

Oblikovanje programske potpore

Ak. god. 2017./2018.

Domaći proizvodi za sve

Dokumentacija, Rev. 2

Grupa: *Fangla*

Voditelj: *Ivan Janjić*

Datum predaje: 18. *siječnja* 2017.

Nastavnik: *dr. sc. Davor Kukolja*

Sadržaj

1. Dnevnik promjena dokumentacije.....	3
2. Opis projektnog zadatka.....	5
3. Pojmovnik.....	7
4. Funkcionalni zahtjevi.....	8
5. Ostali zahtjevi.....	35
6. Arhitektura i dizajn sustava.....	36
6.1. Svrha, opći prioriteti i skica sustava.....	36
6.2. Dijagram razreda s opisom.....	39
6.3. Dijagram objekata.....	41
6.4. Ostali UML dijagrami.....	42
7. Implementacija i korisničko sučelje.....	46
7.1. Dijagram razmještaja.....	46
7.2. Korištene tehnologije i alati.....	47
7.3. Isječak programskog koda vezan za temeljnu funkcionalnost sustava.....	48
7.4. Ispitivanje programskog rješenja.....	52
7.5. Upute za instalaciju.....	56
7.6. Korisničke upute.....	58
8. Zaključak i budući rad.....	63
9. Popis literature.....	64
Dodatak A: Indeks (slika, dijagrama, tablica, ispisa kôda).....	65
Dodatak B: Dnevnik sastajanja.....	67
Dodatak C: Prikaz aktivnosti grupe.....	69
Dodatak D: Plan rada / Pregled rada i stanje ostvarenja.....	71

1.Dnevnik promjena dokumentacije

Rev.	Opis promjene/dodatka	Autor(i)	Datum
0.1.0	Napravljen predložak.	Janjić	23.10.2017.
0.2.0	Dodana većina opisa obrazaca uporabe.	Harmicar	28.10.2017.
0.3.0	Dodan pojmovnik i ostali zahtjevi.	Mihaljčuk	29.10.2017.
0.3.1	Uređeni funkcionalni zahtjevi	Markota	30.10.2017.
0.3.2	Napravljen dijagram obrazaca uporabe i modificirani obrasci	Benčić	31.10.2017.
0.3.3	Opis projektnog zadatka	Maslovar	31.10.2017.
0.3.4	Prepravljene gramatičke pogreške	Janjić	02.11.2017.
0.3.5	Ispravljena greška na slici 4.2.	Benčić	02.11.2017.
0.3.6	Uređivanje obrazaca uporabe	Harmicar	09.11.2017.
0.3.7	Dodana većina sekvencijskih dijagrama i jedan obazac uborabe	Benčić	14.11.2017.
0.4.0	Dodani ostali sekvencijski dijagrami	Mihaljčuk	15.11.2017.
0.5	Skica sustava	Janjić	16.11.2017.
0.5.5	Dovršen odjeljak "Svrha, opći prioriteti i skica sustava"	Janjić	16.11.2017.
0.5.6	Dodani UML dijagrami razreda	Markota	17.11.2017.
0.5.7	UML dijagrami prebačeni na pripadajuće mjesto	Janjić	17.11.2017.
0.6	Dodan opis UML dijagrama razreda	Mihaljčuk	17.11.2017.
0.7	Popunjeni dodaci I uređen dokument	Janjić	17.11.2017.
0.9	Dodan dijagram objekata	Janjić	17.11.2017.
0.9.1	Popravljen dijagram objekata	Markota	17.11.2017.
1.0	Konačna verzija dokumentacije prve revizije	Janjić	17.11.2017.
1.1.0	Dodane korištene tehnologije	Markota	15.01.2018.
1.1.5	Prepravljena dokumentacija po naputcima i napravljena 3 dijagrama	Mihaljčuk	16.01.2018.
1.1.6	Ispravljen pojmovnik i korištene tehnologije	Markota	17.01.2018.
1.2.0	Dodan dijagram komponenti	Janjić	17.01.2018.
1.2.1	Dodani sastanci, aktivnosti i zaključak	Janjić	17.01.2018.
1.3.0	Dodan dijagram razmještaja	Janjić	18.01.2018.
1.3.1	Dodan isječak programskog koda	Harmicar	18.01.2018.
1.4.0	Instalacija i konfiguracija baze podataka	Markota	18.01.2018.
1.4.1	Promjena slika u tekst	Janjić	18.01.2018.
1.4.2	Dodane korisničke upute	Mihaljčuk	18.01.2018.
1.4.3	Dodane upute za instalaciju	Harmicar	18.01.2018.
1.4.4	Dopunjene korisničke upute	Mihaljčuk	18.01.2018.
1.5	Označene slike te dodane u indeksu	Janjić	18.01.2018.
2.0	Konačna verzija dokumentacije druge revizije	Janjić	18.01.2018.

2.Opis projektnog zadatka

Industrijska klima prehrambenog sektora moderne hrvatske države favorizira korporativne divove tako da obiteljskim poljoprivrednim gospodarstvima ne preostaje dovoljan udio u tržištu da bi se mogli neovisno razvijati. Upravo iz tog razloga i zbog pomanjkanja infrastrukture koja bi im olakšala izlazak na tržište, OPG-i su primorani surađivati s korporacijama koje im predstavljaju tržišnu konkurenciju. Proteklih nekoliko godina svjedočili smo kolapsu prehrambenih magnata i Hrvatskoj prijeti opasnost od potpunog srozavanja infrastrukture nabavke namirnica. Poljoprivrednim gospodarstvima potreban je alat kojime i ona mogu postati konkurentna na tržištu. Time bi se smanjila opasnost od monopola nad industrijom i njenim padom zbog posrtanja jednog poslovnog entiteta.

Cilj ovog projekta je razviti Android aplikaciju koja bi učinkovito povezala obiteljska poljoprivredna gospodarstva s njihovim kupcima i time zaobišla posrednike (trgovačke lance) i istovremeno ponudila kupcima alternativu. Aplikacija omogućuje OPG-ima da bez poteškoća plasiraju svoje proizvode na tržište postavljanjem oglasa na aplikaciju i istovremeno kupcima zainteresiranim za domaće proizvode da ih brzo pronađu.

Registrirati se može svatko. Pri registraciji naznačuje prijavljuje li se kao kupac, OPG ili prijevoznik (moguće su i kombinacije), nakon čega se zahtjev za registraciju šalje administratoru na odobrenje. Svaki korisnik mora upisati svoje osobne podatke: ime, prezime, e-mail adresu, broj telefona i lozinku svog korisničkog računa. Prilikom registracije OPG-a, potrebno je unijeti i naziv, adresu i jedinstveni matični identifikacijski broj poljoprivrednog gospodarstva, dok je prilikom registracije prijevoznika potrebno navesti i podatke o vozilu poput modela vozila, registracije i kategorije. Aplikacija ima četiri sučelja: za kupce, OPG-e, prijevoznike i administratore.

Sučelje za kupce se sastoji od tražilice proizvoda i popisa oglasa za iste. Kupac odabirom oglasa stupa u kontakt s OPG-om koji ga je postavio te se dogovara za pojedinih stvari poput cijene, količine i datuma preuzimanja. Kada se postigne dogovor, transakcija se zaključava, OPG ažurira oglas i ukoliko kupac nema prijevoz, stvara se zahtjev za prijevoz koji je vidljiv prijevoznicima.

Sučelje prijevoznika omogućava korisniku pretragu zahtjeva za prijevozom po mjestu polazišta i odredišta. Analogno oglasu OPG-a, prijevoznik odabirom kupčeva zahtjeva za

prijevoz stupa u kontakt s kupcem te s njime dogovara prijevoz. Postizanjem dogovora zahtjev za prijevoz se briše s liste čekanja i on više nije vidljiv ostalim prijevoznicima.

Sučelje OPG-a omogućuje upravljanje vlastitim oglasima i pregled upita za oglase.

Svaki korisnik ima stranicu svog korisničkog računa vidljivu svima u sustavu. Ukoliko su dva korisnika sklapala transakciju preko aplikacije, tada mogu jednom ocijeniti jedan drugoga i ostaviti komentar. Time se osigurava da drugi korisnici mogu prepoznati neodgovorne ili nepoželjne korisnike prije stupanja u interakciju s njima. Korisnici mogu prijaviti druge korisnike, njihove komentare i oglase ako naiđu na neprimjeren sadržaj, sumnjaju u nepostojanje osobe iza korisničkog računa ili su imali neugodna iskustva s korisnikom.

Sučelje administratora omogućuje pregled i upravljanje registracijama i prijavama. Administrator može prihvatiti i odbiti registraciju osobe, ukloniti osobu iz sustava i ukloniti oglase i komentare koje smatra neprimjerenima.

3.Pojmovnik

Baza podataka – zbirka zapisa pohranjenih u računalu na sustavan način, tako da joj računalni program može pristupiti prilikom odgovaranja na zadani problem

HTML – (HyperText Markup Language) prezentacijski jezik za izradu web stranica, omogućuje jednak prikaz stranice bez obzira na korišteni web preglednik, računalu ili operativni sustav

HTTP – (HyperText Transfer Protocol) glavna i najčešća metoda prijenosa informacija na Internetu, osnovna namjena mu je omogućavanje objavljivanja i prezentacije HTML dokumenata

Java – objektno orijentirani programski jezik

Poslužitelj – računalu ili softver koji šalje ili prima podatke od klijenata

DBMS – (DataBase-Management System) računalni program korišten za upravljanje i ispitivanje baze podataka

SQL – (Structured Query Language) najpopularniji računalni jezik za izradu, traženje, ažuriranje i brisanje podataka iz relacijskih baza podataka

UML – (Unified Modeling Language) standardizirani jezik za modeliranje u području programskog inženjerstva koji se koristi za prikazivanje, specificiranje i dokumentiranje sustava programske potpore

4.Funkcionalni zahtjevi

Dionici:

- Kupci
- Prijevoznici
- Obiteljska poljoprivredna gospodarstva (OPG)
- Administrator

Aktori i njihovi funkcionalni zahtjevi:

- Kupac, inicijator
 - Svaki registriran korisnik na aplikaciji
 - Pregledava objavljene oglase koje postavljaju OPG-i
 - Može stupiti u kontakt s prijevoznicima i OPG-ima
 - Dodavanje oglasa za traženje usluge prijevoza robe OPG-a
 - Može urediti ili obrisati vlastiti profil
 - Ima mogućnost uređivanja i brisanja vlastitih objava
- Prijevoznik, inicijator
 - Registrirani korisnik s mogućnošću prihvatanja zahtjeva za prijevoz robe
 - Pregledava objave kupaca za prijevoz robe
 - Može urediti ili obrisati vlastiti profil
- Obiteljsko poljoprivredno gospodarstvo, inicijator
 - Registrirani korisnik sa mogućnošću stavljanja oglasa na stranicu
 - Može urediti ili obrisati vlastiti profil
 - Ima mogućnost uređivanja i brisanja vlastitih objava
 - Može obavljati funkciju prijevoznika
- Administrator, inicijator
 - Korisnik koji je u sustavu od početka (super user)

- Ima ovlasti brisanja bilo koje objave, te brisanja korisničkih profila
- Potvrđuje nove zahtjeve za kategorije proizvoda
- Poslužitelj, sudionik
 - Pohranjuje podatke o svim registriranim korisnicima u bazu podataka (BP)
 - Pohranjuje sve trenutno aktivne objave u BP
 - Pohranjuje sve poruke u chatu između korisnika u BP
- Baza podataka, sudionik
 - Pohranjuje podatke o svim registriranim korisnicima
 - Pohranjuje sve trenutno aktivne objave
 - Pohranjuje sve poruke u chatu između korisnika

Opis obrazaca uporabe:

- UC1 - Registracija:
 - **Glavni sudionik:** Potencijalni korisnik.
 - **Cilj:** Izrada novog korisničkog profila.
 - **Sudionici:** Administrator, baza podataka.
 - **Rezultat:** Korisnik je registriran, te se u bazu sprema novi korisnički profil.
 - **Željeni scenarij:**
 - ◆ Potencijalni korisnik odabire akciju registracije.
 - ◆ Nakon unosa svih potrebnih podataka u obrazac, potvrđuje svoj unos.
 - ◆ Administrator obavlja provjeru unesenih podataka, te provjerava postojanje istog korisnika.
 - ◆ Ako korisnik ne postoji u bazi, administrator odobrava registraciju novog korisnika u bazu podataka, te šalje korisniku na email adresu potvrdu o registraciji.
 - **Mogući drugi scenarij:**

- ◆ Administrator odbije zahtjev za registracijom korisnika jer su uneseni podaci neispravni, ili korisnik već postoji u sustavu.

Aplikacija obavještava korisnika o neuspjeloj registraciji putem emaila.

- UC2 – Prijava:

- **Glavni sudionik:** Registrirani korisnik.
- **Cilj:** Prijava registriranog korisnika.
- **Sudionici:** Korisnik, baza podataka.
- **Rezultat:** Registrirani korisnik se uspješno prijavljuje u sustav, dok je prijava neregistriranog korisnika odbijena.
- **Željeni scenarij:**
 - ◆ Korisnik se prijavljuje u sustav.
 - ◆ Sustav provjerava korisnika i utvrđuje da je registriran.
 - ◆ Korisniku dopušten pristup sustavu u skladu s njegovim ovlastima.
- **Mogući drugi scenarij:**
 - ◆ Korisnik unosi podatke za prijavu u sustav.
 - ◆ Sustav provjerava korisnikove podatke i utvrđuje da ne postoje u bazi.
 - ◆ Korisniku nije omogućen pristup i obavještava ga o neuspješnosti.

- UC3 - Odjava:

- **Glavni sudionik:** Registrirani korisnik.
- **Cilj:** Odjava korisnika.
- **Sudionici:** Korisnik, baza podataka.
- **Rezultat:** Korisnik je uspješno odjavljen iz aplikacije.
- **Željeni scenarij:**
 - ◆ Prekidanje veze sa serverom.
 - ◆ Odjava korisnika iz sustava.

- UC4 - Uređivanje Podataka Na Profilu:

- **Glavni sudionik:** Registrirani korisnik.
- **Cilj:** Promijeniti osobne podatke.
- **Sudionici:** Poslužitelj, baza podataka.
- **Preduvjet:** Korisnik je prijavljen u aplikaciju.
- **Rezultat:** Spremanje promjena na korisničkom profilu
- **Željeni scenarij:**
 - ◆ Korisnik odabere opciju uredi profil.
 - ◆ Korisnik promjeni neke od informacija koje je unio kod registracije.
 - ◆ Korisnik potvrdi unesene promjene, te se one spremaju u bazu podataka.

- UC5 - Obriši Profil:

- **Glavni sudionik:** Registrirani korisnik.
- **Cilj:** Obrisati korisnički profil.
- **Sudionici:** Poslužitelj, baza podataka.
- **Preduvjet:** Korisnik je prijavljen u aplikaciji.
- **Rezultat:** Obrisani korisnički profil.
- **Željeni scenarij:**
 - ◆ Korisnik na profilu odabire opciju obriši profil.
 - ◆ Poslužitelj šalje u bazu podataka zahtjev za brisanje profila.
 - ◆ Korisnički profil je obrisani, ali podatci ostaju u bazi.

- UC6 - Stvaranje Objave OPG:

- **Glavni sudionik:** Obiteljsko poljoprivredno gospodarstvo.
- **Cilj:** Stvaranje nove objave za ponudu robe.
- **Sudionici:** Poslužitelj, baza podataka.
- **Preduvjet:** Korisnik je prijavljen u aplikaciji, ispunjen obrazac za objavu sadržaja.

- **Rezultat:** Stvorena nova objava.
 - **Željeni scenarij:**
 - ◆ OPG bira opciju stvori objavu.
 - ◆ Otvara se obrazac za koji traži podatke za objavu.
 - ◆ OPG potvrđuje svoj unos.
 - ◆ U bazu podataka se pohranjuje nova objava.
 - ◆ U aplikaciji je nova objava vidljiva kupcima.
 - **Mogući drugi scenarij:**
 - ◆ Korisnik nije unio sve potrebne podatke za objavu.
 - ◆ Sustav ga obavještava da treba popuniti sva zadana polja.
- UC7 - StvaranjeObjaveKupac:
- **Glavni sudionik:** Kupac.
 - **Cilj:** Stvaranje nove objave za potražnju prijevoza.
 - **Sudionici:** Poslužitelj, baza podataka.
 - **Preduvjet:** Korisnik je prijavljen u aplikaciji, ispunjen obrazac za objavu sadržaja.
 - **Rezultat:** Stvorena nova objava.
 - **Željeni scenarij:**
 - ◆ Kupac bira opciju stvori objavu za traženje prijevoza.
 - ◆ Otvara se obrazac za koji traži podatke za objavu.
 - ◆ U bazu podataka se pohranjuje nova objava.
 - ◆ U aplikaciji je nova objava vidljiva prijevoznicima.
 - **Mogući drugi scenarij:**
 - ◆ Korisnik nije unio sve potrebne podatke za objavu.
 - ◆ Sustav ga obavještava da treba popuniti sva zadana polja.

- UC8 - BrisanjeObjaveOPG

- **Glavni sudionik:** Obiteljsko poljoprivredno gospodarstvo.
- **Cilj:** Brisanje već postavljene objave OPG-a.
- **Sudionici:** Poslužitelj, baza podataka.
- **Preduvjet:** Korisnik je prijavljen u aplikaciji, vlastita objava koju treba obrisati je aktivna.
- **Rezultat:** Objava je obrisana.
- **Željeni scenarij:**
 - ◆ OPG odabere opciju obriši objavu.
 - ◆ Objava se izbriše iz baze podataka.
- **Mogući drugi scenarij:**
 - ◆ OPG nema ovlasti obrisati objavu.
 - ◆ Ispisuje se greška o nemogućnosti brisanja objave.

- UC9 - BrisanjeObjaveKupac

- **Glavni sudionik:** Kupac.
- **Cilj:** Brisanje već postavljene objave kupca.
- **Sudionici:** Poslužitelj, baza podataka.
- **Preduvjet:** Korisnik je prijavljen u aplikaciji, vlastita objava koju treba obrisati je aktivna.
- **Rezultat:** Objava je obrisana.
- **Željeni scenarij:**
 - ◆ Kupac odabere opciju obriši objavu.
 - ◆ Objava se izbriše iz baze podataka.
- **Mogući drugi scenarij:**
 - ◆ Kupac nema ovlasti obrisati objavu.
 - ◆ Ispisuje se greška o nemogućnosti brisanja objave.

- UC10 - UrediObjavuOPG

- **Glavni sudionik:** Obiteljsko poljoprivredno gospodarstvo.
- **Cilj:** Uređivanje već postavljene objave OPG-a.
- **Sudionici:** Poslužitelj, baza podataka.
- **Preduvjet:** Korisnik je prijavljen u aplikaciji, vlastita objava koju treba urediti je aktivna.
- **Rezultat:** Objava je uređena.
- **Željeni scenarij:**
 - ◆ OPG odabere opciju uredi objavu.
 - ◆ Objava se uređuje te se potvrđuje završetak uređivanja.
 - ◆ Izmjena objave se sprema u bazu podataka.
- **Mogući drugi scenarij:**
 - ◆ OPG nema ovlasti urediti objavu.
 - ◆ Objava je u međuvremenu obrisana.
 - ◆ Ispisuje se greška o nemogućnosti uređivanja objave.

- UC11 - UrediObjavuKupac

- **Glavni sudionik:** Kupac.
- **Cilj:** Uređivanje već postavljene objave kupca.
- **Sudionici:** Poslužitelj, baza podataka.
- **Preduvjet:** Korisnik je prijavljen u aplikaciji, vlastita objava koju treba obrisati je aktivna.
- **Rezultat:** Objava je uređena.
- **Željeni scenarij:**
 - ◆ Kupac odabere opciju uredi objavu.
 - ◆ Objava se uređuje te se potvrđuje završetak uređivanja.
 - ◆ Izmjena objave se sprema u bazu podataka.
- **Mogući drugi scenarij:**

- ◆ Kupac nema ovlasti urediti objavu.
 - ◆ Objava je u međuvremenu obrisana.
 - ◆ Ispisuje se greška o nemogućnosti uređivanja objave.
-
- UC12 - PretražiProizvode
 - **Glavni sudionik:** Registrirani korisnik.
 - **Cilj:** Pronalazak željenog proizvoda.
 - **Sudionici:** Poslužitelj, baza podataka.
 - **Preduvjet:** Korisnik je prijavljen na sustav.
 - **Rezultat:** Informacija o postojanu proizvoda.
 - **Željeni scenarij:**
 - ◆ Korisnik pronalazi željeni proizvod.
-
- UC13 - PretražiKorisnike
 - **Glavni sudionik:** Registrirani korisnik.
 - **Cilj:** Pronalazak željenog korisnika.
 - **Sudionici:** Poslužitelj, baza podataka.
 - **Preduvjet:** Korisnik je prijavljen na sustav.
 - **Rezultat:** Pronalazak željenog korisnika.
 - **Željeni scenarij:**
 - ◆ Prikaz svih korisnika sa zadanim imenom
-
- UC14 - OcijeniKorisnika
 - **Glavni sudionik:** Registrirani korisnik.
 - **Cilj:** Davanje ocjene drugom korisniku.
 - **Sudionici:** Poslužitelj, baza podataka.
 - **Preduvjet:** Korisnik je prijavljen na sustav i ima ovlast ocijeniti drugog korisnika koju dobiva nakon obavljanja transakcije.

- **Rezultat:** Povećana ili smanjena popularnost korisnika.
- **Željeni scenarij:**
 - ◆ Korisnik na profilu drugog korisnika unese ocjenu kojom ocjenjuje iskustvo međusobnog poslovanja.
 - ◆ Promijenjeni rezultat se pohranjuje u bazu podataka.

- UC15 - PrijaviKorisnika

- **Glavni sudionik:** Registrirani korisnik.
- **Cilj:** Prijava korisnika administratoru.
- **Sudionici:** Poslužitelj, baza podataka.
- **Preduvjet:** Korisnik je prijavljen na sustav.
- **Željeni scenarij:**
 - ◆ Administrator prima obavjest o mogućem neprimjerenom ponašanju korisnika.

