

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Grega Mežič

**Mobilna aplikacija za spremljanje
telesne aktivnosti**

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM
PRVE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: viš. pred. dr. Alenka Kavčič

Ljubljana, 2017

COPYRIGHT. Rezultati diplomske naloge so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavo in koriščenje rezultatov diplomske naloge je potrebno pisno privoljenje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

Besedilo je oblikovano z urejevalnikom besedil L^AT_EX.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Mobilna aplikacija za spremljanje telesne aktivnosti (angl. *Mobile application for controlling physical activity*)

Tematika naloge:

V okviru diplomske naloge izdelajte mobilno aplikacijo, ki bo pomagala pri spremljanju izvajanja redne telesne aktivnosti otrok s prekomerno težo. Aplikacija naj omogoča tako beleženje izvedenih športnih aktivnosti kot sistematično kontroliranje telesne teže, hkrati pa naj uporabnika spodbuja k večji aktivnosti preko izzivov prijateljev in zastavljenih nalog zdravnika. Aplikacijo naj sestavljata dva dela: aplikacija na pametni uri naj meri športno aktivnost in porabljeno energijo preko števila prehojenih korakov, aplikacija na tablici pa naj omogoča komunikacijo s spletnim strežnikom, prikaz statistike, izzive prijateljev ter naloge, ki jih postavi zdravnik. Pri zasnovi in realizaciji aplikacije uporabite sodobne tehnologije in orodja za izdelavo mobilnih aplikacij.

Zahvaljujem se mentorici viš. pred. dr. Alenki Kavčič za strokovno pomoč, motivacijo in usmerjanje skozi celotno diplomsko delo. Zahvaljujem se tudi mentorjema na Institutu „Jožef Stefan“ dr. Mitji Luštrek in Božidari Cvetković, univ. dipl. inž. rač. in inf. za pomoč in usmerjanje pri praktičnem delu diplomskega dela. Zahvala gre tudi zdravniku prim. asist. mag. Martinu Bigcu, dr. med. in celotni strokovni ekipi iz Univerzitetnega kliničnega centra Maribor, za vso pomoč pri izvedbi projekta. Rad bi se zahvalil tudi Klemnu Turšiču za delo na strežniku in Urošu Proseniku za delo na dizajnu aplikacije. Še posebej pa bi se rad zahvalil svoji družini in puncu za vso podporo, tako pri diplomskem delu, kot tudi pri celotnem študiju. Hvala, ker ste mi stali ob strani in mi dali še tisti zadnji delček motivacije, še posebej takrat, ko sem to najbolj potreboval.

„A goal without a plan is just a wish.“
– Antoine de Saint-Exupéry

Kazalo

Povzetek

Abstract

1	Uvod	1
2	Funkcionalne zahteve	5
2.1	Uporabniške zgodbe	6
2.2	Diagram primerov uporabe	8
3	Pregled področja	13
3.1	Opis pametne ure	14
3.2	Programska oprema na področju zaznavanja športnih aktivnosti in uravnavanja telesne teže	15
4	Aplikacija	23
4.1	Aplikacija na pametni uri	26
4.2	Aplikacija na tabličnem računalniku	40
5	Zaključek	53
	Literatura	55

Slike

2.1	Diagram primerov uporabe na pametni uri.	10
2.2	Diagram primerov uporabe na tabličnem računalniku.	11
3.1	Začetni zaslon aplikacije Google fit.	17
3.2	Zaslon aplikacije Samsung health.	18
3.3	Začetni zaslon aplikacije Lifelog.	20
4.1	Prikaz delovanja sistema.	23
4.2	Primer notacije JSON.	24
4.3	Primer datoteke „manifest“.	26
4.4	Prijavni zaslon na pametni uri.	27
4.5	Domači zaslon na pametni uri.	28
4.6	Sprejemnik, ki posluša spremembo internetne povezave na pametni uri, v datoteki „manifest“.	29
4.7	Registracija senzorja za štetje korakov na pametni uri.	30
4.8	Zaslon za pregled nalog.	35
4.9	Prikaz zaslona za podrobnejši pregled posamezne naloge.	36
4.10	Prikaz zaslona za obvestilo o opravljeni nalogi.	39
4.11	Prikaz potisnega obvestila.	41
4.12	Prikaz domačega zaslona na tabličnem računalniku.	42
4.13	Prikaz profilnega zaslona.	43
4.14	Prikaz zaslona za ročni vnos aktivnosti.	45
4.15	Prikaz zaslona za pregled zgodovine aktivnosti uporabnika.	46
4.16	Prikaz zaslona za pregled zgodovine telesne teže.	47

4.17 Prikaz zaslona za prijatelje.	48
4.18 Prikaz zaslona za pregled zgodovine telesne teže.	50
4.19 Prikaz zaslona za podrobnejši pregled posamezne naloge na tabličnem računalniku.	51

Seznam uporabljenih kratic

kratica	angleško	slovensko
IJS	Jožef Stefan institute	Institut „Jožef Stefan“
UKC Mari- bor	University Medical Centre Ma- ribor	Univerzitetni klinični center Maribor
UML	unified modeling language	poenoteni jezik modeliranja
GPS	global positioning system	globalni sistem pozicioniranja
API	application programming in- terface	aplikacijski programski vme- snik
JSON	JavaScript object notation	JavaScript objektna notacija
URL	uniform resource locator	enolični krajevnik vira
XML	extensible markup language	razširljiv označevalni jezik
MET	metabolic equivalent of task	metabolni ekvivalent

Povzetek

Naslov: Mobilna aplikacija za spremljanje telesne aktivnosti

Avtor: Grega Mežič

Kot posledica urbanizacije in razvoja sodobne informacijske tehnologije, se je v zadnjih dvajsetih letih, v Sloveniji, močno povečala debelost pri mladostnikih. Mladi se v prostem času gibljejo veliko manj, kot so se pred dvajsetimi leti, kar pa slabo vpliva na zdravstveno stanje mladostnikov. Cilj diplomskega dela je razvoj aplikacije za pomoč otrokom s prekomerno telesno težo. Glavni namen aplikacije je motivirati mladostnike, ki se zdravijo zaradi debelosti, in jim, s pomočjo zdravnika, nuditi vso potrebno pomoč ter nadzor nad opravljanjem športnih aktivnosti. V okviru diplomskega dela smo razvili dve aplikaciji, in sicer aplikacijo za pametno uro, ki na podlagi zajetih podatkov iz senzorjev naprave spremlja ter prepoznava otrokove športne aktivnosti, in aplikacijo za tablični računalnik, na kateri ima otrok pregled nad vsemi aktivnostmi, ki jih je zajela pametna ura. Pričakujemo, da bo aplikacija otrokom s prekomerno telesno težo pomagala, ne samo pri izgubi telesne teže, temveč tudi pri spremembi njihovega življenjskega stila.

Ključne besede: zaznavanje telesnih aktivnosti, štetje korakov, prekomerna telesna teža mladostnikov, debelost, pametna ura, tablični računalnik.

Abstract

Title: Mobile application for controlling physical activity

Author: Grega Mežič

As a result of urbanisation and the development of modern information technology, youngster obesity in Slovenia has severely increased in the last twenty years. Young people tend to exercise much less in their free time than twenty years ago, which results in their poor health. The aim of the diploma paper is to develop an application to help overweight children. The main purpose of the application is to motivate youngsters who are being treated for obesity and to provide them, with the help of a doctor, with all the necessary help and control over their sports activities. In the diploma paper we developed two applications, namely a smart watch application that monitors and recognises a child's sports activities based on the information gathered by the sensors of the device, and a tablet application that enables a child to have an overview of all the activities gathered by the smart watch. It is expected that the application will be of help to overweight children, not only to lose their weight but also to change their lifestyles.

Keywords: activity recognition, step counting, overweight youngsters, obesity, smart watch, tablet computer.

Poglavje 1

Uvod

Gibanje je osnova za zdravo življenje. Le kratek sprehod v naravi ali rekreativni tek lahko pozitivno vplivata na naše zdravje in počutje v današnjem hitrem tempu življenja. Nekdaj, ko ni bilo v navadi se voziti v šolo ali v službo s prevoznimi sredstvi, je bilo teh sprehodov bistveno več. Raziskave kažejo, da tako urbanizacija kot tudi način preživljanja prostega časa negativno vplivata na zdravstveno stanje v današnjem življenju otrok in mladostnikov.

Mladi veliko svojega prostega časa namenijo sodobnim medijem: gledanju televizije, igranju video ali računalniških igric, uporabi mobilnega telefona, gledanju video kaset ali DVD-jev, uporabi interneta ali drugim dejavnostim na računalniku [4]. Zaskrbljujoče je tudi dejstvo, da se povečuje delež mladostnikov, ki se v prostem času ne ukvarjajo s športom. Posledica tega pa je, da se je v Sloveniji, v zadnjih dvajsetih letih, močno povečala debelost pri mladostnikih [1]. Debelost močno poveča tveganje za razvoj sladkorne bolezni, povečuje pa tudi možnost za bolezni srca in ožilja, zato je potrebno dotično problematiko bolje nadzirati in jo pravočasno začeti zdraviti.

Na podlagi zgoraj omenjenih dejstev mnogi menijo, da vsa tehnologija, ki smo jo ustvarili, močno zavira motivacijo h gibanju. Nekateri pa smo mnenja, da je današnja napredno tehnologijo mogoče uporabiti ravno za spodbujanje in motiviranje, tako mlajše kot tudi starejše populacije h gibanju. Tako smo se pri uresničevanju našega cilja, tj. motivirati in spodbuditi mlade h

gibanju, prvič soočili z razvojem aplikacije e-Gibalec [5], ki smo jo razvili na Institutu „Jožef Stefan“ (v nadeljevanju IJS) v sodelovanju z Univerzo v Ljubljani, Fakulteto za šport.

Gre za aplikacijo, ki je namenjena osnovnošolcem, a jo lahko uporabljajo tudi ostali. Aplikacija s pomočjo pospeškometra prepozna telesno aktivnost in oceni njeno intenzivnost. Omogoča dodajanje prijateljev, ki se lahko med seboj izzivajo v različnih športnih izzivih, kar dodatno motivira uporabnike. Ravno zaradi aplikacije e-Gibalec pa se je začela razvijati zgodba o tem diplomskem delu. Zdravnik iz Univerzitetnega kliničnega centra Maribor (v nadaljevanju UKC Maribor) je na predstavitvi projekta e-Gibalec v aplikaciji videl rešitev za problem, s katerim se na pediatrični kliniki v UKC Maribor soočajo že vrsto let.

Vsako leto zdravijo otroke s prekomerno telesno težo in so, po besedah zdravnika, skupno v treh letih obravnavali že okoli štiristo otrok in mladostnikov. Pri zdravljenju debelosti je zdravnik s strokovno ekipo, v kateri so poleg zdravnika še fizioterapevti, kineziologi in psihologi, naletel na kar nekaj težav. Zdravljenja so se lotili tako, da so najprej skupino otrok (približno trideset naenkrat) peljali na Debeli rtič, kjer so otroke ozaveščali o zdravem prehranjevanju, debelosti in nasploh o zdravem načinu življenja. Otroci so se ob učenju tudi veliko gibali in zdravo prehranjevali. Po enotedenskem preživetem času na Debelem rtiču so otroci odšli domov, od zdravnika pa so, tako starši kot tudi otroci, dobili jasna navodila in naloge, ki jih morajo opravljati, vključno z napotki o zdravem prehranjevanju. Otroci naj bi nato preko spleta pošiljali podatke o svoji teži (uporabljali so elektronske tehtnice) in številu korakov (uporabljali so športno zapestnico). Nekaj časa je vse potekalo zadovoljivo. Težava se je pojavila po nekaj tednih, ko so otroci nehali pošiljati zahtevane podatke. Tudi naloge, ki naj bi jih otrok opravljal skupaj z družino (npr. pohod na bližnji hrib, plavanje v bazenu, kolesarjenje itn.), so otroci prenehali opravljati in ravno tukaj se je zgodilo nekaj, česar zdravnik ni pričakoval: starši so, namesto, da bi otroku pomagali in ga spodbujali, celo prikrivali recidiv teže, tako da rezultatev tehtanja

niso niti spreminjali niti pošiljali preko spletne pošte. Kljub temu, da se je zdravnik trudil in klical domov ter skušal spodbuditi otroka in starše, da sodelujejo pri zdravljenju debelosti, je čez čas otrok zdravljenje popolnoma opustil. Zdravnik je mnenja, da razlog za neuspeh tiči v kontroli. Pravi, da je brez neke avtomatizirane kontrole skoraj nemogoče uspeti. Rešitev pa je videl v aplikaciji e-Gibalec. Tako smo stopili v stik in kmalu ugotovili, da e-Gibalec pravzaprav ne bi dobro reševal omenjenega problema, zato smo se strinjali, da je bolje narediti povsem novo aplikacijo, ki bo namenjena izključno reševanju tega problema. Iz aplikacije e-Gibalec smo uporabili zgolj algoritem za prepoznavo športnih aktivnosti in oceno porabe energije, ki pa ni del diplomskega dela.

