityCompat.invalidateOptionsMenu(ContactActivity.this);

  if (selectedContactsItem.contains(o)) {
   selectedContactsItem.remove(o);
   view.setBackgroundColor(Color.TRANSPARENT);
  } else {
   selectedContactsItem.add(pos);
   view.setBackgroundColor(Color.LTGRAY);
  }

  return true;
 }
});
```

Izpis 5.2: Izsek kode, ki omogoča izbiranje v ArrayAdapterju.

UserInfoActivity

Aktivnost, ki služi vnosu osebnih podatkov osebe z namenom natančnejše senzorske meritve. Vnešeni podatki se shranijo v datoteko userData.txt, katera se uporablja za nastavitev parametrov. Aktivnost je neodvisna in ne komunicira z nobeno drugo. Prav tako ne uporablja zunanjih razredov za delovanje (Slika 5.7).



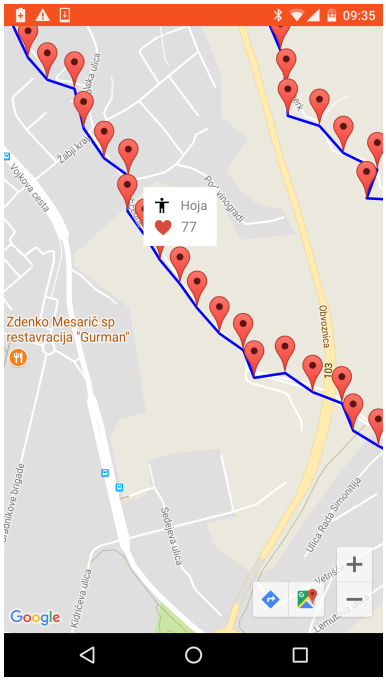
Slika 5.7: Izgled aktivnosti UserInfoActivity.

Naloge aktivnosti:

- Vnos osebnih podatkov.
- Spreminjanje osebnih podatkov.
- Shranjevanje osebnih podatkov.

MapsActivity

MapsActivity je aktivnost, ki v svojo zgradbo implementira Googlov maps API (angl. Application programming interface). Preko tega API-ja se aktivnosti omogoči uporabo in prikaz zemljevida, kjer si uporabnik lahko pregleda prepotovane poti in psihofizično stanje na posameznih odsekih (Slika 5.8). Podatke za prikaz poti in psihofizičnega stanja pridobi iz datoteke userLocation.txt, ki se upravlja preko razreda `GPS_Location.java`.



Slika 5.8: Izgled aktivnosti MapsActivity.

Naloge aktivnosti:

- Nalaganje zemljevida.
- Prikaz prepotovane poti.
- Prikaz srčnega utripa in aktivnosti na poti.

Zgradba aktivnosti

Na `MapsActivity.java` se povezuje pomožni razred:

- `MapUserData.java`

Zaradi uporabe Maps API-ja je potrebno za aplikacijo generirati 39-bitni ključ preko Googlovega računa[1]. Tako lahko Google spremlja in sledi uporabi

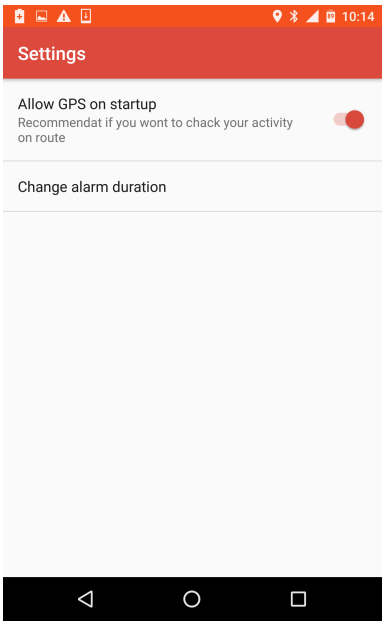
[1]https://console.developers.google.com.

njihove storitve. Maps API ima namreč omejitev na 1000 prikazov lokacij na dan. Za večjo uporabo Google storitev zaračuna oziroma naredi nedostopno. Generirani ključ se zapiše v datoteko AndroidManifest.xml pod metadata in tako zagotovi API-ju potreben podatek za beleženje uporabe.

