



Univerzitet u Nišu, Elektronski fakultet
Katedra za računarstvo



MOBILNA APLIKACIJA I SERVIS ZA DELJENJE PREVOZA

Završni rad

Zadatak:

Proučiti metode, tehnologije i mobilne aplikacije i servise za deljenje prevoza. Razviti mobilnu aplikaciju i odgovarajući servis za deljenje prevoza, omogućiti funkcionalnosti za putnika i vozača kroz postavljanje plana puta i vremena vožnje, pretraživanje prevoza, prijavu na prevoz i ocenu obavljenog prevoza.

Mentor: Prof. dr Dragan Stojanović

Kandidat: Marko Ugrenović 14976

Komisija:

1. _____

Datum prijave: _____

2. _____

Datum predaje: _____

3. _____

Datum odbrane: _____

CONTENTS

1. Uvod	3
2. Deljenje vozila i vožnji	4
2.1 Potreba za aplikacijama i servisima za deljenje vožnje	4
2.2 Principi, metode i tehnologije koje se koriste u ovim sistemima	4
2.3 Real-time aplikacije	5
3. Poznate aplikacije i servisi za ride-sharing	6
4. My Ride	9
4.1 Generalna specifikacija zahteva.....	9
4.1.1 Funkcionalni i nefunkcionalni zahtevi	9
4.1.2 Funkcionalni i nefunkcionalni zahtevi vozača	10
4.2 Arhitektura i dizajn	10
4.2.1 Pregled arhitekture – organizacija paketa i podsistema u slojeve.....	11
4.2.2 Pogled na raspoređivanje sistema.....	12
4.3 Implementacija	13
4.3.1 Baza podataka	13
4.3.2 Serverski deo aplikacije	16
4.3.3 Klijentski deo aplikacije	17
5. My Ride - Opis funkcionalnosti	21
5.1 Početni ekran.....	21
5.2 Ekran za registraciju	22
5.3 Glavni meni.....	23
5.4 Ekran za nalaženje vožnji.....	24
5.5 Ekran za prikaz pronađene vožnje	27
5.6 Ekran za kreiranje nove vožnje	29
5.7 Ekran moje vožnje	30
5.8 Ekran za prikaz ponuđene vožnje	31
5.9 Ekran za podešavanja	32
5.10 Notifikacije	33
6. Zaključak	35
7. Literatura	36

1. UVOD

Inovacije u oblasti razvoja mobilnih tehnologija i Interneta u prethodnih nekoliko godina znatno su povećale mogućnosti koje nam mobilni uređaji pružaju. Povećane performanse, brzina protoka, različiti načini lociranja korisnika i dodatni senzori učinili su ove uređaje sposobnima da pruže korisnicima potrebne servise u najrazličitijim situacijama.

Njihov potencijal prepoznat je u mnogim privrednim granama, pa i u saobraćaju. Da li to bilo za navođenje korisnika ili za nekom vrstom upravljanja vožnjama ili vozilima, razni načini lociranja korisnika (poput GPS-a), raznovrsnost mapa i bežična konekcija na Internet, samo su neke od karakteristika koje se pokazuju kao korisne u ovakvim tipovima aplikacije. Iako postoje mnoge aplikacije iz ove oblasti, razvoj novih tehnologija i javljanje novih potreba korisnika dovodi do toga da se aplikacije stalno unapređuju i poboljšavaju.

Cilj ovog završnog rada je kreiranje jednostavne ali društveno-korisne mobilne aplikacije. Njen zadatak je da poveže ljude koji žele da ponude prevoz jer imaju slobodnih mesta u vozilu sa ljudima kojima je taj prevoz neophodan jer možda nemaju drugih mogućnosti. Glavna ideja jeste da se smanji trošak putovanja po osobi, a sa druge strane da se zaštiti okolina od prevelike količine izduvnih gasova. Pored toga što korisnici mogu da nude prevoz ili pronađu prevoz, imaju mogućnost da nakon svake vožnje ocene ostale putnike kako bi ostali korisnici pored osnovnih informacija imali bolji uvid kakva je osoba sa kojom bi putovali zajedno.

U radu je najpre, u drugom poglavlju, data kratka istorija deljenja vožnje i vozila kao i potreba za aplikacijama koje to omogućavaju. Takođe je dat pregled principa i metoda koje se koriste u ovim sistemima.

U trećem poglavlju dat je pregled već postojećih aplikacija koje se bave carpooling-om a ujedno su i najkorišćenije i najrasprostranjenije širom sveta.

U četvrtom poglavlju predstavljena je aplikacija *My Ride*. Opisana je generalna specifikacija zahteva zajedno sa detaljima arhitekture i implementacije.

U petom poglavlju predstavljeni su rezultati ove implementacije. Detaljno su opisane funkcionalnosti aplikacije uz pojedini prikaz ekrana.

Poslednje, šesto poglavlje, daje nam rezime svega urađenog, kao i smernice za dalje unapređenje ovog projekta.

2. DELJENJE VOZILA I VOŽNJI

Deljenje vožnje (**carpooling, car-sharing, ride-sharing**) je deljenje putovanja automobilom kako bi više od jedne osobe putovalo u automobilu. Imajući u vidu da više ljudi koriste jedno vozilo, smanjuju se putni troškovi po čoveku, kao što su troškovi goriva, putarine, kao i stres same vožnje. Carpooling je takođe ekološki prihvatljiviji i održiviji način za putovanje, jer deljenje putovanja smanjuje zagađenje vazduha, emisije izduvnih gasova, zagušenje saobraćaja na putevima i potrebu za parking mestima. Vlasti često podstiču deljenje vožnji, naročito tokom turističkog perioda, perioda visokog zagađenja ili visokih cena goriva.

Carpooling je prvi put postao istaknut u Sjedinjenim Državama kao racionalna taktika tokom Drugog svetskog rata. Vratio se sredinom sedamdesetih godina, zbog naftne krize 1973. i energetske krize 1979. godine.

Carpooling je naglo opao između 1970-ih i 2000-ih godina, vrhunac u SAD 1970. godine sa učešćem od 20,4%. Do 2011. godine pao je na 9,7%. U velikom delu to se pripisuje dramatičnom padu cena nafte i gasa (45%) tokom osamdesetih godina.

Međutim, nedavno je Internet olakšao rast deljenje vožnje, pa se u 2005. godini povećao na 10,7%. Popularnost Interneta i mobilnih telefona značajno je pomogla carpooling-u da se širi i omogućila ljudima da ponude i pronađu vožnju zahvaljujući jednostavnim i pouzdanim internetskim transportnim tržištima. Ove Web stranice se obično koriste za jednokratne vožnje na daljinu sa visokim troškovima za gorivo.

2.1 POTREBA ZA APLIKACIJAMA I SERVISIMA ZA DELJENJE VOŽNJE

Već neko vreme postoje internet platforme koje omogućavaju upravljanje vožnjama i povezivanje ljudi koji traže ili vozače ili pak putnike. Obično postoji vožnja koju je vozač postavio i putnici koji se za nju prijavljuju. Uglavnom protekne neko vreme dok vozač prihvati te putnike i tek onda oni mogu da se detaljnije dogovaraju o detaljima same vožnje.

Razvojem interneta i mobilnih tehnologija omogućili su i razvoj naprednijih platformi za deljenje vožnje. Upravljanje vožnjama se vrši u realnom vremenu i to na mobilnim uređajima samih vozača ili putnika. Time se olakšava i ubrzava proces deljenja vožnji i popunjavanje slobodnih mesta u bilo kom trenutku i bilo gde. To omogućava organizovanje neplaniranog i iznenadnog putovanja veoma brzo i efikasno.

2.2 PRINCIPI, METODE I TEHNOLOGIJE KOJE SE KORISTE U OVIM SISTEMIMA

Ovakav način deljenja vožnji generalno koristi tri najnovija tehnološka dostignuća:

1. GPS navigacione uređaje za određivanje putanje vozača i organizovanje ponuđenih vožnji
2. Pametne telefone kako bi putnici mogli da zatraže vožnju gde god da se nalaze
3. Socijalne mreže za uspostavljanje poverenja i odgovornosti između vozača i putnika

Ovi elementi se koordiniraju pomoću mrežnog servisa koji može odmah da upravlja plaćanjima vozača i pronalaženju vožnji koristeći algoritam optimizacije.

2.3 REAL-TIME APLIKACIJE

Mobilne, web i desktop onlajn aplikacije mogu biti podeljene u dve kategorije, takozvane “statične” i “*real-time*” aplikacije.

Statične aplikacije su brojnije i obuhvataju one aplikacije čiji se sadržaj menja samo onda kada korisnik sam zatraži novi sadržaj ili stranicu. Dakle, nova informacija je prisutna samo onda kada korisnik to zahteva, i kada je aplikacija aktivna.

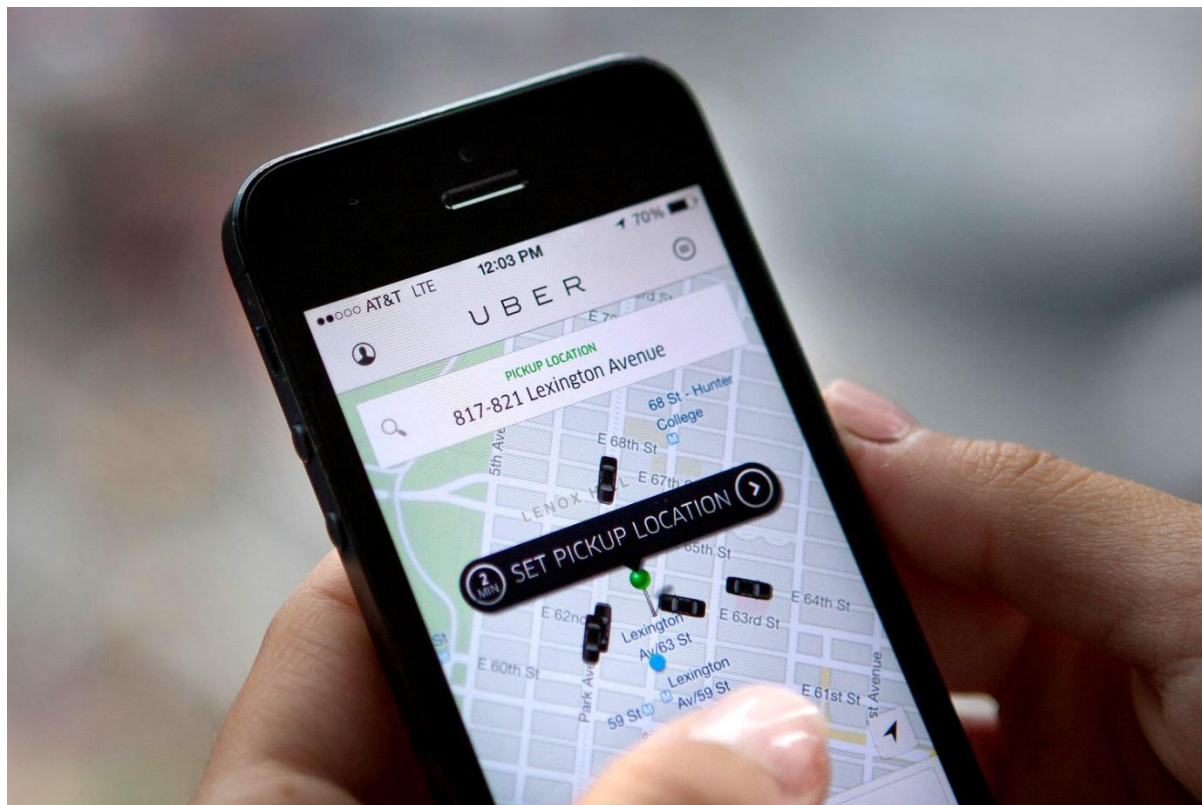
Real-time aplikacije su dosta popularnije danas. Ove aplikacije su uvek aktivne i obaveštavaju korisnika o promenama u trenutku njihovog nastanka. Omogućavaju, na primer, istovremeni rad sa prijateljem putem interneta, pronalaženje prijatelja na mapi, proveravanje lokacije autobusa itd. Omogućavaju da korisnik uvek bude u toku sa stvarima koje se oko njega događaju čak i kada eksplicitno ne pristupa aplikaciji. *Real-time* aplikacije zahtevaju da korisnik bude na mreži, i uglavnom su razvijene korišćenjem servisa koji su uvek aktivni u pozadini, na klijentskoj strani, a zahtevaju i postojanje servera koji će slati obaveštenja klijentu.

Relativno nov, ali veoma popularan, jeste i Firebase *framework* koji je zasnovan na asinhronom izvršavanju događaja i dizajniran je za kreiranje najrazličitijih internet aplikacija. Firebase se zasniva na *event-driven* programiranju. Ovakva vrsta programa koristi događaje (eng. *event*), odnosno promene stanja. Glavna zamisao je da postoji osluškivač koji osluškuje neki *event* i kada se on desi poziva se *callback* funkcija.