Ključne točke so bile jasne, zdravnik potrebuje nadzor nad otroki, aplikacija pa mora samodejno zaznavati športne aktivnosti, šteti korake in ugotovljati, ali je uporabnik uspešno opravil dodeljene naloge. Začeli smo oblikovati sistem za pomoč otrokom s prekomerno telesno težo, ki smo ga poimenovali Dr. Gibalko. Sistem je sestavljen iz spletne aplikacije, preko katere ima zdravnik nadzor nad otroki, in dveh aplikacij: aplikacije na pametni uri in aplikacije na tabličnem računalniku. Preko pametne ure aplikacija zaznava športno aktivnost, porabo energije in število prehojenih korakov ter ugotavlja, ali je uporabnik izpolnil naloge, ki mu jih je dodelil zdravnik. Namen aplikacije na tabličnem računalniku pa je predvsem prikaz vsega, kar zazna ura, omogoča pa tudi izzive s prijatelji. Ključ do uspeha, ki ga skušamo zagotavljati z uporabo aplikacije, je, po mnenju zdravnika, sestavljen iz naslednjih sklopov:

- ni dovolj, da uporabnik naredi eno fizično zahtevnejšo športno aktivnost na teden, potrebno je biti vsak dan vsaj zmerno aktiven. To želimo doseči tako, da uporabnika spodbudimo, da vsak dan prehodi določeno število korakov;
- kljub vsakodnevni zmerni fizični aktivnosti, pa mora vsaj enkrat na teden opraviti fizično zahtevnejšo aktivnost, ki jo v aplikaciji realiziramo

tako, da zdravnik pošlje uporabniku nalogo preko spletne aplikacije, ta pa jo mora v določenem roku opraviti;

- ključ do uspeha je tudi prehranjevanje, ki pa ga zaenkrat še ne prepoznavamo;
- pomembno dejstvo je tudi to, da ni dovolj, da način življenja spremeni samo otrok. Pomembno je, da se spremeni življenjski cilj v celi družini. Zato so naloge oblikovane tako, da jih navadno opravlja cela družina ali pa vsaj del družine, vsekakor pa ne otrok sam.

Cilj celotnega projekta je torej ta, da uporabnik aplikacije med zdravljenjem izgubi toliko telesne teže, kolikor mu je določil zdravnik. Najpomembnejše pri tem pa je, da uporabnik izgubljeno telesno težo tudi ohrani. In ravno to trajnost in trajni nadzor nam ponuja Dr. Gibalko.

Rezultat rešitve omenjenega problema je torej sistem za pomoč otrokom s prekomerno telesno težo, ki je sestavljen iz spletne aplikacije in aplikacije, rezdeležene na dva dela:

- aplikacija na pametni uri, ki zajema podatke o prehojenem številu korakov in zaznava telesne aktivnosti ter oceni porabo energije uporabnike;
- aplikacija na tabličnem računalniku, ki ponuja različne preglede za zgodovine poslanih podatkov, ki jih pošlje pametna ura, in nudi različne metode motiviranja uporabnika h gibanju.

V sklopu diplomeske naloge sta bili izdelani aplikaciji za pametno uro in tablični računalnik. Spletne aplikacije pa ni del diplomskega dela.

Poglavje 2

Funkcionalne zahteve

V tem poglavju si bomo pogledali, kako se je celoten sistem začel razvijati. Glede na to, da je projekt Dr. Gibalko precej obsežen in je na njem sodelovalo več ljudi, smo se odločili, da pred začetkom samega kodiranja najprej posvetimo nekaj časa načrtovanju. Na podlagi več sestankov, tako na IJS-ju kot tudi na UKC Maribor, smo od zdravnika dobili funkcionalne zahteve. Le-te je podal ustno, tako da je prosto govoril o svojih željah, mi pa smo ga s podvprašanji dodatno usmerjali, da smo zajeli celotno funkcionalnost sistema, in tako je nastal dokument v obliki uporabniških zgodb (angl. *user story*). Po sestankih smo ugotovili, da si zdravnik želi skupaj tri aplikacije: aplikacijo na tabličnem računalniku, aplikacijo na pametni uri in spletno aplikacijo. Aplikacija na pametni uri zajema podatke o gibanju, prepoznava telesne aktivnosti, računa porabo energije, šteje korake in vse zajete podatke, ob vzpostavitvi internetne povezave, pošlje na strežnik. Želja zdravnika je, da aplikacija na pametni uri, kar se da malo, interaktira z uporabnikom, zato lahko uporabnik na pametni uri vidi le število prehojenih korakov za trenutni dan in vidi ter opravlja naloge, ki mu jih dodeli zdravnik preko spletne aplikacije. Ideja aplikacije na tabličnem računalniku pa je ravno nasprotna ideji aplikacije na pametni uri. Ponuja namreč veliko interakcije z uporabnikom. Uporabnik lahko na tabličnem računalniku vidi zgodovino prehojenih korakov in porabo energije za različna časovna obdobja, prav tako vidi zgodovino

telesne teže, ki jo preko aplikacije tudi vnese. Pregled dodeljenih nalog je podoben, kot je pregled nalog na pametni uri, le da uporabnik naloge ne more niti začeti in posledično niti končati. Uporabnik lahko tudi doda svoje prijatelje znotraj aplikacije in jih izzove. Zmagovalec izziva je tisti, ki prehodi večje število korakov. Uporabnik lahko vidi svoj profil, na katerem so osnovni podatki o izzivih in o opravljenih nalogah. Sledi še spletna aplikacija, ki ni del diplomske naloge, vendar je vseeno vredna kratkega opisa. Uporabnik spletne aplikacije bo zdravnik sam (morda kasneje tudi nekdo od njegovega osebja). Preko spletne aplikacije ima pregled nad vsemi uporabniki. Za vsakega uporabnika lahko vidi zgodovino opravljenih in neopravljenih nalog ter zgodovino opravljenih in neopravljenih dnevnih ciljev (vsak uporabnik ima določeno normo v obliki števila korakov, ki jo mora izpolniti vsak dan). Zdravnik lahko za vsakega uporabnika določi in spremeni normo dnevnega cilja, dodeli pa mu lahko tudi naloge, katerim določi lokacijo, časovno enoto, in mu izbere enega izmed naslednjih možnih tipov naloge:

- plavanje,
- pohodništvo,
- kolesarjenje.

2.1 Uporabniške zgodbe

Kadar govorimo o razvoju programske opreme, je pojem uporabniške zgodbe predstavljen kot neformalni naravni opis ene ali več funkcij programske opreme. Uporabniške zgodbe so navadno napisane s strani končnega uporabnika, iz njih pa je dobro razvidno, katerih funkcij si uporabnik želi v nekem sistemu [8]. Za zapis uporabniških zgodb se navadno uporablja sledeč vzorec:

Kot *<tip uporabnika>*, si želim *<želja>*, zato da *<korist>*.

Tudi za naš projekt smo se odločili za uporabo uporabniških zgodb, saj smo si želeli agilnega in hitrega pristopa načrtovanja, zavedali pa smo se tudi, da se lahko katera funkcionalna zahteva tekom projekta spremeni.

V sklopu zajema funkcionalnih zahtev je zadnji sestanek potekal na UKC Maribor, kjer smo pridobili še zadnje informacije s strani zdravnika in njegovih sodelavcev na Pediatrični kliniki. Kot rezultat sestankov je nastal dokument s funkcionalnimi zahtevami, v obliki uporabniških zgodb:

- kot uporabnik se lahko prijavim v aplikacijo, zato da lahko dostopam do vseh funkcij aplikacije,
- kot uporabnik se lahko odjavim iz aplikacije, zato ker ne želim več uporabljati aplikacije,
- kot uporabnik lahko vidim dnevni cilj v obliki števila korakov zato, da vem, koliko korakov moram prehoditi,
- kot uporabnik lahko vidim trenutno število korakov v tekočem dnevu zato, da vem, koliko korakov moram še prehoditi, da dosežem dnevni cilj,
- kot uporabnik lahko vidim zgodovino števila korakov v obliki grafa za različno časovno obdobje, zato da lahko spremljam svoj napredek,
- kot uporabnik lahko vidim zgodovino porabe energije v obliki grafa za različno časovno obdobje, zato da lahko spremljam svoj napredek,
- kot uporabnik lahko vnesem svojo težo,
- kot uporabnik lahko vidim zgodovino svoje teže, ker želim videti svoj napredek,
- kot uporabnik lahko vidim naloge, ki mi jih je dodelil zdravnik,
- kot uporabnik lahko opravljam dane naloge, zato da dobim točke,
- kot uporabnik imam lahko pregled vseh prijateljev, da jih lahko izzovem,
- kot uporabnik lahko pošljem prošnjo za prijateljstvo drugemu uporabniku, ker želim imeti novega prijatelja,

- kot uporabnik lahko sprejem/zavrnem prejeto prošnjo za prijateljstvo,
- kot uporabnik lahko odstranim prijatelja s seznama prijateljev, ker ne želim več biti njegov prijatelj,
- kot uporabnik lahko pošljem izziv prijatelju, ker ga želim premagati,
- kot uporabnik lahko izberem različno trajanje posameznega izziva,
- kot uporabnik lahko vidim svoj profil, ker želim videti svoje podatke,
- kot uporabnik lahko ročno vnesem aktivnosti, ker želim, da zdravnik ve, kaj sem počel brez naprav.

Seveda smo dobili tudi ostale funkcionalne zahteve, predvsem za spletno aplikacijo, npr., kako lahko zdravnik pošlje nalogo uporabniku, ima pregled nad uporabniki itn., ki pa ni del diplomske naloge.

Zgoraj naštetе uporabniške zgodbe so bile izdelane na podlagi zdravnikovih želj, ki pa jih je bilo potrebno dopolniti, ko smo začeli s samim razvijanjem aplikacije. V tem trenutku smo dobili dovolj informacij za izdelavo temeljev aplikacije, več manjših funkcionalnosti pa smo dopolnili med samim razvijanjem. Vendar pa iz same strukture uporabniških zgodb ni dobro razvidno, kako se posamezne funkcionalnosti medsebojno povezujejo, prav tako nam ta struktura ne ponuja prikaza odvisnosti med funkcionalnostmi. Kadar želimo izvedeti takšne stvari, imamo na voljo kar nekaj orodij, ki so nam v pomoč pri modeliranju zahtev. Prav tako poznamo različne jezike, s katerimi je mogoče opisati funkcionalne zahteve, med njimi je zelo popularen jezik UML, s katerim smo tudi izdelali vizualizacijo zasnove sistema z *diagramom primerov uporabe*.