- UC16 - OstaviKomentarKorisniku

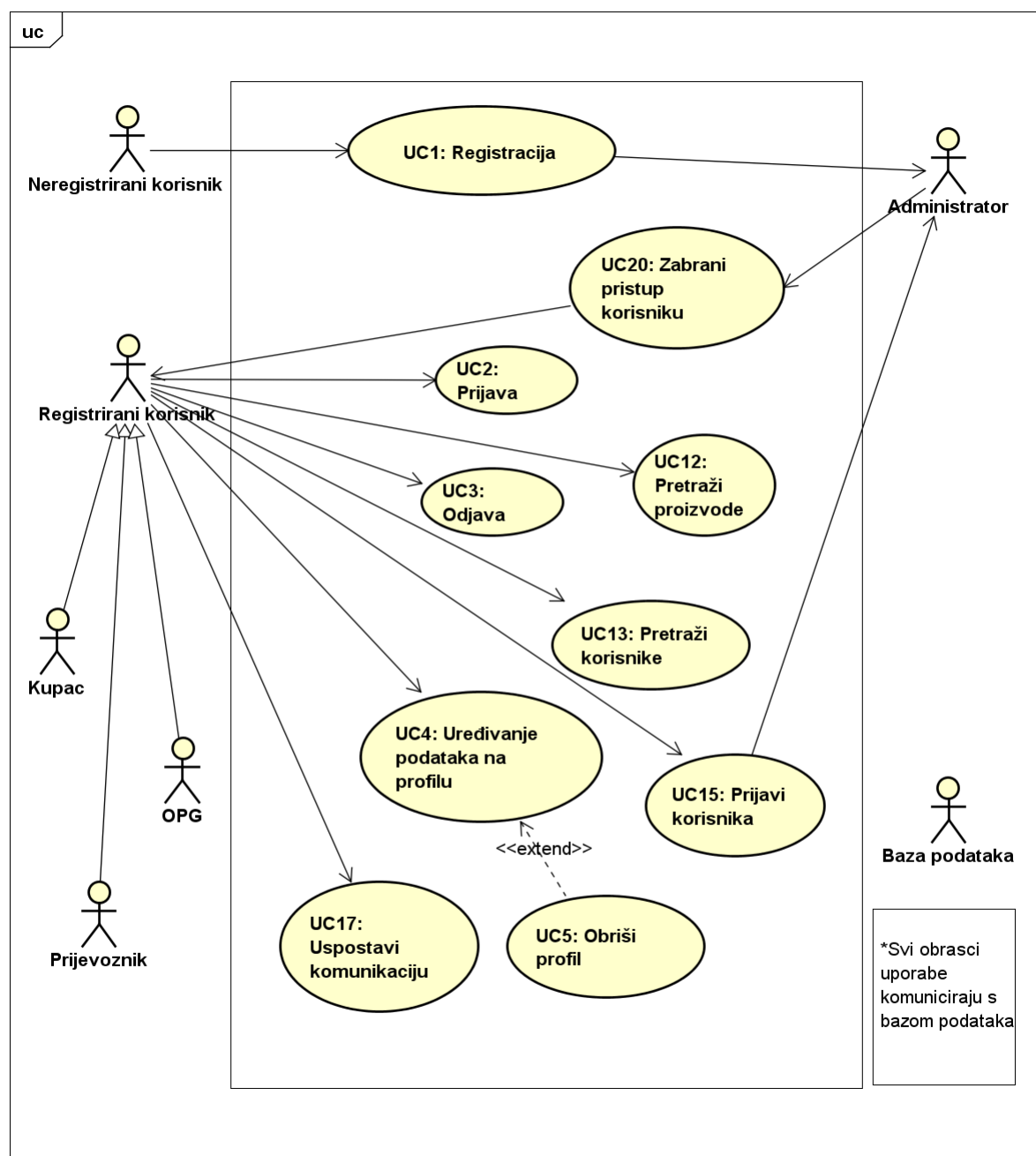
- **Glavni sudionik:** Registrirani korisnik.
- **Cilj:** Ostavljanje javnog komentara drugom korisniku.
- **Sudionici:** Poslužitelj, baza podataka.
- **Preduvjet:** Korisnik je prijavljen na sustav i obavio je transakciju s odabranim korisnikom.
- **Rezultat:** Javni komentar na korisničkom profilu.
- **Željeni scenarij:**
 - ◆ Korisnik na profilu drugog korisnika unese javni komentar kojim opisuje međusobno poslovanje.
 - ◆ Upisani komentar se upisuje u bazu podataka i vidljiv je svim drugim korisnicima.

- UC17 - Uspostavi komunikaciju
 - **Glavni sudionik:** Registrirani korisnik.
 - **Cilj:** Uspostava komunikacije sa drugim korisnikom.
 - **Sudionici:** Poslužitelj, baza podataka.
 - **Preduvjet:** Korisnik je prijavljen na sustav
 - **Rezultat:** Prozor za razgovor između korisnika.
 - **Željeni scenarij:**
 - ◆ Nakon pritiska na gumb, otvara se prozor za komunikaciju sa željenim korisnikom.

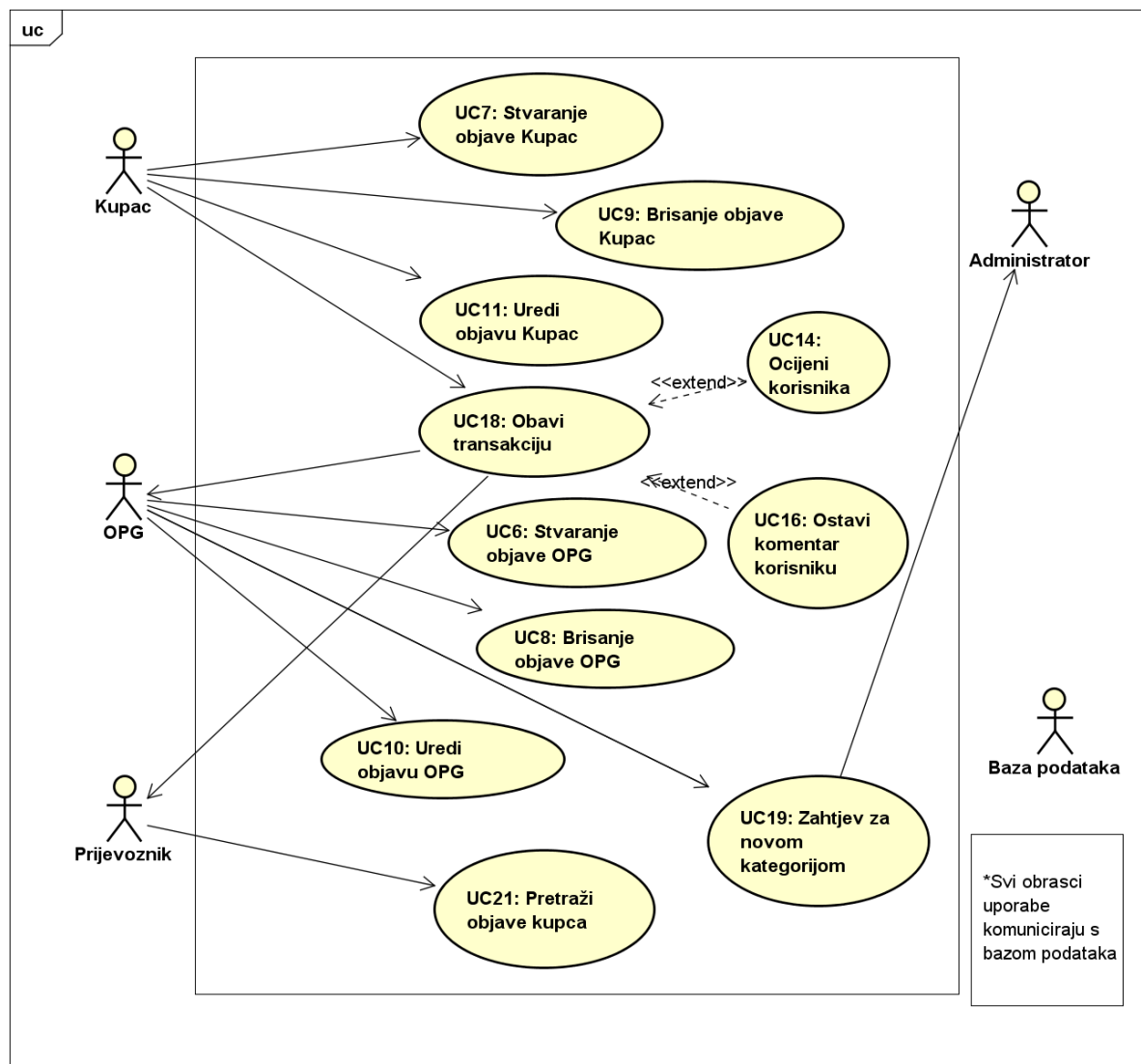
- UC18 - ObaviTransakciju
 - **Glavni sudionik:** Registrirani korisnici.
 - **Cilj:** Obaviti prodaju proizvoda i prijevoz.
 - **Sudionici:** Poslužitelj, baza podataka.
 - **Preduvjet:** Korisnici su prijavljeni na sustav, OPG je postavio objavu i nitko nije napravio transakciju prije ove.
 - **Rezultat:** Dogovorena je prodaja robe stavljene u objavi i dogovoren je prijevoz iste robe ili do prodaje nije došlo.
 - **Željeni scenarij:**
 - ◆ Kupac uočava objavu i šalje zahtjev za kupnju robe, te u tom zahtjevu specificira da ima prijevoz.
 - ◆ OPG prihvaća zahtjev.
 - ◆ Transakcija obavljena.
 - **Mogući drugi scenarij:**
 - ◆ Kupac uočava objavu i šalje zahtjev za kupnju robe, te u tom zahtjevu specificira da nema prijevoz.
 - ◆ OPG prihvaća zahtjev.
 - ◆ Kupac radi objavu transakcije u kojoj traži prijevoz.
 - **Treći mogući scenarij:**
 - ◆ Kupac uočava objavu i šalje zahtjev za kupnju robe.

- ◆ OPG odbija prodati robu.
- UC19 - ZahtjevZaNovomKategorijom
 - **Glavni sudionik:** Obiteljsko poljoprivredno gospodarstvo, Administrator.
 - **Cilj:** Stvaranje nove kategorije proizvoda.
 - **Sudionici:** Poslužitelj, baza podataka.
 - **Preduvjet:** Korisnik je prijavljen na sustav i ima status OPG-a.
 - **Rezultat:** Uspješno prihvaćen zahtjev za novom kategorijom.
 - **Željeni scenarij:**
 - ◆ OPG šalje administratoru zahtjev za novu kategoriju proizvoda.
 - ◆ Administrator odobrava zahtjev.
 - **Mogući drugi scenarij:**
 - ◆ Administrator odbije zahtjev za novom kategorijom.
- UC20 - ZabraniPristupKorisniku
 - **Glavni sudionik:** Administrator.
 - **Cilj:** Privremena/trajna zabrana pristupa.
 - **Sudionici:** Poslužitelj, baza podataka, korisnik.
 - **Preduvjet:** Korisnik ima administratorske ovlasti.
 - **Rezultat:** Registriranom korisniku je zabranjen pristup aplikaciji.
 - **Željeni scenarij:**
 - ◆ Administrator otkrije nedopušteno ponašanje korisnika.
 - ◆ Korisniku zabranjuje pristup aplikaciji.
 - ◆ Zabrana se upisuje u bazu podataka.
- UC21 - PretražiObjaveKupca
 - **Glavni sudionik:** Prijevoznik.
 - **Cilj:** Pronalazak željenih objava za prijevoz.
 - **Sudionici:** Poslužitelj, baza podataka.

- **Preduvjet:** Korisnik je prijavljen na sustav.
- **Rezultat:** Prikaz svih objava sa zadanim polazištima, odredištima ili rutama.
- **Željeni scenarij:**
 - ◆ Prijevoznik pronalazi tražene objave.



Slika 4.1 Dijagram obrazaca uporabe

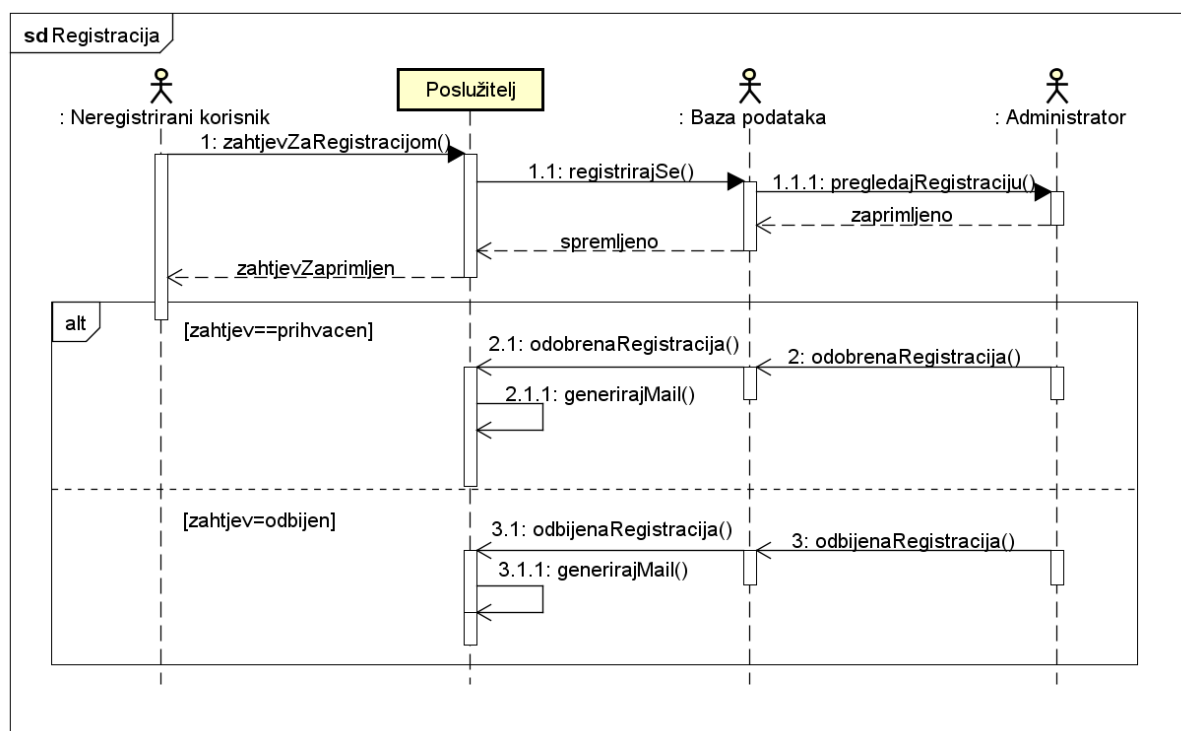


Slika 4.2 Dijagram obrazaca uporabe

Sekvencijski dijagrami

Obrazac uporabe UC1:

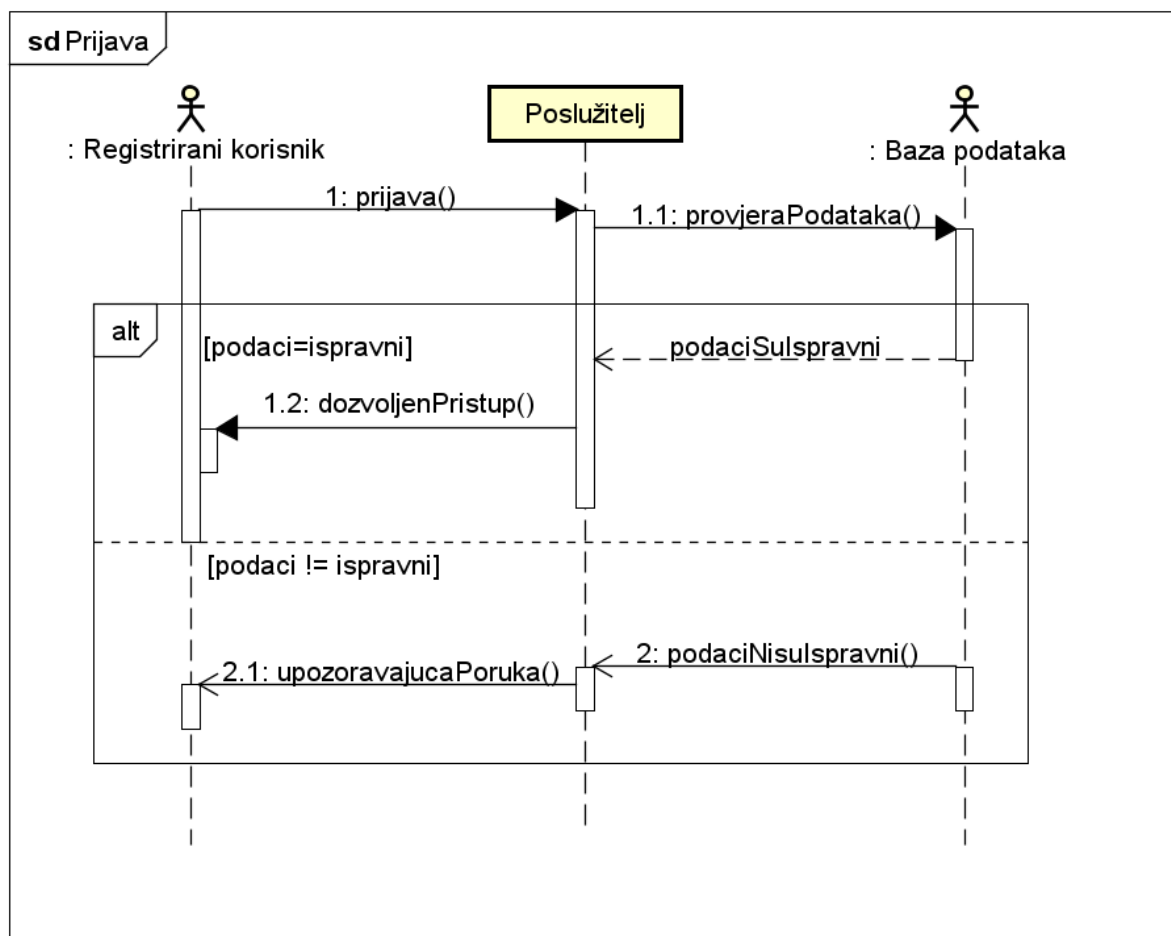
Neregistrirani korisnik preko aplikacije šalje zahtjev za registracijom tako što je ispunio zadani obrazac za registracijom i pritisnuo tipku šalji. Aplikacija administratoru prosljeđuje zahtjev za registracijom koji je prije toga pohranila u bazu podataka. Administrator može odbiti ili potvrditi zahtjev. U oba slučaja se generira mail koji se šalje na adresu koji je korisnik unio prilikom registracije.



Slika 4.3 Sekvencijski dijagram UC1

Obrazac uporabe UC2:

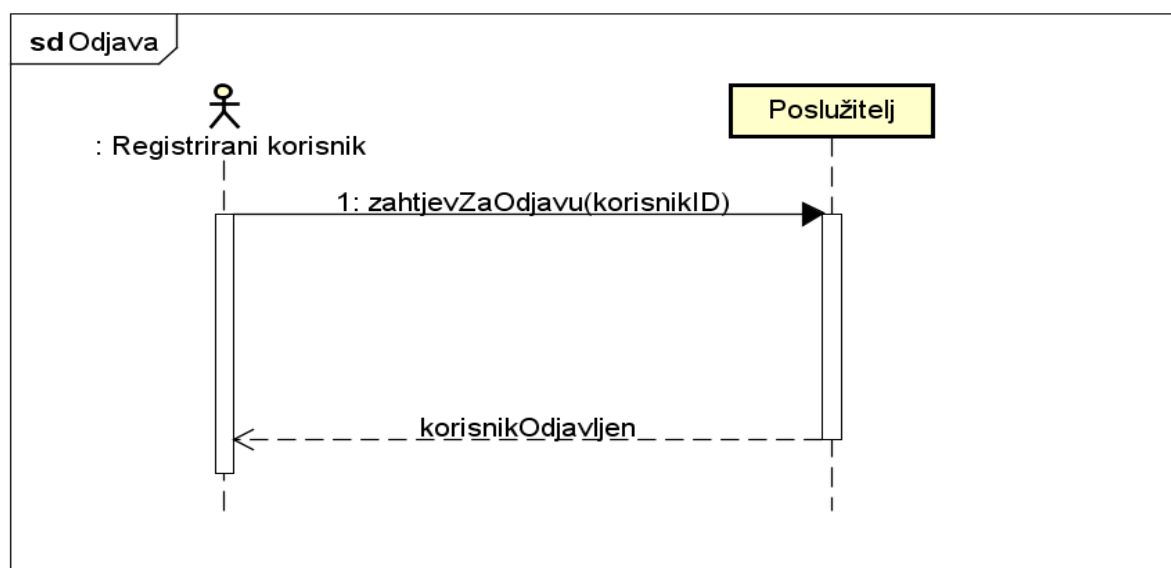
Registrirani korisnik unosi podatke potrebne za prijavu u sustav i šalje ih na obradu aplikaciji. Aplikacija nadalje komunicira s bazom podataka i provjerava ispravnost unesenih podataka. U slučaju ispravno unesenih podataka korisniku se omogućava pristup aplikaciji, a u suprotnom šalje mu se poruka o pogrešnom unosu podataka.



Slika 4.4 Sekvencijski dijagram UC2

Obrazac uporabe UC3:

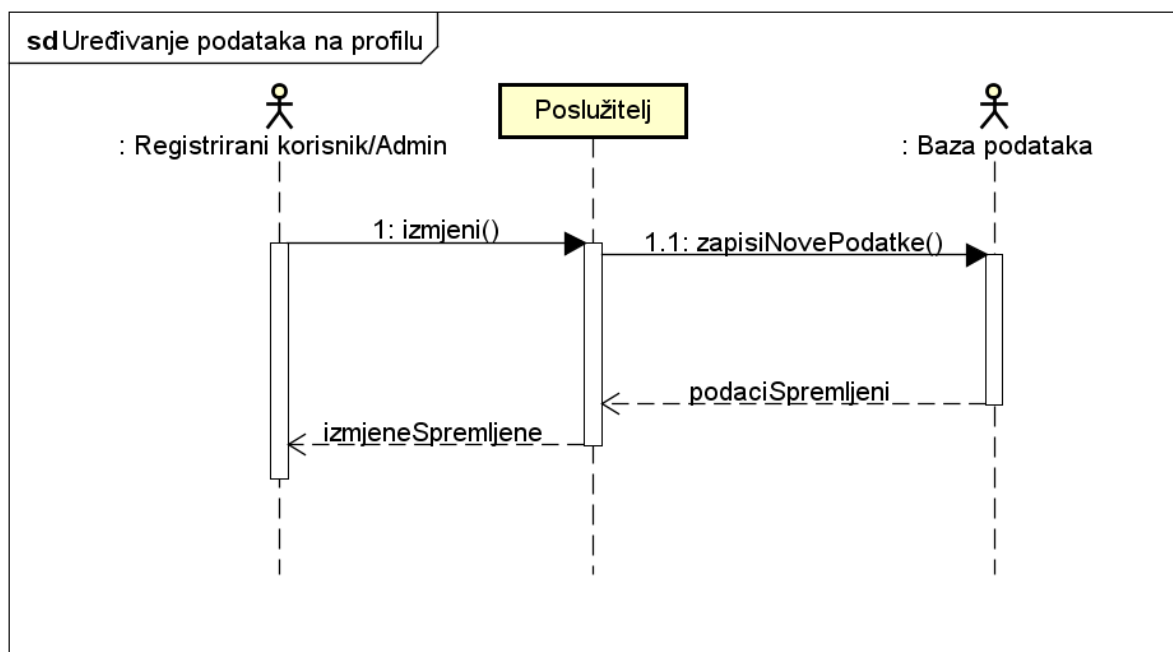
Korisnik šalje aplikacija zahtjev za odjavu, te aplikacija odjavljuje korisnika.



