

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Gregor Kužnik

**Mobilna aplikacija za navigacijo
prilagojena slepim in slabovidnim**

DIPLOMSKO DELO

UNIVERZITETNI ŠTUDIJSKI PROGRAM
PRVE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: prof. dr. Saša Divjak
SOMENTOR: doc. dr. Matija Marolt

Ljubljana, 2016

Rezultati diplomskega dela so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavljanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje 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:

Tematika naloge:

V okviru diplomske naloge razvijte mobilno aplikacijo, ki slepim in slabovidnim olajša navigacijo po mestu. Pri tem preučite druge aplikacije, ki se uporabljajo v podobne namene in upoštevajte standarde za razvoj prilagojenih aplikacij za slepe in slabovidne. Aplikacija naj omogoča shranjevanje priljubljenih poti, enostavno usmerjanje do zelenega cilja in opozarjanje na pomembne točke v bližini.

Zahvaljujem se mentorju prof. dr. Saši Divjaku in somentorju doc. dr. Matiji Maroltu za pomoč in napotke pri izdelavi diplomske naloge.

Prav tako se za pomoč in sodelovanje zahvaljujem Zavodu za slepo in slabovidno mladino Ljubljana.

Zahvaljujem se tudi svoji družini in Urški za vso pomoč in podporo.

Kazalo

Povzetek

Abstract

1	Uvod	1
1.1	Slepi in slabovidni	2
1.2	Predstavitev in analiza obstoječih rešitev	4
2	Uporabljene tehnologije in orodja	7
2.1	Android	7
2.2	Java	9
2.3	OpenStreetMap	10
2.4	MapQuest	12
2.5	GraphHopper	13
2.6	XML	14
2.7	JSON	14
2.8	Pretvorniki besedila v govor	15
3	Mobilna aplikacija Spremljevalec	17
3.1	Delovanje in glavne funkcionalnosti aplikacije	18
3.2	Prilagoditev aplikacije	29
4	Spletna aplikacija	35
5	Sklepne ugotovitve	39

Seznam uporabljenih kratic

kratica	angleško	slovensko
API	Application programming interface	Programski vmesnik
GPS	Global Positioning System	Globalni sistem pozicioniranja
IDE	Integrated development environment	Integrirano razvojno okolje
JSON	JavaScript Object Notation	Standard za izmenjavo podatkov
KML	Keyhole Markup Language	Označevalni jezik za zapis geografskih podatkov
OSM	OpenStreetMap	Prosto dostopni zemljevidi
POI	Point of interest	Točka interesa
SDK	Software development kit	Paket za razvoj programske opreme
TTS	Text To Speech	Pretvorba besedila v govor
W3C	World Wide Web Consortium	Konzorcij svetovnega spleta
WCAG	Web Content Accessibility Guidelines	Smernice za dostopnost spletne vsebine
XML	Extensible Markup Language	Razširljivi označevalni jezik
ZSSM	Institute for Blind and Partially Sighted Children of Ljubljana	Zavod za slepo in slabovidno mladino Ljubljana

Povzetek

Naslov: Mobilna aplikacija za navigacijo prilagojena slepim in slabovidnim

Avtor: Gregor Kužnik

Slepi in slabovidni se vsakodnevno srečujejo s težavami pri sprehajanju na prostem, zato si pri tem pomagajo z različnimi pripomočki. Danes takšen pripomoček predstavljajo tudi pametni telefoni, saj lahko ob uporabi ustreznih aplikacij slepim in slabovidnim hojo na prostem zelo olajšajo. V okviru diplomske naloge je bila v ta namen v sodelovanju z Zavodom za slepo in slabovidno mladino Ljubljana izdelana mobilna aplikacija za navigacijo prilagojena slepim in slabovidnim. Aplikacija sledi že obstoječim aplikacijam v tujih jezikih, upošteva standarde za prilagoditev in vsebuje še dodatne pomeni izdelane funkcionalnosti. Izdelana je v slovenskem jeziku in za naprave z operacijskim sistemom Android. V glavnem omogoča hranjenje najljubših poti, vodenje po poti, prosto sprehajanje in iskanje bližnjih točk interesa.

Ključne besede: navigacija, mobilna aplikacija, dostopnost, slepi in slabovidni.

Abstract

Title: A mobile application for navigation for the blind and visually impaired

Author: Gregor Kužnik

Every day blind and visually impaired people face difficulties when walking outdoors and therefore use a variety of tools for assistance. Smartphones with appropriate applications represent such a tool and can make walking outdoors a lot easier. For this purpose, we collaborated with the Institute for Blind and Partially Sighted Children of Ljubljana and we created a mobile application for navigation for the blind and visually impaired. The application is similar to the already existing applications made in foreign languages, it follows accessibility guidelines and contains additional custom-designed functionalities. It uses Slovenian language and it is made for devices running Android. Its main features include storage of the users favourite routes, route guidance, free walking mode and searching for nearby points of interest.

Keywords: navigation, mobile application, accessibility, blind and visually impaired.

Poglavje 1

Uvod

Danes so po svetu pametni telefoni že zelo razširjeni in množično v uporabi. Te naprave nas spremljajo v našem vsakdanjiku in nudijo širok spekter storitev, ki so lahko osnovne, kot npr. klicanje in pošiljanje sporočil, ali pa bolj napredne kot npr. GPS navigacija. Poleg privzetih storitev lahko preko aplikacij, katerih število na trgu danes hitro narašča, dostopamo tudi do dodatnih storitev.

Sprejemnik GPS je eden izmed številnih dodatkov, ki jih danes lahko najdemo v pametnih telefonih. Iz njega izpeljane aplikacije in storitve mnogim uporabnikom predstavljajo pomemben pripomoček, ki ga nekateri uporabljajo dnevno, nekateri pa le takrat, ko se znajdejo v novem oz. neznanem okolju. GPS sam po sebi nudi različne primere uporabe, ki so lahko enostavni, kot npr. prikaz trenutne lege uporabnika ali pa bolj kompleksni, kot npr. vodenje po željeni poti. Značilno je, da je sprejemnik v napravah povezan z zemljevidi in nam tako podatke predstavi na znan in razumljiv način. Naprave danes v splošnem vsebujejo privzeto aplikacijo za navigacijo in zemljevide. Podobne aplikacije lahko najdemo tudi na trgu aplikacij, ki za razliko od osnovnih, uporabljajo različne ponudnike storitev, ponujajo dodatne funkcionalnosti in so prilagojene za določeno skupino ljudi.

Med uporabniki pametnih telefonov je tudi delež takšnih, ki so slepi oz. slabovidni. Njim je uporaba telefona otežena in včasih predstavlja velik izziv.

Danes je večina pametnih telefonov temu prilagojena, saj vsebuje možnost vklopa načina dostopnosti (angl. *accessibility*), ki nudi dodatne funkcionalnosti in s tem olajša uporabo. Kljub prilagojenosti naprav, pa je na trgu malo primerov aplikacij, ki so prilagojene in namenjene tem uporabnikom. Neprilagojene aplikacije v večini primerov za slepe in slabovidne niso uporabne, ali pa so uporabne samo do neke mere. Njim prilagojene aplikacije temeljijo na govoru, saj se vsa navodila in akcije sporočajo slušno. Jezik posledično predstavlja pomembno vlogo pri tovrstnih aplikacijah. Na trgu trenutno še ne obstaja prilagojena aplikacija za navigacijo slepih in slabovidnih v slovenščini, medtem ko takšne aplikacije v tujih jezikih obstajajo.

V sodelovanju z Zavodom za slepo in slabovidno mladino Ljubljana (ZSSM) smo zastavili projekt, ki obsega izdelavo aplikacije za navigacijo slepih in slabovidnih v slovenskem jeziku. Narejena bi bila po principu podobnih, že obstoječih aplikacij, vendar bi poleg slovenskega jezika nudila še dodatne funkcionalnosti, izdelane glede na predloge ZSSM in uporabnikov.

Projekt predstavlja realen problem, čigar rešitev je prilagojena mobilna aplikacija za navigacijo, ki bi slepim in slabovidnim olajšala gibanje na prostem. V okviru diplomske naloge so bila pridobljena znanja in izkušnje glede omejitev slepih in slabovidnih, razvoja in prilagoditve mobilnih aplikacij ter same orientacije in navigacije.

1.1 Slepi in slabovidni

Glede na podatke Svetovne zdravstvene organizacije, je danes število slepih in slabovidnih ocenjeno na približno 285 milijonov ljudi, od tega jih je 39 milijonov slepih, 246 milijonov pa slabovidnih. Približno 65 % slabovidnih in 82 % slepih je starih nad 50 let [18].

V Sloveniji naj bi po neuradnih podatkih bilo okoli 10000 slepih in slabovidnih, od tega jih je 4000 včlanjenih v Zvezo društev slepih in slabovidnih Slovenije. Delež slepih se zaradi staranja prebivalstva v Sloveniji in po svetu povečuje. Zavod za slepo in slabovidno mladino Ljubljana, ki je sodeloval pri

izdelavi diplomske naloge, je edini zavod v Sloveniji, ki nudi izobraževalne programe za slepe in slabovidne od vrtca do srednje šole. Poleg tega jim nudi tudi podporo, svetovanje in različna usposabljanja [23, 24].

Danes zaradi hitrega razvoja na področju elektronskih naprav, pametni telefoni in tablični računalniki predstavljajo odlično orodje za pomoč slepim in slabovidnim. Priučitev uporabe teh je sicer težavna, vendar pa zaradi vseh prilagoditev in funkcij za dostopnost to več ne predstavlja ovire.

Raziskava iz leta 2012 je pokazala, da imajo slepi in slabovidni željo po priučitvi uporabe mobilnih naprav in aplikacij, pri tem sta glavna razloga za to prenosljivost teh naprav in družbeno sprejetje. To pa kaže tudi na dejstvo, da so aplikacije za dostopnost, ki so razvite na mobilnih napravah bolj uporabljene in se hitreje razširijo, kot tiste, razvite na posebnih napravah [17].