```
<meta-data
android:name="com.google.android.geo.API_KEY"
android:value="@string/google_maps_key" />
```

SettingsActivity

Nekatere aktivnosti so že definirane v okolju Android Studio, med drugim tudi aktivnost SettingsActivity. Aktivnost dovoljuje uporabniku spreminjanje določenih nastavitev v aplikaciji (vklop GPS-a ob zagonu, spreminjanje dolžine alarmnega sporočila). Ker se te nastavitve upravljajo preko posebnega razreda SharedPreferences, je zaradi lažje implementacije ta aktivnost že prenastavljena in je ni potrebo v celoti programirati (Slika 5.9).



Slika 5.9: Izgled aktivnosti SettingsActivity.

Naloge aktivnosti:

- Prikaz nastavitev.
- Beleženje nastavljenih vrednosti.

5.5.3 Načini shranjevanja

Ker se aplikacija ne povezuje na zunanje podatkovne baze, se vsi podatki zapisujejo lokalno na interni oziroma zunanji pomnilnik mobilne naprave. Razlika med internim in zunanjim pomnilnikom je v dostopu do podatkov. V kolikor se podatki shranjujejo v interni pomnilnik, ima dostop do njih samo aplikacija in so za uporabnika naprave nevidni. S tem se prepreči uporabniku spreminjanje strukture zapisa podatkov. katere aplikacija uporablja za pravilno delovanje. Prednost zapisa na interni pomnilnik je tudi uničenje vseh podatkov ob morebitnem izbrisu aplikacije in s tem ohranjanje čistosti naprave. Aplikacija sicer uporablja še en poseben tip shranjevanja, ki ga ponuja sistem Android, imenovan SharedPreferences. Ravno tako gre za shranjevanje na interni pomnilnik, vendar je ta zagotovljen s strani operacijskega sistema. Namenjen je shranjevanju vrednosti nastavitev v aplikaciji, ki jih uporabnik sam nastavi.

Podatke, ki jih aplikacija shranjuje, so zapisani v tekstovnih datotekah v posebej prirejenih formatih zapisa. Razlog za uporabo posebnih formatov je določitev konsistentnosti zapisa, ki nam dovoljuje obdelavo datotek v algoritmih.

Datoteke, ki jih aplikacija uporablja za shranjevanje:

- contactList.txt,
- userData.txt,
- userLocation.txt.

Opis datotek za shranjevanje

contactList.txt je datoteka, ki vsebuje izbrane kontakte uporabnika za obveščanje v primeru nujne pomoči.

```
format zapisa: ime in priimek;telefonska številka;slika
```

Vsaka nova vrsta pomeni nov kontakt. Kontaktov je lahko poljubno število.

userData.txt je datoteka, v kateri se vodijo nekateri osebni podatki osebe, ki uporablja aplikacijo. Ti podatki niso nujno potrebni, saj aplikacija deluje tudi brez njih. V kolikor se uporabnik odloči, da bo svoje podatke shranil, se na podlagi teh spremenijo določeni parametri, ki omogočajo bolj natančno delovanje. Sicer aplikacija uporabi prevzete podatke, ki ustrezajo povprečni osebi.

```
format zapisa: starost;višina;teža;spol
```

Starost in spol vplivata na izračun porabe kalorij in izračun trenutne obremenitve srca, medtem ko se teža uporablja za določitev sile pri padcu. Algoritem za merjenje kalorij deluje samo v prisotnosti senzorja srčnega utripa, v nasprotnem primeru se poraba kalorij ne beleži. Za izračun se uporabljata posebni formuli (5.1), (5.2), ki v izračunu upoštevata tudi spol osebe, pridobljen iz datoteke `userData.txt`.

