

Oblikovanje programske potpore

Ak. god. 2016./2017.

Dokumentacija, Rev. 1

Grupa: *Kavon*

Voditelj: *dr. sc. Miljenko Krhen*

Datum predaje: *27.10.2017.*

Nastavnik: *Miljenko Krhen*

Sadržaj

1. Dnevnik promjena dokumentacije.....	3
2. Opis projektnog zadatka	5
2.1 Projektni prijedlog.....	5
2.2. Zahtjevi za informacijski sustav	5
2.2.1. Korisnici informacijskog sustava i njihove ovlasti/mogućnosti.....	5
2.2.2. Posebne funkcionalnosti sustava.....	5
2.2.3. Ostale funkcionalnosti sustava	7
2.3. Cilj i dugoročna korist rješenja	7
2.4. Potencijalni kupci i korisnici	7
3. Pojmovnik.....	8
4. Funkcijski zahtjevi.....	9
5. Ostali zahtjevi	35
6. Arhitektura i dizajn sustava	36
7. Implementacija i korisničko sučelje	45
8. Zaključak i budući rad	46
Popis literature	47
Dodatak A: Index.....	48
Dodatak B: Dnevnik sastanaka.....	49
Dodatak C: Prikaz aktivnosti grupe.....	51
Dodatak D: Plan rada i ostvarenje.....	54

1. Dnevnik promjena dokumentacije

REV.	OPIS PROMJENA	AUTORI	DATUM
0.1	<ul style="list-style-type: none">• napravljen inicijalni predložak• dodan projektni zadatak	Modrušan	25.10.2017.
0.2	<ul style="list-style-type: none">• dodani početni obrasci upotrebe (Use Case)	Kravrščan	25.10.2017.
0.3	<ul style="list-style-type: none">• dodan dnevnik sastajanja• naknadno ažuriranje i uređivanje dnevnika sastajanja	Paulinović, Galić	26.10.2017.
0.4	<ul style="list-style-type: none">• dodan prijedlog modela baze podataka	Krmpotić-Đurđević	27.10.2017.
0.5	<ul style="list-style-type: none">• završeni obrasci upotrebe (Use Case)• dodani sekvencijski dijagrami	Kravrščan, Paulinović	5.11.2017.
0.6	<ul style="list-style-type: none">• prebacivanje, do sad napravljene, dokumentacije na Gitlab Wiki	Modrušan, Krmpotić-Đurđević, Galić	10.11.2017.
0.7	<ul style="list-style-type: none">• dovršeni svi sekvencijski dijagrami i opisi	Paulinović	14.11.2017.
0.8	<ul style="list-style-type: none">• arhitektura sustava (MVC) dorađen ER model baze podataka	Sodić, Krmpotić-Đurđević	15.11.2017.
1.0	<ul style="list-style-type: none">• kontrola dokumentacije• ispravljene pogreške i dodani detalji• konačna prva verzija	Krmpotić-Đurđević, Sodić, Galić, Paulinović, Modrušan, Kravrščan	17.11.2017.

REV.	OPIS PROMJENA	AUTORI	DATUM
	dokumentacije		

2. Opis projektnog zadatka

2.1 Projektni prijedlog

Tvrtka za čuvanje kućnih ljubimaca nudi sljedeće usluge:

- Dolazak/odlazak po psa na čuvanje
- Odlazak veterinaru
- Šetnje
- Čuvanje pasa
- Hranjenje mačaka
- Čuvanje egzota, glodavaca, dvojezubaca
- Davanje terapije
- Edukacija
- Rezanje noktiju

Za tvrtku je potrebno razviti učinkovit informacijski sustav ostvaren u obliku web aplikacije. Sustav treba zaposlenicima omogućiti lakšu i kvalitetniju organizaciju posla, a klijentima pružiti opciju online rezervacija i naručivanja usluga. Aplikacija uključuje i intranet koji koriste zaposlenici.