Callback je ekvivalent asinhronim funkcijama. *Callback* funkcija se poziva po završetku određenog zadatka. Na primer, funkcija čitanja fajla sa sistema može započeti svoje izvršenje i zatim vratiti kontrolu izvršnom sistemu tako da on može izvršavati ostale zadatke za vreme obavljanja čitanja fajla. Onda kada se U/I operacija čitanja fajla završi, poziva se *callback* funkcija kojoj se prosleđuje pročitani fajl kao parametar. Tako, nema blokiranja ili čekanja U/I operacija što čini Firebase izuzetno brzim i skalabilnim. Firebase može tako procesirati ogroman broj zahteva bez ikakvog čekanja da prethodno aktivirane funkcije vrate rezultate.

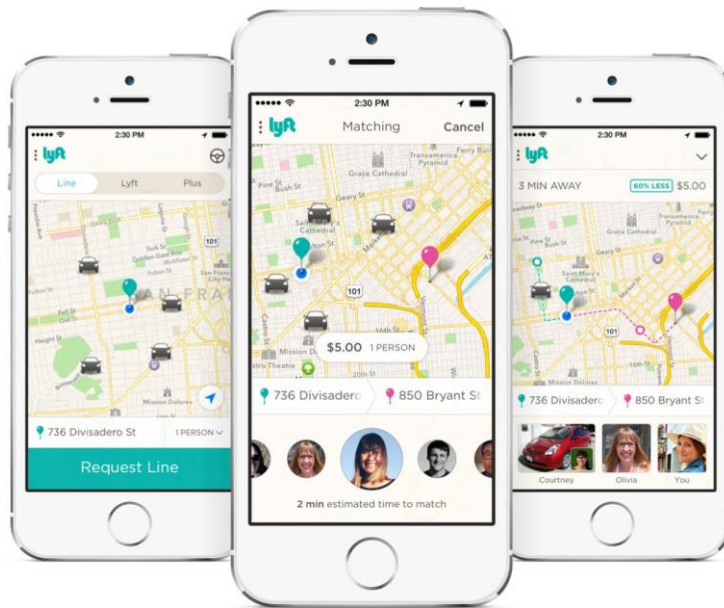
3. POZNATE APLIKACIJE I SERVISI ZA RIDE-SHARING

Ovo su neke od najpoznatijih aplikacija i servisa za deljenje vožnji:



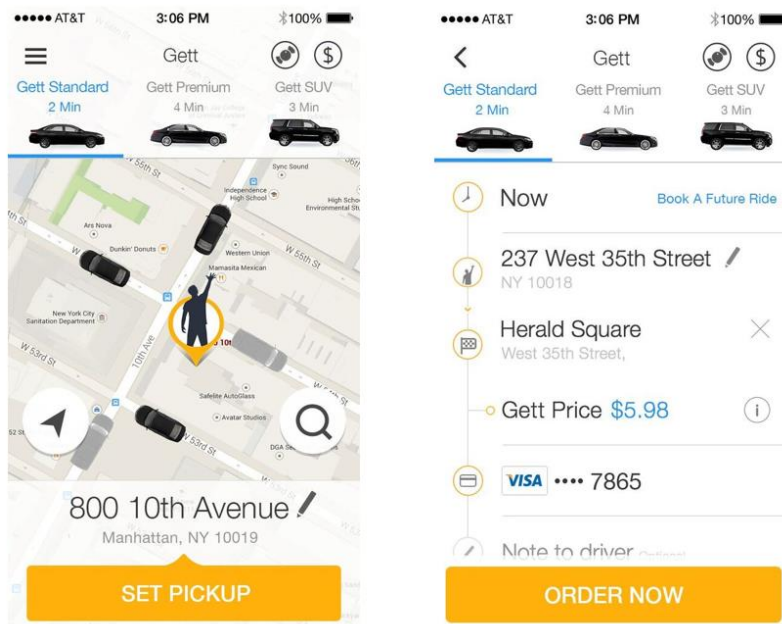
Slika 1. Mobilna Uber aplikacija

Uber Technologies Inc. je američka privatna kompanija za iznajmljivanje sa sedištem u San Francisku, Kalifornija, SAD, koja posluje u 633 grada širom sveta. Razvija, trži i upravlja Uberovim aplikacijama za deljenje vožnji i isporuku hrane. Uber vozači koriste svoja vozila, ili mogu iznajmiti vozilo kako bi vozili za Uber. Ime „Uber“ je referenca na uobičajenu reč „uber“, što znači „najviše“ ili „super“.



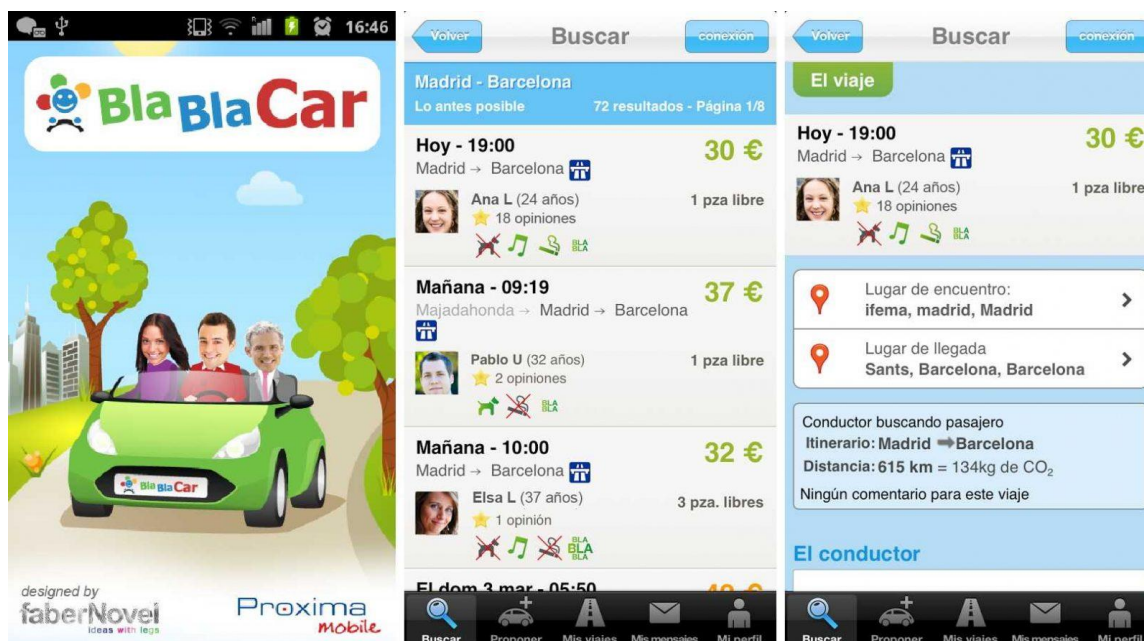
Slika 2. Mobilna Lyft aplikacija

Lift je prevozna mreža sa sedištem u San Francisku, Kalifornija. Razvija, trži i upravlja aplikacijom za prevoz automobila Lift. Pokrenut u junu 2012. godine, Lift posluje u oko 300 američkih gradova, uključujući New York, San Francisco i Los Angeles i pruža 18,7 miliona vozila mesečno. Korisnici moraju preuzeti aplikaciju Lyft na svoj iOS ili Android telefon, prijaviti se, uneti važeći broj telefona i način plaćanja. Putnici mogu zatražiti vožnju od obližnjeg vozača. Kada su potvrđeni, aplikacija prikazuje ime vozača, ocene prošlih putnika i fotografiju vozača i automobila.



Slika 3. Mobilna Gett aplikacija

Gett, ranije poznat kao GetTaxi, je globalna kompanija koja povezuje kupce sa transportom, robom i uslugama. Kupci mogu naručiti taksi ili kurira putem web stranice kompanije ili koristeći GPS baziranu mobilnu aplikaciju. Aplikacija je kompatibilna sa iPhone i Android platformama. Gett trenutno posluje u više od 100 gradova širom Sjedinjenih Država, Ujedinjenog Kraljevstva, Rusije i Izraela.



Slika 4. Mobilna BlaBlaCar aplikacija

BlaBlaCar je najveća svetska zajednica za deljenje vožnje na veće distance. Osmišljena u decembru 2003 od strane Fredrika Mazzella i osnovana 2006 godine, BlaBlaCar povezuje vozače i putnike koji su voljni da putuju zajedno između gradova i dele troškove putovanja. BlaBlaCar ima više od 600 zaposlenih i više od 35 miliona članova u 22 zemlje.

4. MY RIDE

MyRide je mobilna aplikacija i Web servis za prijavu i registrovanje prevoza, pretraživanje i obavljanje zajedničkog prevoza. Za sada je podržana samo na Android platformi.

4.1 GENERALNA SPECIFIKACIJA ZAHTEVA

Prilikom pokretanja aplikacije, od korisnika se traži da se prijavi na svoj nalog. Ako korisnik još uvek nema nalog, ima opciju da kreira nov. Od njega se traži da unese neke osnovne ali neophodne informacije (email, šifra, datum rođenja, ime). Nakon uspešnog registrovanja, korisnik se automatski prijavljuje na sistem. Svaki korisnik može da vidi svoje osnovne informacije i može u opcijama da promeni svoju profilnu sliku. Korisnik zatim može da koristi aplikaciju kako kao vozač, tako kao i putnik. Dovoljan je jedan nalog da istovremeno koristi obe usluge. Isto tako korisnik može da vidi sve vožnje koje je on ponudio i da upravlja s njima, kao i sve vožnje na koje se prijavio. Svaki korisnik ima svoju ocenu od 1 do 5, u zavisnosti od ocenjivanja prethodnih korisnika.

4.1.1 FUNKCIONALNI I NEFUNKCIONALNI ZAHTEVI

Putnik koji želi da nađe određenu vožnju ima mogućnost da pretražuje vožnje po različitim kriterijuma. To su početna i završna destinacija, koje mogu biti unutar grada, međugradska ali čak i u različitim zemljama. Sledeći kriterijum jeste datum i okvirno vreme početka vožnje. Tu se korisniku nudi mogućnost da izabere fleksibilnost od +/- 1,2 ili 3 sata u odnosu na specificirano vreme. Sledeće dve opcije nisu neophodne za pretragu ali mogu značajno da suze izbor vožnji ako se navedu, a to su maksimalni iznos troškova vožnje i minimalan broj praznih mesta. Ako se ova polja ostave prazna ona neće uticati na rezultate pretrage. Na kraju korisnik može da čekira da li želi da su dozvoljeni ljubimci i/ili pušenje u toku vožnje, kao dodatni kriterijum za pretragu vožnje.

Korisniku se, kada klikne na dugme za pretragu, prikazuje lista pronađenih vožnji po zadatim kriterijumima sa osnovnim informacijama. Klikom na neku od vožnji, korisniku se prikazuje detaljnije informacije o samoj vožnji kao i informacije vezano za vozača koji nudi tu vožnju. Ako je korisnik zainteresovan za tu vožnju može da pošalje zahtev vozaču. Kasnije se korisnik obaveštava ako je njegov zahtev prihvaćen ili pak odbijen. Korisnik nema pravo da naknadno poništi zahtev ili odustane od vožnje ako je prihvaćen, niti da ponovo pošalje zahtev ako je prvobitno odbijen.

Ako kojim slučajem ne postoji nijedna pronađena vožnja po zadatim kriterijumima, korisniku se nudi mogućnost da bude naknadno obavešten ako takva vožnja bude ponuđena.

Klikom na obaveštenja o prihvaćenom ili odbijenom zahtevu, ili naknadnoj ponuđenoj vožnji, korisniku se otvara aplikacija i prikazuje detalji te vožnje. Zajedno sa ostalim informacijama, postoji i informacija koja govori korisniku da li je poslao zahtev, ili je prihvaćen/odbijen zahtev za tu vožnju.

Ako je korisnik prihvaćen, nudi mu se opcija da oceni vozača/vožnju od 1 do 5 zvezdica. To može uraditi najkasnije do isteka 24h od završetka vožnje.

4.1.2 FUNKCIONALNI I NEFUNKCIONALNI ZAHTEVI VOZAČA

Vozač ima mogućnost da u bilo kom vremenu ponudi vožnju. Dovoljno je samo da unese informacije vezane za samu vožnju tipa odakle dokle je vožnja, kog datuma i u koje vreme se odvija, koliko je ukupan trošak putovanja kao i broj slobodnih mesta. Na kraju vozač može da dozvoli kućne ljubimce i/ili pušenje u automobilu. Klikom na dugme, vožnja je uspešno kreirana. Ako se desilo da neki putnik čeka na ovakvu vožnju, on će automatski biti obavešten da takva vožnja sada postoji.