Poleg porasta števila naprav, na trgu narašča tudi število prilagojenih aplikacij, kar omogoča uporabo slednjih tudi na drugih področjih. Eno izmed področij, ki slepim in slabovidnim predstavlja veliko težavo, je gibanje in orientacija v neznanem okolju. V znanem okolju in po znanih poteh se gibljejo brez večjih težav, pri čemer si zunaj pomagajo z belo palico, psom vodičem ali pa s spremljevalcem.

Vsako zunanje gibanje, predvsem pa gibanje v neznanem okolju, si lahko olajšamo z uporabo pametnih telefonov ali drugih podobnih naprav. Pri tem se poslužujemo aplikacij na napravi, ki omogočajo prikaz zemljevidov, orientacijo in navigacijo, poleg tega pa koristimo tudi funkcije kot je zvočno sporočanje navodil. Velik del orientacije opravimo na podlagi vizualnih informacij preko zemljevida in okolice. Pri slepih je potrebno te informacije in samo uporabo naprave prilagoditi in jim to predstaviti na njim uporaben način. Tukaj gre predvsem za zvočno sporočanje informacij in prilagoditve aplikacij.

The World Wide Web Consortium (W3C) je skupnost, kjer različne organizacije v sodelovanju z javnostjo razvijajo spletne standarde. Eden izmed teh dokumentov je tudi Web Content Accessibility Guidelines (WCAG), ki

določa smernice za izboljšanje dostopnosti na spletu. Leta 1999 je bil napisan prvi takšen dokument - WCAG 1.0, leta 2008 pa ga je nadomestil WCAG 2.0, ki se uporablja še danes. Ker se je v zadnjem času močno povečala razširjenost mobilnih naprav, je W3C februarja 2015 izdal prvi javni osnutek dostopnosti na mobilnih napravah. Ta opisuje, kako lahko smernice in načela iz dokumenta WCAG 2.0 apliciramo na mobilne aplikacije. Dokument predstavlja le informativne smernice, ne pa tudi zahteve. V grobem sestoji iz štirih glavnih principov, ki so: zaznavanje, operativnost, razumevanje in robustnost [10, 20, 22].

Določene napotke za dostopnost lahko apliciramo tudi na razvoj mobilnih aplikacij za slepe in slabovidne. Zaradi relativno majhne velikosti ekrana mobilnih naprav, je za lažje zaznavanje priporočljivo, da v posameznem prikazu predstavimo samo pomembne podatke, da so elementi in pisava dovolj veliki, da uporabljamo dober kontrast ter da omogočamo spreminjanje velikosti pisave preko nastavitvev uporabnikove naprave. Za dobro delovanje operativnosti je priporočljivo poskrbeti tako, da so interaktivni elementi dovolj veliki in postavljeni na mesta, kjer se do njih lahko enostavno dostopa. Prav tako je tukaj priporočljivo, da se za upravljanje aplikacije uporabljajo enostavne kretnje. Za boljše razumevanje je priporočljivo ohranjati konsistentnost pri postavitvi elementov, nuditi navodila za uporabo zapletenih kretenj ter omogočiti enostavno razpoznavanje elementov, katerim sledijo akcije. Za doseg robustnosti aplikacije se priporoča, da omogočamo ustrezen vnos podatkov npr. preko zvoka, da morebitne vnaprej določene podatke predstavimo v obliki menijev in gumbov ter da aplikacija podpira in uporablja funkcije za dostopnost, ki jih ponuja platforma na kateri je ta razvita [10].

1.2 Predstavitev in analiza obstoječih rešitev

Kot že omenjeno ima trg aplikacij nekaj primerov prilagojenih navigacijskih aplikacij, pomanjkljivost teh pa je jezik v katerem so izdelane. Aplikaciji,

ki bosta predstavljeni v nadaljevanju sta izdelani v angleškem jeziku. Ta sicer večini uporabnikov ne predstavlja težav, vendar pa je uporaba aplikacij, katerih ključna komponenta je govor, veliko lažja v maternem jeziku. S tem med drugim zagotovimo uporabo tudi tistim, ki jim je angleščina tuja oz. tistim, ki tuje jezike razumejo slabše. Analiza aplikacij je bila del raziskovanja in je služila kot pomoč pri razvoju naše aplikacije.

1.2.1 WalkyTalky

Google Play je trgovina aplikacij, ki je namenjena napravam z Android operacijskim sistemom. Project Eyes-Free je na tem trgu eden izmed ponudnikov, ki izdeluje brezplačne aplikacije namenjene slepim in slabovidnim. Med njimi najdemo aplikacijo WalkyTalky [21]. Gre za prilagojeno aplikacijo namenjeno navigaciji, ki je bila s strani ZSSM izpostavljena kot primer in zgled za razvoj poslovenjene verzije.

Aplikacija je slabše dokumentirana in zastarela, zadnje posodobitve so bile izdane leta 2011. Gre za enostavno aplikacijo v angleškem jeziku, ki služi kot nekakšen vmesnik do privzete navigacijske aplikacije telefona. Uporabniku omogoča vnos cilja, ki ga skupaj s podatkom o trenutni lokaciji posreduje privzeti navigacijski aplikaciji, ki nato poskrbi za iskanje in navigacijo po poti. Dodatno nudi tudi možnost shranjevanja in urejanja najljubših poti ter navigacijo do lokacije kontaktov v imeniku, če ti podatki obstajajo. Aplikacija ni najbolje prilagojena, saj vsebuje majhne gumbe ter uporabniku ob uporabi ne daje veliko slušnih povratnih informacij. Implementacija najljubših poti je uporaben dodatek in ideja, ki je bila implementirana tudi v naši aplikaciji.

1.2.2 ViaOpta Nav

Novartis je globalno podjetje, ki se ukvarja z iskanjem rešitev na področju zdravstva. Tudi oni na trgu aplikacij nudijo nekaj brezplačnih aplikacij namenjenih ljudem z motnjami vida. Njihov izdelek, ViaOpta Nav [19], je aplikacija namenjena navigaciji slepih in slabovidnih. Za razliko od aplika-

cije WalkyTalky, je ta izdelana tudi za naprave z operacijskima sistemoma Windows in iOS. Aplikacija je novejša, redno posodabljana in vsebuje kar nekaj dodatnih funkcionalnosti.

Uporabniški vmesnik je dobro prilagojen. Gumbi so veliki, primerno postavljeni in pokrivajo celoten ekran. S tem uporabnikom omogočajo hitro priučitev in enostavnejšo uporabo. Elementi so obarvani z različnimi barvami, kar slabovidnim omogoča lažje ločevanje in razpoznavanje le-teh. Navigacija se izvaja znotraj same aplikacije in ne uporablja privzete aplikacije za navigacijo. Poleg navigacije omogoča dodajanje in urejanje najljubših poti, pri čemer lahko v pot dodamo tudi točke po svoji izbiri. Nudi tudi pridobitev podatkov o tem, kje se trenutno nahajamo in katere interesne točke se nahajajo v bližini. Aplikacija je dobro implementirana in uporabna, njene funkcionalnosti so lahko v pomoč pri razvoju tovrstnih aplikacij in služijo kot zgled.

Poglavje 2

Uporabljene tehnologije in orodja

Aplikacija je bila v okviru diplomskega dela implementirana za operacijski sistem Android. Poleg tega je bila izdelana in na strežnik postavljena spletna aplikacija, ki omogoča dodajanje in urejanje najljubših poti. V nadaljevanju sledi podroben opis tehnologij in orodij uporabljenih pri razvoju aplikacije.

2.1 Android

Operacijski sistem Android razvija podjetje Google in je primarno izdelan in namenjen napravam z zaslonom na dotik. Prva komercialna verzija je izšla septembra 2008, nove verzije pa redno izhajajo še danes. Verzije od 1.5 naprej so poimenovane po abecednem vrstnem redu, verzija 1.5 se tako imenuje Cupcake, zadnja izdana verzija 6.0 pa Marshmallow. Vsaka verzija omogoča uporabo določene stopnje programskega vmesnika (API), zato je pri izdelavi aplikacije potrebno paziti, da z uporabo novejših orodij ne izločimo ciljnih uporabnikov s starejšimi verzijami. Danes je Android najbolj razširjen operacijski sistem za pametne telefone, v svoji trgovini z aplikacijami Google Play ponuja več kot milijon različnih aplikacij.

V praksi razvoj aplikacij poteka v integriranih razvojnih okoljih (IDE) in

z uporabo ustreznega paketa za razvoj programske opreme (SDK). Android aplikacije se v večini razvijajo v programskem jeziku Java in z uporabo Android SDK. Ta nam nudi vrsto orodij, med katere spadajo vzorčni projekti, emulator, knjižnice in osnovna orodja za razvoj. IDE je programsko okolje napisano kot aplikacija, ki tipično vsebuje komponente kot so: urejevalnik kode, prevajalnik, razhroščevalnik in grafični vmesnik. Za razvoj Android aplikacij imamo danes na voljo več okolij. Do sredine leta 2014 je bil Eclipse uraden IDE za razvoj takšnih aplikacij, po letu 2015 pa ga je nadomestil Android Studio. Aplikacijo lahko naložimo in testiramo z emulatorjem ali pa uporabimo dejansko fizično mobilno napravo.

Google prav tako nudi spletno stran Android Developers namenjeno razvijalcem Android aplikacij. Ta nudi pomoč in podatke za zadnje izdane verzije. Na strani lahko najdemo dokumentacijo za knjižnice in razrede, splošna navodila za izdelavo aplikacij, osnovne primere uporabe in primere dobre prakse.

Aplikacija Android nastane tako, da kodo aplikacije, morebitne podatke in vire, Android SDK zapakira v arhivirano datoteko s končnico .apk. To lahko nato namestimo na naprave z Android operacijskim sistemom. Aplikacija na napravi živi v lastnem varnostnem peskovniku in na ta način zadosti principu najmanjših privilegijev. To pomeni, da vsaka aplikacija privzeto dostopa le do komponent, ki jih potrebuje za svoje delovanje, do ostalih delov sistema pa lahko dostopa le z ustreznimi dovoljenji. Razvijalec ob razvoju nastavi zahteve za ustrezna dovoljenja, uporabnik pa se mora ob namestitvi z njimi strinjati.

