

Mobilna aplikacija za organiziranje zajedničke vožnje temeljena na podacima iz otvorenih izvora

Đokić, Nikola

Master's thesis / Diplomski rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Electrical Engineering and Computing / Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:168:923071>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2025-03-20**



Repository / Repozitorij:

[FER Repository - University of Zagreb Faculty of Electrical Engineering and Computing repository](#)



SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 485

**MOBILNA APLIKACIJA ZA ORGANIZIRANJE ZAJEDNIČKE
VOŽNJE TEMELJENA NA PODACIMA IZ OTVORENIH
IZVORA**

Nikola Đokić

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 485

**MOBILNA APLIKACIJA ZA ORGANIZIRANJE ZAJEDNIČKE
VOŽNJE TEMELJENA NA PODACIMA IZ OTVORENIH
IZVORA**

Nikola Đokić

Zagreb, lipanj 2024.

DIPLOMSKI ZADATAK br. 485

Pristupnik: **Nikola Đokić (0036519176)**
Studij: Računarstvo
Profil: Programsko inženjerstvo i informacijski sustavi
Mentor: doc. dr. sc. Slaven Zakošek

Zadatak: **Mobilna aplikacija za organiziranje zajedničke vožnje temeljena na podacima iz otvorenih izvora**

Opis zadatka:

Smanjivanje troškova putovanja, emisije stakleničkih plinova i prometnih zagušenja, najvažniji su motivi za proučavanje alternativnih metoda upravljanja mobilnošću. Jedna od takvih metoda je zajedničko korištenje vozila, odnosno organiziranje zajedničke vožnje (engl. carpooling) u kojem dvije ili više osoba dogovaraju zajedničku vožnju osobnim vozilom na cijeloj ili dijelu zajedničke rute putovanja, pri čemu vozač za tu vožnju nije profesionalno angažiran. Proučiti i opisati karakteristike različitih oblika organizacije dijeljenja vožnje osobnim automobilom i glavne značajke postojećih alata za upravljanje rutama u cestovnom prometu, s naglaskom na alate koji se temelje na otvorenim izvorima podataka. Izgraditi sustav, s mobilnom aplikacijom kao sučeljem, koji će vozaču i zainteresiranim suputnicima omogućiti planiranje i dogovaranje zajedničke vožnje. Vozač opisuje osnovne elemente putovanja koje želi podijeliti (npr. rutu, vrijeme, broj slobodnih mjesta), a sustav treba prepoznati kompatibilne zahtjeve za putovanjima potencijalnih suputnika i omogućiti dogovaranje, a kasnije i međusobno ocjenjivanje zajedničke vožnje.

Rok za predaju rada: 28. lipnja 2024.

Zahvaljujem roditeljima, djevojci i prijateljima na bezuvjetnoj podršci.

Sadržaj

| | |
|---|----|
| Uvod | 1 |
| 1. Koncept zajedničke vožnje | 2 |
| 1.1. Definicija zajedničke vožnje | 2 |
| 1.2. Povijest i razvoj zajedničke vožnje | 2 |
| 1.3. Prednosti i izazovi zajedničkog korištenja vozila..... | 3 |
| 1.3.1. Prednosti | 3 |
| 1.3.2. Izazovi | 4 |
| 1.4. Pregled postojećih aplikacija za organizaciju zajedničke vožnje | 4 |
| 2. Analiza problema i motivacija rješenja | 7 |
| 2.1. Trenutno stanje u organizaciji zajedničke vožnje u Hrvatskoj..... | 7 |
| 2.2. Motivacija za razvoj novog rješenja za organizaciju zajedničke vožnje..... | 8 |
| 3. Tehnologije za usmjeravanje i planiranje ruta..... | 9 |
| 3.1. Otvoreni podaci | 9 |
| 3.2. PostGIS..... | 10 |
| 3.2.1. Prostorni tipovi podataka..... | 10 |
| 3.2.2. Prostorno indeksiranje | 11 |
| 3.2.3. Prostorne funkcije..... | 11 |
| 3.3. OpenStreetMap..... | 12 |
| 3.4. Openrouteservice | 13 |
| 3.4.1. Usluga optimizacije | 14 |
| 4. Koncept rješenja | 17 |
| 4.1. Arhitektura sustava | 17 |
| 4.1.1. Poslužitelj | 18 |
| 4.2. Struktura Baze podataka..... | 18 |
| 4.2.1. Implementacija baze podataka..... | 18 |

| | | |
|--------|---|----|
| 4.2.2. | Model baze podataka | 19 |
| 4.3. | Stvaranje novog prijevoza | 23 |
| 4.4. | Algoritam za prepoznavanje kompatibilnih prijevoza..... | 24 |
| 4.4.1. | Dohvaćanje podataka iz baze | 24 |
| 4.4.2. | Dohvaćanje novih ruta koristeći Openrouteservice..... | 26 |
| 4.5. | Upravljanje rezervacijama | 27 |
| 5. | iOS Mobilna aplikacija..... | 29 |
| 5.1. | Razvoj aplikacije | 29 |
| 5.1.1. | Radni okviri korisničkog sučelja | 29 |
| 5.1.2. | Pohrana podataka..... | 30 |
| 5.1.3. | Prikazivanje karata | 30 |
| 5.2. | Pregled glavnih funkcionalnosti | 32 |
| 5.2.1. | Korisnički račun i recenzije..... | 32 |
| 5.2.2. | Stvaranje prijevoza | 34 |
| 5.2.3. | Rezervacija prijevoza | 35 |
| 5.2.4. | Obavijesti..... | 36 |
| 6. | Testiranje rješenja..... | 38 |
| 6.1. | Točnost prepoznavanja kompatibilnih vožnji..... | 38 |
| 6.2. | Trajanje dohvaćanja kompatibilnih vožnji | 42 |
| | Zaključak | 44 |
| | Literatura | 45 |
| | Sažetak..... | 46 |
| | Summary..... | 47 |

Uvod

Ovaj diplomski rad istražuje važnost i primjenu zajedničke vožnje kroz razvoj mobilne aplikacije koja olakšava organizaciju i provedbu takvih putovanja. Zajednička vožnja (engl. *carpooling*), predstavlja sustav u kojem više osoba putuje zajedno u istom vozilu u istom smjeru, što donosi koristi kako vozačima, tako i putnicima. Glavne prednosti uključuju smanjenje troškova putovanja za sve sudionike, smanjenje ekološkog utjecaja putovanja, smanjenje broja vozila na cesti te povećanje dostupnosti slobodnih parkirnih mjesta.

Razvijena je mobilna aplikacija s ciljem olakšavanja organizacije zajedničke vožnje. Jedan od ključnih izazova s kojim se aplikacija suočava je pružanje fleksibilnosti u odabiru ruta, omogućujući putnicima da se pridruže vožnji iz lokacija koje se nalaze izvan glavne rute putovanja. Ova fleksibilnost postiže se podešavanjem tolerancije za prihvaćanje dodatnih putnika, što omogućuje vozačima da planiraju svoje rute uz minimalan dodatni napor.

Povećanje sigurnosti putnika tijekom vožnje je još jedan ključni aspekt aplikacije, postignut uvođenjem sustava recenziranja. Recenzije pridonose stvaranju povjerenja unutar zajednice korisnika aplikacije te osiguravaju kvalitetno iskustvo svim sudionicima zajedničke vožnje.

Poslužitelj je razvijen koristeći arhitekturu sustava temeljenu na Spring Boot radnom okviru. Podaci se pohranjuju u PostgreSQL bazu podataka, dok se za lokalnu pohranu podataka na mobilnom uređaju koristi Realm radni okvir. Mobilna aplikacija razvijena je za iOS platformu koristeći odgovarajuće Appleove alate.

Za usklađivanje ruta i rješavanje problema usmjeravanja vozila koristi se Openrouteservice API (engl. *Application Programming Interface, API*), skup servisa otvorenog koda, čiji je zadatak stvaranje optimalnih ruta. Osim toga, primijenjeno je i PostGIS proširenje za bazu podataka koje omogućuje korištenje prostornih funkcionalnosti. Prikaz karata u aplikaciji ostvaren je kroz integraciju s Mapbox sustavom, koji većinu podataka dohvaća iz otvorenih izvora, pružajući korisnicima intuitivno i ugodno iskustvo korištenja aplikacije.

1. Koncept zajedničke vožnje

1.1. Definicija zajedničke vožnje

Zajednička vožnja (engl. *carpooling*) je način dijeljenja prijevoza pri kojem više osoba putuje u jednom automobilu, čime se smanjuje broj pojedinačnih vožnji do određene lokacije.

Razlikuje se od usluga naručivanja vožnje (engl. *ridesharing*), gdje ljudi plaćaju vožnju profesionalnom ili honorarnom vozaču putem aplikacije. Naime, u zajedničkoj vožnji vozač nije profesionalac, već privatna osoba koja pristaje podijeliti svoje putovanje (ili dio njega) besplatno ili uz doprinos troškovima putovanja. Osim toga, zajednička vožnja razlikuje se i od dijeljenja automobila (engl. *carsharing*), sheme u kojoj ljudi dobivaju pristup automobilu koji ne posjeduju na određeno vrijeme[1].

Dijeljenje vožnje smanjuje troškove putovanja za svaku osobu jer se dijele troškovi goriva, cestarine i potrošnje vozila. Također, dijeljenje vožnji je ekološki prihvatljiviji i održiviji način putovanja jer smanjuje zagađenje zraka, emisiju ugljika, prometne gužve na cestama i potrebu za parkirnim mjestima.

Zajednička vožnja može obuhvaćati i svakodnevne vožnje koje uključuju članove iste obitelji ili kućanstva. Međutim, u ovom radu naglasak je na zajedničkim vožnjama koje uključuju pojedince koji se ne poznaju.

1.2. Povijest i razvoj zajedničke vožnje

Zajednička vožnja ima dugu povijest koja seže unatrag do početka 20. stoljeća. Prvi oblici zajedničke vožnje pojavili su se tijekom Prvog svjetskog rata kada su ljudi dijelili vožnje kako bi uštedjeli resurse. Međutim, ovaj način prijevoza dobio je veći zamah tijekom Drugog svjetskog rata kada je američka vlada pokrenula veliku kampanju za poticanje dijeljenja vožnji kako bi se štedjeli gorivo i materijali[2].

Nakon rata, popularnost zajedničke vožnje padala je sve do naftne krize 1970-ih godina kada je ponovno doživjela procvat. Tada su vladine mjere potaknule ljude na korištenje zajedničkog prijevoza kako bi se smanjila ovisnost o fosilnim gorivima. S vremenom su

mnogi gradovi uspostavili posebne trake za vozila s više putnika kako bi dodatno potaknuli ovu praksu.

Unatoč promjenama u popularnosti, zajednička vožnja ponovno dobiva na značaju u modernom dobu, osobito uz pomoć tehnologije i pametnih aplikacija koje olakšavaju pronalaženje vozača i putnika.

Prije globalne popularnosti pametnih telefona, glavni izvor organiziranja zajedničke vožnje bile su društvene mreže poput Facebooka. Korisnici su osnivali grupe za često tražene rute i dogovarali se o vremenu polaska, broju slobodnih mjesta u automobilu i cijeni prijevoza u komentarima objava. Popularizacija mobilnih aplikacija omogućila je korisnicima da organiziraju zajedničke vožnje lakše nego ikada prije. Platforma BlaBlaCar, koja je detaljnije opisana u poglavlju 1.4, brzo je postala najkorišteniji način organiziranja dijeljenih prijevoza. Aplikacija se istaknula zbog jednostavnosti kreiranja novih vožnji i odabira željenih prijevoza u Hrvatskoj.

Danas su korisnici zajedničkih vožnji najčešće osobe koje rade izvan mjesta prebivališta, često u područjima s visokim koncentracijama radnih mjesta, te studenti koji pohađaju visokoškolske ustanove u većim gradovima. Radnici koji svakodnevno putuju na posao mogu lakše podijeliti troškove putovanja, dok je studentima zajednička vožnja ekonomičnija opcija u usporedbi s drugim oblicima prijevoza.

1.3. Prednosti i izazovi zajedničkog korištenja vozila

1.3.1. Prednosti

- Glavna prednost dijeljenja prijevoza, često i glavni motiv, jest financijska ušteda. Ovisno o dogovoru između vozača i putnika, trošak vožnje može biti i do dvostruko manji u usporedbi s tradicionalnim oblicima javnog prijevoza.
- Ušteda vremena često je značajna prednost jer je vožnja automobilom obično brža u usporedbi s vožnjom autobusom ili vlakom, ovisno o vrsti putovanja.
- Automobil ne predstavlja samo trošak goriva, već uključuje i troškove održavanja i popravka vozila. Smanjenjem vremena korištenja automobila dodatno se smanjuju ti troškovi.
- Izbjegavanje problema s parkiranjem: smanjen broj vozila na cestama rezultira većim brojem slobodnih parkirnih mjesta.

- Alternativni načini prijevoza u područjima sa slabo razvijenim uslugama javnog prijevoza mogu značajno poboljšati mobilnost stanovništva i smanjiti njihovu ovisnost o osobnim vozilima.
- Smanjenje broja automobila na cestama može znatno smanjiti emisije CO₂ i poboljšati kvalitetu zraka.
- Upoznavanje novih ljudi može smanjiti dosadu i učiniti putovanje zabavnijim.

1.3.2. Izazovi

- Vožnja u zajedničkom prijevozu sa strancima povećava zabrinutost za sigurnost i predstavlja prepreku dijeljenju vozila s nepoznatim osobama.
- Nepristojni putnici mogu pokušati smanjiti cijenu ili čak inzistirati da vozač posjeti mjesto koje nije na planiranom putu.
- Nepristojni vozači također mogu stvarati probleme, na primjer vozači koji prevoze više osoba u malom automobilu i traže visoku cijenu, ili vozači koji često rade neplanirana zaustavljanja i tako produljuju trajanje putovanja.
- U nekim slučajevima vozač mora pokupiti svakog suputnika posebno, što dodatno povećava vrijeme putovanja.
- Fleksibilnost također može biti problem jer može biti vrlo teško prilagoditi zaustavljanja tijekom vožnje.
- Dostupnost je još jedan izazov zajedničkih vožnji. Zajednički prijevoz zahtijeva mnogo sudionika kako bi sustav funkcionirao, šanse za pronalazak suputnika za određeno (ili slično) putovanje su male i zahtijevaju značajan broj aktivnih korisnika[3].