Kada putnici koji su zainteresovani za njegovu vožnju pošalju zahtev, vozaču se prikazuje notifikacija da je neko poslao zahtev. Vozač u svakom trenutku može da prati trenutno stanje svih njegovih ponuđenih vožnji, gde može i da vidi ko je sve od putnika uputio zahtev za datu vožnju. On ima mogućnost da na osnovu osnovnih informacija o putnik prihvati ili odbije njihov zahtev. Vozač nema prava da menja već ponuđenu vožnju, kao ni da je otkaže.

Vozač može svakog prihvaćenog putnika ponaosob da oceni sa 1 do 5 zvezdica ali najkasnije po isteku 24h od završetka vožnje.

4.2 ARHITEKTURA I DIZAJN

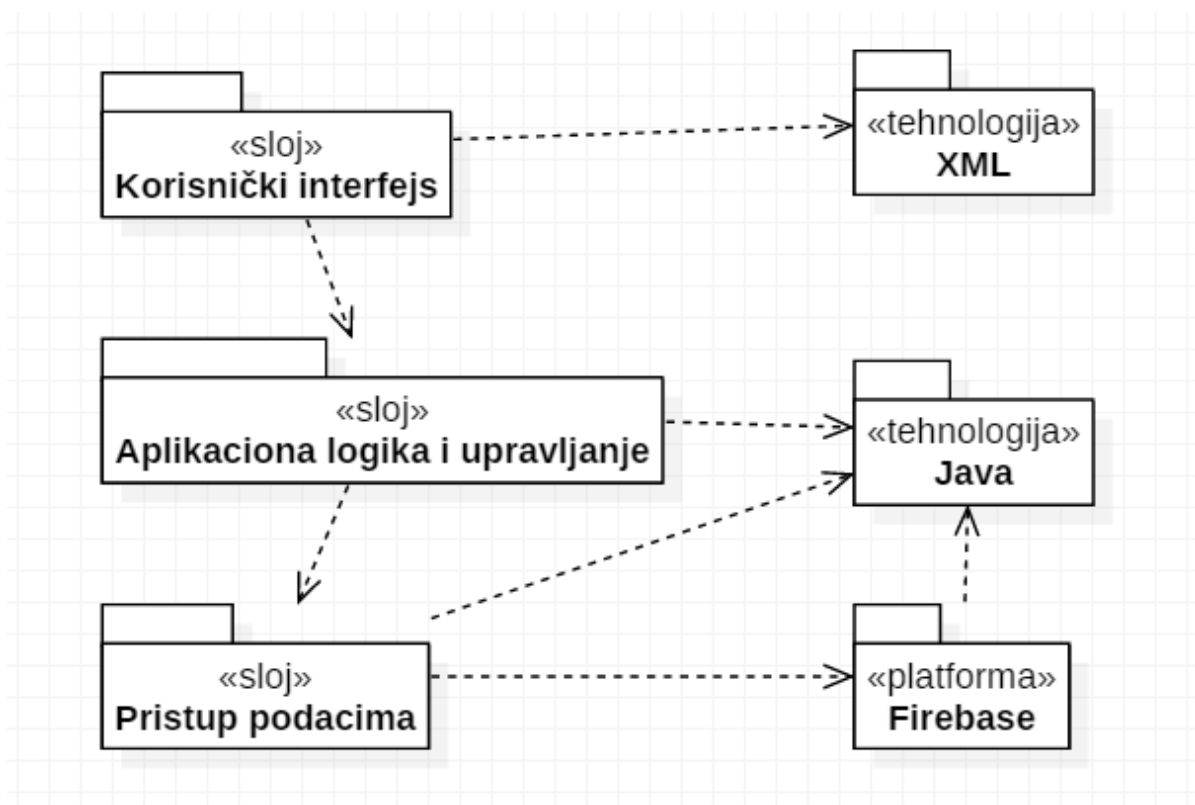
U ovom odeljku je dat pregled logičke arhitekture sistema. Ovaj pogled sadrži opis najznačajnijih klasa, njihove organizacije u pakete i podsisteme, i organizacija podsistema u slojeve. U cilju opisivanja dinamičkih aspekata arhitekture, ovaj odeljak može da uključi opise realizacije najznačajnijih slučajeva korišćenja. Da bi se ilustrovala veza između arhitekturno značajnih klasa, podsistema, paketa ili slojeva moguće je uključiti i odgovarajuće dijagrame klasa.

Logički pogled na aplikaciju My Ride obuhvata 3 glavna paketa: *Korisnički interfejs*, *Aplikaciona logika i upravljanje*, *Pristup podacima*.

Paket *Korisnički interfejs* sadrži ekrane, XML layout-e i multimedijalni sadržaj koji realizuje grafički dizajn i forme preko kojih korisnici sistema interaguju i komuniciraju sa sistemom.

Paket *Aplikaciona logika i upravljanje* predstavlja sloj sistema koji sadrži *Java* klase zadužene za realizaciju funkcionalnosti specifičnih za domen sistema koji se razvija.

Paket *Pristup podacima* sadrži *Java* klase koje predstavljaju interfejs za pristup, dodavanje i ažuriranje podataka koji se čuvaju u bazi podataka na serveru.



Slika 5. Organizacija paketa i podsistema u slojeve

KORISNIČKI INTERFEJS

Ovaj sloj realizuje korisnički interfejs aplikacije. U njemu su sadržani svi XML fajlovi, multimedijalni sadržaji i sve klase koje predstavljaju poglede na podatke.

Sloj korisničkog interfejsa direktno zavisi od sloja aplikacione logike i upravljanja, kao i paketa XML.

APLIKACIONA LOGIKA I UPRAVLJANJE

Sloj aplikacione logike i upravljanja je srednji sloj koji sadrži Java klase koje realizuju funkcionalnost karakterističnu za domen primene aplikacije. Takođe on uspostavlja vezu između korisničkog interfejsa i sloja za pristup podacima.

Ovaj sloj zavisi od sloja za pristup podacima i Java paketa.

PRISTUP PODACIMA

Sloj za pristup podacima se nalazi na dnu troslojne arhitekture i on je zadužen za pribavljanje, dodavanje i ažuriranje podataka koji se čuvaju bazi podataka.

Ovaj sloj ne zavisi od drugih slojeva, ali je zavisian od Firebase platforme i Java paketa.

XML

Tehnologija XML-a definiše gradivne elemente ekrana koji se prikazuju na mobilnom uređaju i koji omogućavaju prikaz formatiranih informacija i realizaciju formi za unos i ažuriranje podataka.

JAVA

Tehnologija Java obezbeđuje mehanizam za pisanje i izvršavanje programskog koda na mobilnom uređaju. Ovaj kod direktno upravlja realizacijom korisničkog interfejsa i pristupu bazi podataka u cilju pribavljanja, unosa i ažuriranja podataka.

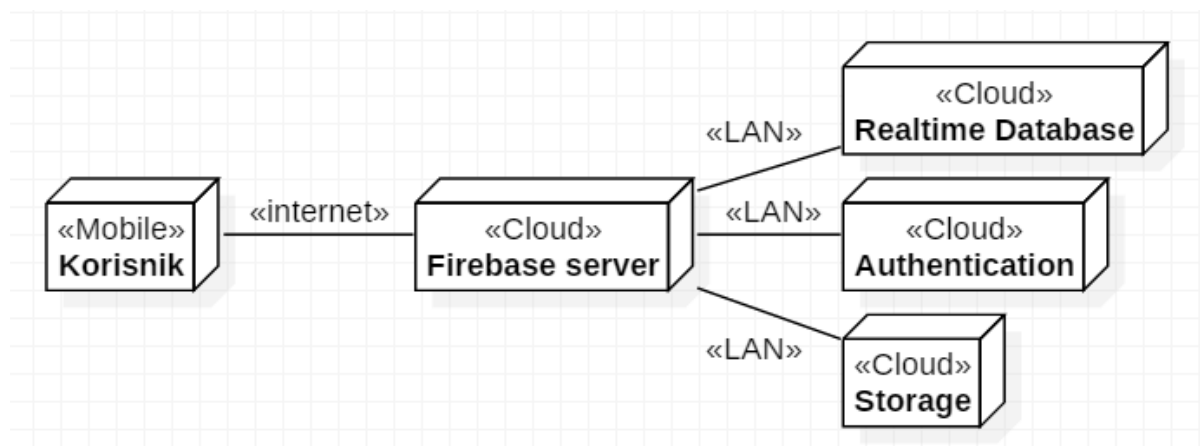
FIREBASE

Mobilna i Web platforma koja omogućava upravljanjem bazom podataka, skladištenjem multimedijalnog sadržaja, autentifikacije korisnika i notifikacijama koje se koriste za realizaciju My Ride mobilne aplikacije.

4.2.2 POGLED NA RASPOREĐIVANJE SISTEMA

Pogled na raspoređivanje sistema prikazuje različite fizičke čvorove za najopštiju konfiguraciju sistema. Fizičkim čvorovima koji predstavljaju procesore vrši se dodeljivanje identifikovanih komponenti.

Na sledećoj slici dat je UML dijagram raspoređivanja My Ride mobilne aplikacije.



Slika 6. UML dijagram rasporeda sistema

KORISNIK

Pristup My Ride mobilnoj aplikaciji se obavlja preko korisnikovih Android mobilnih uređaja. Za povezivanje između korisnika i Firebase servera koristi se Internet infrastruktura tako da nema ograničenja u pogledu lokacije korisnika.

FIREBASE SERVER

Računar na kome se izvršava Web server opslužuje više korisnika koji pristupaju preko Interneta. U opštoj konfiguraciji Realtime Database, Autentication i Storage servisi se izvršavaju na posebnim mašinama koji su zajedno sa Firebase serverom povezani u lokalnoj (LAN) mreži.

REALTIME DATABASE

Računar na kome se izvršava proces koji realizuje funkcionalnost sistema za upravljanje bazom podataka.

AUTENTICATION

Računar na kome se izvršava proces koji realizuje funkcionalnost sistema za upravljanje korisnicima sistema.

STORAGE

Računar na kome se izvršava proces koji realizuje funkcionalnost sistema za skladištenje i upravljanje multimedijalnim sadržajima.

4.3 IMPLEMENTACIJA

Aplikacija My Ride je implementirana kao klijent-server arhitektura, gde mobilni korisnici preko servera pristupaju podacima iz baze podataka i dobijaju asinhrono notifikacije o vožnjama.

Mobilna, klijentska aplikacija razvijena je za Android uređaje, u Android Studio razvojnom okruženju. Podržana je na uređajima koji koriste verzije operativnog sistema nakon Android 4.0 verzije.

Serverski deo aplikacije je razbijen korišćenjem Firebase mobilne platforme, koja među ostalog sadrži i Realtime Database, Autentication i Storage.

4.3.1 BAZA PODATAKA

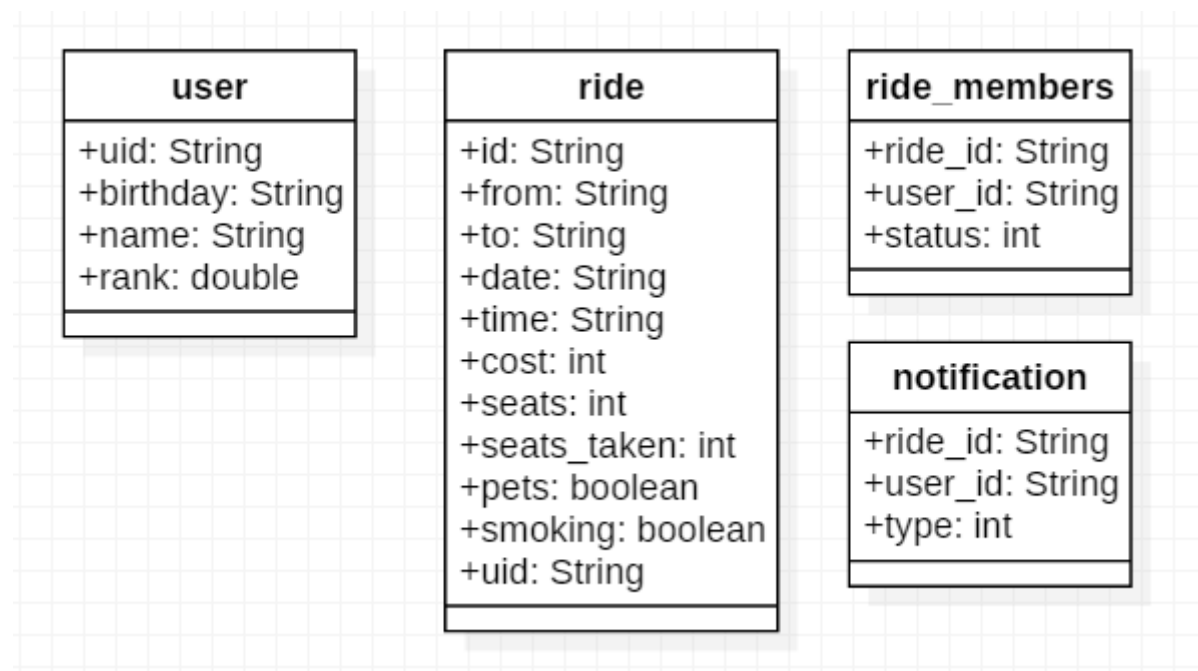
Pošto je Firebase Realtime Database tipa NoSQL baza podataka, ona se u stvari sastoji od jednog običnog JSON stabla, čiji čvorovi predstavljaju tabele i podatke koji se u njoj nalaze.

