

Share Taxi	Verzija: 1.0
Tehnička dokumentacija	Datum: 09.07.2018.

Share Taxi

Tehnička dokumentacija

Verzija 1.0

Student: Vedran Ivić

Kolegij: Razvoj mobilnih aplikacija

Profesor: Josip Balen

Asistent: Bruno Zorić

Share Taxi	Verzija: 1.0
Tehnička dokumentacija	Datum: 09.07.2018.

SADRŽAJ

1. OPIS RAZVIJENOG PROIZVODA	3
2. KORISNIČKI ZAHOTJEVI	4
2.1 Popis zahtjeva	4
2.2 Slučajevi korištenja	5
2.3 Detalji i primjeri	9
2.4 Mock-up	13
3. MODEL PODATAKA	15
4. TEHNIČKE ZNAČAJKE	17
4.1 Korištene tehnologije	17
5. UPUTE ZA KORIŠTENJE	18
6. LITERATURA	21

Share Taxi	Verzija: 1.0
Tehnička dokumentacija	Datum: 09.07.2018.

TEHNIČKA DOKUMENTACIJA

1. OPIS RAZVIJENOG PROIZVODA

Share Taxi je aplikacija namjenjena Android mobilnim uređajima koja predstavlja svojevrsnu virtualnu platformu za dijeljenje vožnji taksijem. Korisnik se pri pokretanju aplikacije prijavljuje sa svojim računom te mu se po uspješnoj prijavi/registraciji nude dva glavna načina korištenja aplikacije. Korisnik može stvoriti i objaviti novu vožnju sa svim relevantnim podacima (polazište, odredište, datum i vrijeme polaksa, broj putnika) te čekati da netko prihvati njegovu vožnju. Moguće je pregledati, uređivati i brisati sve vlastite stvorene vožnje. Također, korisnik može i pretraživati već stvorene i objavljene vožnje drugih korisnika prema željenom polazištu/odredištu te ih prihvatiti ukoliko mu odgovaraju. Nakon što je vožnja prihvaćena, korisnik može izravno kontaktirati vlasnika putem integriranog *chata* aplikacije kako bi se dogovorili oko detalja ili riješili eventualne nejasnoće. Naravno, istu prihvaćenu vožnju moguće je i otkazati. Nadalje, korisniku je omogućen pregled svih razgovora putem direktnih poruka (*chata*) kao i automatski unos trenutne lokacije korisnika prilikom stvaranja nove ili pretrage već postojuće vožnje.



Slika 1.1 Izgled logotipa aplikacije

Share Taxi	Verzija: 1.0
Tehnička dokumentacija	Datum: 09.07.2018.

2. KORISNIČKI ZAHTRAJEVI

2.1 Popis zahtjeva

ID	Status	Prioritet	Opis	UC
			Generalni zahtjevi korisnika	
1	A	1	Prilikom prvog pokretanja aplikacije omogućena je prijava pomoću emaila ili google računa.	-
2	A	1	Korisnik ima pregled vlastitih vožnji te prihvaćenih.	UC1
3	A	1	Korisnik može stvoriti novu vožnju.	UC2
4	A	1	Korisnik može pretraživati i prihvatiti postojeće vožnje.	UC3
5	A	2	Korisniku je omogućeno uređivanje vlastitih vožnji.	UC4
6	A	1	Korisnik može kontaktirati vlasnika vožnje putem direktnih poruka (chata)	UC5
7	A	1	Moguće je obrisati vlastitu stvorenu vožnju.	UC6
8	A	1	Prihvaćenu vožnju moguće je otkazati.	UC7
9	A	2	Korisnik može koristiti vlastitu lokaciju (adresu) kao polazište/odredište na stisak gumba.	UC8
10	A	1	Sustav treba automatski brisati vožnje koje su vremenski istekle.	-
11	A	1	Sustav treba biti napravljen za Android mobilnu platformu	-

Share Taxi	Verzija: 1.0
Tehnička dokumentacija	Datum: 09.07.2018.

2.2 Slučajevi korištenja

ID slučaja	UC1
Ime	Pregled vlastitih i prihvaćenih vožnji
Opis	Korisniku se prikazuje početni zaslon (HomeFragment). Tu pronalazi vlastite vožnje u listi pod „My rides“ a ispod pod „Accepted Rides“ može vidjeti koje je vožnje dosad prihvatio. Svakim novim pritiskom na Home button navigation view, početnih se vraća an Home Scree.
Preduvjet	Korisnik mora biti prijavljen i posjedovati vožnje (vlasitite ili prihvaćene)
Glavni scenarij	<ol style="list-style-type: none"> 1. Registriran korisnik odabire Home button 2. Učitavaju se podaci i prikazuju se vožnje (vlastite pod 'My Rides“, prihvaćene („Accepted Ridees“). 3. Korisniku se prikazuje vrijeme polaska (te datum ako je različit od današnjeg), ime vlasnika, polazište, odredište te trenutni broj putnika.
Alternativni scenarij	<ol style="list-style-type: none"> 1. Otvaranjem aplikacije, navigiranjem nazad nakon liste razgovora ili prvim log-inom se dolazi na isti zaslon

ID slučaja	UC2
Ime	Stvaranje nove vožnje
Opis	Korisnik stvara novu, vlastitu vožnju, s njemu odgovarajućim parametrima, koju će podijeliti kako bi je drugi korisnici mogli prihvatiti.
Preduvjet	Korisnik mora biti prijavljen u aplikaciju
Glavni scenarij	<ol style="list-style-type: none"> 1. Korisnik odabire navigacijski gumb 'Add' 2. Otvara se fragment za upisivanje parametara nove vožnje 3. Korisnik ispunjava sva polja, u skladu s adekvatnim uputama o eventualnim pogreškama pri unosu. 4. Stiskom na gumb 'CREATE' vožnja se objavljuje i postaje javno dostupna drugim korisnicima.