Formula za izračun porabljenih kalorij [11]:

Moški spol

$$Cal = (\frac{(-55.0969 + (0.6309 \times R) + (0.1988 \times W) + (0.2017 \times A))}{4.184}) \times 60 \times t \tag{5.1}$$

Ženski spol

$$Cal = (\frac{(-20.4022 + (0.4472 \times R) - (0.1263 \times W) + (0.074 \times A))}{4.184}) \times 60 \times t \tag{5.2}$$

Parametri:

R: povprečni srčni utrip,

W: teža osebe v kg,

A: starost,

t: čas v urah.

Formula za izračun srčne obremenitve v odstotkih (HP) [13]:

$$HP = \frac{bpm \times 100}{(208 - (\frac{7}{10} \times A))} \tag{5.3}$$

Parametra:

bpm: srčni utrip,

A: starost.

userLocation.txt je datoteka, ki se ob novi seji pobriše. V njej se beležijo podatki o lokaciji uporabnika, kateri so posredovani v Googlov API za izris prepotovane poti na zemljevidu. Poleg koordinat so zraven še zapisi o tipu aktivnosti in srčnem utripu na določeni lokaciji, ki se lahko poleg poti spremljajo na zemljevidu.

```
format zapisa: lat;long;tip aktivnosti;srčni utrip
```

Zaradi velike količine podatkov, generirane s strani sprejemnika GPS, je pridobivanje lokacije programsko omejeno. Omejitev je nastavljena tako, da sprejemnik GPS zahteva pridobitev lokacije na oddaljenosti 50 metrov od prejšnje znane lokacije. S tem se bistveno zmanjša število zapisov v datoteko userLocation.txt.

Poglavje 6

Testiranje

6.1 Mobilne naprave

Aplikacija je bila testirana na dveh mobilnih napravah: Nexus 6p in Lenovo Moto Z Play. Pogoj za pravilno delovanje je operacijski sistem Android 6.0 Marshmalow ter podpora mobilne naprave za Bluetooth Low Energy, različica 4.0. Aplikacija deluje brezhibno na napravi Nexus 6p, ki ustreza vsem predpisanim pogojem. Pri testiranju na napravi Lenovo Moto Z Play se je pojavila motnja v delovanju prenosa podatkov med napravo in senzorjem Polar H7. Vzrok je nova različica operacijskega sistema Android 7.0 Nougat, ki ne podpira enakega načina komunikacije kot Android 6.0. Omenjena težava naj bi se rešila z izdajo nove različice operacijskega sistema 7.1.

6.2 Tipi senzorjev

Integrirani senzorji, kot so pospeškometer in GPS, v aplikaciji delujejo brez težav. Možne so le težave v senzorjih srčnega utripa BLE, kjer se uporabljata dva tipa kodiranja podatkov za prenos: UINT8 in UINT16. Senzorji, ki uporabljajo UINT8, delujejo brez težav. Podpora za format UINT16 je v aplikacijo sicer implementirana, vendar ni testirana, ker je na tržišču teh senzorjev bistveno manj.

6.3 Obravnava prekinitev

Med delovanjem programov oziroma aplikacij pogosto prihaja do prekinitev, ki so generirane s strani operacijskega sistema, uporabnika ali drugih aplikacij. Ker gre za mobilno aplikacijo, je takšnih prekinitev še več in je zagotavljanje delovanja aplikacije, ko do njih pride, bistvenega pomena. Testirane so bile prekinitve, kot so pošiljanje aplikacije v ozadje, izguba povezave s senzorjem BLE, prekinitev ob prejetem klicu, izpad signala GPS in delovanje v stanju pripravljenosti. Testirane prekinitve niso povzročile napake v delovanju aplikacije, zaradi katerih bi prenehala delovati.