Radno vrijeme tvrtke nije fiksno. Prijava ljubimaca za usluge aktivna je uvijek, a oni se zaprimaju/zbrinjavaju ovisno o dostupnosti. Preuzimanje ljubimaca također se vrši u dogovorenim terminima.

2.2. Zahtjevi za informacijski sustav

2.2.1. Korisnici informacijskog sustava i njihove ovlasti/mogućnosti

Informacijski sustav razlikuje tri vrste korisnika: registrirani klijent, zaposlenik i administrator.

Registrirani klijent može rezervirati usluge slanjem zahtjeva. Sustav pamti njegove podatke i podatke o njegovim ljubimcima. Postupak i zahtjev za registraciju klijenata opisani su u odjeljku 2.2.1, a njegov pregled u odjeljku 2.2.2.

Zaposlenik može mijenjati određeni dio vlastitih informacija i svoju lozinku.

Registracija zaposlenika opisana je u odjeljku 2.2.5, a pregled i ovlasti u odjeljku 2.2.6.

Administrator sustava također je zaposlenik, ali s najvećom razinom autorizacije. Njegov pregled i ovlasti opisani su u odjeljku 2.2.8.

2.2.2. Posebne funkcionalnosti sustava

2.2.2.1. Registracija klijenta

Klijent prilikom registracije unosi vlastite podatke. Obavezni podaci su njegovo ime, prezime, OIB, adresa stanovanja i elektroničke pošte te broj telefona. Također se nudi i opcija unošenja pojedinosti o jednom ili više kućnih ljubimaca s ciljem ubrzanja korištenja aplikacije u budućnosti. Ovisno o ljubimcu, sustav može tražiti neke dodatne obavezne informacije. Uz sve obavezne i neobavezne podatke, klijent mora izabrati i lozinku.

Kao korisničko ime koristiti će se unesena adresa elektroničke pošte koja je jedinstvena za svakog klijenta.

2.2.2.2. Pregled registriranog klijenta

Klijent vidi sve svoje zahtjeve i njihove trenutne razine. Postupak zaprimanja, određivanja razina i obrade zahtjeva detaljnije je objašnjen u odjeljcima 2.2.3 i 2.2.7. Klijent može mijenjati dio svojih podataka i lozinku.

2.2.2.3. Prijava ljubimaca za usluge

Nakon spajanja na sustav, registrirani klijent ima mogućnost rezervacije željene usluge. Rezervacija usluge zahtjeva precizno određivanje njenog termina i vrste. Ovisno o odabranoj usluzi, nudi se i preferencijalni odabir zaposlenika kod kojeg klijent želi zbrinuti svog ljubimca. Registrirani klijenti ovdje imaju pristup svojim uobičajenim podacima. Jednom kad klijent unese sve potrebne informacije, može poslati zahtjev čime se razina tog zahtjeva postavlja na 1. Dostupna je dodatna opcija podsjetnika koja je detaljno opisana u odjeljku 2.2.4.

Nakon slanja obrasca, klijent u svom pregledu vidi podneseni zahtjev. Uz zahtjev je navedena i njegova trenutna razina (od 1 do 3), što je pojašnjeno detaljnije u odjeljku 2.2.7. Klijent može mijenjati zahtjev dok god je on na razini 1.

2.2.2.4. Podsjetnik na zakazani termin usluge

Nakon uspješnog završetka prijave, zahtjev prolazi kroz proceduru opisanu u odjeljku 2.2.7. Čim zahtjev dosegne razinu broj 2, klijent dobija e-mail s ponudom. Ukoliko je klijent pri slanju zahtjeva odabrao da želi podsjetnik, sustav mu ga automatski šalje u određeno vrijeme prije termina usluge, također preko adrese elektroničke pošte.