Share Taxi	Verzija: 1.0
Tehnička dokumentacija	Datum: 09.07.2018.

Alternativni scenarij	1. Korisnik na zaslon za stvaranje nove vožnje dolazi uz pomoć tekstualnog linka 'Add' na početnom zaslonu (empty My Rides recyclerview) ili 'Create' na zaslonu pretraživanja.
-----------------------	---

ID slučaja	UC3
Ime	Pretraga i prihvaćanje postojećih vožnji
Opis	Korisnik pretražuje objavljene vožnje ostalih korisnika kako bi pronašao onu koja mu najviše odgovara, pretražujući po polazištu i/ili odredištu. U slučaju odgovarajuće vožnje istu prihvaća stiskom na gumb.
Preduvjet	Prijavljen korisnik
Glavni scenarij	<ol style="list-style-type: none"> 1. Korisnik unosi mjesto (adresu) željenog polazišta 2. Korisnik unosi mjesto (adresu) željenog odredišta 3. Pristiskom na gumb 'Search' prikazuju se sve relevantne vožnje. 4. Korisnik odabire željenu vožnju i prihvaća je stiskom na gumb prihvaćanja.
Alternativni scenarij	<ol style="list-style-type: none"> 1. Korisnik ne unosi adrese polazišta/odredišta i odmah stiže gumb Search 2. Prikazuju se sve trenutno slobodne i važeće vožnje

ID slučaja	UC4
Ime	Uređivanje vožnji
Opis	Korisnik uređuje prethodno stvorenu vlastitu vožnju
Preduvjet	Prijavljen korisnik, s barem jednom prethodno stvorenom vlasitom vožnjom.
Glavni scenarij	<ol style="list-style-type: none"> 1. Korisnik odlazi na početni zaslon 2. Pod 'My Rides' nalazi i odabire svoju vožnju koju želi urediti stiskom na gumb uređivanja

Share Taxi	Verzija: 1.0
Tehnička dokumentacija	Datum: 09.07.2018.

	3. Otvara se novi zaslon za uređivanje nalik onom za stvaranje 4. Ako želi odbaciti promjene, sitsnut će 'CANCEL'. U protivnom odabire 'SAVE' i promjene se odmah spremaju u bazu
Alternativni scenarij	

ID slučaja	UC5
Ime	Kontaktiranje vlasnika
Opis	Korisnik direktno kontaktira vlasnika vožnje nakon što je istu prihvatio i ona se pojavila pod njegovim 'Accepted Rides'. Pokreće chat sučelje gdje razmjenjuje poruke s drugim korisnikom (vlasnikom vožnje) oko detalja ili nejasnoća.
Preduvjet	Prihvaćena vožnja, prijavljen korisnik
Glavni scenarij	1. Korisnik odabire gumb za slanje poruke na prihvaćenoj vožnji 2. Otvara se chat razgovor s vlasnikom vožnje (prethodne poruke se učitavaju ako ih ima) 3. Korisnici međusobno komuniciraju 4. Po završetku razgovora korisnik odabire nazad strelicu.
Alternativni scenarij	1. Korisnik ulazi u chat na drugi način – koristeći messaging ikonu u gornjem desnom kutu, otvara popis svih svojih razgovora 2. Stiskom na gumb za slanje pokraj bilo kojeg od razgovora – pokreće se chat.

ID slučaja	UC6
Ime	Brisanje vlastite vožnje
Opis	Korisnik briše vožnju koju je prethodno dodao
Preduvjet	Prethodno dodana vožnja, prijavljen korisnik
Glavni	1. Korisnik odabire vožnju pod 'My Rides'

Share Taxi	Verzija: 1.0
Tehnička dokumentacija	Datum: 09.07.2018.

scenarij	<ol style="list-style-type: none"> 2. Stiskom na ikonicu za brisanje s desne strane javlja se dijalog koji će ga pitati je li siguran da želi obrisati tu vožnju 3. Odabirom 'YES' na dijalogu briše se vožnja iz baze te trenutno i sa sučelja
Alternativni scenarij	<ol style="list-style-type: none"> 1. Korisnik u dijalogu odabire 'CANCEL' 2. Ništa se ne događa, tj. vožnja nije izbrisana.

Share Taxi	Verzija: 1.0
Tehnička dokumentacija	Datum: 09.07.2018.