Slika 4.5 Sekvencijski dijagram UC3

Obrazac uporabe UC4:

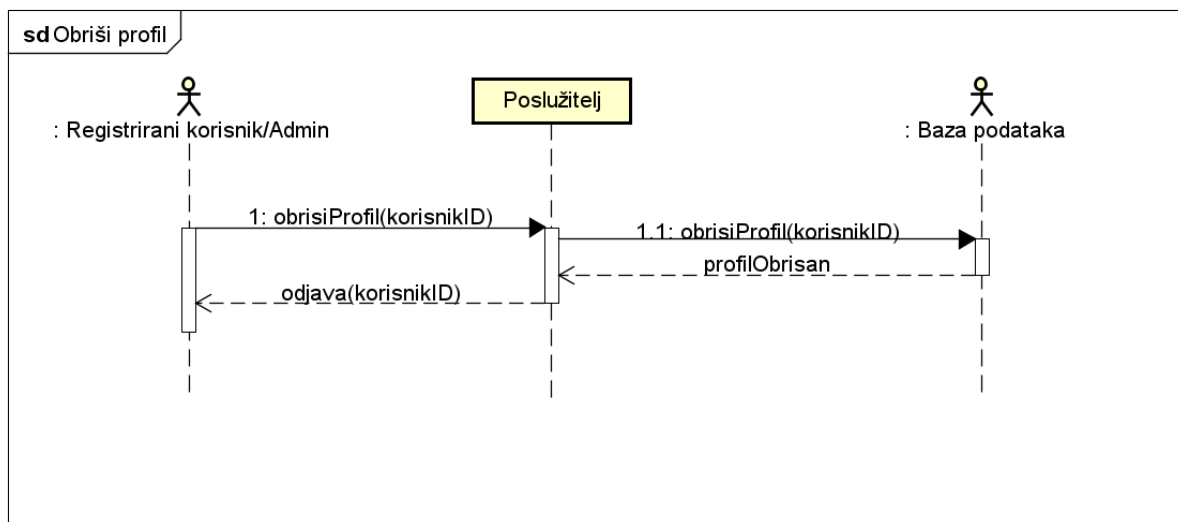
Korisnik u skladu s mogućnostima sučelja unosi promijene na svom profilu. Svaka promjena se šalje bazi podataka koja ju pohranjuje i javlja aplikaciji da je promjena spremljena. Aplikacija tada prosljeđuje korisniku tu poruku.



Slika 4.6 Sekvencijski dijagram UC4

Obrazac uporabe UC5:

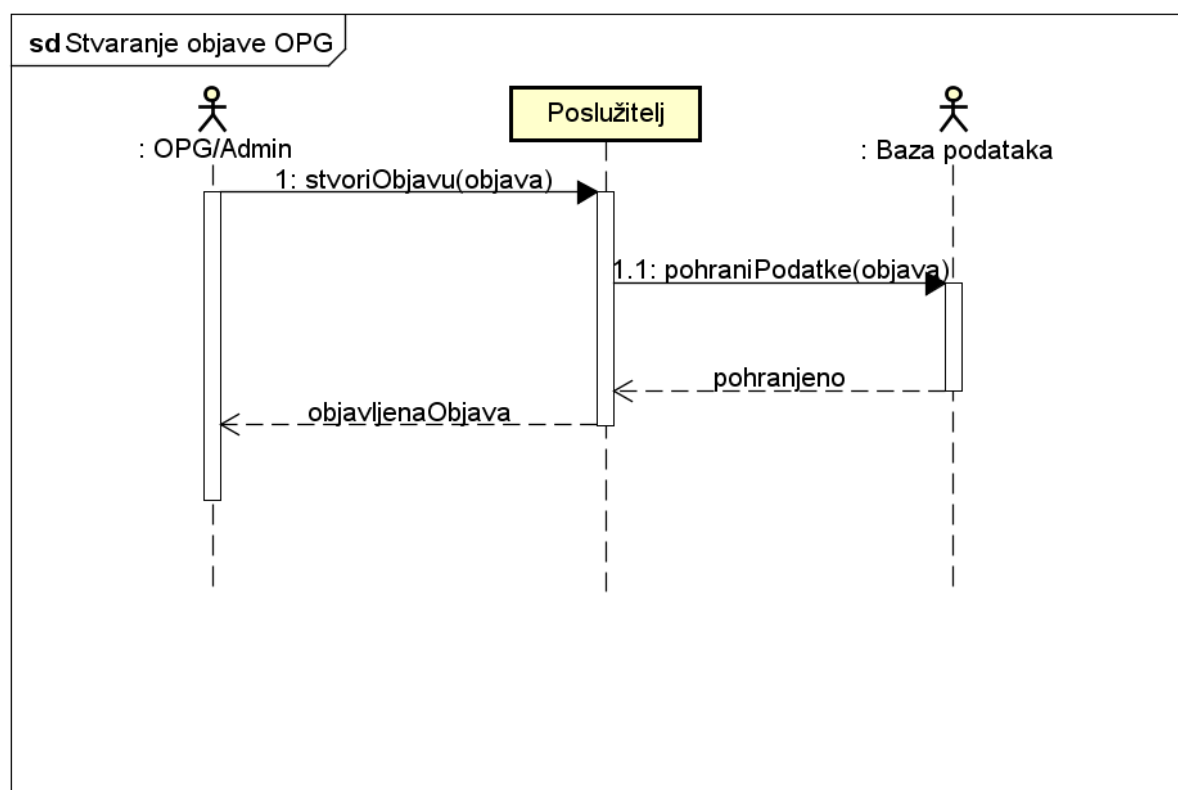
Korisnik ili administrator šalju zahtjev za brisanje profila. Zahtjev se obrađuje preko korisničkog identifikacijskog broja. Aplikacija prosljeđuje zahtjev bazi podataka koja uklanja sve podatke koje je imala o korisniku, te šalje nazad aplikaciji potvrdu o uspješnom brisanju. Aplikacija odjavljuje korisnika u slučaju da je ovaj prijavljen u sustav.



Slika 4.7 Sekvencijski dijagram UC5

Obrazac uporabe UC6:

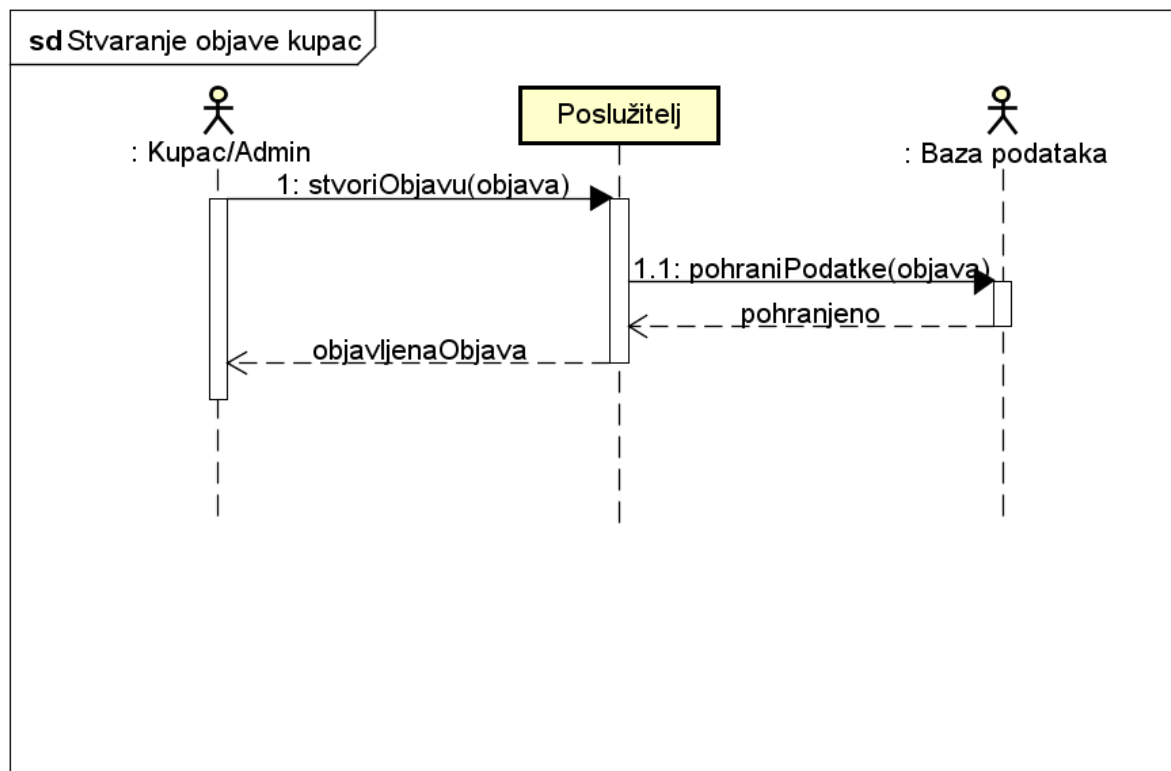
Korisnik koji ima ulogu OPG-a ili Administrator mogu stvoriti objavu na profilu OPG-a. Objavu stvaraju preko sučelja koji nudi aplikacija koja tu objavu pohranjuje u bazu podataka samo onda kada je spojena na Internet.



Slika 4.8 Sekvencijski dijagram UC6

Obrazac uporabe UC7:

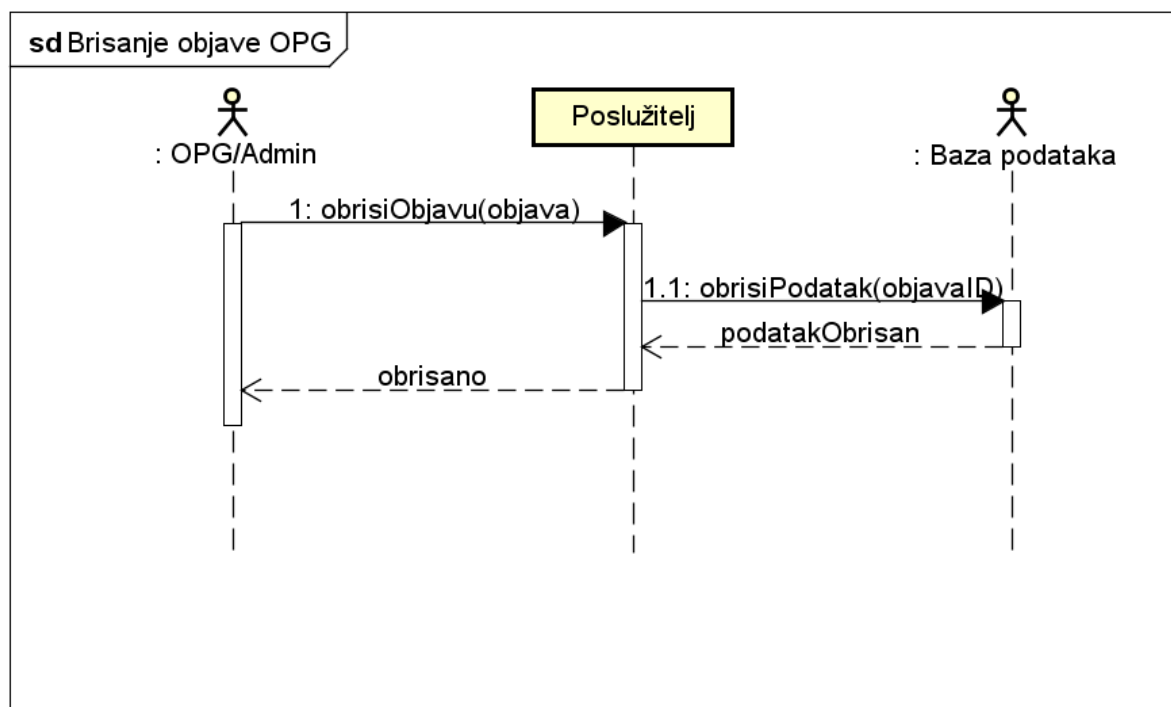
Korisnik koji ima ulogu kupca ili Administrator mogu stvoriti objavu na profilu kupca samo onda kada se dogodila transakcija. Objavu stvaraju preko sučelja koji nudi aplikacija koja tu objavu pohranjuje u bazu podataka samo onda kada je spojena na Internet.



Slika 4.9 Sekvencijski dijagram UC7

Obrazac uporabe UC8:

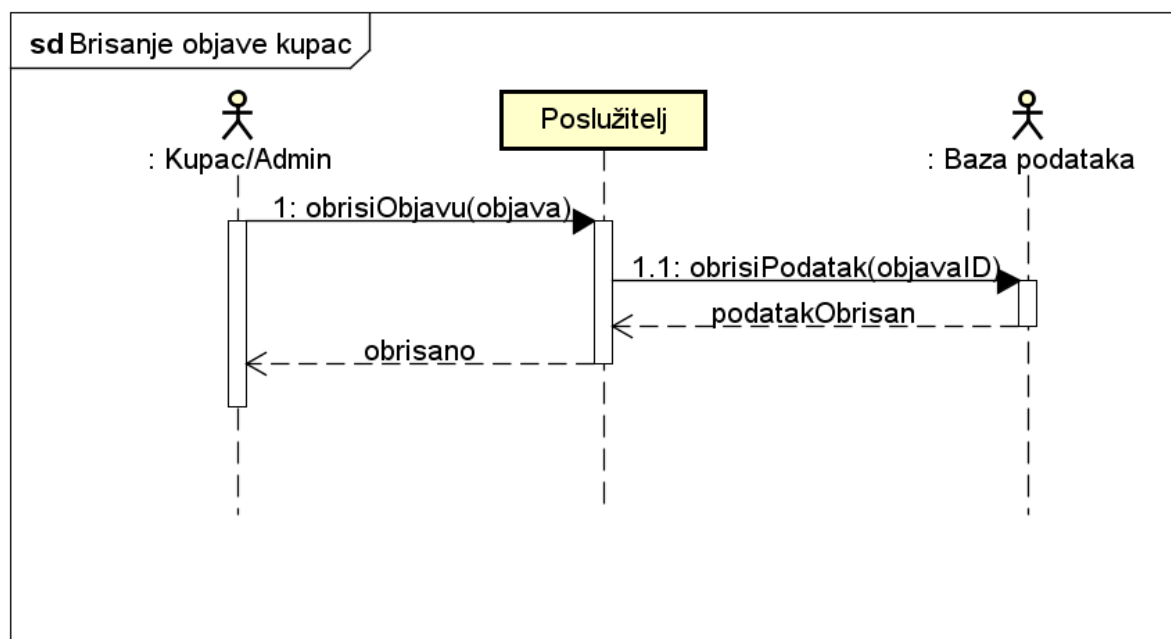
Administrator ili OPG-ovac šalje zahtjev aplikaciji da izbriše određenu objavu. Ako je aplikacija povezana s internetom ona proslijeđuje zahtjev bazi podataka koja briše dotičnu objavu.



Slika 4.10 Sekvencijski dijagram UC8

Obrazac uporabe UC9:

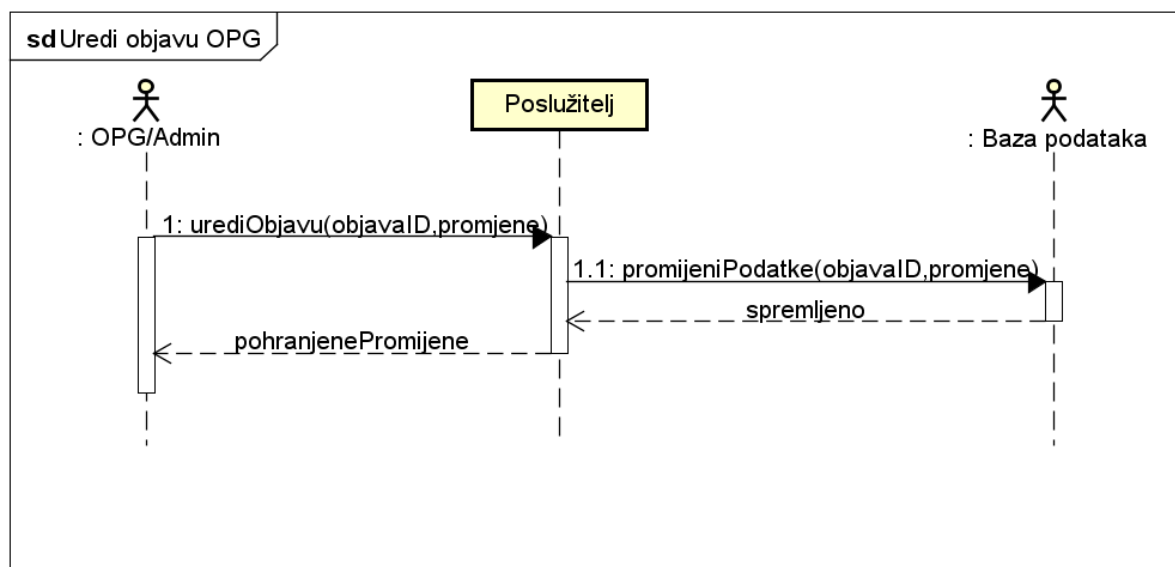
Administrator ili OPG-ovac šalju zahtjev aplikaciji da izbriše određenu objavu. Ako je aplikacija povezana s internetom ona proslijeđuje zahtjev bazi podataka koja briše dotičnu objavu.



Slika 4.11 Sekvencijski dijagram UC9

Obrazac uporabe UC10:

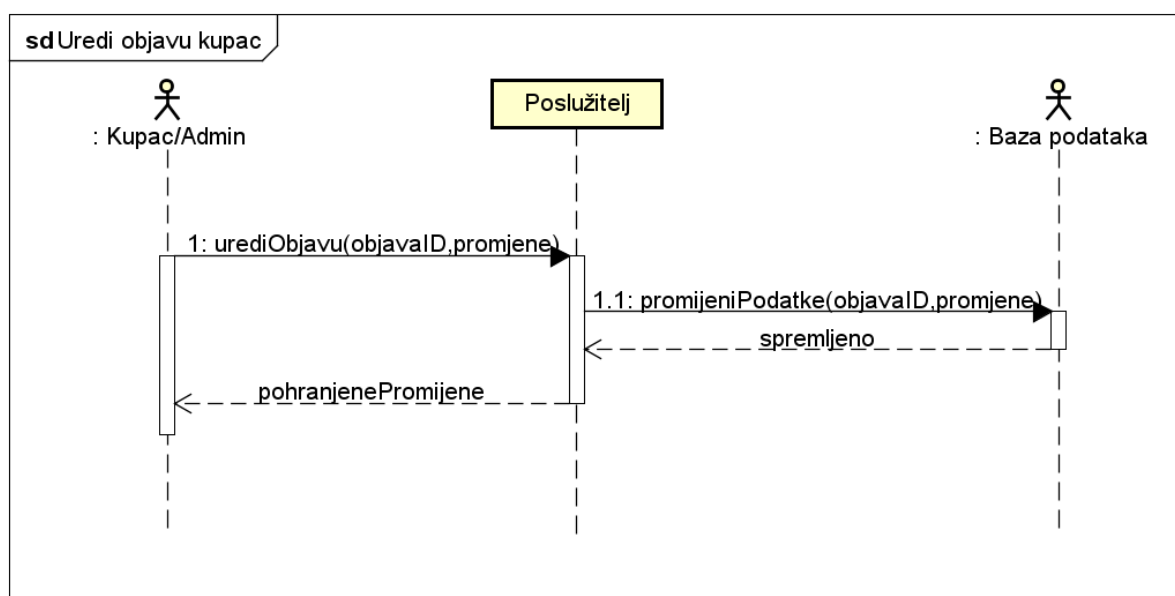
OPG-ovac ili administrator uređuju pojedinu objavu kroz ponuđeno sučelje i napravljene promijene preko aplikacije se prosljeđuju bazi podataka koja ih pohranjuje, te o svakoj takvoj pohrani obavješćuje aplikaciju koja onda to prenosi OPG-ovcu ili administratoru (ovisno koji vrši preinake).



Slika 4.12 Sekvencijski dijagram UC10

Obrazac uporabe UC11:

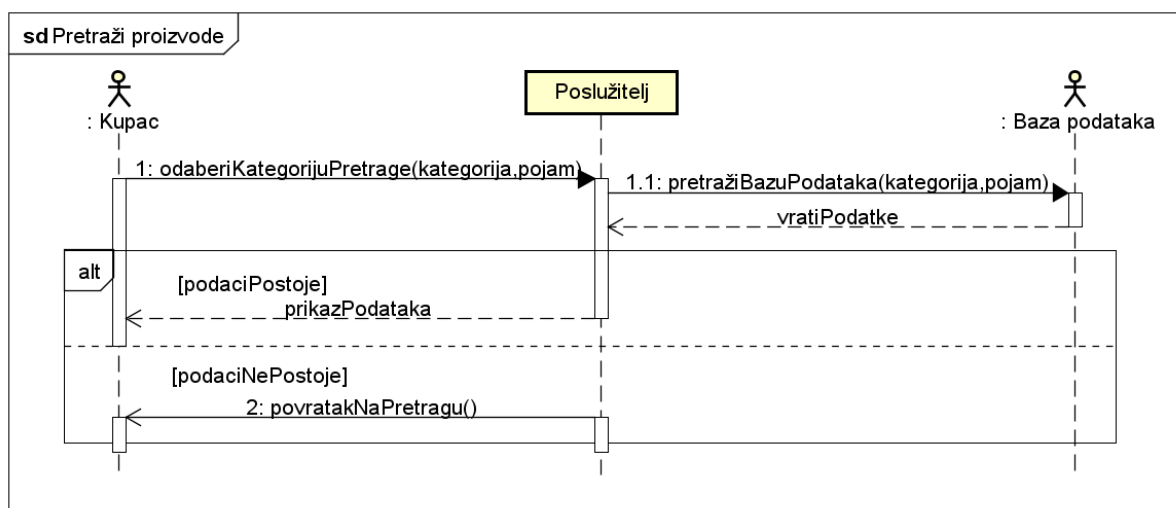
Kupac ili administrator uređuju pojedinu objavu kroz ponuđeno sučelje i napravljene promijene preko aplikacije se prosljeđuju bazi podataka koja ih pohranjuje. Baza podataka šalje obavijest o uspješno spremanju koji aplikacija prenosi korisniku.