1.4. Pregled postojećih aplikacija za organizaciju zajedničke vožnje

Broj aplikacija za organizaciju zajedničkih vožnji značajno je porastao, osobito s pojavom mobilnih aplikacija koje omogućuju brže pregledavanje dostupnih prijevoza, poboljšavaju komunikaciju među korisnicima i pružaju obavijesti o vožnjama. Najpopularnija aplikacija za organizaciju zajedničke vožnje u Hrvatskoj je BlaBlaCar. Osim BlaBlaCar, ukratko su opisane i aplikacija Liftshare, najpopularnija aplikacija u Ujedinjenom Kraljevstvu, i aplikacija Prevoz.org, najpopularnija aplikacija u Sloveniji. Ove aplikacije nude različite

funkcionalnosti i pristupe organizaciji zajedničkih vožnji, pružajući korisnicima niz opcija za optimalno planiranje putovanja.

BlaBlaCar (<https://www.blablacar.hr>)

BlaBlaCar je jedna od najpopularnijih platformi za organiziranje zajedničkih vožnji, pružajući korisnicima jednostavan način povezivanja s vozačima i putnicima koji dijele iste rute. Osnovan 2006. godine u Francuskoj, BlaBlaCar je brzo stekao popularnost zahvaljujući svojoj intuitivnoj aplikaciji koja omogućava lako pronalaženje i rezerviranje vožnji. Platforma je dizajnirana kako bi smanjila troškove putovanja, omogućila ekološki prihvatljivije putovanje te povećala društvenu interakciju među korisnicima.

BlaBlaCar koristi algoritme za uparivanje vozača i putnika na temelju njihovih odredišta, vremena polaska i preferencija tijekom putovanja. Korisnici mogu ocijeniti jedni druge, što povećava sigurnost i pouzdanost sustava. Platforma također nudi opcije za filtriranje rezultata prema različitim kriterijima, poput cijene i udobnosti vožnje.

BlaBlaCar platforma trenutno je dostupna u 21 državi diljem svijeta, a u većini je postala najpopularnija aplikacija za organizaciju zajedničkih vožnji. To potvrđuje i broj aktivnih korisnika kojih je više od 26 milijuna. Osim usluge dijeljenja prijevoza, BlaBlaCar u nekim zemljama nudi i opciju putovanja autobusima, što predstavlja prikladnu alternativu u slučajevima kada nema dostupnih vožnji na željenoj ruti korisnika.

U Hrvatskoj, BlaBlaCar je postao omiljeni izbor za one koji traže povoljan i praktičan način putovanja. Aplikacija omogućava korisnicima brzo kreiranje novih vožnji, odabir željenih prijevoza te primanje obavijesti o dostupnim vožnjama.

Liftshare (<https://liftshare.com/uk>)

Liftshare je vodeća aplikacija za organiziranje zajedničkih vožnji u Ujedinjenom Kraljevstvu, osnovana s ciljem smanjenja troškova putovanja i negativnih utjecaja na okoliš. Platforma omogućuje korisnicima da dijele vožnje i tako uštede novac, te uz to i smanje prometne gužve i emisije CO₂. Liftshare nudi širok raspon funkcionalnosti koje olakšavaju planiranje i koordinaciju zajedničkih vožnji.

Korisnici mogu pretraživati dostupne vožnje na temelju odredišta, datuma i vremena polaska. Platforma također nudi mogućnost pregleda profila vozača i putnika te uključuje sustav ocjenjivanja, što povećava sigurnost i pouzdanost usluge. Liftshare potiče stvaranje

zajednica unutar tvrtki, škola i drugih organizacija, čime olakšava pronalaženje suputnika među poznatim osobama.

Aplikacija Liftshare je jednostavna za korištenje, a njezine napredne funkcionalnosti omogućuju korisnicima da lako organiziraju zajedničke vožnje, bilo da se radi o svakodnevnim putovanjima na posao ili duljim putovanjima.

Prevoz.org (<https://prevoz.org>)

Prevoz.org je najpopularnija aplikacija za organiziranje zajedničkih vožnji u Sloveniji. Osnovana s ciljem povezivanja vozača i putnika koji dijele iste rute, Prevoz.org pruža jednostavan i učinkovit način smanjenja troškova putovanja i negativnih utjecaja na okoliš. Platforma omogućava korisnicima da brzo i lako pronađu dostupne vožnje, dogovore detalje putovanja i podijele troškove.

Prevoz.org je dizajniran kako bi olakšao organizaciju zajedničkih vožnji kroz intuitivno sučelje koje omogućava pretraživanje vožnji prema odredištu, datumu i vremenu polaska. Korisnici mogu pregledavati profile vozača i putnika, uključujući njihove ocjene i recenzije, što povećava sigurnost i pouzdanost sustava.

Platforma je posebno popularna među studentima i radnicima koji često putuju između gradova, omogućujući im ekonomičan i održiv način putovanja.

2. Analiza problema i motivacija rješenja

2.1. Trenutno stanje u organizaciji zajedničke vožnje u Hrvatskoj

U Hrvatskoj, BlaBlaCar je trenutno jedina značajna platforma za organizaciju zajedničke vožnje. Iako je BlaBlaCar popularan i pruža korisne usluge, postoji nekoliko ključnih izazova i ograničenja koja utječu na korisničko iskustvo. Jedan od glavnih problema je nedostatak alternativnih platformi koje bi mogle ponuditi različite opcije i dodatne pogodnosti korisnicima. Ovaj monopolistički položaj može ograničiti konkurenciju i inovacije u sektoru dijeljenih prijevoza.

Osim toga, BlaBlaCar je najavio uvođenje naplaćivanja rezervacija prijevoza preko platforme, što mnogi korisnici doživljavaju kao nepošteno. Ova promjena može dodatno povećati troškove putovanja i obeshrabriti ljude od korištenja zajedničke vožnje. Dodatni troškovi mogu utjecati na smanjenje broja korisnika, što bi moglo dovesti do manjeg broja dostupnih vožnji i otežati organizaciju putovanja.

Trenutno stanje organizacije zajedničke vožnje u Hrvatskoj suočava se s nekoliko ključnih izazova koji utječu na širenje i uspjeh platformi za organizaciju zajedničkih vožnji. Jedan od glavnih problema je stvaranje i održavanje velikog broja aktivnih korisnika. Organizacija zajedničkih vožnji zahtijeva velik broj vozača i putnika kako bi sustav bio funkcionalan i koristan za sve sudionike. Ovaj izazov može se riješiti reklamiranjem i promocijom aplikacije, kako bi se privukli novi korisnici.

Pored toga, reputacija aplikacije igra važnu ulogu u privlačenju novih korisnika. Pozitivna iskustva postojećih korisnika, kao i povratne informacije o sigurnosti i pouzdanosti vozača i putnika, mogu značajno utjecati na izgled aplikacije u javnosti.

Isto tako, edukacija korisnika o prednostima zajedničke vožnje i ekološkim benefitima može pomoći u povećanju svijesti i poticanju ljudi na sudjelovanje u ovoj održivoj praksi.

2.2. Motivacija za razvoj novog rješenja za organizaciju zajedničke vožnje

Motivacija za razvoj nove aplikacije za organizaciju zajedničke vožnje u Hrvatskoj proizlazi iz potrebe za pružanjem alternative postojećim opcijama. Nova aplikacija dizajnirana je tako da odgovara na potrebe korisnika te rješava postojeće probleme i izazove s kojima se susreću.

Jedan od glavnih ciljeva nove aplikacije je omogućiti fleksibilnije stvaranje ruta, odnosno prilagođavanje ruta ovisno o potrebama putnika. Ovo će omogućiti vozačima da pakupe putnike čije lokacije su izvan početne rute, čime će se maksimalno iskoristiti kapacitet vozila i smanjiti nepotrebna putovanja. Prilagođavanje ruta može značajno povećati efikasnost prijevoza i smanjiti troškove putovanja.

Nova aplikacija također će se fokusirati na sigurnost korisnika. Potrebno je uvesti sustav ocjenjivanja korisnika nakon obavljene zajedničke vožnje, što će pomoći u izgradnji povjerenja među korisnicima i omogućiti bolju procjenu sigurnosti prije sklapanja dogovora o prijevozu. Također, korisnici će imati mogućnost postavljanja profilnih fotografija, što će dodatno povećati transparentnost i sigurnost.

Osim toga, cilj novog rješenja za organizaciju zajedničke vožnje je poticanje ekološki prihvatljivijeg oblika prijevoza, smanjivanja emisija CO₂ i prometnih gužva.

3. Tehnologije za usmjeravanje i planiranje ruta

Usmjeravanje ili planiranje ruta, je proces određivanja optimalnog puta kojim se treba kretati kako bi se postigao određeni cilj. Ovaj proces koristi različite algoritme i metode za izračunavanje najkraće ili najefikasnije rute između dviju ili više točaka. Usmjeravanje se široko primjenjuje u logistici, transportu, navigaciji i dostavi, gdje je potrebno precizno planirati putanje za svakakve vrste vozila. Planiranje ruta omogućava učinkovito korištenje resursa, smanjuje vrijeme putovanja i operativne troškove.

3.1. Otvoreni podaci

Otvoreni podaci su digitalni podaci dostupni javnosti. Otvoreni podaci imaju takve tehničke i pravne karakteristike da ih bilo tko, bilo kada i bilo gdje može koristiti, ponovo koristiti i redistribuirati. Opći cilj otvorenih podataka je poticanje i jačanje suradnje koja vodi k usvajanju i implementaciji zajedničkih principa, standarda i dobrih praksi[4].

Otvoreni podaci omogućuju veću transparentnost i odgovornost, što znači da su pristup informacijama i praćenje rada vlada i institucija uvelike olakšani. Osim toga, potiču inovacije i ekonomski rast jer poduzetnici i istraživači mogu koristiti te podatke za stvaranje novih proizvoda i usluga.

Principi otvorenih podataka

Kako bi se podaci smatrali otvorenima, oni trebaju biti objavljeni na način koji je u skladu sa sljedećim principima[4]:

- **Cjelovitost:** Sve javne informacije su dostupne i ne podliježu primjenjivim ograničenjima privatnosti, sigurnosti ili povjerljivosti.
- **Izvornost:** Podaci se prikupljaju na izvoru s najvišim stupnjem detaljnosti.
- **Pravovremenost:** Podaci su dostupni što je prije moguće kako bi se očuvala njihova vrijednost.
- **Pristupačnost:** Podaci su dostupni najširem krugu korisnika za najširi raspon upotreba.
- **Strojna čitljivost:** Podaci su razumno strukturirani kako bi omogućili automatsku obradu.
- **Nediskriminacija:** Podaci su dostupni svima bez potrebe za registracijom.

- **Bez vlasništva:** Podaci su dostupni u obliku nad kojim nitko nema ekskluzivnu kontrolu.
- **Bez licence:** Podaci nisu podložni autorskom pravu, patentu, zaštitnom znaku ili poslovnoj tajni. Razumno ograničenje privatnosti, sigurnosti i povjerljivosti može biti dopušteno.

3.2. PostGIS

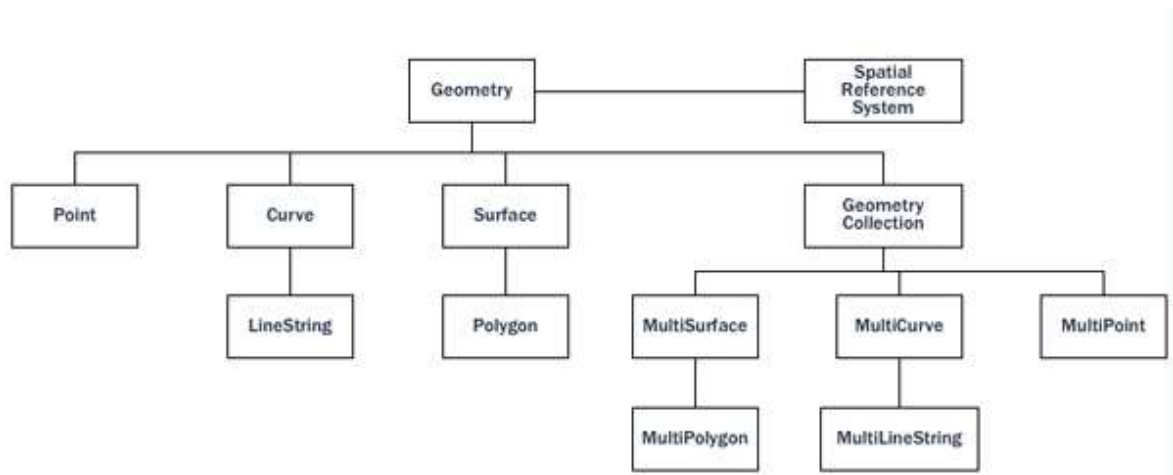
PostGIS je prostorna baza podataka, tj. baza podataka koja pohranjuje i manipulira prostornim objektima kao i bilo kojim drugim objektima u bazi podataka. PostGIS je proširenje za PostgreSQL bazu podataka. Budući da je izgrađen na PostgreSQL-u, PostGIS automatski nasljeđuje važne napredne funkcionalnosti kao i otvorene standarde za implementaciju.

PostGIS pretvara PostgreSQL u prostornu bazu podataka dodavanjem podrške za tri značajke: prostorne tipove, prostorne indekse i prostorne funkcije.

3.2.1. Prostorni tipovi podataka

Obična baza podataka sadrži tekstualne podatke, brojeve i datume. Prostorna baza podataka dodaje prostorne tipove podataka kao što su točka (engl. *point*), linija (engl. *linestring*), poligon (engl. *polygon*), i omogućuje pohranjivanje i obradu geografski distribuiranih podataka[5]. Ti prostorni tipovi podataka apstrahiraju i uključuju prostorne strukture poput granica i dimenzija. U mnogim pogledima, prostorni tipovi podataka se mogu jednostavno shvatiti kao geometrijski oblici.

Prostorni tipovi podataka su organizirani u hijerarhijsku strukturu, pri čemu svaki tip nasljeđuje strukturu i funkcionalnost svog nadređenog tipa ako ga ima. Slika 3.1 prikazuje hijerarhijsku strukturu prostornih tipova podataka.



Slika 3.1 Hijerarhija prostornih tipova podataka[5]

3.2.2. Prostorno indeksiranje

Obični indeksi u bazama podataka su strukture koje se koriste za ubrzanje pretraga i filtriranja podataka. Indeksiranje za standardne tipove obično je jednostavno jer svaka vrijednost je manja od, veća od ili jednaka svakoj drugoj vrijednosti.

No, zbog toga što se poligoni mogu preklapati, mogu biti sadržani jedan u drugome i raspoređeni u dvodimenzionalnom (ili više) prostoru, prostorne baze podataka pružaju mogućnost korištenja prostornih indeksa. Minimalni ograničavajući pravokutnik (engl. *minimal bounding rectangle, MBR*) je najmanji pravokutnik, paralelan s koordinatnim osima, koji može obuhvatiti određenu značajku. Prostorni indeksi omogućuju pretragu tako da provjeravaju koji objekti se nalaze unutar određenog MBR-a, umjesto da uspoređuju vrijednosti.