Generalno, baza podataka aplikacije My Ride je sačinjena od 4 tabele i to:

1. Users
2. Rides
3. Ride_members
4. Notifications

Svaka od ovih tabela se sastoji iz niza podataka istog tipa koji predstavljaju određene objekte. Tabela *users* sadrži objekte tipa User, *rides* tipa Ride itd.

Sledi šema baze podataka:



Slika 7. Šema baze podataka

Tabela *users* sadrži neophodne informacije o korisnicima.

Tabela *rides* sadrži neophodne informacije o vožnjama.

Tabela *ride_members* povezuje određenu vožnju sa njenim putnicima.

Tabela *notifications* sadrži detalje notifikacije za svakog korisnika.

Osnovni entiteti koji se nalaze u bazi podataka su korisnici, vožnje, notifikacije. Detaljno opisani atributi ovih entiteta dati su u nastavku.

Svaki korisnik karakteriše se sledećim podacima:

- **uid** – jedinstveni identifikacioni broj koji se dodeljuje korisniku u toku registracije.
- **birthday** – datum rođenja korisnika.
- **name** – ime korisnika.
- **rank** – srednja ocena korisnika od strane drugih putnika.

Svaka vožnja karakteriše se sledećim podacima:

- **id** – jedinstveni identifikacioni broj koji se dodeljuje vožnji prilikom njenog kreiranja.
- **from** – grad ili mesto odakle kreće vožnja.
- **to** – grad ili mesto gde se završava vožnja.
- **date** – datum vožnje.
- **time** – vreme vožnje.

- **cost** – ukupna cena vožnje koja uključuje cenu goriva, putarine i svega ostalog.
- **seats** – ukupan broj slobodnih mesta u vozilu.
- **seats_taken** – broj trenutno zauzetih mesta u vozilu.
- **pets** – da li je dozvoljeno unošenje kućnih ljubimaca u vozilo.
- **smoking** – da li je dozvoljeno pušenje u vozilu.
- **uid** – jedinstveni identifikacioni broj vozača, odnosno korisnika koji je ponudio vožnju.

Svaka notifikacija karakteriše se sledećim podacima:

- **ride_id** – jedinstveni identifikacioni broj vožnje za koju je notifikacija vezana.
- **user_id** – jedinstveni identifikacioni broj korisnika kome treba da se prikaže notifikacija.
- **type** – tip notifikacije.

Za korišćenje ovog tipa baze podataka i rad sa njom preko Firebase platforme bilo je potrebno uključiti *firebase database* modul u **build.gradle** fajlu. Primer uključivanja ovog modula i pribavljanje reference na određenju tabelu u bazi dat je u nastavku:

```
compile 'com.google.firebase:firebase-database:11.4.2'
```

```
FirebaseDatabase database = FirebaseDatabase.getInstance();
DatabaseReference myRef = database.getReference("users");
```

4.3.2 SERVERSKI DEO APLIKACIJE

Serverski deo, pored baze podataka, sadrži još dve bitne komponente. Jedna je *Authentication* a druga *Storage*.

Authentication je servis koji omogućava kreiranje (registrowanje) novih korisnika, prijavljivanje postojećih korisnika, brisanje korisnika, ažuriranje podataka o korisniku itd. U nastavku je dat primer dodavanja *firebase authentication* modula kako bi mogao da se koristi ali i funkcija za registrowanje korisnika putem email-a i password-a:

```
compile 'com.google.firebase:firebase-auth:11.4.2'
```

```
mAuth = FirebaseAuth.getInstance();

mAuth.createUserWithEmailAndPassword(email, password)
    .addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {
        @Override
        public void onComplete(@NonNull Task<AuthResult> task) {
            if (task.isSuccessful()) {
                // Sign in success, update UI with the signed-in user's information
                Log.d(TAG, "createUserWithEmail:success");
                FirebaseUser user = mAuth.getCurrentUser();
                updateUI(user);
            } else {
                // If sign in fails, display a message to the user.
                Log.w(TAG, "createUserWithEmail:failure", task.getException());
                Toast.makeText(EmailPasswordActivity.this, "Authentication failed.",
                    Toast.LENGTH_SHORT).show();
                updateUI(null);
            }
        }
    });
```

Storage je servis koji omogućava rad sa multimedijalnim fajlovima, odnosno njihovo skladištenje i preuzimanje sa Firebase servera. U nastavku je dat primer dodavanja *firebase storage* modula i funkcija za uploadovanje fajla.

```
compile 'com.google.firebase:firebase-storage:11.4.2'

// Create a storage reference from our app
StorageReference storageRef = storage.getReference();

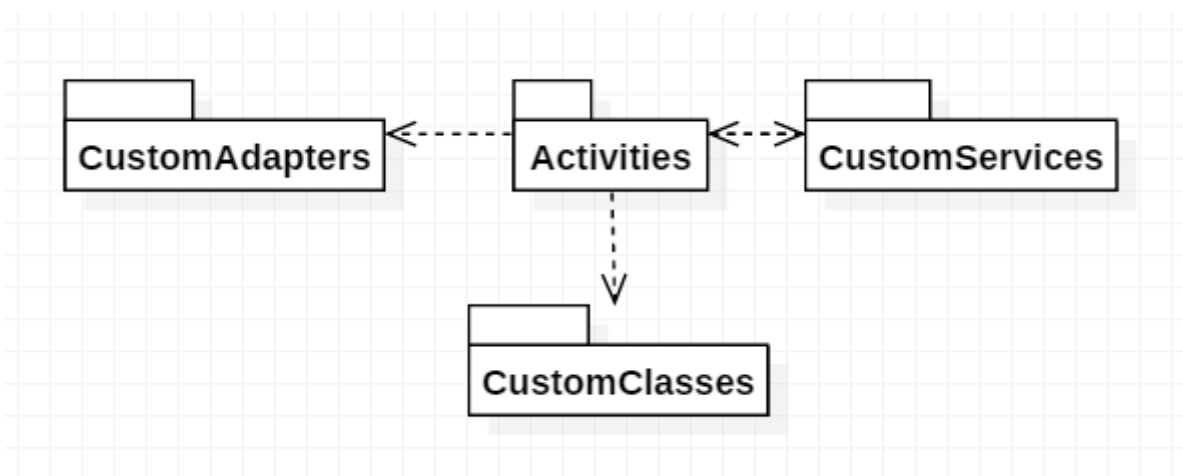
// Create a reference to "mountains.jpg"
StorageReference mountainsRef = storageRef.child("mountains.jpg");

InputStream stream = new FileInputStream(new File("path/to/images/rivers.jpg"));

uploadTask = mountainsRef.putStream(stream);
uploadTask.addOnFailureListener(new OnFailureListener() {
    @Override
    public void onFailure(@NonNull Exception exception) {
        // Handle unsuccessful uploads
    }
}).addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {
    @Override
    public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
        // taskSnapshot.getMetadata() contains file metadata such as size, content-type, and
        // download URL.
        Uri downloadUrl = taskSnapshot.getDownloadUrl();
    }
});
```

4.3.3 KLIJENTSKI DEO APLIKACIJE

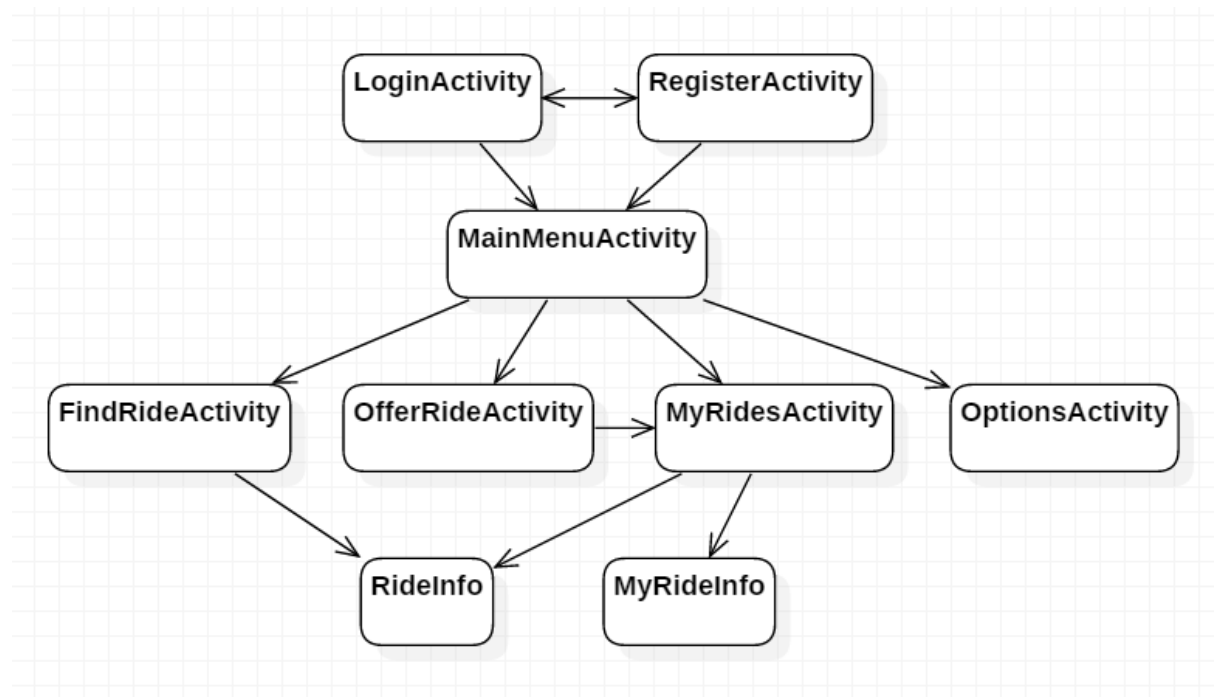
Klijentska aplikacija podeljena je u 4 paketa: *Activities*, *CustomClasses*, *CustomAdapters* i *CustomServices*. Na sledećoj slici je dat šematski prikaz ovih paketa i njihove međusobne zavisnosti.



Slika 8. Šematski prikaz paketa

PAKET ACTIVITIES

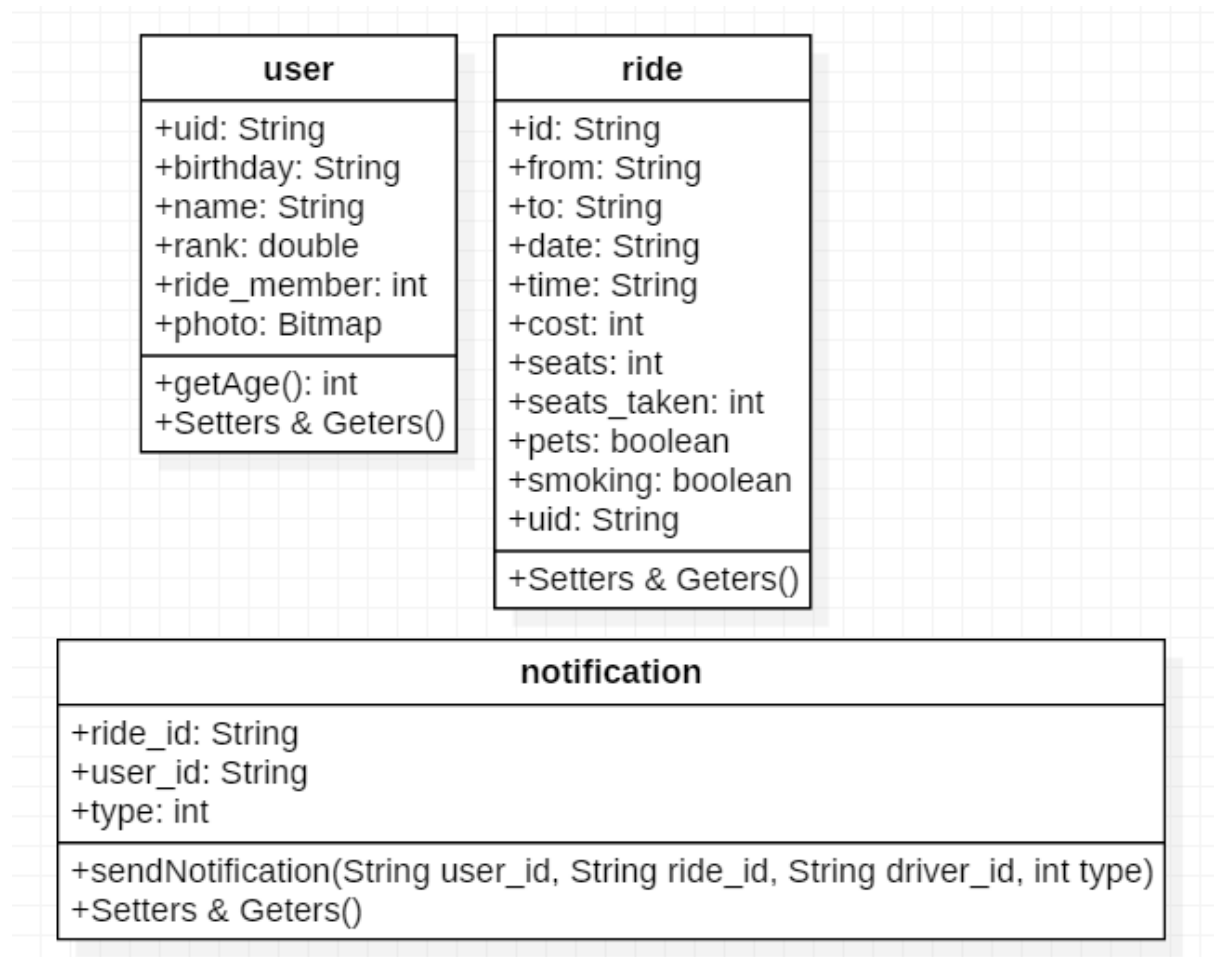
Za svaki ekran aplikacije odgovorna je po jedna **Activity** klasa. **Activity** je osnovna klasa u razvoju Android aplikacija, koja omogućava interakciju sa korisničkim interfejsom. Zbog velikog broja ekrana (i posledično, **Activity** klasa), i visokog stepena povezanosti između njih, sve ove klase smeštene su u jedan paket. Na dijagramu koji sledi prikazani su mogući tokovi aplikacije, gde iz svake aktivnosti strelice vode do onih aktivnosti koje se iz nje mogu pokrenuti.



