Sustav za geolociranje dezorjentiranih starijih osoba

Nikolašević, Sara

Master's thesis / Diplomski rad

2019

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Pula / Sveučilište Jurja Dobrile u Puli**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:137:785894

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-28**

Image not found or type unknown

Repository / Repozitorij:

Digital Repository Juraj Dobrila University of Pula

zir.nsk.hr

Image not found or type unknown

Sveučilište Jurja Dobrile u Puli

Fakultet Informatike u Puli

SARA NIKOLAŠEVIĆ

SUSTAV ZA GEOLOCIRANJE DEZORIJENTIRANIH STARIJIH OSOBA

Diplomski rad

Pula, rujan, 2019.

Sveučilište Jurja Dobrile u Puli

Fakultet Informatike u Puli

SARA NIKOLAŠEVIĆ

SUSTAV ZA GEOLOCIRANJE DEZORIJENTIRANIH STARIJIH OSOBA

Diplomski rad

JMBAG: 0303038038, redoviti student

Studijski smjer: Diplomski sveučilišni studij informatike

Predmet: Izrada informatičkih projekata

Znanstveno područje: Informacijske i komunikacijske znanosti

Mentor: doc.dr.sc. Siniša Sovilj

Pula, rujan, 2019.

IZJAVA O AKADEMSKOJ ČESTITOSTI

Ja, dolje potpisana Sara Nikolašević, kandidat za magistra informatike ovime izjavljujem da je ovaj Diplomski rad rezultat isključivo mojega vlastitog rada, da se temelji na mojim istraživanjima te da se oslanja na objavljenu literaturu kao što to pokazuju korištene bilješke i bibliografija. Izjavljujem da niti jedan dio Diplomskog rada nije napisan na nedozvoljen način, odnosno da je prepisan iz kojega necitiranog rada, te da ikoji dio rada krši bilo čija autorska prava. Izjavljujem, također, da nijedan dio rada nije iskorišten za koji drugi rad pri bilo kojoj drugoj visokoškolskoj, znanstvenoj ili radnoj ustanovi.

Student

_____

U Puli, _____, _____ godine

IZJAVA

o korištenju autorskog djela

Ja, Sara Nikolašević dajem odobrenje Sveučilištu Jurja Dobrile u Puli, kao nositelju prava iskorištavanja, da moj diplomski rad pod nazivom „Sustav za geolociranje dezorijentiranih starijih osoba“ koristi na način da gore navedeno autorsko djelo, kao cjeloviti tekst trajno objavi u javnoj internetskoj bazi Sveučilišne knjižnice Sveučilišta Jurja Dobrile u Puli te kopira u javnu internetsku bazu završnih radova Nacionalne i sveučilišne knjižnice (stavljanje na raspolaganje javnosti), sve u skladu s Zakonom o autorskom pravu i drugim srodnim pravima i dobrom akademskom praksom, a radi promicanja otvorenoga, slobodnoga pristupa znanstvenim informacijama.

Za korištenje autorskog djela na gore navedeni način ne potražujem naknadu.

U Puli, _____, _____ godine

Potpis

_____

SVEUČILIŠTE JURJA DOBRILE U PULI ODJEL ZA INFORMACIJSKO-KOMUNIKACIJSKE TEHNOLOGIJE

Pula, 1. ožujka 2018.

DIPLOMSKI ZADATAK

Pristupnik:	Sara Nikolašević (0303038038)
Studij:	Sveučilišni diplomski studij Informatike
Naslov (hrv.):	Sustav za geolociranje dezorijentiranih starijih osoba.
Naslov (eng.):	System for geolocation of disoriented elderly people.
	Opis zadatka: Zadatak je razviti sustav za geolociranje dezorijentiranih starijih osoba te notifikaciju i alarmiranje skrbnika u slučaju problema. Sustav se treba sastojati od izrade mobilne aplikacije za prikaz lokacije praćene osobe u okruženju Android, web sustava za prikaz lokacije praćene osobe i administriranje u okruženju Laravel te od sklopovskog dijela koji se sastoji od izrade prototipa GPS-GSM modula koji će se koristiti za slanje lokacije praćene osobe (Arduino). U sklopu istraživačkog dijela diplomskog rada usporediti mogućnost korištenja i drugih sklopovskih rješenja za geolociranje.

Mentor:

_____

doc.dr.sc. Siniša Sovilj

UVOD 1
1.Opis sustava 2
2. Tehnologije korištene u razvoju sustava 7
2.1. Android Studio 7
2.2. Java 9
2.3. FCM (Firebase Cloud Messaging) 10
2.4. Arduino 11
2.5. SIM808 13
2.6. PHP 14
2.7. Laravel 15
3. Razvoj sustava za geolociranje dezorijentiranih starijih osoba 18
3.1. Prototip GPS uređaja 18
3.2. Izrada mobilne aplikacije 24
3.3 Web sustav 29
4. Prikaz sučelja web aplikacije 36
4.1. Registracija u sustav 36
4.2. Registracija dezorijentirane osobe 37
4.3. Prijava u sustav 38
4.4. Prikaz lokacije na mapi 39
5. Prikaz sučelja mobilne aplikacije 40
5.1. Prijava u mobilnu aplikaciju 41
5.2 . Pregled lokacije 42
5.3. Ažuriranje podataka dezorijentirane osobe 44
5.4. Pregled statusa GPS uređaja 45
6. Usporedba drugi sklopovskih rješenja za geolociranje 46
6.1. LoRa i LoRaWAN 47
ZAKLJUČAK 50
Literatura 51
Popis slika 53
Sažetak 54
Summary 55

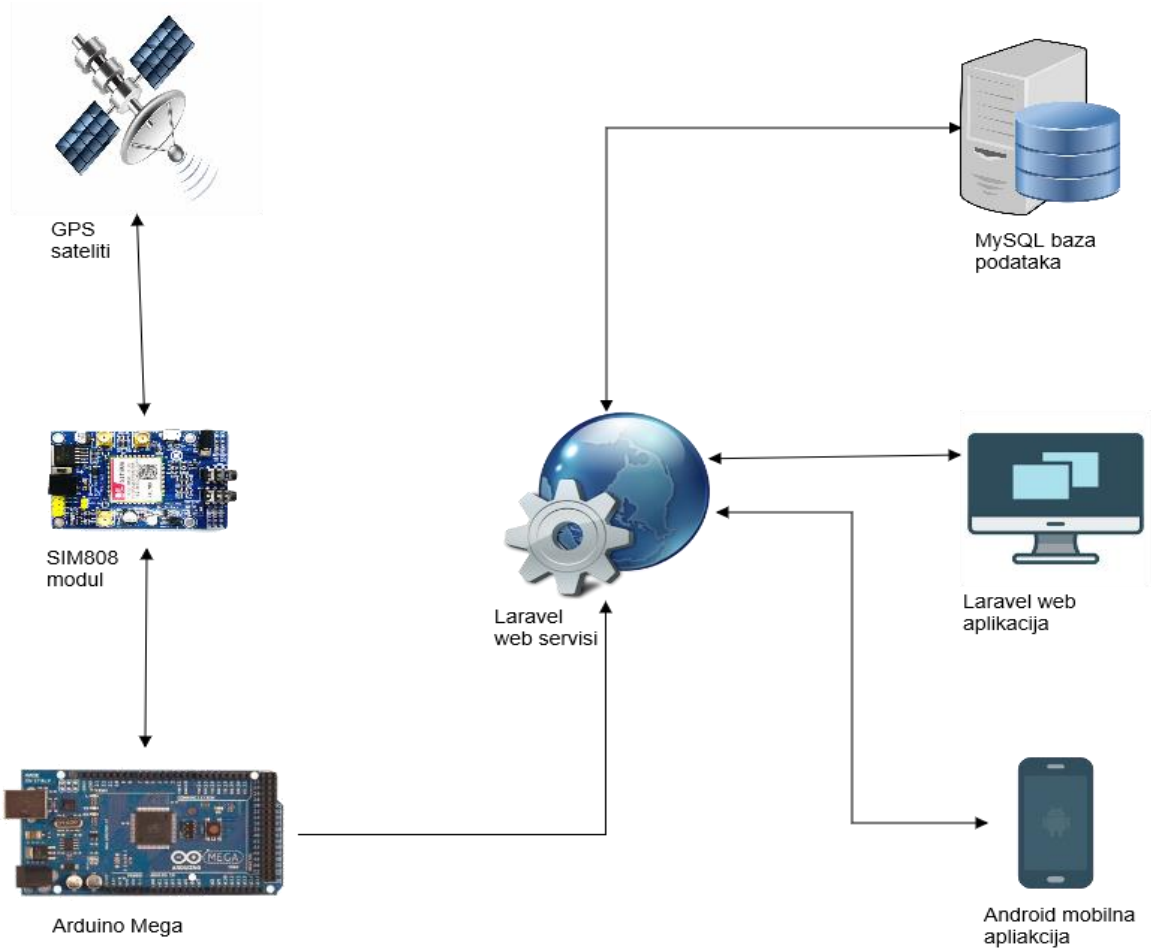
UVOD

Tema ovog rada je izrada sustava za geolociranje dezorijentiranih starijih osoba. Zadatak je razvoj sustava koji se sastoji od tri glavne komponente, a to su prototip GPS - GPRS uređaja, mobilna aplikacija i web aplikacija. Svaka od tri navedene komponente ima svoj zadatak. Zadatak GPS - GPRS uređaja je slanje trenutne lokacije praćene osobe, zadatak mobilne i web aplikacije je prikaz trenutne lokacije praćene osobe uz oznaku lokacije na mapi. Uz prikaz lokacije, glavni zadatak mobilne aplikacije je obavještavanje korisnika o izlasku praćene osobe iz zadanog radijusa dok se registracija korisnika i praćene osobe vrši kroz web aplikaciju. Ovaj sustav rješava problem dezorijentacije dementnih osoba s kojim se često susreću ljudi koji su odgovorni za dezorijentirane osobe, bilo da je riječ o Alzheimerovoj demenciji ili nekom drugom obliku demencije. Dementne osobe se često udalje sa mjesta prebivališta te se upute u nepoznatom smjeru, a ovaj bi sustav trebao pomoć odgovornim osobama da što efikasnije i brže pronađu dezorijentiranu osobu.

U prvom dijelu ovog rada biti će ukratko opisane tehnologije koje su korištene u razvoju ovog sustava. Drugi dio rada opisuje koja komponenta sustava koristi koje od navedenih tehnologija. U nastavku rada je prikazano sučelje sustava sa uputama za uporabu namijenjenu krajnjim korisnicima te usporedba drugih sklopovskih rješenja za geolociranje.

1.Opis sustava

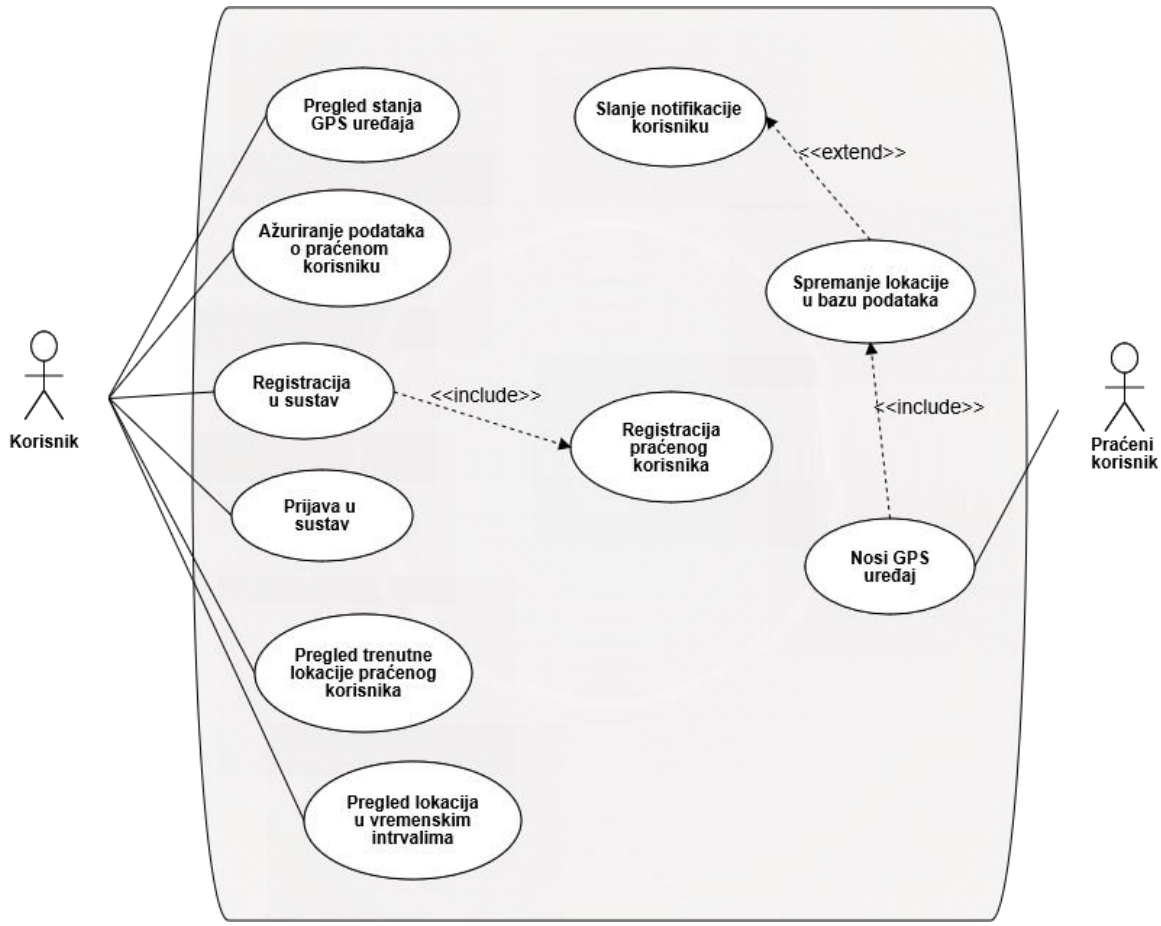
Ovaj sustav je namijenjen praćenju starijih dezorijentiranih osoba. Sustav se sastoji od prototipa GPS uređaja, mobilne aplikacije i web aplikacije. Osoba koja koristi mobilnu i web aplikaciju je korisnik koji je odgovoran za dezorijentiranu osobu. Dezorijentirana osoba koristi GPS uređaj. Glavna značajka i zadaća sustava je dohvaćanje lokacije dezorijentirane osobe. Ti podaci o lokaciji se pohranjuju u bazu podataka te im se pristupa kroz mobilnu ili web aplikaciju kako bi se lokacija dezorijentirane osobe prikazala na mapi. Nakon uspješne registracije u sustav korisnik mora unijeti podatke o osobi kojoj je GPS uređaj namijenjen kao što su ime, prezime, kućna adresa, grad, država te radijus. Prilikom pohrane podataka o dezorijentiranoj osobi u bazu podataka, kućna adresa se pretvara u koordinate odnosno geografsku širinu i dužinu te se u tom formatu spremaju i predstavljaju lokaciju kućne adrese. Radijus je broj koji predstavlja koliko se metara od kućne lokacije dezorijentirana osoba može udaljiti. Prilikom dohvaćanja informacija o trenutnoj lokaciji računa se udaljenost između kućne lokacije i trenutne lokacije. Ukoliko se u nekom trenutku dogodi da je udaljenost od kućne lokacije veća od zadanog radijusa, korisniku mobilne aplikacije se šalje obavijest o udaljavanju iz radijusa s trenutnom adresom. Korisnik može provjeriti korisnikovu trenutnu lokaciju unutar mobilne aplikacije ili preko web aplikacije u bilo kojem trenutku. Ukoliko se pojavi potreba za promjenom informacija o dezorijentiranoj osobi, bilo da se radi o promjeni kućne adrese, grada, države ili pak dopuštene udaljenosti od kućne adrese, korisnik je u mogućnosti to napraviti unutar mobilne aplikacije ili web aplikacije.



Slika 1. Komponente sustava za geolociranje dezorijentiranih osoba

Izvor: izradio autor.