Slika 4.13 Sekvencijski dijagram UC11

Obrazac uporabe UC12:

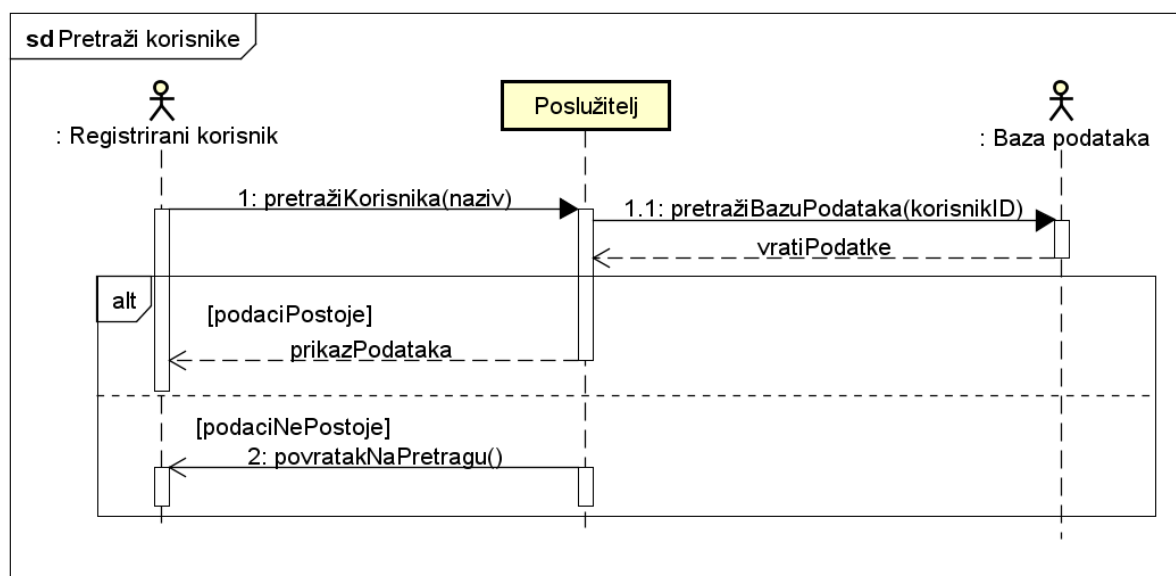
Kupac pretražuje proizvode tako da odabere kategoriju za koju pretpostavlja da sadrži traženi proizvod, te u tražilicu unese naziv proizvoda. Tada aplikacija šalje upit bazi podataka koja onda pretražuje postoji li takav proizvod.



Slika 4.14 Sekvencijski dijagram UC12

Obrazac uporabe UC13:

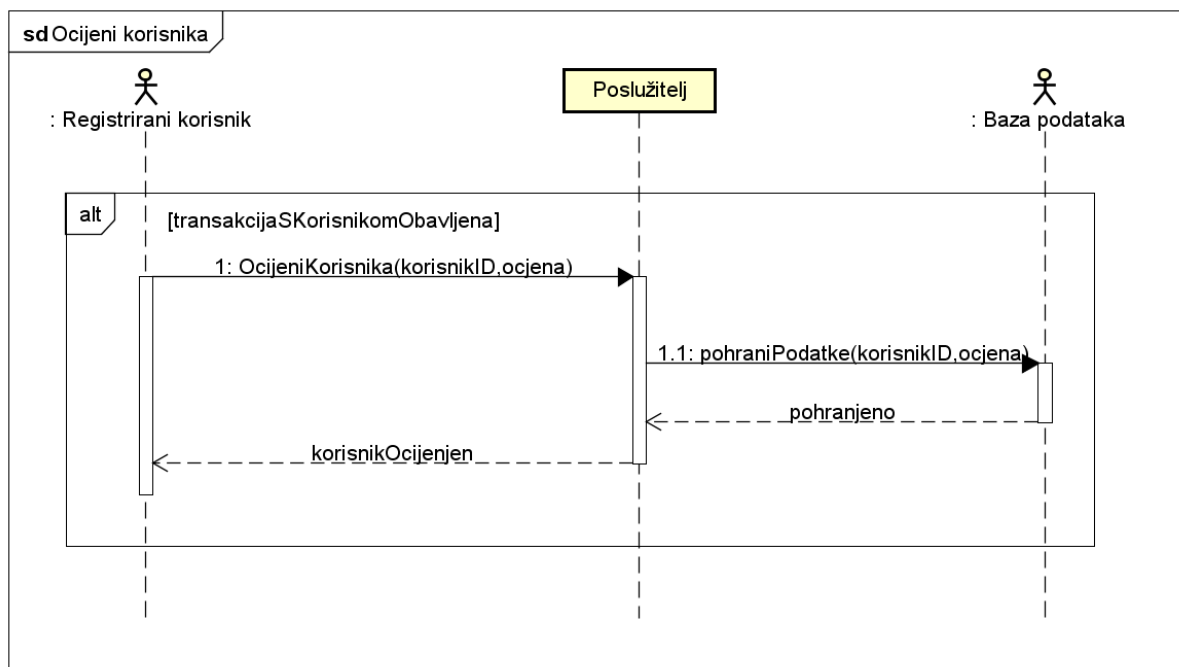
Registrirani korisnik unosi naziv korisnika koga želi pronaći pretragom. Aplikacija pretražuje lokalno imena korisnika i za svakoga koji u dovoljnoj mjeri zadovoljava uvjete pretrage uzima njegov ID i šalje upit bazi podataka koja onda vraća podatke o korisnicima.



Slika 4.15 Sekvencijski dijagram UC13

Obrazac uporabe UC14:

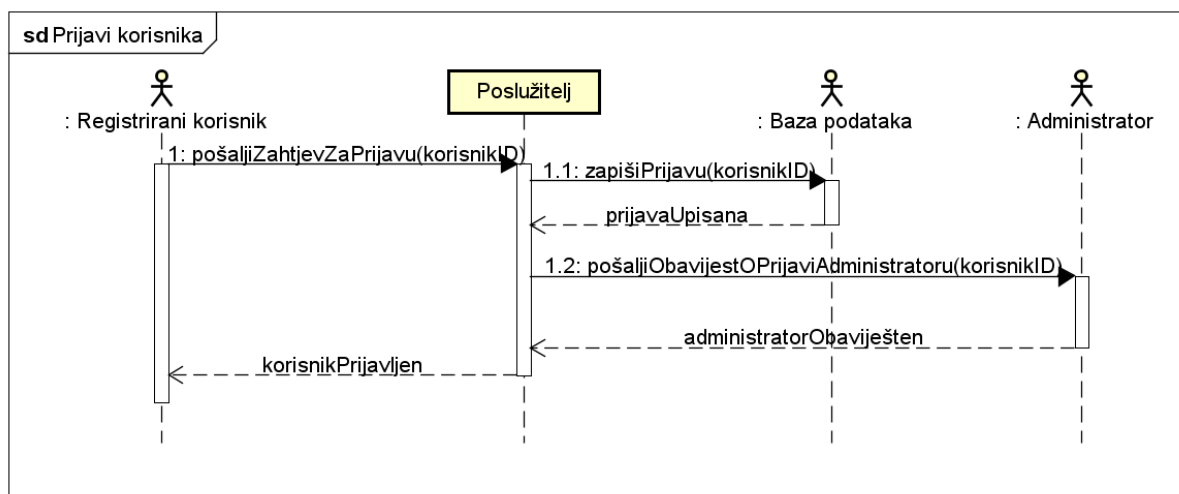
Nakon obavljene transakcije, aplikacija omogućuje registriranim korisnicima da si međusobno dodijele ocjenu kojom ocjenjuju iskustvo međusobnog poslovanja. Ocjena se upisuje u bazu podataka i računa u korisnikovu prosječnu ocjenu.



Slika 4.16 Sekvencijski dijagram UC14

Obrazac uporabe 15:

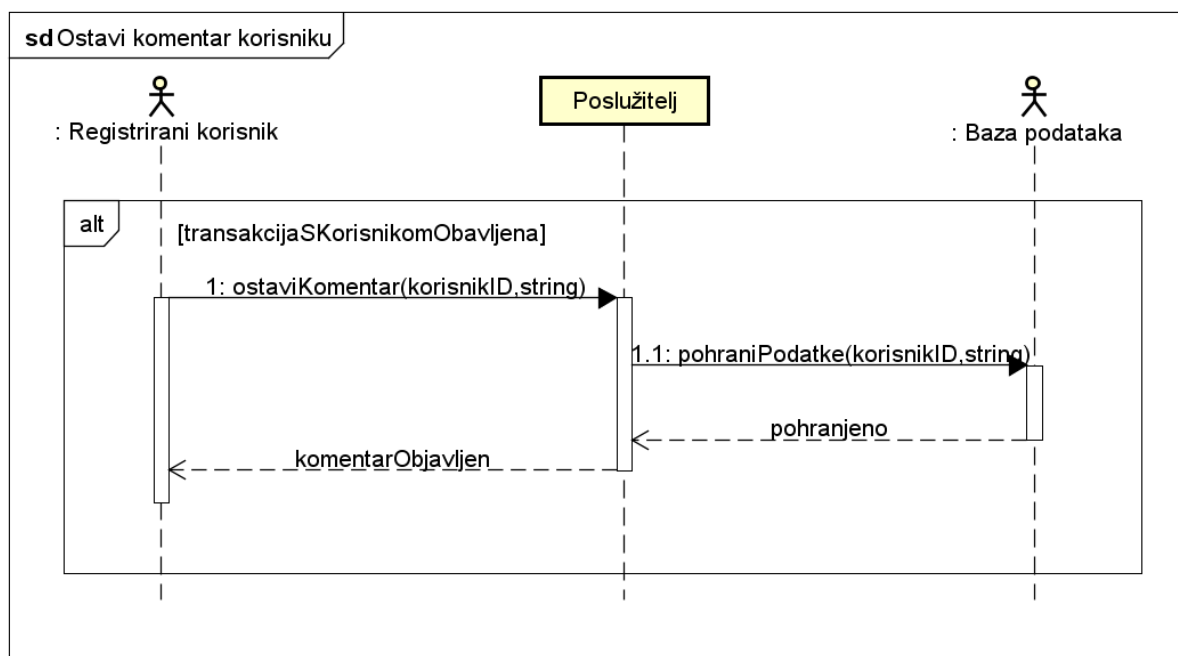
Registrirani korisnik može poslati zahtjev za prijavu drugog korisnika ukoliko primjeti neprimjereno ponašanje. Prijava se zapisuje u bazu podataka i šalje se administratoru na obradu. Administrator prijavu obrađuje i odlučuje o mogućim sankcijama.



Slika 4.17 Sekvencijski dijagram UC15

Obrazac uporabe 16:

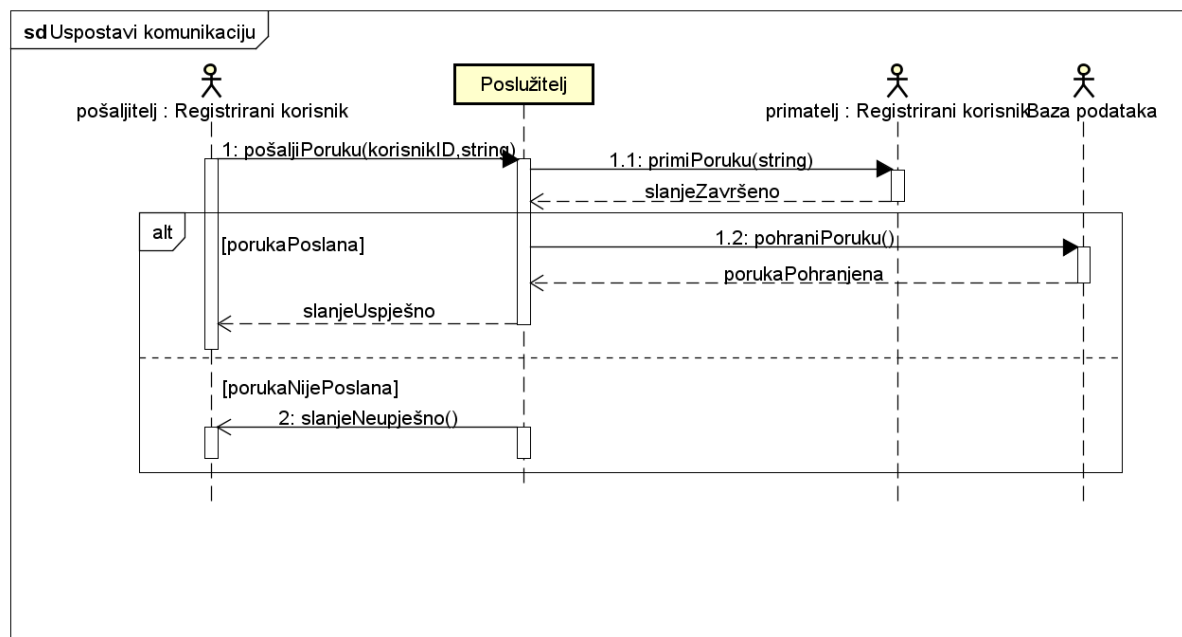
Nakon obavljanja transakcije, registriranim korisnicima se omogućuje da međusobno objavljuju javne komentare o međusobnom poslovnom iskustvu. Komentari se pohranjuju u bazu podataka i jedino ih administrator uz valjan razlog može ukloniti.



Slika 4.18 Sekvencijski dijagram UC16

Obrazac uporabe 17:

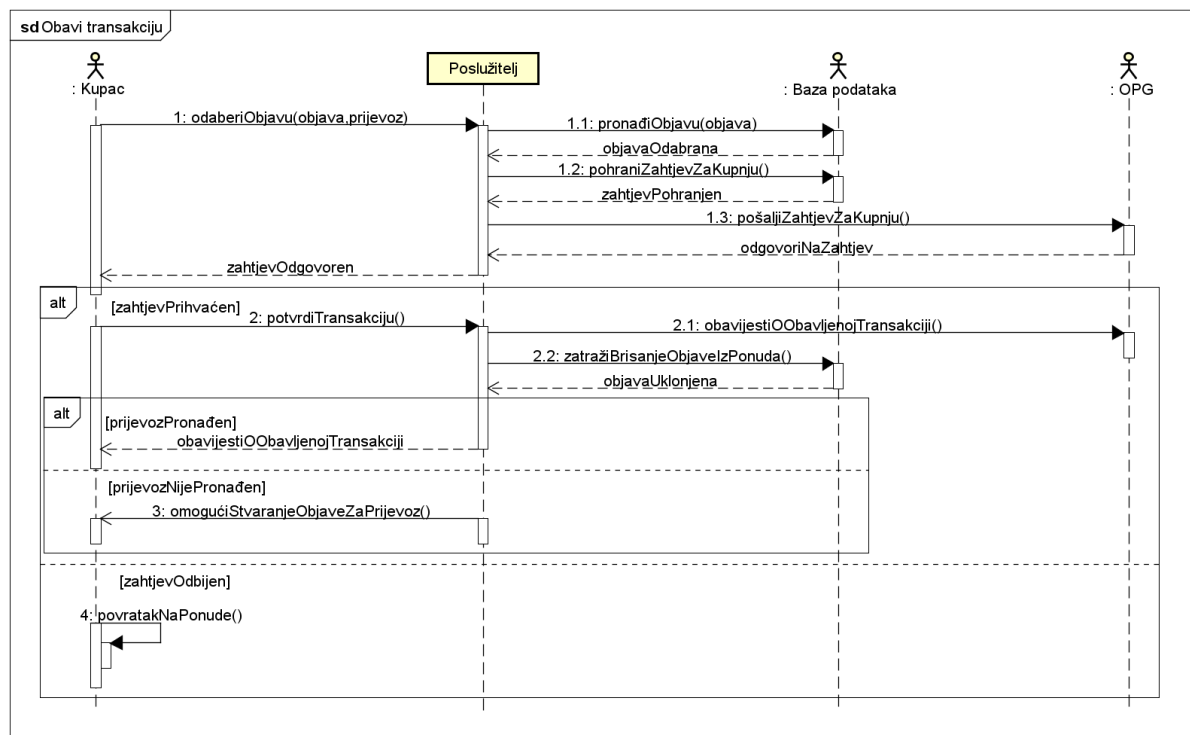
Registrirani korisnik odabire željenog korisnika, pritiskom na gumb otvara prozor za komunikaciju i šalje poruku. Primatelj ne mora biti prijavljen u sustav da primi poruku. Ako je poruku uspješno poslana, zapisuje se u bazu podataka, inače dojavljuje da poruka nije uspješno poslana.



Slika 4.19 Sekvencijski dijagram UC17

Obrazac uporabe 18:

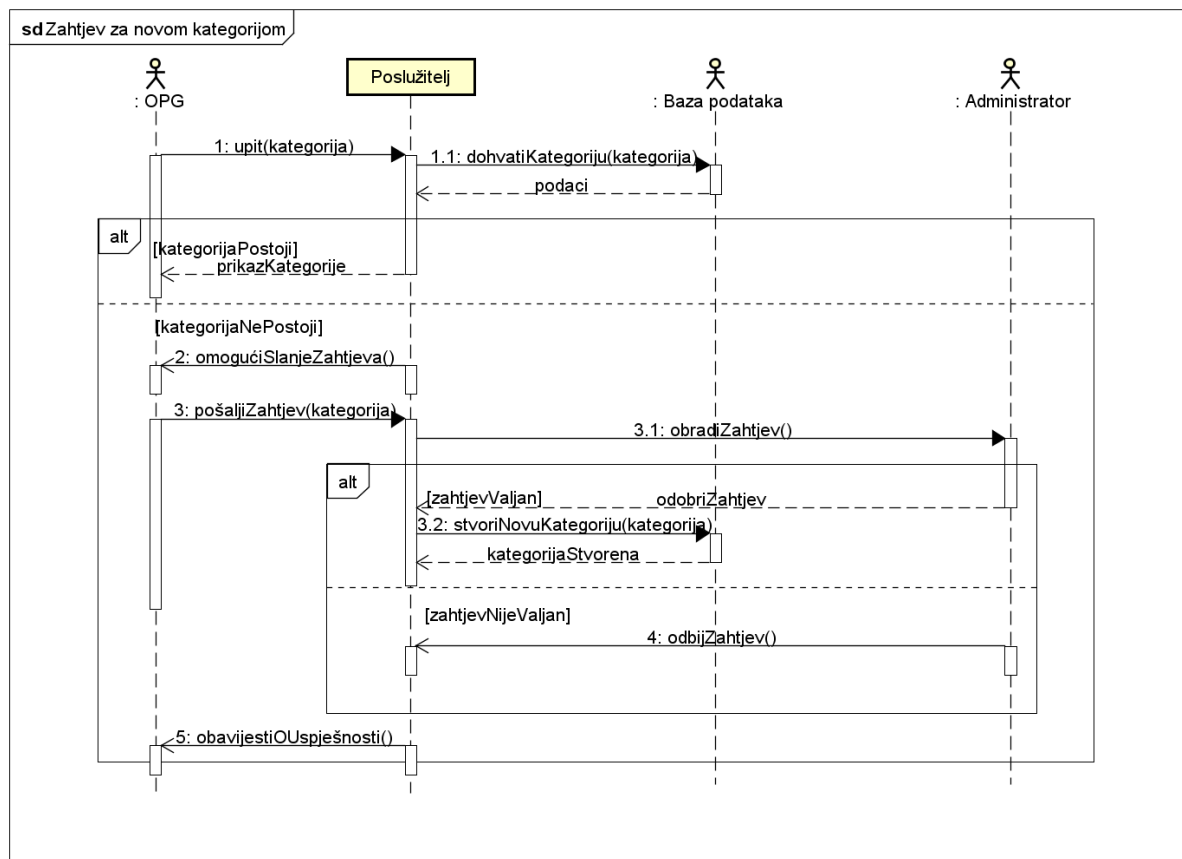
Kupac odabire objavu koju želi kupiti i šalje zahtjev za kupnju OPG-u. U tom zahtjevu ima mogućnost specificirati ima li već dogovoren prijevoz. Nakon dogovora, OPG odgovara na zahtjev. Ako OPG prihvati zahtjev, kupac potvrđuje transakciju i objava se uklanja iz ponude. Ukoliko kupac nema dogovoren prijevoz, otvara mu se prozor za stvaranje objave za prijevoz, inače ga se samo obavještava o uspješno obavljenoj transakciji. Ako OPG iz nekog razloga nije prihvatio zahtjev, kupca se vraća natrag na ponude.



Slika 4.20 Sekvencijski dijagram UC18

Obrazac uporabe 19:

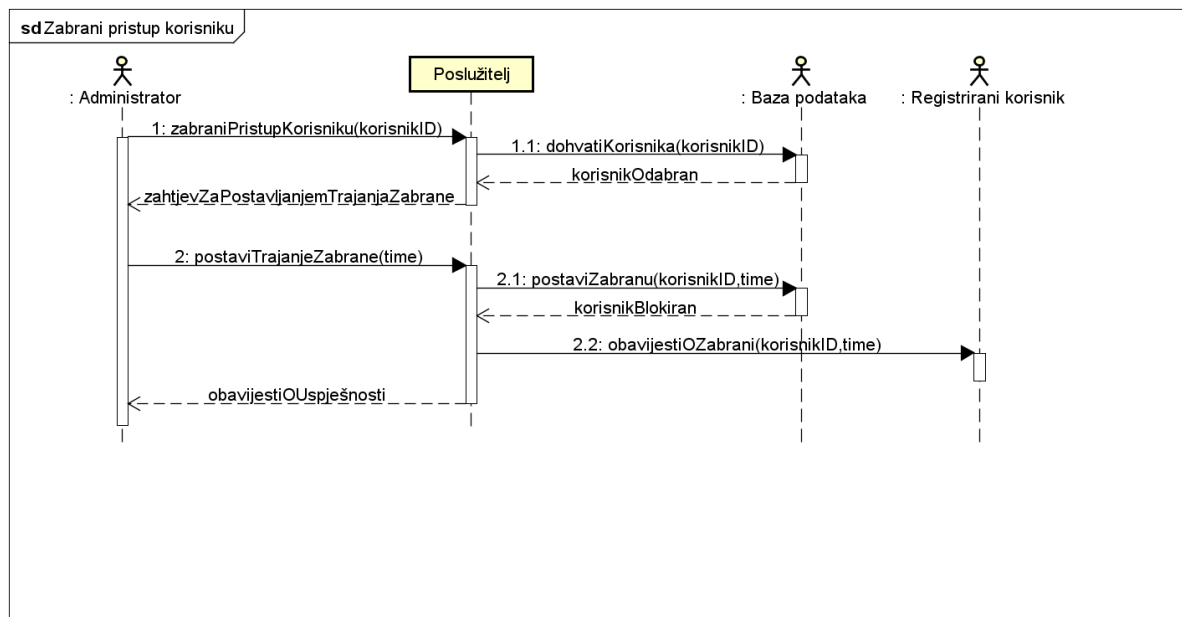
Ako OPG želi detaljnije kategorizirati svoje proizvode, prvo provjerava postoji li već takva kategorija u bazi podataka. Ako ne postoji, OPG-u se omogućuje slanje zahtjeva za novom kategorijom. Zahtjev se proslijeđuje administratoru koji ga obrađuje i odlučuje o njegovoj valjanosti. Ako je zahtjev valjan, administrator ga odobrava i nova kategorija se stvara u bazi podataka, inače ga odbija. O rezultatu se obaviještava OPG.