ID slučaja	UC7
Ime	Otkazivanje prihvaćene vožnje
Opis	Korisnik otkazuje prethodno prihvaćenu vožnju.
Preduvjet	Prihvaćena vožnja, prijavljen korisnik
Glavni scenarij	<ol style="list-style-type: none"> 1. Korisnik odabire vožnju za otkazati 3. Stiskom na gumb 'CANCEL' otvara se dijalog s upitom o brisanju. 4. Odabirom YES konačno briše svoju prisutnost kao putnika na toj vožnji
Alternativni scenarij	<ol style="list-style-type: none"> 1. Korisnik odustaje od otkazivanja vožnje odabirm 'NO' u dijalogu.

ID slučaja	UC8
Ime	Korištenje lokacije
Opis	Korisnik za adresu polazišta/odredišta korisit vlastitu lokaciju
Preduvjet	Odobren pristup lokaciji uređaja, prijavljen korisnik
Glavni scenarij	<ol style="list-style-type: none"> 1. Korisnik stišće ikonicu kraj 'From' ili 'To' polja 2. Traži se pristup lokaciji. 3. Korisnik odobrava pristup 4. Ispisuje se trenutna adresa korisnika na kojoj se nalazi
Alternativni scenarij	<ol style="list-style-type: none"> 1. Može se dogoditi da korisnik ne dopusti pristup lokaciji te funkcionalnost neće raditi sve dok isti ne odobri u postavkama 2. Može se koristiti funkcionalnost i kod stvaranje i kod pretrage

2.3 Detalji i primjeri

Glavni dio aplikacije čini MyRidesActivity – activity koji za sebe veže tri fragmenta koje razlikuje i koji se smjenjuju ovisno o funkcionalnosti – HomeFragment (za prikaz liste vlastitih i prihvaćenih vožnji), CreateFragment za stvaranje nove i uređivanje vlastite vožnje, te konačno SearchFragment za pretragu postojećih vožnji. Pri samom pokretanju HomeFragment se uvijek

Share Taxi	Verzija: 1.0
Tehnička dokumentacija	Datum: 09.07.2018.

učitava i prikazuje prvi. No, prije samog takvog načina rada dolazi proces prijave.

Prijava je odrađena uz pomoć već gotovih metoda za autentikaciju. Najprije se gradi novi UI za prijavu koji je djelomično uređen i prilagođen ovoj aplikaciji (*email button*, logotip, custom tema i sl.).

```
private void login() {
    if(!preferences.contains("UserID")) {
        startActivityForResult(
            AuthUI.getInstance()
                .createSignInIntentBuilder()
                .setAvailableProviders(providers)
                .setLogo(R.drawable.share_taxi_logo)
                .setTheme(R.style.SignInTheme)
                .build(),
            RC_SIGN_IN);
    }
}
```

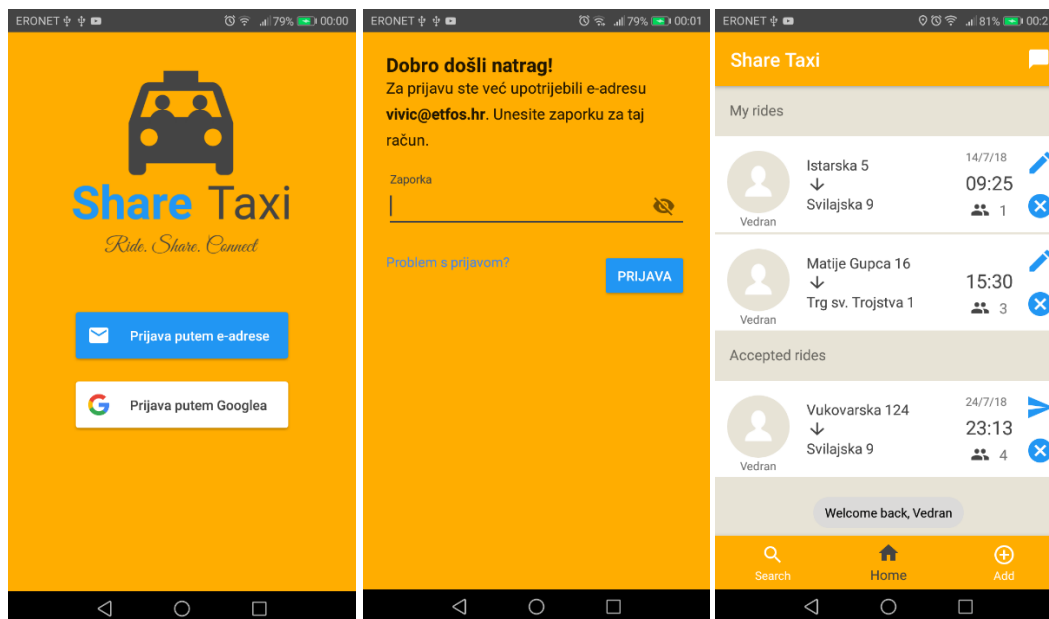
Nakon toga, u `onActivityResult` metodi koja se javlja po obavljenom pokušaju prijave, stvara se novi `FirebaseUser` korisnik koji se sprema u bazu te dobija jedinstveni ID koji se između ostalog sprema i u `sharedPreferences` kako bi se sačuvala informacija o jednom prijavljenom korisniku. Otvara se `HomeFragment` i pozdravlja korisnik (postupak je prikazan na slici 2.1).