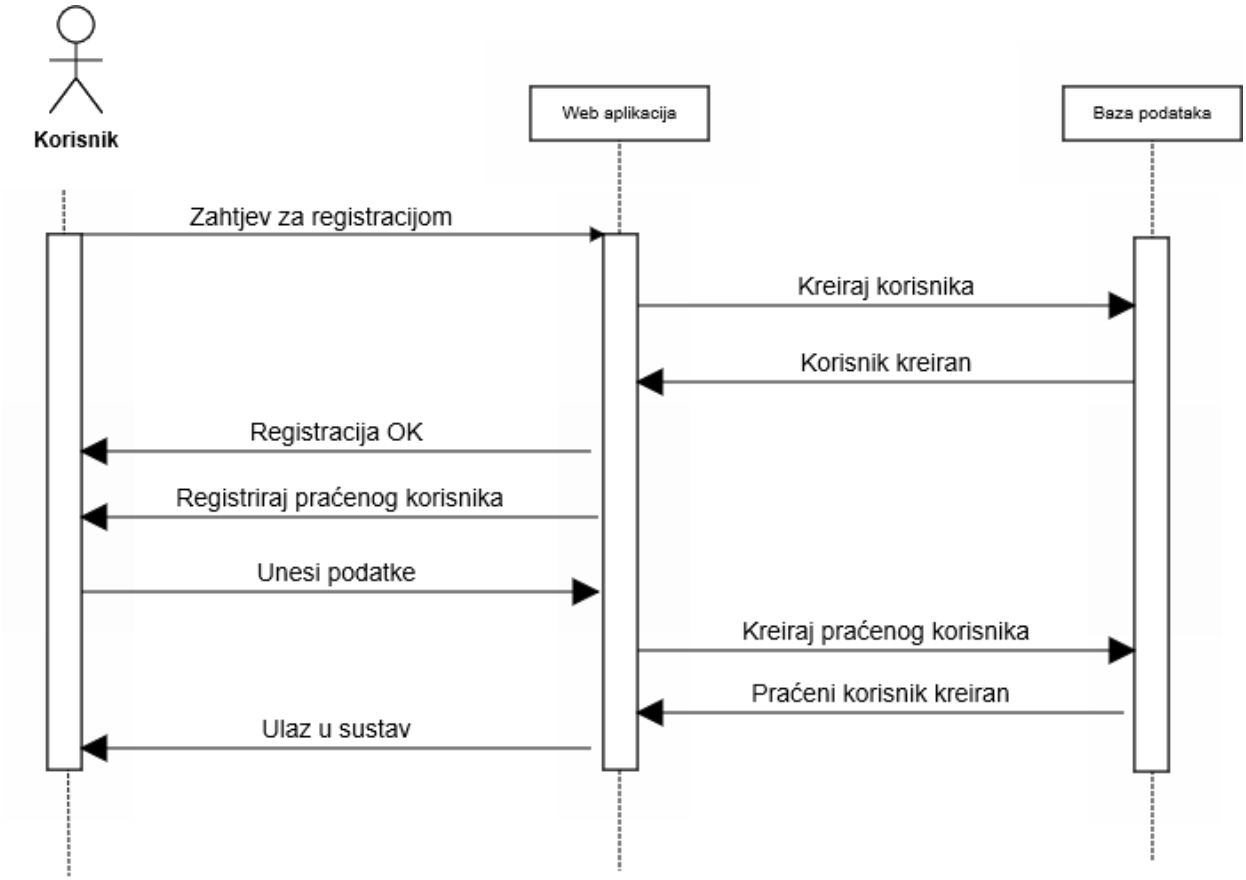
Na slici 1 su prikazane komponente od kojih se sustav sastoji i koje su potrebne kako bi sustav funkcionirao te tijek prikupljanja i pohrane lokacije. Prototip GPS sustava napravljen je spajanjem SIM808 modula sa Arduino Mega mikro kontrolerom. SIM808 prikuplja GPS signale od GPS satelita koji kruže oko Zemlje dok Arduino dohvaća te podatke te ih šalje na web. Zatim se ti podaci spremaju u bazu podataka u tablicu lokacija u obliku geografske dužine i širine. Mobilna aplikacija i web aplikacija korištenje Laravel web servisa upravljaju bazom podataka, odnosno prikupljaju podatke koji su potrebni za prikaz te ažuriraju ukoliko je potrebno. Podatke o lokaciji koriste mobilna aplikacija i web aplikacija za prikaz lokacije dezorijentirane osobe na mapi.



Slika 2. Dijagram korištenja sustava (eng. use-case)

Izvor: izradio autor

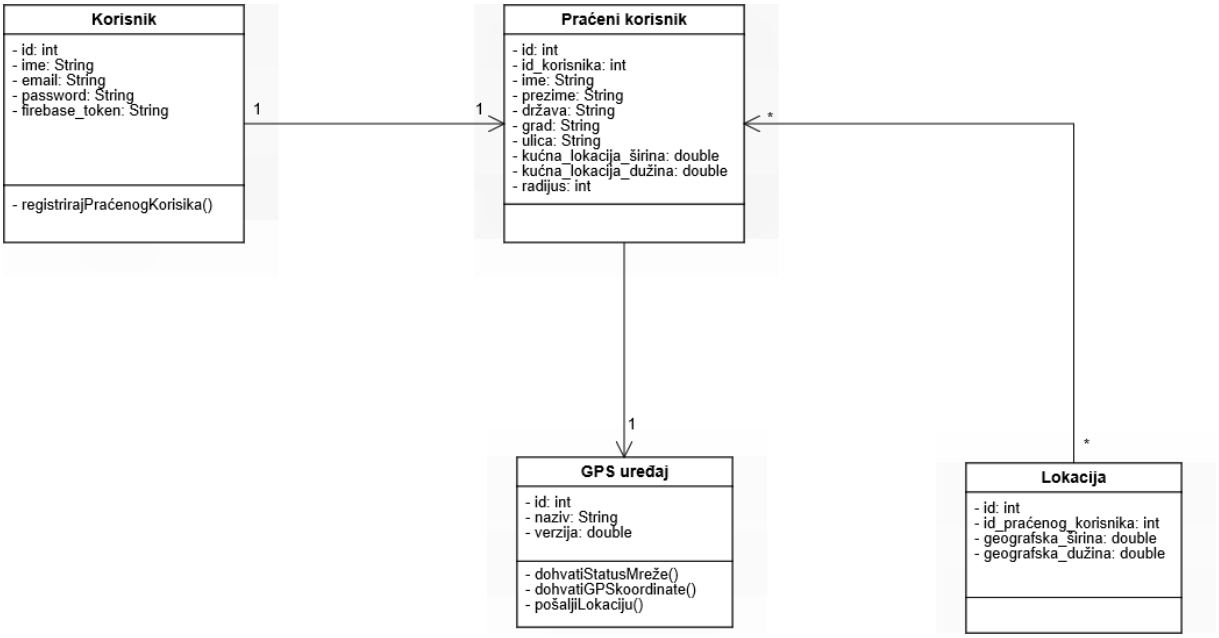
Slika 2 prikazuje dijagram korištenja sustava, aktere koji koriste sustav te mogućnosti sustava. Iz dijagrama vidimo da sustav koristi korisnik, te praćena osoba. Praćena osoba je dezorijentirana osoba koja nosi GPS uređaj. GPS uređaj uključuje dohvaćanje lokacije te slanje iste na web kako bi se spremila u bazu podataka. Slanje lokacije na web opcionalno uključuje slanje notifikacije korisniku. Mogućnosti koje korisnik ima su registracija u sustav što uključuje i registraciju praćenog korisnika, prijava u sustav, ažuriranje podataka o praćenom korisniku, pregled trenutne lokacije, pregled lokacije u vremenskim intervalima te pregled stanja GPS uređaja.



Slika 3. Sekvencijski dijagram

Izvor: Izradio autor

Sekvencijski dijagram prikazan na slici 3 prikazuje postupak registracije u sustav vremenskim redoslijedom izvršavanja aktivnosti. Prvi korak je zahtjev za registracijom, zatim se korisnik kreira unutar baze podataka. Nakon što je novi korisnik registriran u sustav potrebno je unijeti podatke za praćenog korisnika. Nakon registracije praćenog korisnika i podaci o njemu su pohranjeni u bazi podataka korisnik može pristupiti sustavu.



Slika 4. Klasni dijagram sustava

Izvor: Izradio autor

Iz klasnog dijagrama sa slike 4, možemo vidjeti osnovne klase koje su korištene u sustavu te njihove atribute. Na slici vidimo da svaki korisnik ima svoj id koji predstavlja specifičan identifikacijski broj svakog korisnika, ime, email i password koji unosi prilikom registracije u sustav. Zadatak korisnika je registrirati praćenog korisnika koji također ima identifikacijski broj te id korisnika koji ga je registrirao kako bi se mogli povezati praćeni korisnici s korisnicima. Za praćenog korisnika se također unosi ime, prezime, država, grad, ulica te radijus. Jedan praćeni korisnik može biti povezan na točno jednog korisnika i može imati registriran točno jedan GPS uređaj. Svaki GPS uređaj ima također svoj identifikacijski broj te naziv i verziju. Klasa lokacija je klasa koja sadrži sve pohranjene lokacije te ima svoj id, id praćenog korisnika kako bi se pojedina lokacija mogla povezati s točnim praćenim korisnikom te geografsku dužinu i širinu koje predstavljaju točnu lokaciju.

2. Tehnologije korištene u razvoju sustava

Za izradu sustava korištene su različite tehnologije i različita razvojna okruženja. Za izradu mobilne aplikacije korišten je Android operacijski sustav te Java kao programski jezik i Android Studio kao razvojno okruženje. Za izradu web aplikacije korišten je Laravel okvir, te za izradu sklopovskog rješenja korišten je Arduino za slanje lokacije za pohranu u bazu podataka, te SIM808 VB3 2.2 modul za dohvaćanje trenutne lokacije koristeći GPS antenu.

2.1. Android Studio

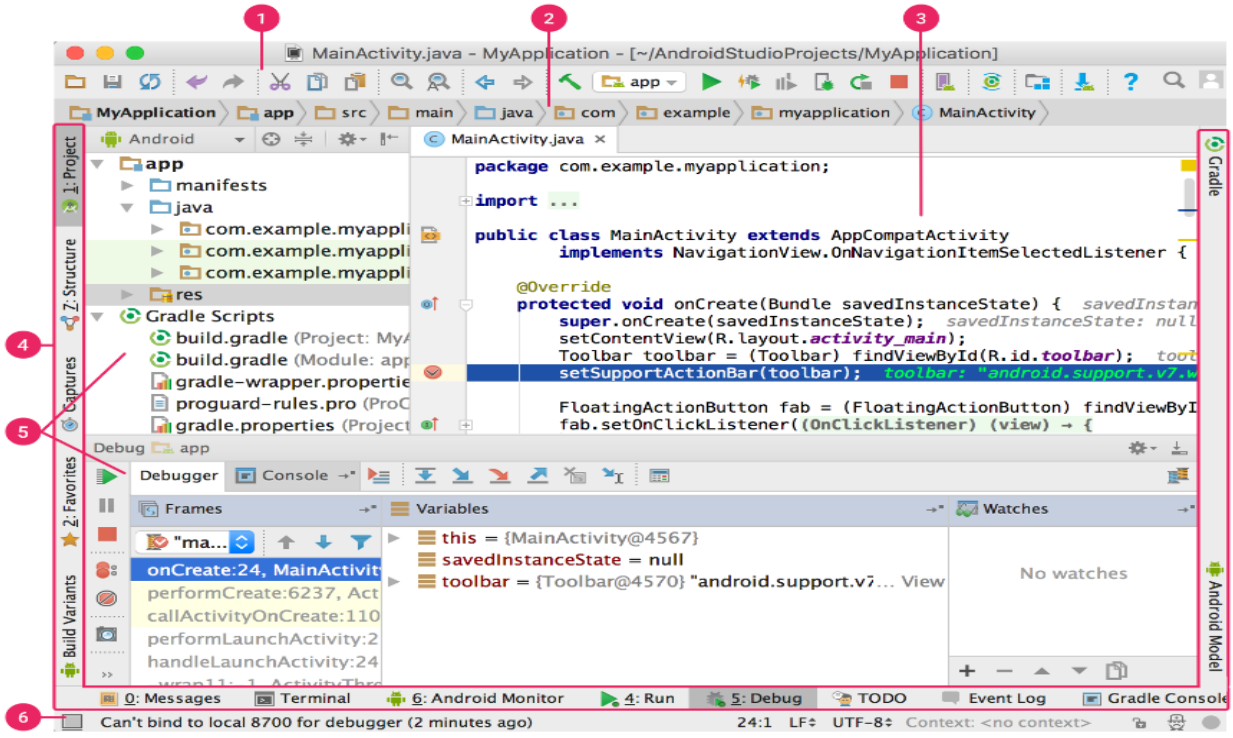
Android Studio je baziran na IntelliJ IDEA-i, službeno integrirano razvojno okruženje za Javu koje je razvio JetBrains, prvobitno poznat kao IntelliJ. Android Studio je dizajniran specifično za Android stoga je i službeno razvojno okruženje za razvoj Android-a. Moguće ga je koristiti na Windows operacijskom sustavu, macOS i Linux operacijskom sustavu. Prva verzija službeno je bila najavljena 15. Svibnja 2016. godine, a prethodnik mu je bio Eclipse Android Development Tools, primarno integrirano razvojno okruženje za razvoj Android aplikacija.

Android Studio sadrži mnogo značajki koje olakšavaju razvoj i povećavaju produktivnost prilikom izrade Android aplikacija. Neke od tih značajki su:

- Podrška za nativno programiranje
- Ugrađena podrška za Google Cloud Platform
- Ujedinjeno okruženje u kojem je moguć razvoj za sve Android uređaje
- Fleksibilan sustav izgradnje zasnovan na Gradle-u
- Brz imitator android uređaja (eng. emulator)
- Instant Run kako bi se izbjegla potreba za ponovnom izgradnjom novog APK-a nakon napravljenih promjena za aktivnu aplikaciju
- GitHub integracija

Korisničko sučelje Android Studio-a izgrađeno je od nekoliko elemenata koje možemo vidjeti na slici 5, a to su:

1. Alatna traka preko koje se obavljaju mnogi zadaci kao što su pokretanje aplikacije, pokretanje otklanjanja nepravilnosti te otvaranje Android alata.
2. Navigacijska traka koja nas vodi kroz projekt i prikazuje otvorene datoteke koje se uređuju.
3. Prozor za uređivanje u koje se piše i modificira kod.
4. Alatna traka prozora koja se pokreće izvan prozora integriranog razvojnog okruženja, a sadrži gumbe koji nam omogućavaju proširenje ili skupljanje pojedinih alatnih prozora.
5. Prozor alata preko kojeg pristupamo specifičnim zadacima kao što je pretraga, kontrola nad projektom i ostalo a također se može proširiti ili skupiti.
6. Traka statusa koja prikazuje status projekta te poruke i upozorenja vezana uz projekt i integrirano razvojno okruženje.



Slika 5. Glavni prozor Android Studio-a

Izvor: https://developer.android.com/studio/intro

2.2. Java

Java je objektno orijentiran programski jezik koji je razvio Sun Microsystems 1991. godine. Izvorni naziv Jave bio je Oak a bio je dizajniran za razvoj softvera za potrošačke elektroničke uređaje kao što su televizori i videorekorderi. Java je prvi put službeno objavljena 1995.godine, a nastala je iz želje da se razvije programski jezik koji će biti jednostavan prenosiv i pouzdan, što su razvojni programeri Jave smatrali glavnim nedostatcima u jezicima kao što su C i C++. Tako su uklonili brojne značajke iz C i C++ koje su smatrali izvorima problema i modelirali Javu. Osim pouzdanosti, jednostavnosti i prenosivosti, željeli su da Java bude jednostavna, kompaktna i interaktivna. Sun Microsystems službeno opisuje Javu s sljedećim značajkama (Balagurusamy, 2007):

- Objektno orijentirana. Skoro sve u Javi je objekt, odnosno cijeli programski kod i podaci unutar programa su smješteni unutar klasa i objekata. Java ima svoj skup klasa koje su smještene unutar paketa te ih se po potrebi može koristiti unutar projekta.
- Robusna i sigurna. Java ima sakupljač smeća (eng. garbage-collector) koji olakšava programerima gotovo sve probleme upravljanja memorijom te uključuje koncept rukovanja iznimkama (eng. Exceptions) koji bilježi pogreške u programskom kodu te rješava rizik od pada sustava. Java provjerava pristupe memoriji, a s obzirom da nema pokazivača, osigurava da program ne može pristupiti memorijskim mjestima bez odgovarajuće autorizacije.
- Distribuirana. Java ima mogućnost distribuiranja podataka i programa. Java aplikacije imaju mogućnost otvaranja i pristupanja objektima na internetu kao i u lokalnom sustavu, a to omogućava da više programera s različitih lokacija zajedno surađuju i rade na jednom projektu.
- Ne ovisna o platformi i prenosiva. Ovo je jedan od najznačajnijih atributa iz razloga što Java se programi mogu jednostavno premještati s jednog računalnog sustava na drugi. Promjene i ažuriranja na razini operacijskog sustava ne zahtijevaju promjene Java programa.
- Jednostavna i poznata. Uklanjanje nekih elemenata iz C i C++ kao što su pokazivači, goto mogućnost i mnogi drugi učinilo ju je jednostavnijom i manjom.

Atribut poznata predstavlja to da je Java modelirana na osnovu C i C++-a što znači da ukoliko programer piše kod u C++-u tada će mu i kod napisan u Javi biti poznat s obzirom na to da je Java u stvari pojednostavljena verzija C++-a.

- Višeslojna i interaktivna. Java omogućuje pisanje više dretvenih programa. Višedretvenost je mogućnost izvršavanja više zadataka istovremeno.
- Dinamična i proširiva. Unutar Jave se jednostavno mogu povezivati nove biblioteke klasa, metoda i objekata. Programi napisani u Javi podržavaju nativne metode, odnosno metode koje su napisane u drugim jezicima kao što su C ili C++.
- Kompajlirana i interpretirana. Java funkcionira na način da prvo kompajler prevodi izvorni kod u bajtkod, a zatim, s obzirom na to da bajtkod nije razumljiv stroju, interpreter generira strojni jezik koji se tada može direktno izvoditi na stroju koji je pokrenuo Java program.

Okruženje za razvoj Java programa uključuje mnogo razvojnih alata te klasa i metoda. Razvojni alati su dio softverskog razvojnog okruženja koji je poznat pod nazivom JDK (eng. Java Development Kit). JDK uključuje i JRE (eng. Java Runtime Environment) skup softverskih alata za razvoj koji kombinira JVM (eng. Java Virtual Machine), temeljne klase platforme te biblioteke podrške. JDK uključuje i interpreter-a (java), kompajler-a (javac), arhiv (jar), generator dokumentacije (javadoc) i ostale alate koji su potrebni za razvoj.

2.3. FCM (Firebase Cloud Messaging)

Firebase Cloud Messaging je prvobitno bilo poznato kao Google Cloud Messaging, a predstavlja rješenje za slanje poruka i notifikacija te se trenutno koristi bez ikakve naplate. Korištenjem FCM-a moguće je pouzdana i učinkovita veza između poslužitelja i uređaja. FCM omogućuje primanje i isporuku poruka i notifikacija na Android i iOS uređaje te na web aplikacije kroz pouzdanu i učinkovitu vezu između poslužitelja i uređaja. FCM ima neke značajke koje obavještavanje korisnika čine jednostavnijom i točnijom kao što su:

- Sa FCM-om je moguće isporučiti poruke odmah ili u neko buduće vrijeme ovisno o lokalnoj vremenskoj zoni u kojoj se primatelj nalazi.