Android aplikacijo sestavljajo t.i. aplikacijske komponente, ki predstavljajo ključne gradnike. Poznamo štiri vrste aplikacijskih komponent, mednje spadajo: dejavnosti (angl. activities), storitve (angl. services), ponudniki vsebine (angl. content providers) in oddajni sprejemniki (angl. broadcast receivers). Za sistem Android je značilno, da lahko aplikacija zažene komponente drugih aplikacij in jih s tem ne rabi implementirati sama. Uporabnik ima občutek, da je uporabljena komponenta kar del aplikacije. Posamezne

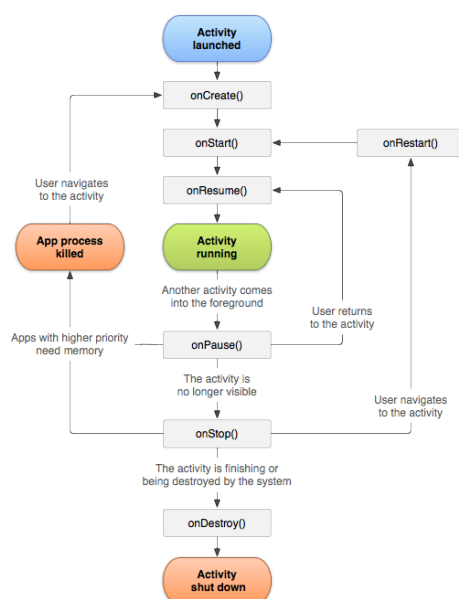
komponente deklariramo v manifestu oz. bolj podrobno v datoteki »AndroidManifest.xml«.

Aplikacijska komponenta dejavnost (angl. activity) je namenjena interakciji z uporabnikom. Dejavnost sama priskrbi okno, v katerem lahko implementiramo naš uporabniški vmesnik. Vsaka dejavnost je najprej zagnana, nato pa prehaja preko stanj dokler se ne zaključi. Potek stanj in metode, ki se prožijo med prehajanjem, nam prikazuje slika 2.1. Večina podrazredov dejavnosti implementira vsaj metodi *onCreate()* in *onPause()*. Ti dve sta eni izmed pomembnejših v življenjskem ciklu dejavnosti, saj se prožita na ključnih mestih. Metoda *onCreate()* se proži takoj po zagonu dejavnosti, tukaj tako vse skupaj inicializiramo in na tem mestu definiramo tudi uporabniški vmesnik. Metoda *onPause()* pa se proži po tem, ko dejavnost zapusti ekran oz. ko dejavnost zapusti stanje teka. Na tem mestu moramo zato preveriti, da so shranjene vse morebitne spremembe ustvarjene s strani uporabnika. V primeru da niso, poskrbimo za njihovo shranjevanje. Življenjski cikel dejavnosti predstavlja pomemben del pri razvoju in ga je potrebno dobro razumeti, saj nam pove, v katerem stanju se dejavnost lahko nahaja in kako med stanji prehaja [1].

2.2 Java

Java je objektno orientiran programski jezik (angl. Object-oriented programming language), uporabljen predvsem za izdelavo spletnih aplikacij. Leta 1995 sta ga razvila James Gosling in podjetje Sun Microsystems, danes pa spada pod podjetje Oracle. Java danes uporablja preko devet milijonov uporabnikov in spada med najbolj popularne trenutno uporabljane programske jezike. Sintaksa jezika je podobna kot pri jezikoma C in C++, z razliko, da je ta lažji za uporabo in uveljavlja model objektno orientiranega programskega jezika [6].

Android je ena izmed platform, kjer se pri programiranju aplikacij tipično in privzeto uporablja Java. Uporaba Java kot programskega jezika, je med



Slika 2.1: Življenjski cikel dejavnosti v Android aplikaciji [1].

drugim eden izmed razlogov za razširjenost Android aplikacij. Android in Java aplikacije se sicer razlikujejo in niso združljive, saj Android aplikacije ne tečejo na navideznem stroju Java (angl. Java Virtual Machine), kot vse Java aplikacije, ampak za to uporabljajo svoj navidezni stroj. Android je do verzije 4.4 uporabljal navidezni stroj Devik, po verziji 5.0 pa je bil privzeto uporabljen Android Runtime [7].

2.3 OpenStreetMap

OpenStreetMap (OSM) je projekt, ki ustvarja in nudi brezplačne geografske podatke. Podatki se dopolnjujejo s strani uporabnikov, po principu delovanja Wikipedije. Danes je registriranih uporabnikov že več kot dva milijona. Pridobljeni podatki se izdajajo pod licenco Open Database License, ki uporabnikom omogoča prosto uporabo, urejanje in deljenje podatkov. Razlog za razvoj tega projekta leži v dejstvu, da imajo ostali ponudniki takšnih storitev pravne in tehnične omejitve [11, 14].

Zemljevide in podatke lahko uporabimo in implementiramo tudi v mobilnih aplikacijah. Danes obstaja že kar nekaj primerov implementacij takšnih aplikacij, nekaj pa jih lahko najdemo tudi v trgovini aplikacij. Večina implementacij za platformo Android je odprtokodnih in se jih sme uporabiti za pomoč pri razvoju lastnih aplikacij.

2.3.1 Osmdroid in OSMBonusPack

Osmdroid [13] je odprtokodna knjižnica namenjena razvoju mobilnih Android aplikacij za interakcijo s podatki iz OSM. Knjižnica nudi orodja za prikaz in delo s podatki iz OSM, ter na ta način predstavlja alternativo knjižnici za delo z Google zemljevidi. Android naprave privzeto vsebujejo Google zemljevide, te in iz njih izpeljane storitve pa se ponavadi uporablja tudi pri razvoju navigacijskih aplikacij.

OSMBonusPack [12] je nadgradnja knjižnice Osmdroid in vsebuje dodatna orodja za delo in prikaz podatkov. Kot glavne razširitve lahko izpostavimo orodja za delo s točkami interesa, Keyhole Markup Language (KML) in GeoJSON formatoma ter navigacijo po poti. KML in GeoJSON formata se uporabljata za predstavitev geografskih podatkov, pri čemer prvi temelji na XML, drugi pa na JSON standardu.

2.3.2 Overpass API

Overpass API je API, ki omogoča bralni dostop do OSM podatkov. Služi kot nekakšna podatkovna baza in je optimiziran za poizvedovanje. Poizvedbe vsebujejo iskalne kriterije in so lahko zapisane v XML ali Overpass QL obliki [15].

Overpass QL je poizvedovalni jezik, ki je bil razvit kot alternativa XML poizvedovanju po OSM podatkih. Overpass QL ima C sintakso, poizvedba je tako sestavljena iz posameznih izjav, katere so med seboj ločene s podpičji in se procesirajo ena za drugo. Overpass QL poizvedbe nam omogočajo, da v enem HTTP zahtevku zapišemo več iskalnih kriterijev in s tem dostopamo

do vseh potrebnih podatkov [16].

Primer uporabe Overpass API-ja je iskanje interesnih točk (POI) v bližini uporabnika. To so točke, ki označujejo uporabniku pomembne lokacije na zemljevidu, kot so npr. bližnja avtobusna postajališča, hoteli, bencinske črpalke itd.

2.4 MapQuest

MapQuest je podjetje, ki nudi zemljevide in geografske storitve, pri tem pa uporablja tudi podatke projekta OSM. Podjetje med svojimi rešitvami ponuja tudi omrežje za razvijalce, kjer lahko dostopamo do brezplačnih programskih vmesnikov in spletnih storitev. Med temi so posebej zanimivi MapQuest Directions API oz. storitev, ki nam vrne navodila za željeno pot, ter MapQuest Geocoding API, ki služi pretvarjanju geografskih koordinat v naslove in obratno. Za uporabo storitev se je potrebno registrirati in pridobiti ključ API, ki nam pri dostopanju služi kot avtentikacija. Posamezen račun in njemu pripadajoč ključ lahko koristi le en uporabnik [8].

Sprejemnik GPS za našo lokacijo vrne koordinati oz. par vrednosti zemljepisne širine in dolžine. Ta podatek ponavadi uporabnikom ne predstavlja neke vrednosti in ni uporaben, zato je potrebno izvesti t.i. geokodiranje, ki ga nudijo aplikacije kot je npr. MapQuest Geocoding API. Pri geokodiranju gre za pretvorbo koordinat v naslov. Poznamo tudi vzvratno geokodiranje, pri čemer pa gre za obratno operacijo, saj se za določen naslov poišče koordinati. Geokodiranje tako uporabljamo kadar želimo uporabniku sporočiti njegovo trenutno lokacijo, vzvratno geokodiranje pa kadar želimo lokacijo predstaviti na zemljevidu ali pa želimo npr. najti pot med dvema točkama.

MapQuest Directions API nam preko zahtevka HTTP omogoča dostop do svojih usmerjevalnih algoritmov. Poleg začetne in končne točke, ter morebitnih vmesnih točk, lahko preko zahtevka nastavimo tudi ostale parametre. Eden izmed parametrov, ki ga lahko nastavimo je tip poti, tako lahko npr. zahtevamo pot za pešce. Nastavimo lahko tudi obliko v kateri naj bo rezul-

tat vrnjen, bodisi XML, bodisi JSON. V zahtevku moramo točke podati kot koordinate, zato je potrebno pred pošiljanjem zahtevka te najprej pretvoriti z uporabo vzvratnega geokodiranja. Podatek o trenutni lokaciji uporabnika namreč lahko dobimo v obliki koordinat, podatke o končni in vmesnih točkah pa uporabniki tipično vnašajo kot naslove.

Storitev kot je MapQuest Directions API, ki nam vrne navodila za pot v načinu »zavoj za zavojem« (angl. turn by turn), je še posebej uporabna za slepe in slabovidne. Takšni uporabniki za razliko od ostalih ne morejo uporabljati vizualnih informacij zemljevidov in se na ta način po njih orientirati. Za njih so uporabna predvsem navodila za usmerjanje, ki jih s sintetizatorjem pretvorimo v govorno obliko in nato sporočimo uporabniku.