Slika 9. Dijagram toka aplikacije

PAKET CUSTOM CLASSES

Ovaj paket sadrži klase koje predstavljaju modele entiteta iz baze podataka. U okviru tih klasa, osim atributa za skladištenje neophodnih informacija, nalaze se i javni getteri i seteri koji među ostalom služe i da mapiraju objekat koji treba da se upiše u bazu podataka, i obrnuto kada se pročitaju iz nje. Sledi dijagram klasa iz ovog paketa na kome su prikazani atributi i najvažnije metode.



Slika 10. Dijagram klasa – paket Custom Classes

PAKET CUSTOM ADAPTERS

U aplikaciji veliki broj ekrana prikazuje različite liste – ponuđenih vožnji, pretraženih vožnji i putnika. Zbog potrebe da svaka lista prikazuje korisnički definisane informacije, bilo je potrebno za svaku listu kreirati odgovarajući adapter, koji nasleđuje **RecyclerViewAdapter**, klasu koja se koristi za popunjavanje **RecyclerView**-a. U ovoj aplikaciji se koristi **RecyclerView** zato što je bolji po performansama od **ListView**-a, jer on „reciklira“ svoje view-ove prilikom prikazivanja liste a ne pravi onoliki broj koliko ima stavki u listi.

Ovaj paket sadrži sledeće klase:

- **RideRecyclerViewAdapter** – ovaj adapter koristi se u okviru klase **FindRideActivity** za prikaz pronađenih vožnji, i u okviru klase **MyRidesActivity** za prikaz ponuđenih vožnji i vožnje za koje se korisnik prijavio
- **MemberRecyclerViewAdapter** – ovaj adapter se koristi u klasi **MyRideInfo** za prikaz liste putnika koji su se prijavili za tu vožnju

PAKET CUSTOM SERVICES

Ovaj paket sadrži samo klasu **FirebaseNotificationService** koja je zadužena za prikazivanje notifikacija korisniku. Ona je direktno povezana sa *notifications* tabelom i automatski prikazuje novu notifikaciju korisniku. Pošto ova klasa nasleđuje **Service** klasu, to joj omogućava da se izvršava u pozadini, čak i kada aplikacija nije otvorena.

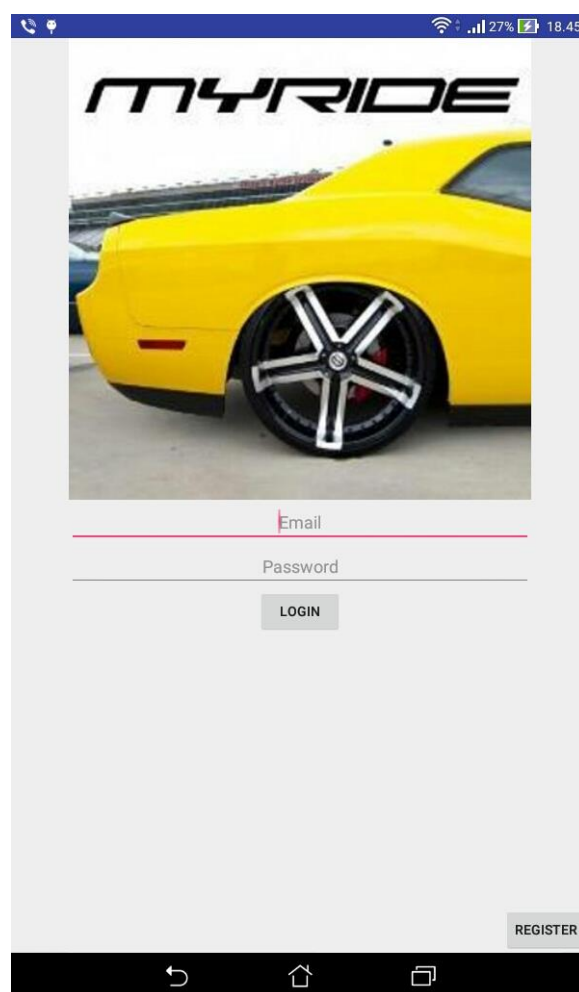
Druga funkcionalnost ove klase jeste proveravanje da li je nova vožnja sa zadatim karakteristikama kreirana i ako jeste obaveštava korisnika. To se koristi kada korisnik pretražuje neku vožnju ali trenutno nema nijedne takve vožnje ponuđene, onda se korisniku nudi mogućnost da ako to želi, bude obavešten kada takva vožnja bude dostupna.

5. MY RIDE - OPIS FUNKCIONALNOSTI

U ovom poglavlju biće dat pregled svih ekrana aplikacije sa detaljnim objašnjenjem funkcionalnosti svakog od njih.

5.1 POČETNI EKRAN

Prilikom prvog otvaranja aplikacije prikazuje se početni ekran sa logom aplikacije My Ride. On je ujedno i ekran za prijavljivanje korisnika na sistem. U donjem desnom uglu je dugme *Register* koje vodi na ekran za registraciju korisnika ukoliko korisnik želi da napravi novi nalog. Da bi se prijavio na sistem, korisnik mora da unese svoj email i šifru u odgovarajuća polja i klikne na dugme *Login*. Ako korisnik nije uneo neku od ovih dveju informacija ili su informacije pogrešne, korisniku se na dnu ekrana prikazuje odgovarajuća poruka sa greškom. Ukoliko je korisnik prethodno bio već prijavljen na sistem, ovaj ekran se neće prikazati već će se korisniku prikazati ekran sa glavnim menijem. Na slici ispod je data slika početnog ekrana.



Slika 11. Početni ekran

5.2 EKRAN ZA REGISTRACIJU

Na ekranu za registraciju je forma koju korisnik mora da popuni kako bi kreirao novi nalog i time mogao da se prijavi na sistem.

Od neophodnih informacija korisnik mora da unese sledeće:

1. Validnu email adresu
2. Ime korisnika - koje će moći da vide ostali korisnici
3. Datum rođenja - kako bi moglo ostalim korisnicima da se prikaže starosne godine
4. Šifra – koja mora biti minimum sačinjena od 6 karaktera

Kada se klikne na dugme *Register*, kreira se novi nalog korisnika i on biva automatski prijavljen na sistem i prikazuje mu se ekran sa glavnim menijem. Ukoliko korisnik nije uneo sve informacije ili neke informacije nisu validne, prikazaće se odgovarajuća poruka na dnu ekrana sa opisom greške. Na slici ispod je data slika ekrana za registraciju.

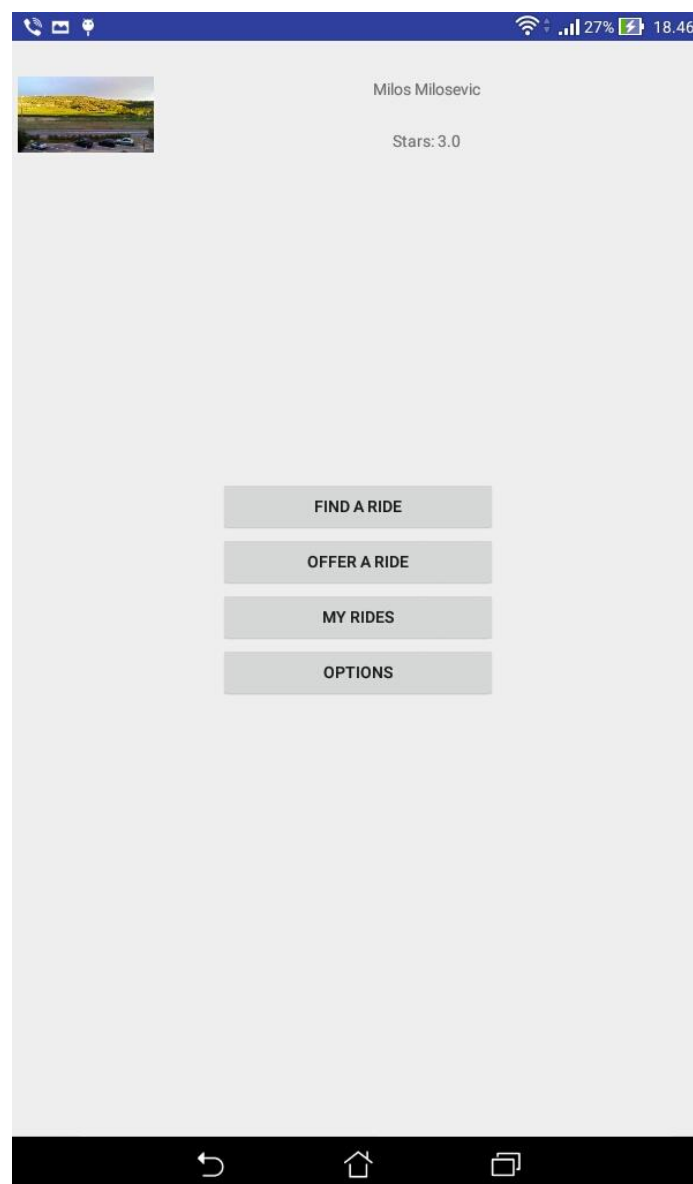
Slika 12. Ekran za registraciju

5.3 GLAVNI MENI

Kada se korisnik prijavi na sistem, otvara mu se ekran sa glavnim menijem. Ukoliko je korisnik prethodno bio već prijavljen, aplikacija će automatski da prikaže ovaj ekran. Na vrhu ekrana se nalazi profilna slika trenutno prijavljenog korisnika (ukoliko je ima) zajedno sa osnovnim informacijama.