2.2 Diagram primerov uporabe

Gre za preprost diagram, ki prikazuje razmerje med uporabnikom in različnimi primeri uporabe. Diagram primerov uporabe lahko identificira različne tipe

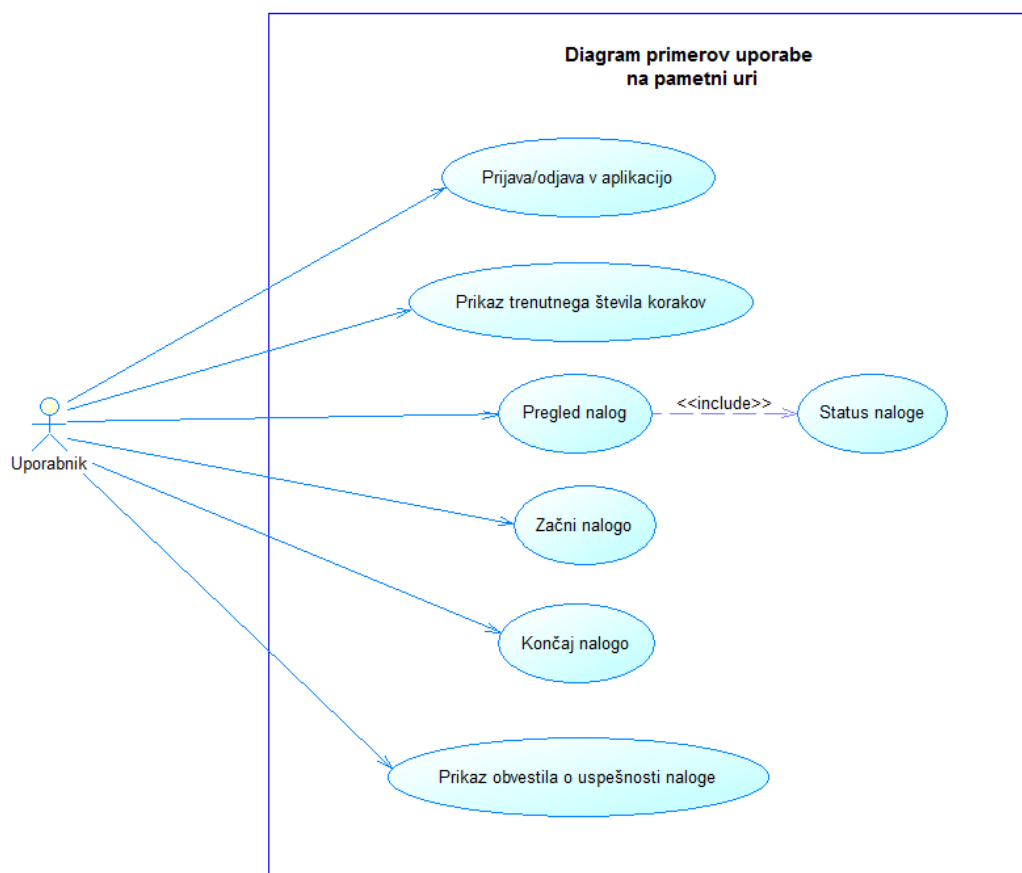
uporabnikov, sistemov in primerov uporabe [7]. Omenjeni diagram je navadno sestavljen iz naslednjih osnovnih gradnikov:

- akter,
- primer uporabe,
- različne oblike relacije:
 - relacija vsebuje (angl. *include*),
 - relacija razširja (angl. *extend*),
 - relacija posplošuje (angl. *generalization*).

Ko smo dobro razumeli uporabniške zgodbe, je sledila izdelava diagrama primerov uporabe. Za samo izdelavo smo uporabili orodje *PowerDesigner 12.5*, ki ponuja enostaven grafični vmesnik za gradnjo diagrama. Na tej točki smo se morali že odločiti, katere funkcionalnosti bo podpirala pametna ura in katere tablični računalnik. Določili smo, da mora ura vsebovati le najnujnejše zaslone, zajemati pa mora seveda vse potrebne podatke iz senzorjev, jih shranjevati in jih ob vzpostavljeni internetni povezavi poslati na strežnik, tablični računalnik pa mora nuditi vso potrebno vizualizacijo podatkov.

Kot prikazuje diagram na sliki 2.1, pametna ura vsebuje minimalno interakcijo med uporabnikom in aplikacijo. Na sliki vidimo, da ima akter (v našem primeru uporabnik) na voljo vse funkcionalnosti, s katerimi je povezan z relacijami. Primer uporabe „Pregled nalog“ še dodatno vsebuje (angl. *include*) „Status naloge“, kar pomeni, da uporabnik pri pregledu nalog vidi, kakšen status ima posamezna naloga (npr. naloga v teku). Seveda aplikacija na pametni uri vsebuje veliko servisov, ki delujejo v ozadju, vendar na diagramu primerov uporabe prikazujemo zgolj funkcionalne zahteve.

Nasprotno pa na sliki 2.2 vidimo veliko več interakcij med uporabnikom in aplikacijo na tabličnem računalniku, kar je tudi ideja in želja zdravnika. Na sliki vidimo, da je akter (uporabnik) povezan z velikim številom primerov uporabe, ki jih veliko razširja (angl. *extend*), vsebuje (angl. *include*) pa tudi



Slika 2.1: Diagram primerov uporabe na pametni uri.

druge primere uporabe. Iz tega diagrama lahko vidimo, da ima uporabnik na voljo vse funkcionalnosti na tabličnem računalniku, kot jih ima na uri, z izjemo dveh; ne more začeti ali končati naloge – nad njimi ima na tabličnem računalniku le pregled in natančen ogled naloge.

Na podlagi diagramov s slik 2.1 in 2.2 smo imeli dovolj podatkov za začetek razvoja aplikacije. V tem trenutku ni bilo zelo pomembno, da smo zajeli vse manjše podrobnosti in manjše funkcionalnosti aplikacije, pomembno je bilo, da smo zajeli glavne temelje, iz katerih je možno razvijati aplikacijo naprej, in ti so bili dobro zasnovani.



Slika 2.2: Diagram primerov uporabe na tabličnem računalniku.

Poglavje 3

Pregled področja

Glede na to, da je danes na področju zaznavanja gibanja in spremljanja športnih aktivnosti s pametnimi napravami razvito precej programske opreme, je smiselno, pred začetkom razvijanja nove aplikacije, narediti natančno analizo tako programske kot tudi strojne opreme na izbranem področju. Današnje pametne naprave so opremljene z velikim številom senzorjev, ki omogočajo zajem podatkov, tako o uporabnikovih vitalnih znakih kot tudi razne pospeške, orientacijo naprave, temperaturo okolja itn. Če naštejemo nekaj najuporabnejših senzorjev, ki jih današnje naprave ponujajo:

- senzor pospeškov,
- senzor za orientacijo,
- senzor bližine,
- senzor za merjenje zračnega pritiska,
- senzor za merjenje srčnega utripa,
- senzor GPS,

vidimo, da lahko aplikacije z inteligentno programsko opremo in z uporabo vseh kombinacij senzorjev, precej dobro zaznajo različne športne aktivnosti,

iz zajetih podatkov pa so zmožne podati tudi dovolj natančno oceno porabe energije uporabnika.

V nadaljevanju tega poglavja se bomo osredotočili predvsem na pregled že obstoječih aplikacij na področju zaznavanja športnih aktivnostih in aplikacij, ki nudijo uporabnikom pregled in uravnavanje uporabnikove teže.

3.1 Opis pametne ure

Pametne ure so na področju računalništva dokaj nova tema, ki pa se precej hitro razvija. Za razvoj naše aplikacije smo uporabljali pametno uro *ZGPAX SPP*. Ura ima nameščen operacijski sistem Android 5.1 in ima na voljo naslednje senzorje:

- senzorji, ki temeljijo na strojni opremi:
 - pospeškomer,
 - senzor bližine,
 - senzor svetlobe,

- senzorji, ki temeljijo na programski opremi:
 - senzor za zaznavanje pomembnih premikov,
 - senzor za detekcijo koraka,
 - senzor za štetje korakov,
 - senzor za detekcijo nagiba,
 - senzor za zaznavanje specifičnega giba, ki prižge zaslon,
 - senzor, ki zaznava, kadar je naprava dvignjena s poljubne površine.

Iz zgornjega seznama lahko vidimo, da se senzorji, na nivoju operacijskega sistema Android, delijo na strojne in programske senzorje. Strojni senzorji so fizični senzorji, ki so vgrajeni v naprave. Ti senzorji pridobivajo podatke direktno z merjenjem specifičnih okoljskih lasnosti. Senzorji, ki temeljijo na

programski opremi, pa pridobivajo podatke iz enega ali več strojnih senzorjev.

Za delovanje aplikacije Dr. Gibalko potrebujemo zgolj strojni senzor pospeškomer in senzor za detekcijo ali štetje korakov.

3.2 Programska oprema na področju zaznavanja športnih aktivnosti in uravnavanja telesne teže

V sklopu diplomske naloge smo dlje časa testirali kar nekaj različnih aplikacij na omenjenem področju. Zanimivo je, da kljub temu, da si aplikacije delijo skupno precej ozko tematiko, se aplikacije dokaj razlikujejo med seboj. Vsaka ima neko dodatno funkcionalnost, inovativen grafični vmesnik in veliko različnih pristopov za zaznavanje športnih aktivnosti, kot tudi različne pristope pomoči uporabniku za uravnavanje uporabnikove telesne teže.

V naslednjih podsklopih se bomo dotaknili treh aplikacij, ki so na izbranem področju zelo priljubljene. Te aplikacije so zanimive predvsem zato, ker imajo določene dele funkcionalnosti, na pogled, zelo podobne naši aplikaciji, vendar pa zaradi specifičnih razlogov ne ustrezajo rešitvi danega problema. Aplikacija Dr. Gibalko je namenjena relativno ozki ciljni skupini uporabnikov, ki imajo skupen problem – debelost. Na trgu je sicer precej aplikacij, ki na takšen ali drugačen način pomagajo uporabniku izgubiti telesno težo, nudijo motivacijo h gibanju, vendar si delijo skupno lastnost: ne ponujajo nadzora strokovne osebe nad opravljanjem danih nalog in izpolnjevanjem dnevnih ciljev, kar je zelo pomembno za reševanje danega problema. Vseeno pa si pogledjmo posebnosti naslednjih treh aplikacij.

3.2.1 Google fit

Google fit je aplikacija za nadzor uporabnikovega zdravja, ki jo je razvil računalniški gigant Google, za operacijski sistem Android (slika 3.1). Apli-

kacija, s pomočjo omenjenih senzorjev, samodejno zazna naslednje aktivnosti:

- vožnjo z vozilom,
- kolesarjenje,
- hojo,
- tek,
- mirovanje,
- nagibanje (kadar se kot naprave močno spreminja glede na gravitacijo).

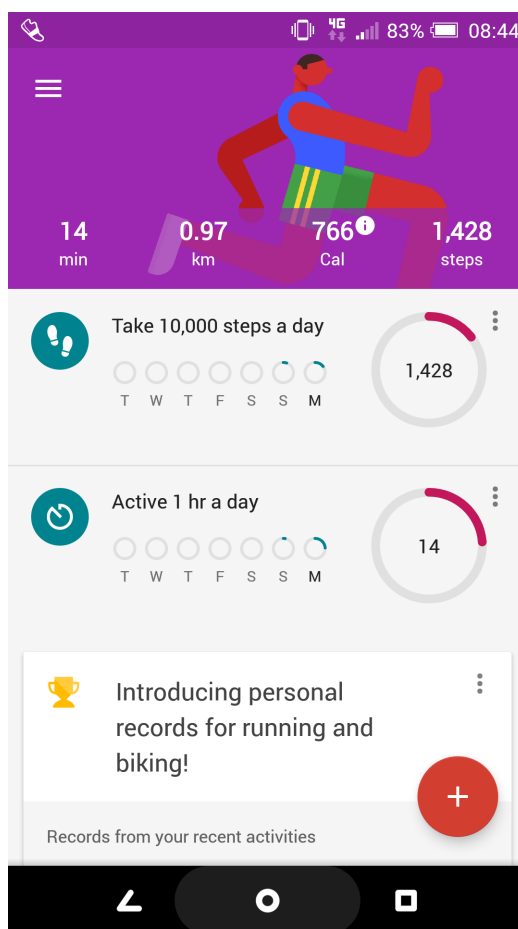
Posebna lastnost te aplikacije je ta, da imajo lahko ostale aplikacije, preko Google fit API-jev, dostop do podatkov o aktivnostih, ki jih zazna aplikacija Google fit. Tako lahko vsak programer razvija svojo programsko opremo na tem področju brez znanja o samem zaznavanju aktivnosti. Aplikacija ponuja preprost pregleden grafični vmesnik, ki vsebuje vse potrebne informacije za uporabnika. Omogoča tudi ročni vnos telesne teže, uporabnik pa ima na voljo tudi pregled zgodovine vnešenih tež in pregled nad zgodovino vseh športnih aktivnosti.

Prednosti:

- na razpolago ponuja API-je drugim aplikacijam,
- v primerjavi z ostalimi, precej natančno zaznava aktivnosti,
- v primerjavi z ostalimi, za prepoznavanje aktivnosti porabi zelo malo električne energije.

Slabosti:

- algoritem za prepoznavo aktivnosti ne zna prepoznati spanja,
- ni mogoče tekmovati ali se primerjati z ostalimi uporabniki,
- v aplikaciji je zelo malo motivacijskih metod.