6.4 Ugotovitve

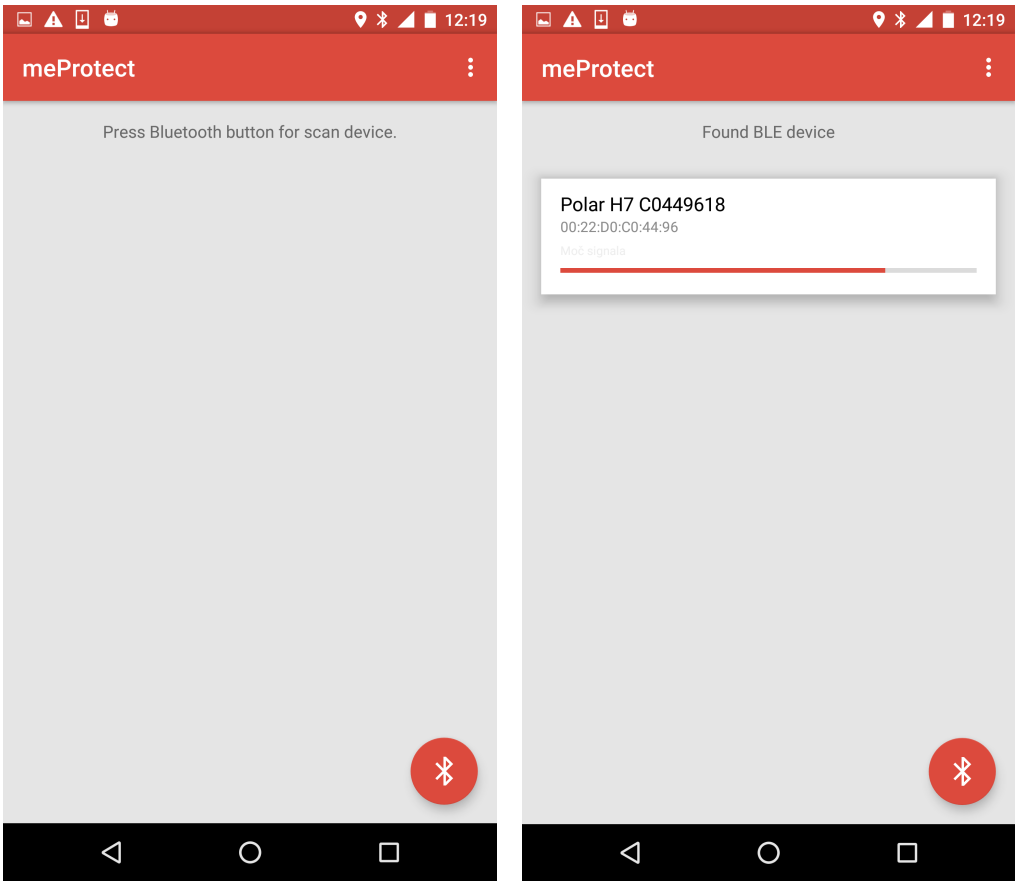
Aplikacija je večino testiranj prestala uspešno. Pri testiranju smo ugotovili še nekaj pomankljivosti, ki so posledica nekompatibilnosti ostalih sistemov z aplikacijo. Zagotavljanje podpore za tako veliko množico različnih sistemov je zelo težavno, če že ne nemogoče. Pomembno je omeniti eno posebnost sistema Android, ki vpliva na delovanje in je bila ugotovljena v fazi testiranja. Gre za nastavitev delovanja naprave v načinu z varčevanjem energije, ki ga lahko uporabnik aktivira. Ta način namreč preprečuje, da bi aplikacija pridobivala lokacijo v stanju pripravljenosti. Posledično se ne beleži lokacije in so poslani podatki v sporočilu SMS nepopolni.

6.5 Primer uporabe

Namen poglavja je predstavitev delovanja in uporabe aplikacije meProtect na realnem primeru, ki ustreza stanju od vklopa do samodejnega pošiljanja sporočila SMS.

Povezava z BLE senzorjem

Ob zagonu aplikacije imamo možnost vzpostavitve povezave z zunanjim senzorjem srčnega utripa BLE (Slika 6.1a). Z dotikom na `Floating Action Button` aktiviramo iskanje rezpoložljivih naprav v okolici (Slika 6.1b). Ob kliku na najdeno napravo se vzpostavi bluetooth povezava in aplikacija preide v glavno aktivnost.