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == RC_SIGN_IN) {
        IdpResponse response = IdpResponse.fromResultIntent(data);
        if (resultCode == RESULT_OK) {
            FirebaseAuth.getInstance().getCurrentUser();
            if (user != null) {
                preferences.edit().putString("UserID",user.getId()).apply();
                USER_ID = user.getId();

                DatabaseReference dbUsers =
                FirebaseDatabase.getInstance().getReference().child("Users");
                User dbUser = new
                User(user.getId(),user.getEmail(),user.getDisplayName(),user.getProviders().get(0));
                dbUsers.child(USER_ID).setValue(dbUser);

                loadFragment(new HomeFragment());
                Toast.makeText(this,"Welcome, "+user.getDisplayName().split(" (?!.*)") [0],Toast.LENGTH_SHORT).show();
            }
        } else{
            Toast.makeText(this, "Error: Sign in failed",Toast.LENGTH_SHORT).show();
        }
    }
}
```

Share Taxi	Verzija: 1.0
Tehnička dokumentacija	Datum: 09.07.2018.



Slika 2.1. Izgled početnog zaslona za prijavu, proces prijave i ulazak u aplikaciju

Za potrebe izmjene fragmenata korišten je BottomNavigationView, sama izmjena obavlja se pomoću fragmentManager odnosno getSupportFragmentManager metoda, u ovom slučaju kao sastavni dio loadFragments() metode:

```
private boolean loadFragment(Fragment fragment) {
    //switching fragment
    if (fragment != null) {
        getSupportFragmentManager()
            .beginTransaction()
            .replace(R.id.fragment_container, fragment)
            .commit();
        return true;
    }
    return false;
}
```

Stvaranje nove vožnje odvija se u CreateFragmentu uz opsežnu validaciju potrebnih polja koja se obavlja u vanjskoj pomoćnoj funkciji.

```
private void createNewRide() {
    String from = etFrom.getText().toString();
    String to = etTo.getText().toString();
    String passengers = etPassengers.getText().toString();
    String date = etDate.getText().toString();
    String time = etTime.getText().toString();
    FirebaseUser owner = FirebaseAuth.getInstance().getCurrentUser();

    if(validateInput(from, to, passengers, date, time)){
        DatabaseReference databaseReference =
        FirebaseDatabase.getInstance().getReference();
        try{
            String id = databaseReference.child("Rides").push().getKey();
        }
    }
}
```

Share Taxi	Verzija: 1.0
Tehnička dokumentacija	Datum: 09.07.2018.

```

        Ride ride = new Ride(
            id,from,to,passengers,date,extraValidateTime(time),
            owner.getUid(),owner.getDisplayName(),null
        );
        databaseReference.child("Rides").child(id).setValue(ride);
        returnHome();
        AlertDialog alertDialog = new AlertDialog.Builder(getActivity()).create();
        alertDialog.setTitle("Your ride has been posted!");
        alertDialog.setMessage("Wait for someone to accept your ride.");
        alertDialog.setButton(AlertDialog.BUTTON_POSITIVE, "OK",
            new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int which) {
                    dialog.dismiss();
                }
            });
        alertDialog.show();
        //Toast.makeText(getContext(),"Your ride has been
        posted.",Toast.LENGTH_SHORT).show();
    }
    catch (Exception e){
        Toast.makeText(getContext(),"Data could not be written. Try again later.",
            Toast.LENGTH_SHORT)
            .show();
    }
}
}

```

Ako su sva polja valjana, stvara se novi objekt klase Ride te se isti pohranjuje u bazu s jedinstvenim ključem. Za skladištenje podatak korištena je Firebase Realtime Database koja podatke pohranjuje u JSON formatu. U skladu s tim napisane su i klase objekata koji će se zapisivati u bazu (vožnje, poruke, razgovori i korisnici).

Pretraga vožnji odvija se u SearchFragmentu, koji za prikaz rezultata iste poziva i prikazuje dodatni ResultsFragment.

```

@OnClick(R.id.btSearch)
public void showResults() {
    Bundle bundle = new Bundle();
    bundle.putString("from", etFrom.getText().toString());
    bundle.putString("to", etTo.getText().toString());
    Fragment resultFragment = new ResultsFragment();
    resultFragment.setArguments(bundle);
    getFragmentManager()
        .beginTransaction()
        .replace(R.id.results_container, resultFragment)
        .commit();
    InputMethodManager imm =
    (InputMethodManager) getContext().getSystemService(Context.INPUT_METHOD_SERVICE);
    imm.hideSoftInputFromWindow(etFrom.getWindowToken(), 0);
    imm.hideSoftInputFromWindow(etTo.getWindowToken(), 0);
}

```

Share Taxi	Verzija: 1.0
Tehnička dokumentacija	Datum: 09.07.2018.

Argumenti se predaju kao Bundle objekt u setArguments metodi fragmenta; ResultsFragment ih dohvaća i na temelju njih filtrira podatke iz baze.

