UNIVERZA V LJUBLJANI
Fakulteta za elektrotehniko

Aleš Pevc

Mobilna aplikacija za pomoč pri reševanju ponesrečencev v gorah

DIPLOMSKO DELO
VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM
PRVE STOPNJE TELEKOMUNIKACIJE

MENTOR: doc. dr. Grega Jakus

Ljubljana 2016

Zahvala

Svoji ženi Andrejki, sinu Nalu, staršem, še posebej mami, in vsem ostalim, ki ste mi med študijem in pripravljanjem diplomske naloge stali ob strani ter me spodbujali, se res najlepše zahvaljujem.
Zahvalil bi rad tudi mentorju, doc. dr. Gregu Jakusu, za vse nasvete, potrpežljivost ter vso strokovno podporo.
Še posebej pa bi se rad zahvalil Sergeju Šorliju, svojemu prijatelju in sodelavcu, ki mi je s svojim strokovnim znanjem vedno priskočil na pomoč, kadar koli je bilo to potrebno.

Vsebina

Seznam slik

Seznam uporabljenih kratic in simbolov

2D	angl. A geometric model with two parameters geometrični model z dvema parametroma
3D	angl. A geometric model with three parameters geometrični model s tremi parametri
3D GIS	angl. Three-dimensional geographic information system tridimenzionalni geografski informacijski sistem
ADB	angl. Android Debug Bridge Androidno komunikacijsko orodje
AML	angl. Advanced Mobile Location napredna mobilna lokacija
API	angl. Application Programming Interface aplikacijski programski vmesnik
AVD	angl. Android Virtual Device Androidna virtualna naprava
BeiDou	angl. Chinese satellite-navigation system kitajski satelitsko-navigacijski sistem
IDE	angl. Integrated Development Environment integrirano razvojno orodje
IMEI	angl. International Mobile Station Equipment Identity edinstvena mednarodna številka mobilne naprave
GALILEO	angl. European Satellite-navigation System evropski satelitsko-navigacijski sistem
GLONASS	angl. Russian Global Navigation Satellite System ruski satelitski sistem globalne navigacije
GPS	angl. Global Positioning System globalni sistem določanja položaja
GSM	angl. Global System for Mobile Communications globalni sistem za mobilne komunikacije
GPRS	angl. General Packet Radio Service

mobilna podatkovna storitev

GNSS	angl. Global Navigation Satellite System globalni satelitski navigacijski sistem
NATO	fr. Organisation du Traité de l'Atlantique Nord Organizacija severnoatlantske pogodbe
OHA	angl. Open Handset Alliance združenje podjetij, ki razvijajo Android
PZS	angl. Alpine Association of Slovenia Planinska zveza Slovenije
SDK	angl. Software Development Kit paket razvojnih orodij za Android
SMS	angl. Short Message Service sistem kratkih sporočil
SQL	angl. Structured Query Language strukturiran programski jezik
USB	angl. Universal Serial Bus univerzalno serijsko vodilo
UMTS	angl. Universal Mobile Telecommunications System univerzalni mobilni telekomunikacijski sistem
ZDA	angl. United States of America Združene države Amerike
Wi-Fi	angl. Technology that allows electronic devices to connect to a wireless LAN brezžična tehnologija, ki napravam omogoča povezavo v omrežje WLAN
WGS-84	angl. – World Geodetic System svetovni geodetski sistem

Povzetek

V diplomskem delu smo obravnavali izdelavo mobilne aplikacije na platformi Android. Namen mobilne aplikacije za pomoč pri reševanju ponesrečencev v gorah je omogočiti lažje poročanje o natančnem položaju ponesrečencev in posredovanje le-tega v Center za obveščanje. Zaradi nepoznavanja točnega položaja s strani ponesrečenca oziroma nenatančnosti položaja, pridobljenega s strani mobilnega operaterja, zahtevajo reševalne akcije večje število vključenih reševalcev, kot bi bilo to potrebno, posledično pa tudi izgubo dragocenega časa.

Osnovna funkcionalnost aplikacije je pošiljanje natančnega položaja uporabnika prek SMS-sporočil, ki deluje tudi v razmerah, ko je podatkovni prenos otežen. V določenih predelih planinskega sveta, ni pokritosti s signalom z baznih postaj, zato smo aplikaciji dodali možnost sledenja v ozadju in samodejno pošiljanje izbranega števila SMS-sporočil o položaju uporabnika v določenih časovnih intervalih (na primer vsakih 15 minut). Na takšen način lahko reševalci v primeru nesreče in morebitne izgube signala rekonstruirajo opravljeno pot ponesrečenca in tako zmanjšajo iskalno območje. Razvita aplikacija ponuja tudi navodila za ravnanje ob nesrečah, hkrati pa smo zaradi preventive z namenom zmanjšanja števila nesreč dodali še ažurne informacije in opozorila o razmerah v gorah.

V uvodnem poglavju diplomske naloge smo raziskali statistične podatke ter utemeljili potrebnost aplikacije. V drugem poglavju smo preučili operacijski sistem Android, njegovo arhitekturo in razvojno okolje Android Studio, v katerem smo aplikacijo tudi razvili. V tretjem poglavju smo preučili tehnologije za določanje položaja mobilnega uporabnika, temelječe na sistemu GPS, ter pomene in možnosti geolokacije. V četrtem poglavju smo se posvetili načrtovanju in izvedbi aplikacije. Opisali smo arhitekturo aplikacije, zasnovo uporabniškega vmesnika ter potek testiranja aplikacije v planinskem svetu. V zadnjem poglavju diplomske naloge pa smo opisali tudi mogoče dodatne funkcionalnosti in nakazali možnosti za nadaljnji razvoj te mobilne aplikacije.

Ključne besede

Android, geolokacija, reševanje v gorah, klic v sili, sledenje

Abstract

In this thesis, we analyze the process of developing a mobile application on the Android platform. The purpose of the application that helps rescue injured persons from the mountains is to facilitate reporting casualty's exact location and transmit it to the Information Centre. Since casualties do not usually know their exact location or because locations obtained by the mobile operator are not precise enough rescue operations require more rescuers than would be necessary and also valuable time is lost.

The basic functionality of the application is sending user's exact location via text message which works well even in circumstances when data transfer is difficult. In certain areas of the alpine world there is no signal coverage, therefore the application was added an option for tracking in the background and automatically sending a selected number of text messages containing user's location in specified time intervals (e.g. every 15 minutes). This way, in case of an accident or loss of signal, the rescuers can reconstruct user's route and reduce the search area. The application also provides instructions on how to handle accidents and, as a preventive measure to reduce the number of accidents, we have added up-to-date information and warnings concerning conditions in the mountain area.

In the introductory chapter of the thesis, we research statistic data and justify the necessity of the application. The second chapter explores the Android operating system and its architecture as well as Android Studio development environment in which the application is developed. In the third chapter, we first examine the technology for determining mobile user's location based on the GPS system and then the importance and possibilities of geolocation. The fourth chapter is dedicated to planning and creating the application. We describe its architecture, user interface design and its testing in the mountains. In the last chapter of the thesis, we describe possible additional functionalities and indicate options for further development of the application.

Keywords

Android, geolocation, mountain rescue, emergency call, tracking

1. Uvod

1.1. Opis problema

Priljubljenost izletov v planine in gore iz leta v leto narašča, s tem pa seveda tudi potencialna možnost za nezgode oziroma nesreče. Po podatkih statističnega urada se vsako leto v Sloveniji odpravi na obisk planin in gora okrog 1.400.000 obiskovalcev [1]. Število nesreč, zaradi katerih intervenirajo reševalci, se na letni ravni giblje okoli 350, število ponesrečencev pa okoli števila 400. Od tega gre v okoli 25 primerih za iskanje pogrešanih oseb [2]. Zgovoren je tudi podatek, da so reševalci za reševanje v slovenskih gorah porabili letno[1] prek 10.000 reševalnih ur, kar znaša v povprečju skoraj 39 reševalnih ur in 9,2 reševalcev na reševalno akcijo [2]. Največjo težavo pri načrtovanju reševalne akcije, poleg organizacije in vremenskih razmer, predstavlja pridobivanje čim bolj točnega položaja ponesrečenca oziroma izrednega dogodka. Ponesrečenci svojega natančnega položaja večinoma ne poznajo, lahko so v šoku, njihov opis položaja pa reševalce lahko celo zavede, saj opis lahko ustreza mnogim podobnim točkam v okolici. Planinci večinoma tudi ne poznajo dobro smeri, po kateri so šli v planine, in težko ocenijo razdaljo, ki so jo prehodili, zato morajo reševalne službe v iskalno akcijo vključiti večje število reševalcev, tudi do 30 in več, ki morajo nato pregledati različne planinske poti.

Reševalci lahko sicer od mobilnega operaterja pridobijo položaj ponesrečenca na podlagi klica na pomoč, a je položaj, pridobljen na tak način, zelo pogosto premalo natančen. Največje težave s pridobivanjem točnega položaja s strani operaterjev se pojavljajo prav v gorskem svetu. Kljub temu da so bazne postaje na tem področju postavljene zelo visoko in zato lahko pokrivajo veliko območje, se zaradi razgibanega reliefa terena pojavljajo področja nepokritosti s signalom.

Da so nenatančni položaji ponesrečencev velik problem, kažejo tudi ocene, da znašajo stroški reševanja zaradi nenatančnih položajev na ravni EU skoraj 4 milijarde evrov [3].

Praktično na dlani se zdi rešitev problema ob razširjenosti pametnih telefonov, ki dosega delež med 60 in 70 % [4]. Z uporabo enega izmed satelitskih navigacijskih sistemov (kot je npr.: GPS – Globalni sistem določanja položaja) omogočajo pametni telefoni sprejemanje natančnega položaja uporabnika, zato jih lahko uporabljamo tudi kot vrsto varnostnega

[1] podatek iz leta 2010

pripomočka, vendar pa sama možnost sprejema GPS-signala še ne pomeni, da lahko ponesrečenci ob izrednem dogodku hitro in enostavno ugotovijo svoj položaj, saj je ugotavljanje natančnega položaja na pametnih telefonih tudi za naprednejše uporabnike pogosto neizvedljivo brez uporabe namenske aplikacije.

Namenske aplikacije, ki bi ustrezno pokrivale področje reševanja v gorah, v slovenskem prostoru še nimamo. V primeru izrednega dogodka, ko je pomembna vsaka minuta, pa je lahko točen in hiter dostop do položaja ponesrečenca odločilnega pomena. Zato smo se odločili, da bomo v diplomski nalogi obdelali področje reševanja v gorah z uporabo mobilne aplikacije, s poudarkom na glavni funkcionalnosti, to je enostavnem pridobivanju natančnega položaja s strani uporabnika in posredovanje le-tega v Center za obveščanje. Pošiljanje položaja v Center za obveščanje je urejeno preko SMS-sporočil, saj ta način komunikacije deluje tudi v primeru šibkega signala, ko je prenos podatkov prek paketnega prenosa (npr. preko GPRS – General Packet Radio Service ali UMTS – Universal Mobile Telecommunications System) otežen.

Dodatni cilj diplomske naloge je bil razviti tudi vrsto varnostnega sledenja. To vključuje periodično pošiljanje podatkov o trenutnem položaju preko SMS-sporočil v 15-minutnih intervalih na izbrano številko prijateljev oziroma svojcev. Ti ob morebitni daljši neodzivnosti s strani planinca v Center za obveščanje posredujejo sprejete položaje, ki jih tam lahko nato uporabijo za rekonstrukcijo poti in dogodkov. Nemalokrat se zgodi, da planinec oziroma turni smučar zdrsne v globel, v kateri ni mobilnega signala, ali se mu izprazni baterija mobilnega telefona in zato v primeru izrednega dogodka ne more sam obvestiti Centra za obveščanje. Pomembna funkcionalnost mobilne aplikacije so tudi navodila za ravnanje ob nesrečah. Čakanje ponesrečenca na reševalce je zagotovo mučno doživetje, saj je ponesrečenec lahko v stresu, ali pa je morda resno poškodovan. V tistem trenutku zato lahko prav taka navodila ponesrečencu nudijo občutek varnosti in ga pomirijo. Ker želimo z aplikacijo tudi aktivno prispevati k preventivi za zmanjšanje števila nesreč in poškodb v gorah, smo dodali možnost ažurnega prikaza razmer v gorah.

1.2. Obstoječe rešitve

Na trgu že obstajajo profesionalni sistemi in naprave za sledenje in tudi mobilne aplikacije za pridobivanje geolokacije (npr. Current GPS location [5], My GPS Location [6]) ter aplikacije za sledenje (npr. GPS Location Track [7], Family Locator [8], 360 life [9]), a večina pohodnikov in planincev tovrstnih mobilnih aplikacij nima naloženih. Te aplikacije večinoma potrebujejo dostop do prenosa podatkov in so velik porabnik baterije. Poleg tega uporabnikom oziroma ponesrečencem niti ne omogočajo neposredne komunikacije z dispečerskim centrom niti ne enostavnega pošiljanja informacij o natančnem položaju prek SMS-sporočil.

Nekateri sodobni centri za obveščanje (pri nas številka 112) imajo že na voljo storitve za napredno določanje položaja, ki osebju v centru omogoča, da v primeru izrednega dogodka samostojno pridobijo natančnejši položaj uporabnika, vendar storitve centra delujejo le prek omogočene podatkovne povezave in ne omogočajo rekonstrukcije opravljene poti. Prav tako ne ponujajo funkcionalnosti, ki deluje v smeri preventive in je pomembna za boljšo varnost v gorah in nenazadnje: pošiljanje položaja v primeru izrednega dogodka ni omejeno samo na reševalce, saj nam velikokrat lahko priskočijo na pomoč tudi drugi planinci, če le poznajo naš položaj. Aplikacija za pošiljanje položaja pa nam je lahko v pomoč tudi na področjih izven gorskega sveta – v primeru, da smo se izgubili oziroma v primeru nesreče, ko želimo reševalcem poslati točen položaj izrednega dogodka.

Mobilna aplikacija za reševanje v gorah, ki je nastala kot rezultat diplomskega dela, bo planincem in reševalcem s svojo funkcionalnostjo v zelo veliko pomoč. Mobilno aplikacijo smo razvili na platformi Android, za njen nadaljnji razvoj in trženje pa bo skrbelo zagonsko podjetje ABI Team.

V nadaljevanju diplomskega dela smo prestavili operacijski sistem Android, razvojno orodje Android Studio ter opisali postopek izdelave aplikacij. Tretje poglavje opisuje tehnologije za določanje položaja mobilnega terminala. V četrtem poglavju predstavljamo razvoj mobilne aplikacije za reševanje v gorah in njenega uporabniškega vmesnika kot praktični izdelek, narejen v okviru diplomskega dela.

2. Razvoj mobilnih aplikacij za operacijski sistem Android

2.1. Operacijski sistem Android

Android je trenutno najbolj razširjeni operacijski sistem za mobilne naprave. V prvi vrsti je namenjen mobilnim telefonom, vse bolj pa se uveljavlja tudi na drugih, predvsem mobilnih napravah, kot so na primer: tablični računalniki, čitalci elektronski knjig, pametne ročne ure, TV naprave ipd. Ime platforme oziroma operacijskega sistema izhaja iz angleške besede »android«, kar pomeni robot, ki izgleda in se obnaša kot človek.

Android temelji na jedru operacijskega sistema Linux. Razvija ga Google v sodelovanju s podjetji združenja Open Handset Alliance (OHA). Je odprtokoden in povsem brezplačen, njegova prednost pa je tudi, da je zelo dobro dokumentiran, zato omogoča cenejše in lažje razvijanje programov v primerjavi z drugimi operacijskimi sistemi. Znatno prednost tega občutijo tudi uporabniki, saj so programi za ta operacijski sistem večinoma brezplačni. Zaradi enostavnosti, brezplačnosti in široke uporabnosti je Android postal najbolj razširjeni operacijski sistem za mobilne naprave. Trenutno se njegov tržni delež na področju mobilnih naprav giblje okrog 80 % [10].

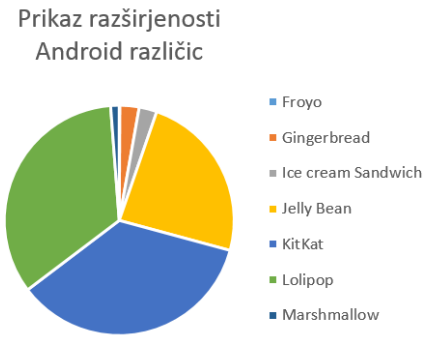
Ena izmed pomembnih prednosti tega operacijskega sistema je tudi ločevanje med strojno in programsko opremo, zaradi česar lahko isto aplikacijo poganjamo na večjem številu različnih naprav. Kljub takšnemu ločevanju omenjena platforma omogoča aplikacijam, da izkoristijo vse strojne zmožnosti naprav, kot so na primer: občutljivi zasloni na dotik, senzorji pospeška, kompas, GPS sprejemnik, kamere, mikrofoni, strojni grafični pospeševalniki itd.