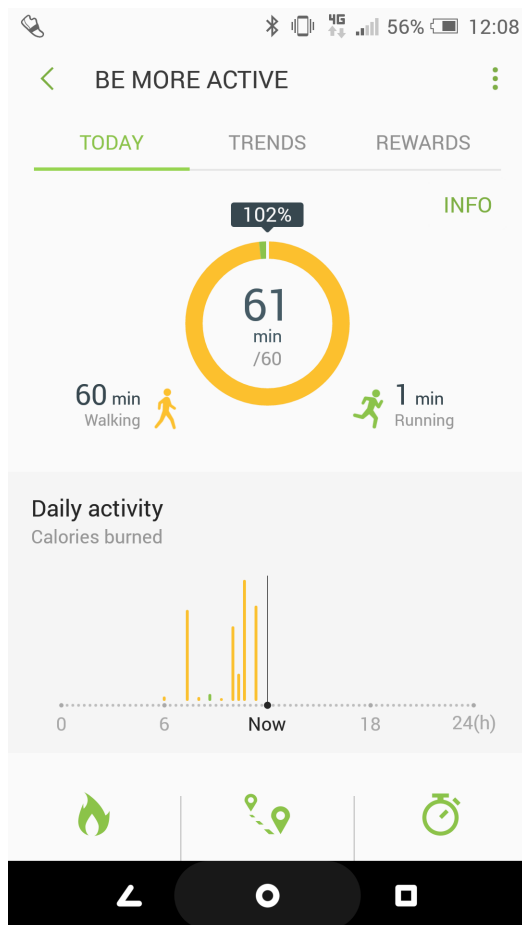


Slika 3.1: Začetni zaslon aplikacije Google fit.

Mnogi bi se lahko vprašali, zakaj za projekt Dr. Gibalko nismo uporabili nabor API-jev, ki jih ponuja aplikacija Google fit. Razlog je predvsem to, da ne želimo biti omejeni na prepoznavanje zgolj tistih aktivnosti, ki jih ponuja API Google fit. V prihodnosti želimo razpoznavati tudi bolj specifične aktivnosti. Najpomembnejša izmed teh je zagotovo prehranjevanje. Na IJS-ju smo že začeli z izvajanjem meritev za različne aktivnosti, ki jih bomo v prihodnosti poizkušali prepoznati, in to je glavni razlog, zakaj uporabljamo lasten algoritem za prepoznavo aktivnosti in porabo energije.

3.2.2 Samsung health

Aplikacija Samsung health (slika 3.2) je sicer namenejena napravam različnih znamk, vendar pa so nekatere funkcije omogočene samo za naprave znamke Samsung. Kljub temu pa ponuja za ostale naprave veliko uporabnih funkcij.



Slika 3.2: Zaslona aplikacije Samsung health.

Aplikacija na domačem zaslonu prikazuje število prehojenih korakov, ponuja pa tudi lep pregled zgodovine števila korakov in porabo energije. Prav tako je možen vnos telesne teže in ostalih osebnih podatkov, kot so: ime in priimek, datum rojstva, telesna teža in osebna subjektivna ocena telesne aktivnosti (od ena do pet, kjer ena pomeni, da uporabnik ni športno aktiven,

in pet pomeni, da je uporabnik zelo športno aktiven).

Aplikacija ponuja tudi možnost povezave z nekaterimi pripomočki znamke Samsung:

- sledilniki aktivnosti:
 - Samsung Gear Fit2,
 - Samsung Gear IconX,
 - Samsung Charm,
- merilnik srčnega utripa:
 - Samsung Gear IconX,
- pametne ure:
 - Samsung Gear S3,
 - Samsung Gear S2,

ki se s povezavo *bluetooth* povežejo z aplikacijo in tako lahko uporabnik preko aplikacije prenese vse podatke, ki jih je posnela naprava, v aplikacijo. Možnost te sinhronizacije z različnimi manjšimi napravami je v praksi zelo dobrodošla, saj je določene športne aktivnosti precej nerodno opravljati s pametnim telefonom.

Prednosti:

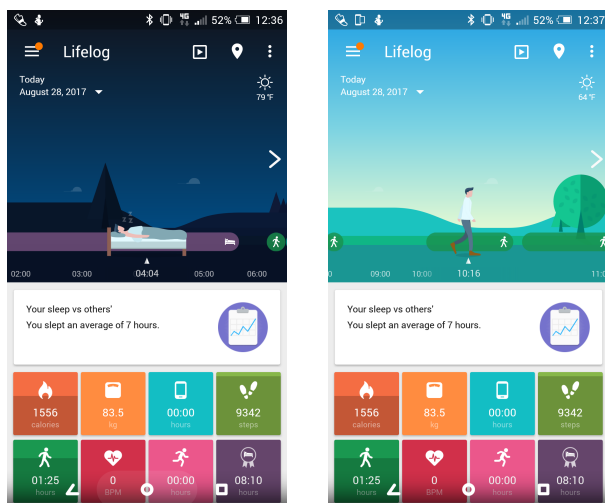
- možnost povezave z ostalimi napravami preko povezave bluetooth,
- motiviranje uporabnika h gibanju v obliki nagrajevanja z značkami, ob zadostnem gibanju in izpolnjevanju ciljev,
- možnost tekmovanja z ostalimi uporabniki.

Slabosti:

- določene funkcionalnosti aplikacije so podprte samo z napravami Samsung,
- določene funkcionalnosti delujejo le ob povezavi z ostalimi napravami.

3.2.3 Lifelog

Lifelog, v primerjavi z ostalimi, ponuja zelo inovativen in uporabniku prijazen grafični vmesnik, ki zelo jasno in lepo prikaže uporabnikove aktivnosti v izbranem dnevu, kot to prikazuje slika 3.3.



(a) Prikazana aktivnost: spanje. (b) Prikazana aktivnost: hoja.

Slika 3.3: Začetni zaslon aplikacije Lifelog.

Funkcionalnosti te aplikacije so v veliki meri podobne kot pri aplikacijah Google fit in Samsung health. Kot dodatek pri zaznavi telesnih aktivnostih pa aplikacija Lifelog zaznava tudi spanje in takoj, ko naprava zazna premake, ki naj bi pomenili konec spanja, aplikacija preko obvestila v orodni vrstici naprave vpraša uporabnika, če je zaznavanje spanja pravilno. Uporabnik lahko nato ročno popravi čas spanja ali pa zaznan spanec potrди samodejno..

Prednosti:

- samodejno zaznavanje spanca,
- primerjava vseh aktivnosti (tudi spanja) z ostalimi uporabniki,
- možnost tekmovanja z ostalimi uporabniki.

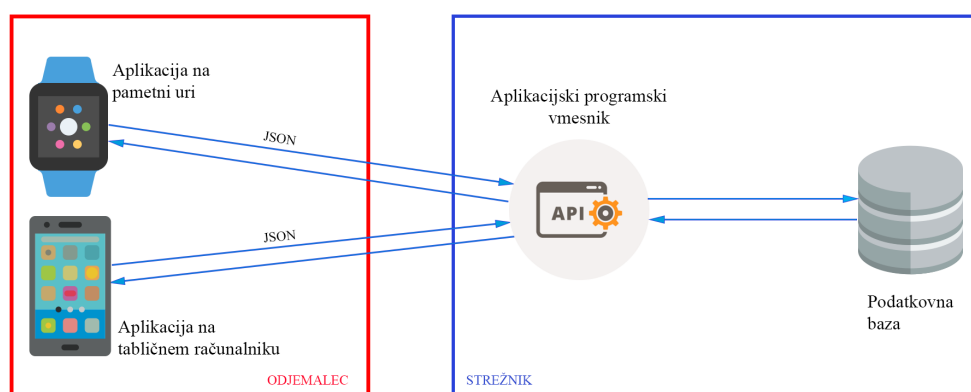
Slabosti:

- v aplikaciji je zelo malo motivacijskih metod,
- med testiranjem smo opazili, da števec korakov, v primerjavi z ostalimi aplikacijami, nekoliko odstopa, in sicer zazna več korakov od ostalih.

Poglavje 4

Aplikacija

Sistem pri projektu Dr. Gibalko je sestavljen iz štirih glavnih komponent, te so: aplikacija na pametni uri, aplikacija na tabličnem računalniku, podatkovna baza na strežniku in aplikacijski programski vmesnik na strežniku, ki povezuje podatkovno bazo ter aplikaciji na pametni uri in tabličnem računalniku. Obe aplikaciji se povezujeta na podatkovno bazo preko API-ja, kot to prikazuje slika 4.1. Za prenos podatkov po povezavi med aplikacijo in API-jem smo izbrali format JSON.



Slika 4.1: Prikaz delovanja sistema.

JSON je standardna človeško berljiva notacija, sestavljena iz parov: ključ,

vrednost, ki se v praksi veliko uporablja. Omogoča poljubno gnezdenje objektov JSON, kot vrednost objekta pa lahko zapišemo tudi seznam (slika 4.2). Takšna notacija nam omogoča enostaven prenos podatkov, tako primitivnih kot tudi kompleksnih (seznami, gnezdeni podatki itn.) [2].

```
{
  "success": 1,
  "rest_data": [
    {
      "time": "1491780504",
      "met": "1.3445301055908",
      "duration": "1"
    },
    {
      "time": "1491776904",
      "met": "1.3426605463028",
      "duration": "54.75"
    }
  ],
  "walk_data": [
    {
      "time": "1491780504",
      "met": "0",
      "duration": "0"
    },
    {
      "time": "1491776904",
      "met": "1.7874666452408",
      "duration": "1.75"
    }
  ],
  "run_data": [
    {
      "time": "1491780504",
      "met": "0",
      "duration": "0"
    },
    {
      "time": "1491776904",
      "met": "0",
      "duration": "0"
    }
  ]
}
```

Slika 4.2: Primer notacije JSON.

Vedno kadar aplikacija, bodisi na tabličnem računalniku bodisi na pametni uri, želi nekaj poslati na strežnik ali pa prebrati nekaj s strežnika, to stori na isti način. Aplikacija (v tem primeru odjemalec) pošlje v formatu JSON na določen naslov URL potrebne parametre (npr. identifikacijsko številko uporabnika) in dobi odgovor, prav tako v formatu JSON, ki najprej vsebuje podatek o tem, ali je bila zahteva pravilno poslana, in v tem primeru dobljen

JSON vsebuje tudi potrebne podatke; v nasprotnem primeru pa vsebuje besedilo o napaki.

Za lažje razumevanje vsebine, ki sledi, si pogledjmo tri glavne komponente na nivoju operacijskega sistema Android, ki jih uporablja naša aplikacija:

- aktivnost (angl. *Activity*) je najbolj uporabljena komponenta. Uporabimo jo vedno, kadar želimo imeti interakcijo med aplikacijo in uporabnikom. Zaslone, kot so npr. prijavnih zaslon, domači zaslon itn., pomenijo posamezno aktivnost v aplikaciji. Sicer ni nujno, da prav vsak zaslon pomeni svojo aktivnost, saj ima lahko ena aktivnost več podzaslonov;
- servis (angl. *Service*) je komponenta, ki jo uporabimo predvsem takrat, ko želimo, da aplikacija opravi neko delo, ki traja dlje časa, in pri tem nima interakcije z uporabnikom;
- sprejemnik (angl. *Receiver*) je komponenta, ki neprestano „poslušá“ določene akcije na nivoju operacijskega sistema, na katere je sprejemnik naročen. V aplikaciji Dr. Gibalko imamo npr. sprejemnik, ki poslušá spremembo vrednosti minute na sistemski uri. Torej se sprejemnik sproži vsako minuto.

Vse aktivnosti, servisi, pa tudi sprejemniki morajo biti definirani v XML¹ datoteki „manifest“². V tej datoteki se nahajajo vse informacije o aplikaciji. Naštete so vse komponente, ki jih vsebuje aplikacija, deklarirati pa je potrebno tudi vsa dovoljenja, ki jih potrebuje aplikacija za delovanje (npr. dovoljenje za uporabo interneta). Definiranim komponentam se lahko določi tudi izrečna pravila, kot je npr. zagonna aktivnost (angl. *launcher activity*) – ta aktivnost se pokliče ob zagonu aplikacije (slika 4.3).

V nadaljevanju si bomo natančneje pogledali oba dela aplikacije Dr. Gibalko, tako na pametni uri, kot tudi na tabličnem računalniku, saj API in podatkovna baza nista del diplomskega dela.