Slika 4.21 Sekvencijski dijagram UC19

Obrazac uporabe 20:

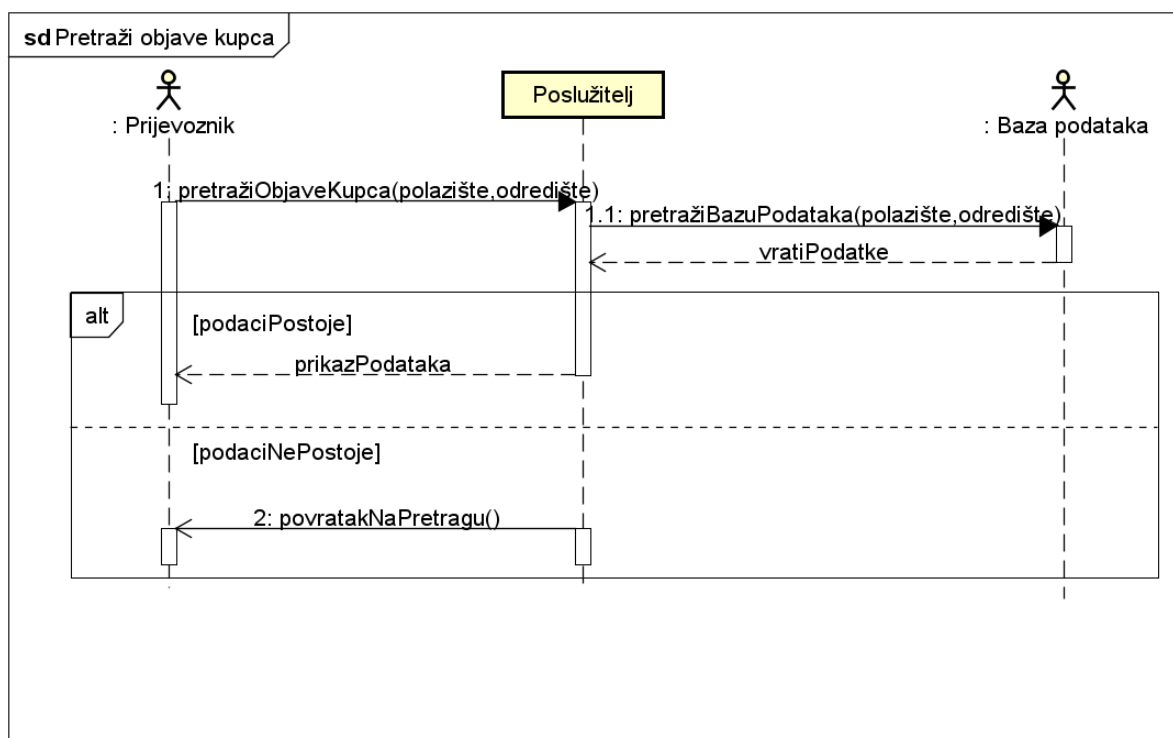
Nakon što primi obavijest o neprimjerenom ponašanju korisnika, administrator odlučuje o zabrani pristupa. Zabrana može biti trajna ili privremena, administrator odlučuje o trajanju zabrane. U bazi podataka se korisnik blokira na odabrano vrijeme i korisnika se obavještava o postavljenoj zabrani.



Slika 4.22 Sekvencijski dijagram UC20

Obrazac uporabe 21:

Prijevoznik pretražuje objave za prijevoz po polazištu i/ili po odredištu. Poslužitelj u bazi podataka pronalazi sve rezultate koji odgovaraju unesenim parametrima i ako pronađe takve podatke vraća ih prijevozniku.



Slika 4.23 Sekvencijski dijagram UC21

5.Ostali zahtjevi

- Sustav mora podržavati paralelni rad više korisnika.
- Sustav mora biti jednostavan da ga svi korisnici mogu koristiti.
- Sustav treba obavijestiti korisnika ako je došlo do promjena u trenutno odabranoj ponudi.
- Sustav mora podržavati hrvatske znakove.
- Nadogradnja ili promjene funkcionalnosti ne smiju narušiti već postojeće funkcionalnosti.
- Nepravilno ponašanje korisnika ne smije narušiti funkcionalnost sustava.
- Korisnici se mogu međusobno ocjenjivati ocjenama od 1 do 5 (5 kao najveća ocjena).
- Korisnici količine proizvoda moraju izražavati u jedinicama SI sustava, a cijenu u HRK.

6.Arhitektura i dizajn sustava

Svrha, opći prioriteti i skica sustava

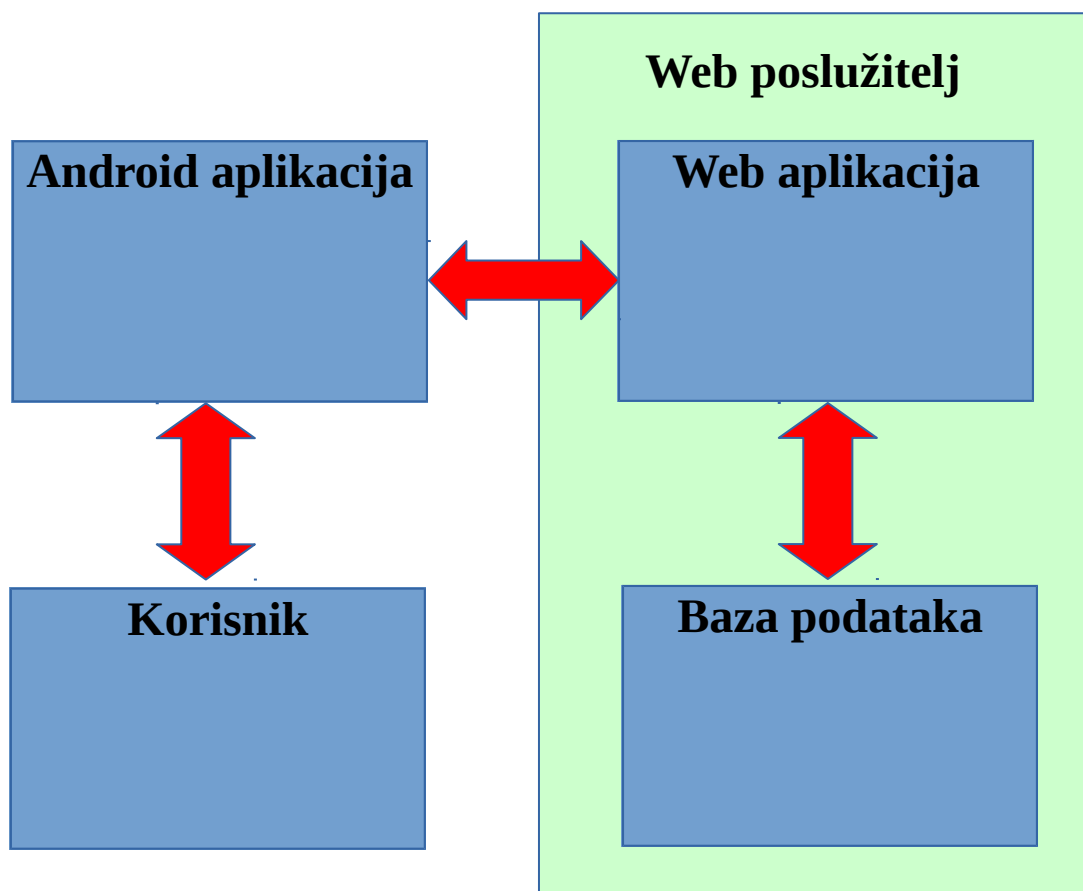
Kako bismo izgradili sustav nužno je bilo odabrati njegovu opću arhitekturu. Uvjeti pri odabiru arhitekture koji se nameću samom naravi aplikacije jesu: olakšano upravljanje sustavom, mobilnost, jednostavna interakcija, pristup više korisnika odjednom. Arhitektura koja ispunjava navedene zahtjeve jest Android aplikacija sa web poslužiteljem.

Arhitekturu sustava možemo podijeliti na četiri podsustava (Slika 6.1.):

- Android aplikacija
- Web poslužitelj
- Web aplikacija
- Baza podataka

Rad sa sustavom odvijati će se na sljedeći način:

- Korisnik putem Android aplikacije šalje zahtjev web poslužitelju
- Web poslužitelj prosljeđuje zahtjev web aplikaciji
- Web aplikacija obrađuje zahtjev te komunicira sa bazom podataka ukoliko je potrebno
- Web poslužitelj vraća korisniku odgovor kojega prikazuje Android aplikacija



Slika 6.1 Skica sustava

Web poslužitelj

Kako bi cijeli sustav ispravno radio potreban nam je web poslužitelj preko kojega će se slati i primati HTTP zahtjevi. Na web poslužitelju se nalaze web aplikacija i baza podataka.

Web aplikacija

Web aplikacija je zadužena za operativni dio sustava. Web aplikacija nam je potrebna kako bi obrađivala zahtjeve koje zaprima i slala ih natrag korisniku. Komunikaciju sa bazom podataka obavlja po potrebi te dodaje, briše ili izmjenjuje postojeće podatke.

Android aplikacija

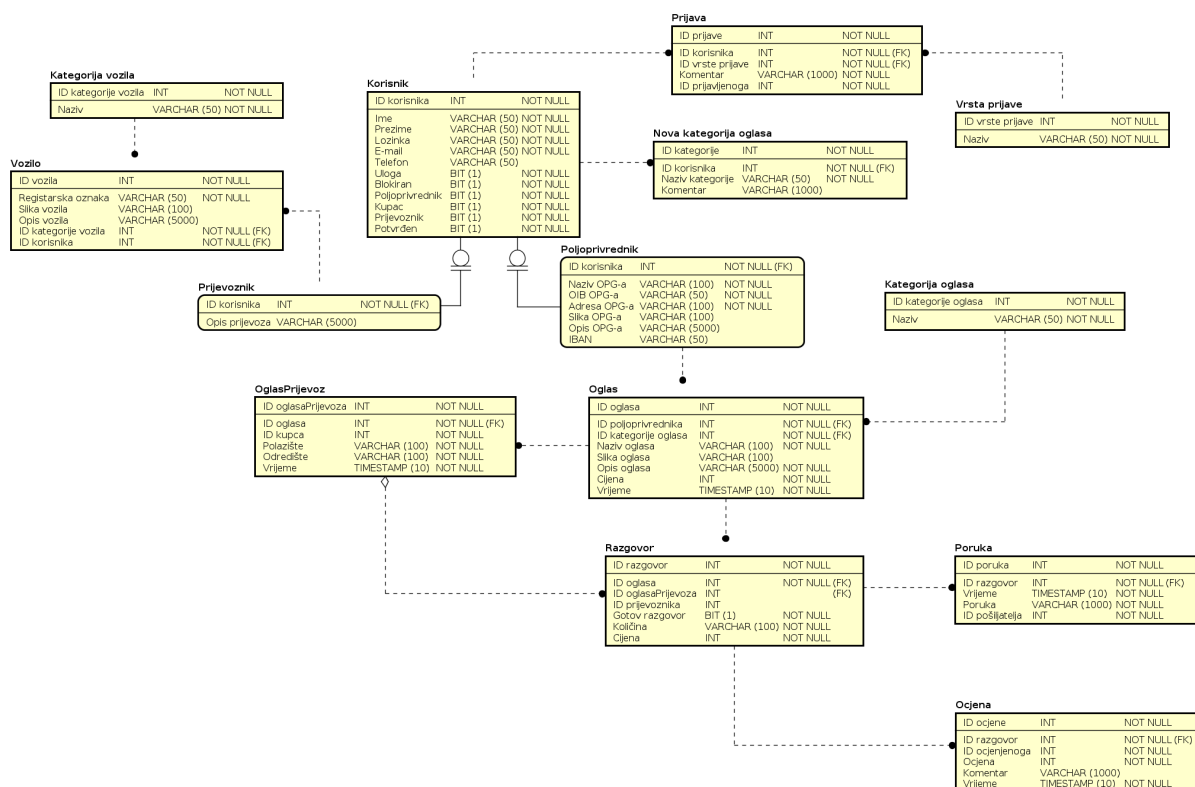
Android aplikacija služi korisniku kao vizualna prezentacija sustava. Preko aplikacije korisnik komunicira sa sustavom, šalje i prima zahtjeve. Android aplikacija omogućuje korisniku mobilnost i jednostavnost korištenja.

Baza podataka

Kako bi svi podaci unutar sustava bili pohranjeni i lako dostupni, imamo bazu podataka. Zadaća baze podataka je da omogućuje jednostavnu izmjenu, dopunu i brisanje podataka. Naša baza podataka je relacijska što znači da ima tablice podataka koje su povezane relacijama (Slika 6.2). Entiteti koje naša baza podataka sadrži su:

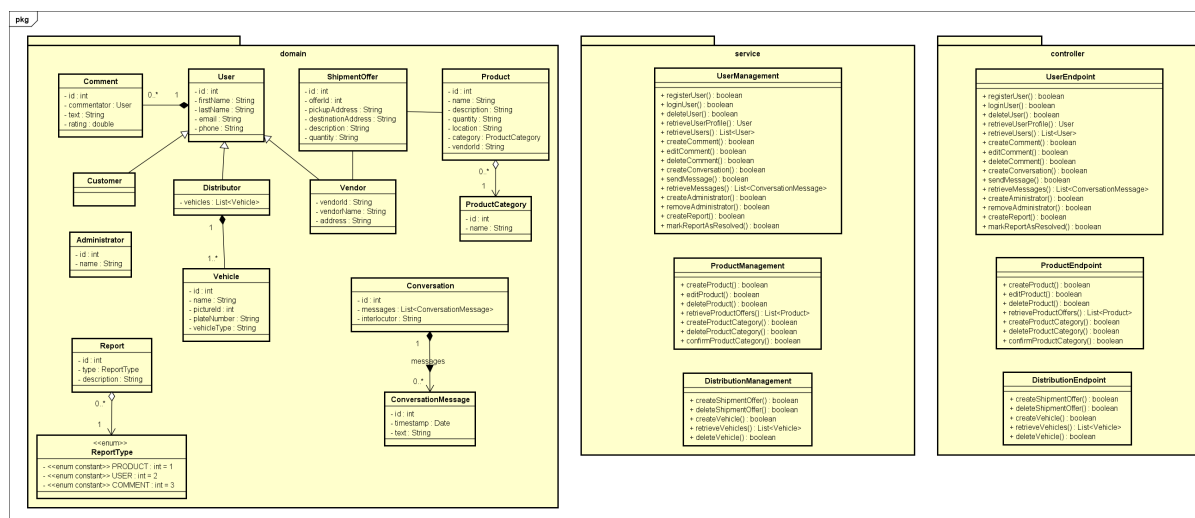
- **Korisnik**
 - Sadrži informacije o korisniku, njegovu razinu pristupa te je li blokiran
- **Prijevoznik**
 - Sadrži opis prijevoza koji određeni prijevoznik nudi
- **Kategorija vozila**
 - Sadrži sve kategorije vozila kojima je moguće prevesti proizvode
- **Vozilo**
 - Sadrži sve potrebne informacije o pojedinom vozilu
- **Poljoprivrednik**
 - Sadrži informacije o obiteljskom poljoprivrednom gospodarstvu
- **Nova kategorija oglasa**
 - Sadrži sve kategorije oglasa koje su korisnici predložili da se dodaju u sustav
- **Kategorija oglasa**

- Sadrži sve kategorije oglasa koje postoje unutar sustava
- **Oglas**
 - Sadrži sve oglase sa proizvodima koje su objavili poljoprivrednici
- **OglasPrijevoz**
 - Sadrži oglase u kojima kupci traže prijevoz za proizvode koje su kupili
- **Razgovor**
 - Sadrži sve razgovore koji su vođeni za dogovor oko oglasa za proizvode i prijevoz
- **Poruka**
 - Sadrži sve poruke koje su poslone unutar razgovora
- **Ocjena**
 - Sadrži sve ocjene i komentare kojima su se korisnici međusobno ocjenjivali i komentirali
- **Vrsta prijave**
 - Sadrži sve vrste prijave korisnika koje su moguće unutar sustava
- **Prijava**
 - Sadrži međusobne prijave korisnika poslane administratoru



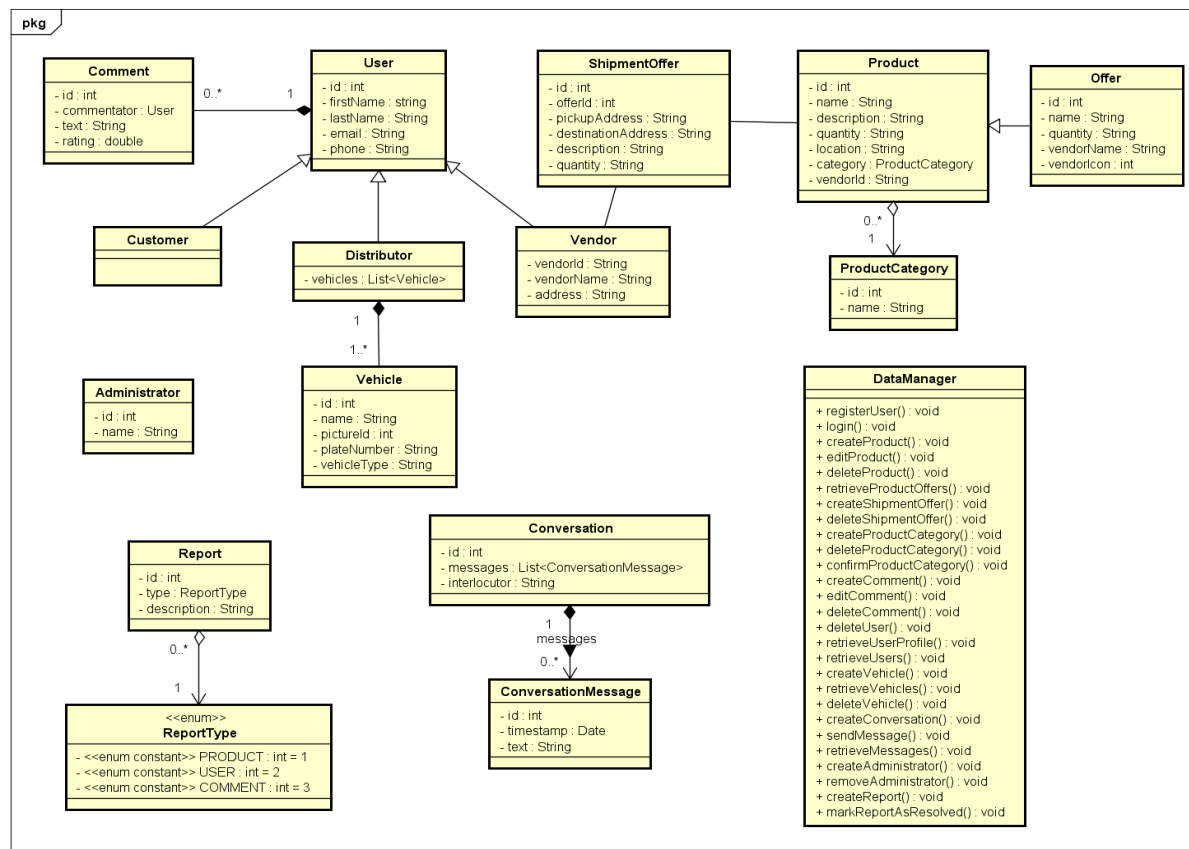
Slika 6.2 ER model baze podataka

Dijagram razreda s opisom



Slika 6.3 Poslužiteljski UML dijagram razreda

Serverski dio aplikacije strukturno je podijeljen na tri paketa: domenski koji sadrži tipove podataka, servisni koju obavlja operacije nad podacima i kontrolerski koji zaprima zahtjeve klijenta. (Slika 6.3.)

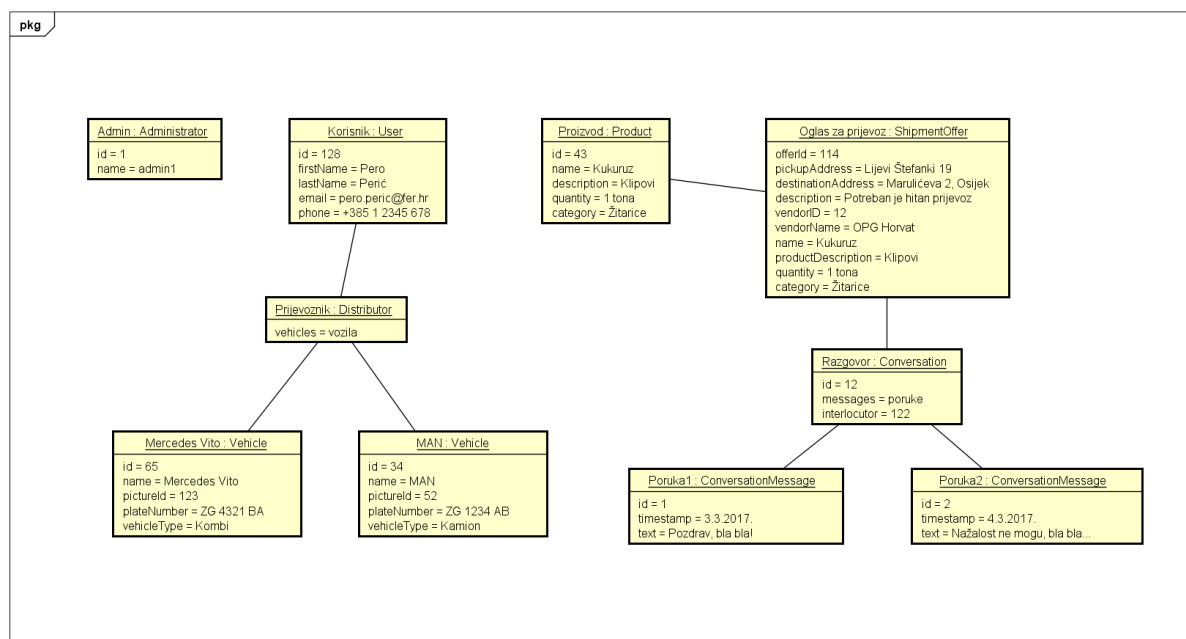


Slika 6.4 Klijentski UML dijagram razreda

Razred 'DataManager' obavlja manipulaciju podacima na klijentskom dijelu aplikacije. Razred **User** je osnovni razred, svaki registrirani korisnik automatski dobija status kupca (**Customer**), ali ima mogućnost odabrati status prijevoznika (**Distributor**) i OPG-a (**Vendor**). Prijevoznik na svom profilu definira vozilo (**Vehicle**) kojim će obavljati dostavu. OPG-ovi imaju mogućnost objavljivanja ponuda (**Offer**) u kojima na prodaju stavljaju određene proizvode (**Product**) i njihovu količinu. Svi proizvodi se definiraju nekom kategorijom (**ProductCategory**). OPG-ovi mogu zatražiti dodavanje detaljnijih kategorija ako žele preciznije definirati svoj proizvod. Nakon dogovorene transakcije, stvara se ponuda za prijevoz (**ShipmentOffer**) kojom se traži dostava proizvoda s adrese OPG-a na adresu kupca. Registrirani korisnici mogu objavljivati javne komentare (**Comment**) o drugim korisnicima koji su vidljivi na njihovim profilima. Razred **Conversation** sadrži poruke (**ConversationMessage**) između korisnika. Korisnici mogu prijaviti druge korisnike ukoliko primjete neprimjereno ponašanje. To rade tako da pošalju zahtjev za prijavu (**Report**) u kojem definiraju tip prijave (**ReportType**). Prijaviti mogu proizvod (loša kvaliteta, lažno oglašavanje,...), korisnika (korištenje neprimjerenog riječnika, pokušaj varanja,...) ili komentar (uvredljiv sadržaj, lažne informacije,...). Zahtjeve tada obrađuje **Administrator** koji donosi odluku o mogućoj kazni. On upravlja sustavom i razrješava probleme između korisnika ili s aplikacijom.