```
private void getResults() {
    final FirebaseUser user = FirebaseAuth.getInstance().getCurrentUser();
    DatabaseReference.child("Rides")
        .addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                matchingRides = new ArrayList<>();
                String from = getArguments().getString("from");
                String to = getArguments().getString("to");
                Ride ride;
                for(DataSnapshot rideSnap : dataSnapshot.getChildren()) {
                    ride = rideSnap.getValue(Ride.class);
                    if(ride.getFrom().toLowerCase().contains(from.toLowerCase())
                        && ride.getTo().toLowerCase().contains(to.toLowerCase())
                        && !ride.getOwnerId().equals(user.getUid())
                        && Integer.parseInt(ride.getPassengers())<8
                        && !ride.getPassengerList().contains(user.getUid())) {
                        matchingRides.add(rideSnap.getValue(Ride.class));
                    }
                }

                MyRidesAdapter rvMatchingRidesAdapter = new MyRidesAdapter(getContext(),
                matchingRides);
                changeItemLayout(rvMatchingRidesAdapter);
                rvMatchingRides.setAdapter(rvMatchingRidesAdapter);
            }

            @Override
            public void onCancelled(@NonNull DatabaseError databaseError) {
                Log.e("FETCH DATA ERROR", "Error populating RecyclerView");
            }
        });
}
```

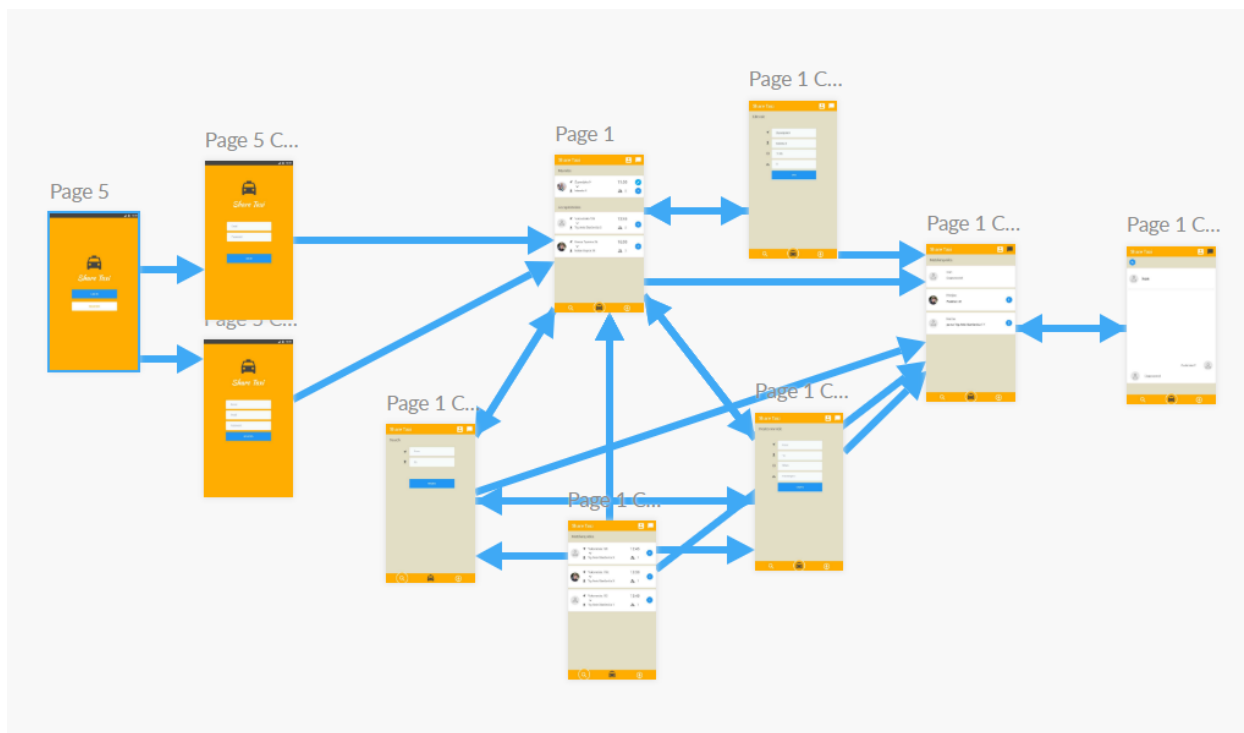