¹<https://www.w3.org/XML/>

²<https://developer.android.com/guide/topics/manifest/manifest-intro.html>

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="si.ijs.drgibalko_wear">

    <uses-feature
        android:name="android.hardware.typeId.watch"
        android:required="false" />

    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />
    <uses-permission android:name="android.permission.WAKE_LOCK" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.VIBRATE" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="Dr. Gibalko"
        android:supportsRtl="true"
        android:theme="@android:style/Theme.DeviceDefault">
        <activity
            android:name=".ui.main.MainActivity"
            android:label="MainActivity"
            android:launchMode="singleTask"
            android:theme="@android:style/Theme.Material.NoActionBar.Fullscreen" />
        <activity
            android:name=".ui.login.LoginActivity"
            android:label="Dr. Gibalko"
            android:theme="@android:style/Theme.Material.NoActionBar.Fullscreen">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

Slika 4.3: Primer datoteke „manifest“.

4.1 Aplikacija na pametni uri

Ideja aplikacije na pametni uri je, da ima uporabnik minimalno interakcijo z aplikacijo. Aplikacija je zasnovana tako, da uporabniku za delovanje aplikacije na pametni uri ni potrebno konstantno klikati ali nastavljati določenih vrednosti. Prav tako smo želeli ustvariti minimalno število zaslonkih mask, kar nam je tudi uspelo. Naredili smo namreč popolnoma funkcionalno aplikacijo z le petimi zaslonskimi maskami:

- prijavni zaslon,
- domači zaslon,

- zaslon za pregled nalog,
- zaslon za podrobnejši pregled posamezne naloge,
- zaslon za obvestilo o opravljeni nalogi.

V nadaljevanju si bomo natančneje pogledali vse funkcionalnosti, ki jih ponuja aplikacija na pametni uri, razdeljene po posameznih, zgoraj naštetih, zaslonih.

4.1.1 Prijavni zaslon na pametni uri

Prijavni zaslon (slika 4.4) ima pravzaprav samo eno funkcionalnost: omogočiti uporabniku, da se prijavi v aplikacijo. Aplikacija preveri, če je uporabnik vpisal tako uporabniško ime kot tudi geslo, in v primeru uspeha pošlje podatke na strežnik, ta pa preko API-ja preveri, če so vnešeni podatki pravilni, in v formatu JSON vrne odgovor. Aplikacija ob uspešni prijavi shrani uporabniško ime in identifikacijsko številko uporabnika,

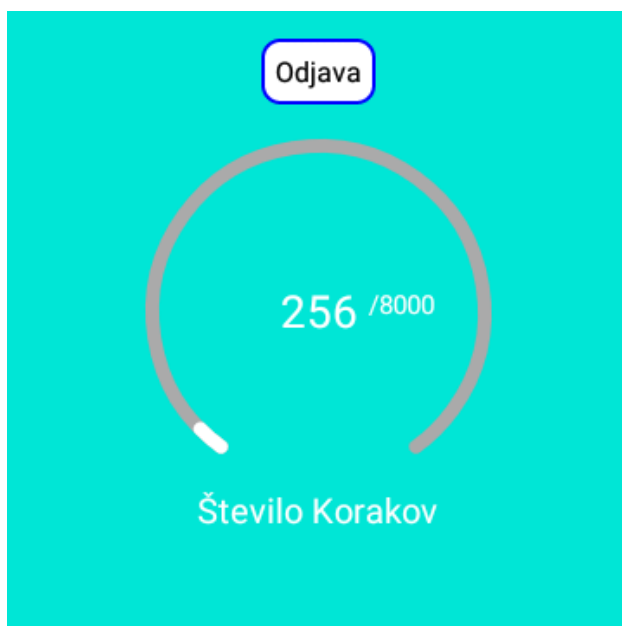


Slika 4.4: Prijavni zaslon na pametni uri.

shrani pa si tudi prijavo, da se uporabniku ni potrebno prijaviti naslednjič, ko želi uporabljati aplikacijo. Če pa se uporabnik odjavi, nas aplikacija preusmeri na domači zaslon.

4.1.2 Domači zaslon na pametni uri

Domači zaslon na prvi pogled deluje, kot da nima nikakršnih posebnih funkcionalnosti, razen prikazovalnika števca korakov in možnosti odjave iz aplikacije, kot to prikazuje slika 4.5. V resnici pa se v ozadju dogaja marsikaj, kar poskrbi za delovanje aplikacije.



Slika 4.5: Domači zaslon na pametni uri.

Ob inicializaciji te aktivnosti aplikacija najprej prebere vrednost trenutno prehojenih korakov, ki je zapisana v temu namenjeni datoteki XML „shared-preferences“. Gre za datoteko, kamor lahko trajno shranimo željene vrednosti primitivnih tipov podatkov v obliki: ključ, vrednost [3]. V tej datoteki so shranjeni tudi ostali podatki, kot so: identifikacijska številka uporabnika, vrednost, ki pove, da je uporabnik prijavljen itn. Po prebrani vrednosti o

prehojenem številu korakov aplikacija prikaže to informacijo na krožnici, kot je prikazano na sliki 4.5. Naslednja, ki je tudi zadnja uporabniku vidna funkcionalnost na tem zaslonu, je možnost odjave. Ob pritisku na gumb „Odjava“ aplikacija preusmeri uporabnika na prijavní zaslon, ob tem pa izbriše vse vrednosti iz prej omenjene datoteke XML.

Hkrati pa se ob inicializaciji tega zaslona zaženejo tudi servisi (angl. *services*) in sprejemniki (angl. *receivers*), ki delujejo v ozadju. Aplikacija Dr. Gibalko ima na pametni uri tri servise:

- servis za štetje korakov,
- servise za prepoznavo aktivnosti in oceno porabe energije,
- servis za opravljanje nalog (ki bo opisan kasneje).

Ima pa tudi tri sprejemnike:

- sprejemnik, ki posluša spremembo internetne povezave (angl. *network state changed*) (slika 4.6),
- sprejemnik, ki v sistemski uri posluša spremembo minutne vrednosti,
- sprejemnik, ki posluša, če se je operacijski sistem zagnal (aktivira se vedno ob ponovnem zagonu sistema).

```
<!--Receiver for network change -->
<receiver android:name=".receivers.NetworkChangeReceiver">
  <intent-filter>
    <action android:name="android.net.conn.CONNECTIVITY_CHANGE" />
  </intent-filter>
</receiver>
```

Slika 4.6: Sprejemnik, ki posluša spremembo internetne povezave na pametni uri, v datoteki „manifest“.

Ob zagonu te aktivnosti se najprej zažene servis za štetje korakov in zaznavo športne aktivnosti, preko tako imenovane servisne povezave (angl.

service connection), ki omogoča, da aktivnost lahko interaktira s samim servisom. To povezavo potrebujemo predvsem zato, ker mora npr. senzor za štetje korakov, ob detekciji koraka, sporočiti to aktivnosti, da se lahko nova vrednost nato prikaže v krožnem prikazovalniku števila korakov na domačem zaslonu. Povezava nam torej omogoča, da lahko servis za štetje korakov preko vmesnika (angl. *interface*) sporoči aktivnosti, da je servis zaznal korak, sporoči pa tudi novo vrednost.

Servis za štetje korakov: glavna naloga tega servisa je štetje korakov in pošiljanje števila korakov na strežnik, za vsako uro. Servis za štetje korakov uporablja temu namenjen senzor. Zanimiva ugotovitev je, da omenjeni senzor uporablja za zaznavo korakov zgolj pospeškomer, ki ga imajo skoraj vse pametne ure, medtem ko imajo senzor za štetje korakov le redke pametne ure. Torej je za delovanje aplikacije potrebno imeti tako pametno uro, ki ima senzor za štetje korakov, kar pa v našem primeru ne povzroča težav, saj bodo, po naših navodilih, za nakup ur poskrbeli na pediatrični kliniki na UKC Maribor. Ob zagonu servisa za štetje korakov, se najprej naredi registracija senzorja za štetje korakov (slika 4.7),

```
//register step counter sensor
private void registerSensors() {
    sensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);
    sensorStepCount = sensorManager.getDefaultSensor(Sensor.TYPE_STEP_COUNTER);

    sensorManager.registerListener(this, sensorStepCount, SensorManager.SENSOR_DELAY_GAME);
}
```

Slika 4.7: Registracija senzorja za štetje korakov na pametni uri.

ta servis pa inicializira tudi sprejemnik, ki v sistemski uri posluša spremembo minutne vrednosti. Ko je senzor za štetje korakov registriran in zazna spremembo – korak, le-ta pokliče metodo *onSensorChanged* s parametrom tipa *SensorEvent*, ki vsebuje informacijo o:

- senzorju, ki je zaznal spremembo,

- časovni oznaki, kdaj je bila sprememba na senzorju zaznana (angl. *timestamp*),
- natančnosti senzorja,
- vsebuje pa tudi tabelo z vrednostmi tipa *float*.

Ko senzor za štetje korakov zazna korak, lahko preko omenjene metode *onSensorChanged* dostopamo do podatkov tabele tipa *float*, ki v tem primeru (kadar gre za senzor štetja korakov) vsebuje le eno vrednost – število korakov, torej je dolžina tabele v tem primeru enaka ena. Vrednost, ki jo vsebuje tabela, nam pove, koliko korakov je senzor zaznal od takrat, ko je bil nazadnje registriran, do tega trenutka. Ker želimo vedeti, koliko korakov je uporabnik prehodil, za vsako uro, hkrati pa želimo vedeti, koliko korakov je prehodil na dan, nam ta vrednost ne ponuja željene informacije. Zato uporabljamo to metodo zgolj za detekcijo koraka (metoda se pokliče vedno, kadar senzor zazna korak). Tako imamo znotraj omenjene metode dva števec korakov: števec, ki šteje korake za posamezno uro in, števec, ki šteje korake za trenutni dan. Vedno, kadar zaznamo korak, najprej preberemo vrednost o prehojenih korakih, tako za trenutno uro kot tudi za trenutni dan, iz datoteke *shared-preferences*, nato povečamo vrednost za ena in nazaj zapišemo obe vrednosti. Tako lahko v vsakem trenutku vemo, koliko korakov je uporabnik prehodil za posamezno uro, in koliko korakov je prehodil na dan. Seveda pa je treba vrednost o številu korakov ob polni uri postaviti na nič. Za to poskrbi že prej omenjeni sprejemnik, ki posluša na sistemsko akcijo o spremembi minutne vrednosti na sistemski uri.

Servis za prepoznavo aktivnosti in oceno porabe energije: struktura tega servisa je zelo podobna tisti, ki jo ima servis za štetje korakov. Pomembno je omeniti, da algoritem, ki prepoznavno telesno aktivnost in ocenjuje porabo energije, ni del diplomskega dela, saj so ga razvili drugi razvijalci na IJS-ju. Pod diplomsko delo pa spada implementacija tega algoritma in pa interakcija z aktivnostjo preko servisne povezave, na enak način kot pri servisu za štetje korakov. Za potrebe diplomske naloge pa je bilo potrebno narediti

tudi algoritem za pošiljanje vseh aktivnosti in za izračunano porabo energije, za vsako uro posebej. Algoritem za pošiljanje podatkov na strežnik najprej naredi poizvedbo v podatkovno bazo, kjer prebere vse zaznane aktivnosti, ki še niso poslani na strežnik. Za vsako zaznano aktivnost v podatkovni bazi hranimo naslednje vrednosti:

- identifikacijsko številko aktivnosti,
- identifikacijsko številko uporabnika,
- časovno oznako (angl. *timestamp*),
- ime aktivnosti,
- porabo energije v MET-ih.

MET je matematično predstavljen kot:

$$1MET = 1 \frac{kcal}{kg * h} \quad (4.1)$$

Enačba pravi, da je en MET enak eni kilokaloriji na kilogram telesne teže na uro. Enačbo lahko torej interpretiramo tudi tako: človek v popolnem mirovanju (1 MET pomeni popolno mirovanje, npr. gledanje televizije) porabi v eni uri toliko kilokalorij, kolikor ima kilogramov. Iz te enačbe lahko torej izračunamo porabo kalorij, če enačbo nekoliko obrnemo:

$$kcal = MET * kg * h \quad (4.2)$$

Aplikacija na pametni uri torej pošlje na strežnik podatke o: mirovanju, hoji in teku. Za vsako aktivnost izračunamo, koliko časa je bila aktivnost zaznana in povprečno vrednost MET-a. Strežnik lahko nato izračuna, koliko kalorij je uporabnik porabil ob izvajanju posamezne aktivnosti, za vsako uro posebej.