2.2.2.5. Registracija zaposlenika

Svi zaposlenici u sustav se prijavljuju kao klijenti i inicijalno imaju samo ovlasti klijenta, zaposliti ih može isključivo administrator. Za svakog se od njih uz osnovne podatke mogu navesti dodatni atributi poput životinja koje je pojedini zaposlenik spreman preuzeti te usluga koje je spreman obaviti.

Nakon što ga administrator zaposli, svaki zaposlenik ima opciju definiranja vremena u kojem nije u mogućnosti ili ne želi preuzeti posao (dalje u tekstu kao interval nedostupnosti).

Zaposleniku se na odabir nudi hoće li elektroničkom poštom primati obavijesti za sve poslove, samo za poslove u kojima je naveden kao preferencijalni odabir ili nikada.

2.2.2.6. Pregled i ovlasti zaposlenika

Nakon uspješne prijave u sustav, zaposleniku se nude tri različita pregleda poslova:

- *Nepreuzeti poslovi* (zahtjevi razine 1)
- *Poslovi na čekanju* (zahtjevi razine 2)
- *Zaključeni poslovi* (zahtjevi razine 3)

Pregled *Nepreuzeti poslovi* zaposleniku prikazuje sve zadatke koje može preuzeti. Ovdje se mogu naći samo zadaci čije se vrijeme ne preklapa ni s intervalom nedostupnosti ni s vremenom bilo kojeg već prihvaćenog zadatka od strane tog zaposlenika. Prikazuju se svi podaci o zadatku koje je registrirani klijent upisao kao i podaci o dotičnom klijentu. Postupak preuzimanja poslova i obrade zahtjeva opisan je u odjeljku 2.2.7.

2.2.2.7. Postupak obrade zahtjeva

Kada klijent zatraži uslugu, taj posao se stavlja u pregled *Nepreuzeti poslovi*. Nakon što neki zaposlenik preuzme posao, on nestaje iz pregleda *Nepreuzeti poslovi* (svim zaposlenicima) te se prebacuje u pregled *Poslovi na čekanju* (samo zaposleniku koji je taj posao preuzeo). U ovom trenutku, klijent o čijem je zadatku riječ dobija e-mail s ponudom te dotični zahtjev dobija oznaku razine 2.

Nakon što klijent plati, šalje potvrdu o plaćanju administratoru koji mora odobriti zadatak. Nakon odobrenja, zaposleniku se zadatak premješta iz pregleda *Poslovi na čekanju* u pregled *Zaključeni poslovi*, te sam zahtjev dobija oznaku razine 3.

2.2.2.8. Pregled i ovlasti Administratora

Administrator ima sve ovlasti nad sustavom:

- Dodavanje, brisanje i uređivanje zaposlenika
- Dodavanje, brisanje i uređivanje klijenata
- Dodavanje, brisanje i uređivanje zadataka
- Uvid u sve zadatke na svim razinama za sve zaposlenike

Administrator jedini može unaprijediti zahtjev na razinu broj 3 i to tek nakon što mu klijent pošalje potvrdu o plaćanju. Ima i opciju odbijanja/otkazivanja zahtjeva bilo koje razine. Za sve odbijene zahtjeve, sustav može automatski poslati odbijenicu putem elektroničke pošte. Administrator može u ime klijenata otvarati nove zahtjeve u slučajevima kada se netko naruči telefonom ili e-mailom, a ne putem web aplikacije.

2.2.3. Ostale funkcionalnosti sustava

Sustav mora omogućiti istovremeni rad administratora, zaposlenika i neograničenog broja registriranih klijenata.

2.3. Cilj i dugoročna korist rješenja

Cilj izrade web aplikacije je olakšati i ubrzati proces zatraživanja usluga te pojednostavljivanje komunikacije između zaposlenika tvrtke. Uz web aplikaciju povećanjem broja klijenata i zaposlenika neće se mijenjati kompleksnost organizacije posla kao što bi to bio slučaj da se nastavi poslovati korištenjem mailova i putem telefona.