Minimalni ograničavajući pravokutnici se koriste jer je odgovor na pitanje "je li A unutar B?" vrlo računalno zahtjevan za poligone, ali vrlo brz u slučaju MBR-a. Sve vrste poligona, čak i najkompleksniji poligoni i linije mogu se prikazati jednostavnim MBR-om, no to znači da će indeksi pružati približne rezultate[5]. Brzina dobivena korištenjem prostornih indeksa dovoljna je za opravdanje dobivanja približnih rezultata.

3.2.3. Prostorne funkcije

Osim mogućnosti stvaranja podataka prostornog tipa i odgovarajućih prostornih indeksa, PostGIS pruža i mogućnost korištenja prostornih funkcija za manipuliranje podacima.

PostGIS pruža potpuni skup funkcija za analizu geometrijskih komponenti, određivanje prostornih odnosa i manipulaciju geometrijama.

Prostorne funkcije služe kao temelj za svaki prostorni projekt, a podijeljene su u pet kategorija[5]:

- **Transformacija:** Funkcije koje transformiraju podatke između geometrija i vanjskih formata podataka.
- **Upravljanje:** Funkcije koje upravljaju informacijama o prostornim tablicama i administraciji PostGIS-a.
- **Dohvaćanje:** Funkcije koje dohvaćaju svojstva i mjerenja geometrije.
- **Usporedba:** Funkcije koje uspoređuju dvije geometrije s obzirom na njihov prostorni odnos.
- **Generiranje:** Funkcije koje generiraju nove geometrije iz drugih.

Posebno je interesantna funkcija `ST_DWithin`, koja se koristi u algoritmu za prepoznavanje kompatibilnih prijevoza i pripada kategoriji prostornih funkcija za usporedbu. Ova funkcija provjerava nalaze li se dvije dane geometrije unutar definirane udaljenosti. Iz optimizacijskih razloga funkcija uključuje usporedbu minimalnih ograničavajućih pravokutnika za sve prostorne indekse koji su dostupni.

`ST_DWithin` provjerava zračnu udaljenost tih geometrija tako da prvo uspoređi udaljenosti MBR-a geometrija kako bi se brzo isključile geometrije koje su previše udaljene. PostGIS izvodi precizniji izračun pravih geometrija za parove čiji se minimalni ograničavajući pravokutnici nalaze unutar specificirane udaljenosti te tek tada određuje završnu vrijednost.

3.3. OpenStreetMap

OpenStreetMap (skraćeno OSM) je globalna inicijativa zajednice koja pruža besplatne, detaljne i prilagodljive kartografske podatke širom svijeta. OSM predstavlja projekt izgradnje besplatne geografske baze podataka svijeta s ciljem zapisivanja svake geografske značajke na planeti.

Iako je projekt započeo s kartiranjem ulica, već je znatno nadmašio tu početnu svrhu uključivanjem pješačkih staza, zgrada, vodotoka, itd. Osim fizičke geografije, projekt uključuje i administrativne granice, detalje o korištenju zemljišta, autobusne rute i druge apstraktne pojmove koji nisu očiti iz samog krajolika[6].

Potaknuti potrebom za slobodno dostupnim geografskim informacijama, OSM okuplja tisuće volontera diljem svijeta koji doprinose prikupljanjem, uređivanjem i provjerom geografskih podataka. Volonteri skupljaju informacije koristeći GPS (engl. *Global Positioning System, GPS*) prijemnike, koje se zatim koriste za stvaranje skupa točaka i linija koje se mogu pretvoriti u karte ili koristiti za navigaciju[6].

Projekt OpenStreetMap ne ograničava se samo na kartiranje, već nudi i razne druge usluge. Usmjeravanje, geokodiranje (pronalaženje koordinata za određeni objekt) i prostorna analiza su dodatne primjene podataka prikupljenih u OSM-u. OSM čak ne nameće ograničenja na izgled karata u aplikacijama, omogućujući korisnicima stvaranje vlastitih karata u bilo kojem stilu.

OSM omogućuje besplatan i neograničen pristup cijelom skupu podataka, s potpunom poviješću promjena, bez ograničenja tko i kako ih može koristiti. Podaci OSM-a objavljeni su pod licencom koja omogućuje kopiranje, mijenjanje i redistribuciju podataka.

Iako je glavni cilj projekta prikupljanje geografskih podataka, članovi projekta također su razvili širok spektar softverskih rješenja (većina ih je otvorenog koda) koji stvara, uređuje, manipulira ili koristi podatke na neki način.

3.4. Openrouteservice

Openrouteservice je besplatna usluga koja omogućuje izradu ruta, analizu vremena putovanja i izračun udaljenosti koristeći podatke iz OSM-a. Openrouteservice pruža više od jednostavnog stvaranja ruta: koristi široku paletu usluga temeljenih na podacima iz OSM-a, koje se mogu primijeniti u raznim aplikacijama i scenarijima.

Trenutno Openrouteservice API nudi sljedeće funkcionalnosti[7]:

- **Upute:** Implementacija detaljnih uputa usmjeravanja za automobile, kamione, različite profile bicikala, pješaćenje, planinarenje ili korištenje invalidskih kolica.
- **Izokrone:** Dohvaćanje izokrona koje pomažu odrediti koje područje objekti mogu doseći unutar određenih vremena ili udaljenosti.
- **Matrica vremena i udaljenosti:** Matrice omogućuju izračunavanje udaljenosti i vremena trajanja ruta između više početnih i više završnih točaka puno brže nego višestrukim korištenjem API-ja za smjernice.

- **Geokodiranje:** Geokodiranje transformira opis lokacije, kao što je naziv mjesta, adresa ili poštanski broj, u normalizirani opis lokacije s koordinatama.
- **Točke interesa:** Openpoiservice omogućuje pronalaženje zanimljivih mjesta oko ili unutar zadanih geografskih koordinata, kao što su ustanove, restorani ili prodavaonice.
- **Visina:** Dohvaćanje trodimenzionalne verzije geometrije na temelju poslanih točkaste ili linijske geometrije, tj. dohvaćanje visine neke koordinate.
- **Optimizacija:** Implementacija problema putujućeg trgovca i drugih problema s usmjeravanjem vozila. Temelji se na projektu Vroom, a ova usluga pruža optimalne rute uzimajući u obzir poslanih specifične zahtjeve.

Openrouteservice API je dostupan svima i besplatan za korištenje. Nakon registracije, korisnici mogu koristiti API za integraciju svih navedenih usluga u svoje aplikacije, čime se omogućuje široka primjena i prilagodba specifičnim potrebama različitih korisnika. Openrouteservice ima dnevni limit za korištenje (API za optimizaciju dopušta 500 pristupa dnevno u standardnoj verziji, a 2500 u suradničkoj), ali korisnici mogu i besplatno napraviti svoju instancu API-ja koja nema ograničenja.

3.4.1. Usluga optimizacije

Optimizacijski API omogućuje rješavanje različitih problema usmjeravanja, a temelji se na projektu VROOM. VROOM je optimizacijski projekt otvorenog koda koji pruža rješenja za različite stvarne probleme usmjeravanja kao što je problem trgovačkog putnika (engl. *Traveling Salesman Problem, TSP*).

Problematika trgovačkog putnika je jedan od najpoznatijih i najintenzivnije proučavanih problema u području optimizacije i teorije grafova. TSP se bavi pronalaženjem najkraće moguće rute koju mora proći trgovački putnik koji treba posjetiti određeni skup gradova i vratiti se u početni grad, pri čemu svaki grad posjećuje točno jednom. Ovaj problem je vrlo sličan problemu prikupljanja i ostavljanja putnika, te on predstavlja osnovu za stvaranje algoritma za prepoznavanje kompatibilnih prijevoza.

Optimizacijski API namijenjen je optimizaciji ruta za vozila, uključujući kamione, dostavna vozila i flote vozila, kako bi se efikasno planirali zadaci poput preuzimanja i isporučivanja. Definiranjem više vozila rezultat će biti optimiziran za cijelu flotu, a zadaci mogu biti definirani kao poslovi, koji su jednokratna preuzimanja ili isporuke.

U slučaju stvaranja ruta za zajedničke vožnje, flota se sastoji od samo jednog vozila, a zadaci označuju lokacije ukrcavanja i iskrcavanja putnika.

Kod 3.1 je primjer tijela HTTP (engl. *Hypertext Transfer Protocol, HTTP*) zahtjeva prema optimizacijskom API-ju Openrouteservice-a. U tijelu je definirano jedno vozilo koje sadrži početnu i završnu lokaciju putovanja, odnosno polazište i destinaciju vozača. Polje `jobs` predstavlja zadatke, što je u ovom slučaju popis lokacija na kojima se putnici ukrcavaju i iskrcavaju. Postavljanje opcije `"g": true` uključuje opciju koja u tijelo odgovora zapisuje šifriranu geometriju rute, što služi za prikazivanje te rute na ekranu.

HTTP odgovor sadrži raznolike korisne informacije, među kojima je ključan popis ruta. Openrouteservice pronalazi optimalnu rutu, a ako su dostupne i alternativne rute. Svaka ruta obuhvaća šifriranu geometriju, vremensko trajanje putovanja u sekundama, udaljenost putovanja u metrima te detaljan niz korak po korak uputa koje opisuju kako reproducirati putovanje.

```
{
  "jobs": [
    {
      "id": 1,
      "location": [
        16.05233,
        45.80290
      ]
    },
    {
      "id": 2,
      "location": [
        16.42007,
        46.38772
      ]
    }
  ],
  "vehicles": [
    {
      "id": 1,
      "profile": "driving-car",
      "start": [
        15.96268,
```

```
        45.79995
      ],
      "end": [
        16.42761,
        46.38680
      ]
    }
  ],
  "options": {
    "g": true
  }
}
```

Kod 3.1 Tijelo HTTP zahtjeva prema optimizacijskom API-ju

4. Koncept rješenja

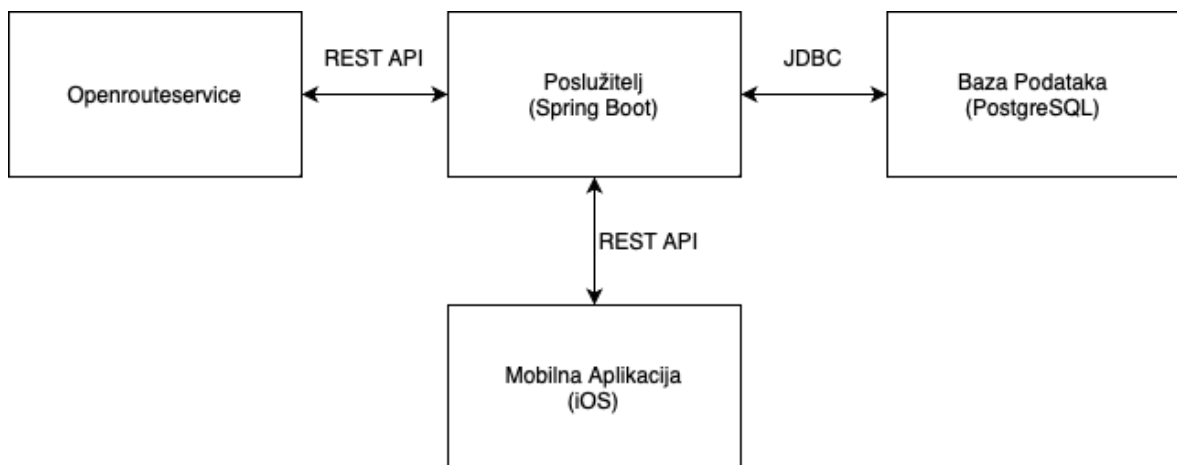
4.1. Arhitektura sustava

Arhitektura sustava temelji se na korištenju Spring Boot radnog okvira koji s pomoću (engl. *Java Database Connectivity, JDBC*) API-ja ostvaruje vezu s PostgreSQL bazom podataka. JDBC omogućuje učinkovitu komunikaciju između poslužitelja i baze podataka, omogućujući upravljanje podacima i izvođenje upita.

Paralelno s tim, poslužitelj komunicira s mobilnom aplikacijom na iOS platformi putem REST API-ja i HTTP zahtjeva. Ovaj sloj omogućuje mobilnoj aplikaciji pristup resursima na poslužitelju, što uključuje čitanje, pisanje i ažuriranje podataka.

Za optimizaciju ruta i rješavanje problema usmjeravanja vozila, sustav koristi Openrouteservice API. Ova integracija omogućuje sustavu da planira optimalne rute za vozila, uzimajući u obzir polazišta i odredišta vozača i putnika.

Slika 4.1 prikazuje arhitekturu sustava, jasno ilustrirajući međusobne veze između poslužitelja, baze podataka, mobilne aplikacije i Openrouteservice API-ja.



Slika 4.1 Arhitektura sustava

4.1.1. Poslužitelj

Implementacija poslužitelja za aplikaciju za organiziranje zajedničke vožnje izvedena je korištenjem Java Spring Boot i Spring Web tehnologija.

Spring Boot olakšava pokretanje, razvoj i testiranje aplikacija uz minimalno kodiranje i konfiguraciju. Spring Web modul unutar Spring radnog okvira se koristi za razvoj web aplikacija i servisa, omogućavajući stvaranje RESTful web servisa koji se lako integriraju s različitim klijentima, uključujući mobilne aplikacije. Koristeći Spring Web, definirane su krajnje točke za različite funkcionalnosti aplikacije za dijeljenje prijevoza, kao što su registracija korisnika, kreiranje vožnji i dohvaćanje obavijesti.

Spring Boot poslužitelj postavljen je na Render, servis za izgradnju i pokretanje aplikacija, web stranica i baza podataka u oblaku. Postupak postavljanja Spring Boot aplikacije na Render platformu uključuje izradu Docker slike aplikacije koja definira okruženje i konfiguraciju potrebnu za izvršavanje. Render automatski dodjeljuje URL (engl. *Uniform Resource Locator, URL*) aplikaciji te omogućuje skaliranje i upravljanje resursima putem weba.

4.2. Struktura Baze podataka

4.2.1. Implementacija baze podataka

Implementacija baze podataka za aplikaciju za organiziranje zajedničke vožnje koristi PostgreSQL, PostgreSQL Administration (pgAdmin) i PostGIS. Ova kombinacija pruža robusnu, skalabilnu i geografski svjesnu bazu podataka koja je ključna za rukovanje podacima i funkcionalnostima aplikacije.

PostgreSQL je moćan, otvoreni sustav za upravljanje objektno-relacijskim bazama podataka, poznat je po svojoj pouzdanosti, integritetu podataka, bogatom skupu značajki i proširivosti. Podržava sve glavne operativne sustave te svojstva atomarnosti, konzistentnosti, izolacije i izdržljivosti (engl. *Atomicity, Consistency, Isolation, Durability, ACID*), što osigurava sigurnost i pouzdanost podataka[8].