MapQuest v svojem osnovnem brezplačnem paketu za svoje spletne storitve omogoča izvedbo 15000 transakcij mesečno, medtem ko npr. Google Maps v brezplačnem paketu za podobne storitve omogoča 2500 transakcij dnevno [4, 9].

2.5 GraphHopper

GraphHopper je tudi eden izmed odprtokodnih ponudnikov, ki poleg zemljevidov GraphHopper Maps ponuja hitro storitev GraphHopper Directions API za usmerjanje po poti. Podatke za zemljevide in ostale storitve pridobi iz projekta OSM [5].

GraphHopper Directions API deluje podobno kot storitev MapQuest Directions API - za podane točke najde pot in vrne navodila za usmerjanje. Za uporabo se je potrebno registrirati in pridobiti ključ API, s katerim nato preko zahtevkov HTTP dostopamo do njihovih storitev. Ključ pripada samo enem uporabniku in omejuje število zahtevkov. Vrnjen rezultat je lahko v obliki JSON ali GPX, kar preko parametra nastavimo v zahtevku. Prav tako lahko v zahtevku nastavimo tudi željen tip poti, kot je npr. pot za pešce.

OSM za iskanje poti za pešce in kolesarje privzeto uporablja storitev GraphHopper Directions API.

2.6 XML

Extensible Markup Language (XML) je označevalni jezik za dokumente, ki vsebujejo strukturirane podatke. Ta izvira iz oz. je podmnožica jezika Standard Generalized Markup Language, ki predstavlja standard za kreiranje strukture dokumenta. Strukturirani podatki so tisti, ki poleg vsebine vsebujejo tudi označbe o njeni vlogi.

Osnovni gradniki XML dokumenta so elementi, ti so definirani z začetno in končno značko. V dokumentu imamo vedno en sam korenski element. Element lahko vsebuje besedilo, attribute in druge elemente ali kombinacijo teh. Ker lahko elementi vsebujejo druge elemente, je s tem omogočena podpora tudi hierarhičnim strukturam. Medtem ko imena elementov opisujejo njihovo vsebino, struktura opisuje njihovo povezavo oz. razmerje.

Danes je značilno, da se dokumenti XML uporabljajo za izmenjavo podatkov preko omrežij. Večina spletnih storitev nam podatke lahko vrne v tej obliki, čeprav se danes za to vedno bolj uporablja anotacija JSON. Primer spletne storitve, ki lahko rezultat vrne tudi v obliki XML, je MapQuest Directions API.

Označevalni jezik XML se uporablja tudi pri razvoju Android aplikacij. V aplikacijskem manifestu »AndroidManifest.xml« se uporablja za deklaracijo komponent in ostalih podatkov aplikacije. Uporablja se lahko tudi za deklaracijo in postavitvev elementov uporabniškega vmesnika.

2.7 JSON

JavaScript Object Notation (JSON) je format za hranjenje in izmenjavo podatkov. Format JSON je enostaven za ljudi in računalnike, ljudem za pisanje in branje, računalnikom pa za razčlenitev in generiranje. Predstavlja alternativo jeziku XML in je načeloma lažji za uporabo.

JSON je zgrajen na podlagi dveh struktur. Prva je zbirka parov imen in vrednosti oz. *object*, druga je urejen seznam vrednosti oz. *array*. *Object* se začne z levim zavitim oklepajem in konča z desnim zavitim oklepajem,

imenom sledi dvopičje, pari imena in vrednosti so ločeni z vejico. *Array* se začne z levim oglatim oklepajem in konča z desnim oglatim oklepajem, vrednosti so ločene z vejico. Posamezna vrednost je lahko različnega tipa.

JSON se danes večinoma uporablja za izmenjavo podatkov med odjemalcem in strežnikom.

2.8 Pretvorniki besedila v govor

V aplikacijah se glasovno sporočanje izvaja preko aplikacij za pretvorbo besedila v govor (TTS). Te aplikacije uporabljajo sintezo govora, ki je računalniško generirana simulacija človeškega govora.

Večina današnjih pametnih telefonov že v osnovi vsebuje eno izmed privzetih aplikacij za pretvorbo besedila v govor, te ponavadi vsebujejo samo nekaj glavnih jezikov. Na trgu aplikacij lahko najdemo še druge implementacije, ki nudijo več jezikov. V privzetih aplikacijah za pretvorbo besedila v govor na platformi Android slovenščina za zdaj še ni vključena.

Trenutno sta v trgovini Google Play dve aplikaciji, ki omogočata pretvorbo besedila v govor v slovenskem jeziku. Prva aplikacija je eSpeak [3], ki je plačljiva in vključno s slovenščino vsebuje 79 jezikov. Druga aplikacija pa je eBralec [2], ki je bila izdelana s strani raziskovalnih organizacij in dveh slovenskih jezikovno-tehnoloških podjetij.

Aplikacijo eBralec se lahko namesti in uporablja na osebnih računalnikih ter mobilnih napravah. Pri mobilni aplikaciji sta na voljo osnovna brezplačna in nadgrajena različica. Nadgrajeno je moč pridobiti z nakupom licence in omogoča namestitvev dodatnih bolj kakovostnih glasov.

Poglavje 3

Mobilna aplikacija

Spremljevalec

V okviru diplomske naloge je bila razvita mobilna aplikacija za platformo Android. Platforma je bila za razvoj izbrana zaradi razširjenosti med uporabniki in lastnega predznanja na tem področju. Programiranje je potekalo predvsem v programskem jeziku Java. Prikazi, animacije in nastavitvene datoteke so napisane v jeziku XML. Za razvoj aplikacije je bilo uporabljeno okolje Android Studio, ki velja za privzeto okolje za razvoj izvirnih (angl. native) Android aplikacij. Določeni grafični elementi so bili posebej narisani in pripravljeni v programu Adobe Illustrator.

Razvoj in testiranje aplikacije je večino časa potekalo na napravi Samsung Galaxy S II Plus, ki ima nameščeno različico sistema Android 4.2.2. Uporabljen je bil tudi tablični računalnik HP Slate 7 z različico sistema Android 4.1.1 in nekaj emulatorjev.

Implementacija aplikacije je na začetku potekala z uporabo Google Maps ter storitvami Google Maps Direction API in Google Maps Geocoding API. Razvoj se je zaradi omejitev v pravilih uporabe teh orodij nato izvedel s ponudnikom OSM. OSM in iz njega izpeljane aplikacije in storitve so v večini odprtokodne in uporabe načeloma ne omejujejo. Kljub temu, da so nekatere izpeljane storitve morda manj kvalitetne in počasnejše, nudijo vse potrebno

za implementacijo aplikacije kot je naša.

Pri implementaciji zemljevidov in ostalih funkcionalnosti z OSM podatki sta bili kot pomoč uporabljeni odprtokodna knjižnica Osmdroid in njena razširitev OSMBonusPack.

Storitev MapQuest Directions API je bila pri razvoju uporabljena kot primarna storitev za pridobitev navodil za pot. Storitev GraphHopper je bila uporabljena kot alternativa in se jo lahko uporabi v primeru, da MapQuest Directions API ne bi deloval.

Za pridobivanje točk interesa je bil uporabljen Overpass API.

Del razvoja aplikacije je bil usmerjen v implementacijo načrtanih funkcionalnosti, del pa v prilagojenost aplikacije za slepe in slabovidne.

3.1 Delovanje in glavne funkcionalnosti aplikacije

Aplikacija je sestavljena iz štirih glavnih delov oz. prikazov: glavni meni, seznam najljubših poti, zemljevid ter orodja in pomoč. Uporabnik lahko med prikazi enostavno prehaja. Ob zagonu aplikacije se uporabniku odpre prikaz z glavnim menijem (slika 3.1), od koder lahko nato dostopa do seznama najljubših poti, proste orientacije oz. prikaza zemljevida ter orodij in pomoči.

3.1.1 Dovoljenja in uporabniški račun

Za dostop do določenih podatkov uporabnikove naprave potrebujemo dovoljenja, s katerimi se mora uporabnik za delovanje aplikacije strinjati. V aplikaciji je bilo potrebno nastaviti zahteve za dostop do omrežja in njegovega stanja, grobe in natančne lokacije ter pisanja v pomnilnik naprave.

Aplikacija je z uporabniškim računom povezana preko unikatne številke pin. Uporabniški račun in pin se generirata ob prvem zagonu mobilne aplikacije, kjer se v spletno aplikacijo pošlje temu namenjen zahtevek. Spletna aplikacija ob prejemu zahtevka poskrbi za kreiranje uporabniškega računa in



Slika 3.1: Glavni meni v mobilni aplikaciji.

številke pin ter podatke vrne mobilni aplikaciji, ta pa pin shrani lokalno v napravo. Shranjena številka pin se v mobilni aplikaciji uporablja za dostop do najljubših poti. Preko generirane številke pin lahko dostopamo tudi do spletne aplikacije, kjer lahko urejamo in dodajamo najljubše poti.

Uporabnik lahko do svoje številke pin dostopa v prikazu orodja in pomoč. Tam lahko z dolgim pritiskom na številko pin proži njeno branje. V prikazu orodja in pomoč lahko uporabnik dostopa tudi do osnovnih navodil za uporabo aplikacije.

3.1.2 Možnosti navigacije - najljubše poti in prosta orientacija

V prikazu najljubših poti se uporabniku na seznamu prikažejo vse najljubše poti, ki so urejene po vrstnem redu kot so bile vnešene v spletni aplikaciji. Prikaz z najljubšimi potmi je prikazan na sliki 3.2. Z dolgim pritiskom na posamezno najljubšo pot se uporabniku prebere naslov poti, ob navadnem



Slika 3.2: Seznam najljubših poti v mobilni aplikaciji.

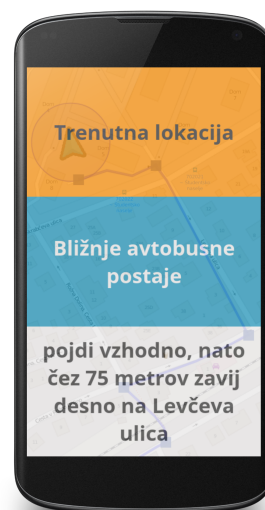
pritisku pa se zažene prikaz zemljevida, kjer se za izbrano pot izvede postopek iskanja in navigacije po poti.