- Moguće je slanje prilagođenih podataka, postavljanje prioriteta, zvukova te datume isteka.
- Poruke obavijesti su u potpunosti integrirane sa Google Analytics za Firebase što omogućuje pristup detaljnom praćenju angažmana te praćenje učinkovitosti sa jedne nadzorne ploče bez potrebe za kodiranjem.
- A/B testiranje poruka obavijesti omogućuje slanje više verzija testnih poruka kako bi se odabrala ona koja najviše odgovara cilju.
- Napredno ciljanje poruka koristeći predefinirane segmente ili kreiranjem vlastitih koristeći demografske podatke.

Kako bi se mogla koristiti usluga slanja i primanja poruka i notifikacija potrebno je dodati Firebase u projekt te se registrirati u Firebase koristeći Google račun.

2.4. Arduino

Arduino je elektronička platforma otvorenog koda. Arduino pločica može biti unaprijed sastavljena prilikom kupnje ili se može samostalno izgraditi s obzirom na to da je dizajn hardvera otvorenog koda. U srcu svake Arduino pločice nalazi se mikro kontroler. Mikro kontroler je samostalan integrirani strujni krug koji sadrži mikro procesor, memoriju za čitanje podataka, memoriju slučajnog pristupa i razne ulazno izlazne sabirnice. Da bi se pločica mogla programirati na računalu je potrebno instalirati integrirano razvojno okruženje za Arduino koji nosi isti naziv kao i elektronička pločica. Arduino pločice mogu čitati ulazne podatke na primjer poruka, te pretvoriti u izlazni podatak na primjer text na ekranu spojenom na pločicu (Hoffman, 2018). Arduino pločica je povezana na računalo putem USB kabla te se na taj način učitava kod sa Arduino softvera na pločicu. Komunikacija sa Arduinom se odvija slanjem skupa instrukcija mikro kontroleru koji se nalazi na pločici. Za komunikaciju sa Arduinom koristi se Arduino programski jezik koji je zasnovan na Wiring - u, te integrirano razvojno okruženje zasnovano na Processing – u.

Arduino programski jezik je jezik koji se koristi za komunikaciju sa pločicom a može biti podijeljen u tri glavna dijela, funkcije, varijable i konstante te strukture. Wiring je programski okvir za mikro kontrolere koji omogućuje pisanje softvera koji mogu biti

korišteni neovisno o platformi i okruženju. Softveri napisani u Wiring-u služe za kontrolu uređaja koji su priključeni na širok raspon mikro kontrolera.

Arduino integrirano razvojno okruženje je softver otvorenog koda koji olakšava pisanje koda koji se kasnije učitava na pločicu. Može se pokrenuti na Windows operacijskom sustavu kao i na Mac OS X i Linux - u, a zasnovan je na Processing-u i ostalim softverima otvorenog koda. Processing je fleksibilna softverska skica i jezik za učenje kodiranja u kontekstu vizualne umjetnosti.

Osim Arduino integriranog razvojnog okruženja moguće je koristiti i alternativna rješenja. Jedna od alternativa je PlatformIO alat koji se može pokrenuti na operacijskim sustavima kao što su Linux, Max OSX i Windows. To je alat za pisanje koda i upravljanje bibliotekama koji je baziran na Python programskom jeziku, a izvršava se kroz komandnu liniju. Osim PlatformIO alata postoji i Ino alat za izgradnju Arduino programa koji se također bazira na Python programskom jeziku. (Hughes, 2016).

Arduino pločice imaju prednost u odnosu na ostale elektroničke pločice zbog sljedećih značajki:

- Prihvatljiva cijena - Arduino pločice su relativno jeftine u odnosu na ostale elektroničke pločice.
- Arduino softver je neovisan o platformi, može se instalirati na Windows, Macintosh OSX i Linux operacijski sustav dok je većina ograničena na Windows operacijski sustav.
- Jednostavno razvojno okruženje.
- Softver je otvorenog koda uz mogućnost nadogradnje.
- Hardver otvorenog koda uz mogućnost nadogradnje.

Postoji više vrsta Arduino elektroničkih pločica. Svaka pločica ima svoje značajke a pločice koje su predviđene za početak učenja programiranja elektroničkih Arduino pločica su Arduino UNO, Arduino Leonardo, Arduino 101, Arduino Esplora, Arduino Micro, Arduino Nano i MKR2UNO adapter. Pločice koje spadaju u skupinu pločica naprednih funkcionalnosti i bržih performansi su Arduino Mega 2560, Arduino Zero, Arduino Due,

Arduino ISP, Arduino Motor Shield, Arduino M0 PRO i mnogi drugi. Arduino pločice mogu se koristiti i za Internet Stvari, tako postoji čitava kolekcija pločica, a neke od njih su Arduino Yun, Arduino Ethernet, Arduino Industrial 101 i mnogi drugi. Pločica koja se smatra najboljim odabirom za početak učenja programiranja elektroničkih pločica je Arduino Uno, dok je Arduino Mega 2560 vrlo čest odabir za projekte gdje su potrebne napredne funkcionalnosti ili brže performanse.

2.5. SIM808

SIM808 je GSM/GPRS modul koji kombinira GPS tehnologiju za navigaciju uz pomoć satelita. Modul ima kompaktan dizajn s integriranim GPRS-om i GPS-om u SMT paketu. Modul ima utor za SIM karticu te je na taj način omogućena komunikacija, odnosno primanje i slanje poruka i poziva te konekcija na internet koristeći mikro kontroler. Na samoj pločici je konektor za GPRS antenu te GPS antenu. Uz pomoć GPS antene se mogu dohvatiti različite informacije o lokaciji. Kontrola SIM808 modul-a te slanje instrukcija moguće je koristeći AT komande. Zadaci vezani uz GPS koji se mogu izvršavati slanjem AT komandi SIM808 modul-u su sljedeće:

- Upravljanje napajanja GPS, odnosno uključiti i isključiti GPS napajanje slanjem AT+CGPSPWR skupa komandi.
- Vraćanje postavki za GPS slanjem AT+CGPSRST skupa komandi.
- Dohvaćanje trenutnih informacija o GPS-u slanjem AT+CGPSINF skupa komandi.
- Dohvaćanje informacija o statusu odnosno pozicioniranju GPS-a slanjem AT+CGPSSTATUS skupa komandi.

Neki od zadataka vezanih uz SMS koji se mogu izvršavati slanjem AT komandi su sljedeći:

- Brisanje SMS poruke slanjem AT+CMGD komande.
- Slanje SMS poruke slanjem AT+CMGS komande.
- Čitanje SMS poruke slanjem AT+CMGR komande.
- Odabir formata SMS poruke slanjem AT+CMGF komande.
- Zapis SMS poruke u memoriju slanjem AT+CMGW komande.

- Mnogi drugi kao što su spremanje postavki SMS-a, slanje SMS poruke iz memorije, vraćanje postavki za SMS itd.

Zadaci vezani uz Internet HTTP uslugu koji se mogu izvršavati slanjem AT komandi su sljedeće:

- Inicijalizacija HTTP usluge slanjem At+HTTPINIT komande.
- Prekid HTTP usluge slanjem AT+HTTPTERM komande.
- Postavljanje HTTP parametara slanjem AT+HTTPPARA komande.
- Postavljanje ulaznih podataka slanjem AT+HTTPDATA komande.
- Postavljanje HTTP akcije metode korištenjem AT+HTTPACTION komande.
- Čitanje HTTP odgovora poslužitelja slanjem AT+HTTPREAD komande.
- Spremanje HTTP aplikacijskog konteksta slanjem AT+HTTPSCONT komande.
- Čitanje HTTP statusa slanjem AT+HTTPSTATUS komande.
- Čitanje HTTP informacija zaglavlja odgovora poslužitelja slanjem AT+HTTPHEAD komande.

2.6. PHP

PHP(rekurzivni akronim za PHP: Hypertext Preprocessor) je skriptni jezik koji se generalno koristi za razvoj aplikacija na web-u, a može biti ugrađen u HTML te na taj način statička HTML stranica postaje dinamička. PHP skripta se izvršava na poslužitelju, generirajući HTML kod koji se zatim šalje klijentu. Nakon pokretanja PHP skripte klijent prima rezultat ali bez ikakvog znanja o temeljnom kodu. Kako bi bili sigurni da temeljni kod ne dođe do krajnjeg korisnika moguće je konfigurirati web poslužitelja da obradi sve HTML datoteke s PHP-om. PHP se koristi za generiranje dinamičkog sadržaja stranice, slanje i primanje kolačića (eng. cookies), prikupljanje podataka iz različitih obrazaca i mnogo više. Čest je odabir kod početnika ali i kod naprednih programera zbog svoje fleksibilnosti i jednostavnosti. Može ga se koristiti na raznim operacijskim sustavima kao što su Microsoft Windows, Linux, neke Unix varijante, macOS te RISC OS. Također postoji podrška za mnoge web servere kao što su Apache, IIS i mnogi drugi. PHP nije ograničen na generiranje isključivo HTML-a već je moguće generiranje slika, PDF

datoteka, XHTML i XML datoteka te formiranje pred memorije na poslužitelju kako bi se pohranile datoteke spremljene u datotečni sustav. Najbitnija značajka PHP-a je podrška za širok raspon baza podataka kao što su MySQL, SQLite, SQLite3, MongoDB i mnoge druge te podrška za druge servise koristeći protokole kao što su HTTP, COM, IMAP, SNMP LDAP i drugi. Jedno od područja na kojima se koristi PHP je programiranje na strani poslužitelja koje je i najpopularnije. Područja na kojima se koristi PHP su razvoj desktop aplikacija, skriptiranje na strani poslužitelja te skriptiranje preko komandne linije. Područje se odabire ovisno o potrebama, međutim najčešće je to skriptiranje na strani poslužitelja kod kojeg nam je potrebno PHP parser, poslužitelj te web preglednik, a rezultatu programa se pristupa preko web preglednika pregledavajući PHP stranicu koja je pohranjena na poslužitelju. (Tatroe i MacIntyre, 2020).

2.7. Laravel

Laravel je web razvojni okvir otvorenog koda baziran na PHP-u, odnosno Simfony okviru, a razvio ga je Taylor Otwell. Trenutno se smatra jednim od najpoznatijih PHP razvojnih okvira. Ovaj okvir nastao je s ciljem da omogući razvoj web aplikacija koristeći MVC (eng. Model-View-Controller) arhitekturu. Izvorni kod Laravel-a se nalazi na GitHub-u pod MIT licencom. Laravel ima mnogo značajki zbog kojih se smatra najpopularnijim okvirom, a neke od njih su ugrađeni predlošci za izradu izgleda stranice, MVC arhitektura, sigurnost, objektno relacijsko mapiranje, Artisan komandni alat te sustav migriranja baze podataka. Ovaj okvir dolazi sa gotovim predlošcima za izgled stranice koji su dizajnirani za stvaranje jednostavnih ali i složenih izgleda stranica s karakterističnim dijelovima. Jednostavan je za korištenje zbog arhitekture koja ima mnogo gotovih i ugrađenih funkcija koje povećavaju sigurnost te poboljšavaju performanse aplikacije. Objektno relacijsko mapiranje u Laravelu omogućuje slanje upita na bazu podataka koristeći PHP sintaksu umjesto uobičajene SQL sintakse. Sigurnost Laravel web okvira smatra se vrlo jakom. Za spremanje lozinka u bazu podataka koriste se mehanizmi za šifriranje tako da lozinka nikad ne može biti spremljena u bazu podataka u obliku običnog teksta, a za generiranje šifrirane lozinke koristi Bcrypt algoritam za šifriranje.

Prva verzija, odnosno beta verzija pojavila se početkom lipnja 2011. godine, a Laravel 1 koji je bio i prva službena verzija izašao je krajem lipnja iste godine. Prve tri verzije bazirane su na istom izvornom kodu sa dodavanjem određenih značajki. Laravel 1 je uključivao ugrađena rješenja za ovjeravanje autentičnosti, lokalizaciju, sesije i usmjeravanje (eng. routing) no arhitektura modela pogleda kontrolera nije bila potpuna s obzirom da je postojala podrška za modele i poglede no ne i za kontrolere. Podrška za kontrolere dolazi tek sa sljedećom verzijom Laravel 2 u studenom 2011. godine. Osim kontrolera, u drugoj verziji dodali su sustav predložaka Blade, programski princip inverzije kontrole te je ukinuta podrška bibliotekama i paketima trećih strana. Treća verzija, odnosno Laravel 3 službeno je objavljen u veljači 2012. godine, a značajke koje su dodali su Artisan sučelje naredbenog retka, podršku za više sustava upravljanja bazama podataka, upravljanje izvornim kodom te sustav pakiranja tzv. Bundles. (Stauffer, 2019).

Laravel 4 nazvan je Illuminate, a izašao je u svibnju 2013. godine. Autori su odlučili da će nova verzija Laravel-a biti ponovno napisana iz temelja te će predstavljati skup komponenti ili paketa koji međusobnom integracijom tvore razvojni okvir. S ovom verzijom Laravel-a uveden je Composer, koji služi za upravljanje paketima i komponentama. Osim upravitelja paketa dodano je i inicijalno kreiranje baze podataka te punjenje baze testnim podacima, mogućnost reda poruka, ugrađeni paketi za slanje e-pošte te brisanje zapisa iz baze koristeći “soft delete” metodu. Nakon Illuminate verzije objavljena je verzija 4.1 u svibnju 2014. godine sa dodanom SSH komunikacijom, a Laravel 4.2 objavljen je u rujnu iste godine sa dodanim podrškama za upravljanje PHP cloud serverima tzv. Laravel Forge te upravljanje pretplatama tzv. Laravel Cashier. (Stauffer, 2019)..

Laravel 5 objavljen je u veljači 2015. godine a poboljšanja su nova struktura mapa, kreiranje pred memorije rute koje drastično ubrzava aplikacije koje koriste više od 100 ruta, upravljački program za redove baze podataka, Laravel Socialite paket koji omogućuje ovjeravanje autentičnosti koristeći OAuth te Flysystem integracija. (Stauffer, 2019). Sljedeća verzija je Laravel 5.1 objavljena u lipnju 2015. godine a predstavlja prvu verziju ikad objavljenu koja uključuje dugotrajnu podršku (eng. LTS) koja podrazumijeva podršku ispravljanja grešaka u trajanju tri godine. Laravel 5.2 objavljen je u prosincu 2015. godine sa podrškom za Multi-Auth poboljšanom ovjeravanje autentičnosti, implicitno

vezivanje modela, poboljšanja validacije nizova te grupe posredničkih softvera. Laravel 5.3 objavljen je u rujnu 2016. godine a značajke koje su dodali su Laravel Passport nativni OAuth poslužitelj za Laravel aplikacije, Laravel Scout upravljački program temeljen na pretraživanju teksta, Laravel Mailable novi pojednostavljeni pristup slanju e-pošte te Laravel Notifications koji omogućuje slanje brzi ažuriranja putem usluga kao što su Slack, tekstualne poruke, e-pošta i drugo, Laravel Echo koji je poboljšanje postojećeg sustava emitiranja događaja koji olakšava rad s web socket-ima i druge. Sljedeća verzija je Laravel 5.4 koji je objavljen u siječnju 2017. godine, a najznačajnija poboljšanja su Laravel Dusk alat za testiranje preglednika, Laravel Mix koji predstavlja novu generaciju Elixir-a i dodatna fleksibilnost kod Blade predložaka. Laravel 5.5 objavljen je u rujnu 2017. godine te spada u sljedeću verziju s podrškom ispravljanja grešaka (eng. LTS) s podrškom za ispravljanje grešaka u trajanju od dvije godine te sigurnosne ispravke u trajanju od tri godine dok generalna manja izdanja imaju podršku ispravljanja grešaka u trajanju od šest mjeseci i sigurnosne ispravke u trajanju jedne godine. Laravel 5.6 objavljen je u veljači 2018. godine te nadograđuje poboljšanja napravljena u Laravel 5.5 verziji dodavanjem poboljšanog sustava za bilježenje, poboljšanja procesa identifikacije serijskog formata modela, raspoređivanje zadataka nad jednim poslužiteljem, poboljšanja formatiranja datuma, Argon2 podrška za šifriranje lozinke i drugo. Laravel 5.7 objavljen je u rujnu 2018. godine te nadograđuje poboljšanja napravljena u prethodnoj verziji, a obuhvaća opcionalnu potvrdu e-pošte, podršku za korisnike “goste” na autorizacijskim prolazima i pravilima, poboljšanja u testiranja konzole, integracija Sympfony dump poslužitelja, lokalizirane obavijesti i niz drugih ispravaka grešaka. Laravel 5.7 je trenutno zadnja službeno objavljena verzija, a sljedeća verzija Laravel 6 predviđa se u kolovozu 2019. godine.

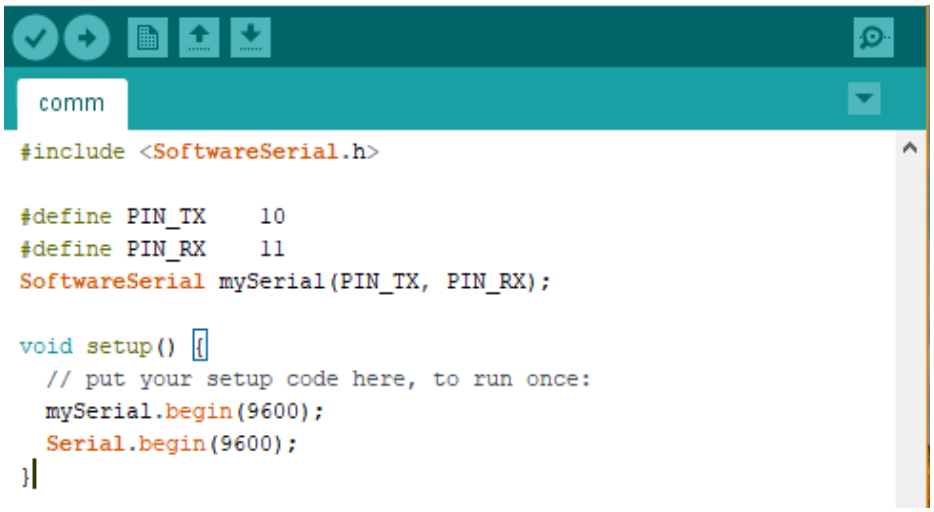
3. Razvoj sustava za geolociranje dezorijentiranih starijih osoba

U ovom poglavlju biti će opisana implementacija prototipa GPS uređaja, mobilne aplikacije te Web sustava. Cilj razvoja ovog sustava je praćenje kretanja željenih, uglavnom starijih osoba te prikaz tih lokacija na mobilnom uređaju kroz mobilnu aplikaciju ili kroz Web aplikaciju. Mobilna aplikacija namijenjena je korisnicima Android uređaja te je iz tog razloga odabran Android Studio kao razvojno okruženje za mobilnu aplikaciju te Laravel razvojno okruženje za Web aplikaciju. Kako bi bio moguć prikaz lokacije praćene osobe potreban je GPS uređaj.