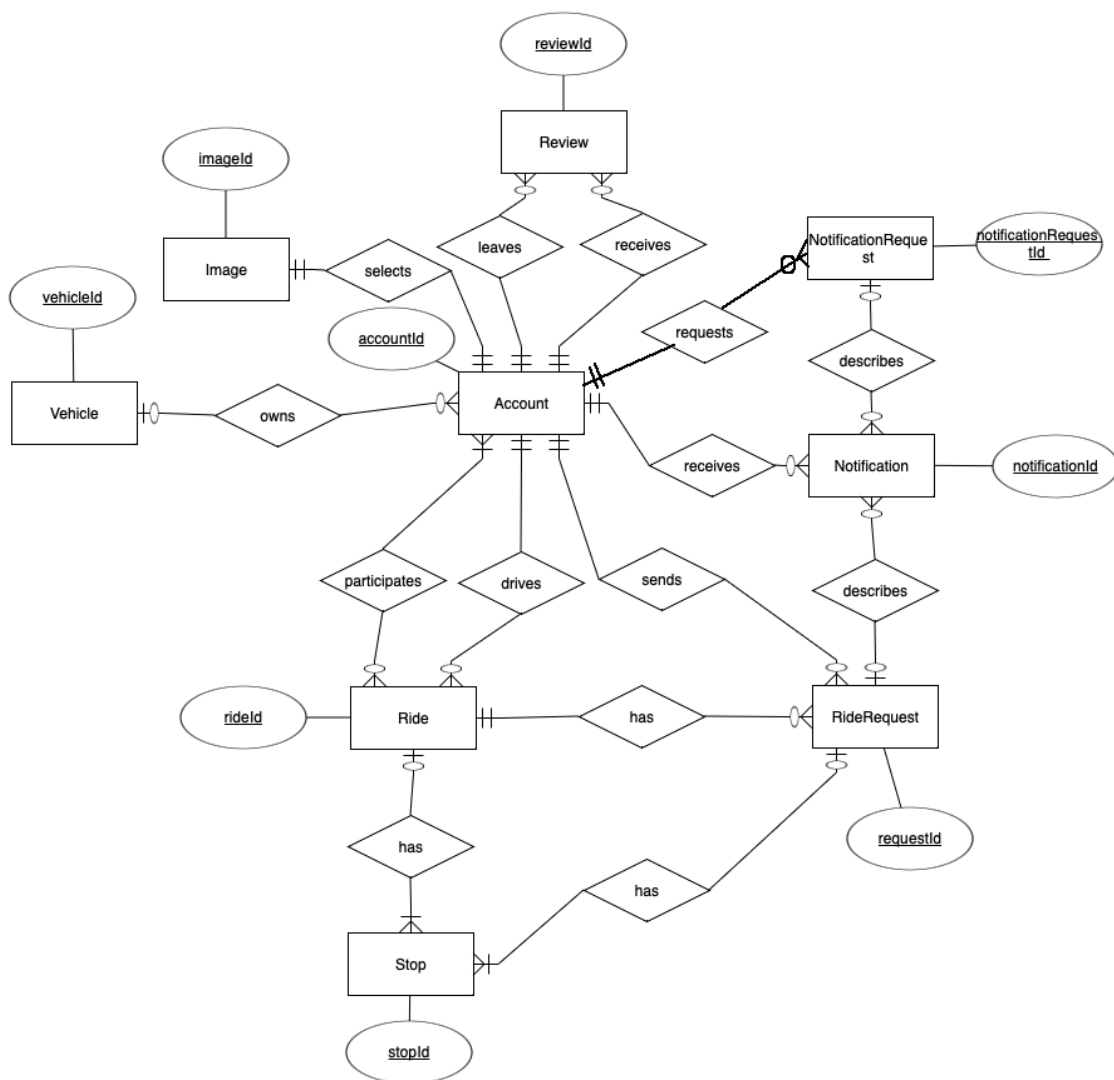
PgAdmin je popularan alat za administraciju PostgreSQL-a, koji pruža grafičko sučelje za upravljanje bazom podataka. Korisnici mogu lako kreirati, pregledavati i modificirati tablice, indekse i ostale objekte baze podataka.

Tablice su kreirane koristeći SQL skripte koje definiraju strukturu baze podataka. Svaka tablica ima primarni ključ za jedinstvenu identifikaciju zapisa i, gdje je potrebno, strane ključeve za povezivanje s drugim tablicama.

PostgreSQL baza podataka je također postavljena na Render. Proces uključuje kreiranje PostgreSQL instance, migraciju podataka i konfiguraciju aplikacije za povezivanje s bazom.

4.2.2. Model baze podataka

ER (engl. *Entity-Relationship*) dijagram je alat za semantičko modeliranje podataka koji služi za apstraktno opisivanje ili prikazivanje podataka[9]. Slika 4.2 prikazuje ER dijagram baze podataka.



Slika 4.2 ER dijagram baze podataka

Account

Tablica `Account` predstavlja korisnički račun i sadrži osnovne podatke o korisnicima. Primarni ključ tablice je stupac `account_id` tipa znak (duljine 36), koji se postavlja pri kreiranju novog zapisa korisnika kao tekstualna reprezentacija univerzalnog jedinstvenog identifikatora (engl. *Universally unique identifier, UUID*). Tablica uključuje osnovne stupce za opis korisnika i njegovog računa, kao što su `name`, `email` i `description`.

Osim ovih osnovnih atributa, tablica sadrži i dva strana ključa: `image_id` i `vehicle_id`. Stupac `image_id` povezuje tablicu s tablicom `Image` stvarajući vezu jedan-prema-jedan (1:1), dok stupac `vehicle_id` povezuje tablicu s tablicom `Vehicle` stvarajući vezu više-prema-jedan (M:1), označavajući korisnikovu profilnu sliku i odabrano vozilo.

Review

Tablica `Review` sadrži recenzije i ocjene koje korisnici ostavljaju jedni drugima. Svaka recenzija sadrži naslov, tekst recenzije i ocjenu (od 1 do 5), predstavljene stupcima `title`, `comment` i `rating`.

Stupac `review_id` je primarni ključ tipa broj koji se automatski generira koristeći strategiju inkrementalne identifikacije.

Povezivanje recenzija s korisnicima obavlja se putem dviju više-prema-jedan (M:1) veza, označenih stupcima `reviewer_id` i `account_id`, koji su strani ključevi tablice `Account`. Stupac `reviewer_id` označava korisnika koji ostavlja recenziju, dok stupac `account_id` označava korisnika koji prima recenziju.

Ride

Tablica `Ride` sadrži sve podatke potrebne za održavanje jedne vožnje, uključujući datum i vrijeme početka, cijenu, broj ukupnih i zauzetih mjesta u vozilu, duljinu i trajanje vožnje.

Primarni ključ je stupac `ride_id`, koji pohranjuje tekstualnu reprezentaciju UUID-a.

Tablica `Ride` ima dvije veze s tablicom `Account`: prva je veza više-prema-jedan (M:1) predstavljena stupcem `driver_id` koji označava vozača, a druga veza je više-prema-više (M:N) koja predstavlja putnike i implementirana je kroz tablicu `Participates`.

Osim osnovnih atributa, `Ride` uključuje i dva stupca prostornih tipova. Stupac `encoded_polyline` sprema šifriranu poliliniiju kao tekst, a stupac `route` pohranjen je

kao geometrijski tip podataka `LineString` iz proširenja `PostGIS`. Geometrijski tip `LineString` predstavlja put između lokacija koji ima oblik uređenog niza od dvije ili više točaka. Stupac `route` predstavlja stvarnu rutu kojom će vožnja prolaziti.

Stupac `route` dodatno je opremljen prostornim indeksom radi optimizacije performansi upita koji koriste prostorne funkcije.

RideRequest

Tablica `RideRequest` predstavlja zahtjev putnika za sudjelovanjem u vožnji, odnosno za osiguranje mjesta u vozilu. Tablica sadrži podatke o broju putnika za koje vrijedi rezervacija u stupcu `passenger_count` i trenutni status zahtjeva u stupcu `status`. Status može biti "PENDING" ako zahtjev još nije odgovoren ili "ACCEPTED" ako je vozač odobrio rezervaciju. Odbijeni zahtjevi se brišu iz tablice.

Primarni ključ je stupac `request_id`, automatski generiran korištenjem strategije inkrementalne identifikacije.

`RideRequest` ima više-prema-jedan (M:1) vezu s tablicom `Account` preko stupca `passenger_id`, koji označava korisnički račun putnika, te jedan-prema-jedan (1:1) vezu s tablicom `Ride` preko stupca `ride_id`, koja predstavlja vožnju za koju putnik želi rezervirati mjesta.

Stop

Tablica `Stop` označava postaju na karti koja može pripadati ili vožnji ili zahtjevu za rezervaciju sjedala. Svaki zapis ima vrstu postaje zapisanu u stupcu `type`, gdje vrste "START" i "END" označavaju početnu i završnu lokaciju vožnje koju postavlja vozač, dok vrsta "STOP" označava dodatnu postaju za prikupljanje ili ostavljanje putnika. U tablicu se spremaju zemljopisna širina i dužina lokacije u stupce `latitude` i `longitude`, te ime postaje u stupcu `name`.

Primarni ključ je stupac `stop_id`, koji pohranjuje tekstualnu reprezentaciju UUID-a.

Tablica `Stop` povezana je s tablicom `Ride` preko veze više -prema-jedan (M:1) stupcem `ride_id` ako postaja pripada određenoj vožnji. Ako postaja pripada zahtjevu za rezervaciju sjedala, tada je to naznačeno više-prema-jedan (M:1) vezom s tablicom `RideRequest` preko stupca `request_id`.

Stupac `location` sprema točnu geografsku lokaciju postaje koristeći prostorni tip podataka `Point` iz proširenja `PostGIS`. `Point` predstavlja jednu lokaciju na Zemlji, prikazanu s pomoću jedne koordinate (uključujući 2-, 3- ili 4-dimenzije). Stupac `location` također sadrži i prostorni indeks za ubrzavanje upita s prostornim funkcijama.

NotificationRequest

Tablica `NotificationRequest` služi za prikupljanje zahtjeva putnika za obavještanjem o novim dostupnim prijevoznim opcijama. Ova tablica sadrži informacije o datumu i vremenu željenog prijevoza, broju putnika koji traže prijevoz te početnoj i završnoj lokaciji putovanja.

Primarni ključ tablice je stupac `notification_request_id`, koji se automatski generira koristeći strategiju inkrementalne identifikacije.

Povezanost između zahtjeva i korisnika ostvaruje se putem više-prema-jedan (M:1) veze, gdje je stupac `account_id` strani ključ koji ukazuje na tablicu `Account`.

Početne i završne lokacije bilježe se kao točne geografske koordinate koristeći prostorni tip podataka `Point` iz proširenja `PostGIS`, a pohranjuju se u odgovarajuće stupce `start_location` i `end_location`. Osim toga `start_location` i `end_location` sadrže odgovarajuće prostorne indekse.

Notification

Tablica `Notification` služi za pohranu obavijesti koje obavještavaju korisnike o novim događajima. Svaka obavijest sadrži naslov, tijelo, datum i tip obavijesti, koji su spremljeni u stupcima `title`, `body`, `date` i `type`. Tipovi obavijesti obuhvaćaju:

- "SEATREQUEST": Vozač dobiva obavijest o novom zahtjevu za rezervaciju sjedala u prijevozu.
- "SEATREQUESTSENT": Putnik dobiva potvrdu o poslanom zahtjevu za rezervaciju sjedala.
- "SEATREQUESTDENIED": Putnik dobiva obavijest o odbijenom zahtjevu za rezervaciju sjedala.
- "SEATREQUESTACCEPTED": Putnik dobiva obavijest o prihvaćenom zahtjevu za rezervaciju sjedala.

- "NEWRIDE": Putnik dobiva obavijest o novoj vožnji koja zadovoljava njegov zahtjev za obavještavanjem.
- "DRIVERCANCEL": Putnik dobiva obavijest da je vozač otkazao vožnju.
- "PASSENGERCANCEL": Vozač dobiva obavijest da je putnik otkazao rezervirano mjesto.

Primarni ključ tablice je stupac `notification_id`, koji se automatski generira koristeći strategiju inkrementalne identifikacije.

Povezanost između obavijesti i korisnika ostvaruje se putem više-prema-jedan (M:1) veze, gdje je stupac `account_id` strani ključ koji referencira tablicu `Account`. Također, radi prikaza detalja obavijesti, postoje i veze više-prema-jedan (M:1) s tablicama `RideRequest` i `NotificationRequest`, pri čemu su stupci `ride_request_id` i `notification_id` označeni kao strani ključevi.

4.3. Stvaranje novog prijevoza

Vozači prilikom kreiranja novog prijevoza odabiru polazišnu i odredišnu lokaciju. Unoseći nazive gradova ili adrese ulica kao lokacijske točke, koordinate tih lokacija dohvaćaju se korištenjem alata za geokodiranje. Platforma Mapbox pruža jednostavan alat za geokodiranje unutar iOS operacijskog sustava, što smanjuje opterećenje na poslužitelj. Kada se utvrde točne koordinate polazišta i odredišta, Mapbox generira preglednu rutu vozila za prijevoz. Pored same rute, sustav također pruža informacije o vremenu trajanja prijevoza u sekundama i udaljenosti u metrima koju će vozilo prijeći. U slučaju da korisnik nije zadovoljan predloženom rutom, moguće je kreirati novu rutu s do pet dodatnih postaja između polazišta i odredišta. Na primjer, ako vozač putuje iz Zagreba u Split i želi obaviti postaju u Zadru, to može učiniti jednostavnim dodavanjem te postaje.

Osim podataka o ruti, potrebno je odrediti i datum i vrijeme polaska, broj slobodnih mjesta u automobilu, cijenu prijevoza te toleranciju za prikupljanje putnika. Tolerancija za prikupljanje putnika označava udaljenost koju je vozač voljan proći izvan svoje osnovne rute kako bi ukrao ili iskrcao putnike. Ova tolerancija može varirati između 1 i 50 kilometara, a bira ju sam korisnik. Toleranciju se nikada ne postavlja na manje od jednog kilometra, kako bi se prijevoz pokazao putnicima koji se nalaze blizu rute putovanja i mogu doći do lokacije kroz koju prolazi vožnja, bez potrebe za promjenom rute vozača. To znači da se putnicima

prikazuju i vožnje u kojima ih vozač neće pokupiti na željenom odredištu već će oni morati doći do mjesta kojim prolazi vožnja.

Stvaranje novog prijevoza završava se slanjem podataka na poslužitelj i zapisivanjem novog retka u tablicu `Ride`, s informacijama o ruti, datumu i vremenu putovanja, cijeni, broju slobodnih mjesta u automobilu i tolerancijom. Poslužitelj tada izračuna točnu rutu kojom će prolaziti vožnja i sprema podatke vožnje u bazu podataka.