Aplikacije se v okviru operacijskega sistema Android izvajajo nadzorovano, odmaknjeno od jedra platforme, kar preprečuje dostop do tistih funkcionalnosti, ki jih ne potrebujejo za svoje delovanje in bi njihova uporaba ogrozila delovanje ostalih aplikacij. Pri namestitvi aplikacij uporabnik zato dobi nabor dovoljenj za funkcionalnosti, ki jih aplikacija potrebuje za svoje delovanje, kar uporabniku daje možnost, da v primeru potencialno škodljivih aplikacij oziroma če mu katera izmed zahtev ne ustreza, prekine namestitev aplikacije.

2.1.1. Različice platforme

Začetek razvoja platforme Android sega v leto 2003, prva različica 1.0 pa je na trg prišla leta 2009. Vse naslednje različice so dobile imena po slaščicah (Slika 1). Vsaka različica tega operacijskega sistema ima pripadajočo različico aplikacijskih programskih vmesnikov (Application Programming Interface, API), s pomočjo katerih lahko razvijalci izdelajo lastne aplikacije. V času pisanja tega diplomskega dela je aktualna različica operacijskega sistema 6.0.1 Marshmallow[2], njemu pripadajoča različica API pa 23.

Različica OS	Ime različice	Različica API	Razširjenost distribucije
2.2	Froyo	8	0,1%
2.3.3-2.3.7	Gingerbread	10	2,7%
4.0.3 4.0.4	Ice cream Sandwich	15	2,5%
4.1.x 4.2.x 4.3	Jelly Bean	16 17 18	8,8% 11,7% 3,4%
4.4	KitKat	19	35,5%
5.0 5.1	Lolipop	21 22	17,0% 17,1%
6.0	Marshmallow	23	1,2%



Slika 1: Različice operacijskega sistema Android in API različic glede na aktualno razmerje razširjenosti med uporabniki [11]

[2] Maj 2016

Operacijski sistem Android omogoča stalne in enostavne posodobitve z novimi različicami, ki jih uporabnik lahko samostojno namesti, ko so na voljo in ko to omogoči proizvajalec mobilne naprave. Sistem je zasnovan tako, da novejše različice razvojnih orodij in programskih knjižnic omogočajo združljivost s starejšimi.

2.2. Arhitektura operacijskega sistema Android

Tehnično je Android osnovan na rahlo prilagojenem operacijskem sistemu Linux, razvitem za arhitekturi ARM in x86. Arhitekturno je razdeljen na štiri nivoje, ti pa so sestavljeni iz dodatnih programskih komponent, kot je to prikazano na Sliki 2.

Slika 2: Arhitektura nivojev platforme Android [12]

Aplikacije

Najvišjo plast predstavljajo aplikacije, ki so lahko prednameščene, ali jih po potrebi dodaja uporabnik. Androidne naprave imajo navadno prednameščenih veliko aplikacij, kot so: brskalnik, fotoaparat, galerija slik, glasba, telefon, sporočila, seveda pa lahko dodamo tudi nove po lastni izbiri. Aplikacije za svoje delovanje uporabljajo funkcionalnost aplikacijskega ogrodja.

Aplikacijsko ogrodje (ang. application framework)

Na tem nivoju se nahajajo sistemski gradniki, ki so neizogibno potrebni za delovanje aplikacij. Uporaba funkcionalnosti ogrodja je razvijalcu na voljo preko API, ki omogočajo aplikacijam dostop do strojne opreme (GPS sprejemnik, Wi-Fi, Bluetooth …), uporabo že pripravljene funkcionalnosti in upravljanje z drugimi viri. Najpomembnejši elementi aplikacijskega okvira so:

- Upravitelj aktivnosti in življenjskega cikla aplikacij (Activity Manager).
- Upravitelj z obvestili v statusni vrstici (Notification Manager).
- Grafični elementi (Views System), ki predstavljajo gradnike uporabniških vmesnikov aplikacij, kot so na primer seznam, vnosno polje, gumb in slika.
- Ponudniki vsebin in podatkov drugih aplikacij (Content Providers).
- Upravitelj z zunanjimi viri (Resource Manager), ki nudi dostop do neprogramskih virov aplikacije, kot so lokalizirana besedila, grafike in datoteke, v katerih je definirana struktura uporabniškega vmesnika.
- Upravitelj namer.

Knjižnice

Knjižnice aplikacijam nudijo dostop do strojne opreme in funkcionalnosti Linux jedra. Nabor programskih knjižnic tega nivoja obsega večpredstavne knjižnice, upravitelja površin (Surface Manager), knjižnice za brskalnik (WebKit), knjižnico, namenjeno prikazu bitne in vektorske pisave (FreeType), OpenGL za prikaz 2D in 3D grafike ter SQLite kot upravitelja relacijskih zbirk podatkov. Razvijalci lahko za razvoj aplikacij uporabijo katero od obstoječih knjižnic, ali pa jih po potrebi napišejo sami.

Izvajalno okolje (ang. Runtime)

Izvajalno okolje se nahaja v istem nivoju kot knjižnice, vsebuje pa navidezni stroj, imenovan Dalvik, ter jedrne javanske knjižnice. Navidezni stroj omogoča, da lahko na isti napravi sočasno teče več aplikacij, saj vsaka teče v lastnem procesu, imenovanem »peskovnik« (ang. Sandbox). Zadolžen je za poganjanje aplikacij, optimiran pa je na način, da porabi čim manj procesne moči in izkoristi čim manj sistemskih virov.

Jedrne knjižnice (ang. Core Libraries), napisane v programskem jeziku java, vsebujejo zbirko vseh razredov, pripomočkov in vhodno/izhodnih vmesnikov, ki jih potrebujemo pri razvoju aplikacij.

Linux jedro

Linux jedro predstavlja osnovni nivo operacijskega sistema, ki je prek gonilnikov povezan s strojno opremo in funkcijami nižjega nivoja. Med drugim zagotavlja varnost, upravljanje s pomnilnikom, upravljanje s procesi in povezavo z omrežjem. Vsebuje gonilnike za osnovne strojne elemente, kot so: kamera, zaslon, USB, zvok, Wi-Fi … Njegova odlika je tudi zanesljivost in robustnost. Osnovno Linuxovo jedro je sicer rahlo prilagojeno potrebam Androida.

2.3. Programske komponente aplikacije

Vsaka aplikacija na operacijskem sistemu Android je samostojna in se izvaja povsem neodvisno od drugih aplikacij. Vse aplikacije vsebujejo:

- aplikacijske komponente s programsko kodo, v kateri je določeno delovanje aplikacije;
- datoteke virov (ang. Resource files), kot so: XML datoteke za določanje barv uporabniškega vmesnika, menijev, vrednosti spremenljivk, besedil, slikovne datoteke in drugo.

Vsaka aplikacija ima vstopno (glavno) aplikacijsko komponento, preko katere sistem vstopa v aplikacijo.
Aplikacije na operacijskem sistemu Android so lahko sestavljene iz sledečih aplikacijskih komponent:

- Aktivnosti (ang. Activity)
- Storitve (ang. Service)
- Sprejemniki namer (ang. Broadcast Receivers)
- Ponudniki vsebin (ang. Content Providers)

Komponente Aktivnost, Storitev, Sprejemnik namer in Ponudnik vsebin operacijski sistem aktivira s pomočjo asinhronega sporočila *Intent*. Vsaka komponenta ima tudi svoj življenjski cikel.

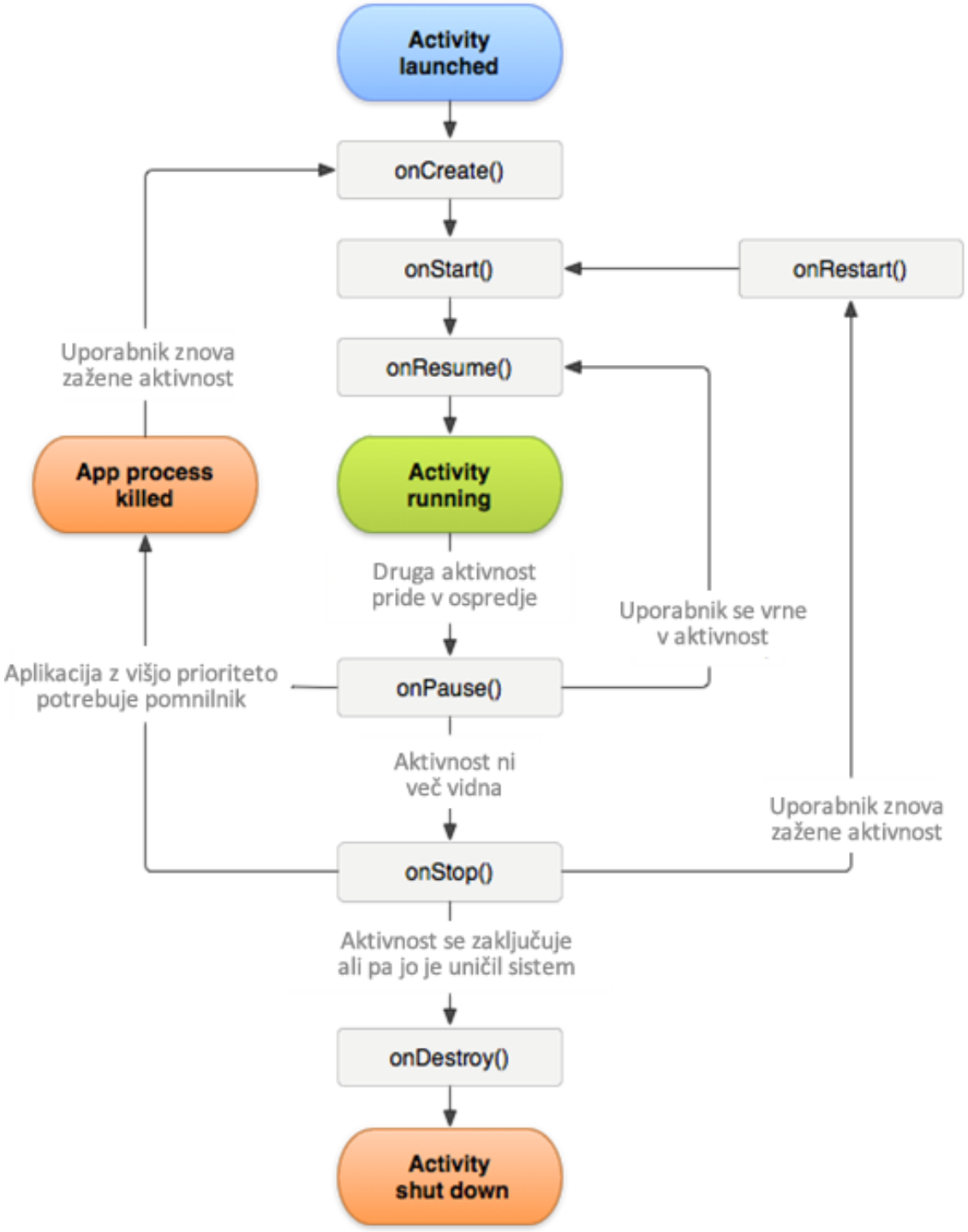
Aktivnost

Aktivnost je aplikacijska komponenta, ki je namenjena vizualni predstavitvi aplikacije. Njen glavni namen je prikaz uporabniškega vmesnika, s katerim uporabnik komunicira. Aplikacija ima lahko več aktivnosti, vsaka pa je praviloma namenjena eni zaključeni nalogi in predstavlja en zaslon uporabniškega vmesnika. Čeprav je vsaka aktivnost samostojna in neodvisna od ostalih, aktivnosti sodelujejo in skupaj predstavljajo aplikacijo. Vsaka aktivnost ima pripadajočo XML datoteko, v kateri je definiran uporabniški vmesnik aktivnosti. Vsaka aktivnost je tudi navedena v manifestu aplikacije in jo lahko uporabljajo tudi druge aplikacije (primer je pošiljanje fotografij prek SMS).

Ob zagonu nove aktivnosti se predhodna zaključi ali začasno ustavi. Ker se to lahko zgodi kadar koli in je potrebno tovrstne dogodke v programski kodi predvideti, je potrebno dobro poznati življenjski cikel aktivnosti.

Življenjski cikel aktivnosti

Življenjski cikel aktivnosti (Slika 3) se nanaša na stanja, v katerih se aktivnost lahko nahaja. Ta so: nadaljevanje, začasna zaustavitev in ustavitev. Med prehodi med stanji operacijski sistem pokliče ustrezno izmed metod *onCreate*, *onStart*, *onResume*, *onPause*, *onStop*, *onDestroy* in da s tem razvijalcu možnost odzvati se na skorajšnjo spremembo stanja. Klica metod: *onCreate* in *onDestroy* določata začetek in konec življenjskega cikla aktivnosti. Uporabniški vmesnik je viden med klicema metod *onStart* in *onStop*, odziven pa v stanju *onResume*.



Slika 3: Življenjski cikel aktivnosti [13]

Ker se aplikacija lahko kadar koli ustavi ali zaključi, je ob zaključku ali ustavitvi aktivnosti potrebno vsakokrat shraniti tudi uporabnikove podatke.

Storitev (ang. Service)

Storitve opravljajo naloge »v ozadju«, brez uporabe uporabniškega vmesnika. Primer storitve je predvajanje glasbe, med katerim lahko uporabljamo drugo aplikacijo. Storitve so namenjene izvedbi dolgotrajnih operacij, večinoma komunicirajo z drugimi komponentami sistema Android, preko njih pa aplikacija lahko ponudi tudi del svojih storitev drugim aplikacijam.

Sprejemnik sporočil in namer (ang. Brodcast Receiver)

Sprejemnik je namenjen odzivanju aplikacije na sistemska sporočila in namere. Sprejemnik v primeru dogodka dobi sporočilo od operacijskega sistema in lahko ustvari obvestilo uporabniku v statusni vrstici. Vsako sporočilo je dostavljeno v obliki sporočila *Intent*.

Sporočila delimo na dve vrsti.

- Običajna sporočila (ang. normal broadcast):
    - Le-ta se pošiljajo z uporabo metode *Context.sendBroadcast*.
    - Sporočilo dostavimo vse sprejemnikom, lahko tudi vsem hkrati.

- Naročena sporočila (ang. order broadcast):
    - Poslano z uporabo metode *Context.sendOrderBroadcast*.
    - Sporočilo pošiljamo zaporedno določenim sprejemnikom.
    - Posamezen sprejemnik lahko nato drugim pošlje svoje rezultate, ali pa prekine nadaljnje pošiljanje sporočila.
    - Atribut *android: priority* določa vrstni red sprejemnikov.

Ponudniki vsebin (ang. Content providers)

Ponudniki vsebin se uporabljajo za upravljanje, shranjevanje oziroma dostop do strukturiranih podatkov znotraj aplikacije, lahko pa se uporabljajo tudi za izmenjavo in deljenje podatkov z ostalimi aplikacijami. Deljenje podatkov med aplikacijami lahko poteka samo prek ponudnika vsebin, do njih pa dostopajo prek istega vmesnika.

Operacijski sistem Android vsebuje številne ponudnike vsebin za različne tipe podatkov (avdio, video, slike, kontakti, koledar ...). Podatki se v povezavi s ponudnikom vsebin shranjujejo v relacijske podatkovne zbirke sistema SQLite.

Intent je objekt, s katerem lahko izvedemo interakcijo med različnimi aplikacijskimi komponentami. Uporabljamo ga v treh osnovnih primerih: za zagon aktivnosti, za zagon storitve in, kot že omenjeno, za dostavo sporočil.

2.3.1. Android manifest

Vsaka aplikacija potrebuje datoteko *AndroidManifest.xml*, ki je v korenski mapi. Osnovna naloga datoteke je informiranje sistema o komponentah aplikacije. V njej so deklarirane informacije o strukturi aplikacije in metapodatki, ki jih operacijski sistem potrebuje, da aplikacijo lahko zažene. Operacijski sistem preko manifesta ocenjuje tudi zahtevano konfiguracijo in določi potrebne zmogljivosti.