U centru ekrana se nalaze sledeće stavke:

- **Find a Ride** – dugme koje vodi na ekran za pronalaženje vožnji po zadatim kriterijumima
- **Offer a Ride** – dugme koje vodi na ekran gde trenutni korisnik može da ponudi svoju vožnju
- **My Rides** – dugme koje vodi na ekran u kome se izlistavaju sve aktuelne vožnje za koje se korisnik prijavio i vožnje koje je on ponudio
- **Options** – dugme koje vodi na ekran sa podešavanjima gde korisnik može da promeni svoju profilnu sliku ili da se odjavi sa sistema



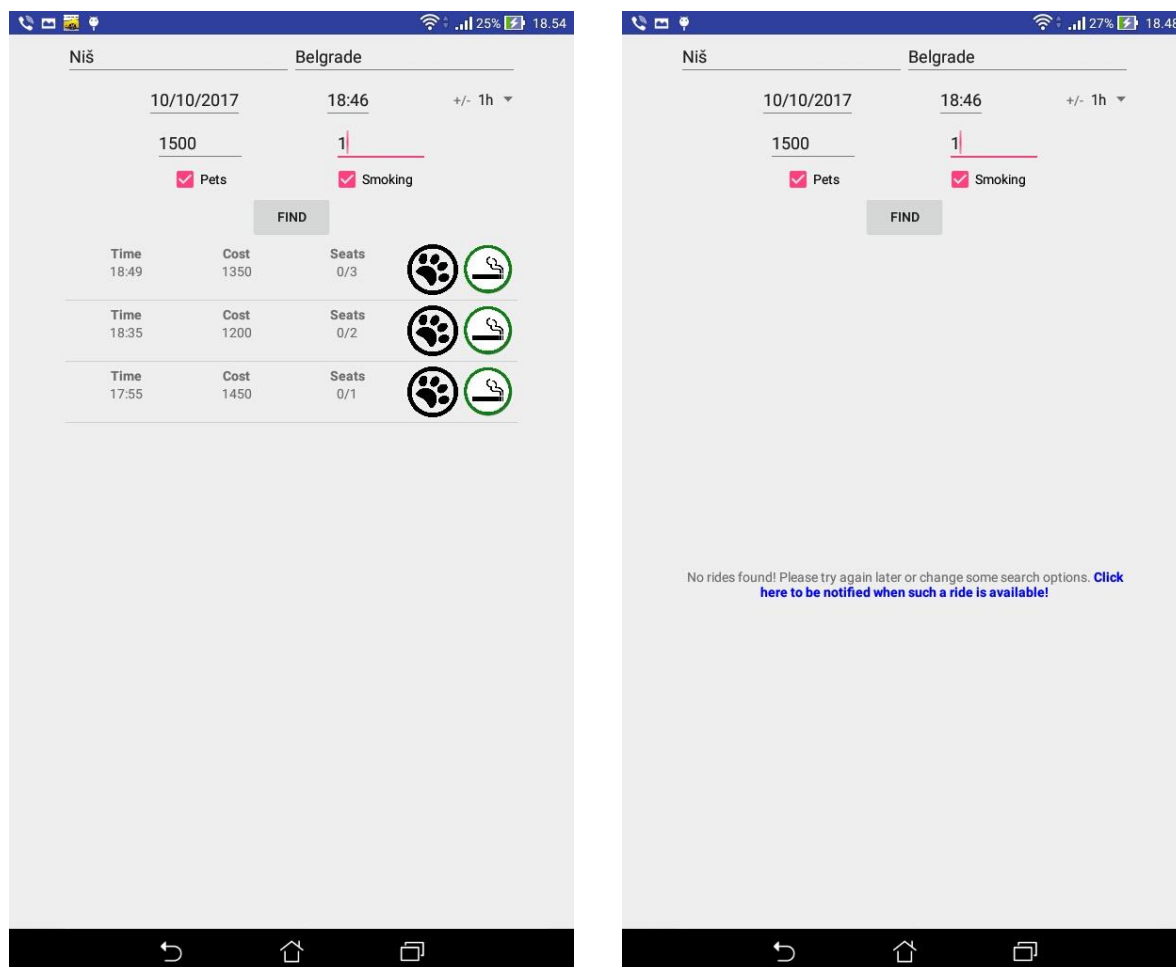
Slika 13. Ekran glavnog menija

5.4 EKRAN ZA NALAŽENJE VOŽNJI

Na ekranu se nalaze polja koja korisnik popunjava kako bi pretražio i našao specifičnu vožnju koja njemu odgovara. Stavke koje su neophodne za pretragu su mesto početka i kraja vožnje, datum i okvirno vreme vožnje. Klikom na polje za datum i vreme se otvara dijalog gde korisnik bira odgovarajuću stavku. Ako korisnik pak pokuša da izabere neki pređašnji datum, prikazaće se poruka o grešci. Pored polja za vreme postoji i dodatna mogućnost gde korisnik može da izabere fleksibilnost od +/- 1,2 ili 3 sata u zavisnosti od vremena kojeg je prethodno izabrao. Korisniku se nude i dodatne stavke koje mogu da suze pretragu vožnji ako on to želi. Tu spadaju maksimalni iznos troškova vožnje, minimalan broj sedišta koji je slobodan i na kraju može da štiklira ako želi da pretraži vožnje koje dozvoljavaju kućne ljubimce i/ili pušenje u vozilu. Klikom na dugme *Find* korisniku se prikazuje lista pronađenih vožnji po zadatim kriterijumima, ukoliko je sve popunio kako treba, inače se ispisuje odgovarajuća greška. Sledi slika ekrana za nalaženje vožnji

Slika 14. Ekran za nalaženje vožnji

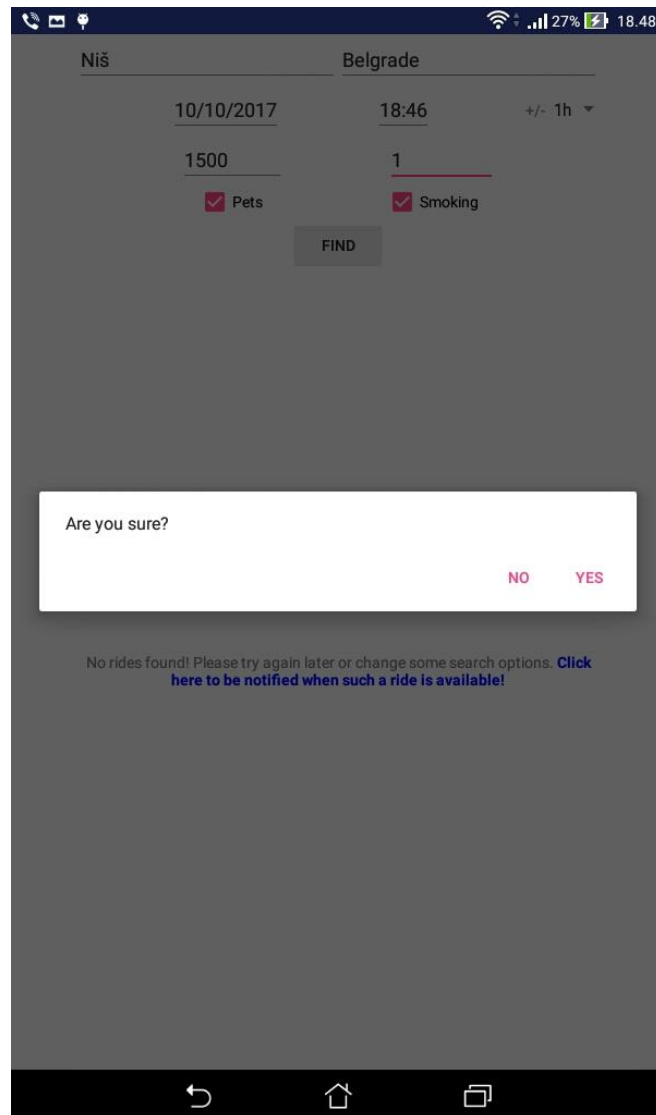
Slede slika ekrana na koje je prikazana lista pronađenih vožnji, kao i slika ekrana ako nije pronađena ni jedna vožnja po zadatim kriterijumima.



Slika 15. Lista pretraženih vožnji

Ukoliko ima pronađenih vožnji one se prikazuju u listi sortirane po najbližem vremenu od zadatog. Pored vremena početka vožnje, prikazuju se i dodatne informacije tipa ukupna cena troškova, popunjenost mesta u vozilu, kao i da li je dozvoljeno unošenje kućnih ljubimaca i pušenje u vozilu. Klikom na neku vožnju se otvara ekran sa svim informacijama vezanih za tu vožnju, gde korisnik može da pošalje zahtev vozaču ako to želi.

Ukoliko nije pronađena baš ni jedna vožnja, korisniku se ispisuje poruka da proba da promeni neki parametar pretrage ili jednostavno da pokuša malo kasnije. Nudi mu se i mogućnost da bude obavešten ako uskoro bude ponuđena vožnja koja ispunjava kriterijume pretrage. Korisniku se klikom na plavi deo poruke prikazuje dijalog gde se od korisnika traži da potvrdi tu akciju. Sledi slika ekrana gde je prikazan ovaj dijalog.



Slika 16. Dijalog u kome se zahteva potvrda radi obaveštenja

Ako korisnik potvrdi biće obavešten čim takva vožnja bude bila dostupna, a ako odbije dijalog se svakako zatvara. Korisnik u bilo kom trenutku može da isključi buduće obaveštenje za konkretnu vožnju ako prihvati da bude obavešten za neku drugu ili pak odbije, što će prebrisati trenutnu vožnju.

5.5 EKRAN ZA PRIKAZ PRONAĐENE VOŽNJE

Na ovom ekranu se prikazuje detaljne informacije o pronađenoj vožnji. U gornjoj polovini se nalazi profilna slika vozača koji nudi datu vožnju, zajedno sa njegovim ličnim informacijama. U donjoj polovini su sve informacije vezane za izabranu vožnju. Na dnu ekrana se nalaze dva dugmeta, *Close* koje zatvara ovaj prozor i vraća se na prethodni ekran, a drugo *Join ride* koje će da pošalje zahtev vozaču da trenutni korisnik želi da putuje sa njim.

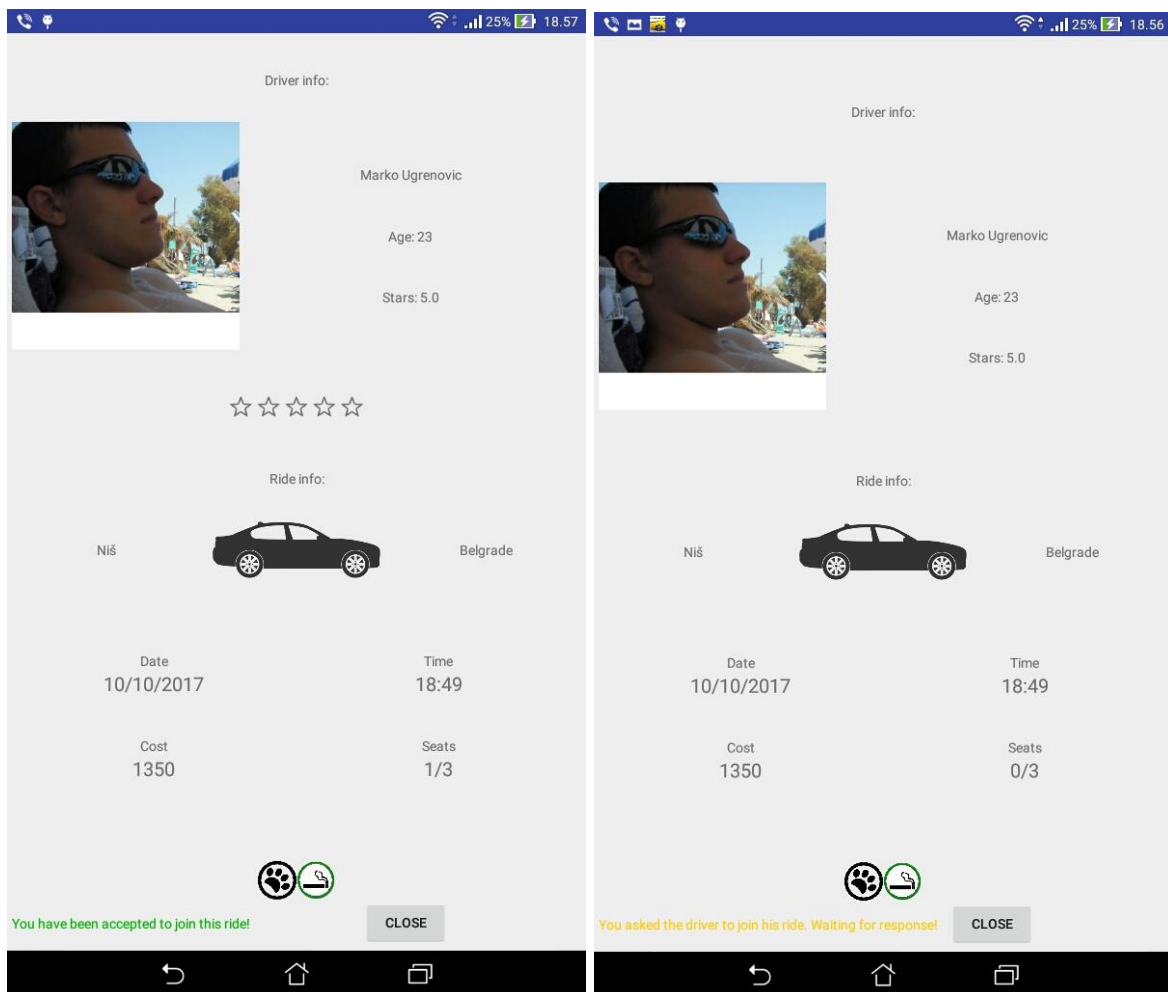
Kada korisnik pošalje zahtev, umesto tog dugmeta se ispisuje statusna poruka koja može biti sledećeg tipa:

- **Čekanje na odgovor** – korisnik je poslao zahtev vozaču koji biva obavešten o tome
- **Odbijen** – vozač je odbio korisnikom zahtev
- **Prihvaćen** – vozač je odobrio korisnikov zahtev

Dok čeka na odgovor, korisnik može u bilo kom trenutku da prati promene za tu vožnju iz liste prijavljenih vožnji na ekranu *Moje vožnje*. Svakako, korisnik će biti obavešten o prihvaćenom ili odbijem zahtevu čim vozač to bude uradio.

Ako je korisnikov zahtev odbijen, on neće moći ponovo da pošalje zahtev za istu vožnju.