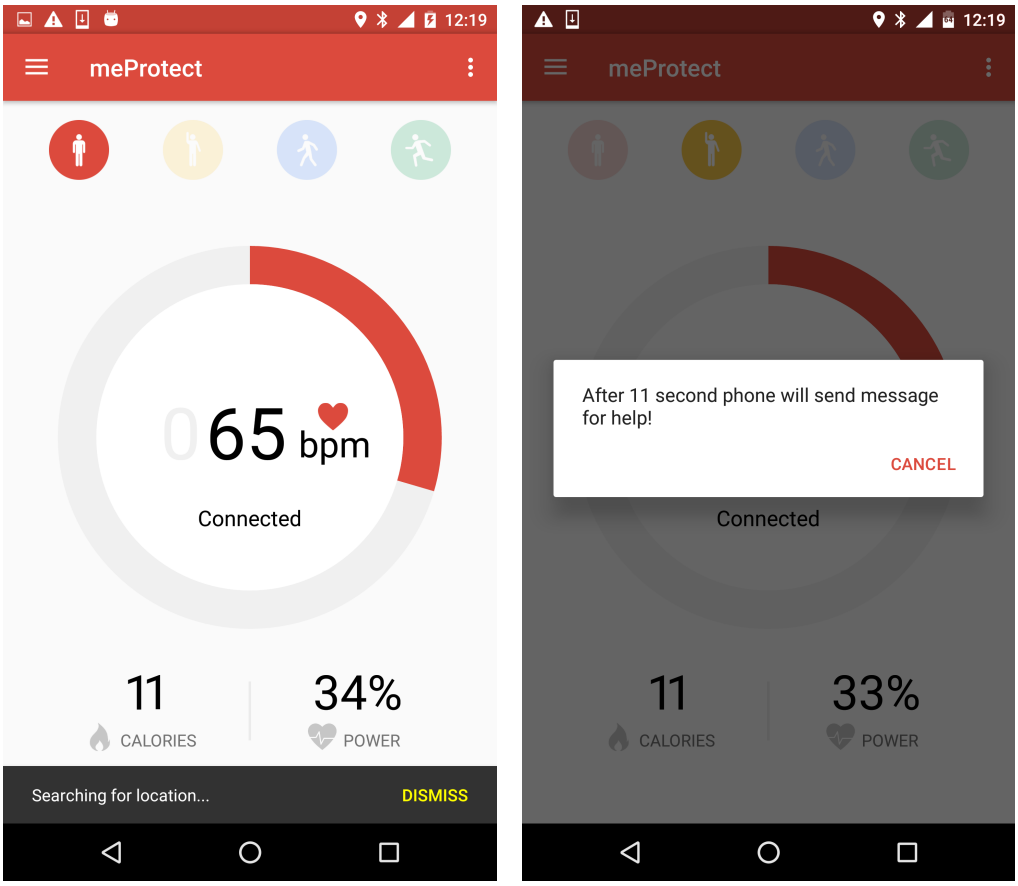


(a) Pred iskanjem BLE naprav (b) Po iskanju BLE naprav

Slika 6.1: Povezovanje s senzorjem (Polar H7).

Spremljanje in detektiranje

V glavni aktivnosti aplikacija prikazuje prejete senzorske podatke (Slika 6.2a). V kolikor nastopi padec oziroma srčni zastoj, se aplikacija odzove z opozoritvenim sporočilom (Slika 6.2b). Ob preklicu sporočila aplikacija nadaljuje z delovanjem, sicer se po izteku intervala posreduje sporočilo SMS.