4.4. Algoritam za prepoznavanje kompatibilnih prijevoza

Najvažnija funkcionalnost sustava je povezivanje korisnika koji nude prijevoz i korisnika koji žele rezervirati mjesto u vozilu.

Putnici biraju svoje polazišne i odredišne lokacije na isti način kao i vozači. Ostali obvezni filteri za putnike uključuju željeni datum i vrijeme putovanja te broj putnika za koje žele rezervirati mjesto. Nije nužno da svaki putnik za kojeg se rezervira vožnja ima kreiran račun na platformi, već jedna osoba može izvršiti rezervaciju za više ljudi.

Podaci o željenoj vožnji šalju se putem HTTP zahtjeva na poslužitelj, čiji algoritam za prepoznavanje kompatibilnih prijevoza zatim filtrira odgovarajuće prijevoze u dva koraka: dohvaćanjem podataka iz baze i dohvaćanjem novih ruta koristeći `Openrouteservice`.

4.4.1. Dohvaćanje podataka iz baze

Prvi dio algoritma je čitanje vožnji iz baze podataka. Primjer SQL upita za dohvaćanje prijevoza iz baze podataka prikazan je u kodu Kod 4.1. Iz baze podataka potrebno je dohvatiti samo one prijevoze koji odgovaraju putnikovim zahtjevima, a provjera prijevoza odvija se u tri koraka, svedena u jedan SQL upit:

1. **Kompatibilnost prijevoza:** Kompatibilni prijevozi su oni koji odgovaraju jednostavnim putničkim zahtjevima. Ti zahtjevi su datum i vrijeme željenog putovanja te broj mjesta koja putnik želi rezervirati. Iz baze podataka dohvaćaju se samo one vožnje koje polaze na dani datum, nakon danog vremena i s dovoljnim brojem slobodnih sjedala.
2. **Filtriranje po ruti:** Koristeći funkcije iz `PostGIS`-a, filtriraju se vožnje koje prolaze u blizini putnikovog polazišta i odredišta, eliminirajući one rute koje očigledno ne prolaze u blizini tih lokacija. Usporedba udaljenosti između rute vožnje i putnikovih

lokacija vrši se korištenjem funkcije `ST_DWithin` iz PostGIS-a. Ova funkcija koristi se za usporedbu tolerancije vožnje s udaljenosti rute i polazišta putnika, kao i za usporedbu tolerancije vožnje s udaljenosti rute i odredišta putnika. U osnovi, funkcija uspoređuje zračnu udaljenost rute i lokacija putnika s predviđenom tolerancijom. Vožnje koje ne zadovoljavaju ovu provjeru (gdje udaljenost između lokacija i rute premašuje tolerancijski okvir) ne ulaze u popis kompatibilnih vožnji.

3. **Provjera smjera vožnje:** Posljednja provjera nad podacima iz baze odnosi se na smjer vožnje, kako bi se eliminirale vožnje koje su prošle filtriranje po ruti, ali putuju u suprotnom smjeru od željenog smjera putnika. Ova provjera se provodi korištenjem funkcije `ST_LineLocatePoint` iz PostGIS-a. Funkcija određuje relativnu poziciju zadane točke duž linijske geometrije rute, vraćajući vrijednost između 0 i 1, koja predstavlja udio ukupne duljine rute od početne točke do zadane točke. Ako je relativna pozicija polazišne točke manja od relativne pozicije odredišne točke, to znači da vožnja putuje u smjeru u kojem putnik želi putovati.

```
SELECT *
FROM Ride ride
WHERE ride.date = '2024-06-27'
AND ride.time >= '08:00:00'
AND ride.seats >= 2
AND ST_DWithin(
    ride.route,
    ST_GeomFromText('POINT(15.981945.8150)', 4326),
    ride.tolerance,
    TRUE
)
AND ST_DWithin(
    ride.route,
    ST_GeomFromText('POINT(16.4323 46.3844)', 4326),
    ride.tolerance,
    TRUE
)
AND ST_LineLocatePoint(
    ride.route,
    ST_GeomFromText('POINT(15.9819 45.8150)', 4326))
<
ST_LineLocatePoint(
    ride.route,
```



```

        ST_GeomFromText ('POINT(16.4323 46.3844)', 4326)
    )
ORDER BY ride.date ASC,
        ride.time ASC;

```

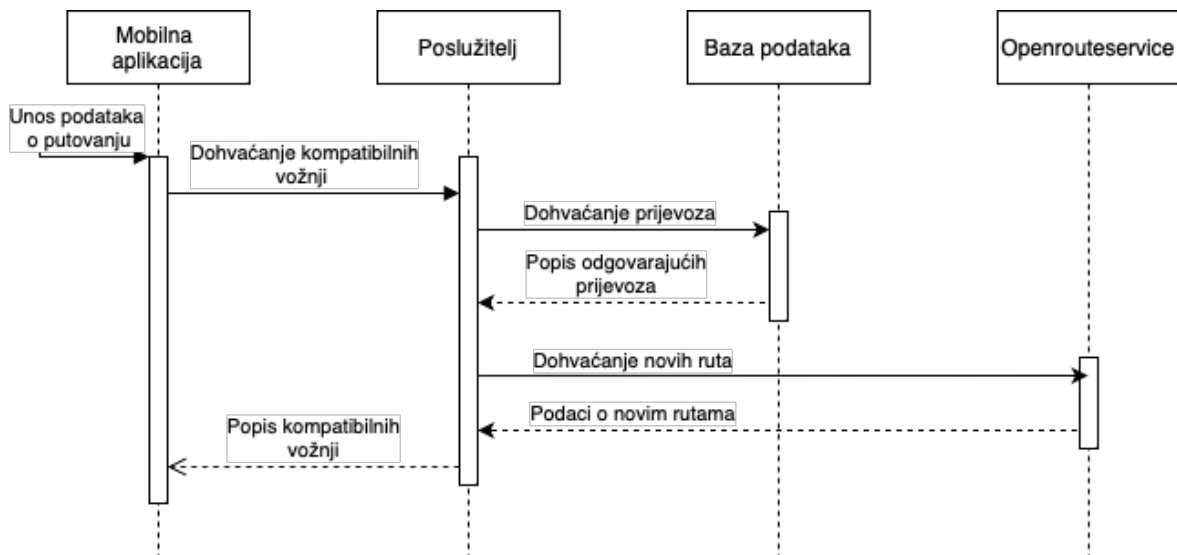
Kod 4.1 SQL upit za odgovarajuće prijevoze

4.4.2. Dohvaćanje novih ruta koristeći Openrouteservice

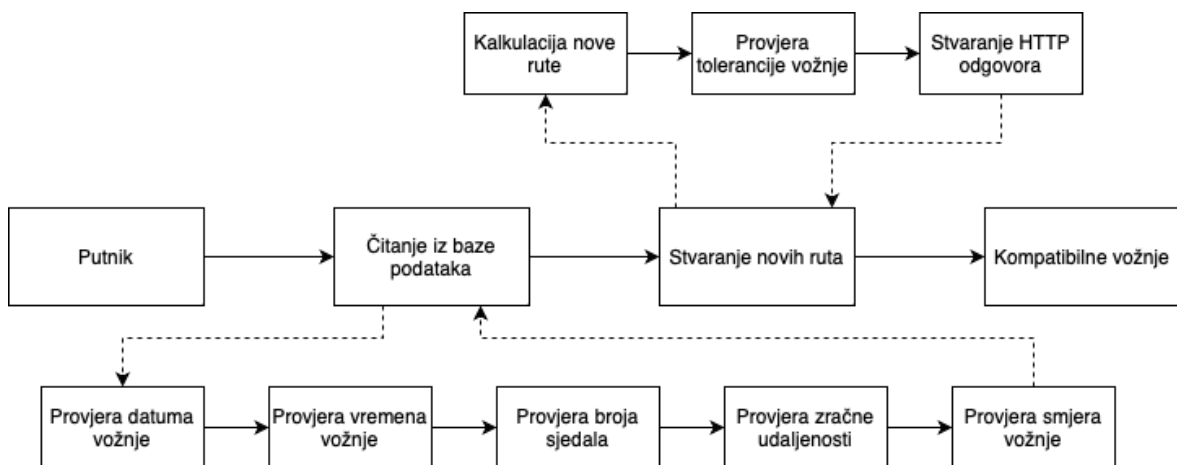
Drugi dio algoritma je dohvaćanje novih optimalnih ruta koristeći Openrouteservice API za optimizaciju. Nove rute sadržavat će i postaje putnika koji traže prijevoz, što znači da se ukupna udaljenost rute mijenja i treba provjeriti pripada li ruta vožnje još uvijek u vozačev tolerancijski okvir. Kalkulacija i provjera nove rute odvijaju se u sljedeća 3 koraka:

1. **Kalkulacija nove rute:** Za svaku vožnju dobivenu iz baze podataka potrebno je izračunati novu rutu kojom će prolaziti ako će ukrcavati i iskrcavati novog putnika. Za svaku vožnju šalje se HTTP zahtjev prema optimizacijskom API-ju Openrouteservice-a. Taj HTTP zahtjev sadrži polazišne i odredišne lokacije svih putnika, u ovom slučaju i novog putnika, na popisu mjesta (poslova) kroz koja ruta mora proći. Kao rezultat, u HTTP odgovoru dobivena je nova optimalna ruta koja uključuje i lokacije novog putnika.
2. **Provjera tolerancije:** Kompatibilnost vožnje određuje se provjerom ukupne udaljenosti nove rute. Vozač prilikom kreiranja nove vožnje odabire toleranciju, tj. udaljenost koju je voljan proći izvan svoje osnovne rute kako bi ukrao ili iskrao putnike. U ovom koraku uspoređuje se razlika između tolerancije i promjene u udaljenosti rute nakon dodavanja novih postaja. Ako se ruta promijenila za manji broj kilometara nego što je definirana tolerancija, smatra se da je vožnja kompatibilna. To znači da se putnicima prikazuju samo vožnje s rutama koje su unutar granica tolerancije vozača.
3. **HTTP odgovor:** Nakon dobivenih odgovora iz Openrouteservice-a i provedenih provjera tolerancije, kreira se HTTP odgovor koji sadrži popis kompatibilnih vožnji. Taj popis šalje se prema mobilnoj aplikaciji.

Slika 4.3 prikazuje sekvencijski dijagram koji ilustrira interakciju sustava prilikom prepoznavanja kompatibilnih prijevoza, a Slika 4.4 prikazuje pojednostavljen proces algoritma.



Slika 4.3 Sekvencijski dijagram prepoznavanja kompatibilnih prijevoza



Slika 4.4 Proces algoritma za prepoznavanje kompatibilnih prijevoza

4.5. Upravljanje rezervacijama

Nakon odabira željenog prijevoza, putnici nisu zajamčeni sjedalo u vozilu jer njihovu rezervaciju potvrđuje vozač. Postoje dva glavna razloga za izradu sustava upravljanja rezervacijama: sigurnost i sloboda vozača. Kada vozač primi novi zahtjev za rezervaciju, u obavijesti su mu prikazani izgled nove rute s lokacijama tog putnika i njegov korisnički profil.

Sigurnosni razlog obuhvaćen je prikazivanjem informacija o putniku. Na isti način na koji putnici mogu provjeriti korisnički profil vozača prije rezervacije, vozači mogu provjeriti profil putnika koji želi rezervirati mjesto u vozilu. To omogućava vozačima pregledavanje

osobnih informacija i recenzija putnika, te im olakšava donošenje odluke o prihvaćanju rezervacija tih putnika.

S druge strane, sloboda vozača podrazumijeva mogućnost odabira putnika na temelju njihovih mjesta ukrcaja i iskrcaja. Ako nova ruta ne odgovara vozaču ili prolazi kroz mjesta koja on želi izbjeći, on može odbiti zahtjev za rezervaciju. Broj zahtjeva za rezervaciju nije ograničen brojem sjedala u vozilu, što omogućava situaciju gdje postoji više putnika koji žele rezervirati isto mjesto u vozilu. Tako vozač može imati više opcija i odabrati one putnike koji će mu omogućiti najkraću ili najbržu rutu.

5. iOS Mobilna aplikacija

5.1. Razvoj aplikacije

Mobilna aplikacija razvijena je isključivo za iOS platformu. U ožujku 2024. godine, tržišni udio iOS operacijskog sustava iznosio je 28,58 %, što implicira da bi, u cilju postizanja veće popularnosti aplikacije u budućnosti, bilo nužno razviti i verziju za Android platformu[10].

iOS aplikacija je izrađena koristeći najnoviju verziju Appleovog razvojnog okruženja, XCode 15. Posljednja verzija iOS operacijskog sustava je 17.5, a s obzirom na to da korisnici Appleovih mobilnih uređaja redovito ažuriraju svoje sustave, aplikacija podržava najnoviju dostupnu verziju operacijskog sustava u vrijeme početka razvoja, a to je verzija 17.2.

Cilj ove aplikacije je popularizacija koncepta dijeljenja prijevoza, čime se nastoji korisnicima omogućiti lakše i brže pronalaženje odgovarajućih prijevoza i suputnika. Time se pokušava ne samo smanjiti troškove putovanja, već i povećati društvenu interakciju i učiniti putovanja ugodnijima i zabavnijima.

Ova aplikacija pruža platformu koja povezuje vozače i putnike s istim odredištem, olakšavajući koordinaciju zajedničkih putovanja. Tako korisnici mogu efikasnije upravljati sa svojim vremenom i resursima, smanjujući financijske izdatke za gorivo i održavanje vozila. Osim ekonomskih prednosti, aplikacija potiče ekološku osviještenost smanjenjem broja vozila na cestama, što doprinosi smanjenju emisije štetnih plinova i prometnih gužvi.

5.1.1. Radni okviri korisničkog sučelja

Razvoj korisničkih sučelja za iOS aplikacije trenutno nudi dvije opcije: UIKit i SwiftUI.

UIKit, pokrenut 2008. godine, pruža unaprijed definirane komponente za obradu korisničkih interakcija, poput unosa podataka i upravljanja gestama. Ključna komponenta UIKit-a je `ViewController` koja upravlja događajima kao što su dodiri, geste i prijenos podataka na korisničko sučelje. Također djeluje kao posrednik između modela podataka aplikacije i korisničkog sučelja te upravlja koordinacijom s drugim objektima unutar aplikacije[11].

S druge strane, SwiftUI, predstavljen 2019. godine, donosi modernije pristupe programiranju, poput deklarativne sintakse. Ovaj okvir omogućuje jednostavnu i intuitivnu

definiciju korisničkih sučelja, oslobađajući programere od brige o detaljima implementacije[11].