V tej datoteki definiramo tudi dovoljenja in uporabniške pravice, ki jih aplikacija potrebuje. Glavni elementi manifesta so:

- Poimenovanje javanskih paketov v aplikaciji. Paketna imena služijo kot unikaten identifikator.
- Predstavitev in opis aplikacijskih komponent, kot so: aktivnosti, storitve, sprejemniki namer, ponudniki vsebin.
- Določitev, kateri proces bo gostil komponente aplikacij.
- Določitev, katera dovoljenja mora imeti aplikacija za dostop do zaščitenih delov API in interakcije z drugimi aplikacijami.
- Določitev dovoljenj in pravic, ki so potrebne za komunikacijo z drugimi komponentami aplikacije.
- Navajanje razredov, ki definirajo profile in ostale pomembnosti za zagon aplikacije.
- Definiranje minimalne različice API, ki jo aplikacija potrebuje za delovanje.
- Vsebuje nabor knjižnic, ki jih aplikacija potrebuje za svoje delovanje.
- Določi strojne in programske zahteve (na primer: uporaba tehnologije Bluetooth, kamere itd.).

Manifest torej skrbi za varno okolje, v katerem imajo aplikacije dostop le do tistih delov sistema, ki jih dejansko potrebujejo in za katere imajo dovoljenje, hkrati pa omogoča operacijskemu sistemu izmenjavo podatkov med aplikacijami.

2.4. Android SDK

Android SDK (ang. Android Software Development Kit) je paket razvojnih orodij, ki se uporablja za razvoj aplikacij za omenjeni operacijski sistem. Vsebuje knjižnice in ostala potrebna orodja za ustvarjanje in povezovanje mobilnih aplikacij.

Osnovni namen paketa Android SDK je prevajanje programske kode (napisane v programskem jeziku java) v arhivski paket APK. Na ta način vso programsko kodo aplikacije združimo v eno datoteko, ki se uporablja za nameščanje aplikacije na mobilni napravi.

Vsebino paketa Android SDK sestavljajo:

- knjižnice in programska orodja za pripravo aplikacij (komponente za podporo aplikacijam in njihovim uporabniškim vmesnikom, knjižnice za dostopnost, ravnanje s podatki, omrežno povezljivost itd.);
- razvojna orodja, ki med drugim omogočajo prevajanje, nameščanje in zaganjanje aplikacij, komunikacijo z mobilnimi napravami in razhroščevanje na napravi ali emulatorju;
- dokumentacija orodij in knjižnic;
- primeri s programsko kodo;
- sistemske slike naprav (ang. system images) za vsako različico operacijskega sistema;
- emulator – program, ki posnema delovanje dejanskih naprav.

2.4.1. Android Debug Bridge (ADB)

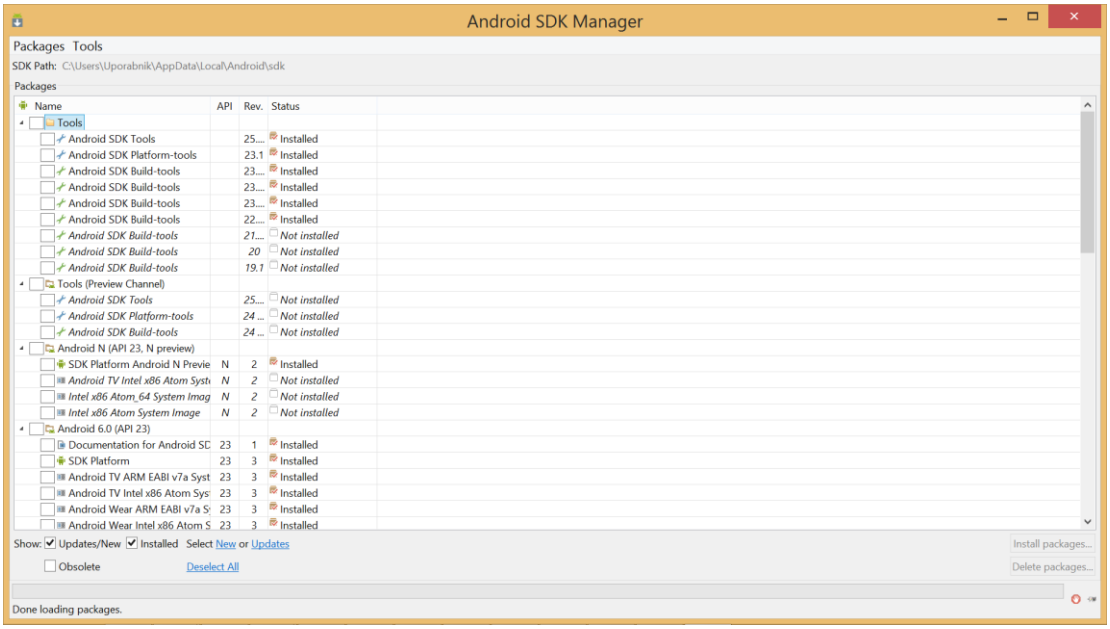
ADB je orodje, ki omogoča komunikacijo z emulatorjem oziroma s priključeno mobilno napravo z operacijskim sistemom Android. Uporablja se za iskanje napak znotraj aplikacije, za prenašanje datotek, dostop do lupine operacijskega sistema ter nameščanje in odstranjevanje aplikacij.

ADB deluje po načelu odjemalec-strežnik in ga sestavljajo tri komponente:

- proces, ki teče v ozadju na emulatorju oziroma napravi;
- odjemalec, ki teče na razvojnem računalniku;
- strežnik, ki teče na razvojnem računalniku in upravlja s komunikacijo med odjemalcem in procesom na emulatorju ali napravi.

2.4.2. Upravitelj SDK

Upravitelj SDK (ang. Android SDK Manager) nam olajša namestitev izbrane različice SDK ter drugih knjižnic in orodij, poleg tega pa omogoča tudi posodobitev ali odstranitev obstoječih starejših različic (Slika 4).



Slika 4: Upravitelj Android SDK

2.5. Android Studio

Za lažji razvoj aplikacij za operacijski sistem Android ponuja Google integrirano razvojno orodje (IDE – Integrated Development Environment), imenovano Android Studio, ki omogoča lažjo uporabo orodij in knjižnic Android SDK. Ima številne prednosti pred drugimi razvijalnimi orodji, saj med drugim ponuja:

- podporo procesom, ki delujejo v ozadju;
- bogato zbirko knjižnic za uporabniški vmesnik;
- podporo 2 D in 3 D grafiki;
- dostop do datotečnega sistema;
- podpora relacijskim podatkovnim zbirkam sistema SQLite;
- fleksibilen gradnik sistema (ang. Gradle);
- ustvarjanje APK paketov;
- grafične in vsebinske predloge za najpogostejše tipe aplikacij;
- urejevalnik grafičnega vmesnika, ki podpira funkcije povleci in spusti;
- samodejno sinhronizacijo z Googlovimi storitvami;
- orodje Lint za doseganje boljših zmogljivosti in večje uporabnosti ter združljivosti različic in za reševanje ostalih problemov;
- podporo za Googlovo platformo v oblaku, ki omogoča enostavno integriranje Googlovega oblačnega sporočanja in jedra aplikacije.

Android Studio vključuje namenske urejevalnike za posamezne vrste datotek, ki sestavljajo mobilno aplikacijo. Večina teh datotek temelji na jeziku XML. Urejevalniki pogosto omogočajo preklop med tekstovnim in grafičnim načinom urejanja dokumentov ter upravljanje projektov – vse od izvorne kode aplikacije do prevajanja, namestitve in seveda testiranja. Vsak projekt vsebuje enega ali več različnih vrst modulov, kot so na primer: aplikacijski modul, knjižni modul (ki omogoča ustvarjanje javanskih knjižnic) in testni modul.

Android Studio lahko prenesemo iz uradne strani. Pred namestitvijo moramo na računalnik namestiti ustrezno različice jave.

2.5.1. Razvojni proces

Razvojni proces (Slika 5) se nanaša na osnovne razvojne korake pri razvoju vseh vrst mobilnih aplikacij z uporabo Android Studia ali brez njega.

Faza 1: Postavitev okolja

V tej fazi vzpostavimo razvojno okolje. Nastavimo tudi emulator z navidezno napravo z želeno različico operacijskega sistema ali priključimo napravo, na katero lahko namestimo našo aplikacijo.

Faza 2: Postavitev projekta in razvoj

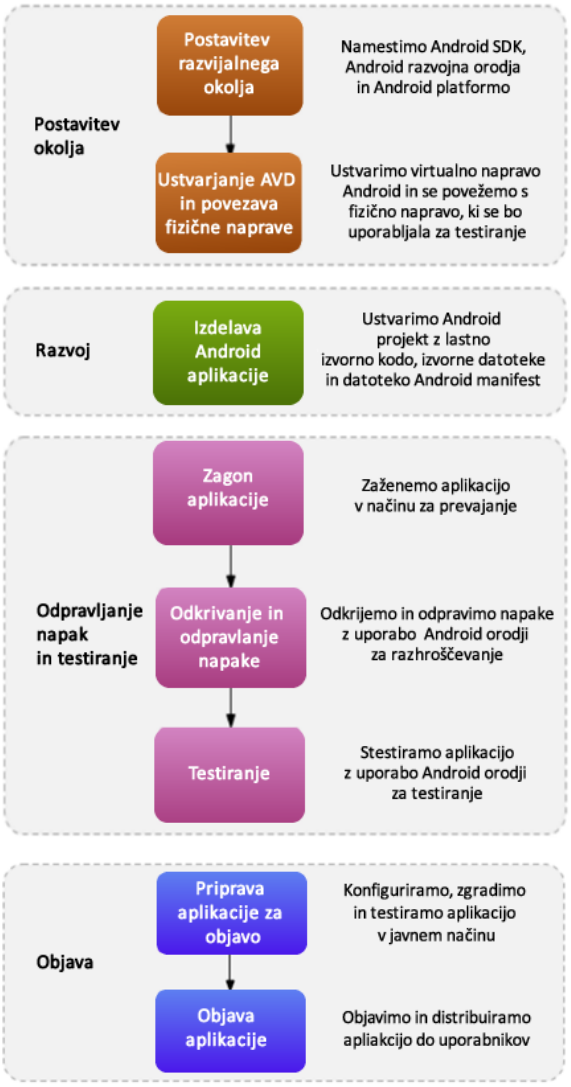
Skozi to fazo postavimo in razvijemo aplikacijo v Android Studiu skupaj z aplikacijskimi moduli, ki vsebujejo izvorno kodo in datoteke virov za to aplikacijo.

Faza 3: Prevajanje, razhroščevanje in testiranje

V tej fazi zgradimo svoj projekt, ga zapakiramo v APK paket, ki ga lahko namestimo na emulatorju ali napravi. Na koncu sledi še testiranje aplikacije z uporabo različnih orodij.

Faza 4: Objava

V tej fazi odpravimo morebitne napake in dokončno pripravimo aplikacijo za objavo in distribucijo do uporabnikov.

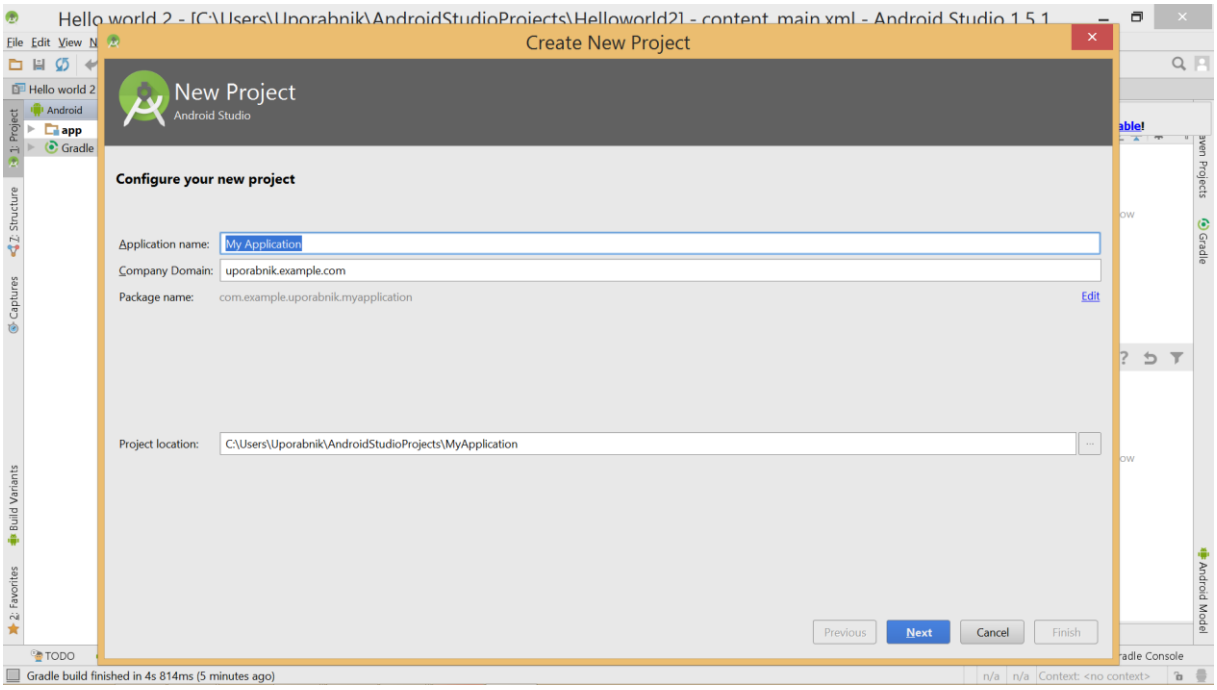


Slika 5: Razvojni proces aplikacije [14]

2.5.2. Ustvarjanje aplikacij z Android Studiem

Korak 1: Ustvarimo nov projekt

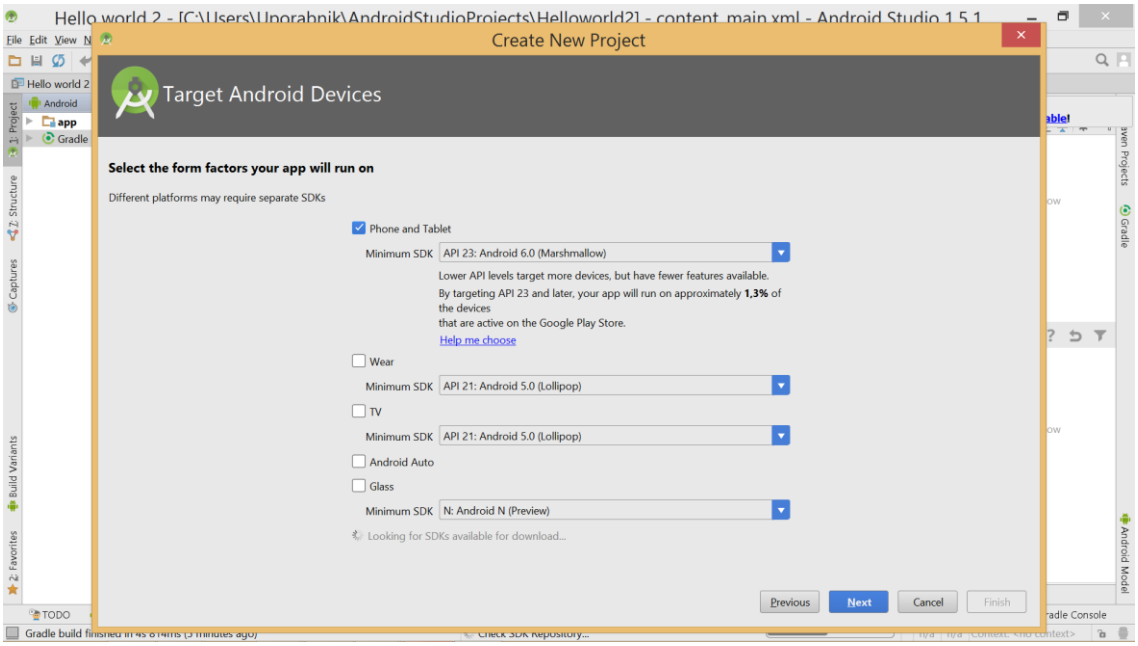
Odpremo Android Studio in izberemo: File > New Project > New Project. Čarovnik nam prek obrazca ponudi možnost izbire naprave, za katero bomo razvijali aplikacijo, hkrati pa namesti vse za čim hitrejši »začetek«. V tem koraku izberemo tudi ime aplikacije in javanskega paketa, v katerega bo umeščena (Slika 6).