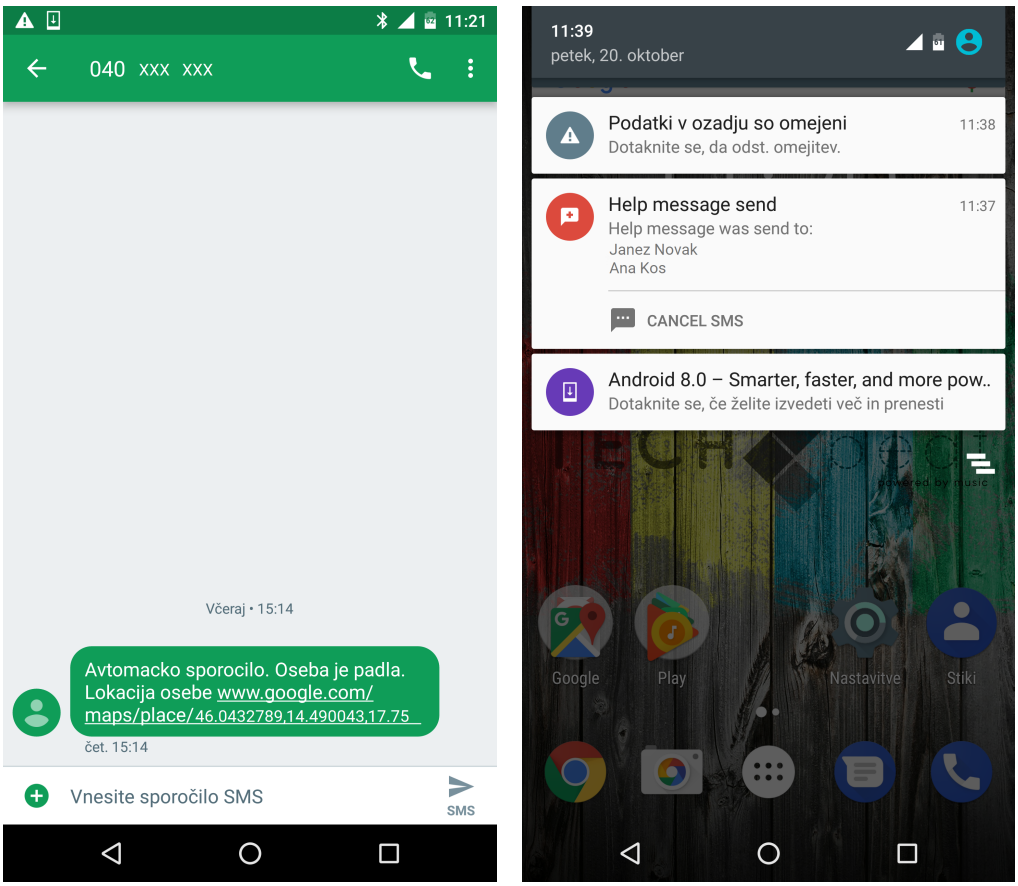


(a) Prikaz senzorskih vrednosti (b) Prikaz alarmnega sporočila

Slika 6.2: Aplikacija v delovanju in detekciji anomalije.

Poslano sporočilo SMS in preklic

Poslano sporočilo se na prejemnikovi napravi prikaže kot ostala sporočila SMS. Njegova vsebina vsebuje vzrok in aktivno povezavo z lokacijo do Googlovega zemljevida (Slika 6.3a). Po vsakem samodejno poslanem sporočilu ima možnost uporabnik med obvestili poslano sporočilo preklicati s pošiljanjem sporočila o preklicu vsem uporabnikom, ki so prejeli predhodnje sporočilo (Slika 6.3b).



(a) Poslano sporočilo SMS (b) Obvestilo o pošiljanju sporočila SMS

Slika 6.3: Oblika sporočila in obvestila.

Poglavje 7

Sklepne ugotovitve

V diplomskem delu je predstavljena mobilna rešitev za nadzor psihofizičnega stanja človeka, ki obravnava detekcijo padca in srčnega zastoja z uporabo senzorskih tehnologij. Pri razvoju so bile uporabljene novejše tehnologije, kot so Bluetooth Low Energy in nekatere že standardne (GPS, integrirani senzorji). Tovrstne tehnologije so pravzaprav dokaj kompleksne in posledično predstavljajo težjo implementacijo. Za integracijo z mobilno napravo zahtevajo veliko raziskovanja in razumevanja delovanja protokolov, ki so predpisani s strani globalnih združenj za standardizacijo. Dokumentacija, ki je na voljo razvijalcem pri uporabi omenjenih tehnologij, je težko razumljiva in primernejša za izkušene poznavalce, zato smo imeli kar nekaj težav v fazi definiranja primarnega ogrodja mobilne rešitve, saj je bilo njegovo delovanje temelj za nadaljni razvoj ostalih modulov.