Unatoč relativnoj nezrelosti i potencijalnim izazovima SwiftUI-a, odlučeno je da se aplikacija razvije koristeći oba radna okvira. Ova odluka temelji se na želji za iskorištavanjem robusnosti i pouzdanosti UIKit-a, kao i brzine i jednostavnosti razvijanja koju pruža SwiftUI.

Odluka o kombiniranju radnih okvira olakšala je odabir arhitekture jer se MVC (engl. *Model, View, Controller, MVC*) arhitektura prirodno uklapa s radnim okvirom UIKit.

Uz MVC, korišten je i koordinacijski obrazac, koji omogućuje uklanjanje zadatka navigacije iz kontrolera, što pomaže učiniti ih upravljivijima i više ponovno upotrebljivima, dok istovremeno omogućuje prilagodbu toka aplikacije kada je potrebno[12].

5.1.2. Pohrana podataka

Pohrana podataka na uređaju odrađuje se koristeći radni okvir Realm zbog njegove jednostavnosti i performansi. Realm je napredno rješenje baze podataka koje teži brzini i performansama, pružajući sve potrebne pogodnosti za rad s podacima u iOS aplikacijama.

Realm nije relacijska baza podataka, već se temelji na upotrebi klasa za definiranje sheme podataka i rad s objektima, što ga čini alternativom klasičnim relacijskim bazama podataka. Ovaj pristup omogućuje jednostavnu manipulaciju podacima, bez potrebe za složenim SQL upitima, što rezultira učinkovitijim i fleksibilnijim razvojem aplikacija za iOS platformu[13].

Podaci prikazani korisniku na ekranu uglavnom se dohvaćaju iz Realm baze podataka, osim u slučaju kada ti podaci ubrzo postanu irelevantni, poput dohvaćanja kompatibilnih prijevoza za putnika. Ekрани imaju promatrače na Realm koji ih obavještavaju o svakoj izmjeni podataka. To znači da se podaci dobiveni putem HTTP zahtjeva ne prikazuju izravno na ekranu, već se prvo spremaju u Realm, koji potom obavještava ekrane o novim informacijama.

5.1.3. Prikazivanje karata

Za implementaciju prikaza karata u iOS aplikaciji koristi se Mapbox, neovisan pružatelj kartografskih podataka koji nudi razvojne alate za iOS, Android i web platforme. Mapbox omogućuje jednostavan prikaz karata na različitim uređajima i nudi širok spektar dodatnih

možnosti. Dostupne su različite cjenovne opcije za korištenje Mapbox usluga, uključujući besplatnu verziju s određenim ograničenjima. Uključen je API za usmjeravanje i geokodiranje koji podržava do 100,000 besplatnih upita mjesečno.

Mapbox omogućuje jednostavnu integraciju globalnog kartografskog sučelja u aplikaciju uz minimalan kod. Također, omogućuje dodavanje slojeva na prikazanu kartu, što uključuje iscrtavanje ruta putovanja i označavanje postaja.

Iscrtavanje ruta se ostvaruje korištenjem `PolylineAnnotation` strukture koja prihvaća podatke u obliku `LineString`. Geometrije ruta koje se razmjenjuju između mobilne aplikacije i poslužitelja uvijek su šifrirane kako bi se optimizirala veličina podataka. Potrebno je dešifrirati primljene rute iz tekstualnog formata u odgovarajuću geometriju, u ovom slučaju `LineString`, radi prikaza na karti. Ovaj proces šifriranja i dešifriranja provodi se koristeći Googleov algoritam koji šifrira binarne vrijednosti u base64 ASCII znakove.

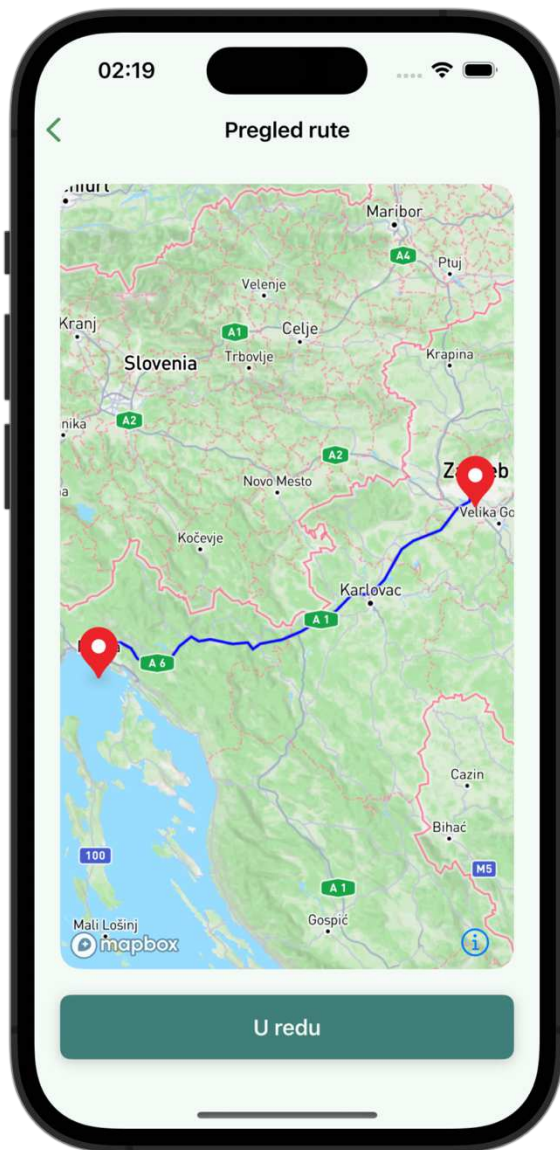
Iscrtavanje oznaka ostvaruje se korištenjem `PointAnnotation` strukture koja prima koordinate kao parametre. Ova struktura omogućuje postavljanje odabrane slike na specificirane koordinate te se koristi za prikaz svih važnih postaja. Važne postaje uključuju polazište i odredište vožnje, ali i sve ostale lokacije na kojima se putnici prikupljaju ili iskrcavaju.

Osim iscrtavanja karata, Mapbox sadrži i integrirane funkcionalnosti kao što su stvaranje ruta i geokodiranje.

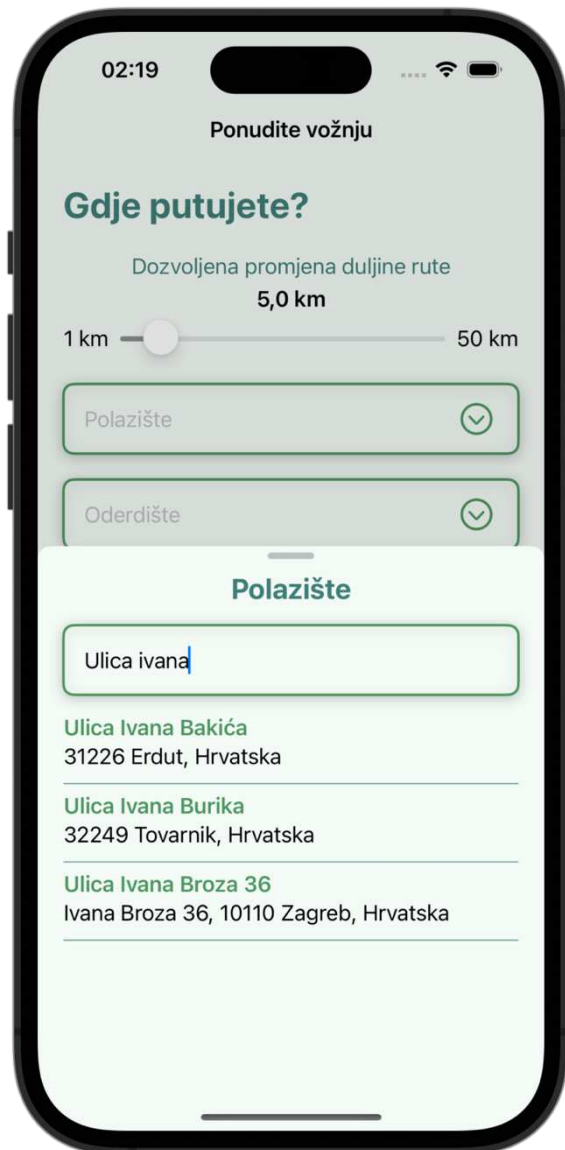
Prilikom stvaranja nove vožnje koristi se Mapbox API za usmjeravanje. Potrebno je konfigurirati vožnju postavljanjem koordinata kroz koje ruta prolazi, a zatim se dohvaća ruta s pomoću funkcije `Directions.shared.calculate(routeOptions)`. Ova funkcija šalje HTTP zahtjev na API za usmjeravanje kako bi se stvorila nova ruta za vožnju. Nakon dobivenog odgovora, ruta se prikazuje vozaču.

Geokodiranje je postupak mapiranja točnih koordinata na temelju naziva mjesta ili adrese ulice. Mapbox nudi mogućnost automatskog dovršavanja adresa i naziva gradova, što omogućuje korisnicima pristup postojećim adresama na temelju unesenog teksta, umjesto ručnog pretraživanja adrese. Kako bi se poboljšala brzina pretraživanja adresa, usluga geokodiranja ograničena je na mjesta, adrese i mjesta interesa (engl. *point of interest*) unutar Hrvatske. Geokodiranje pruža korisnicima mogućnost brzog pronalaska lokacija unutar Hrvatske, što znači da mogu precizno locirati adrese i važne točke interesa bez potrebe za detaljnim unosom podataka.

Slika 5.1 prikazuje korisničko sučelje s kartom na kojoj se nalazi ruta iz Zagreba u smjeru Rijeke, a Slika 5.2 korisničko sučelje za odabir početne lokacije vozača i automatskog dovršavanja adresa.



Slika 5.1 Ekran za prikaz rute



Slika 5.2 Ekran za odabir polazišta

5.2. Pregled glavnih funkcionalnosti

5.2.1. Korisnički račun i recenzije

Kako bi se postigla veća razina sigurnosti i omogućio bolji dojam o suputnicima, implementiran je sustav pregleda korisničkih računa i recenziranja.

Putnici, prilikom odabira prijevoza, mogu pregledati detalje o vozaču i već prihvaćenim sputnicima. Vozači, s druge strane, imaju mogućnost pregleda detalja o korisnicima koji žele rezervirati prijevoz nakon što prime obavijest o zahtjevu za rezervaciju. Ekran s detaljima o korisniku uključuje informacije kao što su ime, starost, kontakt podaci (e-mail i broj telefona), profilna slika, odabrano vozilo te posljednje dvije recenzije. Uz osnovne podatke, korisnici na ovom ekranu imaju pristup i gumbu za kontakt, koji omogućuje slanje SMS poruka ili uspostavljanje telefonskih poziva. Slika 5.3 prikazuje ekran s detaljima o korisniku.

Recenzije su ključne za izgradnju povjerenja unutar zajednice korisnika. One omogućuju transparentnost i olakšavaju donošenje informiranih odluka. Putnici i vozači mogu koristiti recenzije kako bi dobili uvid u prijašnja iskustva drugih korisnika, što pomaže u smanjenju rizika od neugodnih situacija ili sukoba. Osim toga, sustav recenziranja potiče korisnike na odgovorno i pristojno ponašanje, jer su svjesni da će njihovo ponašanje biti ocijenjeno i javno dostupno drugim korisnicima. Recenzije također služe kao povratne informacije za platformu, omogućujući kontinuirano poboljšanje usluga i iskustava svih korisnika.

Nakon završene vožnje, korisnici koji su putovali u istom vozilu imaju mogućnost međusobnog ocjenjivanja. Svaka recenzija sastoji se od ocjene (od 1 do 5), naslova i tijela. Ekran s detaljima korisnika prikazuje posljednje dvije dobivene recenzije, ali sadrži gumb za otvaranje ekrana s popisom svih dobivenih recenzija. Slika 5.4 prikazuje ekran za stvaranje nove recenzije.



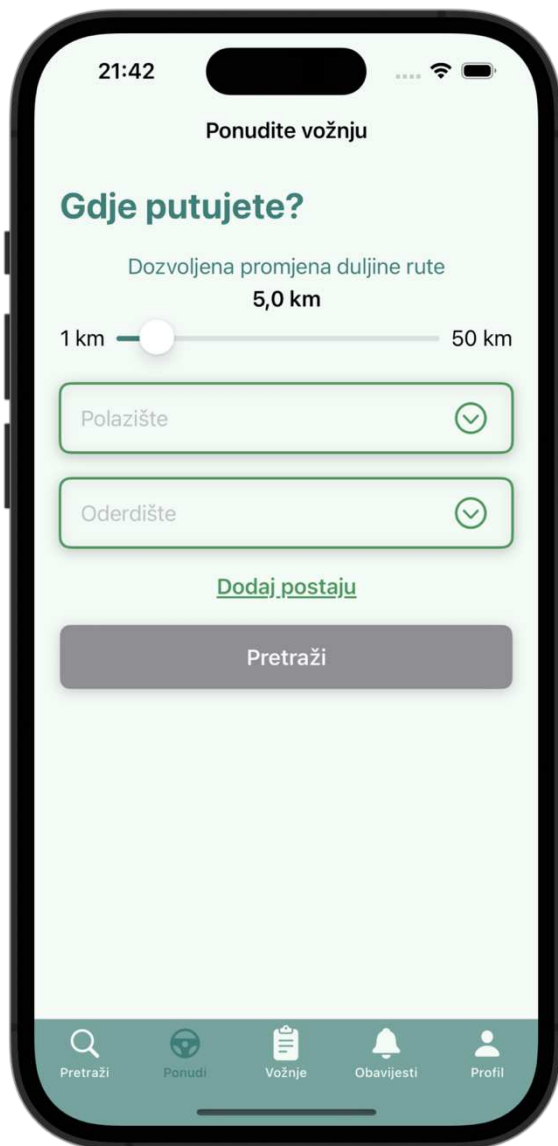
Slika 5.3 Ekran s detaljima korisnika



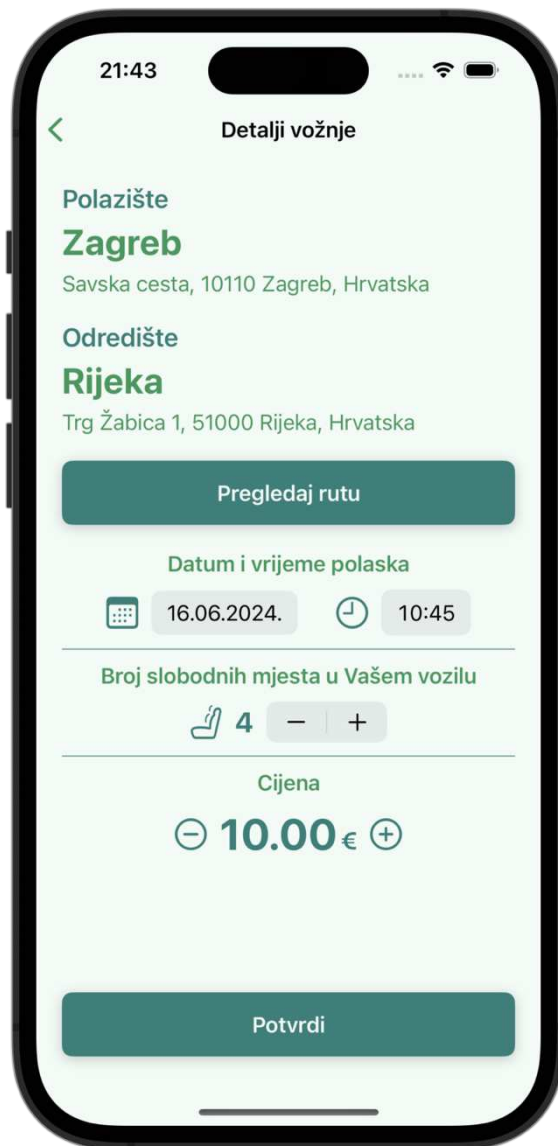
Slika 5.4 Ekran za stvaranje recenzije

5.2.2. Stvaranje prijevoza

Vozači mogu u bilo kojem trenutku odlučiti podijeliti svoju vožnju s drugim putnicima na jednostavan način. Na prvom ekranu prikazanom na slici Slika 5.5, vozač odabire polazište, odredište te, po potrebi, postaje između. Osim toga, određuje i željenu toleranciju promjene rute. Ekran prikazan na slici Slika 5.6 pokazuje detalje vožnje, gdje vozač može pregledati stvorenu rutu, odrediti datum i vrijeme polaska te broj mjesta u vozilu. Na kraju, vozač određuje cijenu putovanja i potvrđuje stvaranje novog prijevoza pritiskom na gumb. Proces završava prikazom potvrde ili ekrana s informacijom o grešci, ovisno o odgovoru poslužitelja.