Ob kliku gumba »Prosta orientacija« aplikacija prav tako preide v prikaz zemljevida, s tem da se za razliko od prehoda iz najljubših poti tukaj ne išče in izriše pot, ampak se prikazuje samo trenutna lokacija uporabnika. V tem načinu so nekoliko spremenjene tudi hitre akcije.

V prikazu zemljevida lahko uporabnik s tresenjem naprave prikliče meni s hitrimi akcijami. V tem oknu so na voljo tri akcije: trenutna lokacija, bližnje avtobusne postaje in navodila za pot oz. dostop do najljubših poti. Zadnja akcija se razlikuje glede na to, ali je uporabnik v načinu navigacije ali v načinu proste hoje. V primeru da je uporabnik v načinu proste hoje, je zadnja akcija povezava do najljubših poti. S tem uporabniku omogočimo, da lahko iz načina proste hoje v dveh klikih preide v način navigacije. V primeru da je uporabnik v načinu navigacije, je zadnja akcija trenutno navodilo za pot. Uporabnik v tem primeru ob kliku na akcijo proži branje navodila. Hitri meni in razlike med akcijami lahko vidimo na slikah 3.3 in 3.4.



Slika 3.3: Hitri meni v načinu proste hoje.



Slika 3.4: Hitri meni v načinu navigacije.

Prikaz zemljevida je implementiran v eni Android aktivnosti in omogoča dva načina uporabe. Prvi način uporabe je prosta hoja, drugi način pa navigacija po poti. Aktivnost glede na vhodne parametre ob klicu izbere način uporabe in ustrezno inicializira potrebne dele. Pri obeh načinih je začetni del inicializacije enak, saj gre tukaj za nalaganje zemljevidov, izris elementov in zagon poslušalcev za spremembe lokacije ter kretenj.

Skupni del, ki se izvaja pri obeh načinih, je sestavljen iz več korakov in se zažene v metodi *onCreate()*.

Inicializacija zemljevida

Najprej se inicializira zemljevid, kjer se glede na podanega ponudnika pridobi t.i. ploščice (angl. tiles), ki se jih združuje in izrisuje v zemljevid. Pri implementaciji je bilo za upodabljanje zemljevida uporabljeno odprtokodno orodje Mapnik. V tem koraku se nastavi tudi privzete vrednosti, omejitve in vklop ali izklop funkcij in elementov. V našem primeru so dodatni elementi

na zemljevidu izključeni, saj ciljnim uporabnikom ne predstavljajo nobene dodane vrednosti. Uporabniku bi navadni, neprilagojeni gumbi bili kvečjem v napoto. Minimalna povečava je omejena, saj gre za aplikacijo, ki služi hoji po mestu, kjer ni potreben pogled na večje dele zemljevida. Prav tako je omejitev smiselna zaradi nalaganja ploščic in varčevanja z mobilnimi podatki, saj se v primeru, da uporabnik nehote zmanjša povečavo, ne nalagajo novi deli zemljevida.

Uporabnikova lokacija

Sledi dodajanje in izris uporabnikove lokacije na zemljevid. V tem koraku preverimo ali obstaja podatek o zadnji lokaciji uporabnika. Če podatek obstaja, ga uporabimo za prvi izris lokacije uporabnika. V primeru da ne obstaja, pa element za prikaz lokacije skrijemo in ga prikažemo kasneje, ko ta podatek dobimo. Omembe vredno je tudi to, da se v primeru, da obstaja podatek iz sprejemnika GPS zaradi večje natančnosti uporabi ta, namesto podatka iz omrežnega sprejemnika.

Zagon poslušalcev

V naslednjem koraku, ki se izvaja v metodi *onResume()*, se zaženejo poslušalci. Metoda *onResume()* se med drugim kliče tudi ko se uporabnik vrne, po tem ko je zapustil aplikacijo. V metodi *onPause()*, ki se kliče kadar uporabnik zapusti pogled, pa poslušalce začasno ugasnemo. Poslušalca, ki ju zaženemo sta poslušalec za lokacijo uporabnika in poslušalec za zaznavanje kretenj.

Pri poslušalcu za kretnje gre za razred, v katerem se zaznava premike naprave in se ob ustreznem premiku to sporoča aktivnosti, ki ga je zagnala. Premike naprave poslušalec spremlja s senzorjem za pospešek. Vsakič ko se naprava premakne, se v poslušalca pošljejo vrednosti pospeškov za vse tri osi v prostoru. Iz teh vrednosti se izračuna skupna sila, ki se uporabi za določanje tresenja. Za določanje tresenja definiramo prag sile, s čimer določimo, kdaj je uporabnik namerno stresel napravo.

Pri poslušalcu za lokacijo gre za pridobivanje podatkov iz dveh različnih ponudnikov. Prvi ponudnik lokacije je GPS, drugi pa omrežja, bodisi omrežje Wi-Fi bodisi mobilno omrežje. Najbolj natančni, uporabni in zaželeni so podatki, ki jih pridobimo iz sprejemnika GPS. Vseeno pa se v primeru lokacij iz omrežja uporabljajo tudi te, čeprav imajo lokacije pridobljene iz GPS-a prednost. V tej točki za vsakega trenutno aktivnega ponudnika zaženemo poslušalec lokacije, ki ob spremembi lokacije kliče metodo *onLocationChanged()*. Metoda *onLocationChanged()* je prekrmljena iz razreda *LocationListener*, ki ga naša aktivnost razširjuje. Ta metoda na nek način predstavlja možgane delovanja te aktivnosti, saj se tukaj izvajajo vse glavne spremembe in klici. V grobem skrbi za tri dele, prvi del je pridobivanje in filtriranje lokacije, drugi del so klici za izris sprememb na zemljevidu, tretji del pa je navigacija po poti v primeru, da gre za način navigacije.

Filtriranje podatkov

Ker so podatki o lokaciji velikokrat nenatančni in napačni, je dobro izvajati nekakšno filtriranje. V ta namen je bilo implementirano osnovno filtriranje, pri čemer gre za izločanje manj pomembnih lokacij.

Ko dobimo nov podatek o lokaciji, ga tako najprej filtriramo glede na časovni razmak in ponudnika. V primeru, da takoj za lokacijo pridobljeno iz sprejemnika GPS dobimo lokacijo iz omrežja, jo ignoriramo. Ignoriramo jo, ker je manj natančna in ker lahko predpostavimo, da bo iz sprejemnika GPS v kratkem prišla nova lokacija. Ignoriramo tudi podatke o lokaciji med katerimi je zelo kratek časovni razmak, saj nam tako majhni premiki ne pomagajo in kvečjemu upočasnjujejo delovanje aplikacije. Pridobljeno lokacijo nastavimo kot trenutno ter spremenimo pozicijo uporabnika na zemljevidu. Na tem mestu nastavimo še posodobljena podatka o smeri uporabnika in natančnosti GPS podatkov, ki je na zemljevidu predstavljena v obliki modrega kroga okoli uporabnika.

Navigacija

Sledi del, ki skrbi za navigacijo. Tukaj najprej preverimo ali aktivnost teče v načinu navigacije, saj v nasprotnem primeru ta del preskočimo in nadaljujemo z naslednjim. Če gre za način navigacije preverimo ali je bila navigacija po poti že zagnana. V primeru da še ni, to pomeni, da smo uporabnikovo lokacijo prejeli prvič in moramo na tem mestu poskrbeti za iskanje poti in njen zagon. Za iskanje poti je bilo implementirano asinhrono opravilo, ki teče v ozadju in ne moti delovanja ostalih delov aktivnosti. Opravilo kot vhodni parameter sprejme seznam točk, za katere želimo najti pot. Točke v seznamu si morajo slediti po vrstnem redu, v katerem jih želimo obiskati. Kot prvo točko tako dodamo trenutno lokacijo uporabnika, kot zadnjo pa željeno destinacijo oz. naslov najljubše poti. Najljubše poti uporabnik preko spletne aplikacije vnaša kot naslove, točke v seznamu pa so zapisane v obliki koordinat. Posledično je potrebno naslov najljubše poti pretvoriti v par koordinat, pri čemer se uporablja proces geokodiranja. Po pretvorbi vse točke dodamo v seznam, ki ga posredujemo opravilu ob njegovem zagonu.

V opravilu za iskanje poti se preko razreda za upravljanje s potmi nastavi podatke in nastavitvene parametre, ki se pretvorijo v zahtevek HTTP GET. V aplikaciji je za iskanje poti uporabljen ponudnik MapQuest. Del MapQuest API-ja omogoča iskanje poti, tukaj v zahtevek HTTP vključimo različne parametre in podatke na podlagi katerih se potem izvede iskanje. Za dostop do MapQuest API-ja potrebujemo t.i. API ključ, ki služi za avtentikacijo. Posamezen ključ mesečno omejuje število zahtevkov na uporabnika. V naši aplikaciji v zahtevku nastavimo začetno in končno točko poti, format vrnjenega rezultata, ki je XML ter tip poti, ki je v našem primeru pot za pešce. V zahtevku je nastavljen tudi parameter *narrativeType*, ki določa v kakšni obliki bodo vrnjena navigacijska navodila. Trenutno je na voljo samo besedilni tip, ki je tudi privzeto nastavljen.

Ko je zahtevek pripravljen se preveri ali ima uporabnik dostop do omrežja. V primeru, da ga ima, se izvede klic API-ja z zahtevkom HTTP GET.

Primer zahtevka HTTP GET za OpenMapQuest API:

```
http://open.mapquestapi.com/guidance/v1/route?key=
MYfJLyZ1gpij3SUZfG7JQd6hQA4nr0c1&from=46.0489536,14.4744304&to=
46.050017,14.46903&outFormat=xml&shapeFormat=cmp&narrativeType=
text&unit=k&fishbone=false&routeType=pedestrian
```

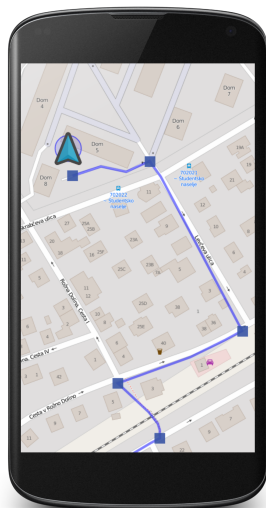
Gre za zahtevek, katerega rezultat je v obliki XML. Tip poti je nastavljen na pešce, enota so kilometri. V zahtevku se opazi tudi primer ključa API.