Slika 6: Ustvarjanje novega projekta

Korak 2: Izbor naprave in API različice

V drugem koraku izberemo vrsto naprave (telefon, tablica, ura ali očala), ki ji je naša aplikacija namenjena (Slika 7), in različico API. Android Studio samodejno prenese izbrano različico s pomočjo upravitelja SDK, če le ta še ni nameščena.

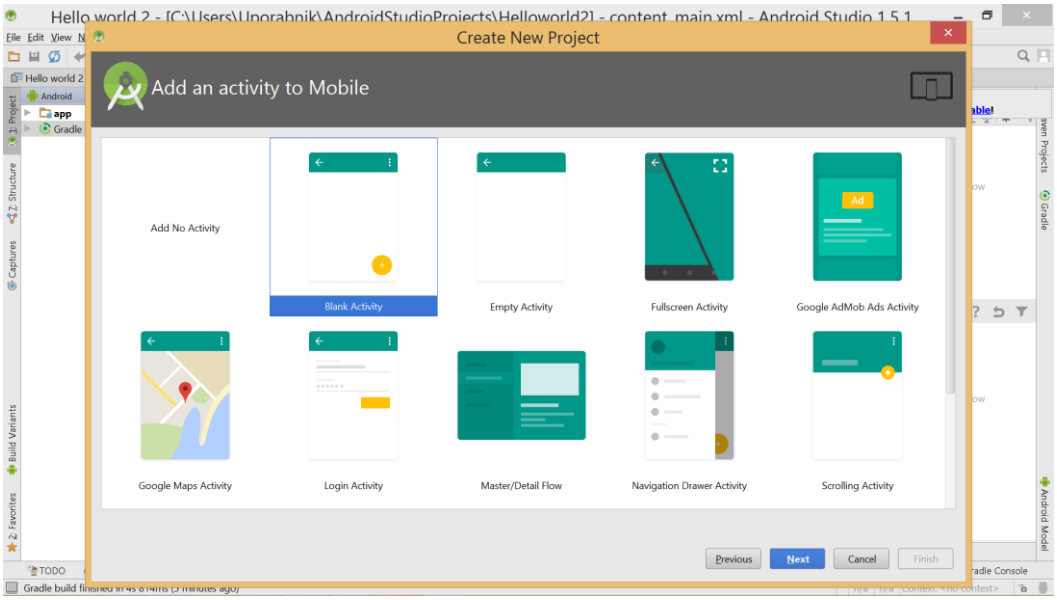


Slika 7: Izbor naprave in API različice

Ob izboru različice API se prikaže obvestilo, kolikšen je delež mobilnih naprav, ki ima nameščeno pripadajočo različico operacijskega sistema. V kolikor smo v dvomih, katero različico bi izbrali, lahko pogledamo v razdelek »Pomagajte mi izbrati«, kjer so naštete vse funkcionalnosti, ki jih vsebuje določena različica. Če izberemo nižjo različico API, dosežemo večje število naprav. Odločimo pa se lahko tudi za višjo različico, z uporabo katere bomo sicer imeli na voljo več novih funkcionalnosti, ampak bo posledično našo aplikacijo lahko uporabljalo manjše število uporabnikov.

Korak 3: Dodajanje Aktivnosti

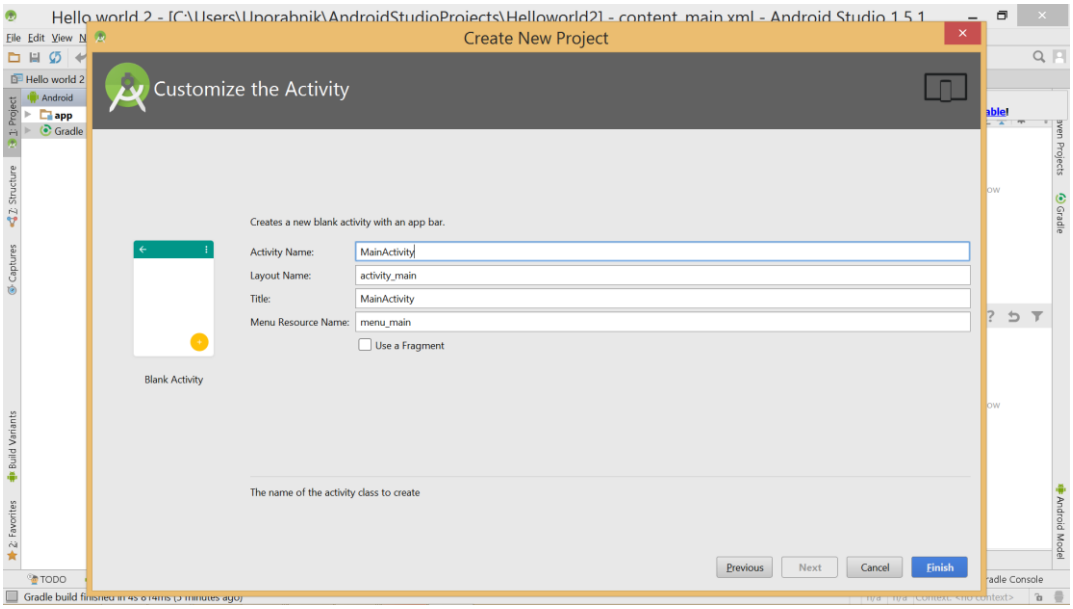
V tem koraku s klikom na izbrano predlogo izberemo predlogo glavne aktivnosti v aplikaciji (Slika 8).



Slika 8: Izbor predloge aktivnosti

Korak 4: Poimenovanje aktivnosti in njenih virov

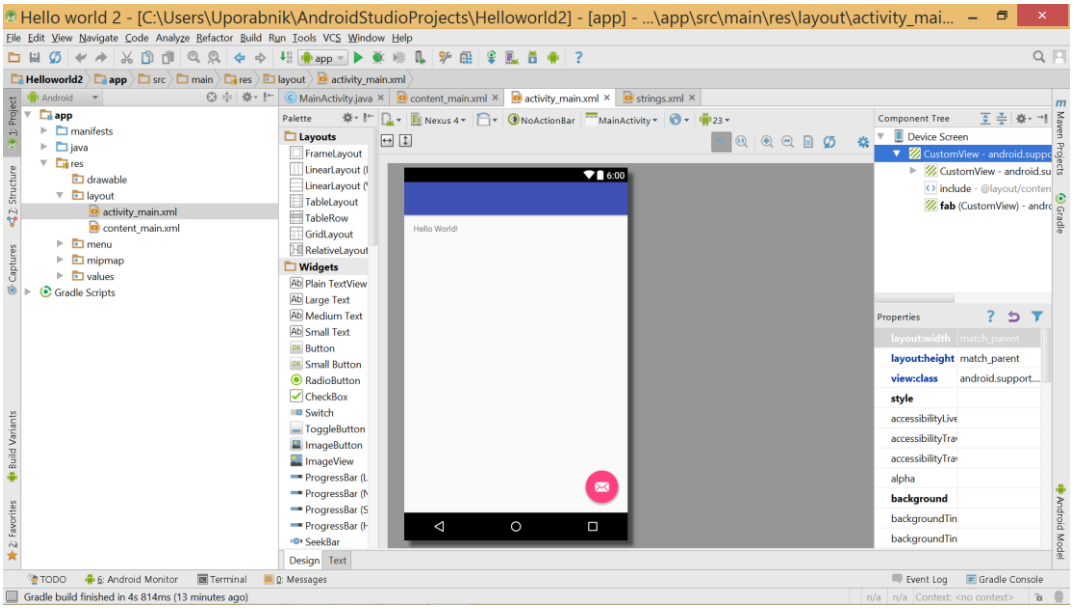
V četrtem koraku (Slika 9) izberemo ime aktivnosti in nekaterih pripadajočih virov (npr.: javanskega razreda s programsko kodo, datoteke s strukturo uporabniškega vmesnika in menijev).



Slika 9: Poimenovanje Aktivnosti in njenih virov

Korak 5: Razvijanje aplikacij

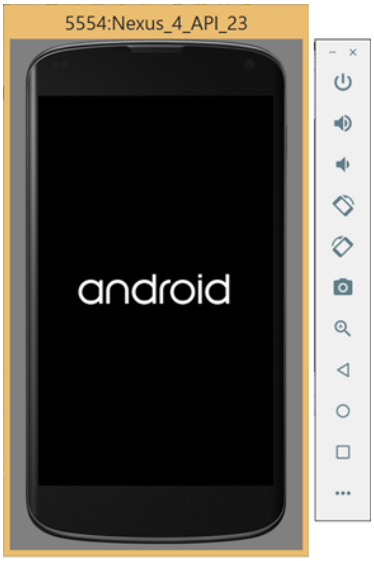
V tem zaključnem koraku Android Studio ustvari osnovno strukturo za nov projekt in prikaže razvojno okolje (Slika 10). Za aplikacije, ki so razvite za več naprav, Android Studio ustvari več modulov s posebnimi različicami virov za vsakega izmed njih.



Slika 10: Grafični urejevalnik uporabniškega vmesnika v razvojnem okolju Android Studio

2.5.3. Zagon aplikacije v emulatorju

Android SDK med svojimi orodji nudi tudi emulator izbrane mobilne naprave (Slika 11), imenovan AVD (Android Virtual Device – AVD). Le-ta posnema resnično napravo z operacijskim sistemom Android. Z uporabo AVD lahko testiramo aplikacije na različnih konfiguracijah mobilnih naprav in različicah operacijskega sistema. AVD ima različne prednaložene modele mobilnih telefonov in naprav, tako da lahko simuliramo naprave po velikosti ločljivosti zaslona.



Slika 11: Emulator mobilne naprave

S preprostim klikom na ikono AVD v orodni vrstici lahko ustvarimo novo virtualno napravo, aplikacijo pa lahko testiramo na več različnih napravah hkrati.

2.5.4. Namestitev aplikacije na zunanjo napravo

Aplikacije, ki smo jih razvili z Android Studiem, lahko namestimo oziroma poganjamo tudi na dejanskih napravah – kot na primer na osebnem mobilnem telefonu.

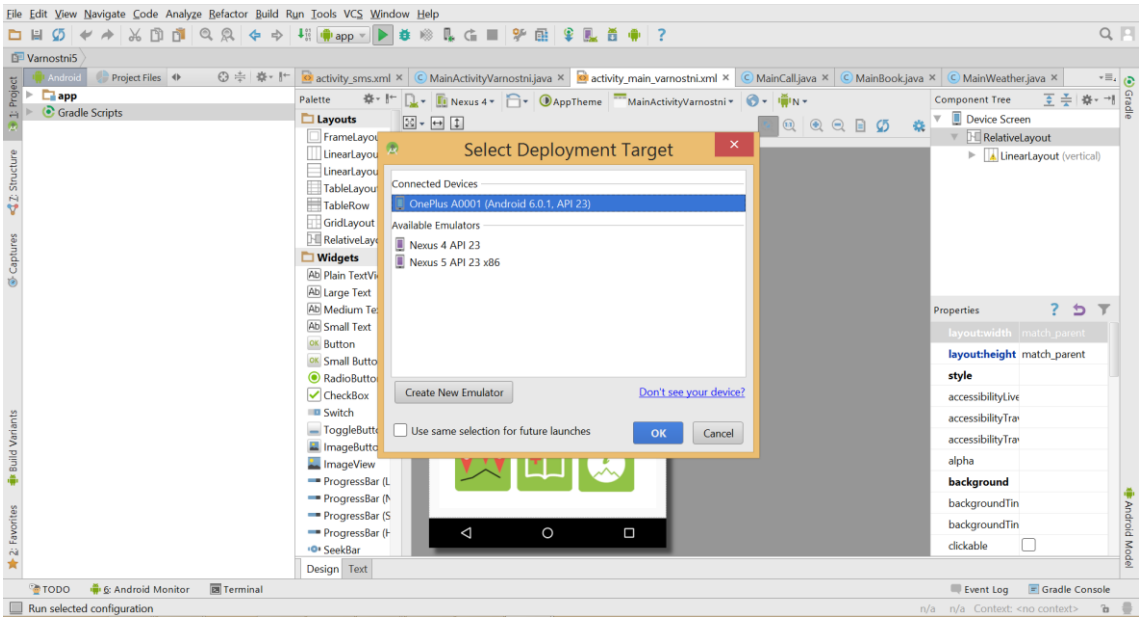
Zagon aplikacij na znanji napravi izpeljemo na naslednji način:

1. Napravo prek USB-vmesnika priključimo na računalnik, na katerem je pognan Android Studio.
2. Na njej izberemo rubriko »O napravi«, znotraj nje pa: »Delovna različica« (ang. Build Number), na katero moramo pritisniti sedemkrat. V nastavitvah se pojavi rubrika »Možnosti za razvijalce«.
3. Izberemo rubriko »Možnosti za razvijalce«, v kateri omogočimo »Razhroščevanje USB«.

V primeru, da uporabljamo operacijski sistem Windows oziroma katerega od drugih operacijskih sistemov, je potrebno naložiti še USB-gonilnik, kar storimo tako, da:

1. Priklopimo napravo na računalnik preko USB-vtičnika.
   2. Odpremo Upravitelja naprav.
   3. Odpremo rubriko »prenosne naprave«.
   4. Izberemo možnost »posodobi gonilnik«.
   5. Namestimo ustrezen gonilnik.

Ko smo omogočili USB razhroščevanje in namestili ustrezen gonilnik, lahko v Android Studiu ob zagonu aplikacije izberemo želeno napravo (Slika 12).



Slika 12: Prikaz izbire mobilne naprave ali emulatorja za poganjanje aplikacije

3. Tehnologije za določanje položaja mobilnega terminala

3.1. Geolokacija

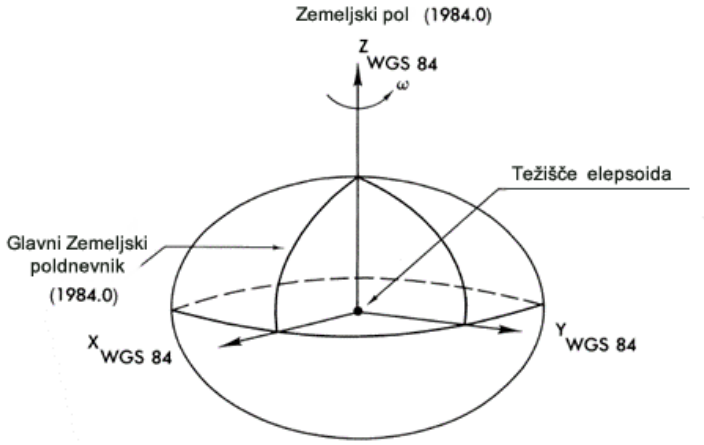
Geolokacija je proces pridobivanja geografskega položaja naprave, na primer za mobilni telefon, na internet priključen računalnik, radar ali katero drugo napravo v realnem času. Izraz geolokacija se lahko nanaša na postopek ugotavljanja položaja ali oceno položaja, ki postane rezultat takega postopka.

Položaj sodobne mobilne naprave lahko načeloma določimo na vsaj dva načina. Najbolj natančen položaj naprave dobimo z uporabo vgrajenega sprejemnika signalov satelitskih navigacijskih sistemov. Če signali satelitov niso na voljo, lahko geolokacijske aplikacije uporabijo informacijo o oddaljenosti do najbližjih baznih postaj mobilnega omrežja in z uporabo postopka, imenovanega trilateracija, določijo približen položaj naprave. Ta metoda sicer ni tako natančna kot ugotavljanje položaja z uporabo satelitskega navigacijskega sistema, vendar v zadnjih letih zaradi vedno obširnejših zbirk položajev baznih postaj določanje položaja na ta način zelo napreduje.

3.1.1. Koordinatni sistem za podajanje položaja

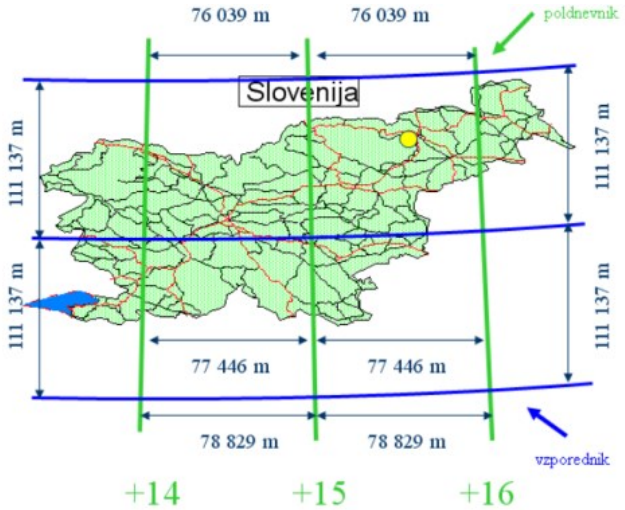
Poznamo več zemljepisnih koordinatnih sistemov, v tej diplomski nalogi pa se bomo osredotočili na geografski koordinatni sistem, ki ga uporablja sistem GPS. To je globalni koordinatni sistem z izhodiščem v težišču Zemlje, ki se vrti skupaj z njo. Položaj točke v tem sistemu je določen s kartezičnimi koordinatami (x, y, z) ali z geografskimi koordinatami (λ - lambda, φ – geografska širina, dolžina in elipsoidna višina).