Dohvaćanje iz Firebaseove Realtime baze se obavlja uz pomoć DatabaseReference objekta pri čemu se nad odgovarajućim čvorom postavlja ValueEventListener koji mora „overridati“ metode onDataChange te onCancelled.

U svim slučajevim gdje se obavlja prikaz podataka o vožnjama (HomeFragment My Rides, HomeFragment Accept Rides, ResultsFragment) korišten je MyRidesAdapter – napisana klasa adapter koja naslijeđuje i proširuje klasu adaptera za RecyclerView.

2.4 Mock-up

Mock-up je izrađen koristeći Fluid UI, alat koji služi za dizajn mobilnog sučelja. Na slici 2.5. prikazan je početni mock-up aplikacije.

Share Taxi	Verzija: 1.0
Tehnička dokumentacija	Datum: 09.07.2018.

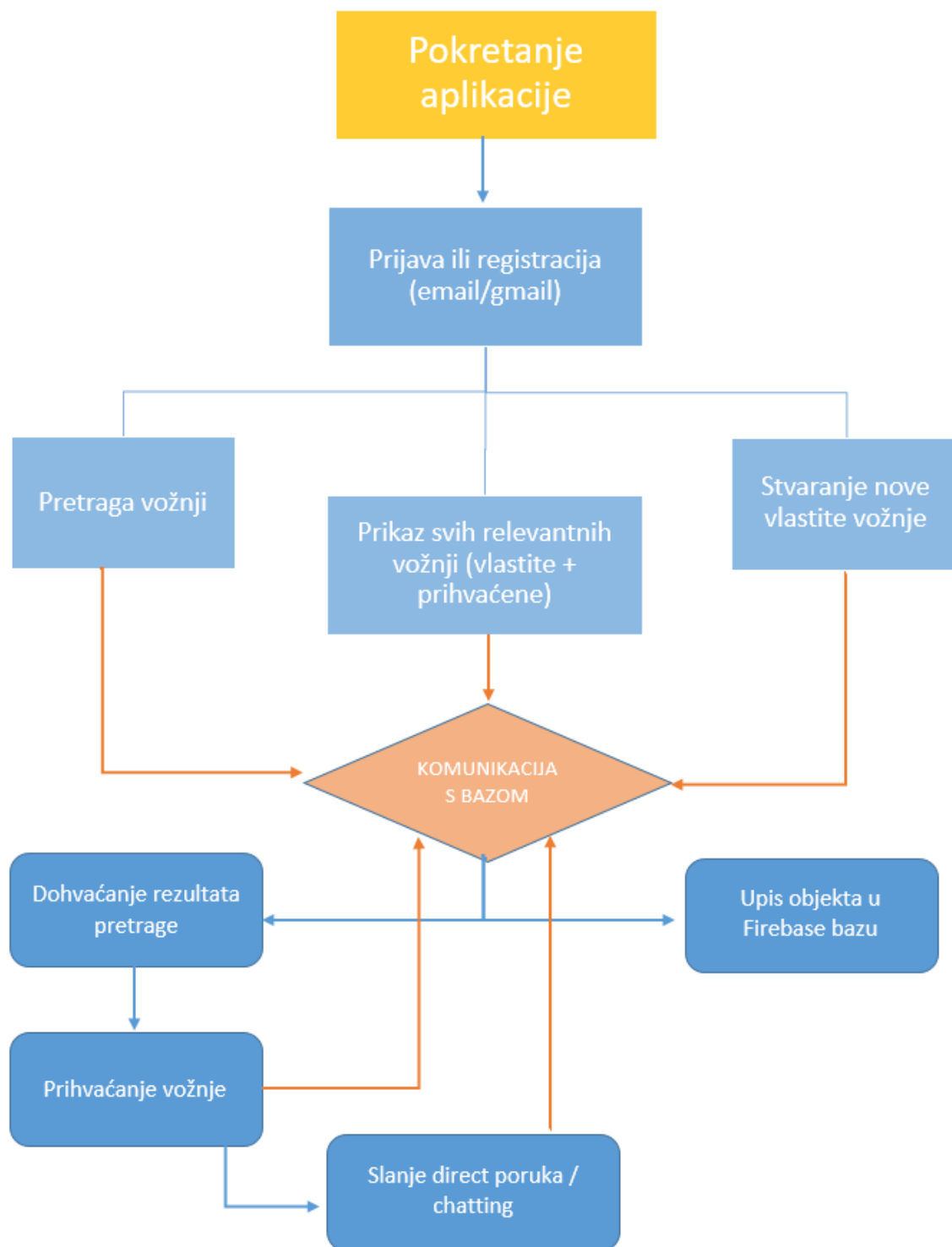


Slika 2.5. Mock-up aplikacije

Prema priloženom mock-upu su radena sučelja i dizajn samo aplikacije.

Share Taxi	Verzija: 1.0
Tehnička dokumentacija	Datum: 09.07.2018.

3. MODEL PODATAKA



Slika 3.1. Prikaz dijagrama toka

Share Taxi	Verzija: 1.0
Tehnička dokumentacija	Datum: 09.07.2018.

Na slici 3.1. prikazan je dijagram toka. Nakon pokretanja pojavljuje se zaslon za prijavu/registraciju putem Firebase Authentication-a. Nakon prijave javlja se početni zaslon te korisnik dodaje novu vožnju ili pretražuje postojeće. Za sve je nužna komunikacija s bazom podataka u stvarnom vremenu što Firebase Realtime database i omogućava. Model podataka od kojih je sastavljena prikazan je na slici 3.2.



Slika 3.2. Prikaz modela podataka

Slika 3.2 prikazuje model aplikacije pomoću kojeg je realizirana većina funkcionalnosti aplikacije. Glavna klasa modela je klasa Ride koja predstavlja vožnju za koju će se zvati taksi i koji treba dijeliti. Sadrži sve potrebne parametre vožnje, kao i informacije o vlasniku i putnicima. Više informacija o korisnicima moguće je pronaći u Users referenci. Chats i Messages zajedno funkcioniraju kako bi omogućili realtime komunikaciju.