Sprejemnik za zaznavo spremembe minutne vrednosti na sistemski uri: je „naročen“ na akcijo t. i.

TIME_TICK.

To je sistemska akcija, ki se sproži na nivoju operacijskega sistema Android in obvesti vse „naročene“ sprejemnike, da se je sistemski uri spremenila minutna vrednost. Naš sprejemnik najprej preveri, če je vrednost minute enaka nič (če je polna ura), in v tem primeru pošlje podatke za trenutno uro na strežnik. Ker pa vemo, da pametna ura vseskozi ne bo imela vzpostavljene povezave, je potrebno vrednost o številu korakov za posamezno uro najprej zapisati v podatkovno bazo. Po uspešnem vnosu vrednosti za posamezno uro sprejemnik preveri, če je vzpostavljena internetna povezava, in nato naredi poizvedbo v podatkovno bazo za vse še neposlane vrednosti o številu prehojenih korakov za posamezno uro. Ob uspešno poslani zahtevi si aplikacija v datoteko *sharedpreferences* shrani podatek, kdaj je bilo število korakov nazadnje poslano na strežnik, in postavi število prehojenih korakov za tekočo uro na nič. Podobna funkcionalnost se zgodi ob polnoči, ko poklican sprejemnik ugotovi, da je tako vrednost ure kot tudi minute enaka nič. Za dodatek pa takrat v datoteki *sharedpreferences* ponastavi še vrednost o številu prehojenih korakov za trenutni dan na nič.

Sprejemnik za zaznavo spremembe internetne povezave: je prav tako „naročen“ na eno izmed sistemskih akcij operacijskega sistema Android. Akcija, na katero je „naročen“ ta sprejemnik, se imenuje:

`android.net.conn.CONNECTIVITY_CHANGE,`

pokliče pa se vedno, kadar pride do spremembe internetne povezave na napravi. Ta sprejemnik mora najprej preveriti, do kakšne spremembe je prišlo; če je prišlo do prekinitve internetne povezave, sprejemnik samo zaključi izvajanje, če pa je prišlo do vzpostavitve internetne povezave, sprejemnik opravi neko delo. Želja je, da uporabnik, ko po neki športni aktivnosti pride domov, v trenutku vidi svoj napredek. Zato ob vzpostavitvi internetne povezave sprejemnik prebere časovno oznako (angl. *timestamp*), od nazadnje poslane zahtevke o prehojenem številu korakov, in v kolikor je razlika do trenutne časovne oznake večja kot ena ura, tudi ta sprejemnik naredi poizvedbo v podatkovno bazo prehojenih korakov in le-te pošlje na strežnik, na enak

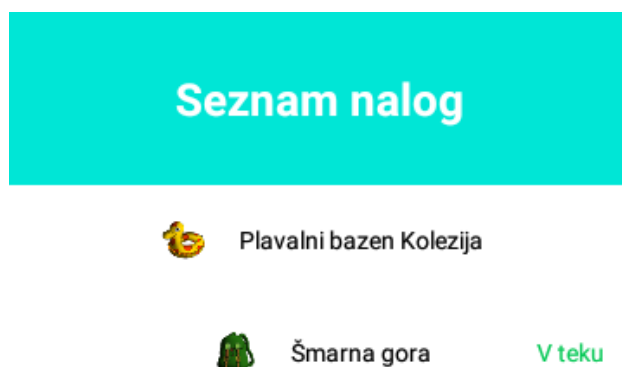
način, kot to naredi sprejemnik za zaznavo spremembe minutne vrednosti na sistemski uri.

Ta sprejemnik, ob vzpostavitvi internetne povezave, pošlje tudi vse podatke iz podatkovne baze za naloge, ki si jih bomo pogledali v naslednjih podsklopih.

Še zadnji sprejemnik in tudi zadnja komponenta, ki teče v ozadju aplikacije Dr. Gibalko, je **sprejemnik za zaznavo zagona sistema**. Sprejemnik se aktivira vedno ob zagonu operacijskega sistema Android. Njegova edina naloga pa je ta, da zažene oba zgoraj omenjena servisa: servis za zaznavanje aktivnosti in izračun porabe energije ter servis za štetje korakov. S tem zagotovimo, da ta dva servisa delujeta neprestano tudi ob ponovnem zagonu operacijskega sistema Android.

4.1.3 Zaslona za pregled nalog na pametni uri

Na tem zaslonu lahko uporabnik vidi seznam nalog, ki mu jih je preko spletne aplikacije dodelil zdravnik (slika 4.8). Posebnost pri tem zaslonu je ta, da mora biti viden tudi, kadar pametna ura nima internetne povezave, saj so naloge zelo pogosto na krajih, kjer ni brezžičnega omrežja, in je prikaz tega seznama nujen, da lahko uporabnik sploh začne z opravljanjem naloge. Zato si aktivnost tega zaslona vedno, kadar je vzpostavljena internetna povezava, v datoteko XML *sharedpreferences* shrani ves seznam, ki ga dobi v formatu JSON s strežnika.



Slika 4.8: Zaslón za pregled nalog.

Če smo v prejšnjem podsklopu dejali, da se lahko v datoteko *sharedpreferences* shrani samo primitivne tipe podatkov, tukaj pa shranjujemo celoten seznam, bi se lahko mnogi vprašali, kako je to mogoče. V resnici pa je zadeva precej preprosta, saj lahko format JSON, ki vsebuje seznam nalog s strežnika, shranimo v datoteko *sharedpreferences* kot niz (angl. *string*), ki pa ima identično strukturo kot format JSON. Pravzaprav tudi podatke s strežnika dobimo kot niz in jih nato pretvorimo v Javanski objekt *JSONObject* in enako naredimo, kadar ni internetne povezave, ko iz datoteke *sharedpreferences* preberemo nazadnje osvežen niz s strežnika in ga prav tako pretvorimo v Javanski objekt *JSONObject*.

Ob izbiri naloge s seznama, aplikacija za podrobnejši pregled posamezne naloge, preusmeri uporabnika na zaslon.

4.1.4 Zaslona za podrobnejši pregled posamezne naloge na pametni uri

Na tem zaslonu lahko uporabnik poleg naslova naloge, ki ga je videl že na prejšnjem zaslonu, vidi tudi natančnejši opis naloge. V tem opisu lahko zdravnik uporabniku natančneje opiše zahteve za uspešno opravljeno nalogo. Kot prikazuje slika 4.9, lahko na spodnjem delu zaslona vidimo, da ima uporabnik možnost, da nalogo bodisi začne opravljati bodisi jo konča. V primeru, da uporabnik želi začeti z opravljanjem naloge, kljub temu, da je naloga že v teku, ga aplikacija o tem opozori, enako velja, če želi zaključiti nalogo, ki še ni v teku.



Slika 4.9: Prikaz zaslona za podrobnejši pregled posamezne naloge.

Kot je že omenjeno v poglavju Funkcionalne zahteve, trenutno aplikacija Dr. Gibalko zaznava in ponuja opravljanje treh tipov nalog:

- pohodništvo,
- kolesarjenje,

- plavanje.

Zdravnik preko spletne aplikacije vnese različna pravila, ki so zahtevana za uspešno opravljeno nalogo: najprej s seznama izbere uporabnika, kateremu je naloga dodeljena, nato izbere ime in opis naloge, sledi izbira enega izmed treh (zgoraj naštetih) tipov nalog, nato določi rok, do katerega je nalogo potrebno opraviti, lokacijo s koordinatami GPS, in najmanjše zahtevano trajanje naloge ter točke, ki si jih uporabnik prisluži, če nalogo uspešno opravi.

Ta servis iz aktivnosti za podrobnejši pregled naloge prejme naslednje podatke o nalogi: identifikacijsko številko naloge, tip naloge in identifikacijsko številko uporabnika; hkrati pa inicializira tudi senzor GPS, iz katerega lahko razberemo lokacijo uporabnika. Servis nato na intervalu ene minute prebere vrednosti *latitude* in *longitude* iz senzorja GPS. Tako si vsako minuto v podatkovno bazo shrani vse podatke o nalogi, ki je v teku: identifikacijsko številko naloge, tip naloge, identifikacijsko številko uporabnika, časovno oznako, za vsako minuto posebej, in koordinati GPS.

Ob kliku na gumb „STOP TASK“ (zaključek naloge) servis shrani še zadnje koordinate GPS in se zaključi. Po tem, ko se je servis zaključil, aktivnost za podrobnejši pregled naloge, v primeru vzpostavljene internetne povezave, pošlje prav vse podatke iz podatkovne baze na strežnik. Če so podatki uspešno poslani, ta aktivnost izbriše vse že poslane podatke iz podatkovne baze. V primeru, da v trenutku, ko se servis zaključi, ni internetne povezave, se bodo ti podatki poslali, ko bo sprejemnik za zaznavo spremembe internetne povezave zaznal, da je vzpostavljena internetna povezava; podatki pa se iz sprejemnika pošljejo na enak način, kot se pošljejo iz te aktivnosti.

Za vsak tip naloge smo določili pravila, kako bomo preverjali, če je uporabnik zares opravil dano nalogo. Sicer so preverjanja med različnimi tipi nalog precej podobna:

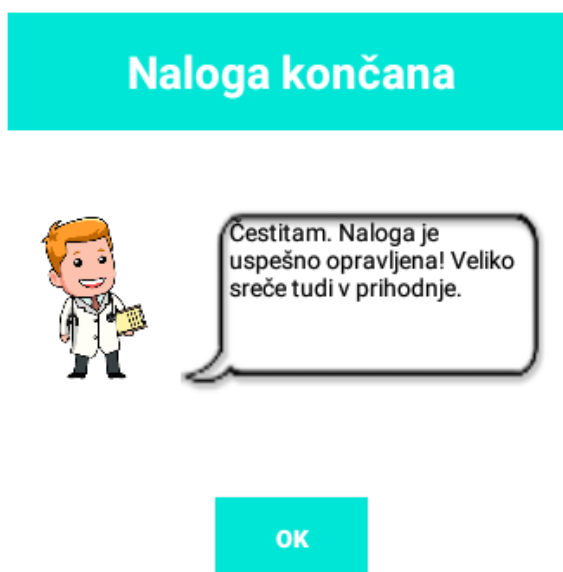
- pohodništvo: zdravnik določi koordinati GPS in uporabnik mora z vsaj eno izmed zabeleženih minutnih koordinat biti v radiusu dvestotih metrov od določene lokacije. Časovno ta naloga ni omejena;

- kolesarjenje: zdravnik določi več lokacij, ki jih uporabnik mora prevoziti, določi pa tudi čas trajanja, ki odmeri, koliko časa mora uporabnik opravljati aktivnost. Na ta način preprečimo uporabnikom, da bi ga starš ali druga oseba peljala z drugim prevoznim sredstvom;
- plavanje: za ta tip naloge zdravnik določi le eno lokacijo (lokacijo bazena), določi pa tudi primerno minimalno trajanje naloge.

To preverjanje se izvede na strežniku na podlagi poslanih podatkov iz pametne ure. Strežnik nam nato vrne status o zaključeni nalogi, ki ga kot obvestilo o opravljeni nalogi prikažemo na novem zaslonu.

4.1.5 Zaslon za obvestilo o opravljeni nalogi na pametni uri

Aktivnost tega zaslona je v primerjavi z ostalimi aktivnostmi na pametni uri najenostavnejša. Edina naloga te aktivnosti je, da na podlagi statusa o opravljeni nalogi prikaže ustrezno besedilo v oblaku, kot je to prikazano na sliki 4.10.



Slika 4.10: Prikaz zaslona za obvestilo o opravljeni nalogi.

Možni statusi, ki jih vrne strežnik, so:

- naloga je uspešno opravljena,
- naloga ni uspešno opravljena, saj je lokacija napačna, trajanje izvajanja naloge pa je v redu,
- naloga ni uspešno opravljena, saj je trajanje izvajanja naloge prekratko, lokacija pa je pravilna,
- naloga ni uspešno opravljena, saj sta, tako trajanje izvajanja naloge kot tudi lokacija napačna.