Globalni položajni sustav (eng. Global Positioning System - GPS) sastoji se od mreže satelita koji kontinuirano šalju kodirane informacije pomoću kojih je moguće precizno određivanje položaja na Zemlji. GPS uređaj određuje svoj položaj primajući radio signale koje mu odašilju sateliti koji kruže oko Zemlje. Položaj se zapisuje u obliku koordinata, gdje je jedan broj zemljopisna dužina te definira udaljenost točke od nultog meridijana prema istoku ili zapadu, a drugi broj je zemljopisna širina te definira udaljenost točke od ekvatora prema sjeveru ili jugu.

3.1. Prototip GPS uređaja

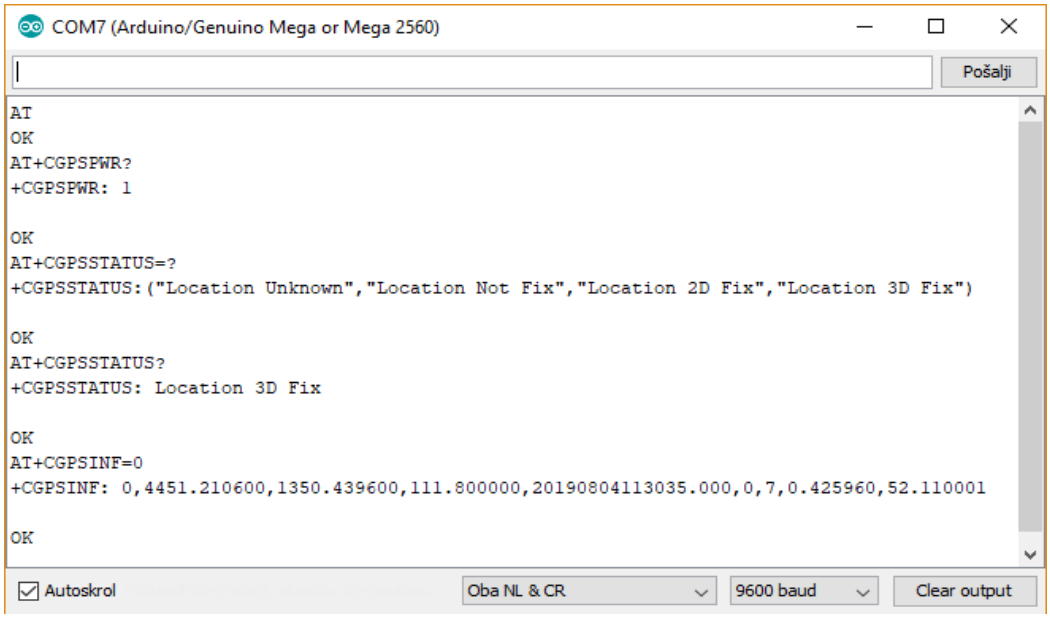
Za prototip GPS uređaja korišten je SIM808 GSM/GPRS modul. Modul ima utor za SIM karticu te također podršku za TCP/IP protokol te je na taj način moguće razmjenjivanje podataka, odnosno slanje podataka o lokaciji u bazu podataka. Informacije o lokaciji se dohvaćaju preko GPS antene koja je priključena na površini elektroničkog sklopa. Platforma koja se koristi za komunikaciju sa SIM808 modulom je Arduino Mega. Za prototip je odabrana ta platforma zato što je vrlo dobro dokumentirana i cijenom pristupačna te najpristupačnija platforma u trenutku izrade sustava. Kodiranje Arduino elektroničke pločice uspostavljeno je preko Arduino IDE softvera. Kako bi bilo moguće slanje instrukcija SIM808 modul-u sa Arduino MEGA pločice bilo je potrebno uspostaviti komunikacijske kanale odnosno povezati pinove za primanje i slanje podataka sa Arduina na SIM808 modul i obrnuto. Nakon što se fizički spoje komunikacijski kanali potrebno ih je inicijalizirati unutar Arduino IDE softvera te učitati kod na Arduino pločicu kako bi se mogla uspostaviti komunikacija.



Slika 6. Inicijalizacija komunikacije unutar Arduino IDE

Izvor: Izradio autor

Nakon uspostavljanja komunikacijskih kanala moguće je slanje instrukcija SIM808 modul-u korištenjem AT komandi. Za testiranje komunikacije između Arduina i SIM808 modula koristi se serijski monitor preko kojeg simuliramo slanje instrukcija SIM808 modul-u. Kroz serijski monitor šaljemo instrukcije koje su namijenjene izvršavanju određenih zadataka kao što je slanje tekstualnih poruka, dohvaćanje lokacije preko GPS antene, dohvaćanje stanja o napajanju modul-a, stanje računa na SIM kartici te inicijalizacija HTTP protokola. Kako bismo bili sigurni da je komunikacija između Arduina i SIM808 modul-a dobro uspostavljena šalje se AT komanda te ukoliko dobijemo odgovor OK komunikacijski kanali su uspješno spojeni i inicijalizirani.

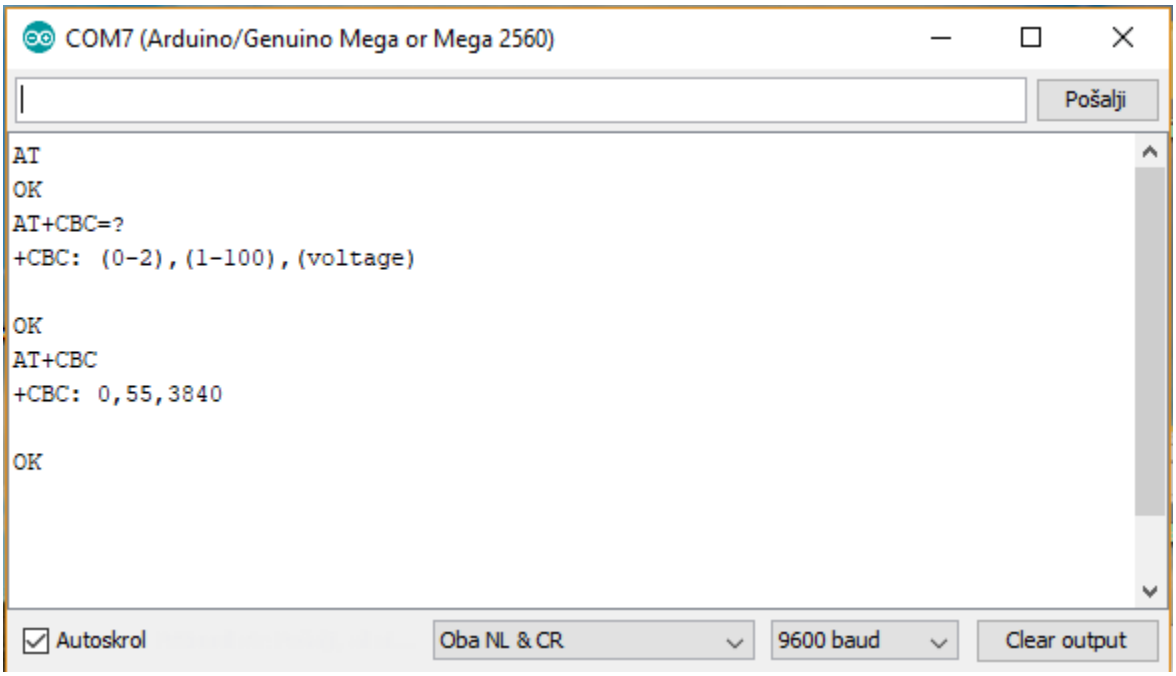


Slika 7. Slanje instrukcija za GPS

Izvor: izradio autor

Na slici 7 vidimo uspostavljanje komunikacije između Arduina i SIM808 modul-a preko serijskog monitora. Prva komanda koja je poslana je AT komanda te odgovor “OK” znači da je komunikacija uspješna te možemo nastaviti slati željene komande. Prva komanda za GPS koja je poslana “AT+GPSPWR?” predstavlja upit o statusu napajanja GPS antene koja je priključena na SIM808 modul. Odgovor koji možemo dobiti je 0 ukoliko je GPS antena ugašena ili 1 ukoliko je GPS antena aktivna. Ako dobijemo odgovor da je GPS antena ugašena tada ju je, prije sljedećih komandi o lokaciji, potrebno aktivirati slanjem komande “AT+CGPSPWR=1”. Nakon što je GPS antena aktivna moguće je zatražiti informacije o statusu te informacije o lokaciji. Komanda “AT+CGPSSTATUS?” vraća status GPS uređaja. Ukoliko dobijemo odgovor “Location Unknown” znači da GPS antena nije uključena, “Location Not Fix” znači da je GPS uključen ali nije još dohvatio informacije o lokaciji, odnosno nije pozicioniran. Statusi “Location 2D Fix” i “Location 3D Fix” predstavljaju mjere točnosti položaja, 2D mjera treba minimalno 3 satelita za dohvatiti dužinu i širinu dok 3D mjera treba minimum 4 satelita za dohvaćanje širine, dužine te visine. Posljednja komanda koja je prikazana na Slika 2 je “AT+CGPSINF=0” te odgovor na te instrukcije su informacije posljednje uspješne GPS lokacija, gdje je prva vrijednost odgovora vrsta informacije koju smo postavili u komandi “AT+CGPSINF=0” te je to

najčešće 0 za dohvaćanje dužine i širine. Vrijednost lokacije dohvaćamo iz druge i treće vrijednosti iz odgovora gdje je dužina vrijednost na drugoj, a širina vrijednost na trećoj poziciji u odgovoru. Osim geografske dužine i širine iz odgovora možemo pročitati vrijednost geografske visine, vrijeme i datum, TTFF (eng. Time to first fix), odnosno vrijeme u sekundama u kojem je uhvaćena prva informacija o lokaciji, broj satelita u trenutku dohvaćanja lokacije, brzina nad zemljom te tijek iznad zemlje. Nakon što nam informacije o lokaciji više nisu potrebne možemo ugasiti napajanje za GPS slanjem komande “AT+CGPSPWR=0”.

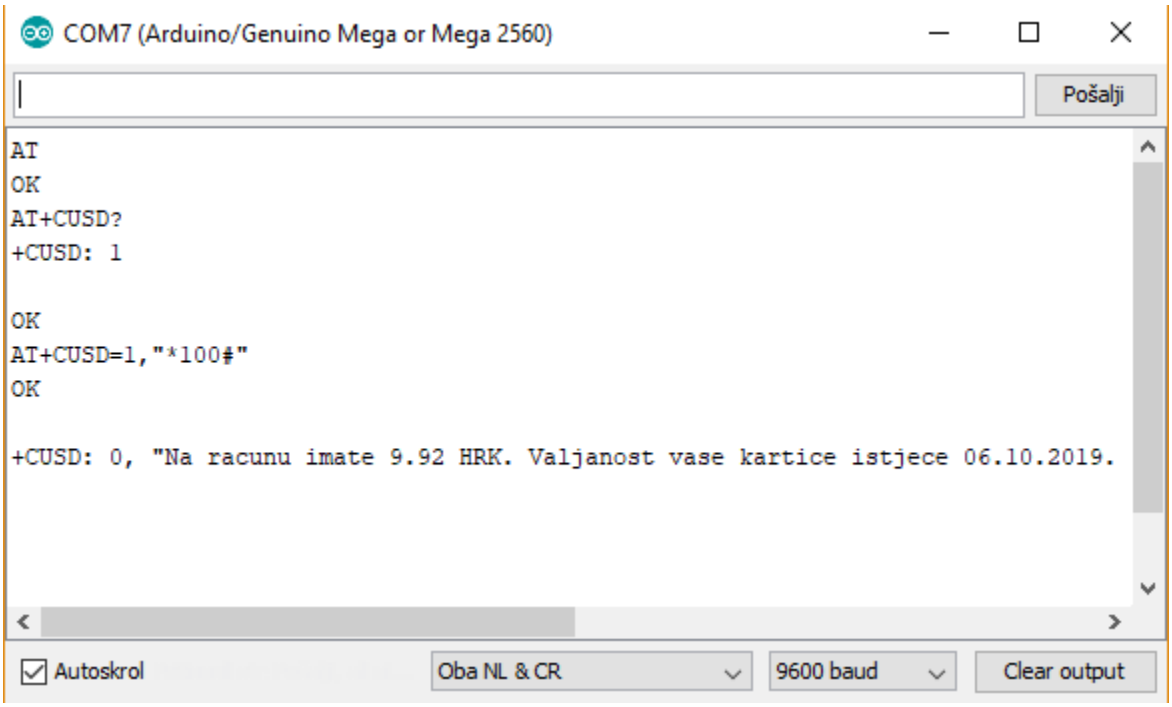


Slika 8. Slanje instrukcija za bateriju

Izvor: Izradio autor

Na slici 8 je prikazano slanje instrukcija za dohvaćanje stanja baterije. Kao i obično prvo se šalje AT komanda da provjerimo status, ukoliko je status “OK” komunikacija s SIM808 modul-om je moguća. Slanjem “AT+CBC=?” komande dobivamo odgovor koji predstavlja status napajanja uređaja. Prvi odgovor je niz brojeva od 0 do 2, a predstavlja listu podržanih načina napajanja. Odgovor 0 označava da se uređaj ne napaja, 1 označava da se uređaj napaja i 2 označava da je napajanje završeno. Drugi odgovor je niz brojeva od

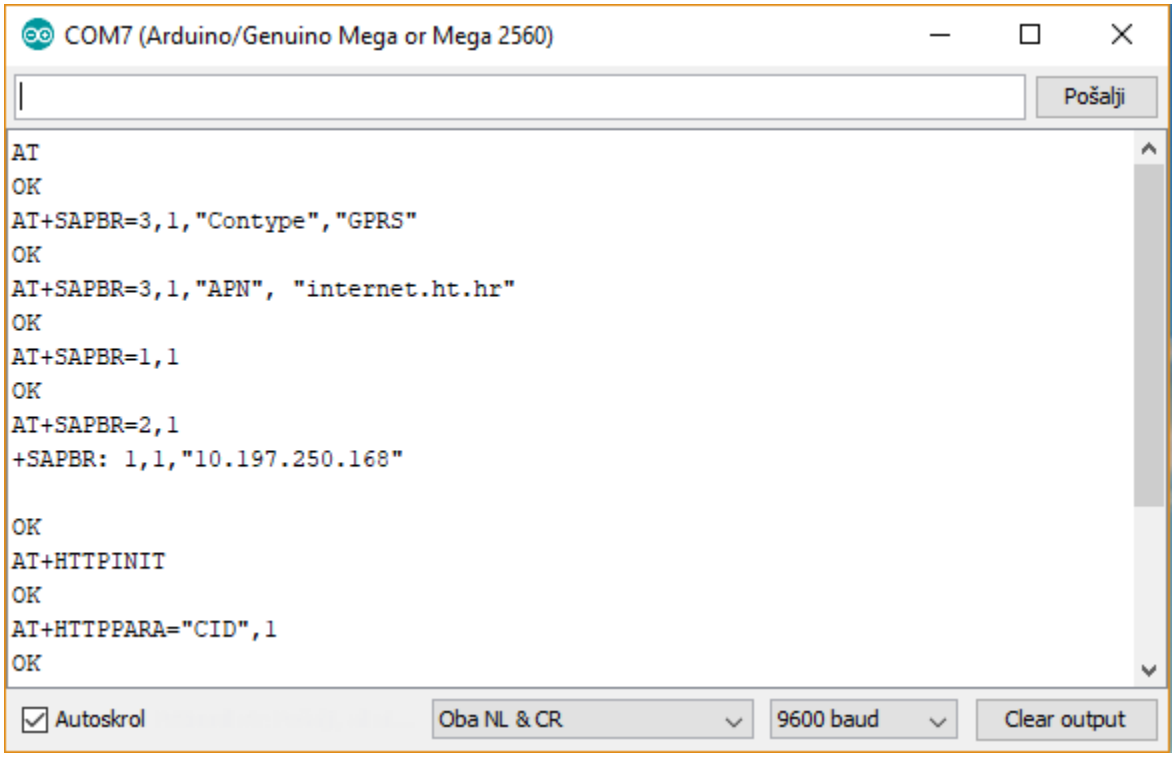
1 do 100, a predstavlja postotak baterije. Sljedeća komanda koja je poslana je “AT+CBC” a predstavlja instrukciju čiji odgovor nam daje stanje napajanja, postotak baterije te voltažu.



Slika 9. Slanje instrukcija za stanje računa

Izvor: Izradio autor

Na slici 9 je prikaz slanja instrukcija preko serijskog monitora za dobivanje stanja računa na SIM kartici koja se nalazi u SIM808 modul-u. Komanda “AT+CUSD?” koja se vidi na slici 8 kao odgovor vraća moguća stanja za slanje USSD komande. Mogući odgovori su 0 i 1 gdje 0 znači da opcija za slanje USSD koda nije omogućena te 1 koja znači da je opcija omogućena. Kako bi se mogao poslati kod za provjeru stanja računa sa modul-a potrebno je prvo omogućiti slanje USSD koda slanjem instrukcije “AT+CUSD=1”. Nakon toga šalje se instrukcija za slanje USSD koda koji se razlikuje ovisno o operateru u ovom slučaju “AT+CUSD=1, *100# ”, a kao odgovor dobiva se poruka o stanju računa.