Share Taxi	Verzija: 1.0
Tehnička dokumentacija	Datum: 09.07.2018.

4. TEHNIČKE ZNAČAJKE

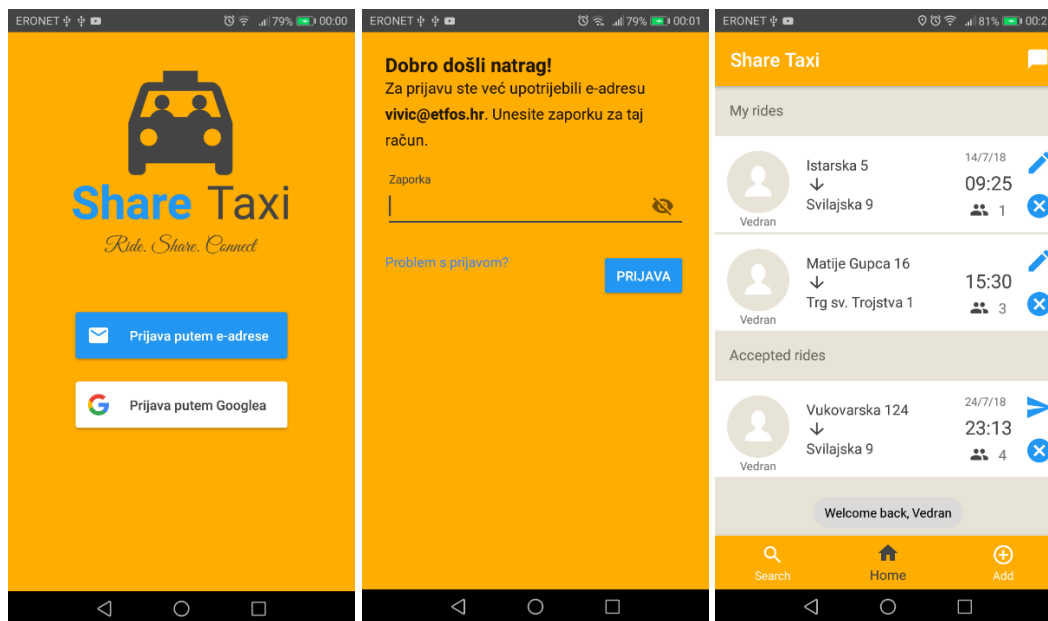
4.1 Korištene tehnologije

Razvijena je Android aplikacija s funkcionalnostima opisanim u prethodnim poglavljima (stvarnovremenska baza podataka, lokacijske usluge određivanja adrese...). Android aplikacija razvijena je u Android Studio razvojnom okruženju, u programskom jeziku Java. Za instalaciju je potrebno upotrijebiti .apk datoteku koja je dana kao prilog. Od mogućnosti Android sustava korišten je Shared Preferences kako bi korisnik mogao pohraniti svoj userID. Također je korišten i pristup lokaciji korisnika kako bi se ista mogla prikazati u obliku adrese polazišta ili odredišta. Za potrebe pisanja, čitanja i bilo kakvih operacija s podacima korištena je Firebase Realtime Database koja se zasniva na principu JSON čvorova. Uz to, korištene su Firebase Cloud funkcije pisane u Javascriptu koje se samostalno izvršavaju na serveru (primjerice funkcija za brisanje isteklih vožnji i sl.).

Share Taxi	Verzija: 1.0
Tehnička dokumentacija	Datum: 09.07.2018.

5. UPUTE ZA KORIŠTENJE

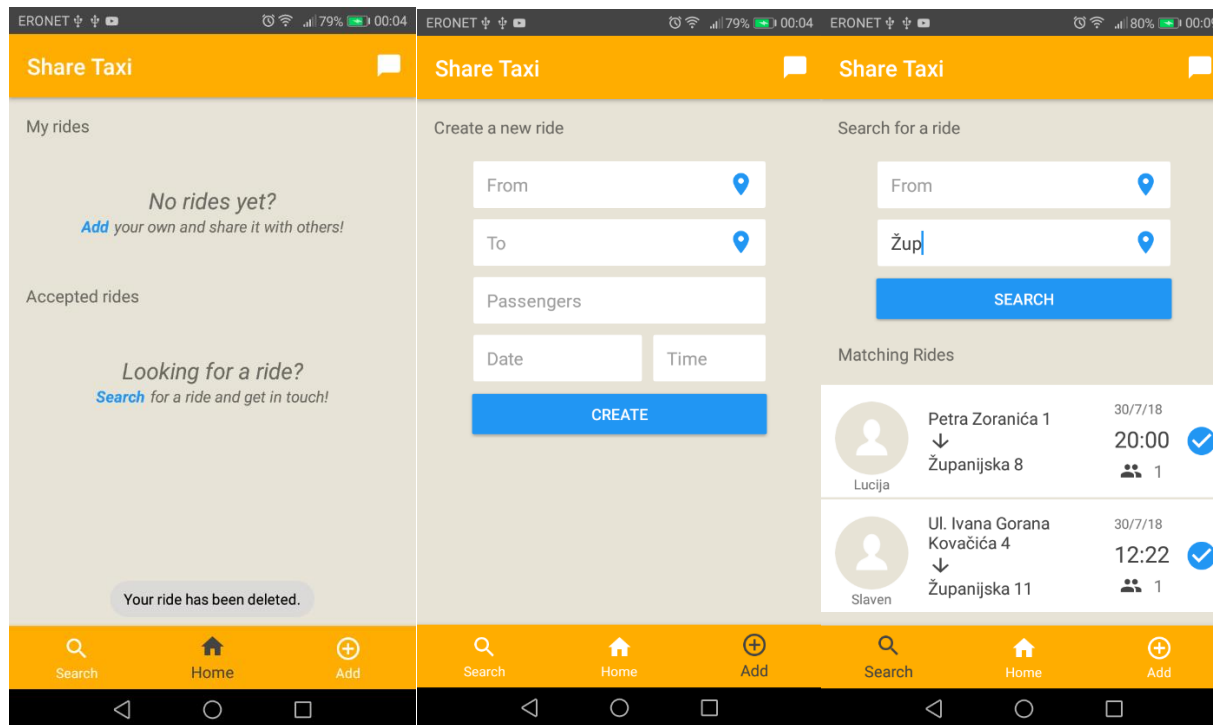
1. Ponajprije je potrebno pokrenuti aplikaciju pritiskom na ikonu aplikacije. Pojavljuje se zaslon za prijavu gdje je potrebno odabrati želi li se korisnik prijaviti uz pomoć emaila i lozinke ili uz Google gmail račun. Po ispravnoj prijavi/registraciji korisnik dolazi do početnog zaslona, gdje su mu prikazane vlastite i prihvaćene vožnje.