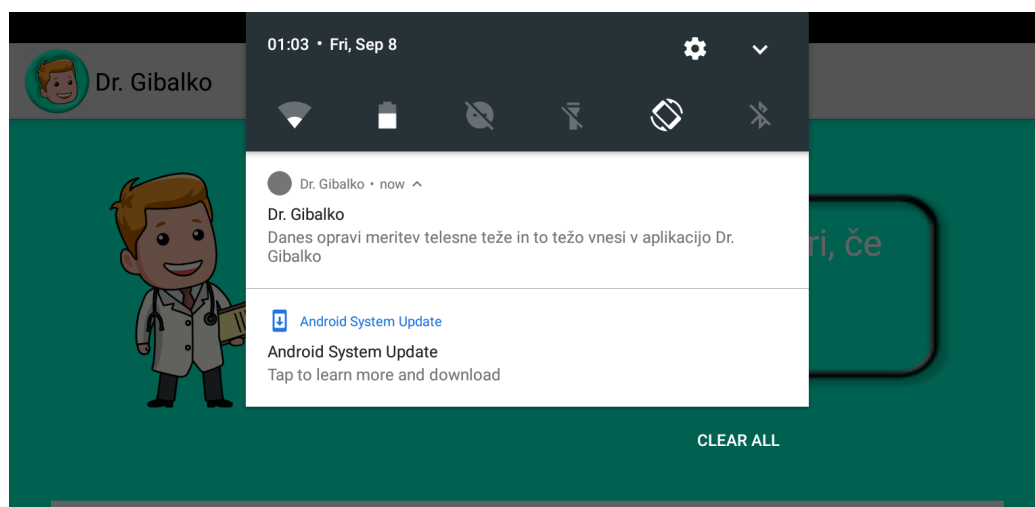
V zgoraj opisanih sklopih in podsklopih smo na kratko opisali vse servise, sprejemnike in aktivnosti, ki jih ima aplikacija Dr. Gibalko na pametni uri. Vidimo, da ima na prvi pogled aplikacija zelo malo funkcionalnosti, vendar se v ozadju skriva precej komponent, ki vseskozi delajo in (nenehno) shranjujejo podatke v podatkovno bazo, le-te pa pošiljajo na strežnik na različne načine glede na stanje internetne povezave.

4.2 Aplikacija na tabličnem računalniku

Cilj aplikacije na tabličnem računalniku je, da ima uporabnik pregled nad vsemi aktivnostmi, ki jih je opravil s pametno uro. Aplikacija na tabličnem računalniku ima tako precej več interakcije z uporabnikom kot aplikacija na pametni uri. Posledica tega pa je, da ima ta aplikacija tudi večje število zaslonskih mask, a hkrati nima nobenih servisov ali sprejemnikov, ki bi delovali v ozadju.

Aplikacija na tabličnem računalniku je, v nasprotju z aplikacijo na pametni uri, še nekoliko bolj v prototipnem stanju. Na začetku je pomembnejše, da lahko začnemo testirati samo zbiranje podatkov, ki jih zajema aplikacija na pametni uri, kako točno bodo le-ti prikazani, pa se bomo še dogovorili v bližnji prihodnosti. Kljub temu je v sklopu diplomske naloge narejena aplikacija za tablični računalnik z osnovnimi funkcionalnostmi.

Preko aplikacije na tabličnem računalniku je lahko uporabnik, preko potisnih obvestil (angl. *push notifications*), tudi obveščen, ko prejme novo nalogo od zdravnika in ob uspešno opravljeni nalogi. Preko potisnih obvestil pa lahko zdravnik tudi pošlje npr. motivacijsko sporočilo ali pa ga opomni na kakšen dogodek, kot prikazuje slika 4.11. Za pošiljanje potisnih obvestil smo uporabili platformo *Firebase*.

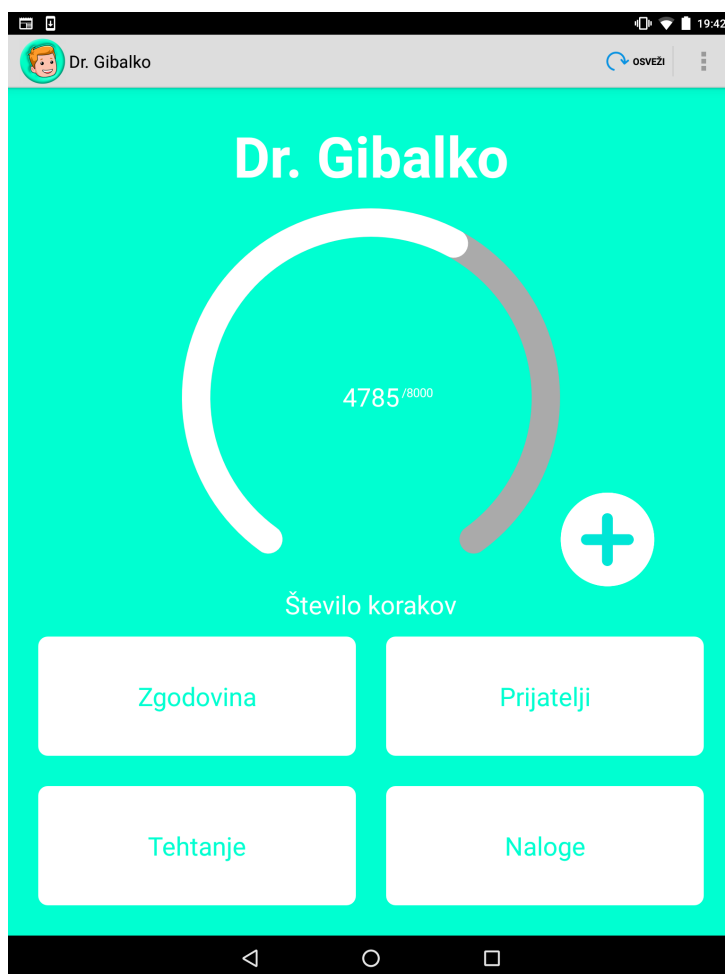


Slika 4.11: Prikaz potisnega obvestila.

4.2.1 Domači zaslon na tabličnem računalniku

Ob uspešni prijavi uporabnika na prijavnem zaslonu (delovanje prijavnega zaslona na tabličnem računalniku je popolnoma enako kot delovanje prijavnega zaslona na pametni uri) nas aplikacije preusmeri na domači zaslon. Na tem zaslonu (slika 4.12) lahko uporabnik vidi število prehojenih korakov, ki jih je zaznala pametna ura. Edina razlika med krožnim prikazovalnikom na pametni uri in na tabličnem računalniku je, da se na pametni uri posodablja takoj ob detekciji koraka, na tabličnem računalniku pa le takrat, kadar pametna ura pošlje podatke za preteklo uro.

Uporabnik lahko v zgornji orodni vrstici osveži podatke, preko menija pa se lahko odjavi ali zapre aplikacijo. Ob krožnem prikazovalniku korakov je slika v obliki znaka plus „+“, ki uporabnika preusmeri na zaslon, kjer lahko ročno vnese aktivnosti, klik na krožni prikazovalnik prehojenih korakov pa uporabnika preusmeri na zaslon, kjer vidi svoj profil. V spodnjem delu aplikacije so štirje gumbi, ki uporabnika preusmerijo do štirih glavnih komponent aplikacije na tabličnem računalniku.



Slika 4.12: Prikaz domačega zaslona na tabličnem računalniku.

4.2.2 Zaslona za prikaz uporabnikovega profila

Na tem zaslonu uporabnik vidi svoj profil. Izpiše se njegovo ime, uporabniško ime in priimek. Vidi tudi svojo trenutno telesno težo, ki se primerja z njegovo prvo vnešeno težo v aplikacijo Dr. Gibalko. Izpiše se mu tudi podatek o njegovi starosti. Na sliki 4.13 pa vidimo tudi tri medalje. Medalje oziroma različne barve medalj, uporabnik dobiva z opravljanjem dodeljenih nalog. Uporabnik ima za vsak tip naloge štiri različne možnosti prikaza:

- brez medalje,

- bronasta medalja,
- srebrna medalja,
- zlata medalja.



Slika 4.13: Prikaz profilnega zaslona.

Če uporabnik ni opravil še nobene naloge nekega tipa, potem medalje nima (plavanje na sliki 4.13). Že ko opravi prvo nalogo določenega tipa, uporabnik pridobi bronasto medaljo. Za pridobitev medalje srebrne barve, mora uporabnik opraviti dvajset nalog, za zlato pa petdeset nalog. Te vrednosti se bodo lahko čez čas še spremenile. Veliko smo se pogovarjali tudi

o tem, ali bi za omenjene nivoje imeli konstantne vrednosti ali pa bi bilo bolje imeti nivoje razporejene procentualno (npr. najboljših deset procentov uporabnikov ima zlato medaljo itn.). Vendar pa ta način omogoča možnost odvzema medalje nekomu, ki se sicer trudi, vendar mogoče ni fizično tako sposoben, kot so njegovi prijatelji. O tej temi smo se že pogovarjali s psihologi iz UKC Maribor, vendar še nismo prišli do končne rešitve.

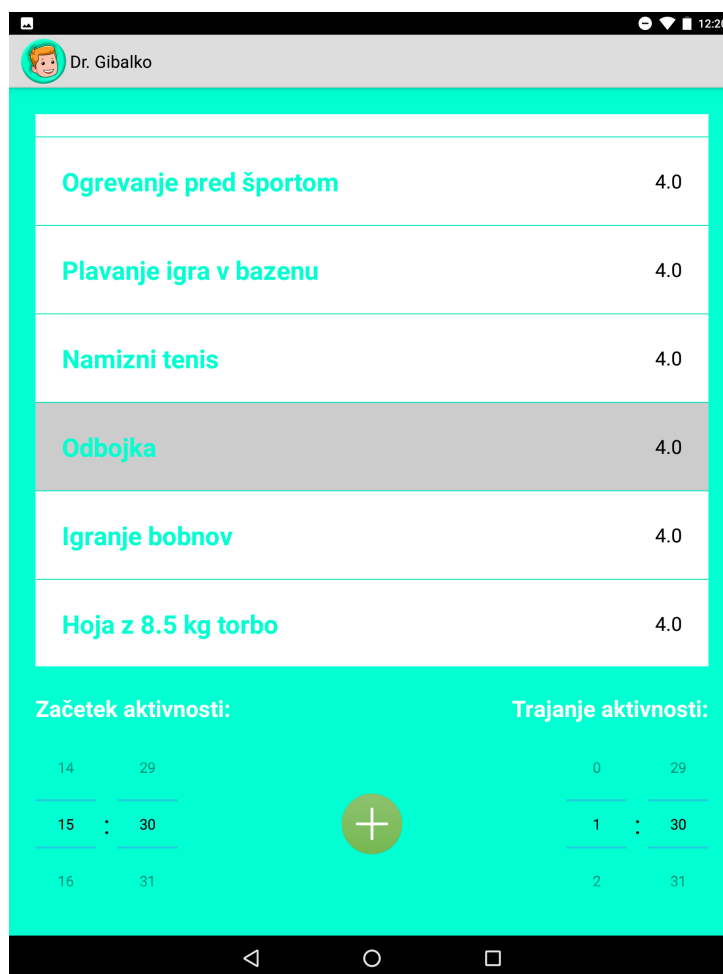
4.2.3 Zaslona za ročni vnos aktivnosti

Na tem zaslonu lahko uporabnik doda aktivnosti, ki jih je opravil v tekočem dnevu (slika 4.14). Te aktivnosti lahko doda, če pri opravljanju aktivnosti pri sebi ni imel pametne ure. Vendar pa bi, po mnenju zdravnika, otroci lahko to funkcionalnost izkoriščali. Zato smo se odločili, da se ročno dodane aktivnosti ne štejejo v isto kategorijo kot npr. naloge ali dnevni cilj. Prav tako se te aktivnosti ne bodo prikazovale na grafu pri pregledu zgodovine aktivnosti. Te vnešene aktivnosti bodo prikazane le zdravniku v spletni aplikaciji. Zdravnik bo tako lahko imel pregled nad vsemi aktivnostmi uporabnika, vendar tem vnešenim aktivnostim, ne bo dal velike prioritete.

Za ročni vnos aktivnosti, mora uporabnik s seznama izbrati eno aktivnost, ki ji določi čas začetka in trajanje. Ob pritisku na gumb „+“ se aktivnost pošlje na strežnik. Od Fakultete za šport smo, za vsako aktivnost na seznamu pridobili podatek o MET-u aktivnosti. Ker poznamo trajanje aktivnosti, vrednost MET in uporabnikovo težo, lahko izračunamo tudi porabo energije, ki jo je uporabnik pri opravljanju aktivnosti porabil.