Slika 10. Slanje HTTP instrukcija

Izvor: Izradio autor

Na slici 10 vidljive su komande koje su korištene za konfiguraciju nositelja mreže te inicijalizaciju HTTP usluge. Ove su komande korištene kako bi se poslala lokacija, stanje računa i stanje baterije te kako bi se to spremilo u bazu podataka za daljnje korištenje u sustavu. Prije inicijalizacije HTTP protokola potrebno je konfigurirati nositelja mreže te otvoriti GPRS kontekst kako bi bilo omogućen prijenos podataka bežičnim putem kroz GSM mrežu. To se izvršava slanjem skupa “AT+SAPBR” instrukcija kao što se vidi na slici. Nakon otvaranja GPRS konteksta moguća je inicijalizacija HTTP usluge slanjem komande “AT+HTTPINIT” te ukoliko dobijemo odgovor “OK” inicijalizacija je uspješna postavljamo parametre HTTP sesije komandama “AT+HTTPPARA = “CID”, 1” i “AT+HTTPPARA=”URL”, “http://foundyou.eu/api/storeLocationBat/1/13.849560/44.877350/2/50/S”. Sesija metode GET započinje slanjem instrukcije “AT+HTTPACTION=0” te očekujemo odgovor “OK” za uspješan početak sesije. Podaci iz odgovora se čitaju pomoću komande “AT+HTTPREAD”, a usluga se prekida slanjem “AT+HTTPTERM” komande.

3.2. Izrada mobilne aplikacije

Mobilna aplikacija je izrađena u Android Studio razvojnom okruženju. Glavna značajka i namjena mobilne aplikacije je prikaz trenutne lokacije na mapi te alarmiranje korisnika mobilne aplikacije ukoliko se praćena osoba udalji iz zadanog radijusa. Za potrebe prikaza lokacije na mapi korišten je su Google API, točnije Maps SDK za Android. Za potrebe alarmiranja odnosno obavještavanje korisnika mobilne aplikacije korištena je Firebase usluga. Maps SDK (Software development kit) omogućuje dodavanje mapa u Android aplikaciju koje se temelje na Google Kartama. Korištenjem ovog API-a omogućeno je postavljanje markera na mapu, poligona, linija, te promjena stila prikaza mape. API automatski rješava pristup serverima Google Karti, preuzimanje podataka te prikaz mape.

Dodatne mogućnosti unutar mobilne aplikacije su ažuriranje podataka o praćenom korisniku, pregled stanja na SIM kartici sa GPS uređaja, stanje baterije GPS uređaja, informacije o zadnjem izlasku iz radijusa te pregled kretanja u vremenskim intervalima.

Kako bi se koristile usluge Google karti potreban je API ključ sa restrikcijom za Android aplikacije. API ključ je jedinstveni identifikator koji se koristi kako bi se potvrdili zahtjevi vezani uz projekt u kojem se koristi. API ključ se dobiva prilikom registracije projekta na Google Cloud Platform konzoli te se taj ključ dodaje u Android aplikaciju. Ključ se dodaje unutar AndroidManifest.xml datoteke unutar meta-data oznake. Osim dodavanja API ključa potrebno je i omogućiti željene Google API usluge unutar Google Developer konzole.

```
<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="Found You"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/AppTheme">

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

Slika 11. Dodavanje API ključa unutar Manifest.xml datoteke

Izvor: Izradio autor

Kako bi se prikazala mapa potrebno je kreirati klasu koja implementira OnMapReadyCallback sučelje i koja nadjačava metodu onMapReady() kako bi se postavila mapa tada kada je GoogleMap objekt dostupan.

```
@Override
public void onMapReady(GoogleMap googleMap) {

    mMap = googleMap;
    mMap.setMinZoomPreference(6.0f);
    mMap.setMaxZoomPreference(20.0f);

    //Current location -> marker
    LatLng current_location = new LatLng(latitude_01, longitude_01);
    MarkerOptions markerOptions = new MarkerOptions().position(current_location).title("Current location");

    //Home location -> marker
    LatLng home = new LatLng(home_latitude, home_longitude);
    MarkerOptions homeMarker = new MarkerOptions().position(home).title("Home");
    model.styleAndAddMarkerOptions(mMap, markerOptions, homeMarker);

}
```

Slika 12. Postavljanje lokacije na mapu

Izvor: Izradio autor

Na slici 12 je prikazano postavljanje minimalnog i maksimalnog povećanja mape te spremanje trenutne lokacije odnosno geografske širine i dužine u LatLng objekt i postavljanje oznake na mapi za tu lokaciju. Osim trenutne lokacije dohvaća se zadnja pohranjena kućna adresa i postavlja se oznaka na mapi za tu lokaciju. Na osnovu trenutne i kućne adrese računa se trenutni radijus kretanja te se uspoređuje sa zadanim radijusom. Ukoliko je trenutni radijus kretanja veći obavještava se korisnik mobilne aplikacije o prekoračenom radijusu.

Geo kodiranje i obrnuto Geo kodiranje u aplikaciji omogućeno je korištenjem Geocoder klase. Geo kodiranje je proces pretvaranja ulične adrese u geografske koordinate odnosno u zemljopisnu širinu i dužinu koje se onda koriste za prikaz lokacije na mapi, a obrnuto Geo kodiranje je proces pretvaranja geografske širine i dužine u adresu koja je razumljiva čovjeku. Funkcija korištena za dobivanje adrese u obliku teksta je getFromLocation koja kao parametre prima zemljopisnu širinu i dužinu iz lokacije te maksimalan broj adresa koji će vratiti.

```
public void getAddressInfoFromLocation(){

    List<Address> addresses;
    Geocoder geocoder = new Geocoder(mContext, Locale.getDefault());

    try {
        addresses = geocoder.getFromLocation(current_latitude, current_longitude, maxResults: 1);
        address = addresses.get(0).getAddressLine( index: 0);
        city = addresses.get(0).getLocality();
        state = addresses.get(0).getAdminArea();
        country = addresses.get(0).getCountryName();
        postalCode = addresses.get(0).getPostalCode();
        knownName = addresses.get(0).getFeatureName();

    }catch (IOException e){
        Log.e(TAG, e.toString());
    }

}
```

Slika 13. Upotreba Geocoder klase

Izvor: Izradio autor

Na slici 13 je prikazano korištenje Geocoder klase te funkcije getFromLocation kako bi se iz geografske širine i dužine dobila adresa koja opisuje područje koje neposredno okružuje zadanu širinu i dužinu. Dobivena adresa se sprema te se kasnije koristi za prikaz adrese koja je razumljiva korisniku mobilne aplikacije.

Obavještavanje korisnika o izlasku praćene osobe iz radijusa riješeno je korištenjem Firebase Cloud Messaging usluge. Kako bi se dodao Firebase u android projekt potrebno je instalirati ili ažurirati Android Studio na zadnju verziju, ciljan API razina mora biti minimum 16 odnosno Jelly Bean te Gradle verzija 4.1 ili kasnija. Osim navedenih zahtjeva potrebno je kreirati Firebase projekt koji ćemo moći povezati sa Android aplikacijom. Nakon kreiranja Firebase projekta možemo registrirati Android aplikaciju unutar tog projekta. Kroz postupak registracije aplikacije preuzimamo konfiguracijsku datoteku naziva “google-services.json” koja se zatim dodajemo u projekt. Kako bi omogućili Firebase proizvode unutar aplikacije potrebno je dodati google-services dodatak unutar Gradle datoteke na razini projekta kako je prikazano na slici 14, te unutar Gradle datoteke na razini aplikacije kako je prikazano na slici 15.

```
buildscript {
    repositories {
        google()
        jcenter()

    }
    dependencies {
        classpath 'com.android.tools.build:gradle:3.4.2'
        classpath 'com.google.gms:google-services:4.3.0'

    }
}
```

Slika 14. Google-services dodatak unutar Gradle datoteke na razini projekta

Izvor: Izradio autor

Unutar Gradle datoteke na razini aplikacije potrebno je dodati i podršku za Cloud Messaging Android biblioteku kao što je prikazano na slici 15.

```
dependencies {

    implementation 'com.google.firebase:firebase-core:17.0.1'
    implementation 'com.google.firebase:firebase-messaging:19.0.1'

    apply plugin: 'com.google.gms.google-services'
}
```

Slika 15. Google-services dodatak unutar Gradle datoteke na razini aplikacije

Izvor: Izradio autor

Unutar AndroidManifest.xml datoteke potrebno je dodati uslugu koja je pod klasa FirebaseMessagingService klase. To je potrebno kako bi mogli rukovati porukama i obavijestima dok je aplikacija aktivna i kako bi se obavijesti mogle prikazivati korisniku dok aplikacija nije aktivna.

```
<service
    android:name="$Model.Utility.MyFirebaseMessagingService">
    <intent-filter>
        <action android:name="com.google.firebase.MESSAGING_EVENT"/>
    </intent-filter>
</service>
```

Slika 16. Dodavanje FirebaseMessagingService usluge u AndroidManifest.xml

Izvor: Izradio autor

Kako bi obavijest bila poslana pravom korisniku potrebno je dohvatiti registracijski token uređaja. Prilikom inicijalnog pokretanja aplikacije generira se registracijski token za instancu klijentske aplikacije. Tom tokenu pristupamo nadjačavanjem funkcije onNewToken. Funkcija onNewToken se pozove svaki put kad je generiran novi token. Nakon inicijalnog pokretanja aplikacije moguća je promjena tokena te se preporuča dohvaćanje zadnjeg ažuriranog registracijskog tokena. Registracijski token se može promijeniti u ovim slučajevima:

- Korisnik deinstalira i ponovno instalira aplikaciju.
- Korisnik očisti podatke aplikacije.
- Aplikacija se vrati na novom uređaju.

Kada treba dohvatiti trenutni token pozove se funkcija sa slike 17.



Slika 17. Dohvaćanje trenutnog registracijskog tokena

Izvor: Izradio autor

Unutar ove funkcije dohvaća se zadnji registrirani token za taj uređaj te se on ažurira u bazi podataka za tok korisnika po identifikacijskom broju korisnika. Nakon šta su napravljene sve navedene postavke moguće je slanje i primanje obavijesti i poruka unutar mobilne aplikacije.

3.3 Web sustav

Web sustav je razvijen u Laravel okruženju. Glavna namjena web sustava je registracija korisnika u sustav te komunikacija sa bazom podataka odnosno dohvaćanje podataka te pohranjivanje i ažuriranje zapisa unutar baze podataka. Dodatna opcija web sustava je također pregled trenutne lokacije korisnika na mapi, pregled kretanja u vremenskim intervalima te ažuriranje podataka o praćenom korisniku. Kako bi bilo moguće korištenje mobilne aplikacije potrebna je registracija u sustav. Nakon registracije u sustav ispunjavaju se podaci o korisniku koji će koristiti GPS uređaja tzv. praćenom korisniku te se nakon toga otvara prikaz mape i mogućnost ažuriranja podataka o praćenom korisniku.

Ovjera autentičnosti korisnika prilikom prijave u registracije u sustav riješena je korištenjem Laravel-ove ovjere autentičnosti. Laravel ima unaprijed ugrađene kontrolere

za ovjeru autentičnosti, a to su “RegisterController” koji se koristi kod registracije novog korisnika, “LoginController” koji rješava ovjeru autentičnosti, “ForgotPasswordController” koji šalje e-poštu s poveznicama za resetiranje lozinke i “ResetPasswordController” koji sadrži logički dio za resetiranje lozinke. Ovi se kontroleri uglavnom ne trebaju mijenjati jer sadrže sve što je potrebno kako bi izvršili potrebne metode. Registracijski kontroler sadrži dvije unaprijed postavljene metode, metoda create() i metoda validator(). Metoda” create” je odgovorna za kreiranje novog korisnika aplikacije. Tu metodu se može mijenjati ako postoji potreba, na primjer ako se dodaju stupci koje je potrebno popuniti prilikom registracije. Metoda “validator” sadrži validacijska pravila za novog korisnika aplikacije i također se može mijenjati ukoliko postoji potreba. Izvršavanjem “php artisan make:auth” komande kreiraju se pogledi za registraciju i pogledi za prijavu u sustavu unutar “resources/views/auth” direktorija kao i rute za krajnje točke ovjere autentičnosti te migracijske datoteke za tablice u bazi podataka. Moguće je mijenjati, prilagođavati i dodavati stupce u tablice unutar migracijskih datoteka za željenu tablicu ukoliko postoji potreba za time. Nakon što smo zadovoljni strukturom tablica u bazi podataka pokreće se “php artisan migrate” komanda kako bi se novo kreirane migracije pohranile u bazu podataka u obliku tablice. Kontroleri za registraciju i prijavu korisnika automatski nakon uspješne ovjere autentičnosti preusmjeravaju na “/home” rutu. Ukoliko želimo promijeniti rutu nakon registracije ili prijave u sustav potrebno je promijeniti varijablu preusmjeravanja unutar odgovarajućeg kontrolera.

```
/**
 * Where to redirect users after login.
 *
 * @var string
 */
protected $redirectTo = '/tracking';
```

Slika 18. Preusmjeravanje nakon prijave u sustav

Izvor: Izradio autor

Na slici 18 možemo vidjeti promjenu varijable preusmjeravanja na rutu koja odgovara potrebama sustava. Rute se definiraju unutar datoteka koje su smještene u Routes direktoriju, a omogućavaju nam usmjeravanje svih zahtjeva aplikacije na odgovarajući

kontroler. Nakon prijave u sustav korisnik će biti preusmjeren na prikazanu rutu “/tracking” koja zatim poziva funkciju iz kontrolera koji je dodijeljen toj ruti.

```
Auth::routes();

Route::group(['middleware' => 'auth'], function () {

    // All my routes that needs authentication

    Route::get('/tracking', 'TrackingController@index')->name('tracking');


});
```

Slika 19. Prikaz tracking rute

Izvor: Izradio autor

Na slici 19 vidimo tracking rutu na koju sustav preusmjerava korisnika nakon uspješne prijave u sustav. Prilikom preusmjeravanja na tu rutu izvršava se index() funkcija koja se nalazi unutar TrackingController-a i koja sadrži daljnju logiku za prikaz podataka korisniku.

Prikaz lokacije na mapi omogućen je korištenjem Google API-a točnije Maps JavaScript API-a i Geocoding API-a. Kao i za mobilnu aplikaciju potreban je Google API ključ kako bi koristili Google API-je. Korištenjem Maps JavaScript API-a omogućeno je prilagođavanje mape vlastitim sadržajem i slikama za prikaz na web stranicama. Sadrži četiri osnovne vrste prikaza karte a to su hibridni prikaz, satelitski prikaz, terenski prikaz i prikaz plana puta te se mogu mijenjati korištenjem raznih slojeva i stilova, kontrola i događaja te različitih usluga i biblioteka. Geocoding API je usluga koja omogućuje Geo kodiranje i obrnuto Geo kodiranje adresa. Za integraciju Google mape s Laravel - om korišten je Googlmapper paket koji pruža razne mogućnosti kako bi se ubrzalo korištenje mape. Prvo je potrebno instalirati taj paket kroz Composer dodavanjem sljedećeg unutar composer.json datoteke:

```
"require": {

"cornford/googlmapper": "2.*"

}
```

Nakon što se dodao zahtjev za instalacijom paketa potrebno je izvršiti ažuriranje Composer-a kroz terminal upisivanjem komande “composer update”. Nakon što je ta operacija izvršena potrebno je unutar konfiguracijske datoteke aplikacije dodati pružatelja usluge unutar polja usluga upisivanjem sljedećeg:

```
'providers' => [
        “ Cornford\Googlmapper\MapperServiceProvider::class “
],
```

te unutar alias polja dodati fasadu Googlmapper klase odnosno naziv po kojem ćemo koristiti instancu Googlmapper klase upisivanjem sljedećeg:

```
 'aliases' => [
        “ 'Mapper' => Cornford\Googlmapper\Facades\MapperFacade::class “
 ],.
```

Zadnji korak je dodati konfiguracijske datoteke za taj paket upisivanjem sljedećeg u terminal:

“php artisan vendor:publish --provider="Cornford\Googlmapper\MapperServiceProvider" --tag=googlmapper”.

Nakon svih koraka može se konfigurirati Googlmapper unutar konfiguracijske datoteke tog paketa te omogućiti ili onemogućiti sve opcije koje su nam potrebne te dodati Google Maps API ključ.