Izsek iz vrnjenega XML dokumenta za zgornji zahtevek:

```
<node>
  <maneuverType>7</maneuverType>
  <nodeType>0</nodeType>
  <infoCollection>
    <info>Turn right on Jamnikarjeva ulica</info>
    <info>Turn right onto Jamnikarjeva ulica.</info>
  </infoCollection>
  <linkIds>
    <linkId>2</linkId>
  </linkIds>
</node>
```

Vrnjen izsek predstavlja eno izmed vozlišč na seznamu vseh, ki nam jih za določeno pot vrne API. Posamezno vozlišče nam pri navigaciji predstavlja eno izmed opornih točk na poti. Vrnjena vozlišča lahko vsebujejo podatek o tipu manevra, ki nam pove kakšen manever mora uporabnik v tem vozlišču narediti. Poleg tega lahko vsebuje seznam informacij, kjer nam prva informacija v seznamu predstavlja besedilo, ki je pripravljeno za branje t.i. »TTS ready« besedilo, sledeče informacije pa so bolj primerne za prikaz. Iz seznama vrnjenih vozlišč se določeni podatki uporabljajo pri navigaciji uporabnika po poti.

Po prejetju rezultata se za posamezno vozlišče izvede pretvorba navodil v slovenski jezik. Tukaj gre za zamenjavo besed in tvorjenje stavkov, ki jih kasneje združujemo in uporabljamo za navigacijo uporabnika po poti. »TTS ready« navodila, ki jih dobimo iz vozlišč, so sestavljena iz ponavljajočih se besed in besednih zvez, zato lahko z zamenjavo le-teh navodila smiselno



Slika 3.5: Izris poti na zemljevidu v mobilni aplikaciji.

prevedemo v slovenščino.

V naslednjem koraku se pridobljena pot in posamezna vozlišča izrišejo. Za izris poti se uporabi lomljena črta, za izris posameznih vozlišč oz. opornih točk pa se uporabijo oznake. Del izrisane poti na zemljevidu vidimo na sliki 3.5, kjer je z modro črto označena pot, oporne točke pa so predstavljene z modrimi kvadrati.

Po obdelavi prejetih podatkov se ustvari začetno navodilo, to uporabniku preberemo in zapišemo v gumb hitrih akcij. S pritiskom na ta gumb lahko nato uporabnik poljubno proži branje navodil. Pri nastavljanju navodil se preverja katero vozlišče je uporabnik že obiskal in se na podlagi tega pripravi naslednja. Za pripravo navodil se uporablja pridobljene podatke iz seznama vozlišč, poleg tega pa se uporablja tudi podatek o razdalji uporabnika do naslednje točke na poti. Tako se v nekem trenutku vzame razdaljo od uporabnika do naslednje točke in se k temu doda navodilo za manever, ki ga mora uporabnik narediti v naslednji točki. Ko je uporabnik dovolj blizu naslednje točke, se mu manever ponovno sporoči. Ob prehodu čez pa se pripravi

naslednje navodilo, ki se uporabniku prav tako zvočno predstavi in zapiše v gumb.

Na koncu, po izvedbi vseh delov, se zemljevidu nastavi stanje razveljavitve, kar pomeni, da se bo zemljevid ponovno izrisal in s tem izrisal vse spremembe in na novo dodane grafične elemente.

3.1.3 Trenutna lokacija in bližnje avtobusne postaje

Poleg zadnje akcije na seznamu hitrega menija, ki nam prebere trenutno pot, imamo še dve akciji: trenutna lokacija in bližnje avtobusne postaje. Akcija trenutna lokacija služi temu, da lahko uporabnik med sprehajanjem ali med navigacijo v vsakem trenutku preveri svojo lokacijo. S tem uporabniku omogočamo, da si s pomočjo informacije o trenutni lokaciji oz. ulici na kateri se nahaja, pomaga pri orientaciji. Podatek o lokaciji uporabnika, ki ga pridobimo iz naprave, je zapisan v koordinatah zemljepisne širine in dolžine. Ta podatek je potrebno zato pretvoriti v uporabniku razumljivo informacijo in mu jo sporočiti. Za pretvorbo para koordinat v naslov ulice se uporablja proces vzvratnega geokodiranja.

Ker lahko ta proces traja nekaj časa, je smiselno, da se izvaja ločeno od glavne niti aplikacije. Za ta namen je bila implementirana *IntentService* storitev. Storitvi ob zagonu posredujemo lokacijo uporabnika zapisano v koordinatah, pred tem pa nastavimo poslušalca, ki bo skrbel za sprejem rezultata. Storitev za pretvorbo uporablja razred *Geocoder*, ki je del *android.location* knjižnice. Proces vzvratnega geokodiranja nam lahko vrne več rezultatov ali pa nobenega. V primeru da je vrnjenih rezultatov več, se uporabi in zvočno sporoči prvi, v primeru da rezultata ni, pa se to prav tako sporoči uporabniku.

Zaradi potrebe po obeh oblikah zapisa lokacije, se v aplikaciji procesa geokodiranja in vzvratno geokodiranja uporabljata na večih mestih.

Akcija bližnje avtobusne postaje služi iskanju interesnih točk, v našem primeru bližnjih avtobusnih postaj. Za to specifično akcijo smo se odločili na podlagi naših ciljnih uporabnikov, saj jim ta predstavlja najbolj uporabno točko interesa. Do akcije uporabnik dostopa preko hitrega menija. Za iskanje

točk interesa je bil uporabljen Overpass API, ki nam omogoča poizvedovanje po OSM podatkih. Do njega lahko dostopamo preko HTTP zahtevkov, v katere z uporabo jezika XML ali Overpass QL zapišemo naše iskalne kriterije. V našem primeru so bili uporabljeni zahtevki HTTP GET, katerih iskalni kriteriji so bili zapisani v jeziku Overpass QL.

Klic do Overpass API-ja se izvaja v posebnem razredu, ki se ob klicu izvaja asinhrono izven glavne niti. S tem preprečimo blokiranje glavne niti, kjer teče ostali del aplikacije. Uporabnik lahko tako med tem, ko se izvaja iskanje bližnjih točk interesa, nemoteno uporablja aplikacijo. V razredu za dostop do Overpass API-ja najprej zgeneriramo zahtevek, kateremu nastavimo potrebne parametre. Najprej nastavimo oznako po kateri bomo podatke filtrirali, ta je sestavljena iz para ključa in vrednosti. V našem primeru je ključ »highway«, vrednost pa »bus_stop«. Ta par nam iz OSM podatkov vrne točke na zemljevidu, kjer potniki lahko čakajo na avtobus. Drugi parameter, ki ga nastavimo je prostorski obseg (angl. bounding box), ta določa obseg iskanja. V našem primeru iz trenutne lokacije uporabnika izračunamo nek smiselno velik obseg poizvedovanja, ki zajema oddaljenost približno 500 metrov.

Primer enostavnega zahtevka HTTP GET zapisanega v jeziku Overpass QL:

```
http://overpass-api.de/api/interpreter?data=[out:json]
;node["highway"]="bus_stop"(46.044591999999994,14.
481741999999999,46.054592,14.491741999999999);out;
```

V zahtevku je najprej nastavljen format vrnjenega rezultata, ki je JSON. Nato je nastavljena oznaka za iskanje avtobusnih postaj, ključ »highway«, vrednost »bus_stop«. Sledi še parameter za prostorski obseg.

Izsek iz vrnjenega rezultata:

```
{
  "type": "node",
  "id": 1641047464,
  "lat": 46.0457839,
  "lon": 14.4900137,
  "tags": {
    "highway": "bus_stop",
    "name": "602032 ~ Hajdrihova"
  }
}
```

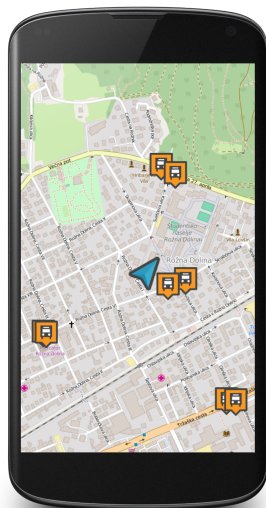
V izseku je vidno eno izmed vozlišč na vrnjenem seznamu in predstavlja eno avtobusno postajo. Iz vozlišča lahko razberemo koordinati oz. lokacijo postaje ter oznake, v katerih je zapisano ime avtobusne postaje.

Vrnjen odgovor v obliki JSON razčlenimo in iz njega preberemo vse bližnje avtobusne postaje. Za posamezno postajo nato izračunamo oddaljenost do uporabnika. V naslednjem koraku na zemljevidu izrišemo točke (Slika 3.6), prikažemo seznam najdenih postaj in o tem obvestimo uporabnika. Postaje na seznamu vsebujejo tudi podatek o oddaljenosti, po katerem so naraščajoče urejene. S tem uporabniku omogočimo hiter dostop do najbližje avtobusne postaje, saj je ta na seznamu na prvem mestu. Uporabnik z dolgim pritiskom na posamezno pot proži branje naslova in oddaljenosti, z enojnim pa proži iskanje poti in zagon navigacije. Tako lahko iz seznama izbere željeno postajo, za katero se na podlagi njegove trenutne lokacije najde pot, po kateri ga nato usmerjamo. Uporabnik se lahko med drugim brez izbire postaje, s pritiskom nazaj, vrne na zemljevid.

3.2 Prilagoditev aplikacije

Pri prilagoditvi aplikacije Spremljevalec je bil poudarek predvsem na govornem sporočanju informacij ter postavitvi, barvi in velikosti elementov, implementirano pa je bilo tudi zaznavanje in uporaba kretenj.