Slika 5.5 Ekran za stvaranje novog prijevoza



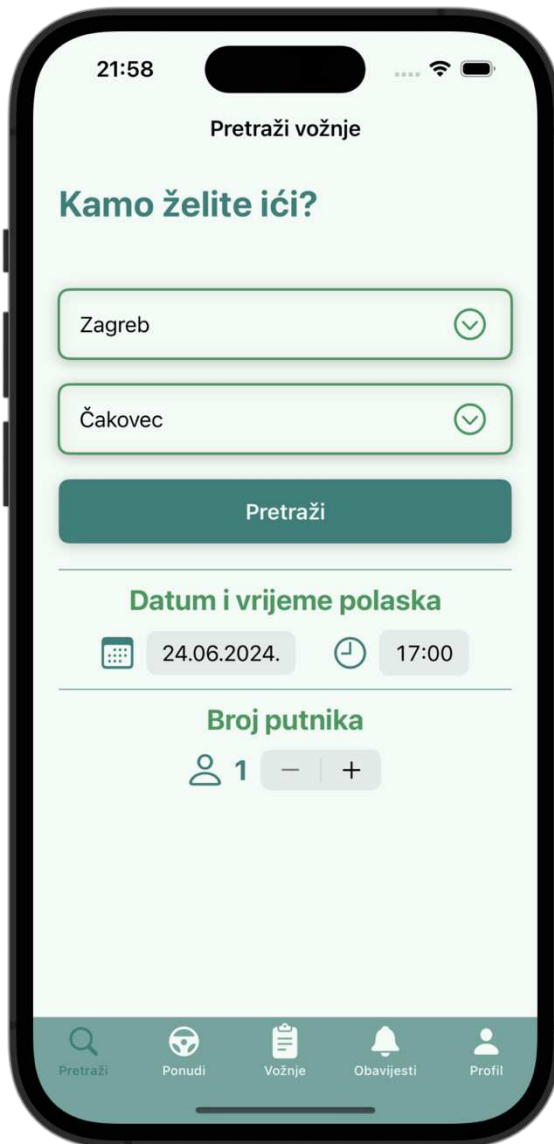
Slika 5.6 Ekran s detaljima vožnje

5.2.3. Rezervacija prijevoza

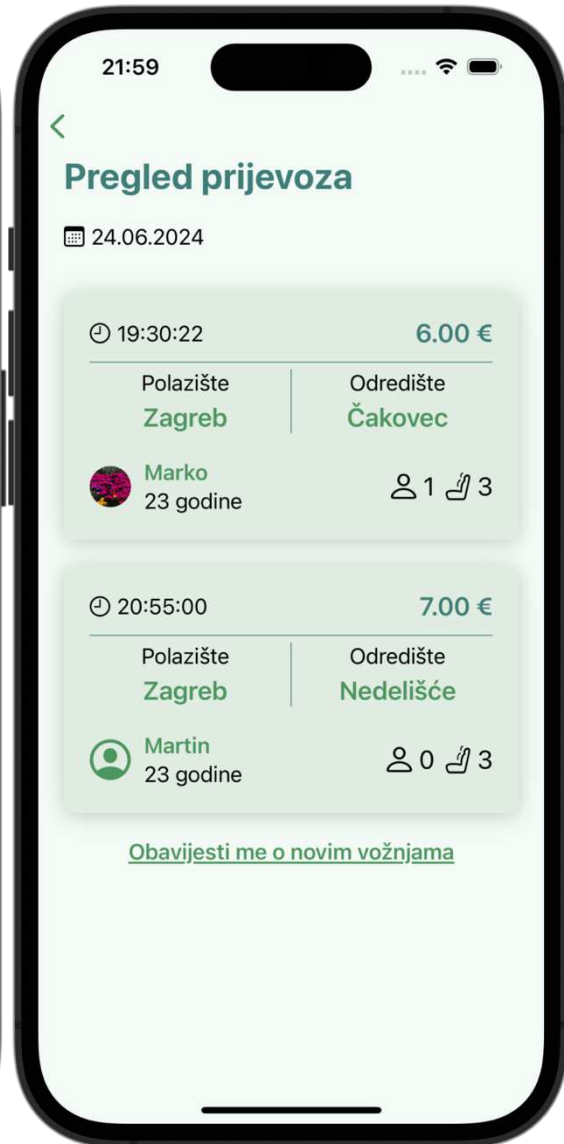
Putnici započinju pretragu prijevoza na isti način kao što vozači započinju stvaranje vožnje, odabirom polazišta i odredišta. Osim lokacija, putnik mora odabrati broj putnika za koje želi rezervirati mjesto u vozilu, kao i datum i vrijeme željenog prijevoza. Ekran za pretragu prijevoza prikazan je na slici Slika 5.7.

Ako sustav pronađe odgovarajuće prijevoze, oni će se korisniku prikazati u popisu koji prikazuje najvažnije detalje vožnje. Putnik može odabrati prijevoz kako bi vidio sve detalje, poput rute, nakon čega može poslati zahtjev za rezervaciju sjedala u odabranoj vožnji. Ako nema prijevoza koji mu odgovaraju, putnik ima opciju zahtjeva za obavještanje o stvaranju

novog prijevoza koji odgovara njegovim kriterijima. Klikom na gumb, njegov zahtjev za obavještanje šalje se na poslužitelj, koji će prilikom kreiranja novog prijevoza provjeravati zadovoljava li kriterije bilo kojeg od zahtjeva za obavještanjem i, ako je potrebno, stvoriti novu obavijest za odgovarajuće korisnike. Ekran s popisom kompatibilnih prijevoza prikazan je na Slika 5.8.



Slika 5.7 Ekran za pretragu prijevoza



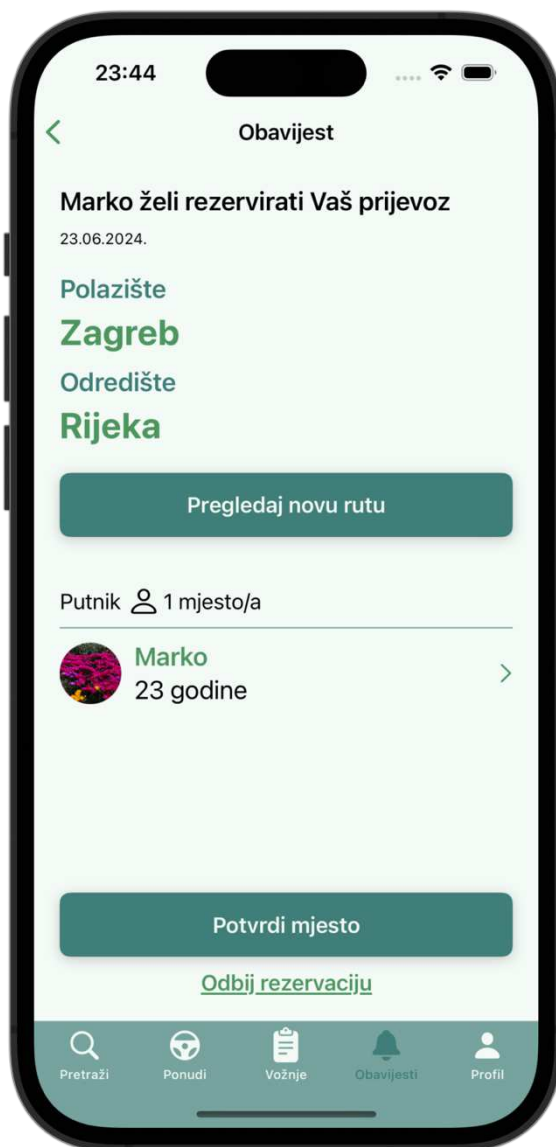
Slika 5.8 Ekran s popisom kompatibilnih prijevoza

5.2.4. Obavijesti

Obavijesti služe za komunikaciju između sustava i korisnika. Popis tipova obavijesti nalazi se u poglavlju 4.2.2. Za mobilnu aplikaciju to znači da je potrebno dizajnirati ekran i

omogućiti interakciju za svaki od tih tipova. Dva najvažnija tipa obavijesti su "SEATREQUEST" i "NEWRIDE", čiji su ekrani prikazani na slikama Slika 5.9 i Slika 5.10.

Važno je napomenuti da push obavijesti nisu implementirane u iOS aplikaciji, jer je za korištenje te usluge potrebno imati aktivan Apple Developer račun, čija cijena iznosi \$99 godišnje, a to je nepotreban trošak za stvaranje prve verzije aplikacije. Umjesto dohvaćanja popisa obavijesti s poslužitelja nakon dolaska nove push obavijesti, taj popis se dohvaća svaki put kada korisnik otvori ekran s popisom obavijesti.



Slika 5.9 Ekran s obavijesti tipa "SEATREQUEST"



Slika 5.10 Ekran s obavijesti tipa "NEWRIDE"

6. Testiranje rješenja

Testiranje sustava provedeno je na dva načina. Prvi pristup obuhvaća analizu točnosti prepoznavanja kompatibilnih vožnji, dok se drugi fokusira na evaluaciju performansi sustava, odnosno vrijeme potrebno za dohvaćanje kompatibilnih vožnji.

6.1. Točnost prepoznavanja kompatibilnih vožnji

Provjera točnosti prepoznavanja kompatibilnih vožnji izvršena je kroz stvaranje tri vožnje koje su bazirane na poznatim rutama. Točnost se ocjenjuje na temelju ponude vožnje putniku koji traži prijevoz za svoja specificirana polazišta i odredišta. Ovo se ostvaruje izračunavanjem novih ruta s novim postajama, te provjerom nalaze li se nove rute unutar zadanog tolerancijskog okvira za te vožnje. Popis kreiranih vožnji prikazan je u tablici Tablica 6.1.

Tablica 6.1 Popis kreiranih vožnji za testiranje

| Polazište vozača (točna adresa) | Odredište vozača (točna adresa) | Tolerancija (u metrima) | Duljina (u metrima) rute |
|------------------------------------|---|----------------------------|-----------------------------|
| Zagreb (Savska cesta 41) | Čakovec (Ulica Zrinsko - Frankopanska 14) | 10000 | 104858 |
| Zagreb (Savska cesta 41) | Čakovec (Ulica Zrinsko - Frankopanska 14) | 30000 | 104858 |
| Zagreb (Savska cesta 41) | Čakovec (Ulica Zrinsko - Frankopanska 14) | 5000 | 104858 |

U sljedećim tablicama prikazani su rezultati za pretraživanja vožnji s različitim polazištima i odredištima, a posljednji stupac označuje rezultat prikazivanja te vožnje korisniku. Tablica 6.2 prikazuje rezultate za prijevoz Zagreb-Čakovec (tolerancija 10000 metara) iz tablice Tablica 6.1, Tablica 6.3 za prijevoz Zagreb-Čakovec (tolerancija 30000 metara), a Tablica 6.4 za prijevoz Zagreb-Čakovec (tolerancija 5000 metara).