```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Mapper;

class TrackingController extends Controller
{
    public function index()
    {
        $map = Mapper::map($home_lat,$home_long,['zoom' => 16,'animation' => 'DROP','icon'=>
            'http://maps.google.com/mapfiles/kml/pal2/icon2.png','title'=>'HOME']);

        $map->circle([['latitude' => $home_lat, 'longitude' => $home_long]], ['strokeColor' =>
            'darkblue', 'strokeOpacity' => 0.3, 'strokeWeight' => 2, 'fillColor' =>
            '#ADD8E6','fillOpacity'=>0.5, 'radius' => $radius]);

        //trenutni marker
        $map->marker($curr_lat, $curr_long,['animation' => 'BOUNCE','icon'=>
            'http://maps.google.com/mapfiles/kml/paddle/blu-circle.png']);

        //informacija o kuci
        $map->informationWindow($home_lat, $home_long, 'Home', ['open' => false, 'maxWidth'=> 100]);

        //informacija o udaljenosti
        $map->informationWindow($curr_lat, $curr_long, $distance.' m udaljenost od kuće', ['open' =>
            true, 'maxWidth'=> 100]);

        return view('tracking')->with(compact('page_name','home_lat','home_long','curr_lat','curr_long','radius','distance'));

    }
}
?>
```

Slika 20. Inicijalizacija mape

Izvor: Izradio autor

Slika 20 prikazuje inicijalizaciju mape te pripadajuće metode. Metode korištene nad inicijaliziranom mapom koje se vide na slici 14 su “map”, “circle”, “marker” i “information window” a omogućuju sljedeće:

- Metoda “map” omogućuje kreiranje mape s prikazom lokacije u odnosu na geografsku širinu i dužinu te s opcionalnim parametrima za razne opcije kao što je uvećavanje mape ili postavljanje određene slike kao oznaku lokacije.
- Metoda “circle” omogućuje dodavanje kruga oko zadanih koordinata sa opcionalnim parametrima kao što su boja, prozirnost i debljina obruba kruga, boja ispune kruga, širina kruga te prozirnost kruga.
- Metoda “marker” omogućuje dodavanje oznake na zadanu lokaciju i opcionalne parametre za razne opcije kao na primjer animacija i određena slika za prikaz oznake na mapi.
- Metoda “Information window” omogućuje prikaz prozora na mapi iznad oznake lokacije s pripadajućim informacijama.

Za pretvaranje koordinata lokacije u adresu koja je razumljiva korisniku te za pretvaranje ulične adrese u geografsku dužinu i širinu prilikom pohrane u bazu podataka koristi se Geocoding API. Potrebno ga je omogućiti unutar Google developers konzole kako bi ga bilo moguće koristiti. Opcije za Geo kodiranje i obrnuto Geo kodiranje koristiti će se preko Spatie Geocoder paketa. Instalacija paketa se vrši kroz composer upisivanjem sljedećeg u terminal:

“ composer require spatie/geocoder “.

Unutar konfiguracijske datoteke aplikacije potrebno je dodati pružatelja usluge unutar polja usluga:

```
'providers' => [
    “ Spatie\Geocoder\GeocoderServiceProvider::class “
],
```

te unutar polja aliasa dodati fasadu, odnosno naziv po kojem ćemo koristit instancu Spatie Geocoder klase:

```
'aliases' => [
    “ 'Geocoder' => Spatie\Geocoder\Facades\Geocoder::class “
],.
```

Zadnji korak je dodati konfiguracijske datoteke za taj paket upisivanjem sljedećeg u terminal:

“php artisan vendor:publish --provider="Spatie\Geocoder\GeocoderServiceProvider" --tag="config" “. Zadnji korak je dodavanje Google API ključa unutar generirane konfiguracijeske datoteke kako bi se mogle koristiti usluge Geo kodiranja.

```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Geocoder;

class EditUserController extends Controller
{
    public function update(Request $request)
    {
        $country = $request->get('country');
        $street = $request->get('street');
        $city = $request->get('city');
        $adress=$street.','.$city.','.$country;

        $result=Geocoder::getCoordinatesForAddress($adress);

        //latitude => $result['lat']
        //longitude =>$result['lng']

        return response()->json($result);