Ako je korisnikov zahtev prihvaćen, njemu se ispod informacija vozača nudi dodatna opcija da ocenu vozača odnosno vožnju. On to može da učini odmah, ako već ima iskustva sa tim vozačem, ili tek nakon uspešno završene vožnje. Korisnik može da oceni vozača sa 1 do 5 zvezdica, ali najkasnije dva dana od datuma početka vožnje. U tom slučaju korisnik neće moći više da oceni vozača jer će se automatski vožnja izbrisati sa servera. Korisnik nije u obavezi da oceni vozača-vožnju.



Slika 17. Primer ekrana kada je putnik prihvaćen i dok čeka na potvrdu

5.6 EKRAN ZA KREIRANJE NOVE VOŽNJE

Na ovom ekranu se nalazi forma koju korisnik mora da popuni sa neophodnim informacijama kako bi kreirao (ponudio) novu vožnju. Klikom na dugme *Offer*, vožnja se kreira i korisniku se prikazuje ekran *Moje vožnje* gde može da vidi sve vožnje koje je ponudio. Ako korisnik nije uneo sve informacije ili neke informacije nisu validne ispisuje se poruka sa greškom na dnu ekrana. Sledi slika ekrana sa proizvoljno unetim informacijama vožnje.

The screenshot shows a mobile application interface for creating a new trip. At the top, there is a status bar with icons for signal, Wi-Fi, and battery (25%), and the time 19:00. The form itself is a light gray rectangle with the following fields and elements:

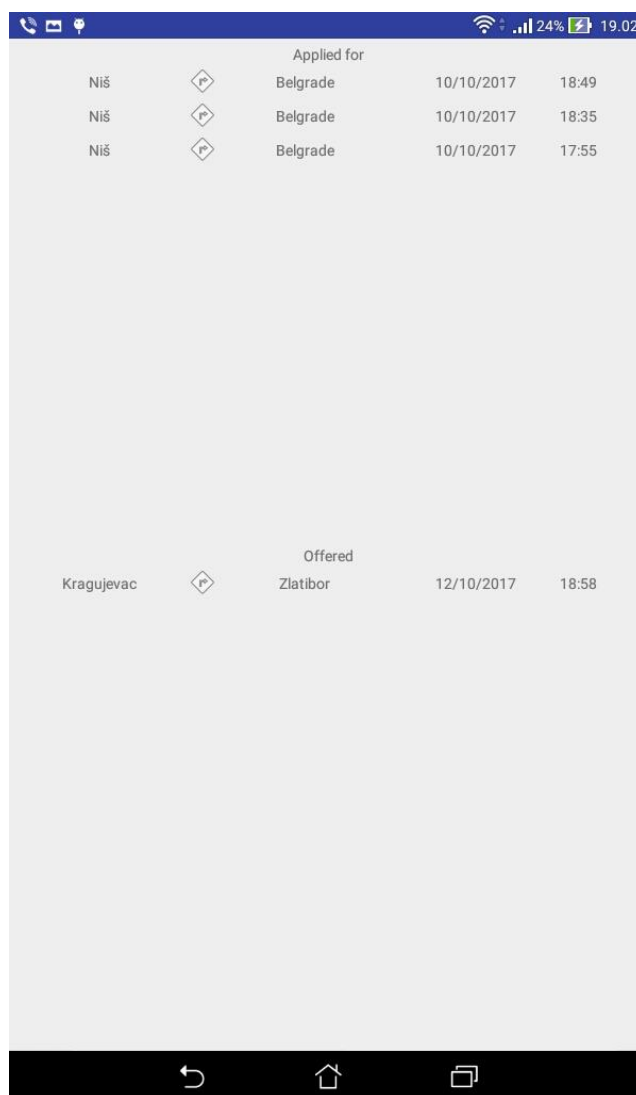
- Origin: A text input field containing "Kragujevac".
- Destination: A text input field containing "Zlatibor".
- Date: A text input field containing "12/10/2017".
- Time: A text input field containing "18:58".
- Price: A text input field containing "2680".
- Number of seats: A label "Number of seats:" followed by a text input field containing "3".
- Pets: A checkbox labeled "Pets", which is unchecked.
- Smoking: A checkbox labeled "Smoking", which is unchecked.
- Submit button: A gray button labeled "OFFER" at the bottom center.

At the very bottom of the screen is a black navigation bar with three white icons: a back arrow, a home icon, and a recent apps icon.

Slika 18. Ekran za kreiranje nove vožnje

5.7 EKRAN MOJE VOŽNJE

Na ovom ekranu su prikazane dve liste. Prva je lista vožnji na koje se korisnik prijavio ili na kojima je njegov zahtev prihvaćen, dok je ispod lista vožnji koje je on ponudio. Vožnje su u listi prikazane samo sa osnovnim informacijama i sortirane su po datumu. Ako korisnik želi da vidi detaljnije informacije o konkretnoj vožnji, klikom na određenu vožnju se otvara ekran za prikaz svih informacija vezanih za tu vožnju. Sledi slika ekrana.

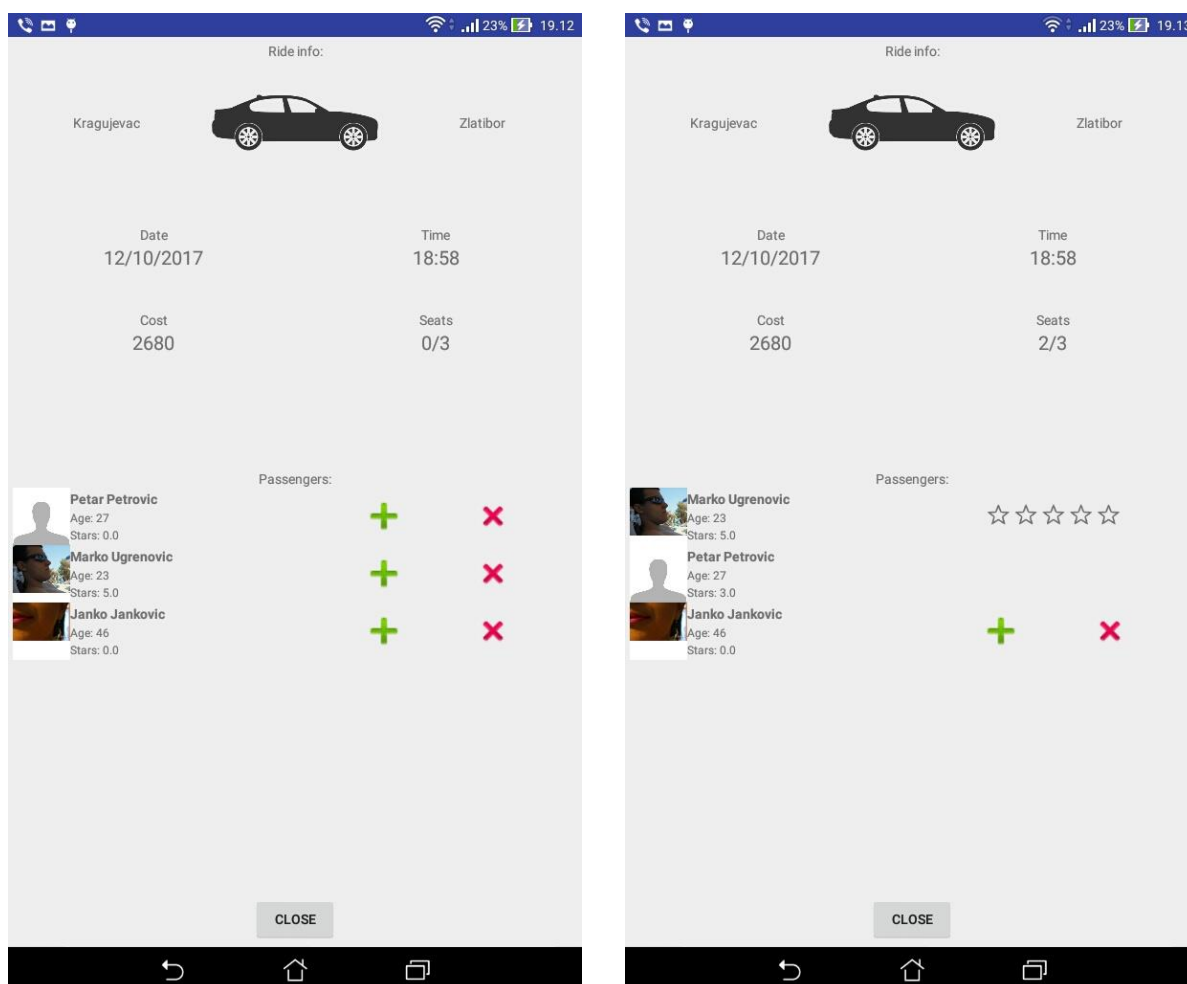


Slika 19. Ekran – Moje vožnje

5.8 EKRAN ZA PRIKAZ PONUĐENE VOŽNJE

Na ovom ekranu se u prvom delu prikazuju informacije vezane za vožnju koju je korisnik ponudio, dok u drugom delu je spisak korisnika (putnika) sa osnovnim informacijama, koji su poslali zahtev ili kojima je zahtev već prihvaćen. Korisniku (u ovom slučaju vozaču) se daje mogućnost da prihvati korisnikov zahtev ili ga odbije. Ukoliko odbije nečiji zahtev (klikom na crveni iks), taj korisnik nestaje sa liste. Ukoliko vozač prihvati nečiji zahtev (klikom na zeleni plus), umesto ovih ikonica se pojavljuju zvezdice koje daju mogućnost vozaču da oceni tog putnika. Ista pravila važe kao i kad sam putnik ocenjuje vozača. On može ali i ne mora da oceni nekog putnika, kao i da to može učiniti odmah ili tek nakon završetka vožnje, ali svakako najkasnije do 2 dana po završetku vožnje kada će vožnja biti uklonjena sa servera. Ako se prijavio veći broj od slobodnih mesta u vozilu, kada se popuni poslednje mesto svi ostalo zahtevi ostalih korisnika se automatski odbijaju. Korisnicima se u svakom slučaju šalje obaveštenje da li su prihvaćeni ili odbijeni.

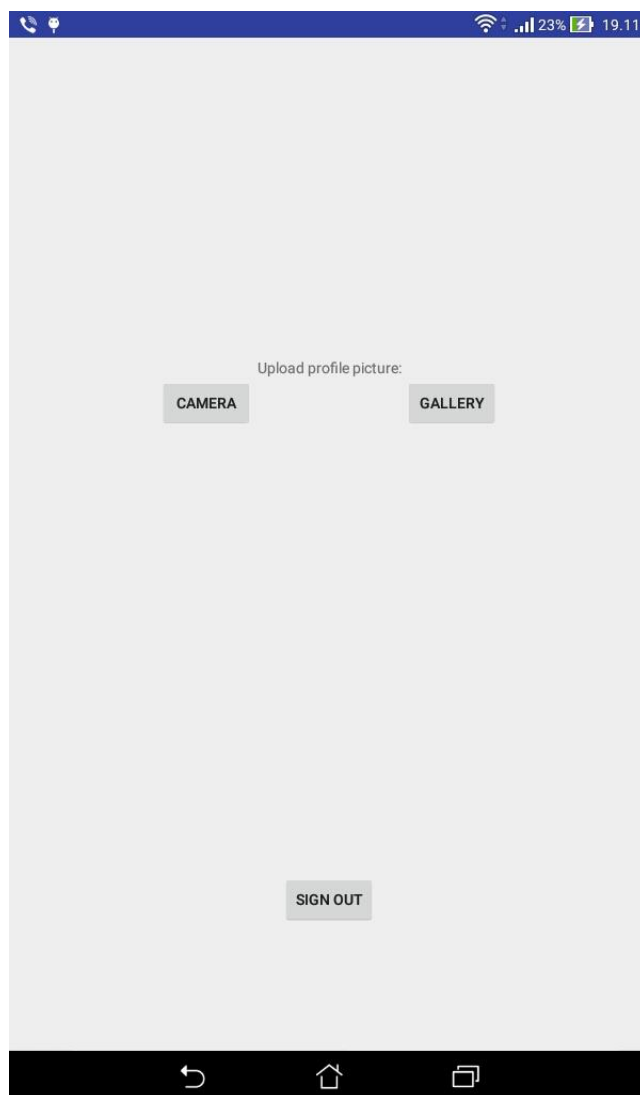
Slede dve slike primera ekrana gde korisnik vidi listu putnika, od kojih je neke odbio a neke prihvatio i može da ih oceni.



Slika 20. Prikaz ekrana za ponuđenu vožnju

5.9 EKRAN ZA PODEŠAVANJA

Na ovom ekranu se nude dve funkcionalnosti. Prva da korisnik promeni svoju profilnu sliku, da li biranjem fotografije iz galerije ili direktno sa kamere, a druga da se odjavi sa sistema u kom slučaju će se otvoriti početni ekran za ponovno prijavljivanje na sistem. Sledi slika ekrana sa osnovnim podešavanjima.