Dijagram objekata

U slijedećem dijagramu objekata prikazan je korisnik koji je prijevoznik. Taj prijevoznik posjeduje kamion i kombi za vršenje usluge prijevoza robe. Prijevoznik je pronašao oglas o potrebi prijevoza kukuruza s lokacije Lijevi Štefanki 19 do lokacije Marulićeva 2, Osijek te je stupio u kontakt s kupcem tog proizvoda kako bi dogovorio cijenu, način i vrijeme prijevoza. No, iz priloženih poruka vidimo da do dogovora nije došlo.

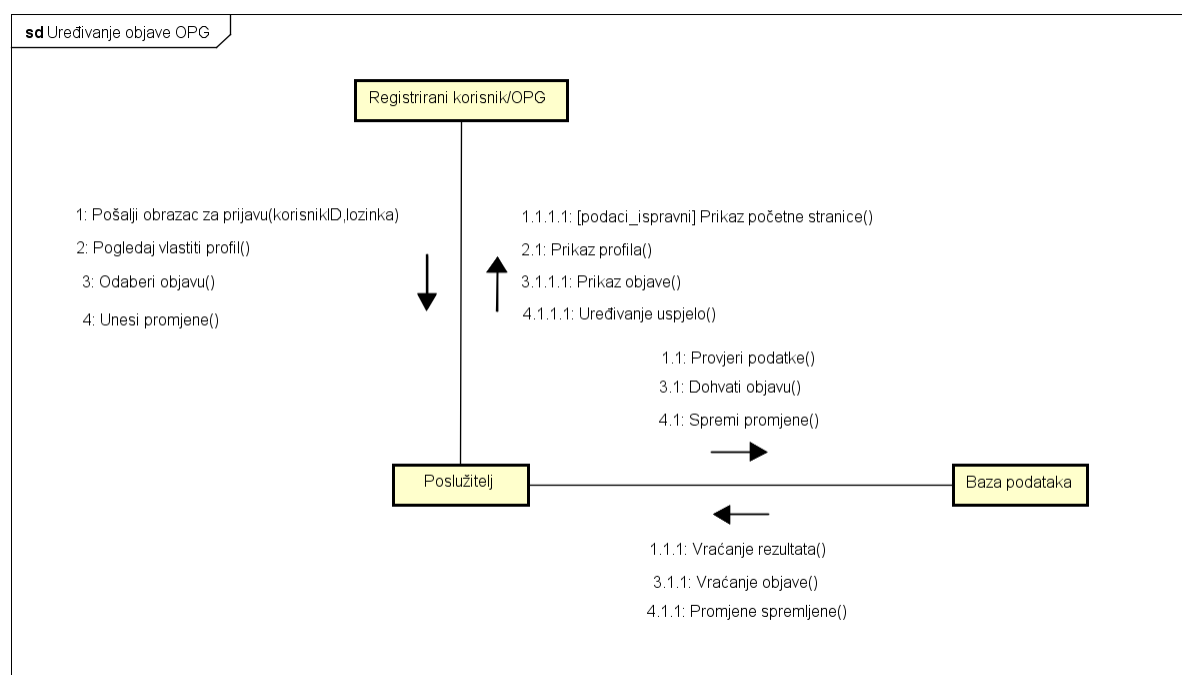


Slika 6.5 Dijagram objekata

Ostali UML dijagrami

Komunikacijski dijagram

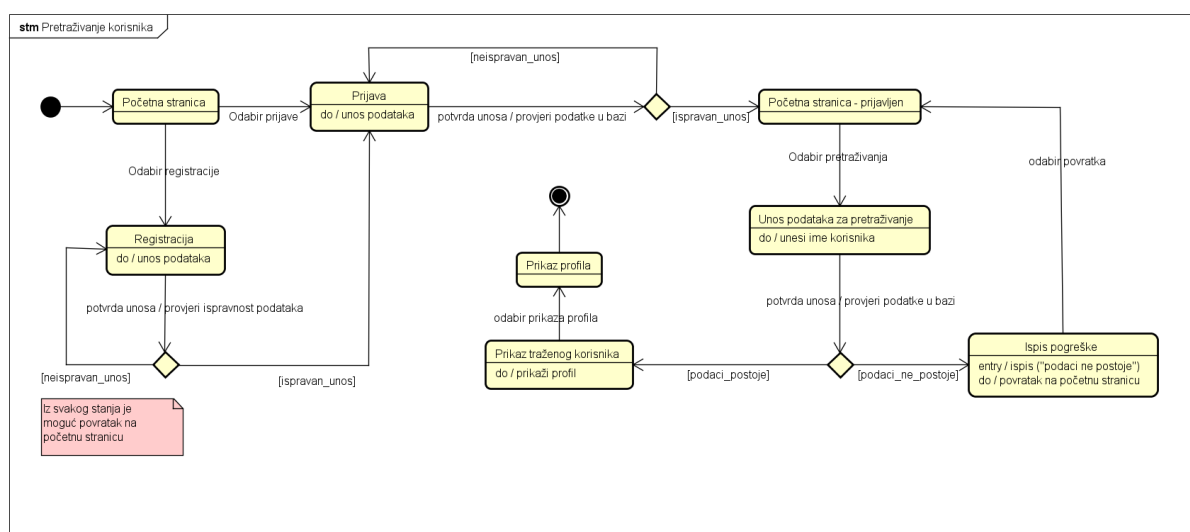
U komunikacijskom dijagramu na slici 6.6 prikazan je vremenski slijed komunikacije između registriranog korisnika, koji ima status OPG-a, poslužitelja i baze podataka u situaciji kada korisnik želi urediti već postavljenu objavu. Nakon što korisnik unese podatke, poslužitelj ih provjerava u bazi i ako što su ispravni, omogućuje mu pristup sustavu. Korisnik otvara pregled svog profila i odabire objavu koju želi urediti. Na odabranoj objavi unosi željene promjene i potvrđuje ih. Poslužitelj te promjene prosljeđuje bazi podataka gdje se one spremaju. Korisnika se obavještava o uspješno unesenim promjenama.



Slika 6.6 Komunikacijski dijagram

Dijagram stanja

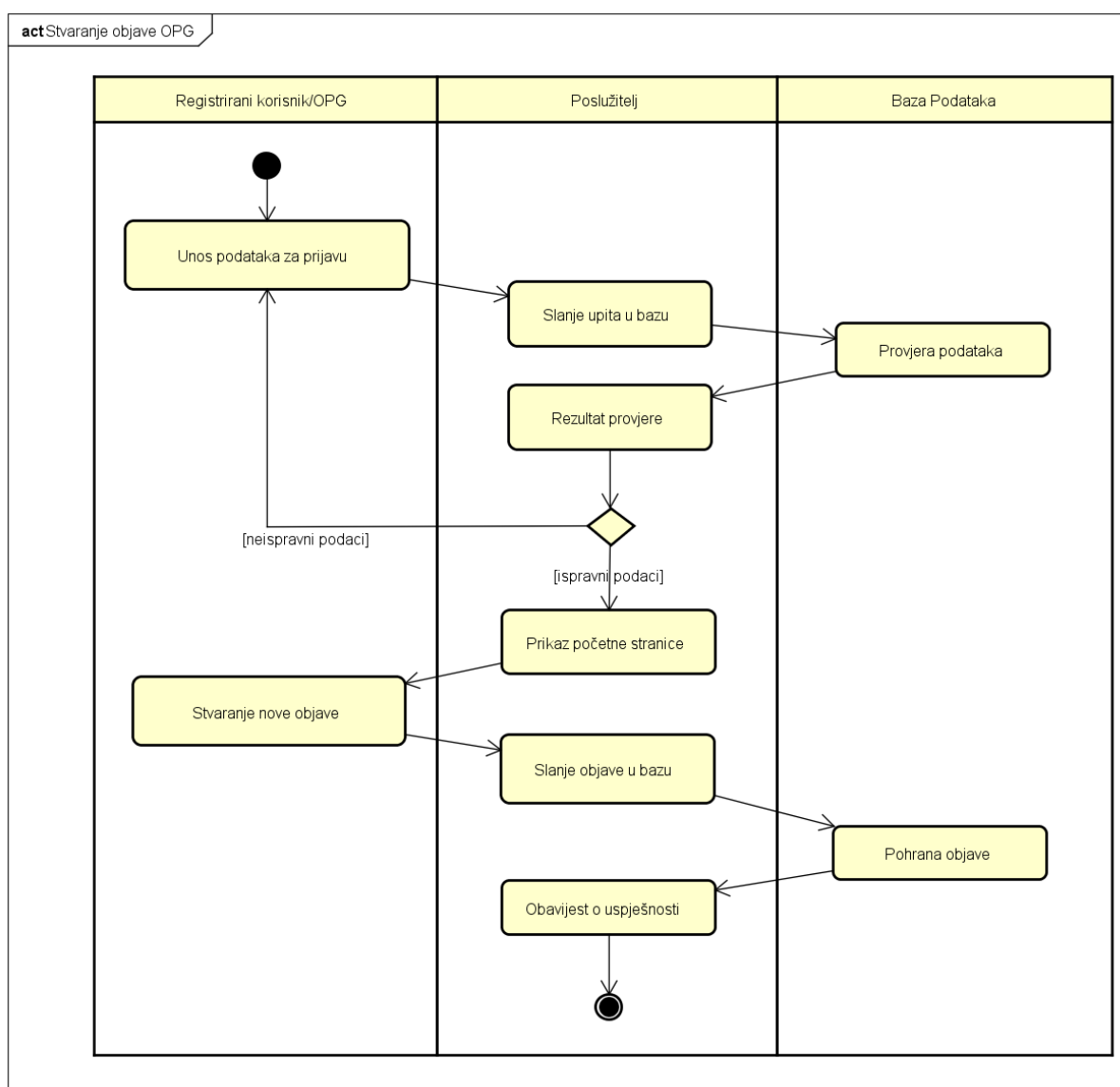
Ukoliko korisnik želi pretraživati druge korisnike aplikacije, mora se prvo registrirati. Na slici 6.7 prikazan je dijagram stanja koji opisuje tu funkcionalnost. Nakon što otvori početnu stranicu aplikacije, ponuđene su mu dvije opcije, prijava i registracija. Neregistrirani korisnik mora odabrati registraciju gdje onda unosi tražene osobne podatke. Poslužitelj te podatke provjerava u bazi podataka, ako su podaci već pohranjeni u bazi, znači da ih drugi korisnik već koristi pa će korisnika vratiti natrag na ponovnu registraciju. Ako su podaci ispravni, vratit će ga na stranicu za prijavu. Korisnik unosi potrebne podatke i potvrđuje ih. Poslužitelj provjerava podatke u bazi i ako su neispravni vraća korisnika na stranicu za prijavu, u suprotnom mu omogućuje pristup sustavu. Korisnik u traku za pretraživanje unosi ime traženog korisnika i potvrđuje svoj unos. Ukoliko podaci o traženom korisniku postoje, poslužitelj ih dohvaća iz baze i prikazuje korisniku. Ako podaci ne postoje, poslužitelj o tome obaviještava korisnika. U slučaju uspješne pretrage korisnik može pregledati profil pronađenog korisnika.



Slika 6.7 Dijagram stanja

Dijagram aktivnosti

Registrirani korisnici koji imaju ulogu OPG-a imaju mogućnost postavljanja objava za prodaju svojih proizvoda. Na slici 6.8 dijagramom aktivnosti opisan je model toka tog procesa. Registrirani korisnik se prvo treba prijaviti u sustav kako bi mogao postaviti objavu. Nakon što unese i potvrdi podatke za prijavu, poslužitelj ih provjerava u bazi i ako su ispravni omogućuje korisniku pristup sustavu, inače ga vraća natrag na prijavu. Kad je prijavljen u sustav korisnik odabire opciju stvaranja nove objave. Tu unosi sve potrebne podatke i objavljuje ju. Poslužitelj objavu šalje u bazu podataka gdje se zapisuje i sprema. Nakon toga obavještava korisnika o uspješnom postavljanju objave.

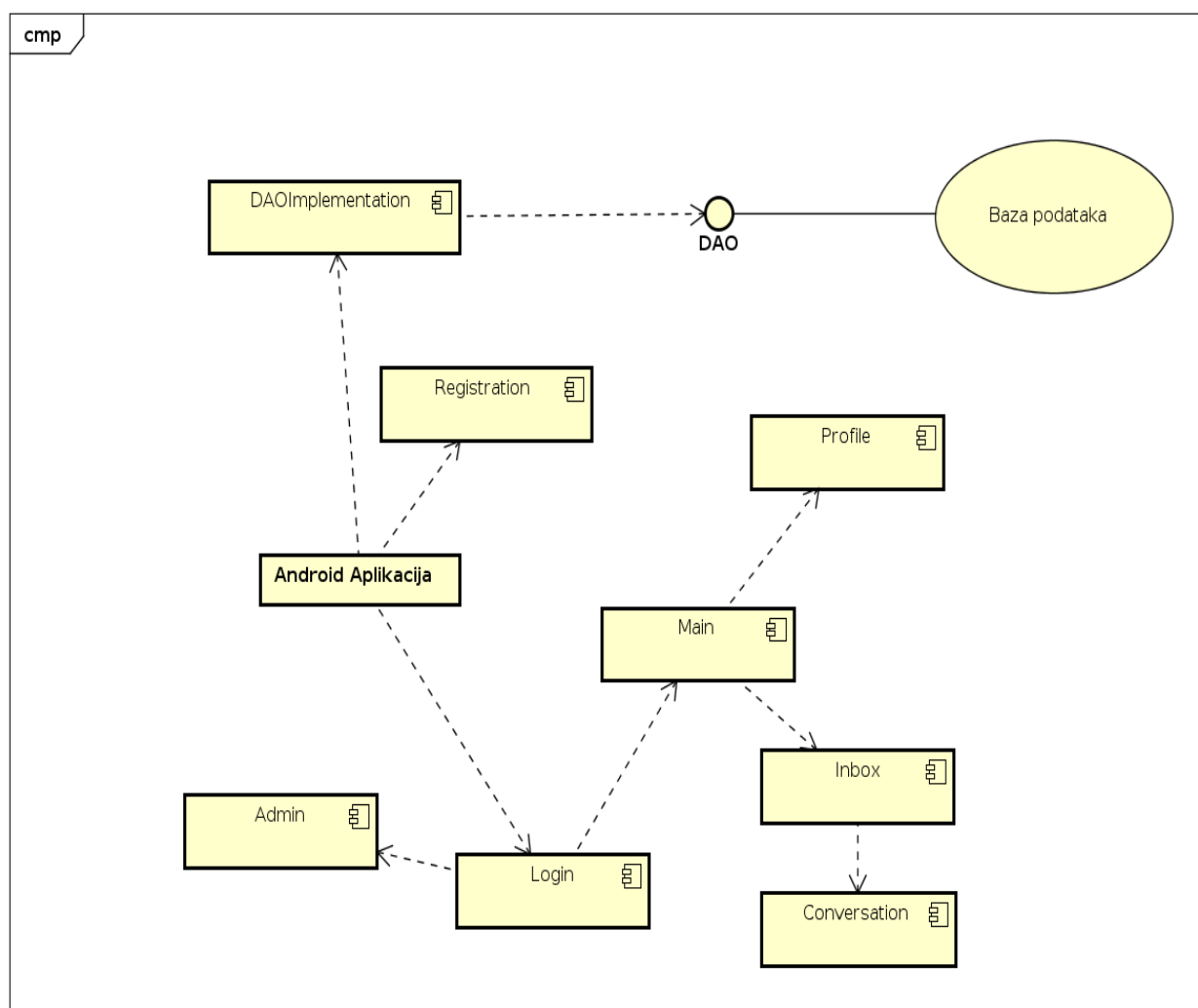


Slika 6.8 Dijagram aktivnosti

Dijagram komponenti

Sustav se sastoji od komponenti od kojih je svaka zaslužna za jedan pogled je vidljiv korisniku na Android uređaju. Komponente su sljedeće:

- **Registration:** Zadužen za obavljanje registracije novog korisnika
- **Login:** Prijava postojećeg korisnika ili administratora
- **Admin:** Administratorsko sučelje za administratorske poslove
- **Main:** Osnovni izgled aplikacije nakon prijave sa izbornikom
- **Profile:** Korisnički profil sa podacima i kontaktom
- **Inbox:** Popis primljenih i poslanih poruka korisnika s ostalim korisnicima
- **Conversation:** Razgovor s određenim korisnikom
- **DAOImplementation:** Zadužen za komunikaciju sa serverom i bazom podataka

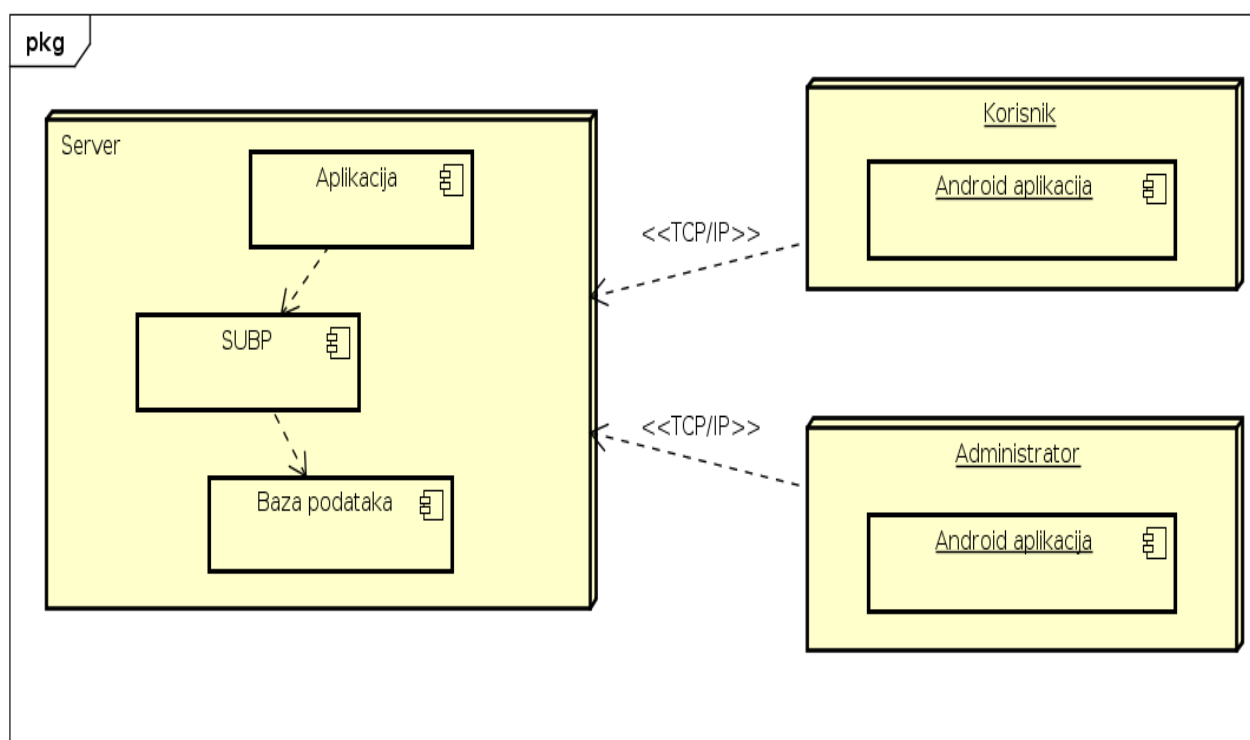


Slika 6.9 Dijagram komponenti

7. Implementacija i korisničko sučelje

Dijagram razmještaja

Dijagram razmještaja prikazuje fizički razmještaj našega sustava. Korisnikova jedina komponenta je Android aplikacija. Preko korisničke Android aplikacije se u pozadini pristupa putem TCP/IP veze na server. Na istom principu radi i administratorska verzija aplikacije. Server je konstantno upogonjen te prima upite na svoju internu aplikaciju koja preko SUBP-a komunicira s bazom podataka te obavlja upite koji su poslani od korisnika ili administratora.