Višina v tem sistemu je določena kot pravokotna oddaljenost točke nad elipsoidom WGS-84, ki aproksimira Zemljo kot telo (Slika 13).



Slika 13: Višina glede na WGS-84 [15]

Geografski koordinatni sistem določa dva kota, merjena od težišča Zemlje – njegovega koordinatnega izhodišča. Prvi kot, imenovan zemljepisna širina, podaja kot med daljico, ki povezuje težišče in poljubno točko na zemlji, ter ekvatorialno ravnino. Drugi kot, imenovan zemljepisna dolžina, pa podaja kot med začetnim poldnevnikom in poldnevnikom, na katerem se nahaja opazovana točka na Zemlji. V večjem delu sveta so za začetni poldnevnik sprejeli tistega, ki gre skozi mesto Greenwich v Veliki Britaniji. Čez Slovenijo potekajo 14., 15. in 16. vzporednik in poldnevnik s 46 stopinjami severne zemljepisne širine (Slika 14).



Slika 14: Potek vzporednikov in poldnevnikov čez slovensko ozemlje [16].

3.2. GPS – Globalni sistem določanja položaja

Globalni sistem določanja položaja (angleško Global Positioning System, GPS) je satelitski navigacijski sistem, ki se uporablja za določanje točnega položaja in časa kjer koli na Zemlji ali na zemeljski tirnici. Sistem je zasnovalo obrambno ministrstvo ZDA v začetku 70 let prejšnjega stoletja, ki ga tudi upravlja. Na začetku je bil razvit za potrebe ameriške vojske, kasneje pa je bil prilagojen tudi za civilne uporabnike. Prosto ga lahko uporablja vsakdo, ki ima ustrezen sprejemnik, kar pomeni, da ni naročnine ali pristojbine za uporabo sistema GPS.

Sistem GPS je razdeljen je na tri odseke: vesoljskega, nadzornega in uporabniškega. Vesoljski odsek vključuje satelite, nadzorni pa zemeljske postaje, ki skrbijo za nadzorovanje poti satelitov, usklajevanje njihovih atomskih ur in upravljanje s podatki, ki jih sateliti oddajajo. Uporabniški odsek sestavljajo civilni in vojaški GPS sprejemniki, ki razberejo časovne podatke iz večjega števila satelitov in njihovi podlagi izračunajo lego sprejemnikov s postopki trigonometrije.

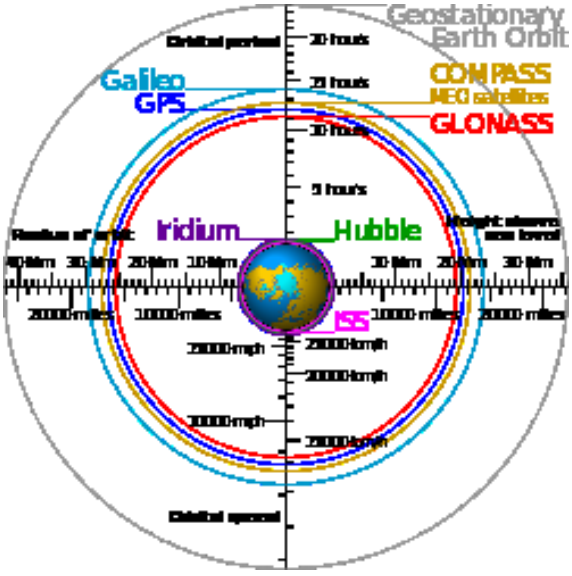
Sistem sestavlja najmanj 24 satelitov (Slika 15) v šestih ravninah tirnic (v zaporedju 30, 105, 120 in 105 stopinj razmika). Vsak od njih Zemljo obkroži dvakrat dnevno na višini 20200 km. Satelit neprestano oddaja čas lastne atomske ure in podatke o tirnici gibanja za zemeljske opazovalnice.

Slika 15: Prikaz razporeditve satelitov sistema GPS [17]

Osnovna funkcija GPS sprejemnika je izračun točnega položaja (geografskih koordinat sprejemnika. V idealnem primeru lahko na zemeljski površini naenkrat sprejemamo signale polovice satelitov. Takrat je seveda natančnost meritve najboljša, solidne rezultate pa dobimo že s sprejemom signalov s štirih do šestih satelitov. Če so dobro razporejeni po vidnem delu neba, lahko GPS sprejemnik izračuna položaj celo s signali le-treh.

Do leta 2000 so sateliti oddajali premaknjen čas in podatke o tirnici, poleg njih pa še šifrirani signal, ki je sporočal to namenoma povzročeno napako. Ta signal so lahko dešifrirali samo sprejemniki Oboroženih sil ZDA (verjetno tudi NATA). Zaradi napake so civilni sprejemniki prikazovali koordinate, ki so bile natančne do 120 metrov. Po letu 2000 pa teh motenj ne vnašajo več, zato se je tudi natančnost položaja izboljšala na 2 metra, v nekaterih primerih z boljšimi sprejemniki (dvofrekvenčnimi z zunanjo anteno) in z diferencialno metodo izmere celo na nekaj centimetrov.

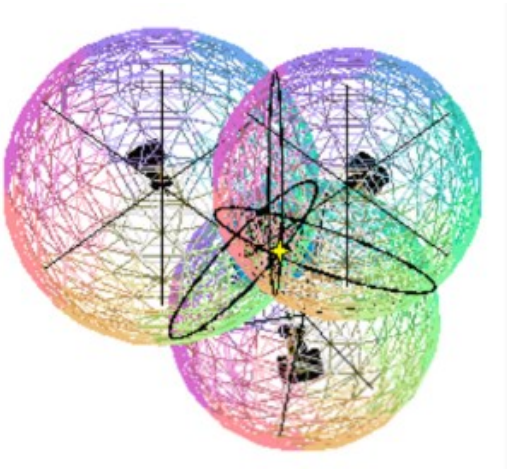
Poleg sistema GPS obstajajo tudi nekateri drugi sorodni navigacijski sistemi. Ruska različica se imenuje GLONASS, evropska GALILEO, kitajska pa BeiDou. Vsi omenjeni satelitski sistemi na nadnacionalni ravni tvorijo GNSS – Globalni navigacijski satelitski sistem (Slika 16). Možnost sočasnega sprejemanja signalov različnih satelitskih sistemov lahko zelo poveča natančnost določanja položaja.



Slika 16: Grafični prikaz navigacijskih sistemov [18]

3.2.1. Izračun položaja z uporabo sistema GPS

Za pridobitev podatkov o zemljepisni dolžini in širini, nadmorski višini ter točnem času potrebujemo signale vsaj štirih satelitov. Iz razlike med časom sprejema signala in časom njegove oddaje lahko določimo razdaljo med sprejemnikom in satelitom. Nato iz njihovih signalov (dveh nosilnih frekvenc v frekvenčnem pasu L) in notranje baze podatkov ugotovimo mesta satelitov ob času oddajanja. Sprejemnik se torej nahaja na sferi, katere središče je satelit in katere polmer je določen z razdaljo, ki jo prepotujejo radijski signali v času od trenutka oddajanja do trenutka sprejemanja signala.



Slika 17: Prikaz trilateracije v treh dimenzijah [19]

Ker sprejemnik hkrati sprejema signale več satelitov, je mogoče določiti njegovo lego na osnovi tridimenzionalne trilateracije. Trilatelacija je postopek pridobivanja relativnega položaja na presečišču treh krogov s poznanimi radiji in središči.

GPS sprejemnik mora sprejemati signal vsaj treh satelitov, da lahko izračuna položaj na zemeljski površini (zemljepisno širino in dolžino) in sledi gibanju. S štirimi ali več sateliti lahko sprejemnik določi položaj uporabnika v prostoru (zemljepisno širino, dolžino in višino), kar je zelo pomembno prav v gorah. GPS sprejemnik iz sprejema štirih ali več satelitov lahko izračuna še druge informacije, kot so hitrost, smer, tirnica, razdalja potovanja, oddaljenost do cilja in še nekatere druge količine. Od več satelitov sprejemnik lahko sprejema, večja je natančnost določanja položaja, na katerem je uporabnik.

3.3. Mobilna omrežja

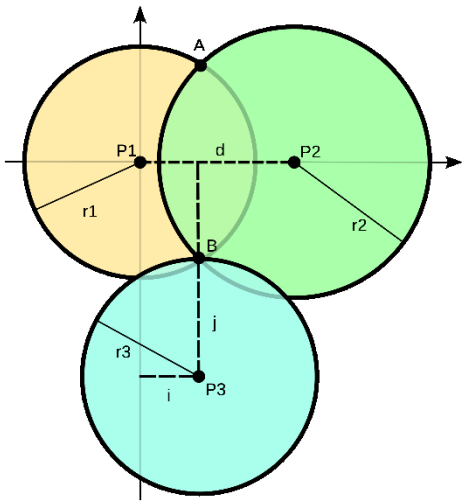
Ker so viri radijskega kanala (čas in frekvenca) omejeni, je mobilno omrežje razdeljeno na celice, ki omogočajo ponovno uporabo omenjenih virov. Glavna elementa dostopnega dela mobilnega omrežja sta:

- mobilni terminal (telefon uporabnika) in
- bazna postaja, ki se navadno nahaja v središču celice.

Bazna postaja in mobilna naprava komunicirata z uporabo radijskih valov.

3.3.1. Pridobivanje podatkov o položaju izrednega dogodka s strani operaterja

Podatke o položaju mobilnega telefona lahko pridobi tudi operater prek omrežja, pri čemer prav tako kot uporabnik pridobi podatke treh mobilnih postaj (t. i. trialateracije), s katerimi je uporabnik v stiku. V primeru pridobivanja položaja znotraj mobilnega omrežja prek trilateracije je v središču kroga bazna postaja. Tirnice treh baznih postaj se sekajo v skupni točki, le-ta pa predstavlja položaj mobilne naprave (Slika 18). Položaj mobilne naprave pridobljen s pomočjo trilateracije prek baznih postaj je manj natančen od določanja položaja preko GPS-sistema, zato je le pogojno uporaben.



Slika 18: Grafični prikaz trilateracije v dveh dimenzijah[20]

Določeni sodobni Centri za obveščanje imajo na voljo tudi napredno opremo za pridobivanje položaja s strani operaterjev, imenovano pametni lokator oziroma AML (napredna mobilna lokacija). Lokalna reševalna služba na uporabnikov telefon posreduje SMS-sporočilo z ustrezno povezavo. Ko uporabnik odpre povezavo, ga operater zazna in reševalna akcija lahko steče.

Lokator, ki deluje v tridimenzionalnem geografskem informacijskem sistemu (3 D GIS), klicatelja v nekaj sekundah locira na podlagi GSM-povezave, baznih postaj ali točk Wi-Fi. Njegovo delovanje je neodvisno od operaterjev in proizvajalcev telefonov. Da spletno orodje lahko določi položaj klicatelja, mora le-ta imeti vključen prenos podatkov in določanje lokacijskih storitev [21].

4. Mobilna aplikacija za pomoč pri reševanju v gorah

V nadaljevanju diplomskega dela je opisana mobilna aplikacija za pomoč pri reševanju v gorah. Razvita je bila z namenom olajšati dostop do natančnega položaja izrednega dogodka, kot sta na primer nesreča oz. poškodba planinca in pošiljanje sporočila v Center za obveščanje. Glavna naloga aplikacije je poiskati in posredovati položaj izrednega dogodka v planinah ali gorah, zato smo tej funkcionalnosti in pripadajočemu uporabniškemu vmesniku posvetili največjo pozornost. Posredovanje položaja smo izvedli z uporabo SMS-obvestil, ki potrebujejo najmanj virov – tako baterije, časa pošiljanja, kot tudi obremenitve omrežnih kapacitet, poleg tega pa so tudi z vidika uporabnika najbolj enostavna za uporabo. Aplikaciji smo dodali še nekaj dodatnih funkcionalnosti. Ena izmed njih je sistem varnostnega sledenja v ozadju, s katerim lahko poskrbimo za dodatno varnost ob samostojnem odpravljanju v gore, saj aplikacija omogoča 15-minutno intervalno pošiljanje položaja prek SMS sporočil. Dostop do podatkov o posredovanem položaju je omejen na vnaprej določeno osebo. Ta jih v primeru, da se planinec ne javi v dogovorjenem času, ali če se prekine samodejno redno posredovanje podatkov, posreduje reševalnim ekipam. Te podatke lahko uporabijo za rekonstrukcijo opravljene poti ter analizo nastalega položaja.

Zaradi stresnih okoliščin, v katerih se lahko znajde ponesrečenec, smo v aplikacijo vključili tudi navodila za ravnanje v primeru nesreče. Aplikacija z možnostmi sporočanja o ažurnih razmerah in stanju v gorah deluje tudi kot preventiva.

V skladu z zahtevami aplikacije smo zasnovali njeno arhitekturo, razvili njene komponente in oblikovali ustrezen uporabniški vmesnik. Vsako od komponent smo preizkusil v ločenem projektu.
Aplikacijo smo razvili s pomočjo razvojnega okolja Android Studio za različico operacijskega sistema Android 6.0.1 Marshmallow oziroma različico API 23.

Aplikacijo smo izdelovali po sklopih, ki so vključevali izvedbe posameznih funkcionalnosti in pripadajočega uporabniškega vmesnika. V prvem sklopu smo obdelali klice v Center za obveščanje na številko 112 in pošiljanje informacij o položaju izrednega dogodka prek SMS-sporočil. V drugem sklopu smo obravnavali sistem za pridobivanje trenutne lokacije in za intervalno sporočanje lokacij na izbrano številko. V tretjem pa smo se pretežno lotili elementov z informacijami in obvestili, kot so navodila za ravnanje ob nesrečah v gorah in prikazi opozoril

ter aktualnih vremenskih razmer, ki smo jih zajeli s spletne strani ARSO (Agencije Republike Slovenije za okolje).

V zaključku smo navedli tudi nekatere smotrne in možne izboljšave mobilne aplikacije, namenjene reševanju v gorah.

4.1. Uporabniški vmesnik

Uporabniški vmesnik je eden ključnih elementov mobilne aplikacije. Odločilen je za to, ali bo raba aplikacije zaživela ali pa bo postala le ena od številnih aplikacij, »pozabljenih« v Google Play trgovini. Uporabnik prek uporabniškega vmesnika upravlja z aplikacijo, hkrati pa ga vodi do rezultatov, ki jih od nje pričakuje. Dober uporabniški vmesnik odlikuje funkcionalno načrtovana modularna zasnova, ki vse dele poveže v logično celoto.

Ker je potrebno skrbeti tudi za nadaljnji razvoju aplikacije, je že ob sami zasnovi nujno predvideti možnosti za bodoče nadgradnje uporabniškega vmesnika.

4.1.1. Zasnova uporabniškega vmesnika aplikacije

Uporabniški vmesnik mora biti pri razvijani aplikaciji zaradi narave njene rabe še posebej skrbno načrtovan. Zaradi tega smo pri njegovi zasnovi poleg inovativnosti imeli v mislih tudi enostavnost in učinkovitost rabe. Izbrali smo kontrastne barve in logično razporeditev ključnih funkcionalnosti.

Bistveni elementi uporabniškega vmesnika so gumbi, prek katerih lahko uporabnik nedvoumno upravlja s svojimi zahtevami za odzivanje aplikacije. Gumbi morajo imeti minimalistično oblikovanje in morajo uporabniku nedvoumno sporočati, kaj se bo zgodilo ob pritisku nanj. Hkrati morajo biti dovolj veliki in smiselno razporejeni, da omogočajo rokovanje vsem starostnim skupinam.

Z razporeditvijo gumbov smo sledili potrebam, ki se pojavijo ob izrednih dogodkih. Potrebe v takšnih dogodkih tudi nedoumno sledijo zelo jasni prioriteti, ki smo ji sledili tudi sami z njihovo vertikalno postavitvijo. Prednostne gumbe smo postavili višje od ostalih.