Tablica 6.2 Rezultati pretraživanja za vožnju Zagreb- Čakovec s tolerancijom 10000m

| Polazište putnika (točna adresa) | Odredište putnika (točna adresa) | Duljina (u metrima) nove rute | Rezultat |
|---|---|---|-------------------------|
| Zagreb (Savska cesta 41) | Čakovec (Ulica Zrinsko - Frankopanska 14) | 104858 | Vožnja je ponuđena |
| Zagreb (Slavonska avenija 11d) | Čakovec (Ulica Zrinsko - Frankopanska 14) | 105173 | Vožnja je ponuđena |
| Zagreb (Trg Kralja Tomislava 12) | Rijeka (Trg Žabica 2) | Nije izračunata (Ne zadovoljava provjeru zračne udaljenosti) | Vožnja nije ponuđena |
| Zagreb (Trg Kralja Tomislava 12) | Varaždin (Ulica Grada Lipika 15) | 101334 | Vožnja je ponuđena |
| Zagreb (Slavonska avenija 11d) | Novi Marof (Varaždinska ulica 7) | 113776 | Vožnja je ponuđena |
| Čakovec (Ulica Zrinsko - Frankopanska 14) | Zagreb (Savska cesta 41) | Nije izračunata (Ne zadovoljava provjeru smjera) | Vožnja nije ponuđena |

Tablica 6.3 Rezultati pretraživanja za vožnju Zagreb- Čakovec s tolerancijom 30000m

| Polazište putnika (točna adresa) | Odredište putnika (točna adresa) | Duljina (u metrima) nove rute | Rezultat |
|---|---|---|-------------------------|
| Zagreb (Savska cesta 41) | Čakovec (Ulica Zrinsko - Frankopanska 14) | 104858 | Vožnja je ponuđena |
| Zagreb (Slavonska avenija 11d) | Čakovec (Ulica Zrinsko - Frankopanska 14) | 105173 | Vožnja je ponuđena |
| Zagreb (Trg Kralja Tomislava 12) | Rijeka (Trg Žabica 2) | Nije izračunata (Ne zadovoljava provjeru zračne udaljenosti) | Vožnja nije ponuđena |
| Zagreb (Trg Kralja Tomislava 12) | Varaždin (Ulica Grada Lipika 15) | 101334 | Vožnja je ponuđena |
| Zagreb (Slavonska avenija 11d) | Novi Marof (Varaždinska ulica 7) | 113776 | Vožnja je ponuđena |
| Čakovec (Ulica Zrinsko - Frankopanska 14) | Zagreb (Savska cesta 41) | Nije izračunata (Ne zadovoljava provjeru smjera) | Vožnja nije ponuđena |

Tablica 6.4 Rezultati pretraživanja za vožnju Zagreb- Čakovec s tolerancijom 5000m

| Polazište putnika (točna adresa) | Odredište putnika (točna adresa) | Duljina (u metrima) nove rute | Rezultat |
|---|---|---|-------------------------|
| Zagreb (Savska cesta 41) | Čakovec (Ulica Zrinsko - Frankopanska 14) | 104858 | Vožnja je ponuđena |
| Zagreb (Slavonska avenija 11d) | Čakovec (Ulica Zrinsko - Frankopanska 14) | 105173 | Vožnja je ponuđena |
| Zagreb (Trg Kralja Tomislava 12) | Rijeka (Trg Žabica 2) | Nije izračunata (Ne zadovoljava provjeru zračne udaljenosti) | Vožnja nije ponuđena |
| Zagreb (Trg Kralja Tomislava 12) | Varaždin (Ulica Grada Lipika 15) | 101334 | Vožnja je ponuđena |
| Zagreb (Slavonska avenija 11d) | Novi Marof (Varaždinska ulica 7) | Nije izračunata (Ne zadovoljava provjeru zračne udaljenosti) | Vožnja nije ponuđena |
| Čakovec (Ulica Zrinsko - Frankopanska 14) | Zagreb (Savska cesta 41) | Nije izračunata (Ne zadovoljava provjeru smjera) | Vožnja nije ponuđena |

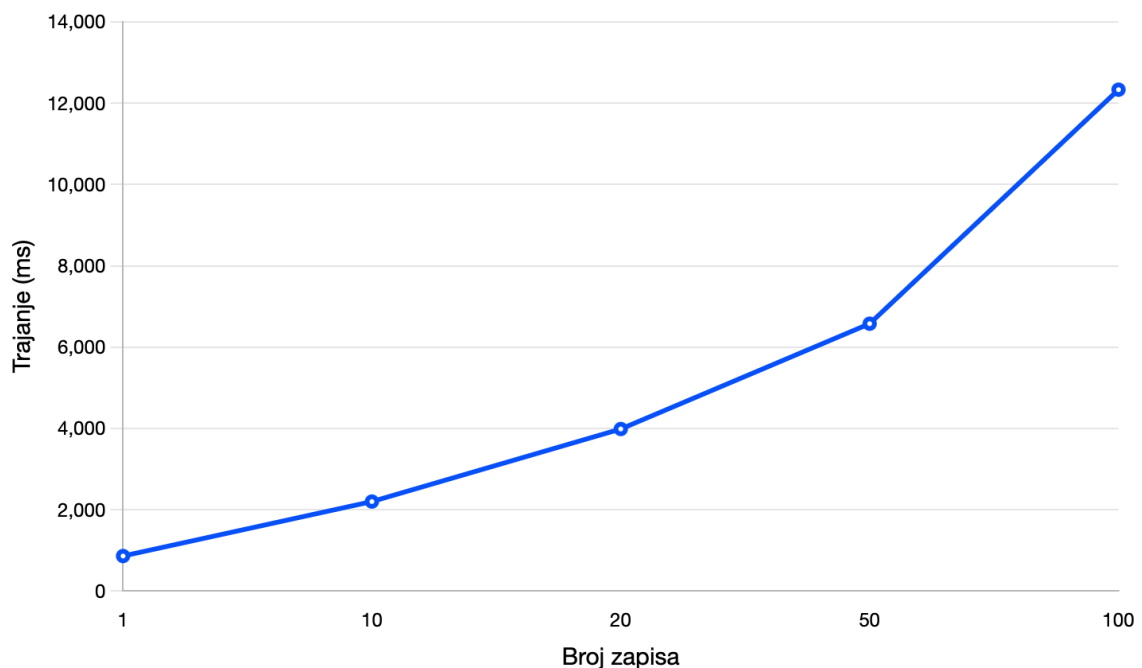
Rezultati testiranja prikazuju da algoritam za prepoznavanje kompatibilnih vožnji demonstrira zadovoljavajuću preciznost i funkcionalnost. Stvaranje tri vožnje temeljene na poznatim rutama omogućilo je evaluaciju sustava. Utvrđeno je da algoritam uspješno nudi prijevoz putnicima koji traže vožnju između njihovih odabranih polazišta i odredišta, uz pravilno izračunate nove rute koje zadovoljavaju zadane tolerancije. Osim toga,

ustanovljeno je da PostGIS funkcije uspješno filtriraju vožnje koje ne odgovaraju putničkim zahtjevima te ih izbacuju iz završnog rezultata. To rezultira smanjenjem broja potrebnih poziva prema Openrouteservice optimizacijskom API-ju.

6.2. Trajanje dohvaćanja kompatibilnih vožnji

Za evaluaciju vremena trajanja HTTP zahtjeva za dohvaćanje kompatibilnih vožnji s poslužitelja, stvoren je novi prijevoz s rutom od Zagreba do Nedelišća. Testiranje je provedeno u pet slučajeva s različitim brojem zapisa u bazi podataka: 1, 10, 20, 50 i 100 zapisa ove vožnje. Posebna pažnja posvećena je najdužem trajanju algoritma, gdje je putnikova pretraga konstruirana tako da zadovoljava filtraciju u bazi podataka. Ovime se osigurava generiranje HTTP zahtjeva prema Openrouteservice-u radi izračuna nove rute. Konkretno, putnik traži prijevoz s polazištem u Zagrebu i odredištem u Nedelišću te zahtjeva jedno slobodno mjesto u vozilu.

Rezultati testiranja su prikazani prosječnim trajanjem od 3 HTTP zahtjeva upućena prema poslužitelju za svaku od testnih situacija, odnosno za različit broj zapisa prijevoza u bazi podataka. Rezultati testiranja prikazani su u grafu na slici Slika 6.1.



Slika 6.1 Graf rezultata trajanja dohvaćanja kompatibilnih vožnji

Rezultati pokazuju da vrijeme potrebno za dohvaćanje kompatibilnih vožnji raste s povećanjem broja zapisa u bazi podataka iz nekoliko ključnih razloga.

Prvo, s većim brojem zapisa potrebno je izvršiti više PostGIS provjera kako bi se filtrirali prijevozi koji zadovoljavaju putničke zahtjeve. Ove provjere uključuju filtriranje prema datumu i vremenu polaska, raspoloživim mjestima u vozilu te provjeru rute korištenjem funkcije `ST_DWithin` radi eliminacije vožnji koje ne prolaze u blizini putnikovih lokacija. Dodatno, svaki odgovarajući prijevoz zahtijeva transformaciju podataka u odgovarajući format za HTTP odgovor koji se šalje korisniku. Nadalje, svaka odgovarajuća vožnja zahtijeva HTTP zahtjev prema Openrouteservice-u radi izračuna nove optimalne rute koja uključuje dodatne postaje za putnika.

S obzirom na ove izazove, ideja optimizacije u budućnosti je implementacija strategija kao što je paginacija rezultata, gdje se vožnje dohvaćaju u manjim skupinama ili stranicama, umjesto u jednom velikom upitu. Ovo bi smanjilo opterećenje baze podataka i vanjskog API-ja, poboljšavajući tako performanse sustava i korisničko iskustvo.

Zaključak

U ovom diplomskom radu istražena je važnost i primjena zajedničke vožnje kroz razvoj mobilne aplikacije koja olakšava organizaciju i provedbu takvih putovanja. Zajednička vožnja predstavlja ekonomičan, ekološki prihvatljiv i praktičan način putovanja koji donosi koristi vozačima i putnicima. Glavne prednosti uključuju smanjenje troškova putovanja, smanjenje ekološkog utjecaja, smanjenje broja vozila na cesti te povećanje dostupnosti parkirnih mjesta, a glavni izazovi zajedničkih vožnji su fleksibilnost i sigurnost, koji su pokriveni u konačnom rješenju.

Rezultati razvijenog rješenja su mobilna aplikacija i sustav za jednostavnu organizaciju zajedničke vožnje. Ključni aspekti ovih rješenja su prepoznavanje potencijalnih sudionika u zajedničkim vožnjama, korištenje podataka iz otvorenih izvora kroz integraciju s Openrouteservice platformom te sustav recenziranja za povećanje sigurnosti i povjerenja među korisnicima.

Razvijen je algoritam za prepoznavanje kompatibilnih prijevoza koji omogućuje korisnicima da lako pronađu prikladne vožnje prema svojim preferencijama, uključujući datum i vrijeme putovanja, broj slobodnih mjesta u vozilu te toleranciju za prikupljanje putnika. Implementiranje Openrouteservice API-ja, koji omogućuje rad s otvorenim podacima, pruža funkcionalnost za stvaranje optimalnih ruta prema specifičnim zahtjevima korisnika.

Cilj povećanja sigurnosti postignut je upravljanjem rezervacijama i sustavom recenziranja. Upravljanje rezervacijama omogućava vozačima da odaberu putnike na temelju njihovih preferencija i sigurnosnih razmatranja, čime se osigurava kvalitetno iskustvo za sve sudionike zajedničke vožnje. Recenzije igraju ključnu ulogu u stvaranju povjerenja i transparentnosti, potičući odgovorno ponašanje korisnika i kontinuirano poboljšanje usluga.

U konačnici, razvijeni sustav predstavlja solidan prvi korak u izradi nove platforme za organizaciju zajedničke vožnje. Međutim, za postizanje optimalne funkcionalnosti sustava potrebno je provesti dodatnu optimizaciju algoritma za prepoznavanje prijevoza, uključujući mogućnosti paginacije i poboljšanja točnosti rezultata. Nadalje, s obzirom na ograničenu pokrivenost korisnika samo s aplikacijom za iOS operacijski sustav, neophodno je razviti Android i web verzije aplikacije kako bi se postigla šira dostupnost i prihvaćenost među korisnicima.

Literatura

- [1] Aguiléra A., Eleonore P. The Future and Sustainability of Carpooling Practices: An Identification of Research Challenges. *Sustainability*. 13, 2021, 11824.
- [2] Cozza J., The History of Carpooling, from Jitneys to Ridesharing, (2012). Poveznica: <https://www.shareable.net/the-history-of-carpooling-from-jitneys-to-ridesharing/>; pristupljeno 18. lipnja 2024.
- [3] Pinto dos Santos R. A. Wiride - An application for a carpooling system. Diplomski rad. Instituto Superior Técnico, University of Lisbon, 2014.
- [4] Terzić R., Majstorović M. N. Open data concept, its application and experiences. *Vojnotehnički glasnik*. 67, 2019, 347-364.
- [5] Ramsey P., Leslie M., Introduction to PostGIS, (2012-2023). Poveznica: <http://postgis.net/workshops/postgis-intro/>; pristupljeno 11. lipnja 2024.
- [6] Bennett J. OpenStreetMap: Be your own Cartographer. 1. izdanje. Birmingham: Packt Pub Ltd, 2010.
- [7] OpenStreetMap contributors, OpenRouteService, (2023). Poveznica: https://wiki.openstreetmap.org/wiki/Openrouteservice#Openrouteservice_API; pristupljeno 21. lipnja 2024.
- [8] PostgreSQL Global Development Group., Introduction to PostgreSQL, (2024). Poveznica: <https://www.postgresql.org/about/>; pristupljeno 11. lipnja 2024.
- [9] Bagui S., Earp R. Database Design Using Entity-Relationship Diagrams. 2. izdanje. Boca Raton: Auerbach Publications, 2011.
- [10] Backlinko Team, iPhone vs Android Statistics, (2024). Poveznica: <https://backlinko.com/iphone-vs-android-statistics>; pristupljeno 16. lipnja 2024.
- [11] Ndungu D., SwiftUI vs. UIKit: What is the best choice for building an iOS user interface in 2024?, (2024). Poveznica: <https://sendbird.com/developer/tutorials/swiftui-vs-uikit>; pristupljeno 16. lipnja 2024.
- [12] Hudson P., How to use the coordinator pattern in iOS apps, (2019). Poveznica: <https://www.hackingwithswift.com/articles/71/how-to-use-the-coordinator-pattern-in-ios-apps>; pristupljeno 16. lipnja 2024.
- [13] Todorov M. Realm: Building Modern Swift Apps with Realm Database. 2. izdanje. McGaheysville: Razeware LLC, 2019.

Sažetak

Mobilna aplikacija za organiziranje zajedničke vožnje temeljena na podacima iz otvorenih izvora.

Ovaj diplomski rad istražuje važnost zajedničke vožnje kroz razvoj mobilne aplikacije koja olakšava organizaciju takvih putovanja. Analizirane su prednosti i izazovi zajedničkih vožnji, potreba za dijeljenjem prijevoza te trenutno stanje u Hrvatskoj. Opisani su neki od alata i sustava otvorenog koda za usmjeravanje vozila i stvaranje ruta. Vozači kreiraju prijevoze koji se pohranjuju u bazu podataka i prikazuju putnicima čija polazišta i odredišta ne odstupaju značajno od tih ruta. Stvaranje ruta i implementirani algoritam za prepoznavanje kompatibilnih prijevoza koriste postojeća rješenja otvorenog koda: PostGIS i Openrouteservice. Implementiran je i sustav recenzija radi povećanja povjerenja među korisnicima. Opisane su glavne funkcionalnosti mobilne aplikacije i odgovarajuća korisnička sučelja. Prikazani su rezultati preciznosti i efikasnosti algoritma za prepoznavanje kompatibilnih prijevoza.

Ključne riječi: Zajednička vožnja, usmjeravanje vozila, optimizacija ruta, Openrouteservice, iOS

Summary

Mobile Application for Organizing Carpooling Based on Open-Source Data

This thesis explores the importance of carpooling through the development of a mobile application that facilitates the organization of such journeys. The advantages and challenges of carpooling, the need for carpooling, and the current situation in Croatia have been analysed. Several open-source tools and systems for vehicle routing and route creation are described. Drivers create rides that are stored in a database and displayed to passengers whose starting points and destinations do not deviate significantly from those routes. Route creation and the implemented algorithm for recognising compatible rides use existing open-source solutions: PostGIS and Openrouteservice. A review system has also been implemented to increase user trust. The main functionalities of the mobile application and the corresponding user interfaces are described. The results of the precision and efficiency of the algorithm for recognising compatible rides.

Keywords: carpooling, vehicle routing, route optimization, Openrouteservice, iOS