Slika 21. Ekran sa podešavanjima

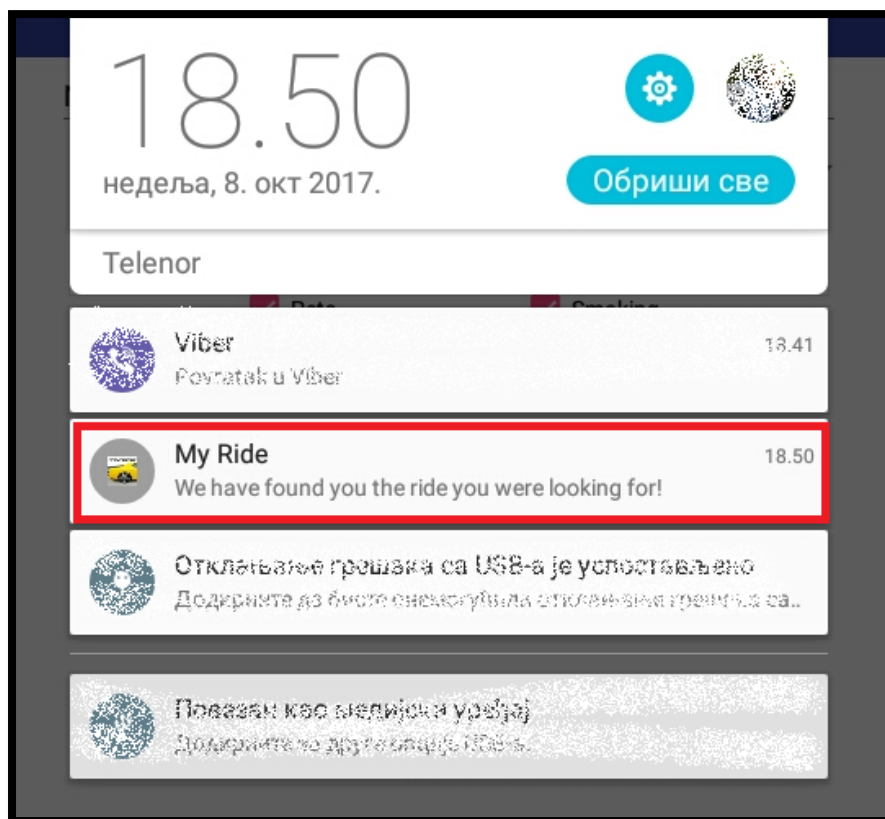
5.10 NOTIFIKACIJE

Notifikacije (obaveštenja) u aplikaciji My Ride su implementirane kao servis. To znači da će korisniku izaći notifikacija čak i ako nije trenutno u aplikaciji. Generalno, u aplikaciji My Ride postoje 4 vrsta obaveštenja:

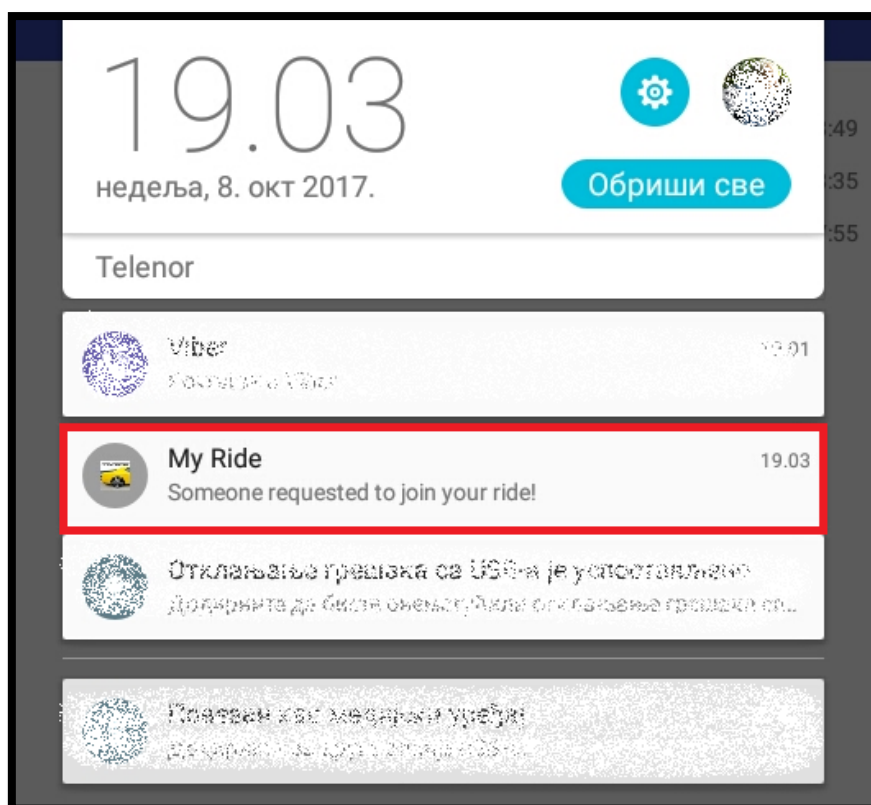
- Obaveštenje za vozača koje mu kazuje da je neko od putnika poslao zahtev za njegovu vožnju
- Obaveštenje za putnika ako mu je vozač odbio zahtev za vožnju
- Obaveštenje za putnika ako mu je vozač prihvatio zahtev za vožnju
- Obaveštenje za korisnika ako je ponuđena neka nova vožnja koja zadovoljava kriterijume koje je on zadao prilikom pretrage a nije tada našao nijednu takvu vožnju, pa je izabrao da bude obavešten kada ona bude bila dostupna.

Notifikacije se sastoje od ikonice aplikacije, imena aplikacije i poruke koja zavisi od vrste notifikacije. Notifikacije se korisniku prikazuju u standardnom delu za notifikacije Android uređaja i klikom na nju korisniku se otvara aplikacija My Ride i prikazuje mu se odgovarajući ekran u zavisnosti od vrste notifikacije. Ako je obaveštenje bilo za vozača, otvara mu se *ekran za prikaz ponuđene vožnje* sa informacijama njegove vožnje i listom putnika i korisnika koji su uputili zahtev za vožnju. Ako je obaveštenje bilo za putnika ili korisnika, vodi ga na *ekran za prikaz pronađene vožnje* gde se prikazuje informacije o vožnji za koju je bio zainteresovan ili je njegov zahtev prihvaćen/odbijen.

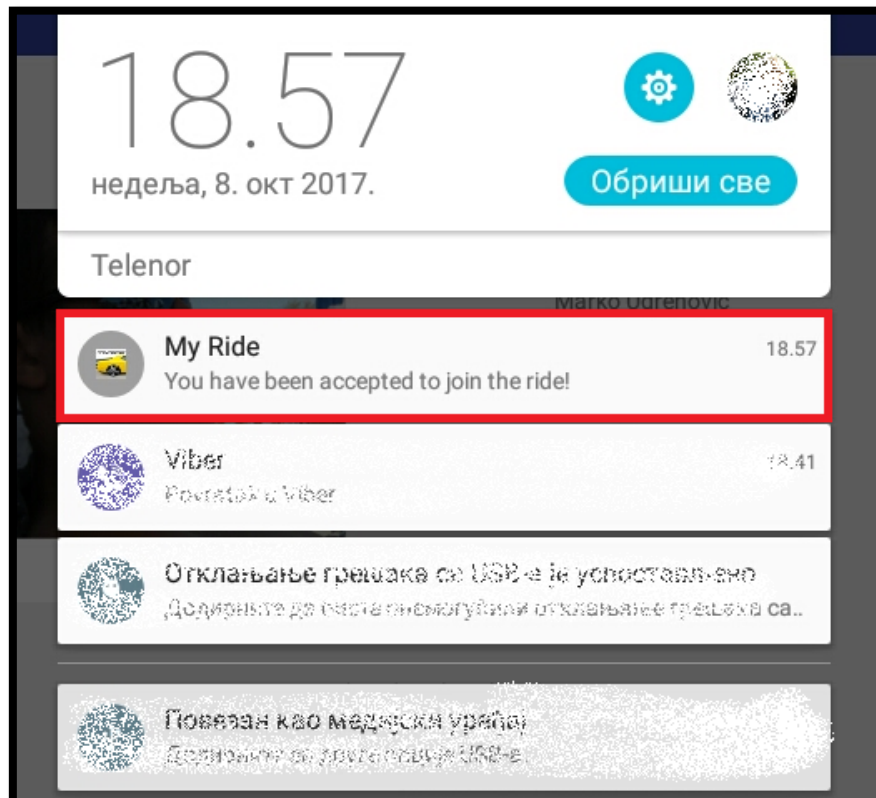
Slede primeri ekrana na kojima su prikazane neke od ovih notifikacija:



Slika 22. Obaveštenje o pronalasku nove vožnje



Slika 23. Obaveštenje da je novi putnik poslao zahtev



Slika 24. Obaveštenje da je korisnikov zahtev za vožnju prihvaćen

6. ZAKLJUČAK

U ovom radu predstavljen je značaj korišćenja mobilnih aplikacija u saobraćaju i turizmu. One pomažu putnicima da lakše pronađu mesta koja žele da posete, rezervišu smeštaj, isplaniraju rute ili pronađu odgovarajući prevoz.

Posebna pažnja posvećena je kategoriji mobilnih aplikacija za deljenje vožnje. Njihov glavni cilj je da se postaraju da svaki putnik nađe odgovarajući prevoz, a svaki vozač nađe svoje putnike. Uglavnom su to aplikacije koje omogućavaju korisnicima da brzo i jednostavno podele neku vožnju ili pak nađu vožnju koja im odgovara.

Karakteristike koje najviše doprinose kvalitetu i uspehu ovakvih aplikacija su mogućnosti geopozicioniranja korisnika, kao i asinhrona komunikacija sa korisnicima preko Interneta.

U radu su spomenute neke od reprezentativnih aplikacija ovog tipa. Aplikacija *My Ride*, razvijena za potrebe ovog rada, poseduje slične funkcionalnosti kao i navedene aplikacije. Ipak, glavni cilj ove aplikacije bio je da korisnici što lakše i brže nađu ili podele prevoz. To im omogućava brzi odziv aplikacije koja u realnom vremenu prati promene podataka na serveru i prikazuje te podatke korisniku. Pored toga, ova aplikacija poseduje i asinhrona obaveštenja koja pomažu korisnicima da brže primete kada je neka vožnja dostupna ili kada je neko od korisnika zainteresovan sa njihovu vožnju.

Aplikacija zahteva internet konekciju kako bi u realnom vremenu ažurirala i sinhronizovala korisnikove zahteve i promene, ali nije i neophodna za pokretanje aplikacije i rad delovima aplikacije koji ne zahtevaju pristup internetu. Dobra stvar što svaka korisnikova akcija, ako nije povezan na internet, se pamti lokalno i biće prvom prilikom kada se korisnik poveže na internet ažurirana na serveru.

Kao dodatna funkcionalnost, koja nažalost trenutno nije implementirana, jeste konkretan način kako bi vozači mogli da se detaljnije dogovore sa svojim putnicima oko vožnje. To bi moglo da se uradi tako što bi kada vozač prihvati putnika mogao da vidi njegov broj mobilnog telefona, i obrnuto putnik vozačev broj, kako bi mogli da se čuju i ugovore detalje oko organizovane vožnje. Sa druge strane, ako korisnici ne bi želeli da dele svoj lični broj mobilnog telefona, mogao bi da se implementira *chat* sistem u kojem bi bili uključeni trenutno prihvaćeni putnici zajedno sa vozačem. Tako bi svi lakše mogli da se dogovaraju oko prevoza.

Naravno, dizajn same aplikacije bi definitivno bio sledeći za dopunu, jer se generalno danas aplikacije rangiraju po korisnikovom iskustvu i doživljaju same aplikacije, a dizajn tu igra veliku ulogu. Još jedna stvar koja nedostaje jeste personalizovanost. Sve je više aplikacija koje uz pomoć određenih *recommender* sistema, a na osnovu korisnikovih ranijih aktivnosti, procenjuju njegove preference. Na taj način bi *My Ride* aplikacija mogla da daje korisniku predloge i obaveštenja o vožnjama koje bi se njemu najverovatnije dopala.

7. LITERATURA

Real-time apps - <http://venturebeat.com/2013/04/05/why-your-app-needs-to-be-real-time/>

Carpooling - <https://en.wikipedia.org/wiki/Carpool>

Carsharing - <https://en.wikipedia.org/wiki/Carsharing>

Firebase Authentication for Android - <https://firebase.google.com/docs/auth/android/start/>

Firebase Realtime Database for Android - <https://firebase.google.com/docs/database/android/start/>

Firebase Storage for Android - <https://firebase.google.com/docs/storage/android/start>

The 11 Best ridesharing apps - <https://www.digitaltrends.com/mobile/best-ride-sharing-apps/>