4.2.4 Zaslona za prikaz zgodovine aktivnosti na tabličnem računalniku

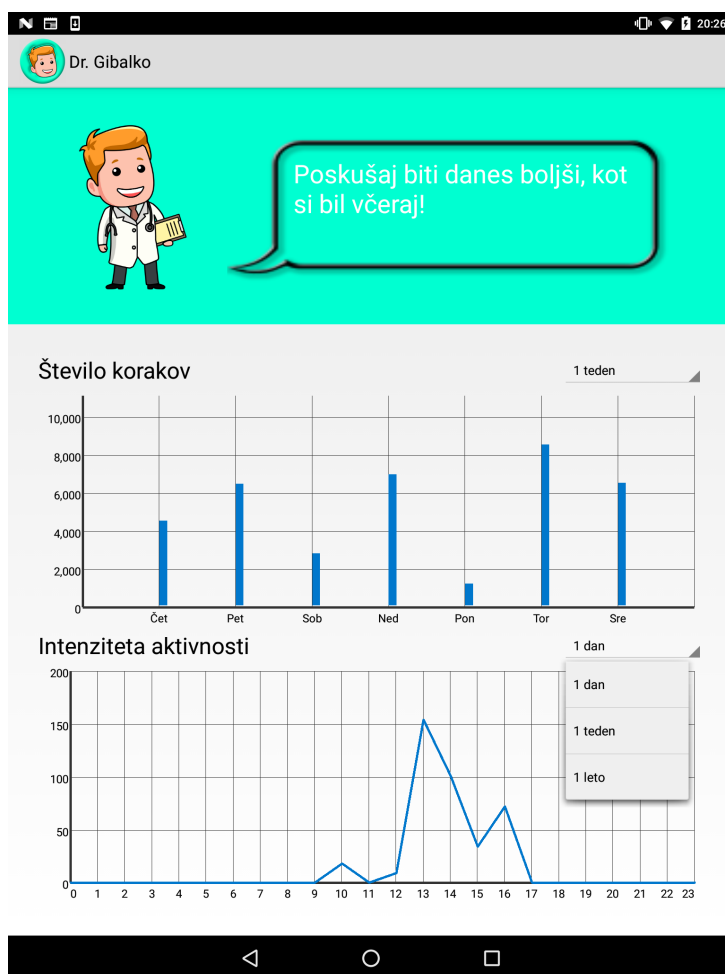
Ta zaslon je namenjen prikazu zgodovine aktivnosti uporabnika za različna časovna obdobja. Uporabnik lahko na zgornjem grafu vidi število prehojenih korakov, na spodnjem pa porabo kalorij (slika 4.15). Zaslon ima zgoraj tudi motivacijsko sporočilo, ki, skladno s prikazano tematiko, motivira upo-



Slika 4.14: Prikaz zaslona za ročni vnos aktivnosti.

rabnika. Uporabnik lahko za prikaz aktivnosti izbira med tremi časovnimi obdobji:

- en dan,
- en teden in
- eno leto.



Slika 4.15: Prikaz zaslona za pregled zgodovine aktivnosti uporabnika.

4.2.5 Zaslona za prikaz zgodovine teže na tabličnem računalniku

Na tem zaslonu lahko uporabnik na grafu (slika 4.16) vidi svojo zgodovino vnešene telesne teže. S pritiskom na gumb „Vnesi težo“ pa se odpre dialog, v katerem lahko uporabnik vnese nov podatek o telesni teži. Z zdravnikom iz UKC Maribor smo se o vnosu telesne teže veliko pogovarjali. Izrazil je željo, da bi naredili nek sistem ali preverjanje, ki bi zagotavljajo, da uporabniki pri vnosu telesne teže ne bi vnašali lažnih ali napačnih podatkov. Zato bomo v

prihodnosti aplikacijo, preko povezave *bluetooth* povezali z elektronsko tehtnico in bo vnos telesne teže omogočen le preko tehtnice.



Slika 4.16: Prikaz zaslona za pregled zgodovine telesne teže.

4.2.6 Zaslون za prijatelje na tabličnem računalniku

Na tem zaslonu, ki ga prikazuje slika 4.17, lahko uporabnik vidi seznam vseh svojih prijateljev. Seznam prijateljev prikazuje tri kategorije:

- prejete prošnje za prijateljstvo: vsebuje seznam drugih uporabnikov, ki so uporabniku poslali prošnjo za prijateljstvo,

- prijatelji: vsebuje seznam drugih uporabnikov, s katerimi je uporabnik že prijatelj,
- poslane prošnje za prijateljstvo: vsebuje seznam drugih uporabnikov, katerim je uporabnik poslal prošnjo za prijateljstvo, vendar jo drugi uporabniki še niso sprejeli.



Slika 4.17: Prikaz zaslona za prijatelje.

Uporabnik lahko na tem zaslonu sprejme ali zavrne prejeta prošnjo za prijateljstvo, lahko pošlje novo prošnjo za prijateljstvo ali pa prekliče poslano prošnjo za prijateljstvo.

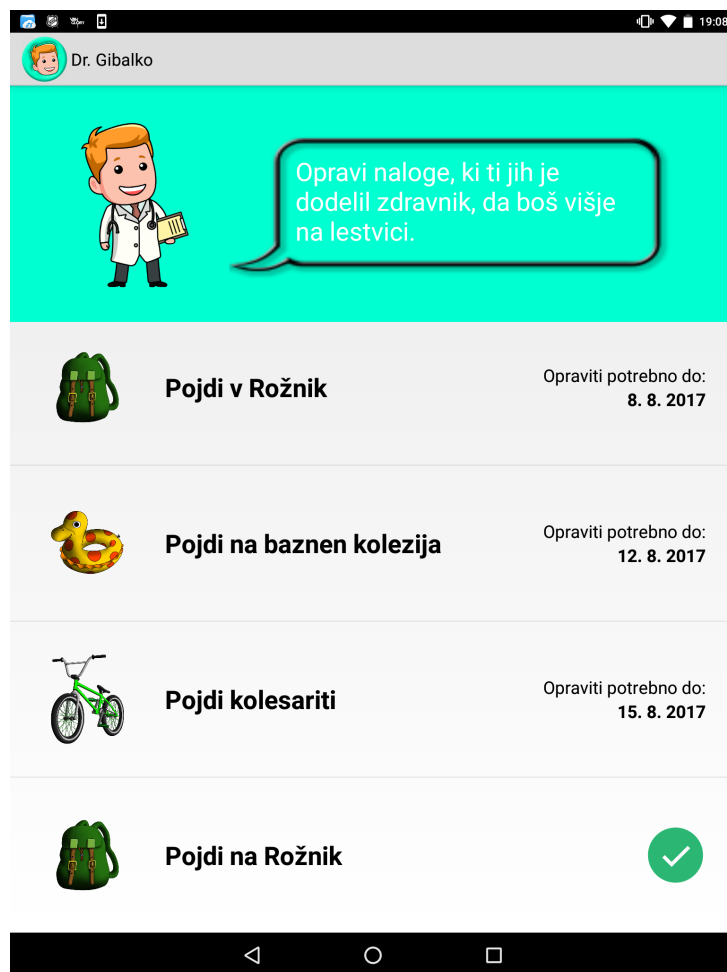
Uporabnik bi na tem zaslonu lahko izzval svoje prijatelje, vendar izzivi v tej prototipni verziji še niso popolnoma dokončani. Izzivi bodo sicer potekali v številu prehojenih korakov. Izzivalec si bo lahko izbral tudi trajanje izziva. Izziv se začne v trenutku, ko uporabnik izziv sprejme, in traja toliko časa, kot je to določil izzivalec.

4.2.7 Zaslón za prikaz nalog na tabličnem računalniku

Na tem zaslonu uporabnik vidi seznam vseh nalog, ki mu jih je dodelil zdravnik. Seznam nalog pa je lahko različen, kakor seznam nalog na aplikaciji pametne ure. Na pametni uri namreč, uporabnik vidi samo neopravljene naloge, ki jim še ni potekel datum, do katerega mora biti naloga opravljena. Pri aplikaciji na tabličnem računalniku pa vidi seznam vseh, tako opravljenih kot tudi neopravljenih, nalog, ki jim še ni potekel datum. Seznam nalog je tudi urejen, tako da so na vrhu seznama še neopravljene aktivnosti, spodaj pa opravljene aktivnosti. Znotraj opravljenih in neopravljenih nalog pa so naloge urejene po datumu zapadlosti, kot to prikazuje slika 4.18.

4.2.8 Zaslón za podrobnejši pregled posamezne naloge na tabličnem računalniku

Za razliko od aplikacije na pametni uri, kjer lahko uporabnik na tem zaslonu začne ali ustavi izvajanje naloge, je ta zaslon namenjen zgolj za podrobnejši ogled naloge (slika 4.19). Uporabnik vidi naslov naloge, opis naloge in datum, do katerega mora opraviti dodeljeno nalogo.



Slika 4.18: Prikaz zaslona za pregled zgodovine telesne teže.



Slika 4.19: Prikaz zaslona za podrobnejši pregled posamezne naloge na tabličnem računalniku.

Poglavje 5

Zaključek

V sklopu diplomske naloge smo idejo zdravnika, z uporabo sodobnih tehnologij in orodij, najprej zapisali, nato pa naredili model v obliki diagrama primerov uporabe in iz njega razvili dve aplikaciji za pomoč otrokom s prekomerno telesno težo. Največji izziv je bil razvoj aplikacije za pametno uro, ki omogoča zajem, obdelavo, zaznavanje in pošiljanje različnih podatkov na strežnik. Pri aplikaciji za tablični računalnik pa je bil največji izziv narediti vizualizacijo vseh podatkov, ki jih zajema aplikacija na pametni uri. Pri prikazu teh podatkov smo si želeli, da je otrok še dodatno motiviran h gibanju.

V sklopu diplomskega dela smo izdelali prototip, katerega pa bomo posodabljali še naprej. Prvi večji korak, ki se ga bomo lotili, je omogočiti vnos telesne teže preko elektronske tehtnice, saj si želimo, da so podatki o telesni teži uporabnika resnični. Na IJS-ju smo že začeli s snemanjem učnih podatkov za prehranjevanje in želja je, da bomo lahko v sklopu prepoznavanja aktivnosti zaznali tudi prehranjevanje uporabnika. Pogovarjali smo se tudi o spanju, saj je spanje pomemben dejavnik za otrokovo zdravje. Slab spanec pa je pomemben problem na področju pediatrije [6].

Zdravnik je z aplikacijo zadovoljen in kmalu bomo začeli s testiranjem celotnega sistema na prvi skupini otrok. Na testiranjih bomo dobili tudi povratne informacije s strani otrok, na podlagi katerih bomo lahko aplikacijo še izboljšali in tako prišli korak bliže k željenemu cilju: zagotoviti učinkovitejše

zdravljenje debelosti pri mladostnikih.

Literatura

- [1] Magdalena Avbelj, Nada Saje-Hribar, Margareta Seher-Zupancic, Polona Bracar, Primoz Kotnik, Andrej Irsic, Nina Bratanic, Ciril Krzisnik, in Tadej Battelino. Prevalenca čezmerne prehranjenosti in debelosti med pet let starimi otroki in 15 oziroma 16 let starimi mladostnicami in mladostniki v Sloveniji. *Zdravniski Vestnik*, 74(12), 2005.
- [2] Douglas Crockford. The application/json media type for javascript object notation (json). Dosegljivo: <https://tools.ietf.org/html/rfc4627>. [Dostopano 20. 8. 2017] , 2006.
- [3] Sheran Gunasekera. *Android Apps Security*. Apress, 2012.
- [4] G Jurak, M Kovač. *Ali kurikularne spremembe dohajajo spremembe v življenjskih slogih otrok?*. Sodobna pedagogika, 2009.
- [5] Mitja Luštrek, Božidara Cvetković, Vito Janko, Boro Štrumbelj, Jožef Štihec, and Tanja Kajtna. Mobilna aplikacija za spodbujanje gibanja šolarjev in učinkovitejšo športno vzgojo. VIVID, 2017
- [6] B G Stražišar. Spanje in motnje spanja pri otrocih in mladostnikih. *Slov Pediatr, letnik 19, 2012/2*, stran 195.
- [7] Wikipedia. Use case diagram — Wikipedia, the free encyclopedia, 2017. [Dostopano: 23. 8. 2017].
- [8] Wikipedia. User story — Wikipedia, the free encyclopedia, 2017. [Dostopano: 23. 8. 2017].