    }
}
?>
```

Slika 21. Dohvaćanje koordinata iz adrese

Izvor: Izradio autor

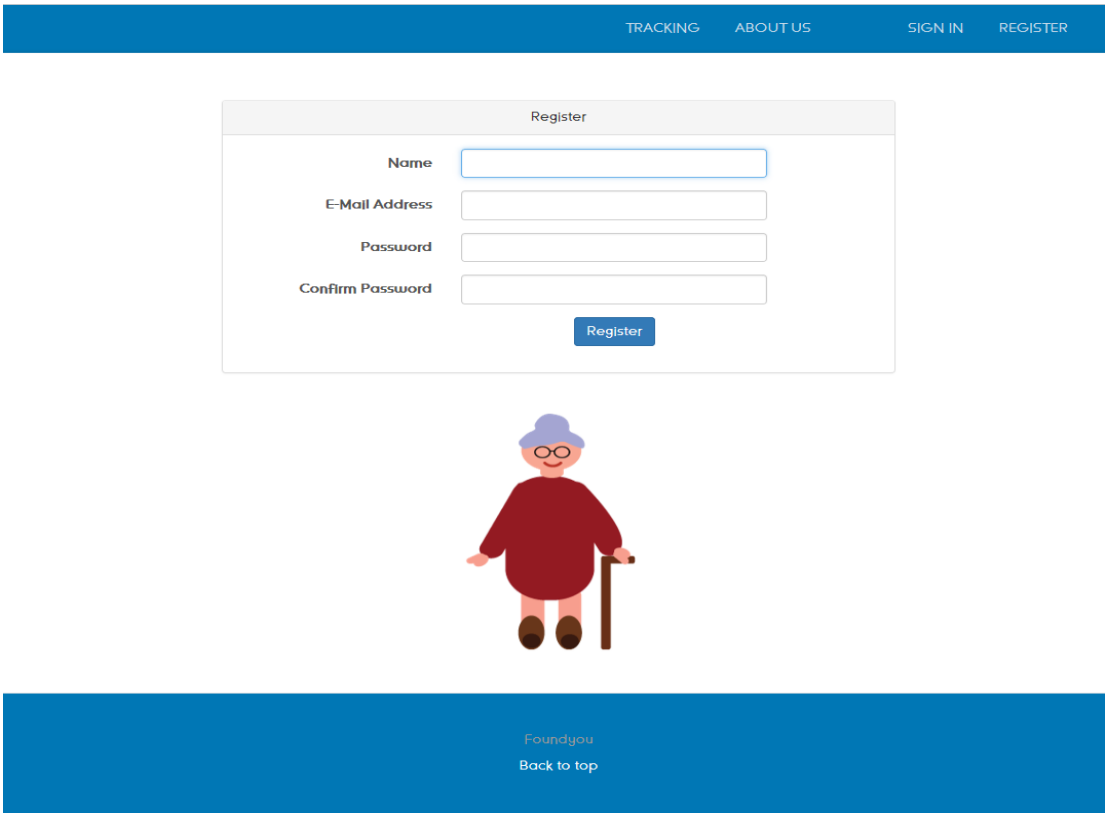
Na slici 21 je isječak koda koji prikazuje korištenje “getCoordinatesForAddress” metode kako bi se iz adrese, odnosno ulice, grada i države dobile točne koordinate u obliku geografske dužine i širine. U ovom projektu Geocoding API se koristi prilikom ažuriranja podataka o korisniku. Lokacija u bazi podataka je pohranjena u obliku dvije koordinate od kojih jedna predstavlja geografsku širinu a druga geografsku dužinu. Prilikom ažuriranja podataka o korisniku u postavkama potrebno je upisati uličnu adresu. Kako bi se ta lokacija mogla prikazati na mapi potrebno je pretvoriti uličnu adresu u koordinate lokacije te se geografska dužina i širina spremaju u tom formatu u bazi podataka.

4. Prikaz sučelja web aplikacije

U ovom poglavlju biti će prikazano sučelje web aplikacije te korisničke upute. Glavna zadaća web aplikacije je registracija u sustav te prikaz lokacije na mapi. Web aplikacija se sastoji od pogleda za registraciju u sustav, prikaza lokacije na mapi te pogleda za ažuriranje podataka o praćenom korisniku.

4.1. Registracija u sustav

Prilikom registracije u sustav korisnik mora upisati svoje ime, adresu elektroničke pošte i lozinku. Elektronička pošta i lozinka služe za identifikaciju korisnika prilikom sljedećih prijava. Nakon što se popune sva potrebna polja klikom na gumb za registraciju korisnika se preusmjerava na pogled za registraciju dezorijentirane osobe koja će nositi GPS uređaj.

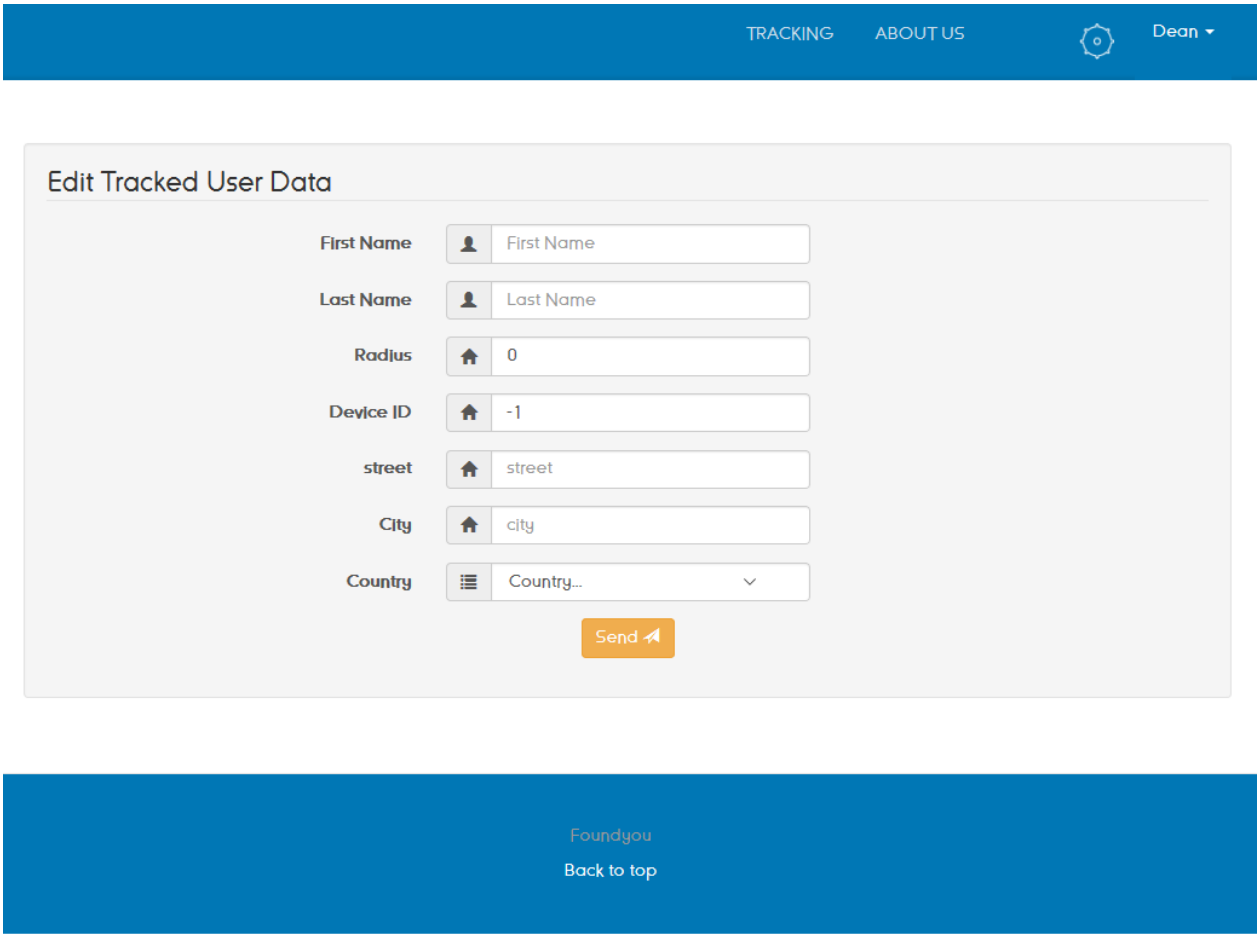


Slika 22. Registracija korisnika

Izvor: Izradio autor

4.2. Registracija dezorijentirane osobe

Nakon što je korisnik uspješno registriran u sustav prikazuje mu se pogled prikazan na slici 23. Korisnika će se prilikom svake prijave u sustav preusmjeriti na taj pogled i neće biti u mogućnosti koristiti web aplikaciju dok ne unese potrebne podatke o osobu čiju bi lokaciju želio pratiti. Nakon što popuni podatke o tom korisniku može pristupiti pogledu za pregled lokacije na mapi. Ukoliko u nekom trenutku korištenja aplikacije postoji potreba za promjenom podataka o dezorijentiranoj osobi, tada se klikom na gumb za postavke opet otvara ovaj pogled unutar kojeg je moguće ažurirati potrebna polja.

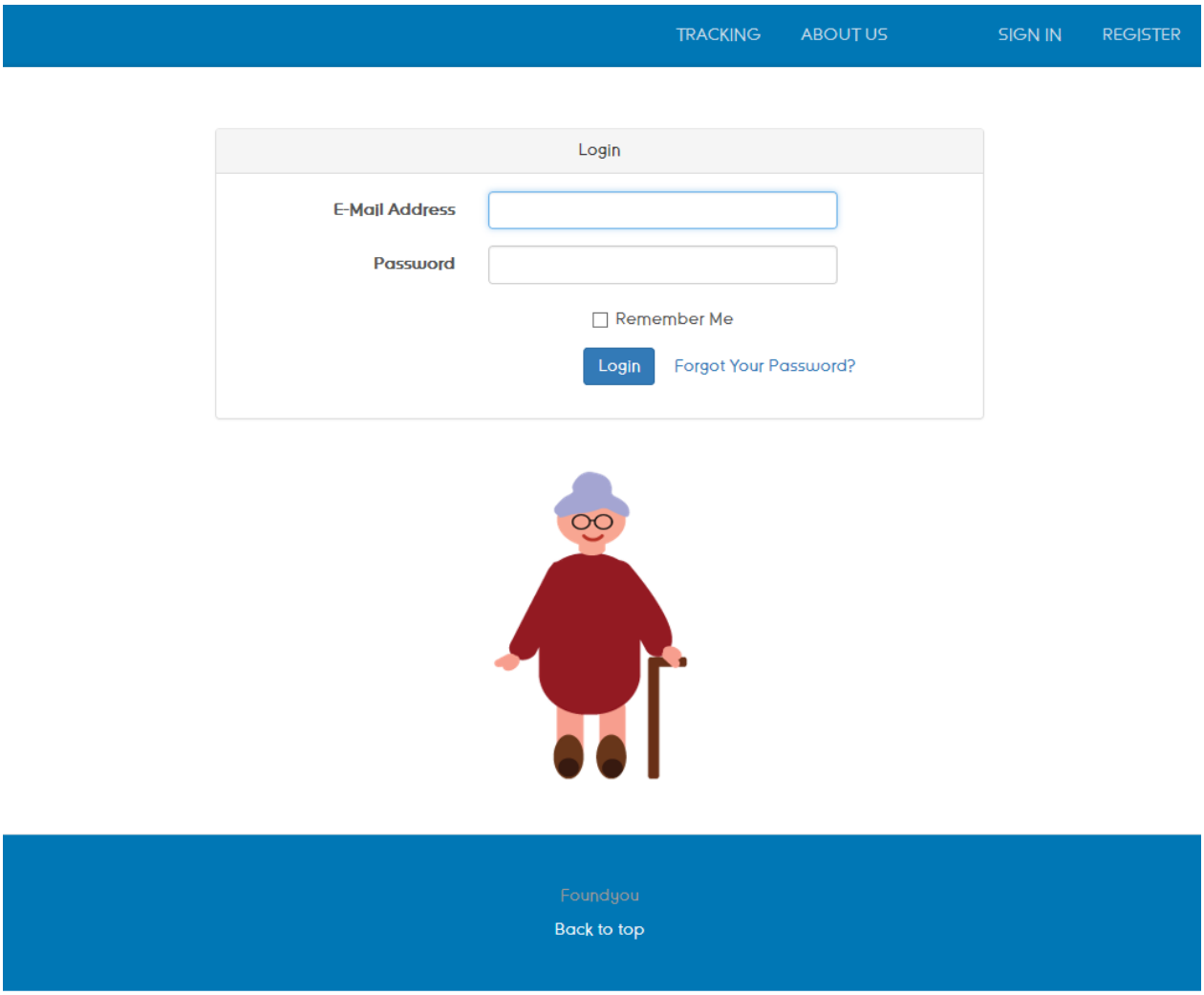


Slika 23. Unos podataka o dezorijentiranoj osobi

Izvor: Izradio autor

4.3. Prijava u sustav

Pogled za prijavu u sustav prikazan je na slici 24. Za prijavu u sustav korisnik mora upisati adresu elektroničke pošte koju je upisao prilikom registracije te točnu lozinku. Zatim slijedi autorizacija korisnika i ukoliko autorizacija uspješna tada se korisnika preusmjerava na pogled koji prikazuje trenutnu lokaciju praćenog korisnika.

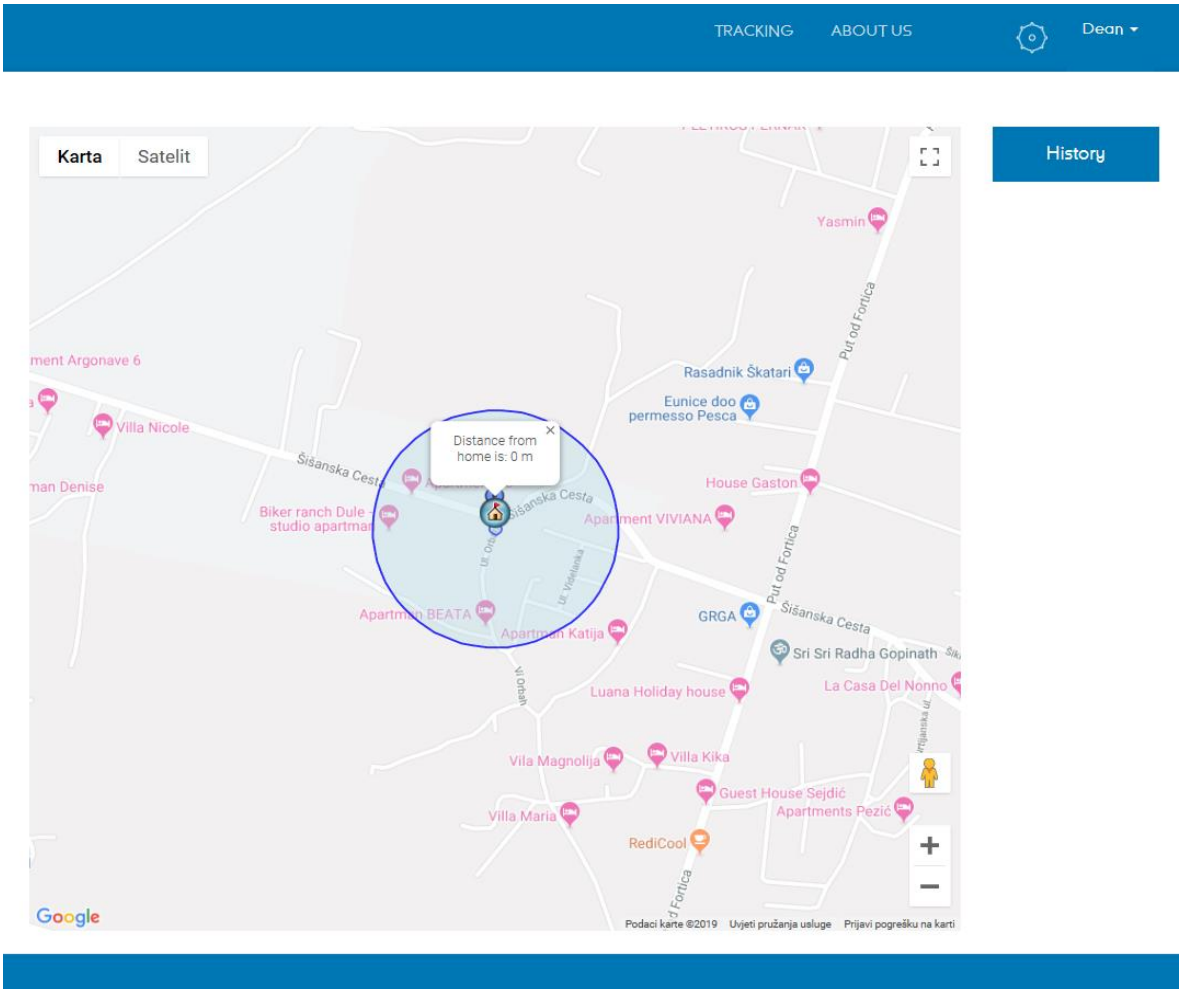


Slika 24. Prijava u sustav

Izvor: Izradio autor

4.4. Prikaz lokacije na mapi

Nakon registracije u sustav i popunjavanja podataka o dezorijentiranoj osobi korisnik može vidjeti lokaciju te osobe na mapi kao što se vidi na slici 25. Prilikom svake sljedeće prijave u sustav korisniku se prvo prikazuje upravo ovaj pogled unutar kojeg mu se prvo prikazuje trenutna lokacija koja je prikazana oznakom sa logom i kućna lokacija koja ima ikonu kuće. Iznad oznake trenutne lokacije korisnik vidi kolika je udaljenost od kućne adrese te trenutna adresa na kojoj se praćena osoba nalazi. Sa desne strane moguće je odabrati prikaz lokacije po vremenskim intervalima pritiskom na gumb “History”. Nakon što korisnik klikne na željeni vremenski interval na mapi će biti iscrtan pravac između početne lokacije odabranog intervala do zadnje poznate lokacije.

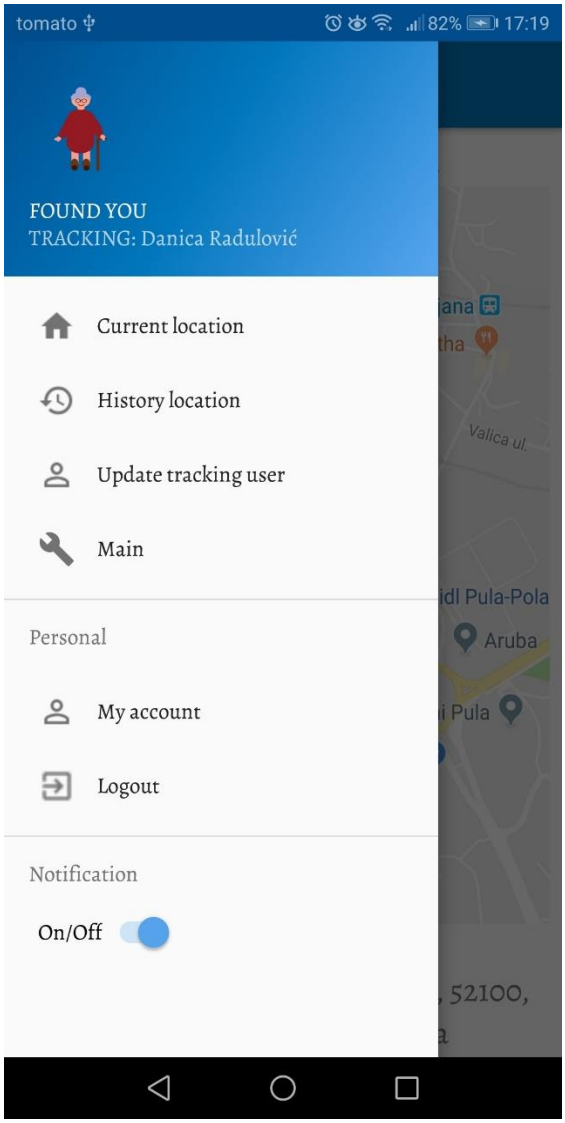


Slika 25. Prikaz lokacije na mapi

Izvor: Izradio autor

5. Prikaz sučelja mobilne aplikacije

Mobilna aplikacija se sastoji od nekoliko aktivnosti. Da bi ju korisnik mogao koristiti potrebna je prijava u sustav koristeći adresu elektroničke pošte i lozinku kao i kod prijave u web aplikaciju. Unutar aplikacije korisnik može pregledavati trenutnu lokaciju na mapi, lokacije u odabranom vremenskom intervalu, ažurirati podatke o praćenom korisniku ukoliko je potrebno te pregledavati status GPS uređaja. Korisnik se može odjaviti iz sustava pritiskom na “Logout” gumb.

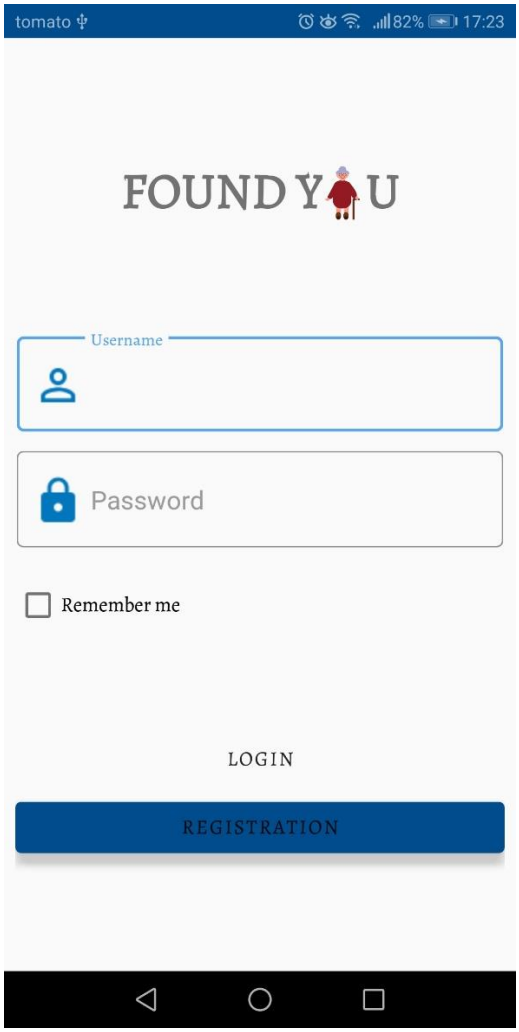


Slika 26. Glavni izbornik

Izvor: Izradio autor

5.1. Prijava u mobilnu aplikaciju

Prvi pogled koji se korisniku pojavljuje prilikom prvog pokretanja aplikacije je pogled za prijavu u aplikaciju. Sastoji se od polja za upis elektroničke pošte te lozinke. Korisnik ima mogućnost označiti polje “Remember me” pa će se prilikom sljedećeg pokretanja aplikacije smatrati prijavljenim u aplikaciju. Ukoliko korisnik nije registriran, pritiskom na “Register” button biti će preusmjeren na lokaciju za registraciju. Klikom na “login” button korisnik se autorizira te ukoliko je autorizacija uspješna vodi ga na sljedeću aktivnost.

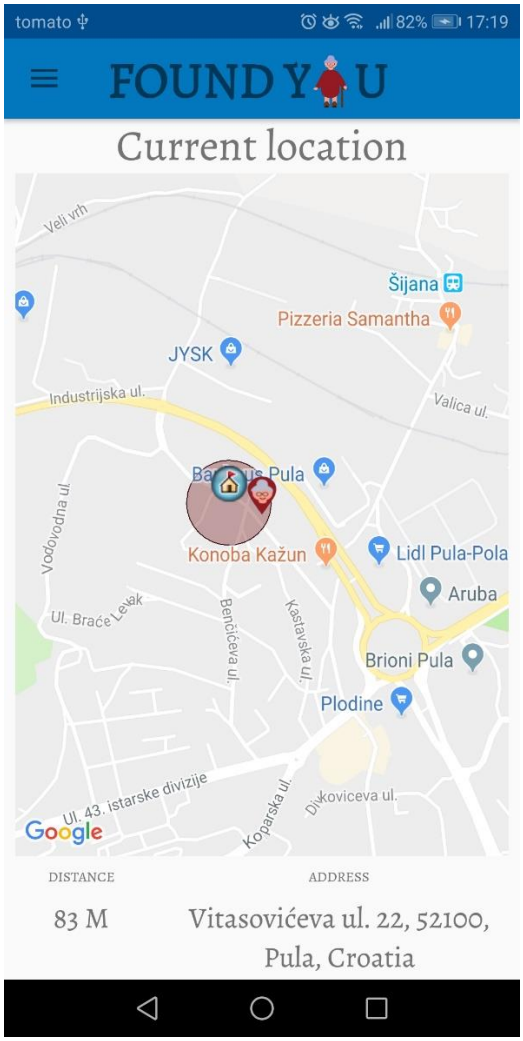


Slika 27. Prikaz pogleda za prijavu aplikaciju

Izvor: Izradio autor

5.2 . Pregled lokacije

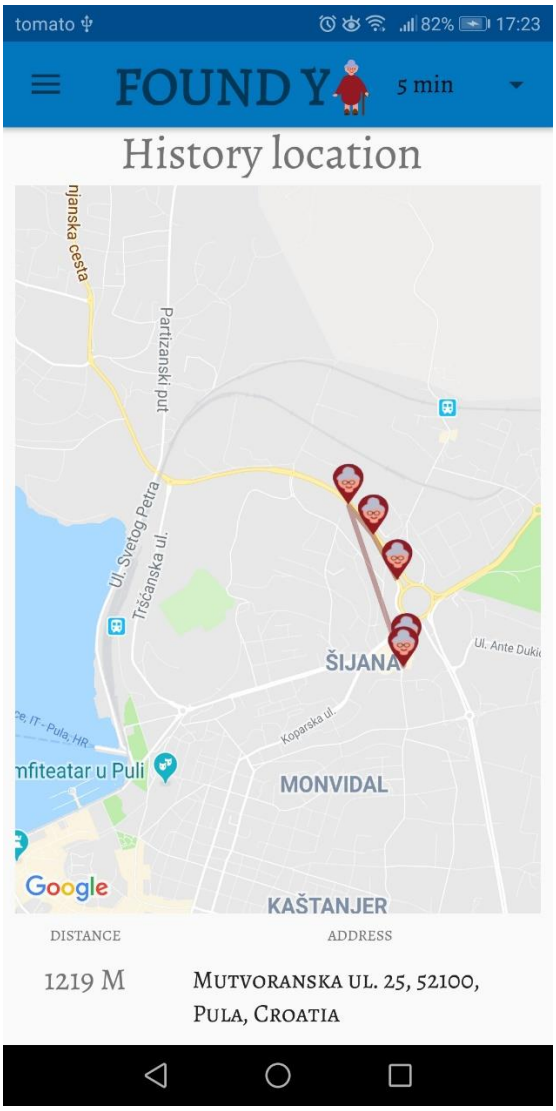
Nakon što se korisnik uspješno prijavio u sustav prikazuje se trenutna lokacija dezorijentirane osobe. Unutar glavnog izbornika korisnik može birati između pregleda trenutne lokacije i pregleda lokacije u odabranim vremenskim intervalima. Oznaka trenutne lokacije prikazana je slikom loga dok je kućna lokacija označena slikom kuće. U dnu ovog pogleda nalazi se udaljenost između trenutne i kućne lokacije i adresa na kojoj se trenutno dezorijentirana osoba nalazi. U gornjem lijevom kutu nalazi se gumb koji na pritisak otvara glavni izbornik prikazan na slici 26. Slika 28 prikazuje pogled trenutne lokacije.



Slika 28. Pregled trenutne lokacije korisnika

Izvor: Izradio autor

Unutar pogleda za pregled lokacije u vremenskim intervalima korisnik može birati između pregleda lokacija u zadnjih pet minuta, trideset minuta, jednog sata i 2 sata. Prilikom učitavanja pogleda povijesti kretanja početno je prikazana lokacija za vremenski interval od zadnjih pet minuta. Unutar pogleda povijesti kretanja također se nalazi gumb za glavni izbornik te udaljenost između kućne i trenutne lokacije te adresa zadnje zabilježene lokacije. Na slici 29 je prikazan pogled povijesti kretanja.

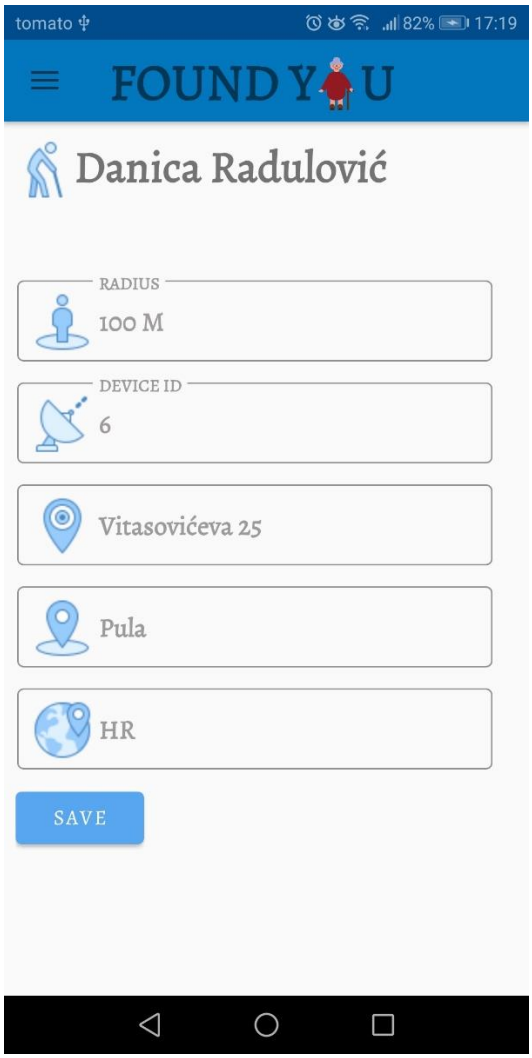


Slika 29. Prikaz kretanja u vremenskom intervalu od 5 minuta

Izvor: Izradio autor

5.3. Ažuriranje podataka dezorijentirane osobe

Ukoliko se unutar glavnog izbornika izabere aktivnost “Update tracked user” tada aplikacija preusmjerava korisnika na pogled ažuriranja podatak dezorijentirane osobe. Unutar te aktivnosti korisnik može promijeniti bilo koju od sljedećih informacija o dezorijentiranoj osobi: ime, prezime, radijus, kućnu adresu, grad i državu. Pritiskom na “save” button informacije će biti ažurirane.



Slika 30. Prikaz pogleda za ažuriranje podataka dezorijentirane osobe

Izvor: Izradio autor

5.4. Pregled statusa GPS uređaja

Unutar ove aktivnosti korisnik može vidjeti informacije o statusu GPS uređaja kao što je preostalo stanje računa na SIM kartici, postotak baterije te je li uređaj aktivan ili nije. Također korisnik može vidjeti informacije o zadnjem udaljavanju iz radijusa kao što je ime i prezime osobe koja je izašla iz radijusa, adresa izlaska iz radijusa, datum, vrijeme i udaljenost od kućne adrese.



Slika 31. Prikaz informacija o GPS uređaju

Izvor: Izradio autor

6. Usporedba drugi sklopovskih rješenja za geolociranje

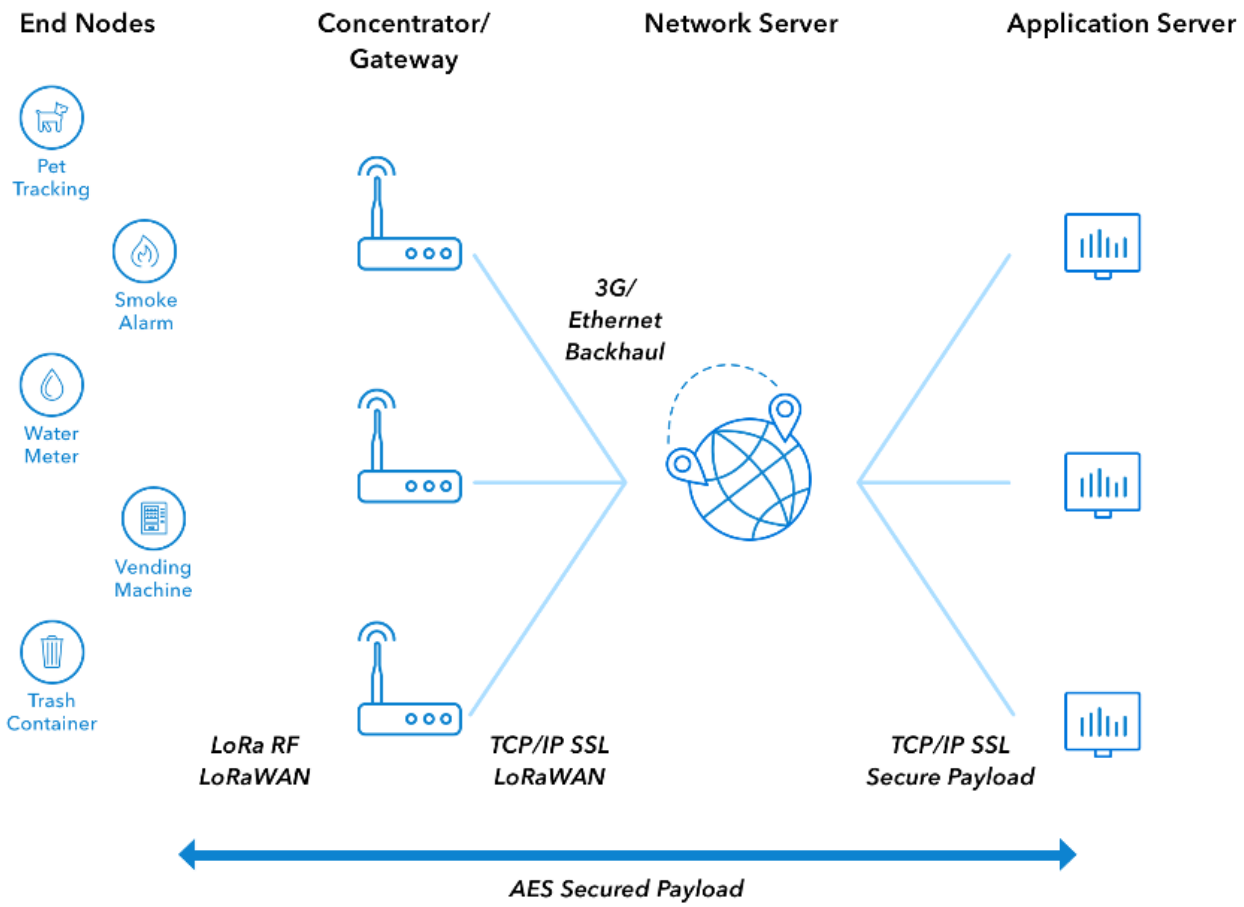
Geolociranje je moguće ostvariti korištenjem raznih tehnologija. Najčešće korištena tehnologija je geolociranje korištenjem globalnih navigacijskih satelitskih sustava te globalnih položajnih sustava. Takvi sustavi pružaju mogućnost geoprostornog pozicioniranja odnosno omogućuju elektroničkim prijamnicima određivanje njihove lokacije koristeći signale dobivene od satelita. Prednost tih sustava je visoka točnost lokacije s odmakom od svega nekoliko metara, smatraju se najboljom tehnologijom za određivanje lokacije s obzirom na točnost i preciznost. Veliki nedostatak takvih sustava je nemogućnost dohvaćanja lokacije u zatvorenim prostorima kao što su zgrade i kuće te relativno visoka potrošnja baterije. Geolociranje je moguće korištenjem Wi-Fi sustav pozicioniranja koji koristi karakteristike obližnjih Wi-Fi žarišnih točaka i drugih bežičnih pristupnih točaka kako bi otkrio gdje se uređaj nalazi. Tipični parametri koji se koriste prilikom geolociranja bežičnim pristupom su SSID i MAC adresa uređaja, a točnost ovisi o broju pristupnih točaka koji se nalaze u blizini. Koristi se za pozicioniranje odnosno određivanje lokacije u unutarnjim prostorima. Ovakvi sustavi imaju dobru točnost međutim potrošnja baterije je relativno visoka te ovise o Wi-Fi pristupnim točkama što nije pogodno na mjestima gdje nema dovoljno pristupnih točaka. Problem korištenja jednog ili drugog od navedenih tehnologija je mogućnost dohvaćanja lokacije unutar zatvorenog prostora ili obrnuto. Ukoliko je potrebno praćenje starije osobe tada će korištenjem globalnih navigacijskih satelitskih sustava biti moguće locirati osobu u trenutku kad se ona udalji od poznate lokacije ukoliko se kreće otvorenim prostorima i ta će lokacija biti vrlo točna, no problem nastaje kada osoba zaluta u zatvoreni prostor iz razloga što se tada gubi GPS signal. U tim slučajevima rješenje je Wi-Fi sustav za pozicioniranje, međutim ukoliko se osoba udalji u smjeru neke nenastanjene livade ili šume neće ju biti moguće locirati. Potrošnja baterija u oba slučaja je relativno visoka te postoji mogućnost da se GPS uređaj ugasi prije lociranja osobe. Rješenje za takve sustave s minimalnom potrošnjom baterije je LoRa.

6.1. LoRa i LoRaWAN

LoRa je fizički sloj ili bežična modulacija koja se koristi za stvaranje komunikacijske veze dugog dosega. Mnogi bežični sustavi koriste modulaciju u kojoj se digitalna informacija prenosi kroz diskretne promjene frekvencije nositelja signala kao fizički sloj jer je to vrlo učinkovita metoda i treba vrlo malo snage. LoRa se temelji na “chirp” tehnici raširenog spektra (eng. CSS - chirp spread technique) koja za šifriranje informacija koristi “chirp” širokopojasne linearne frekvencije. Chirp je sinusoidni signal ili frekvencija koja se s vremenom povećava ili smanjuje. Takva tehnika raširenog spektra zahtijeva jednako malo energije ali znatno povećava domet komunikacije. Prednost LoRa tehnologije je upravo u mogućnosti širokog dometa. Jedan pristupnik ili bazna stanica može pokriti čitav grad ili stotine kvadratnih kilometara. Iako domet uvelike ovisi o okruženju ili preprekama na određenim mjestima, LoRa i LoRaWAN imaju proračun veze koji je veći od bilo koje druge standardizirane komunikacijske tehnologije. Proračun veze koji se obično računa u decibelima je glavni faktor u određivanju raspona u danom okruženju. Proračun veze predstavlja obračun svih gubitaka i dobitaka od odašiljača preko medija (slobodan prostor, kabel, vlakna) do prijamnika. Formula za računanje proračuna veze je :