Slika 5.1. Izgled početnih zaslona aplikacije

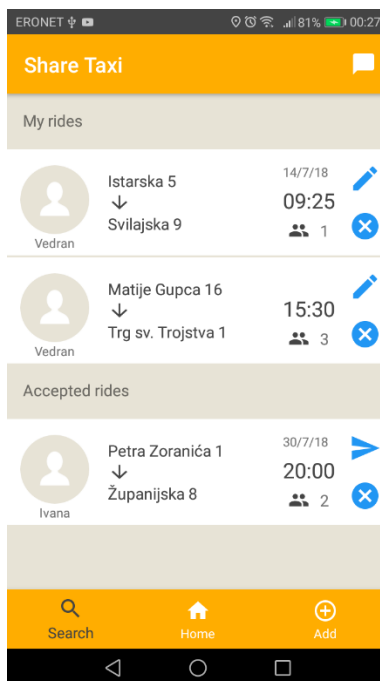
2. U slučaju prvog korištenja obje liste će biti prazne te će se pojaviti dio layouta za prazne liste, koje ujedno sadrže i link za stvaranje odnosno pretrage vožnji. Korisnik bira hoće li stvoriti vlastitu vožnju i čekati da je neko prihvati ili će je tražiti među ostalim, već objavljenim vožnjama. Naravno, korisnik može učiniti i oboje. Pritiskom na ikonu lokacije u kućici odredišta ili polazišta, upisat će se automatski lokacija korisnika u obliku željene adrese.

Share Taxi	Verzija: 1.0
Tehnička dokumentacija	Datum: 09.07.2018.



Slika 5.2. Početne aktivnosti korisnika

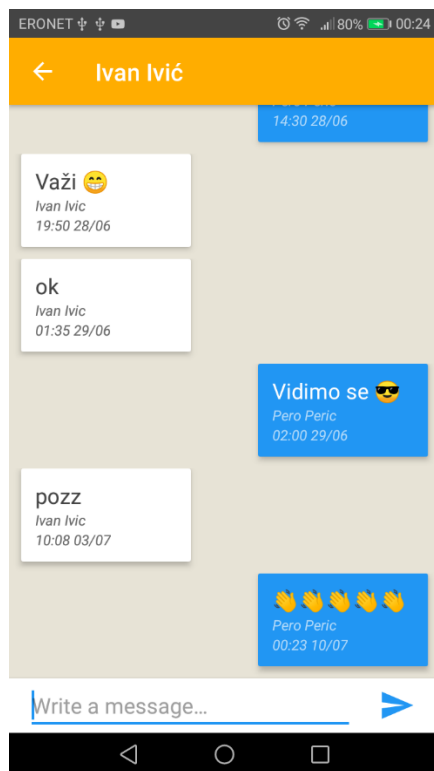
3. Sada korisnik već može uređivati ili brisati svoje vožnje, prihvaćati ili otkazivati vožnje drugih korisnika ili ih pak kontaktirati te u konačnici podijeliti vožnju.



Slika 5.3. Izgled glavnog zaslona s prihvaćenim i vlastitim vožnjama

Share Taxi	Verzija: 1.0
Tehnička dokumentacija	Datum: 09.07.2018.

4. Pri kontaktiranju, otvara se chat s osobom vlasnikom vožnje te se mogu razmjenjivati instant poruke.



Slika 5.4. Chat u aplikaciji

5. Po dogovoru u direktnim porukama, vožnja će se uspješno podijeliti i jedan proces dijeljenja vožnje je završen.

Share Taxi	Verzija: 1.0
Tehnička dokumentacija	Datum: 09.07.2018.

6. LITERATURA

1. Predlošci za laboratorijske vježbe iz kolegija „Razvoj mobilnih aplikacija“
2. <https://stackoverflow.com/>
3. <https://developer.android.com/index.html>
4. <https://firebase.google.com/docs/>
5. <https://codelabs.developers.google.com/>