Razvoj modula za detekcijo padca je predstavljal poseben izziv. Postopek namreč temelji na fazi procesiranja signala in fazi praktičnega testiranja. Testni podatki, pridobljeni z merjenjem simulacij padcev, so bili osnova za sestavo pravilno delujočega algoritma. Prav v segmentu konstrukcije algoritma je še prostor za izboljšave v delovanju mobilne rešitve. Že uporaba nosljivega pospeškometra bi pripomogla k natančnejšim podatkom, kot jih sedaj beleži integriran pospeškometer v mobilni napravi. Uporabljeni bi lahko bili tudi napredni pristopi analize in programiranja, kot sta umetna

inteligenca in strojno učenje.

Posebna pozornost je bila posvečena grafični zasnovi in uporabniški izkušnji. Čeprav je aplikacija namenjena delovanju v ozadju, se jo lahko uporablja tudi za monitoring srčnega utripa, pregled nad porabo kalorij, srčne obremenitve in trenutne dejavnosti. Zasnovana je na "Material" dizajnu, ki je posebna oblikovna struktura, razvita pri Googlu z namenom zagotavljanja enake *look and feel* izkušnje med aplikacijami. "Material" dizajn je možno uporabiti samo na novejših operacijskih sistemih Android, ker zahteva posebne gradnike, ki jih starejši ne podpirajo. Za lepšo ponazoritev podatkov srčnega utripa je poleg "Material" dizajna implementirana še knjižnica MPAndroidChart. Ta nam omogoča enostavnejši in procesorsko manj obremenjujoč izris "Donut" grafa, kot je ta z apliciranje slik in računanjem krožnega izseka. Optimizacija delovanja je v mobilnih aplikacijah veliko pomembnejša, saj se izvaja na strojni opremi, ki ni tako zmogljiva kot osebni računalniki. To še posebej velja za aplikacije, kot so meProtekt, ki procesirajo velike količine podatkov v realnem času in precej obremenjuje glavni proces.

Rešitve, kot so meProtect, zaradi povezljivosti z različnimi tehnologijami zahtevajo bistveno več testiranja, saj se zaradi kompleksnosti delovanja in nedostopnosti storitev lahko posredujejo napačne informacije drugim osebam. Tekom razvoja smo konstantno preverjali delovanje in odzivanje aplikacije na različne situacije, ki se lahko zgodijo med delovanjem. Zaznanih težav ni bilo, razen nedostopnosti GPS-a v zaprtih prostorih stavb zaradi izgube signala s satelitom. Omenjena težava bi bila lahko odpravljena z IPS (angl. Indoor Positioning System), ki pa je predmet drugih tem in tehnologij.

Izdelana aplikacija odpira nek nov pristop k nadzoru in varnosti tako starejših ko mlajših oseb, predvsem pa je usmerjena v področje, ki pridobiva vse večji pomen v družbi. Na "Wearable Health Technology" se v zadnjih letih daje precej poudarka in bi lahko pomembno vplivalo na dosedanje oblike zdra-

vljenj, oskrbe, nadzora in varnosti ljudi. Že dejstva, da nekatere bolnišnice vključujejo podobne aplikacije v svoje sisteme, obetajo temu področju vse večji razvojni potencial.

Razvita aplikacija meProtect je le ena od mnogih, ki delno posega v "Wearable Health Tehnology" in s tem dokazuje, da so na tem področju možne rešitve, ki odpravljajo celo vrsto težav povezane z zdravjem ljudi.

Literatura

[1] Razširjenost android platform. Dosegljivo: https://developer.android.com/about/dashboards/index.html. [Dostopano: 07. 09. 2017].

[2] Zunanji pospeškometer. Dosegljivo: http://wiki.aprbrother.com/wiki/ASensor. [Dostopano: 15. 10. 2017].

[3] B.S. Beauvais, Vincent Rialle, and Juliette Sablier. Myvigi: An android application to detect fall and wandering. pages 156–160, 01 2012.

[4] Delovanje ble komunikacije. Dosegljivo: https://devzone.nordicsemi.com/question/47911/current-measurement-in-nrf51422-development-kit/. [Dostopano: 08. 09. 2017].