Celotna aplikacija deluje v pogledu portreta in onemogoča rotiranje prika-



Slika 3.6: Izris avtobusnih postaj na zemljevidu v mobilni aplikaciji.

zov aplikacije. S tem uporabniku olajšamo izkušnjo in omogočimo, da lahko akcije pričakuje vedno na istem mestu in se njihove pozicije priuči.

3.2.1 Glasovno sporočanje informacij

Za sintezo govora se v Android aplikacijah uporabljajo t.i. TTS aplikacije. Pri izdelavi so bile uporabljene privzete aplikacije za pretvorbe besedila v govor in brezplačna različica aplikacije eBralec. Aplikacija eBralec za razliko od privzetih, ponuja izgovorjavo besedila v slovenskem jeziku. Kljub temu, da obstaja slovenski sintetizator zvoka, nekateri uporabniki uporabljajo na svojih napravah angleški sintetizator zvoka in so nanj navajeni. V naši aplikaciji se ta nastavi v primeru, da naprava nima sintetizatorja s slovenskim jezikom.

Ker so vsa besedila aplikacije v slovenskem jeziku, za sintezo govora pa se uporablja tudi angleški jezik, prihaja do težav pri izgovorjavi določenih besed. Težave pri izgovorjavi sicer ne predstavljajo prevelikih omejitev, saj

so uporabniki na to vajeni, vendar so določene besede včasih vseeno nerazumljive. Predvsem se pojavljajo težave pri besedah, ki vsebujejo šumnike, saj angleščina teh črk ne podpira. Izgovorjavo lahko s tem, da določene dele besed zamenjamo z deli, ki zvenijo boljše, nekoliko izboljšamo oz. popravimo.

Implementirana je bila enostavna metoda, kjer se določene črke v besedilu zamenjajo s črkami, ki v angleščini zvenijo podobno kot v slovenščini. Primer take pretvorbe je, da črko »č« zamenjamo s črkama »ch«, kar v angleščini zveni približno kot »č«. Kakovost izgovorjave je odvisna tudi od aplikacije TTS, tako za določene pretvorbe pri nekaterih ponudnikih dobimo boljše, pri nekaterih pa slabše rezultate. Zamenjave črk poleg tega ne proizvajajo enako dobrih rezultatov, saj določene besede s popravki zvenijo bolje, druge pa zvenijo še bolj popačeno. Glede na to, da se v aplikaciji večinoma uporabljajo enake besede in besedne zveze, je takšna enostavna metoda uporabna in omogoča boljše razumevanje izgovorjenih besed.

Aplikacija Spremljevalec se za zvočno sporočanje poslužuje aplikacije za pretvorbo besedila v govor, katero ima uporabnik nastavljeno kot privzeto. Za inicializacijo in uporabo pretvorbe besedila v govor je bila narejena storitev, ki se zažene ob prvi zahtevi po izgovorjavi. Storitev nato deluje v ozadju, dokler uporabnik ne zapusti aplikacije. Ob zagonu storitve se nastavi jezik, ki je v našem primeru privzeto slovenščina, v primeru da ta ne obstaja, pa se nastavi angleščina. Ko uporabnik izvede akcijo, kateri naj bi sledile zvočne informacije, se izvede klic storitve, kjer se besedilo sintetizira v govor.

V naši aplikaciji se sinteza govora uporablja skozi celotno aplikacijo. Gumbi v meniju in na seznamih aplikacije uporabniku ob dolgem pritisku preberejo vsebino. S tem uporabniku omogočimo, da lahko preveri katero akcijo želi izvesti, če razporeditve gumbov še ne pozna. Prav tako se ob dolgem pritisku na posamezno najljubšo pot uporabniku prebere naslov poti in mu omogoča izbiro ustrezne.

Zvočne informacije se uporabniku posredujejo tudi med samo navigacijo ob ustreznih dogodkih, npr. ob začetku, koncu ter med prehodom skozi oporne točke vzdolž poti. Uporabniku se zvočno posredujejo tudi informacije,

ki jih proži iz okna za hitre akcije. Ob določenih akcijah, ki za izvedbo potrebujejo dlje časa, se uporabniku to tudi glasovno sporoči in ga s tem obvesti o trenutnem stanju aplikacije.

3.2.2 Postavitev, velikost in barva elementov

Implementirane so bile tudi prilagoditve glede postavitve, velikosti in barve elementov. Gumbi in ostali elementi v aplikaciji so veliki in skupaj zasedajo celotno širino in višino ekrana. Glavni meni (Slika 3.1) je razdeljen na 3 dele oz. gumbe, med katerimi ni razmaka in skupaj zasedajo celoten ekran. Enak princip je bil uporabljen v meniju za hitre akcije. Seznam najljubših poti je razdeljen na četrtine (Slika 3.2), kjer vsak del predstavlja eno najljubšo pot. Med gumbi s potmi ni razmaka in skupaj zasedajo celoten ekran.

Vsa besedila v aplikaciji uporabljajo dobro berljivo in veliko pisavo. Pisava uporabljena v aplikaciji je Open Sans Bold, ki ustreza navodilom za izdelavo prilagojenih aplikacij. Pisava je pokončna, krepka, neserifna in je med drugim prilagojena za prikaz na vmesnikih mobilnih naprav. Pisava Tiresias, ki je bila izdelana posebej za slabovidne, zaradi pomanjkanja podpore vseh črk v naši aplikaciji ni bila uporabljena.

Poleg velikosti in pisave so prilagojene tudi barve elementov in besedila. Za slabovidne je dobro, da so elementi v aplikaciji različno obarvani in da se barve med seboj čim bolj razlikujejo. Prav tako je pomembna barva besedila na elementih, ki se mora čim bolj razlikovati od barve ozadja elementa. Za elemente so bile uporabljene štiri barve: oranžna, modra, bela in črna. Za besedila sta bili uporabljeni črna in bela barva, kjer se bela barva uporablja na modrem in črnem, črna pa na oranžnem in belem ozadju. Zgoraj naštetе vizualne prilagoditve lahko vidimo na vseh vključenih slikah aplikacije.

3.2.3 Kretnje

Zaradi dostopnosti in uporabnosti, se med prosto hojo ali navigacijo do menija s hitrimi akcijami dostopa s tem, da napravo stresemo. Hiter meni se po

tem, ko uporabnik izvede eno izmed akcij, zapre. Priklic menija s tresenjem omogoča, da lahko zemljevid zaseda celoten ekran, saj za dostop do menija ne potrebujemo nobenih gumbov. Slepim in slabovidnim je prav tako med hojo lažje izvesti kretnjo, kot pa iskanje in pritisk gumba na ekranu.

Zaznavanje tresenja je bilo v aplikaciji implementirano in prilagojeno tako, da se hitri meni med hojo in normalno uporabo naprave ne sproži, kljub temu pa za proženje ni potrebno močno tresenje.

Poglavje 4

Spletna aplikacija

Poleg mobilne Android aplikacije je bila izdelana tudi spletna aplikacija, ki služi kot vmesnik za dostop do najljubših poti in upravljanje z njimi. Razvoj aplikacije je potekal v Microsoft .NET ogrodju, v tehnologiji ASP.NET ter z uporabo okolja Visual Studio. Prikazi so bili izdelani v jeziku HTML5 in oblikovani s CSS knjižnico Bootstrap, zaledje aplikacije pa je bilo napisano v programskem jeziku C#.

Za hranjenje podatkov o uporabnikih in njihovih najljubših poteh je bila ustvarjena podatkovna baza Microsoft SQL. Podatkovna baza je enostavna in sestoji iz dveh tabel. V prvi tabeli hranimo uporabniške podatke, v drugi pa hranimo najljubše poti, ki so preko identifikacijske številke uporabnika povezane z uporabniškim računom.

Aplikacija in podatkovna baza sta bili postavljeni na strežniku AppHarbor, ki nudi brezplačno gostovanje za .NET aplikacije. Poleg drugih, AppHarbor nudi tudi gostovanje podatkovnih baz Microsoft SQL. AppHarbor omogoča, da z uporabo Git-a potisnemo kodo na strežnik, kjer ta poskrbi za gradnjo aplikacije. Če je gradnja uspešna, lahko aplikacijo vidimo na spletu, stanje gradnje pa lahko ves čas spremljamo preko nadzorne plošče.

Na strežniku se postavijo in zaženejo tudi storitve (angl. services), ki jih potrebujemo za komunikacijo med mobilno in spletno aplikacijo.

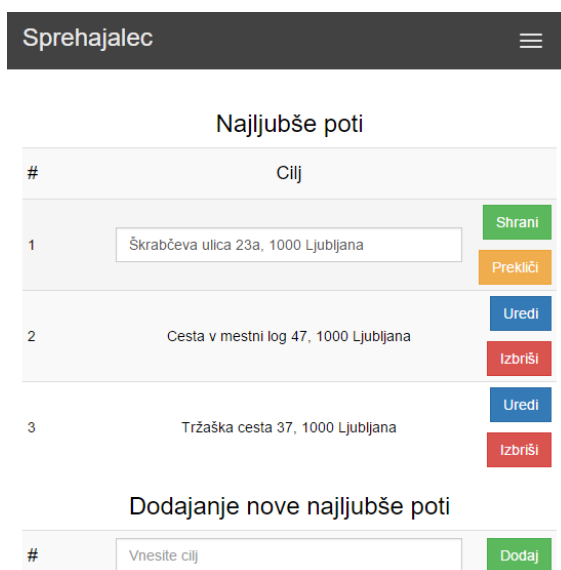
Spletna aplikacija ima dva glavna namena. Prvi namen je uporabniku

omogočiti dostop do svojih najljubših poti in njihovo urejanje neodvisno od mobilne aplikacije. To pomeni, da lahko uporabniku nekdo drug vnaša in ureja najljubše poti brez neposrednega dostopa do mobilne aplikacije in mu s tem olajša delo. Uporabniku lahko prav tako doda ali ureja najljubše poti, medtem ko je ta že na poti.