Gumbe uporabniškega vmesnika smo predstavili v obliki ikon, ki so bile izdelane posebej za naša aplikacijo. Zanje smo zbrali zeleno barvo, saj le-ta s psihološkega vidika pomirja uporabnika in tako prispeva k dodatni zbranosti ob uporabi aplikacije pri izrednih dogodkih. Primeri gumbov so prikazani na (Sliki 19).

SMS v sili

Osrednji grafični element na gumbu predstavlja standardna ikona za SMS-sporočilo, ki ji je dodan označevalec položaja. Uporabnik lahko z gumba razbere, da gre za funkcionalnost, s katero bo posredovan položaj prek SMS- sporočila.

Gumb za klic številke 112.

Osrednji grafični element na gumbu predstavlja slušalka, dodano pa je besedilo »Call 112«, ker bi se brez jasnega sporočila, katero številko bo poklical, uporabnik lahko zmedel.

Sledenje v ozadju

Osrednji grafični element na gumbu predstavljajo trije označevalci položaja, ki so dodani nad pobočje. Uporabnik z gumba lahko razbere, da gre za sledenje oziroma ugotavljanje položaja v časovnih intervalih.

Navodila za ravnanje v primeru izrednih dogodkov

Osrednji grafični element na gumbu predstavlja odprta knjiga, ki ji je dodan rdeč križ. Uporabnik z gumba razbere, da gre za zapisana navodila o ravnanju v primeru poškodb oz. nesreč in je potrebna prva pomoč ali reševanje.

Opozorila in vremenske informacije

Osrednji grafični element na gumbu predstavlja črka »i« nad goro (natančneje nad Triglavom). Uporabnik z gumba razbere, da gre za pridobivanje informacij o stanju v gorah.

Slika 19: Gumbi, uporabljeni v aplikaciji

V stresnih situacijah uporabniku zelo olajšajo položaj zmanjšana možnost izbir, vključenost ključnih podatkov v SMS-sporočila in že izpolnjeni vnosni obrazci. Vse to mu prihrani čas, hkrati pa zmanjša možnosti za napake pri vnosu. Uporabnik, ki obvladuje tehnološki

pripomoček oz. mobilno aplikacijo, namreč deluje precej bolj zbrano in umirjeno, saj ve, da je naredil vse, kar je bilo v njegovi moči in da bo pomoč kar se da hitro prišla.

Pomembno je tudi, da uporabnika pri uporabi aplikacije ne zmedemo, zato aplikacija ne sme imeti vklopljene možnosti za spremembo prikaza ob spremenjeni orientaciji naprave (zaradi spremenjene lege iz navpične v vodoravno). To bi zmotilo uporabnika in otežilo čas dostopa do funkcionalnosti.

4.2. Arhitektura aplikacije

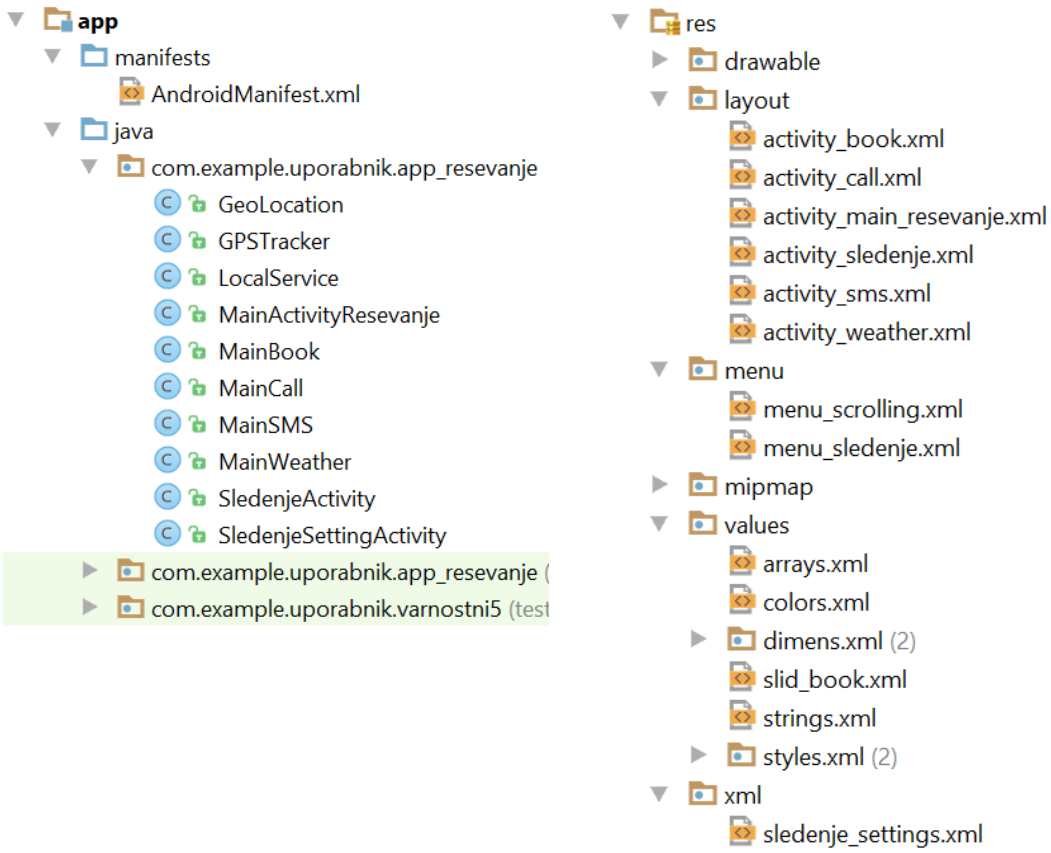
Aplikacija za reševanje v gorah je glede na ključne funkcionalnosti razdeljena na 6 modulov:

1. Vstopna stran aplikacije s prikazom položaja uporabnika
2. Klic številke 112 – Centra za obveščanje in reševanje
3. Pošiljanje GPS-položaja v Center za obveščanje in reševanje preko SMS-sporočil
4. Intervalno sporočanje o položaju s sledenjem v ozadju
5. Navodila za ravnanje v primerih izrednih dogodkov
6. Opozorila in informacije o stanju v gorah

Na podlagi te razdelitve smo zasnovali tudi uporabniški vmesnik, ki ima šest oken, kot to prikazuje Slika 20. Puščice na sliki nakazujejo prehod med okni uporabniškega vmesnika oziroma med funkcionalnostmi, ki jih ta predstavljajo.

Slika 20: Pregled in povezanost modulov ter njihovih uporabniških vmesnikov

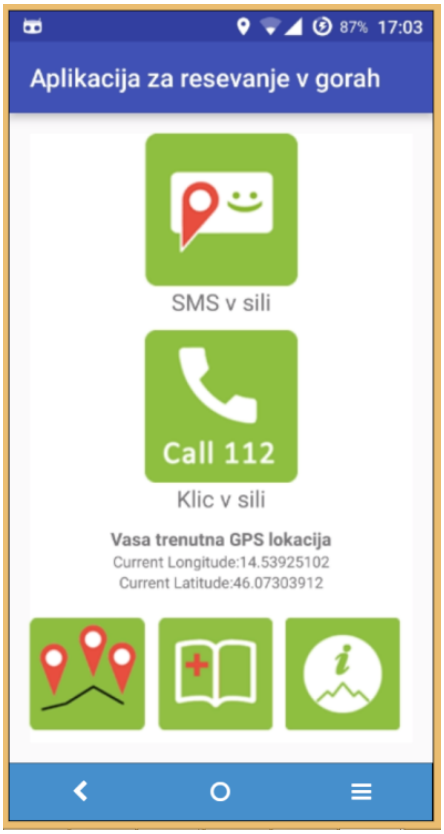
Na Sliki 21 je prikazana razporeditev virov aplikacije. Te sestavljajo manifest in javanski razredi programskih modulov aplikacije (Slika 21, levo) in datoteke uporabniškega vmesnika, menijev, vrednosti raznih spremenljivk ter nastavitve (Slika 21, desno).



Slika 21: Struktura aplikacije. Levo: manifest in javanski razredi, desno: datoteke uporabniškega vmesnika, besedil, menijev, vrednosti spremenljivk in nastavitve

4.2.1. Vstopna stran aplikacije

Glavna naloga vstopne strani je jasen prikaz in dostop do funkcionalnosti, ki jo aplikacija ponuja. Vstopno stran sestavlja pet gumbov, na njej pa se prikaže tudi trenutni položaj uporabnika (Slika 22).



Slika 22: Vstopna stran aplikacije

Vstopna stran aplikacije je izdelana v aktivnosti *MainActivityVarnostni*. Le-to smo določili v manifestu aplikacije kot glavno oziroma vstopno točko (ang. Launcher Activity), zato se ob zagonu aplikacije izvede prva (Slika 23).

```
<activity android:name=".MainActivityVarnostni">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
```

Slika 23: Določitev vstopne točke aplikacije v manifestu

Uporabniški vmesnik vstopne strani je definiran v datoteki *activity_main_resevanje.xml*. Ta vsebuje pet gumbov in področje za izpis trenutnega položaja uporabnika. Gumbe definiramo z uporabo gradnikov *ImageButton*, kar je prikazano na Sliki 24.

```
<ImageButton
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Call Button"
    android:onClick="goToActivityCall"
    android:id= "@+id/Call_button_zacetna"
    android:src="@drawable/call"
    android:background="#000000"
    android:layout_marginTop="20dp"
    android:layout_marginRight="10dp"
    android:layout_marginBottom="10dp" />
```

Slika 24: Določitev gumba z uporabo gradnika ImageButton

Ob pritisku na želeni gumb se izvede metoda v javanskem razredu aktivnosti, ki jo določa vrednost atributa *onClick*. V našem primeru vrednost tega atributa določa, da se ob kliku na gumb pokliče metoda *goToActivityCall* v javanskem razredu *MainActivityVarnostni* (Slika 25). V tej metodi s pomočjo sporočila *Intent* poženemo drugo aktivnost. V spodnjem primeru je to aktivnost za klic v sili, imenovana *MainCall*.

```
public void goToActivityCall(View view) {
   Intent i = new Intent(MainActivityVarnostni.this, MainCall.class);
   startActivity(i); //Zagon aktivnosti, definirane s sporočilom
Intent}
```

Slika 25: Priklic aktivnosti za klic v sili z metodo goToActivityCall

Priklice ostalih aktivnosti (pošiljanje SMS-sporočil na številko 112; sledenje v ozadju; pregled informacij o ravnanju v primeru nesreče; informacij in obvestil o stanju v gorah) izvajamo na podoben način.

4.2.2. Prikaz lokacije na vstopni strani

Ko poženemo aplikacijo, se zaženejo elementi za uporabo GPS-modula, ki začne s pridobivanjem položaja na podlagi sprejemanja signala s satelitov. Za celotno operacijo je potrebnih okoli 10 s. Ko aplikacija pridobi ustrezne podatke, se le-ti prikažejo na zaslonu pod gumbom za SMS-sporočilo.

Ob zagonu glavne aktivnosti *MainActivityVarnostni* se v vstopni metodi *onCreate* pokliče storitev *LocationManager*, ki je namenjena dostopu do sistemske lokacijske storitve. Le-ta aplikacijam ponuja periodično osveževanje geografskega položaja. V okviru storitve *LocationManager* določimo tudi način in čas osveževanja položaja (Slika 26).

```
locationManager = (LocationManager) getSystemService(Context.LOCATION_SERVICE);
Criteria criteria = new Criteria();
mprovider = locationManager.getBestProvider(criteria, false);
if (mprovider != null && !mprovider.equals("")) {
    if (ActivityCompat.checkSelfPermission(this,
        Manifest.permission.ACCESS_FINE_LOCATION) !=
        PackageManager.PERMISSION_GRANTED &&
        ActivityCompat.checkSelfPermission(this,
        Manifest.permission.ACCESS_COARSE_LOCATION) !=
        PackageManager.PERMISSION_GRANTED) {
            return; //preverjanje dovoljenja za GPS
        }
     Location location = locationManager.getLastKnownLocation(mprovider);
     //pridobivanje GPS položaja uporabnika
     locationManager.requestLocationUpdates(mprovider, 15000, 1, this);
     //osveževanje položaja po določenem x času oz. prehojenih x metrih
     if (location != null)
         onLocationChanged(location);
     //izpis položaja na zaslonu
     else
     Toast.makeText(getBaseContext(), "Ne zaznam nobene lokacije, preverite
nastavitve", Toast.LENGTH_SHORT).show();
    } //obvestilo uporabniku, v primeru, da ni mogoče določiti položaja
}
```

Slika 26: Uporaba storitve LocationManager in osveževanje položaja

Storitev *LocationManager* sporoča osveženi položaj z izvedbo povratnega klica metode *onLocationChanged*, ki pridobljeni položaj izpiše na zaslonu (Slika 27).

```
public void onLocationChanged(Location location) {
    TextView longitude = (TextView)
    findViewById(R.id.textViewLokacija1);
    TextView latitude = (TextView)
    findViewById(R.id.textViewLokacija2);

    longitude.setText("Current Longitude:" + location.getLongitude());
    latitude.setText("Current Latitude:" + location.getLatitude());
}
```

Slika 27: Izpis osveženega položaja s pomočjo metode onLocationChanged

Položaj prikazujemo z rabo elementa *TextView, ki ga definiramo* v datoteki *Activity_main_resevanje.xml*, v kateri je določen uporabniški vmesnik vstopne aktivnosti (Slika 28).

```
<TextView
    android:id="@+id/textViewLokacija1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="10sp" />
```

Slika 28: Določitev elementa TextView za prikazovanje položaja uporabnika

4.2.3. Klic številke 112 – Centra za obveščanje in reševanje

Aktivnost »Klic 112« oziroma »Call 112« je namenjena vzpostavitvi klica s Centrom za obveščanje in reševanje. Uporabnik vzpostavi povezavo s pritiskom na gumb, ki je prikazan na Sliki 29.

Slika 29: Zaslon za klic Centra za obveščanje na številko 112

Aktivnost je izdelana v razredu *MainCall* in je dostopna z vstopne strani aplikacije. Uporabniški vmesnik je definiran v datoteki *activity_call.xml*, vsebuje pa en sam gumb za klic številke 112 – Centra za obveščanje in reševanje (Slika 30).

```
<ImageButton
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/Call_button"
    android:src="@drawable/call"
    android:background="#000000"
    android:layout_below="@+id/textView4"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="73dp" />
```

Slika 30: Določitev gumba za klic 112

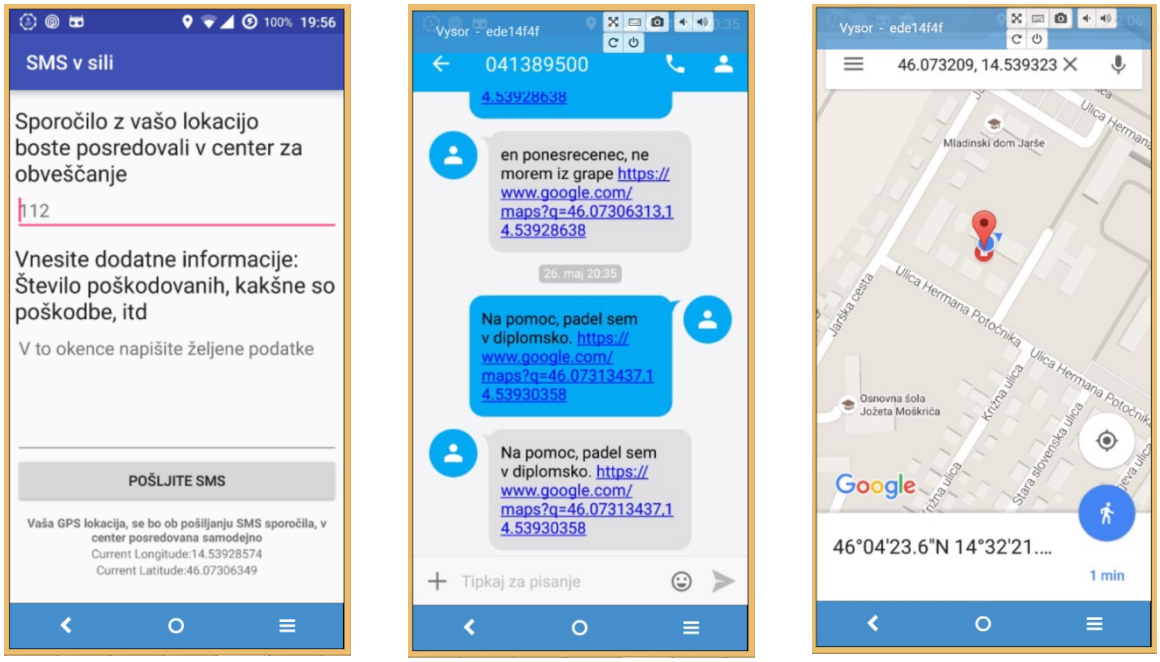
V javanskem razredu *MainCall* smo definirali metodo *onCreate*, znotraj katere smo programsko določili, da se ob pritisku na gumb izvede metoda call. V metodi *call* smo z rabo sporočila *Intent* zagnali aktivnost za izvedbo telefonskega klica in ji posredovali številski parameter 112 kot telefonsko številko, na katero naj se klic opravi (Slika 31).

```
private void call() {
    try {
        Intent callIntent = new Intent(Intent.ACTION_CALL);
        callIntent.setData(Uri.parse("tel:112"));
        startActivity(callIntent); // Zagon nove aktivnosti, ki
smo jo definirali v sporočilu Intent
    } catch (ActivityNotFoundException activityException) {
        Toast.makeText(getApplicationContext(),"Vaša aktivnost se
ni vzpostavila",
         //prikaz obvestila (ang. notification massage) na
zaslonu, v kolikor se aktivnost ne vzpostavi
        Toast.LENGTH_SHORT).show();
    }
}
```

Slika 31: Metoda call za izvedbo klica v sili

4.2.4. Pošiljanje SMS sporočila s položajem izrednega dogodka

Ob pritisku na gumb za pošiljanje SMS sporočil se požene aktivnost, ki na številko 112 pošlje SMS sporočilo z GPS-položajem. Hkrati s pošiljanjem podatkov o položaju lahko ponesrečenec posreduje tudi druge podatke, ki so pomembni za organizacijo reševanja. Uporabniški vmesnik aktivnost za pošiljanje SMS-sporočila ima tri elemente: polje z že vpisano telefonsko številko, na katero bo poslano SMS-sporočilo, polje za vnos ponesrečenčevega sporočila in gumb za izvedbo pošiljanja (Slika 32).