[5] Bluetooth uradna dokumentacija. Dosegljivo: https://www.bluetooth.com/specifications/gatt/services. [Dostopano: 26. 09. 2017].

[6] Delovanje protokola gap. Dosegljivo: https://learn.adafruit.com/introduction-to-bluetooth-low-energy/gap. [Dostopano: 12. 09. 2017].

[7] Delovanje protokola gatt. Dosegljivo: https://learn.adafruit.com/introduction-to-bluetooth-low-energy/gatt. [Dostopano: 12. 09. 2017].

[8] Delovanje sistema gps. Dosegljivo: http://www.gps.gov/systems/gps/performance/accuracy/. [Dostopano: 20. 09. 2017].

[9] Polar senzor. Dosegljivo: http://fittechnica.co.uk/2013/08/polar-h7-heart-rate-sensor-review/. [Dostopano: 21. 09. 2017].

[10] Texas Instruments India. Bluetooth low energy l1 p1. Dosegljivo: https://youtu.be/YuEWZ20zDZI, 2017. [Dostopano: 20. 09. 2017].

[11] Formula za izračun kalorij. Dosegljivo: http://www.shapesense.com/fitness-exercise/calculators/heart-rate-based-calorie-burn-calculator.shtml. [Dostopano: 26. 09. 2017].

[12] Denis Laure. Heart rate measuring using mobile phone's camera. In *Proceedings of the 12th Conference of Open Innovations Association FRUCT and Seminar on e-Travel. Oulu, Finland*, pages 272–273, 2012.

[13] Formula za izračun obremenjenosti srca. Dosegljivo: https://www.runnersworld.com/ask-the-sports-doc/what-is-my-maximum-heart-rate. [Dostopano: 26. 09. 2017].

[14] Matlab programsko orodje. Dosegljivo: https://www.mathworks.com/?s_tid=gn_logo. [Dostopano: 15. 10. 2017].

[15] Signala padca. Dosegljivo: https://github.com/BharadwajS/Fall-detection-in-Android/blob/master/Fall%20detection.pdf. [Dostopano: 21. 09. 2017].

[16] Delovanje urnih merilcev srčnega utripa. Dosegljivo: https://exist.io/blog/fitness-trackers-heart-rate/. [Dostopano: 29. 09. 2017].

[17] Delovanje polar h7 ble senzorja. Dosegljivo: https://developer.polar.com/wiki/H6,_H7_and_H10_Heart_rate_sensors. [Dostopano: 28. 08. 2017].

[18] Delovanje pospeškometra. Dosegljivo: http://blog.hackerearth.com/2016/08/how-does-fitness-tracker-work.html. [Dostopano: 21. 09. 2017].

[19] Delovanje ppg. Dosegljivo: https://www.eeweb.com/electronics-quiz/heart-rate-sensor-technology. [Dostopano: 21. 09. 2017].

[20] Weihao Qu, Feng Lin, Aosen Wang, and Wenyao Xu. Evaluation of a low-complexity fall detection algorithm on wearable sensor towards falls and fall-alike activities. In *2015 IEEE Signal Processing in Medicine and Biology Symposium (SPMB)*, pages 1–6, Dec 2015.

[21] Statistika razširjenosti mobilnih naprav. Dosegljivo: https://www.statista.com/statistics/274774/forecast-of-mobile-phone-users-worldwide/. [Dostopano: 02. 09. 2017].

[22] Rr interval srčnega utripa. Dosegljivo: https://lifeinthefastlane.com/ecg-st-segment-evaluation/. [Dostopano: 16. 09. 2017].

[23] Delovanje senzorjev utripa pametnih ur. Dosegljivo: https://arstechnica.com/gadgets/2017/04/how-wearable-heart-rate-monitors-work-and-which-is-best-for-you/. [Dostopano: 29. 08. 2017].

[24] Benfano Soewito, Irwan, Anna Antonyová, and Fergyanto E. Gunawan. Fall detection algorithm to generate security alert. *Procedia Computer Science*, 59(Supplement C):350 – 356, 2015. International Conference on Computer Science and Computational Intelligence (ICCSCI 2015).