Slika 7.1 Dijagram razmještaja

Korištene tehnologije i alati

Sustav za upravljanje izvornim kodom

- GitLab mrežni repozitorij
- Git, verzija 2
- GitHub Desktop grafičko sučelje, verzija 1
- GitKraken grafičko sučelje, verzija 3

Poslužitelj

- Linux-based Ubuntu Server, verzija 16.04
- Java Server SE JRE, verzija 8
- Java SE JRE, verzija 8
- Java SE JDK, verzija 8
- Eclipse razvojno sučelje, verzija Neon 2

Baza podataka

- PostgreSQL baza podataka, verzija 10
- DBeaver grafičko sučelje, verzija 4
- pgAdmin 4 grafičko sučelje, verzija 2

Android mobilna aplikacija

- Android Studio razvojno sučelje, verzija 3
- Android SDK:
 - minimalna podržana verzija: 4.4 KitKat (API level 19)
 - verzija kompiliranja: 8 Oreo (API level 26)
- Google Places API verzija 11.8.0

Isječak programskog koda vezan za temeljnu funkcionalnost sustava

Server

Klasa server sadrži metodu main koja se vrti u beskonačnoj petlji, te čeka korisničke konekcije. Svaka konekcija dobiva vlastitu dretvu za obavljanje posla, dok glavna dretva samo čeka na nove konekcije.

```
public class Server {
    private static final int PORT=8080;
    private static HashMap<String, ShellCommand> commands;
    static {
        commands=new HashMap<>();
        ShellCommand[] cc= {
            new Help(),
            new Terminate(),
            new RegisterUser(),
            new LoginUser(),
            new DeleteUser(),
            new RetrieveUserProfile(),
            new CreateComment(),
            new CreateConversation(),
            new CreateProduct(),
            new CreateProductCategory(),
            new EditComment(),
            new DeleteComment(),
            new SendMessage(),
            new RetrieveMessages(),
            new CreateReport(),
            new EditProduct(),
            new DeleteProduct(),
            new RetrieveProductOffers(),
            new DeleteProductCategory(),
            new RetrieveProductCategories(),
            new ConfirmProductCategory(),
            new GetConversations(),
            new CreateShipmentOffer(),
            new DeleteShipmentOffer(),
            new CreateVehicle(),
            new RetrieveVehicles(),
            new DeleteVehicle(),
            new RetrieveShipmentOffers(),
            new GetAdminTasks(),
            new RetrieveProductOffersByVendor()
        };
        for (ShellCommand shellCommand : cc) {
            commands.put(shellCommand.getCommandName(), shellCommand);
        }
    }

    public static void main(String[] args) throws IOException {
        ServerSocket socket=new ServerSocket(PORT);
        socket.setSoTimeout(1000);
        BlockingQueue<Runnable> workingThreadQueue = new
        ArrayBlockingQueue<Runnable>(20);
        ExecutorService threadPool =new ThreadPoolExecutor(4, 8, 1000,
        TimeUnit.MILLISECONDS, workingThreadQueue);

        while(true) {
            System.out.println("Waiting");
            ClientWorkerShell worker=null;
```

```

        try {
            EnvironmentImpl impl=new
EnvironmentImpl(socket.accept(),commands);
            worker= new ClientWorkerShell(impl);
            System.out.println("conenction established");
            threadPool.execute(worker);
        }
        catch (Exception e) {
            System.out.println("No Connection");
        }
    }
}

```

Ispis kôda 7.2 Server

ClientWorkerShell

Klasa ClientWorkerShell implementira sučelje runnable jer definira standard obavljanja posla za svaku konekciju. Dretva čeka na korisnički zahtjev, te ovisno o zahtjevu modificira bazu podataka ili vraća neke podatke iz baze. Za komunikaciju sa korisnikom koristi klasu EnvironmentImpl.

```

public class ClientWorkerShell implements Runnable {
    private EnvironmentImpl environment;

    public ClientWorkerShell(EnvironmentImpl environment) {
        this.environment=environment;
    }

    @Override
    public void run() {
        ShellCommand command;
        while(true) {
            String inputLine=environment.getText().trim();
            if (inputLine.equals("null")||inputLine.equalsIgnoreCase("fail")) {
                environment.close();
                return;
            }
            JSONObject input=new JSONObject(inputLine);
            String inputCommand=input.getString("command").toUpperCase();
            command=environment.getCommand(inputCommand);
            if (command==null) {
                environment.sendText("Unsupported command.");
                continue;
            }
            if (!
command.execute(environment,input).toString().equalsIgnoreCase("Continue")) {
                break;
            }
        }
    }
}

```

Ispis kôda 7.3 ClientWorkerShell

EnvironmentImpl

Klasa EnvironmentImpl se instancira za svaku novu korisničku konekciju na server. Nasljeđivanjem sučelja Environment se osigurava da klasa environment sadrži sve potrebne metode za komunikaciju sa korisnikom. Klasa pomoću predane socket konekcije instancira komunikacijske kanale za slanje podataka od servera prema korisniku. Klasa također tijekom nastajanja se spaja na bazu podataka tako da svaki korisnik ima vlastitu konekciju na bazu podataka.

```
public class EnvironmentImpl implements Environment {
    private static final String FAIL="fail";
    private Socket accessPoint=null;
    private PrintWriter writeTo=null;
    private BufferedReader readFrom=null;
    private DataOutputStream writeToByte=null;
    private DataInputStream readFromByte=null;
    private StringBuilder buffer=new StringBuilder();
    private HashMap<String, ShellCommand> commands;
    private Connection connection=null;

    public EnvironmentImpl(Socket socket, HashMap<String, ShellCommand> commands) {
        accessPoint=socket;
        this.commands=commands;
        initializeStreams();
        createDatabaseConnection();
    }
    private void createDatabaseConnection() {
        try {
            Class.forName("org.postgresql.Driver");
            connection = DriverManager
                .getConnection("jdbc:postgresql://165.227.175.217:5432/f
anglabd",
                                "development",
                                "vladimirPutin");
        } catch (Exception e) {
            System.out.println("Database can't open");
            System.exit(0);
        }
    }
    @Override
    public void sendText(String text) {
        writeTo.println(text);
    }
    @Override
    public String getText() {
        try {
            buffer.append(readFrom.readLine()+"\n");
            while(readFrom.ready()) {
                buffer.append(readFrom.readLine()+"\n");
            }
        } catch (IOException e) {
            System.out.println("Unable to get text from client");
            return FAIL;
        }
        String text=buffer.toString();
        buffer.setLength(0);
        return text;
    }
    @Override
    public boolean sendImage(String path) {
        byte[] imageInBytes;
        try {
            imageInBytes=convertImageToByte(path);
            writeToByte.writeInt(imageInBytes.length);
        }
    }
}
```

```

        writeToByte.write(imageInBytes);
    } catch (IOException e) {
        System.out.println("Unable to read picture");
        return false;
    }
    return true;
}
@Override
public BufferedImage getImage() {
    BufferedImage fromByte=null;
    try {
        int len=readFromByte.readInt();
        if (len>0) {
            byte[] image=new byte[len];
            readFromByte.readFully(image,0,image.length);
            InputStream input=new ByteArrayInputStream(image);
            fromByte=ImageIO.read(input);
        }
    } catch (IOException e) {
        System.out.println("Reading image from stream error.");
        return null;
    }
    return fromByte;
}
@Override
public void initializeStreams() {
    try {
        writeTo=new PrintWriter(accessPoint.getOutputStream(),true);
        readFrom=new BufferedReader(new
InputStreamReader(accessPoint.getInputStream(), StandardCharsets.UTF_8));
        writeToByte=new DataOutputStream(accessPoint.getOutputStream());
        readFromByte=new DataInputStream(accessPoint.getInputStream());
    } catch (IOException e) {
        System.out.println("Couldn't initialize IO streams. Terminating
connection");
        System.exit(-1);
    }
}
@Override
public void close() {
    try {
        writeTo.close();
        readFrom.close();
        readFromByte.close();
        writeToByte.close();
        accessPoint.close();
    } catch (IOException Ignorable) {}
}
private byte[] convertImageToByte(String path) throws IOException {
    BufferedImage original=ImageIO.read(this.getClass().getResource(path));
    ByteArrayOutputStream baos = new ByteArrayOutputStream();
    ImageIO.write(original, "jpg", baos);
    baos.flush();
    byte[] array=baos.toByteArray();
    baos.close();
    return array;
}
public HashMap<String, ShellCommand> getCommands() {
    return commands;
}
@Override
public ShellCommand getCommand(String command) {
    if (command!=null && commands.containsKey(command)) {
        return commands.get(command.toUpperCase());
    }
    return null;
}
@Override
public Connection getDatabase() {
    return connection;
}
}

```

Ispis kôda 7.4 EnvironmentImpl

Ispitivanje programskog rješenja

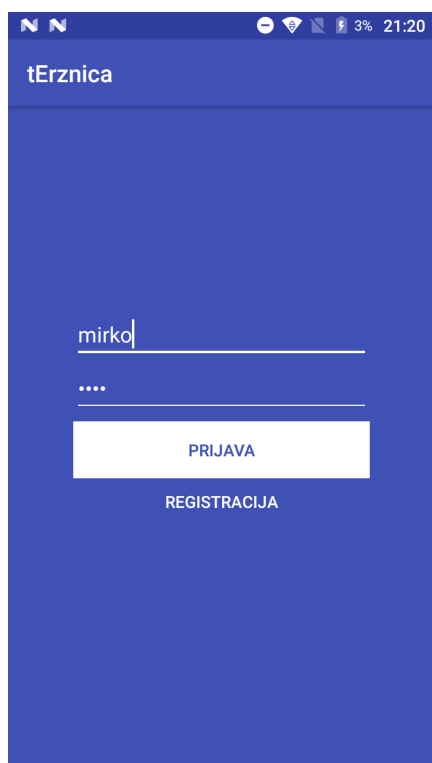
Ispitujemo osnovne funkcionalnosti i rubne slučajeve svih razina sustava za sve vrste korisnika kako bi mogli spriječiti rušenje sustava ili neautorizirani pristup korisnika.

Slučaj 1:

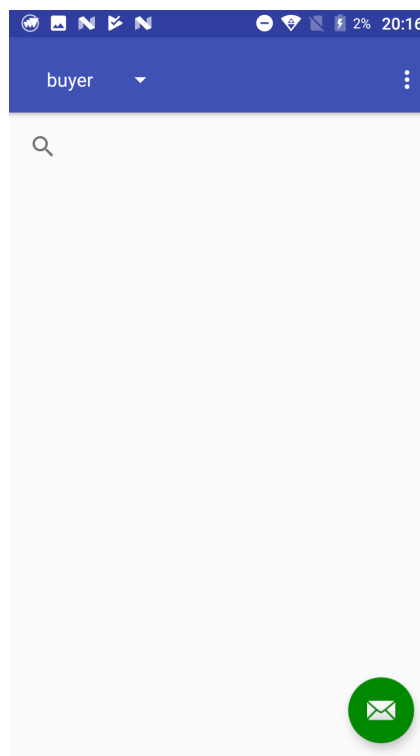
Opis: Korisnik se prijavljuje na sustav.

Očekivani rezultat: Ukoliko su korisničko ime i lozinka ispravni, korisnik je prijavljen na sustav. Dobiveni rezultat: U slučaju točno upisanih podataka, korisnik je prijavljen u sustav dok je pri upisivanju krivih podataka korisniku ispisana poruka o grešci.

Dobiveni rezultat: Za upisivanje točnih podataka korisnik je uspješno prijavljen u sustav.



Slika 7.5 Slučaj 1a



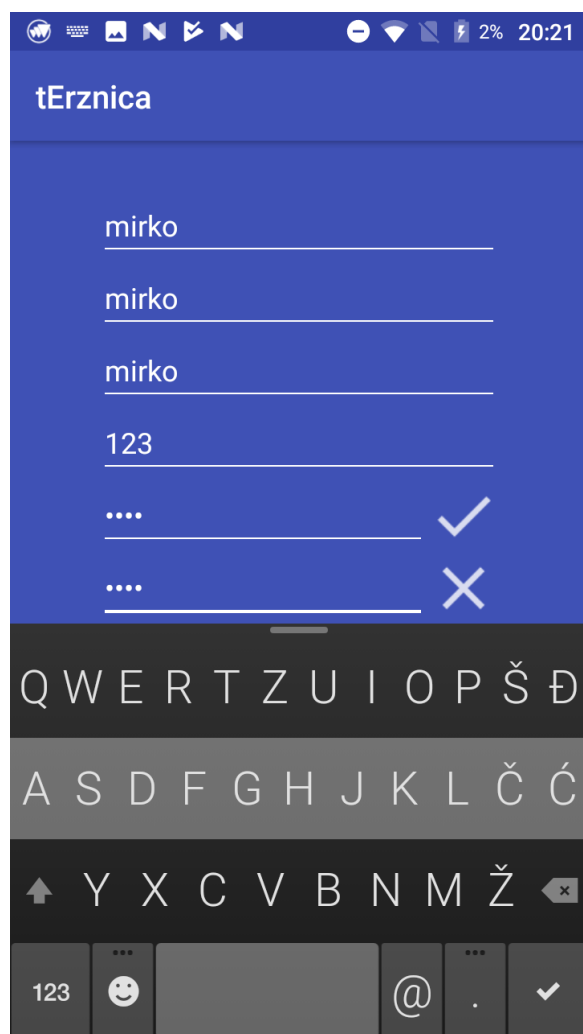
Slika 7.6 Slučaj 1b

Slučaj 2:

Opis: Korisnik prilikom registracije unosi lozinku koju mora onda ponovno unijeti.

Očekivani rezultat: Sustav signalizira korisniku da nije uspješno unio lozinku drugi put.

Dobiveni rezultat: Sustav uistinu sugerira korisniku da nije dobro unio lozinku.



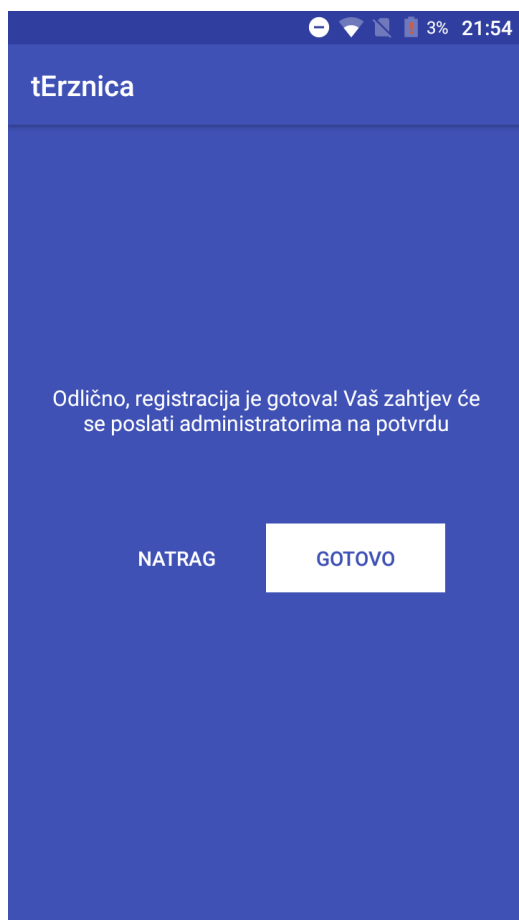
Slika 7.7 Slučaj 2

Slučaj 3:

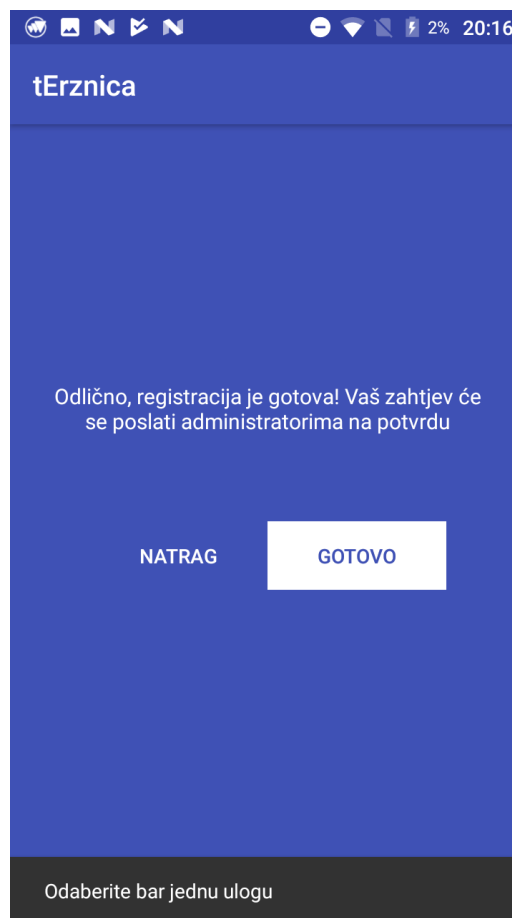
Opis: Korisnik se registrira u sustav i pritom ne odabire niti jednu ulogu.

Očekivani rezultat: Sustav spriječava korisnika da se prijavi.

Dobiveni rezultat: Sustav je spriječio korisnika da se prijavi i upozorio ga da nije odabrao ulogu.



Slika 7.8 Slučaj 3a



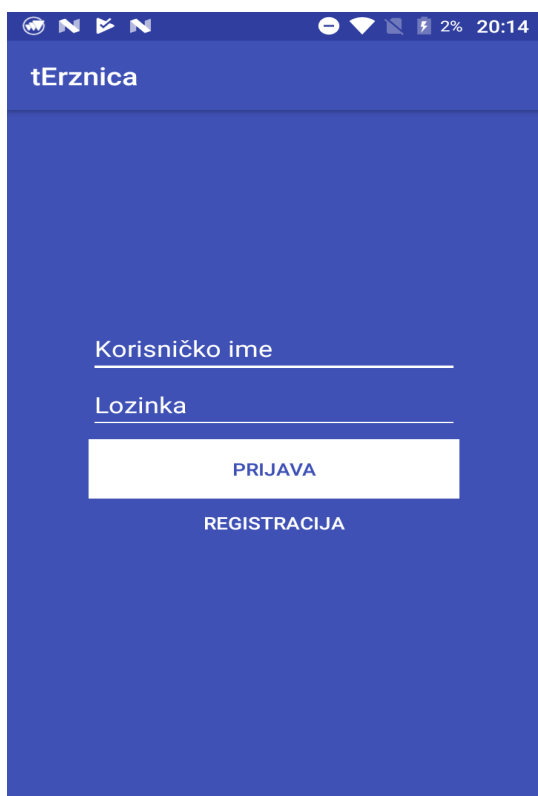
Slika 7.9 Slučaj 3b

Slučaj 4:

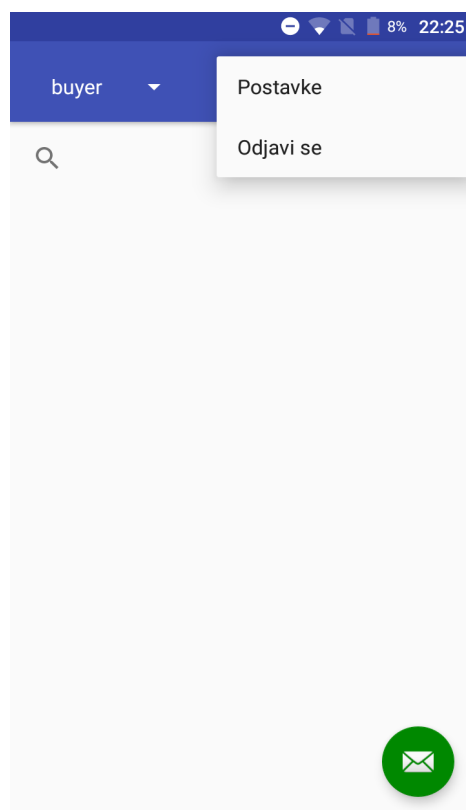
Opis: Korisnik odabire opciju odjava i sustav ga stavlja na početni ekran u kojem se traži unos podataka.

Očekivani rezultat: Nakon pritiska tipke odjava korisnik odjavljen iz sustava.

Dobiveni rezultat: Dobiven je očekivani rezultat.



Slika 7.10 Slučaj 4a



Slika 7.11 Slučaj 4b

Upute za instalaciju

Instalacija poslužitelja

Poslužitelj je konfiguriran preko DigitalOcean-a. Za operacijski sustav izabran je Linux Ubuntu server x64 koji je ponuđen kao standardni operacijski sustav na stranici DigitalOceana.

Pračene su upute napisane na DigitalOceanu, te je sam proces instalacije automatiziran od strane DigitalOcean-a za minimalno opterećivanje korisnika.

Instalacija baze podataka

Potrebno je pokrenuti naredbeni redak („terminal“), te izvršiti naredbu: „**sudo apt-get update**“, a zatim i naredbu „**sudo apt-get install postgresql**“.

Nakon instalacije PostgreSQL-a, potrebno je kreirati novog korisnika. Upišimo naredbu „**sudo -u postgres createuser --interactive**“, te zatim traženo ime novog korisnika te „y“ da korisnik ima vlasnička prava. Zatim kreirajmo korisnika istog imena i unutar operacijskog sustava naredbom „**sudo adduser [ime korisnika]**“.

Sada se možemo ulogirati kao novi korisnik u sustav upravljanja bazom podataka naredbom „**sudo -[ime korisnika] psql postgres**“.

Novu bazu podataka kreirajmo SQL naredbom „**CREATE DATABASE [ime nove baze];**“.

Potrebne tablice možemo kreirati izravnim pisanje SQL sintakse, no preporučujemo da se to izvrši korištenje grafičkog sučelja pgAdmin 4 verzije 2.

Da bismo se mogli spojiti s vlastitog računala na novokreiranu bazu podataka, potrebno je modificirati konfiguraciju sustava upravljanja bazom podataka na poslužitelju.

Potrebno je otkriti lokaciju konfiguracijske datoteke PSQL-a izvršenjem naredbe „**SHOW config_file;**“ unutar PSQL-a. Zatim u datoteci „**postgresql.conf**“ za vrijednost atributa „**listen_addresses**“ postaviti „*“, tj. „**listen_addresses = ***“. Zatim restartajmo PSQL servis naredbom „**/etc/init.d/postgresql [stop/start]**“.

Zatim u istom direktoriju na kraj datoteke „**pg_hba.conf**“ dodajmo dvije retka: „**host all all 0.0.0.0/0 md5**“, te „**host all all ::/0 md5**“. Restartajmo PSQL servis.

Postavljanje i pokretanje poslužiteljskog dijela aplikacije

Poslužitelj je napisan u programskom jeziku Java. Prvo je potrebno exportati kod u Runnable JAR file.

File-> Export -> Runnable JAR file, te se zatim izabere koja konfiguracija se želi exportati, te na koju lokaciju i za kraj se pritisne Finish.

Zatim se jar file prebaci na server, te se na serveru pokrene u komandnoj liniji naredbom „**java -jar name.jar**“.

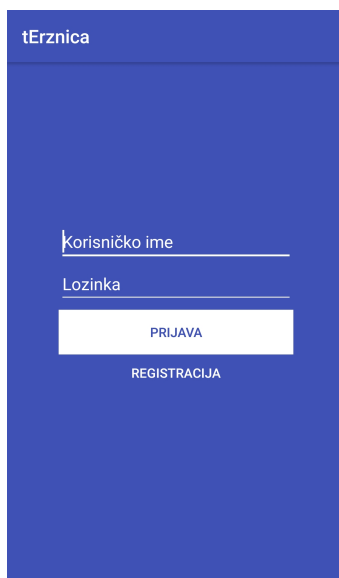
Za zaustavljanje servera u komandnoj liniji dovoljno je samo pritisnuti Ctrl + C.

Preuzimanje i instalacija Android aplikacije na Android uređaj

Aplikaciju je potrebno preuzeti sa Amazon App trgovine i instalirati na Android uređaj.

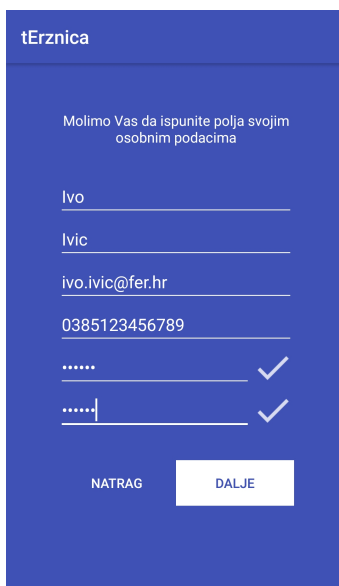
Korisničke upute

Nakon završetka instalacije i pokretanja aplikacije, korisniku se otvara prozor za prijavu i registraciju kao na slici 7.12.