Slika 32: Aktivnosti za pošiljanje SMS-sporočila. Levo: pošiljanje SMS-sporočila s GPS-položajem, v sredini: prikaz poslanega SMS-sporočila v aplikaciji za SMS-sporočila, desno: prikaz položaja uporabnika v aplikaciji Google Maps

Aktivnost za pošiljanje SMS-sporočil je pognana z vstopne aktivnosti aplikacije in definirana v javanskem *MainSMS*, njen uporabniški vmesnik pa v datoteki *activity_sms.xml* (Slika 33).

```
<TextView
    android:paddingTop="@dimen/activity_vertical_margin1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/sms_sp_res"
    android:textAppearance="?android:attr/textAppearanceLarge" />

<EditText
    android:id="@+id/smsBody"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:gravity="top"
    android:hint="@string/sms_vpis_text"
    android:inputType="textMultiLine"
    android:lines="5" />


<Button
    android:id="@+id/send"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_weight="5"
    android:text="Pošljite SMS" />
```

Slika 33: Določitev uporabniškega vmesnika aktivnosti za pošiljanje SMS sporočila

V razredu *MainSMS* smo določili, da se v primeru pritiska na gumb izvede metoda *onClick*. V njej preberemo vpisano telefonsko številko in uporabnikovo besedilo ter pridobimo uporabnikov GPS-položaj. Nato pokličemo upravljalnik SMS sporočil *SmsManager*, s pomočjo katerega pošljemo SMS-sporočilo z uporabnikovim položajem in sporočilom. Uporabnik dobi obvestilo o uspešnem pošiljanju SMS-sporočila (ang. notification massage) prek obvestila Toast (Slika 34).

```
SmsManager smsManager = SmsManager.getDefault();
    smsManager.sendTextMessage(number, null, sms, null, null);
    //pošiljanje SMS s pomočjo objekta SmsManager,
    //SMS sporočilo sestavimo z uporabo metode sendTextMessage

    Toast.makeText(getApplicationContext(),
    "SMS je bil uspešno poslan, pomoč je na poti!",
    //Uporabnik dobi obvestilo, če je bil SMS uspešno poslan
            Toast.LENGTH_LONG).show();
}
```

Slika 34: Pošiljanje SMS-sporočila z informacijami o položaju

4.2.5. Varnostno sledenje

Modul za varnostno sledenje je namenjen intervalnemu pošiljanju uporabnikovega položaja na izbrano številko v obliki SMS-sporočil. Poslana sporočila služijo kot rezervna varnostna obvestila o položaju spremljanega udeleženca. V primeru izrednih dogodkov se iz poslanih SMS-sporočil lahko rekonstruira pot in ugotovi zadnji uporabnikov položaj. Aktivnost za varnostno sledenje je namenjena vklopu oziroma izklopu intervalnega pošiljanja informacij o položaju na izbrano telefonsko številko.

Aktivnost za varnostno sledenje je izdelana v razredu *SledenjeActivity*, njen uporabniški vmesnik pa v datoteki *activity_sledenje.xml*. Uporabniški vmesnik vključuje gumba za vklop in izklop storitve, vnosno polje za vpis telefonske številke, na katero pošiljamo sporočila, in prikaz položaja uporabnika (Slika 35).

Slika 35: Zaslon za upravljanje z aktivnostjo za varnostno sledenje

V razredu *SledenjeActivity* s klicem metode *startNewService* (Slika 36) zaženemo glavni del modula – storitev za intervalno pošiljanje sporočil, ki poteka v ozadju.

```
public void startNewService(View view) {
    if (phoneEditText.getText().length() == 0){
        Toast.makeText(this, "Prosim vnesite telefonsko številko,
kamor se bodo pošiljale vaše lokacije!", Toast.LENGTH_LONG).show();
    }else {
        startupService();
    }
}
```

Slika 36: Klic metode startNewService

Storitev intervalnega pošiljanja je izdelana v ločenem javanskem razredu *LocalService*. Ko uporabnik vklopi sledenje v ozadju, nastavimo osveževanje položaja v intervalu 15 minut (klic metode *getLocation* na Sliki 37). V primeru pridobitve svežega podatka o položaju platforma izvede povratni klic metode *locationChanged*, ki z rabo (zgoraj že predstavljenega) upravljalnika SMS-sporočil *SmsManager* pošlje SMS na nastavljeno telefonsko številko (Slika 38).

```
gps.getLocation(prefSMSSendInterval * 1000, 0);


gps.addLocationListner(new GPSTracker.ChangedLocation() {
    @Override
    public void locationChanged(Location location, int isGPSFix) {
double latitude = location.getLatitude();
    double longitude = location.getLongitude();
    smsMessage = "https://maps.google.com/maps?q=" + latitude + ",
    " + longitude;// sestavljanje SMS s povezavo do prikaza položaja
na zemljevidu

     counterMaxSMS++; // preverjanje največjega dovoljenega števila
poslanih SMS sporočil
     if (prefSMSSendMax >= counterMaxSMS) {
            sendMessageToUI("SMS send to phone: " + phoneNumber);
            Thread t = new Thread(mUpdateResults); // pošiljanje SMS
            t.start();
        }
    }
});
```

Slika 37: Storitev za sledenje

```
final Runnable mUpdateResults = new Runnable() {
    public void run() {
         SmsManager smsManager = SmsManager.getDefault();
        //pošiljanje sporočila
        smsManager.sendTextMessage
       (phoneNumber, null, smsMessage, null, null);
    }
}
```

Slika 38: Pošiljanje sporočila pri intervalnem pošiljanju sporočil

Nastavitve modula za varnostno sledenje določimo v okviru aktivnosti *SledenjeSetingsActivity*, uporabniški vmesnik smo določili v datoteki *sledenje_settings.xml* (Slika 39).

```
<PreferenceCategory android:title="@string/pref_update_setting" >

<EditTextPreference
    android:title="@string/pref_send_interval"
    android:summary="@string/pref_send_interval_summary"
    android:key="prefSMSSendInterval"
    android:inputType="number"
    android:defaultValue="20"/>

<EditTextPreference
    android:title="@string/pref_max_number_of_sms"
    android:summary="@string/pref_max_number_of_sms_summary"
    android:key="prefSMSSendMax"
    android:inputType="number"
    android:defaultValue="100"/>
```

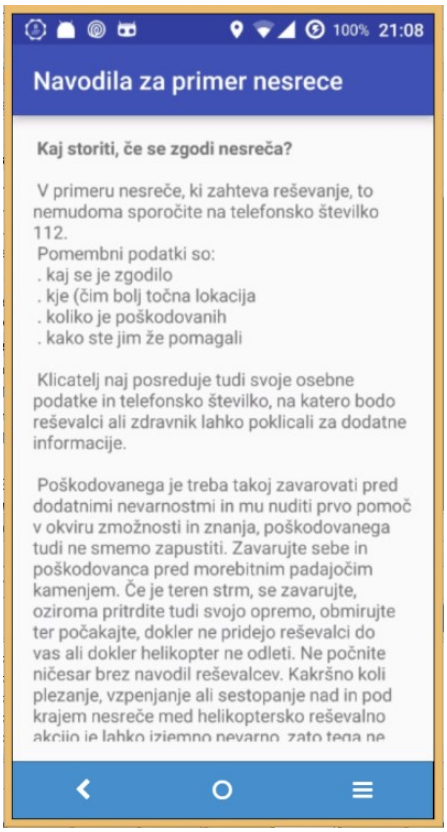
Slika 39: Uporabniški vmesnik aktivnosti za določanje nastavitev

Dostop do nastavitev uporabniku omogočamo prek menija, določenega v datoteki *menu_sledenje.xml*, ki se prikaže v desnem zgornjem kotu uporabniškega vmesnika z modulom za sledenje v ozadju (tri pikice, Slika 35).

Uporabnik lahko dostopa do nastavitev časa intervalnega pošiljanja SMS-sporočil in nastavitve največjega dovoljenega števila poslanih SMS-sporočil. Prav tako lahko dostopa tudi do nastavitev vklopa in izklopa sprejemanja GPS-položaja.

4.2.6. Navodila za ravnanje v primeru nesreče

V primeru nesreče je za planinca nujno upoštevanje določenih navodil, ki morajo biti kratka in jasna [22]. Modul z navodili o ravnanju ob izrednem dogodku (Slika 40) je izveden v aktivnosti *MainCall*, ki jo prikličemo z vstopne strani aplikacije. Uporabniški vmesnik je definiran znotraj datoteke *activity_book.xml* in vsebuje osnovne napotke in informacije o ravnanju v primeru nesreče.



Slika 40: Navodila za ravnanje ob izrednih dogodkih

Z uporabo elementa *ScrollView* (Slika 41) lahko prikažemo daljša besedila, v katerih se pomikamo z drsenjem po zaslonu. Besedilo z navodili je v našem primeru shranjeno v datoteki *slid_book.xml* v mapi *values*.

```
<ScrollView
    android:layout_width="match_parent"
    android:layout_height="match_parent">

<TextView

        android:id="@+id/tv_long"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:text="@string/slid_book_text" />

</ScrollView>
```

Slika 41: Prikaz vsebine z gradnikom ScrollView

4.2.7. Modul z opozorili in informacijami o vremenu v gorah

Modul vsebuje najnovejša opozorila in informacije o gorskem svetu, ki jih črpa s strežnikov ARSO (Slika 42). Opozorila in informacije o vremenu v gorah uporabniku pomagajo pri odločitvi, ali naj se odpravi v gore ali naj počaka na ugodnejšo priložnost. Na ta način z razvito aplikacijo delujemo tudi preventivno.

Slika 42: Opozorila in informacije o stanju v gorskem svetu

Aktivnost z opozorili in informacijami je določena v razredu *MainWeather*, njen uporabniški vmesnik pa v datoteki *activity_weather.xml*. Vsebino prikazujemo v obliki spletne strani z uporabo gradnika *WebView*. Slika 43 prikazuje umestitev gradnika *WebView* v uporabniški vmesnik, Slika 44 pa programsko kodo za prikaz izbrane spletne strani v omenjenem gradniku.

```
<WebView android:id="@+id/webview1"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:layout_weight="1.0" />
```

Slika 43: Uporaba gradnika WebView za prikaz spletne strani v uporabniškem vmesniku

```
webview.loadUrl("http://meteo.arso.gov.si/met/sl/weather/bulletin/mountain/");
    webview.requestFocus(); //odpremo željeno spletno stran
```

Slika 44: Izbira spletne strani za prikaz v gradniku WebView

4.2.8. Potrebna dovoljenja aplikacije

V datoteki *AndroidManifest.xml* moramo določiti različna dovoljenja, ki jih aplikacija potrebuje za svoje delovanje. Ta vključujejo dovoljenja za uporabo interneta, dostop do natančnega položaja naprave, izvedbo klica, ugotavljanje stanja telefona in pošiljanje SMS-sporočil (Slika 45).

```
<uses-permission android:name="android.permission.INTERNET" />
<!--  dovoljenje za dostop do spleta -->
 <uses-permission android:name="android.permission.SEND_SMS" />
 <!--  dovoljenje za pošiljanje SMS -->
 <uses-permission
android:name="android.permission.ACCESS_FINE_LOCATION" />
 <!--  dovoljenje za dostop do GPS  -->
 <uses-permission android:name="android.permission.CALL_PHONE" />
 <uses-permission android:name="android.permission.READ_PHONE_STATE" />
 <!--  dovoljenje za klic -->
```

Slika 45: Dovoljenja za dostop do funkcionalnosti, potrebnih za delovanje aplikacije

4.3. Preizkus aplikacije

Preizkus aplikacije smo izvedli na Šmarni gori, kjer smo s skupino prijateljev preizkusili vse funkcionalnosti, ki jih omogoča. Ob odločitvi, da se odpravimo v gore, smo doma z aplikacijo najprej preverili, ali so za ta dan izdana kakšna posebna opozorila oziroma informacije o stanju v gorskem svetu. Ker jih ni bilo, smo se s prijatelji odločili za pot.

Ob vznožju smo na telefonu vklopili modul za varnostno sledenje. Ob zagonu je GPS-modul začel s pridobivanjem položaja, kar je trajalo nekaj sekund. V modulu za varnostno sledenje smo nato določili številko, na katero bodo pošiljana SMS-sporočila, in nastavili intervalno pošiljanje na 10 minut. Ob vklopu storitve za varnostno sledenje se je na zaslonu prikazalo obvestilo o tem, da je sledenje v ozadju vklopljeno. V nadaljevanju smo se razdelili v dve skupini. Prva skupina se je takoj odpravila na pot, druga pa je počakala 20 minut in nato začela z vzponom.

Na določeni točki vzpona je prvi del skupine, v katerem se bil tudi sam, zašel s poti in simuliral izredni dogodek. Ob tem smo »ugotovili«, da se nam je izpraznila še baterija in da ni mogoče poklicati na pomoč. Na mestu simuliranega izrednega dogodka smo počakali na pomoč prijatelja, ki smo mu v intervalih pošiljali SMS-sporočila z informacijami o položajih točk s poti, po kateri smo hodili. Prijatelj je iz prejetih SMS-sporočil uspešno rekonstruiral zadnji del naše poti in prišel v našo neposredno bližino. Ugotovili smo, da so položaji poslanih SMS v celoti ustrezali prehojeni poti.

Nato se je del skupine odpravil naprej proti vrhu. Čez nekaj časa se je prijatelj »poškodoval« in zaradi zvina noge ni mogel nadaljevati poti. Prek aplikacije smo poklicali v »Center za obveščanje« (v našem primeru prijatelje, ki so čakali na položaju, kjer smo se »izgubili«) in jih prosili za pomoč. Njihovo vprašanje je bilo, kje se je dogodek zgodil, zato smo jim posredovali SMS z informacijami o položaju dogodka.

Med tem, ko smo čakali na prihod »reševalcev«, smo si pogledali še navodila za ravnanje ob nesrečah v gorah. Prijatelji so prek prejetega SMS-sporočila prišli v neposredno bližino lokacije izrednega dogodka in nam pomagali pri vrnitvi v dolino.

4.3.1. Testiranje uporabniškega vmesnika

Uporabniški vmesnik smo ovrednotili na vzorcu 10 oseb. Vse osebe so ugotovile namen posameznih gumbov, pohvalile pa so tudi preglednost in jasnost aplikacije. Vsi sodelujoči so uspeli uporabiti vse funkcije aplikacije brez zapletov in pomoči. Menili so, da bi v primeru odhoda v gore bila to zagotovo aplikacija, ki bi jo uporabili ob izrednem dogodku, priporočili pa bi jo tudi drugim.