Uporabnik za dostop do spletne aplikacije potrebuje številko pin, ki se skupaj z uporabniškim računom ustvari ob prvem zagonu mobilne aplikacije. Uporabnik jo lahko nato prebere iz mobilne aplikacije v prikazu orodja in pomoč. Vhodno stran spletne aplikacije predstavlja stran za prijavo, kjer uporabnik za nadaljevanje vnese le številko pin. Po uspešni prijavi je uporabnik preusmerjen na glavno stran oz. nadzorno ploščo, kjer lahko upravlja z najljubšimi potmi. Stran je sestavljena iz seznama najljubših poti in polja za dodajanje novih najljubših poti, kar je prikazano na sliki 4.1. S klikom na gumb »Izbriši« uporabnik izbrano pot odstrani s seznama, pri čemer mora akcijo še dodatno potrditi. S klikom na gumb »Uredi« se prikaže vnosno polje, v katerem lahko uporabnik pot ureja in spremembe nato bodisi shrani, bodisi zavrže. V polju za dodajanje uporabnik vnese novo najljubšo pot in jo s klikom na gumb »Dodaj« doda na seznam. Pri vnosu novih najljubših poti je potrebno paziti, da poleg naslova ulice dopišemo tudi kraj oz. poštno številko, saj lahko v nasprotnem primeru pri geokodiranju pride do napačnih rezultatov.

Drugi namen aplikacije je, da služi kot vmesnik med podatki uporabnika in mobilno aplikacijo. Mobilna aplikacija tako preko storitev spletne aplikacije dostopa do podatkov in funkcij za delo s podatki. Mobilna aplikacija do storitev dostopa z zahtevki HTTP, v katere zapiše parametre, spletna aplikacija pa poskrbi za izvedbo zahtev in vrne rezultat.

Eden izmed pomembnih klicev spletne storitve se zgodi ob prvem zagonu mobilne aplikacije, kjer se pošlje zahtevki HTTP za kreiranje novega računa uporabnika. Pri kreiranju gre za enostavno metodo, kjer se za uporabnika ustvari unikatna 6 mestna številka pin sestavljena iz velikih črk in števil, ki se shrani v podatkovno bazo in se v obliki JSON kot odgovor vrne mobilni



Slika 4.1: Stran z najljubšimi potmi v spletni aplikaciji

aplikaciji. Ta odgovor razčleni in pridobljene podatke lokalno shrani. Mobilna aplikacija je tako povezana z uporabniškim računom in lahko dostopa do ostalih podatkov.

Za pridobitev najljubših poti se iz mobilne aplikacije pošlje zahtevek HTTP, v katerem se posreduje pin uporabnika. Spletna aplikacija zahtevek razčleni in glede na številko pin iz podatkovne baze ustrezno pridobi uporabnikove najljubše poti. Seznam najljubših poti se zapakira v format JSON in se kot odgovor vrne mobilni aplikaciji, ta medtem čaka odgovor in ob prejemu poskrbi za razčlenitev in prikaz najljubših poti.

Poglavje 5

Sklepne ugotovitve

V sklopu diplomske naloge je bila kot glavna rešitev izdelana prilagojena mobilna aplikacija za pomoč pri orientaciji slepih in slabovidnih. Vsebuje funkcionalnosti kot so vodenje po najljubših poteh, prosto sprehajanje ter iskanje in navigacija do bližnjih interesnih točk. Pri implementaciji prilagoditev so bile upoštevane obstoječe smernice za izdelavo tovrstnih aplikacij ter omejitve, zahteve in predlogi slepih in slabovidnih. Zaradi potrebe po slovenski verziji aplikacije, je za zapis besedil in glasovno sporočanje informacij kot privzeto uporabljen slovenski jezik. Poleg mobilne aplikacije je bila izdelana tudi spletna aplikacija, ki omogoča dostop do uporabnikovih podatkov in njihovo urejanje brez neposrednega dostopa do mobilne naprave. S tem omogočimo, da lahko uporabniku pri dodajanju in urejanju najljubših poti pomaga tudi nekdo drug. Izvedeno je bilo tudi raziskovanje na področju že obstoječih podobnih rešitev, prilagoditve aplikacij in same orientacije slepih in slabovidnih.

Aplikacija lahko slepim in slabovidnim služi kot pripomoček pri sprehajanju na prostem in v določenih primerih nadomesti spremljevalca. Poleg sprehajanja po znanih poteh odpira tudi možnost sprehajanja po novih poteh in možnost prostega sprehajanja. Aplikacija se lahko uporabi tudi za pridobitev informacij o bližnjih avtobusnih postajah in navigacijo do njih. Aplikacija je zaradi razširjenosti mobilnih naprav in njihove uporabe med

slepimi in slabovidnimi enostavno dostopna in uporabna. Prav tako je za slepe in slabovidne dostopna in uporabna zaradi vseh implementiranih prilagoditev.

Izdelana rešitev predstavlja osnovno aplikacijo s ključnimi funkcionalnostmi, pri čemer pa ostaja veliko prostora za nadgradnje in implementacijo dodatnih funkcionalnosti. Nadgradili bi lahko hranjenje najljubših poti, kjer bi poleg končne točke omogočali tudi vnos in hranjenje vmesnih točk in s tem omogočili natančnejšo določitev poti. Dodali bi lahko možnost izbiranja vrste in količine sporočenih informacij, s čimer bi uporabniku omogočili, da lahko aplikacijo prilagodi lastnim potrebam in željam. Vključili bi lahko tudi dodatne kretnje in s tem izboljšali dostopnost in prilagojenost aplikacije. Največ nadgradenj pa bi lahko naredili na področju orientacije in navigacije, kjer bi lahko implementirali vrsto algoritmov za boljše določanje lokacije in natančnejšo navigacijo po poti. Prav tako ostaja odprto vprašanje, kako slepim in slabovidnim najbolje opišemo okolico in kakšne podatke uporabimo za čimbolj optimalno orientacijo. V aplikacijo bi lahko kot dodatne funkcionalnosti vključili tudi vnos podatkov preko govora in razširili nabor interesnih točk.

Ker rešitev predstavlja prvo tovrstno aplikacijo v slovenskem jeziku, lahko služi kot osnova za nadaljnji razvoj. Med drugim lahko služi tudi kot pomoč pri implementaciji drugih prilagojenih mobilnih aplikacij za slepe in slabovidne. Predvsem pa predstavlja enostaven pripomoček, ki lahko slepim in slabovidnim olajša vsakodnevno sprehajanje na prostem.

Literatura

- [1] Android Activity. Dosegljivo: <https://developer.android.com/reference/android/app/Activity.html>, 2016. [Dostopano: 15. 7. 2016].
- [2] eBralec. Dosegljivo: <http://ebralec.si/>, 2016. [Dostopano: 12. 8. 2016].
- [3] eSpeak for Android. Dosegljivo: <http://reecedunn.co.uk/espeak-for-android>, 2014. [Dostopano: 9. 8. 2016].
- [4] Google Maps APIs Pricing and Plans. Dosegljivo: <https://developers.google.com/maps/pricing-and-plans/>, 2016. [Dostopano: 9. 8. 2016].
- [5] GraphHopper. Dosegljivo: <https://graphhopper.com/>, 2016. [Dostopano: 9. 8. 2016].
- [6] Java. Dosegljivo: <https://www.java.com/en/about/>, 2015. [Dostopano: 14. 8. 2015].
- [7] B. Kurniawan. *Java for Android, Second Edition*.: Brainy Software Incorporated, 2015.
- [8] MapQuest Developer. Dosegljivo: <https://developer.mapquest.com/>, 2016. [Dostopano: 8. 8. 2016].
- [9] MapQuest Pricing and Plans. Dosegljivo: <https://developer.mapquest.com/plans>, 2016. [Dostopano: 9. 8. 2016].

-
- [10] Mobile Accessibility: How WCAG 2.0 and Other W3C/WAI Guidelines Apply to Mobile. Dosegljivo: <https://www.w3.org/TR/mobile-accessibility-mapping/>, 2015. [Dostopano: 3. 8. 2016].
- [11] OpenStreetMap. Dosegljivo: <https://www.openstreetmap.org/about>, 2016. [Dostopano: 5. 8. 2016].
- [12] OSMBonusPack. Dosegljivo: <https://github.com/MKergall/osmbonuspack>, 2015. [Dostopano: 18. 8. 2015].
- [13] OsmDroid. Dosegljivo: <https://github.com/osmdroid/osmdroid>, 2015. [Dostopano: 18. 8. 2015].
- [14] OpenStreetMap Stats. Dosegljivo: <http://wiki.openstreetmap.org/wiki/Stats>, 2016. [Dostopano: 5. 8. 2016].
- [15] Overpass API. Dosegljivo: http://wiki.openstreetmap.org/wiki/Overpass_API, 2016. [Dostopano: 18. 8. 2016].
- [16] Overpass API/Overpass QL. Dosegljivo: http://wiki.openstreetmap.org/wiki/Overpass_API/Overpass_QL, 2016. [Dostopano: 18. 8. 2016].
- [17] Nektarios Paisios. *Mobile accessibility tools for the visually impaired*. PhD thesis, Citeseer, 2012.
- [18] Donatella Pascolini and Silvio Paolo Mariotti. Global estimates of visual impairment: 2010. *British Journal of Ophthalmology*, 2011.
- [19] ViaOpta Nav. Dosegljivo: <https://play.google.com/store/apps/details?id=com.novartis.blind>, 2015. [Dostopano: 4. 7. 2016].
- [20] W3C. Dosegljivo: <https://www.w3.org/>, 2016. [Dostopano: 3. 8. 2016].
- [21] WalkyTalky. Dosegljivo: <https://play.google.com/store/apps/details?id=com.googlecode.eyesfree.walkytalky>, 2011. [Dostopano: 1. 7. 2016].

-
- [22] Web Content Accessibility Guidelines (WCAG) 2.0. Dosegljivo: <https://www.w3.org/TR/WCAG20/>, 2008. [Dostopano: 3. 8. 2016].
- [23] ZDSSS Okvare vida. Dosegljivo: <http://www.zveza-slepih.si/okvare-vida/>, 2015. [Dostopano: 2. 8. 2016].
- [24] ZSSM O zavodu. Dosegljivo: <http://582.gvs.arnes.si/wordpress/o-zavodu/>, 2012. [Dostopano: 2. 8. 2016].