Primljena snaga (dB) = Prenesena snaga(dB) + Dobitak(dB) - Gubitak(dB).

LoRaWAN je protokol za kontrolu pristupa medijima za širokopojasne mreže. LoRaWAN predstavlja komunikacijski protokol i arhitekturu sustava za mrežu dok LoRa fizički sloj omogućuje komunikaciju širokog raspona. Značajka tog protokola je niska potrošnja ali obuhvaća široko područje. Dizajniran je za bežično povezivanje uređaja koji koriste prijenosnu bateriju kao izvor napajanja s internetom u regionalnim, nacionalnim ili globalnim mrežama. Cilja ključne zahtjeve Interneta stvari kao što e dvosmjerna komunikacija, potpuna sigurnost te mobilnost i usluge lokalizacije. Mrežna arhitektura raspoređena je u zvjezdastoj tipologiji kod koje pristupnici prenose poruke između krajnjih uređaja i centralnog mrežnog poslužitelja. Pristupnici su povezani na mrežni poslužitelj putem standardnih IP veza te djeluju kao nevidljivi most pretvarajući RF (Radio frequency) pakete u IP pakete i obrnuto. Bežična komunikacija koristi karakteristike dugog dometa LoRaÒ fizičkog sloja, te omogućuje vezu s jednim preskokom između krajnjeg uređaja i jednog ili više pristupnika.



Slika 32. LoRaWAN arhitektura.

Izvor: https://www.thethingsnetwork.org/docs/lorawan

Slika 32 prikazuje LoRaWAN arhitekturu koja se sastoji od:

- "End Nodes" odnosno krajnjih čvorova ili uređaja. Krajnji čvor ili uređaj predstavlja objekt s ugrađenim komunikacijskim uređajem male snage.
- "Gateway" odnosno pristupnika. Pristupnici su ustvari antene koje primaju signale s krajnjih uređaja i šalju podatke natrag krajnjim uređajima.
- "Network Server" odnosno mrežni poslužitelj. Mrežni poslužitelj usmjerava poruke s krajnjih uređaja na aplikaciju i natrag.
- "Application server" odnosno aplikacija. Aplikacija je dio softvera pokrenut na poslužitelju.
- Poruke koja se šalje s uređaja na aplikaciju te
- Poruke koja se šalje s aplikacije na uređaj.

Unutar LoRaWAN mreže, čvorovi nisu povezani s specifičnim pristupnikom već su podaci koji se prenose s čvora primljeni od više pristupnika. Svaki će pristupnik proslijediti primljeni paket s krajnjeg čvora na mrežni poslužitelj temeljen na oblaku. Logika i kompleksnost je smještena na mrežni poslužitelj koji upravlja mrežom i filtrira redundantne primljene pakete, izvodi sigurnosne provjere i izvodi brzinu prilagođavanja podataka. Ukoliko je čvor mobilan odnosno u pokretu tada primopredaja od pristupnika nije potrebna. Visok mrežni kapacitet postiže se korištenjem adaptivne brzine prijenosa podataka i korištenjem više kanalnog multi-modemskog primopredajnika u pristupniku tako da se istovremeno mogu primiti poruke na više kanala. Kritični čimbenici koji utječu na kapacitet su broj istodobnih kanala, brzina prijenosa podataka, duljina korisnog opterećenja i koliko često čvorovi prenose podatke. Ako se čvor nalazi blizu pristupnika i ima dobru vezu onda ne koristi najnižu brzinu podataka kako ne bi popunjavao raspoloživi spektar duže nego što je potrebno. Pomicanjem brzine prijenosa podataka otvara se više potencijalnog prostora za prijenos. Sve te značajke omogućuju LoRaWAN mreži vrlo visok kapacitet. Mreža se može rasporediti s minimalnom količinom infrastrukture, može se dodati više pristupnika s obzirom na to da je potreban kapacitet te na taj način smanjiti količinu pregrijavanja na druge pristupnike.

LoRa geolociranje je način lociranja uređaja u ruralnim i urbanim područjima bazirano na LoRa platformi. Cilj razvoja ove platforme bio je smanjiti potrošnju baterije te su se iz tog razloga odlučili odmaknuti od tradicionalnog geolociranja preko GPS senzora koji može mnogo povećati potrošnju baterija LPWAN (Low Power Wide Area network) senzora. Ideja je da nakon što je LoRa signal s pojedinog uređaja pokupilo barem 3 pristupnika moguće je triangulirati položaj uređaja koristeći vremensku oznaku. LoRa čipovi na pristupnicima imaju vrlo točne satove koji točno bilježe vrijeme, do unutar Nano sekundi, prijema prijenosa. Svaki pristupnik ima svoju lokaciju koja je poznata te ukoliko je poznato točno vrijeme kada primaju signal sa uređaja, moguće je izračunati poziciju uređaja.

ZAKLJUČAK

Dezorijentacija predstavlja gubitak sposobnosti neke osobe da se snađe u vremenu i prostoru te je vezana uz poremećaj svijesti. Dezorijentaciju prati umanjena mogućnost opažanja, shvaćanja, zaključivanja i mišljenja, a nastaje kao posljedica nekog blažeg ili intenzivnijeg poremećaja ili bolesti te može biti prolazna ili trajna. Dezorijentirane osobe se često udalje s adrese trenutnog prebivališta te ne znaju naći put natrag s obzirom na gubitak sposobnosti koji nastaju kao posljedica dezorijentacije. Osobe koje su odgovorne za dezorijentirane osobe u trenutku udaljavanja moraju brzo reagirati te odlučiti u kojem pravcu će započeti svoju potragu. Ukoliko krenu u krivom pravcu proces potrage će biti duži, a mogućnost pronalaska svakim trenutkom sve manja. Sustav za geolociranje dezorijentiranih starijih osoba pomaže odgovornim osobama u procesu traganja za dezorijentiranom osobom u trenutku udaljavanja s trenutnog mjesta prebivališta. U trenutku udaljavanja sustav šalje obavijest odgovornoj osobi o udaljavanju te trenutnoj adresi što olakšava pronalazak dezorijentirane osobe. Ukoliko dođe do tehničkih poteškoća s GPS uređajem, moguće je vidjeti zadnje lokacije kretanja te je na taj način proces traženja pojednostavljen s obzirom da je uz informacije o zadnjoj lokaciji lakše odrediti u kojem pravcu se potraga nastavlja. Ovaj sustav omogućuje odgovornim osobama da u bilo kojem trenutku sa bilo kojeg mjesta pregledaju trenutnu lokaciju dezorijentirane osobe koja nosi GPS uređaj.

Literatura

[1] Balagurusamy, E. (2007) Programming with Java - A Primer. Third Edition. New Delhi: Tata McGraw-Hill

[2] Hoffman, J. (2018) Mastering Arduino. United Kingdom: Packt Publishing Ltd

[3] Stauffer, M. (2019) Laravel: Up & Running. A Framework for Building Modern PHP Apps. Second Edition. United States of America: O'Reilly Media,Inc.

[4] Tatroe, K. i MacIntyre, P. (2020). Programming PHP. Creating Dynamic Web Pages.Early Release - Raw & Unedited. Fourth Edition. United States of America: O'Reilly Media, Inc.

[5] Hughes, J. M. (2016). Arduino. A Technical Reference. A HANDBOOK FOR TECHNICIANS, ENGINEERS, AND MAKERS. First Edition. United States of America: O'Reilly Media, Inc.

[6] https://developer.android.com/studio/intro [Pristupljeno: 19.07.2019]
[7] https://www.jetbrains.com/idea/ [Pristupljeno: 19.07.2019]
[8] https://firebase.google.com/products/cloud-messaging [Pristupljeno: 21.07.2019]
[9] https://www.arduino.cc/en/guide/introduction [21.07.2019]
[10] https://www.circuito.io/blog/arduino-code/ [21.07.2019]
[11] http://wiring.org.co/ [21.07.2019]
[12] https://processing.org/ [21.07.2019]
[13] https://learn.sparkfun.com/tutorials/what-is-an-arduino/all [Pristupljeno 23.07.2019]
[14] https://store.arduino.cc/usa/arduino-uno-rev3 [23.07.2019]
[15] https://www.elecrow.com/wiki/index.php?title=SIM808_GPRS/GSM%2BGPS_Shield_v1.1 [23.07.2019]
[16] https://cdn-shop.adafruit.com/datasheets/SIM808_GPS_Application_Note_V1.00.pdf [23.07.2019]
[17] ttps://www.elecrow.com/wiki/images/2/20/SIM800_Series_AT_Command_Manual_V1.09.pdf [23.07.2019]

[18] https://laravel.com/ [24.07.2019].
[19] https://laravel-news.com/laravel-v6-announcement [28.07.2019]
[20] https://www.php.net/manual/en/refs.database.php [28.07.2019]
[21] https://developer.android.com/reference/android/location/Geocoder [06.08.2019]
[22] https://firebase.google.com/docs/android/setup [06.08.2019]
[23] https://www.thethingsnetwork.org/docs/lorawan/[11.08.2019]
[24] https://lora-alliance.org/about-lorawan [11.08.2019]

Popis slika

Slika 1. Komponente sustava za geolociranje dezorijentiranih osoba 3
Slika 2. Dijagram korištenja sustava (eng. use-case) 4
Slika 3. Sekvencijski dijagram 5
Slika 4. Klasni dijagram sustava 6
Slika 5. Glavni prozor Android Studio-a 8
Slika 6. Inicijalizacija komunikacije unutar Arduino IDE 19
Slika 7. Slanje instrukcija za GPS 20
Slika 8. Slanje instrukcija za bateriju 21
Slika 9. Slanje instrukcija za stanje računa 22
Slika 10. Slanje HTTP instrukcija 23
Slika 11. Dodavanje API ključa unutar Manifest.xml datoteke 24
Slika 12. Postavljanje lokacije na mapu 25
Slika 13. Upotreba Geocoder klase 26
Slika 14. Google-services dodatak unutar Gradle datoteke na razini projekta 27
Slika 15. Google-services dodatak unutar Gradle datoteke na razini aplikacije 27
Slika 16. Dodavanje FirebaseMessagingService usluge u AndroidManifest.xml 28
Slika 17. Dohvaćanje trenutnog registracijskog tokena 29
Slika 18. Preusmjeravanje nakon prijave u sustav 30
Slika 19. Prikaz tracking rute 31
Slika 20. Inicijalizacija mape 33
Slika 21. Dohvaćanje koordinata iz adrese 35
Slika 22. Registracija korisnika 36
Slika 23. Unos podataka o dezorijentiranoj osobi 37
Slika 24. Prijava u sustav 38
Slika 25. Prikaz lokacije na mapi 39
Slika 26. Glavni izbornik 40
Slika 27. Prikaz pogleda za prijavu aplikaciju 41
Slika 28. Pregled trenutne lokacije korisnika 42
Slika 29. Prikaz kretanja u vremenskom intervalu od 5 minuta 43
Slika 30. Prikaz pogleda za ažuriranje podataka dezorijentirane osobe 44
Slika 31. Prikaz informacija o GPS uređaju 45
Slika 32. LoRaWAN arhitektura. 48

Sažetak

Zadatak ovog diplomskog rada bio je razvoj sustava za geolociranje dezorijentiranih starijih osoba. Cilj tog sustava je olakšano geolociranje dezorijentirane osobe prilikom udaljavanja s adrese prebivališta. Sustav se sastoji od tri segmenta, a to su prototip GPS uređaja, mobilna aplikacija i web aplikacija. Geolociranje je omogućeno kroz prototip GPS uređaja koji šalje trenutnu lokaciju dezorijentirane osobe. Sustav koriste dvije osobe, dezorijentirana osoba i odgovorna osoba. Dezorijentirana osoba nosi GPS uređaj dok odgovorna osoba koristi mobilnu ili web aplikaciju. Registracija odgovorne i dezorijentirane osobe odvija se unutar web aplikacije, a pregled trenutne lokacije moguć je kroz mobilnu i web aplikaciju.

Summary

The purpose of this thesis was to develop a system for geolocating disoriented older people. The goal of this system is to make it easier to geolocate a disoriented person when moving away from their place of residence. The system consists of three segments, namely a GPS prototype device, a mobile application and a web application. Geolocation is enabled through a prototype of GPS device that sends the current location of the disoriented person. The system is used by two people, a disoriented person and a responsible person. A disoriented person carries a GPS device while the responsible person uses a mobile or web application. The registration of a responsible and disoriented person takes place within the web application, and viewing the current location is possible through the mobile and web application.