The screenshot shows a blue-themed login and registration window for 'tErznica'. It features two input fields: 'Korisničko ime' (Username) and 'Lozinka' (Password). Below these fields are two buttons: 'PRIJAVA' (Login) and 'REGISTRACIJA' (Registration). The window has a title bar with the text 'tErznica'.

Slika 7.12 Početna stranica

Kako je korisniku ovo prvo korištenje aplikacije, još nije prijavljen u sustav, zato mora odabrati opciju 'Registracija'. Tada mu se otvara novi prozor kao na slici 7.13 gdje se od njega traži da unese svoje osobne podatke.

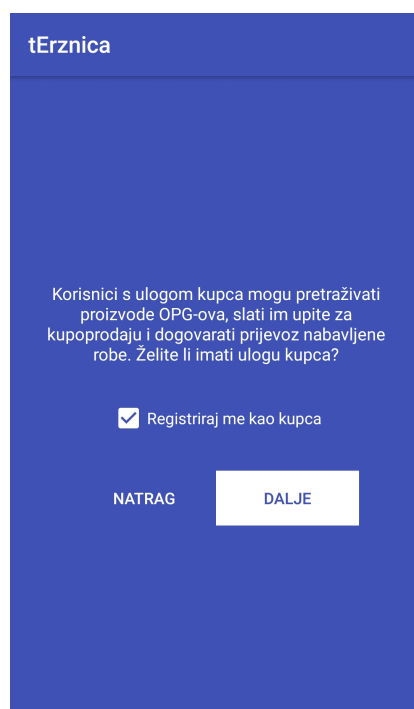
The screenshot shows a registration window for 'tErznica'. It contains a message: 'Molimo Vas da ispunite polja svojim osobnim podacima' (Please fill in the fields with your personal data). Below this are several input fields: 'Ivo', 'Ivic', 'ivo.ivic@fer.hr', and '0385123456789'. There are two password fields, each with a checkmark icon indicating password strength. At the bottom, there are two buttons: 'NATRAG' (Back) and 'DALJE' (Next).

Slika 7.13 Registracija

Nakon potvrde svojih podataka, otvaraju se novi prozori na kojima se korisniku nudi mogućnost da odabere uloge koje će imati:

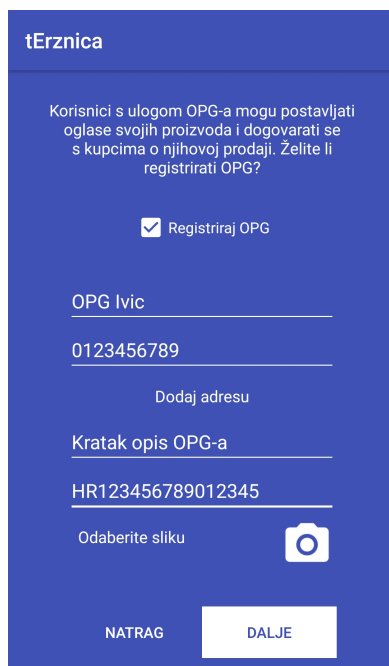
- Kupac
- OPG
- Prijevoznik

Svaku ulogu koju želi imati, korisnik potvrđuje prilikom registracije. Na prvom prozoru, kao što je prikazano na slici 7.14, korisniku je opisana uloga kupca i koje mogućnosti ima. Da bi dobio ulogu kupca, korisnik mora označiti *check box* kraj kojeg piše 'Registriraj me kao kupca'.



Slika 7.14 Registracija kupca

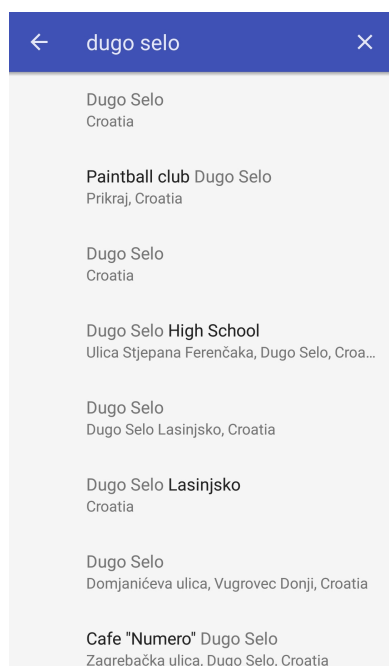
Sljedeća uloga koja se nudi je uloga OPG-a. Korisnik koji želi prodavati svoje proizvode putem ove aplikacije mora se registrirati kao OPG. To će učiniti tako da označi *check box* kraj kojeg piše 'Registriraj OPG'. Time se otvara padajući izbornik, kao na slici 7.15, u kojem se traži od korisnika da unese podatke o svom OPG-u. Odabirom opcije 'Dodaj adresu' otvara se novi pretraživač, kao na slici 7.16, koji je povezan s Google Maps i omogućuje korisniku lakši pronalazak svoje lokacije. Dodatno, odabirom opcije 'Odaberite sliku' korisniku se nudi mogućnost da postavi sliku svog OPG-a koja će biti vidljiva na njegovom profilu.



The screenshot shows a registration form titled 'tErznica'. It contains the following elements:

- A header bar with the text 'tErznica'.
- A paragraph of text: 'Korisnici s ulogom OPG-a mogu postavljati oglase svojih proizvoda i dogovarati se s kupcima o njihovoj prodaji. Zelite li registrirati OPG?'.
- A checkbox labeled 'Registriraj OPG' which is checked.
- A text input field containing 'OPG Ivic'.
- A text input field containing '0123456789'.
- A text input field with the placeholder 'Dodaj adresu'.
- A text input field containing 'Kratak opis OPG-a'.
- A text input field containing 'HR123456789012345'.
- A text input field with the placeholder 'Odaberite sliku' and a camera icon.
- Two buttons at the bottom: 'NATRAG' and 'DALJE'.

Slika 7.15 Registracija OPG-a



The screenshot shows a dropdown menu for selecting a location. The title bar at the top says 'dugo selo' with a back arrow on the left and a close 'X' on the right. The list of options includes:

- Dugo Selo
Croatia
- Paintball club Dugo Selo
Prikraj, Croatia
- Dugo Selo
Croatia
- Dugo Selo High School
Ulica Stjepana Ferenčaka, Dugo Selo, Croa...
- Dugo Selo
Dugo Selo Lasinjsko, Croatia
- Dugo Selo Lasinjsko
Croatia
- Dugo Selo
Domjanićeva ulica, Vugrovec Donji, Croatia
- Cafe "Numero" Dugo Selo
Zagrebačka ulica, Dugo Selo, Croatia

Slika 7.16 Odabir lokacije

Sljedeća uloga je uloga prijevoznika. Na slici 7.17 opisana je uloga prijevoznika i njegove mogućnosti. Korisnik se može registrirati kao prijevoznik tako da označi *check box* kraj kojeg piše 'Registriraj me kao prijevoznika'. Tada mu se otvara padajući izbornik gdje može dodati opis usluge i vozila kojima će obavljati prijevoz. Odabirom opcije 'Dodaj vozilo' otvara se pomoćni prozor kao na slici 7.18, gdje korisnik unosi podatke o vozilu.

tErznica

Korisnici s ulogom prijevoznika mogu odgovarati na zahtjeve kupaca za prijevozom kupljenih dobara i dogovarati se s njima za prijevoz. Želite li se registrirati kao prijevoznik?

☒ Registriraj me kao prijevoznika

Opis

+ Dodaj vozilo

NATRAG DALJE

Slika 7.17 Registracija prijevoznika

tErznica

OG1234HR

Volkswagen Transporter

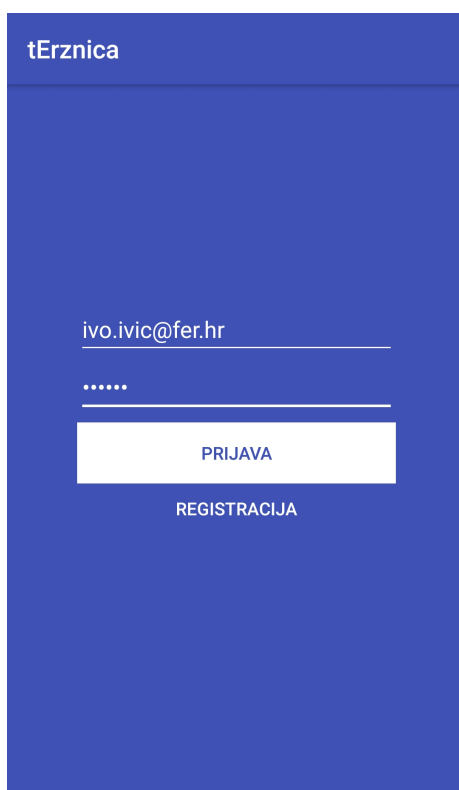
Kategorije

- A
- B1
- B
- C1
- C
- D1
- D
- BE
- C1E
- CE
- D1E
- DE
- F, G

NATRAG DALJE

Slika 7.18 Dodavanje vozila

Nakon toga potvrđuje svoju registraciju i spreman je korištenje sustava. Upisom e-maila kao korisničko ime i lozinke prijavljuje se na sustav.

The image shows a login interface for a website called 'tErznica'. The header is a solid blue bar with the text 'tErznica' in white. Below the header, the background is a lighter blue. In the center, there is a white rectangular area containing the login form. The form has two input fields: the first one contains the email address 'ivo.ivic@fer.hr' and the second one contains six dots '.....'. Below these fields are two buttons: a white button with the text 'PRIJAVA' in blue, and a blue button with the text 'REGISTRACIJA' in white.

Slika 7.19 Prijava

Ako je odabrao ulogu OPG-a, korisnik može objavljivati ponude svojih proizvoda. Uloga kupca omogućuje korisniku pretraživanje željenih proizvoda. Kada pronađe željeni proizvod kod odabranog prodavača, može putem poruka stupiti s njim u kontakt te se detaljnije dogovoriti oko kupovine. Nakon obavljanja transakcije, kupac može tražiti prijevoz robe tako što stvori objavu za prijevoz. U njoj specificira polazište, odredište i opis proizvoda koji se treba dostaviti. Korisnici s ulogom prijevoznika pretražuju te objave te se u dogovoru s kupcem dogovaraju za cijenu prijevoza. Kupac i OPG također mogu uređivati i brisati svoje objave.

8. Zaključak i budući rad

U dosadašnjem radu smo napravili opis projektnog zadatka, dijagrame obrazaca uporabe, sekvencijske dijagrame, opis arhitekture sustava, dijagram razreda te dijagram objekata.

Za budući rad je potrebna dobra organizacija oko implementacije samog projekta kako bi bili što efikasniji.

U drugoj fazi projekta podijeljen je posao oko implementacije same aplikacije, podešavanja serverskog dijela, izrade serverske aplikacije i postavljanja baze podataka. Kada se implementacija bližila kraju podijeljen je posao oko korekcije i dopune dokumentacije. Dodano je nekoliko dijagrama te kako se instalira i kako funkcionira sama aplikacija.

Što se tiče budućnosti i nadogradnje same aplikacije, mogućnosti su široke. Aplikacija ima potencijala da postane korištena u stvarnom svijetu jer bi se našla publika za nju. Također Android aplikacija za domaće proizvode još ne postoji prema našim saznanjima.

9. Popis literature

- ¹ Oblikovanje programske potpore, FER ZEMRIS, <http://www.fer.hr/predmet/opp>
- ² Oblikovanje programske potpore, FER ZEMRIS,
<http://www.zemris.fer.hr/predmeti/opp>
- ³ Astah Community, <http://astah.net/editions/community/>

10.Dodatak A: Indeks (slika, dijagrama, tablica, ispisa kôda)

- Slika 4.1 Dijagram obrazaca uporabe
- Slika 4.2 Dijagram obrazaca uporabe
- Slika 4.3 Sekvencijski dijagram UC1
- Slika 4.4 Sekvencijski dijagram UC2
- Slika 4.5 Sekvencijski dijagram UC3
- Slika 4.6 Sekvencijski dijagram UC4
- Slika 4.7 Sekvencijski dijagram UC5
- Slika 4.8 Sekvencijski dijagram UC6
- Slika 4.9 Sekvencijski dijagram UC7
- Slika 4.10 Sekvencijski dijagram UC8
- Slika 4.11 Sekvencijski dijagram UC9
- Slika 4.12 Sekvencijski dijagram UC10
- Slika 4.13 Sekvencijski dijagram UC11
- Slika 4.14 Sekvencijski dijagram UC12
- Slika 4.15 Sekvencijski dijagram UC13
- Slika 4.16 Sekvencijski dijagram UC14
- Slika 4.17 Sekvencijski dijagram UC15
- Slika 4.18 Sekvencijski dijagram UC16
- Slika 4.19 Sekvencijski dijagram UC17
- Slika 4.20 Sekvencijski dijagram UC18
- Slika 4.21 Sekvencijski dijagram UC19
- Slika 4.22 Sekvencijski dijagram UC20
- Slika 4.23 Sekvencijski dijagram UC21
- Slika 6.1 Skica sustava
- Slika 6.2 ER model baze podataka
- Slika 6.3 Poslužiteljski UML dijagram razreda
- Slika 6.4 Klijentski UML dijagram razreda
- Slika 6.5 Dijagram objekata
- Slika 6.6 Komunikacijski dijagram
- Slika 6.7 Dijagram stanja
- Slika 6.8 Dijagram aktivnosti

- Slika 6.9 Dijagram komponenti
- Slika 7.1 Dijagram razmještaja
- Ispis kôda 7.2 Server
- Ispis kôda 7.3 ClientWorkerShell
- Ispis kôda 7.4 EnvironmentImpl
- Slika 7.5 Slučaj 1a
- Slika 7.6 Slučaj 1b
- Slika 7.7 Slučaj 2
- Slika 7.8 Slučaj 3a
- Slika 7.9 Slučaj 3b
- Slika 7.10 Slučaj 4a
- Slika 7.11 Slučaj 4b
- Slika 7.12 Početna stranica
- Slika 7.13 Registracija
- Slika 7.14 Registracija kupca
- Slika 7.15 Registracija OPG-a
- Slika 7.16 Odabir lokacije
- Slika 7.17 Registracija prijevoznika
- Slika 7.18 Dodavanje vozila
- Slika 7.19 Prijava

11.Dodatak B: Dnevnik sastajanja

1.Sastanak - 5.10.2017.

Prisutni: Benčić, Janjić, Markota, Maslovar

Sažetak: Upoznavanje i dogovaranje osnovnih stvari oko projekta.

Ishod: Dogovorena platforma(Android) i rad bez Erasmus studenata.

2.Sastanak - 9.10.2017.

Prisutni: Benčić, Harmicar, Janjić, Markota, Maslovar, Mihaljčuk

Sažetak: Dogovor oko ideje za izradu aplikaciju.

Ishod: Izabrana ideja "Domaći proizvodi za sve"

3.Sastanak - 13.10.2017.

Prisutni: Benčić, Maslovar

Sažetak: Izmjene i dopune projektnog zadatka

Ishod: Ispravljene nedorečenosti u projektnom zadatku

4.Sastanak - 17.10.2017.

Prisutni: Benčić, Harmicar, Janjić, Maslovar

Sažetak: Izmjene i dopune projektnog zadatka

Ishod: Ispravljene nedorečenosti u projektnom zadatku

5.Sastanak - 25.10.2017.

Prisutni: Benčić, Harmicar, Janjić, Markota, Maslovar, Mihaljčuk

Sažetak: Dogovor oko podjele posla

Ishod: Proučavanje dodijeljenog dijela posla

6.Sastanak - 27.10.2017.

Prisutni: Benčić, Harmicar, Janjić, Markota, Maslovar, Mihaljčuk
Sažetak: Crtanje korisničkih sučelja i dogovor oko obrazaca uporabe
Ishod: Jasnija slika projekta i intenzivniji rad na dokumentaciji

7.Sastanak - 7.11.2017.

Prisutni: Benčić, Harmicar, Janjić, Markota, Maslovar, Mihaljčuk
Sažetak: Dogovor oko daljnje podjele posla
Ishod: Intenzivniji rad na dovršavanju dokumentacije

8.Sastanak - 11.12.2017.

Prisutni: Benčić, Harmicar, Janjić, Markota, Maslovar, Mihaljčuk
Sažetak: Dogovor oko implementacije aplikacije
Ishod: Podjeljen posao implementacije

9.Sastanak - 8.1.2018.

Prisutni: Benčić, Harmicar, Janjić, Markota, Maslovar, Mihaljčuk
Sažetak: Priprema alfa inačice aplikacije
Ishod: Podjela preostalog posla

10.Sastanak - 10.1.2018.

Prisutni: Benčić, Harmicar, Janjić, Markota, Maslovar, Mihaljčuk
Sažetak: Dogovor oko podjele posla i pisanja dokumentacije
Ishod: Intenzivniji rad na aplikaciji i pisanje dokumentacije

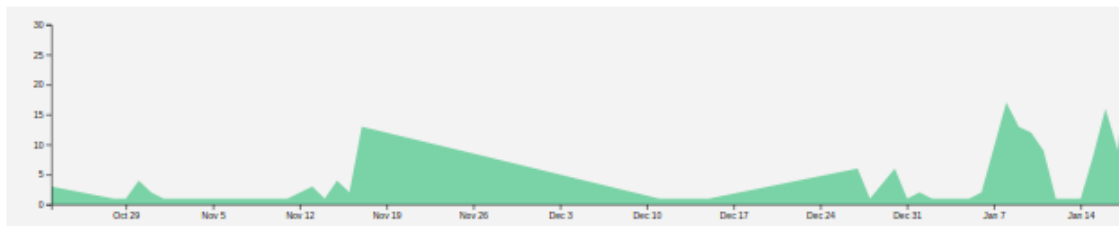
12.Dodatak C: Prikaz aktivnosti grupe

Popis aktivnosti	Članovi grupe (abecednim redom)					
	Ivan Janjić	Stjepan Benčić	Luka Harmicar	Domagoj Markota	Domagoj Maslovar	Marko Mihaljčuk
Upravljanje projektom	100%	-	-	-	-	-
Opis projektnog zadatka	-	50%	-	-	50%	-
Rječnik pojmova	-	-	-	-	-	100%
Opis funkcionalnih zahtjeva	5%	40%	40%	5%	5%	5%
Opis ostalih zahtjeva	-	-	-	-	-	100%
Arhitektura i dizajn sustava						
Svrha, opći prioriteti i skica sustava	100%	-	-	-	-	-
Dijagram razreda s opisom	-	-	20%	50%	20%	10%
Dijagram objekata	70%	-	-	30%	-	-
Ostali UML dijagrami	25%	-	-	-	-	75%
Implementacija i korisničko sučelje						
Dijagram razmještaja	100%	-	-	-	-	-
Korištene tehnologije i alati	-	-	-	100%	-	-
Isječak programskog kôda	-	-	100%	-	-	-
Ispitivanje programskog rješenja	-	100%	-	-	-	-
Upute za instalaciju	-	-	50%	50%	-	-
Korisničke upute	-	-	-	-	-	100%
Plan rada	100%	-	-	-	-	-
Pregled rada i stanje ostvarenja	100%	-	-	-	-	-
Zaključak i budući rad	100%	-	-	-	-	-
Popis literature	100%	-	-	-	-	-
Dodaci						
Indeks	100%	-	-	-	-	-
Dnevnik sastajanja	100%	-	-	-	-	-

Pregled pohrana kroz vrijeme trajanja projekta:

October 23, 2017 – January 18, 2018

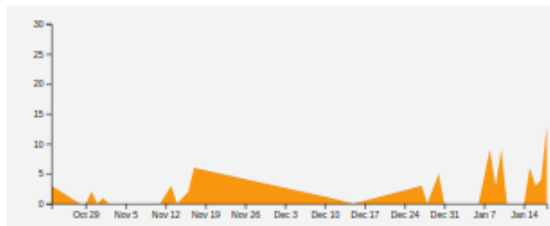
Commits to master, excluding merge commits. Limited to 6,000 commits.



ijanjić

74 commits

ivan.janjić@fer.hr



Maslovar

27 commits

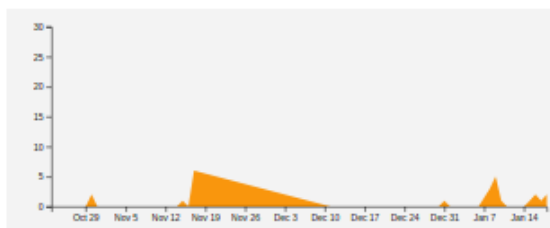
domagoj.maslovar@fer.hr



Domagoj Markota

25 commits

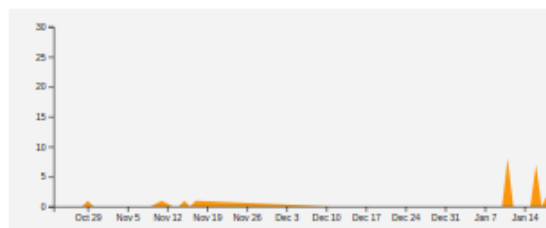
domagoj.markota@fer.hr



Marko Mihaljcuk

21 commits

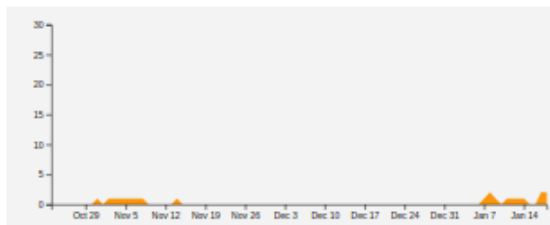
marko.mihaljcuk@fer.hr



stjepanB

15 commits

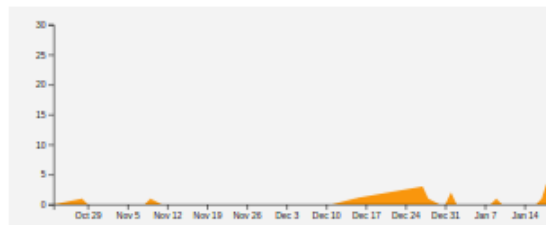
stjepan.bencic@fer.hr



Luka

13 commits

lu48838@fer.hr



13.Dodatak D: Plan rada / Pregled rada i stanje ostvarenja

Plan rada za rev. 2 je ravnomjerno raspodijeliti posao, implementirati sustav na Android platformi te nadopuniti dokumentaciju potrebnim informacijama.

Projekt je uspješno izvršen u rev. 2, implementirana je zamišljena aplikacije te je za nju napisana adekvatna dokumentacija.