4.4. Politika zasebnosti – varovanje podatkov

Zbiranje in uporaba podatkov o položaju uporabnika odpira številna vprašanja o zasebnosti, zato je potrebno v aplikaciji opisati t. i. »politiko zasebnosti«. V okviru slednje opredelimo, na kakšen način aplikacija uporablja geolokacijske podatke, kakšno raven natančnosti dosega in s kom podatke deli. Geolokacijski podatki so v splošnem precej občutljivi, saj razkrivajo uporabnikove položaje, iz katerih lahko rekonstruiramo zgodovino in vzorce uporabnikovega gibanja in obnašanja.

V izdelani aplikaciji so vsi prenosi podatkov od končnega uporabnika varni, saj pošiljanje podatkov poteka samo na številke, ki jih je uporabnik sam določil.

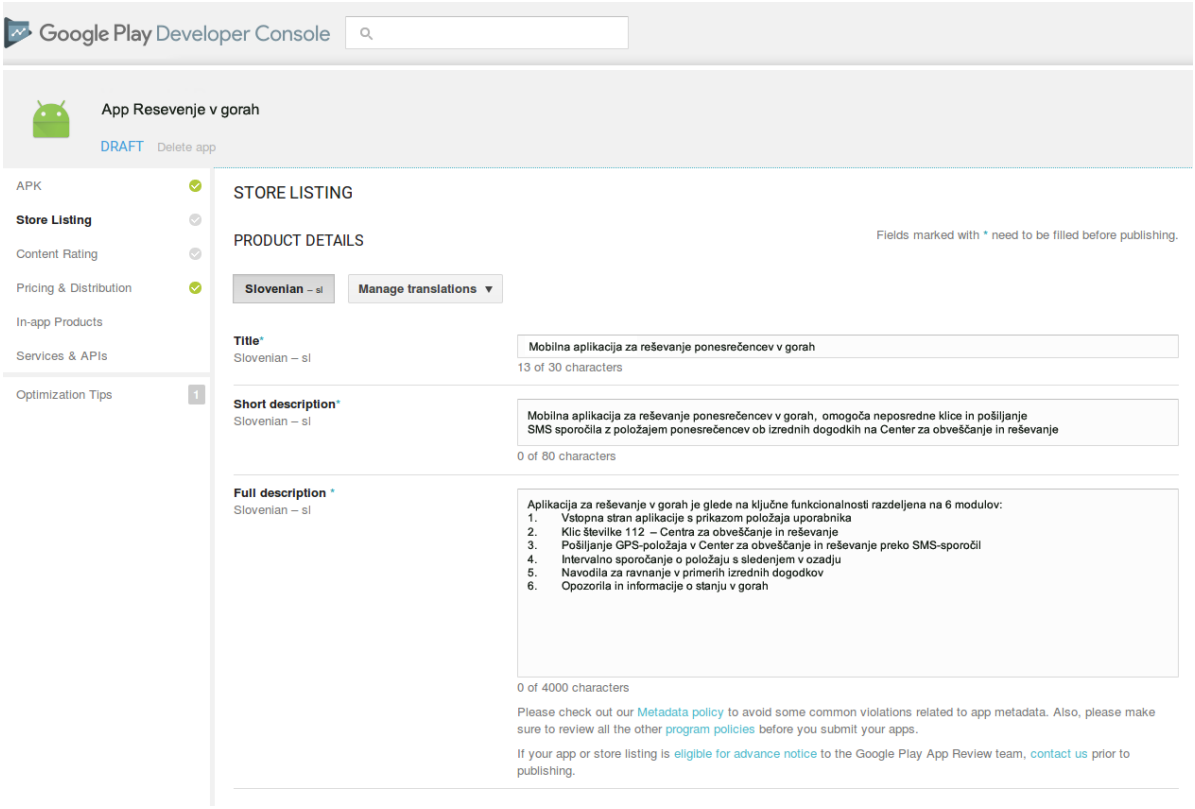
Aplikacija shranjuje poslana SMS-sporočila z informacijami o položaju uporabnika. Prav tako se ta SMS-sporočila shranjujejo tudi pri prijateljih oziroma znancih, ki smo jim jih posredovali po predhodnem dogovoru.

4.5. Google Play

S pomočjo storitve Google Play (nekoč Android Market) lahko razvijalci ponudijo svoje aplikacije uporabnikom naprav z operacijskim sistemom Android, slednji pa lahko hitro, enostavno, brez pomoči in rabe osebnega računalnika nalagajo želene aplikacije. V Google Play trgovini je neposredno z mobilne naprave mogoče brskati po seznamih aplikacij in iger, jih nalagati na mobilno napravo in jih ocenjevati. Uporabniki lahko nekatere aplikacije namestijo brezplačno, druge pa morajo kupiti. Storitev Google Play omogoča samodejne nadgradnje,

kadar razvijalci naložijo novo različico aplikacije, oziroma o obstoju nove različice aplikacije obvestijo uporabnike. Google Play omogoča tudi nadgradnjo storitev in knjižnic za razvijalce aplikacij. Tak primer so Googlovi zemljevidi, ki se osvežujejo oziroma sinhronizirajo po namestitvah novejših različic operacijskega sistema.

Aplikacijo v Google Play lahko dodamo tako, da se najprej prijavimo v račun za razvijalce. Nato aplikacijo naložimo prek konzole za razvijalce Google Play (Slika 46). V meniju izberemo našo aplikacijo, ime aplikacije in privzeti jezik. Aplikacijo naložimo z izbiro: »Nalaganje APK«.



Slika 46: Nalaganje aplikacije, preko konzole za razvijalce

4.6. Možnosti za nadaljnji razvoj aplikacije

Aplikacija ima veliko možnosti za nadaljnji razvoj – naj si bo to z vključevanjem novih funkcionalnosti (tistih že obstoječih, ki se sedaj izvajajo le prek spletne strani Planinske Zveze Slovenije), ali pa tistih, ki še čakajo na digitalizacijo.

Potencialne bodoče funkcionalnosti tako vključujejo:

- Zemljevide z označenimi planinskimi kočami in zavetišči.
- Možnosti »Izgubil sem se« ali »Točka srečanja«, ki omogočata sporočanje položaja brez internetne povezave. – Uporabnik s pomočjo telefona z vgrajenim GPS-modulom pošlje svoj položaj prek SMS-sporočila na izbrano telefonsko številko. Prejemnik uporabi informacijo o sporočenem položaju za navigacijo s pomočjo ene izmed klasičnih navigacijskih aplikacij, kot so: Sygic, Google Maps, Here Maps ali podobnih.
- Seznami opravil (ang. check list). – Aplikacija ima lahko prednaloženih nekaj osnovnih seznamov opravil (kot so seznam stvari, ki jih moramo vzeti s seboj v planine, na pohod, v gore ….). – Vse sezname lahko uporabnik prilagaja po svoje.
- Opozarjanje na nevarne točke na poti, po kateri hodi uporabnik. – Točke lahko v realnem času dopolnjujejo tudi drugi planinci in gorniki.
- Vremenska slika mikrolokacije. – Natančna vremenska slika lahko planincu v gorah zelo veliko pomeni, saj se lahko nenadoma znajde v spremenljivih vremenskih razmerah ali ga ujame tema.
- Vremenski alarm glede na položaj uporabnika. – Na zaslonu se prikaže obvestilo v primeru pričakovane hitre spremembe vremenskih razmer v uporabnikovem območju.
- Možnost »Najdi svoj avto« – za primere, ko gremo v gore po eni, vračamo pa se po drugi poti.

Nadaljnji razvoj aplikacije gre lahko tudi v smeri zahtevnejših funkcionalnosti, kot so: planinske poti, vodiči, elektronska knjižica osvojenih vrhov, georazglednice, članske ugodnosti itd., vendar te v večini zahtevajo večje število razvijalcev in druge dodatne vire.

5. Zaključek

V diplomskem delu smo obravnavali razvoj mobilne aplikacije za reševanje v gorah. V prvem delu diplomske naloge smo se posvetili teoretičnemu spoznavanju operacijskega sistema Android ter sistema GPS, ki ga uporabljamo za natančno določanje položajev na zemeljski obli. V drugem delu smo predstavili razvoj mobilne aplikacije v razvojnemu okolju Android Studio. Na podlagi zahtev za aplikacijo smo razvili uporabniški vmesnik in izbrali ustrezne komponente, nato pa v razvojnem orodju izdelali pet modulov z glavnimi funkcionalnostmi aplikacije.

Rezultat diplomskega dela je mobilna aplikacija za reševanje v gorah, prek katere lahko uporabnik enostavno z SMS-sporočili posreduje podatke o položaju izrednega dogodka na Center za obveščanje. Aplikacija uporabniku omogoča tudi, da lahko s pritiskom na gumb neposredno pokliče na številko 112. Ena od glavnih funkcionalnosti je modul za sledenje v ozadju, prek katerega sorodnikom ali prijateljem v SMS-sporočilih pošiljamo podatke o položaju v določenih intervalih, ki jih le-ti lahko posredujejo v Center za obveščanje v primeru izrednega dogodka oz. izgube stika s ponesrečencem, s katerimi je mogoče rekonstruirati pot, ki jo je uporabnik prehodil. V primeru izrednega dogodka lahko uporabnik v aplikaciji prebere tudi navodila ob nesrečah. Za preventivo je na spletni strani ARSO pred odhodom v gore mogoče preveriti vreme in opozorila za gorski svet.

Aplikacijo smo testirali tudi na terenu. Sklepna ugotovitev je, da so se vse funkcionalnosti v praksi odlično obnesle. »Reševalne ekipe« so v obeh primerih prispele v našo neposredno bližino in nas brez težav našle, prav tako pa smo si med čakanjem na »reševalce« prebrali navodila o ravnanju ob nesrečah. Preden smo se odpravili v gore smo si ogledali tudi vremensko napoved in opozorila za gorski svet.

Mobilna aplikacija za pomoč pri reševanju ponesrečencev v gorah lahko zelo veliko doprinese k večji varnosti in boljšemu dostopanju do ponesrečencev v gorah. Za uporabnike bo v osnovi povsem brezplačna in brez oglaševalskih vsebin.

Nadaljnji razvoj aplikacije bi lahko šel v smeri dodajanja funkcionalnosti, kot so zemljevidi s položaji planinskih koč, možnosti sporočanja o točkah srečanj s sopotniki (v primeru, ko smo

zašli s poti), ali v razvoj zahtevnejših funkcionalnosti, kot so knjižice s potrditvami o doseženih vrhovih, vodiči po planinskih poteh, georazglednice, članske ugodnosti itd.

Seveda si nadaljnjega razvoja aplikacije in njene širše uporabe ne predstavljamo brez sodelovanja z organizacijami, kot sta Planinska zveza Slovenije (PZS) ali Gorska reševalna služba. Skupni cilj pri razvoju aplikacije pa je, da le-ta s pomočjo partnerjev resnično postane aplikacija, ki bi jo v primeru odhoda v gore uporabilo čim več planincev. Naš končni cilj je, da aplikacija postane vrsta standarda (podobno kot je to varnostni pas, s katerim se pred vožnjo pripnemo v avtomobilu).

Literatura in viri

[1]. Predstavitev Planinske zveze Slovenije, 2014. Dosegljivo na: http://www.pzs.si/javno/mediji/Novinarska_konf-10-6-2014/PZS-Predstavitev_Planinske_zveze_Slovenije-junij2014.pdf
[Dostopano: 20. 3. 2016]

[2] Pregled statistike reševalnega dela: Janez Kosec, 2011. Dosegljivo na: http://www.grzs.si/ftp/NoviceGRZS/KINFO/2011/analiza/Statistika-resevalnega%20dela-2010-Kosec-18-1-2011.pdf
[Dostopano: 20. 3. 2016]

[3] Do lokacije z SMS-sporočilom: Gašper Završnik, 2014. Dosegljivo na: http://www.zurnal24.si/do-lokacije-z-sms-sporocilom-clanek-230289
[Dostopano: 10.4. 2016]

[4] Android OEM profitability and the most surprising number from Q4's smartphone market, 2015. Dosegljivo na: https://theoverspill.wordpress.com/2015/02/09/android-oem-profitability-and-the-most-surprising-number-from-q4s-smartphone-market/
[Dostopano: 30. 4. 2016]

[5] Current GPS Location, 2012. Dosegljivo na: https://play.google.com/store/apps/details?id=com.cbsb.sendyourlocation
[Dostopano: 30. 3. 2016]

[6] My GPS location, 2015. Dosegljivo na: https://play.google.com/store/apps/details?id=com.sktapp.sharemylocation
[Dostopano: 30. 3. 2016]

[7] GPS Location Tracker, 2016. Dosegljivo na: https://play.google.com/store/apps/details?id=com.triladroid.locationshare
[Dostopano: 30. 3. 2016]

[8] Family Locator - GPS Tracker, 2016. Dosegljivo na:
https://play.google.com/store/apps/details?id=com.sygic.familywhere.android
[Dostopano: 30. 3. 2016]

[9] Family Locator - Life360, 2016. Dosegljivo na:
https://play.google.com/store/apps/details?id=com.life360.android.safetymapd
[Dostopano: 30. 3. 2016]

[10] Smartphone OS Market Share, 2015 Q2. Dosegljivo na:
http://www.idc.com/prodserv/smartphone-os-market-share.jsp
[Dostopano: 30. 3. 2016]

[11] Marshmallow distribution figures (and older Android versions), 2016. Dosegljivo na:
http://forums.androidcentral.com/android-6-0-marshmallow/639507-marshmallow-distribution-figures-older-android-versions-feb-2016-a.html
[Dostopano: 10. 4. 2016]

[12] Introduction to Android development with Android Studio: Lars Vogel, 2015.
Dosegljivo na:
http://www.vogella.com/tutorials/Android/article.html
[Dostopano: 10.3. 2016]

[13] Android Activity. Dosegljivo na:
http://developer.android.com/reference/android/app/Activity.html
[Dostopano: 30. 3. 2016]

[14] Developer Workflow Basics. Dosegljivo na:
https://developer.android.com/studio/workflow.html
[Dostopano: 10. 4. 2016]

[15] Svetovni geodetski sistem 84, 2011. Dosegljivo na:
http://code7700.com/wgs84.html
[Dostopano: 10. 5. 2016]

[16] GPS/AVL v slovenski policiji: Albin Gradišnik, Miha Ristič, 2011. Dosegljivo na:
http://www.fvv.um.si/dv2011/zbornik/policijska_dejavnost/gradisnik-ristic.pdf
[Dostopano: 20. 4. 2016]

[17] Kako pametni telefon določi tvojo lokacijo? Boštjan Kop, 2015 Dosegljivo na: http://bostjankop.eu/kako-pametni-telefon-doloci-tvojo-lokacijo/
[Dostopano: 25.4. 2016]

[18] GNSS (Global Navigation Satellite Systems), 2012-2016. Dosegljivo na: http://xenon.colorado.edu/spotlight/index.php?action=kb&page=27
[Dostopano: 20. 3. 2016]

[19] Koliko je satelita potrebno za određivanje lokacije pomoću GPS uređaja? Predrag Supurović, 2009. Dosegljivo na: http://pedja.supurovic.net/koliko-je-satelita-potrebno-za-odredivanje-lokacije-pomocu-gps-uredaja/
[Dostopano: 10. 4. 2016]

[20] Marko Čulibrk. Diplomska naloga – Spletne storitve na Windovs mobile in Android, 2010. Dosegljivo na: https://dk.um.si/Dokument.php?id=15559
[Dostopano: 10. 4. 2016]

[21] SmartLocator. Dosegljivo na: http://www.xlab.si/products/smart-locator/
[Dostopano: 30. 3. 2016]

[22] Gorski policisti bodo na planinskih poteh skrbeli za varen obisk naših gora, 2015 Dosegljivo na: http://www.policija.si/index.php/component/content/article/35-sporocila-za-javnost/78785-gorski-policisti-bodo-na-planinskih-poteh-skrbeli-za-varen-obisk-naih-gora?tmpl=component&print=1&page=&lang=
[Dostopano: 10. 4. 2016]

[23] Uporaba GPS sledilnih naprav in varstvo osebnih podatkov, 2010. Dosegljivo na: https://www.ip-rs.si/fileadmin/user_upload/Pdf/smernice/GPS_smernice_net.pdf
[Dostopano: 20. 4. 2016]