2.4. Potencijalni kupci i korisnici

Dosadašnje korisnike uputiti će se na novi način naručivanja usluga, odnosno korištenje web aplikacije, a modernizacija načina poslovanja mogla bi dovesti nove korisnike i tako povećati promet tvrtke

3. Pojmovnik

Java: Objektno orijentiran jezik nezavisan o platformi koji se može koristiti za izradu web aplikacija.

UML (Unified Modeling Language): razvojni jezik modela opće svrhe čiji je cilj slikovno predstavljanje dizajna sustava.

SQL: Najpopularniji jezik za rad s bazom podataka. Objedinjuje funkcije jezika za definiciju podataka i jezika za rukovanje podacima.

Git: Distribuirani jezik za upravljanje izvornim kodom.

Spring Framework: Spring aplikacijski okvir može se koristiti za izradu web aplikacija u Java EE platformi.

Eclipse: Programska razvojna okolina (IDE) koja se najčešće koristi razvoj aplikacija u Javi. Sastoji se od osnovnog radnog prostora te ima mogućnost dodavanja raznih proširenja za uređivanje okoline.

JavaScript: Skriptni jezik koji se koristi u pretraživačima s ciljem stvaranja dinamičkog sadržaja.

Model-view-controller: Programski obrazac arhitekture za implementaciju korisničkih sučelja na računalima. Aplikacija se dijeli na tri međusobno povezana dijela kako bi se odvojila unutarnja reprezentacija informacije od načina na koji je ta informacija prikazana korisniku.

4. Funkcijski zahtjevi

Dionici:

- Vlasnik tvrtke
- Zaposlenici tvrtke
- Klijenti tvrtke
- Administratori web aplikacije
- Inženjeri koji održavaju sustav

Aktori i njihovi funkcionalni zahtjevi

- Administrator (inicijator)
- Klijent (inicijator)
- Zaposlenik (inicijator)
- Baza podataka (sudionik) - pohranjuje podatke o:
 - registriranim klijentima
 - registriranim zaposlenicima
 - postojećim uslugama
 - ljubimcima
- Neregistrirani korisnik (inicijator) - može se registrirati na web mjesto

Opisi obrazaca uporabe

UC1 – Registracija

- **Glavni sudionik:** neregistrirani korisnik
- **Sporodni sudionik:** poslužitelj, baza podataka
- **Preduvjeti:** -
- **Cilj:** dodati novog klijenta u sustav
- **Rezultat:** klijent dodan u sustav
- **Željeni scenarij:**
 1. Neregistrirani korisnik odabire opciju za registraciju
 2. Neregistrirani korisnik upisuje sve potrebne podatke
 3. Provjerava se ispravnost podataka
 4. Novonastali korisnik pohranjuje se u bazu podataka i inicijalno dobiva ulogu klijenta
- **Mogući drugi scenarij:**
 1. Korisnik s unesenom adresom elektroničke pošte već postoji u bazi podataka
 2. Sustav javlja poruku o grešci

UC2 – Dodavanje Zaposlenika

- **Glavni sudionik:** administrator
- **Sporodni sudionik:** poslužitelj, baza podataka
- **Preduvjeti:** administratorske ovlasti, administrator se nalazi u pregledu podataka korisnika (UC4)
- **Cilj:** dodati novog zaposlenika u sustav
- **Rezultat:** korisnik zaposlen
- **Željeni scenarij:**
 1. Administrator odabire opciju za zapošljavanje korisnika

2. Uloga korisnika promiče se u ulogu zaposlenika

UC3 – Prijava

- **Glavni sudionik:** korisnik (administrator/zaposlenik/klijent)
- **Sporedni sudionik:** poslužitelj, baza podataka
- **Preduvjeti:** korisnik postoji u sustavu
- **Cilj:** prijava korisnika u sustav
- **Rezultat:** korisnik prijavljen u sustav
- **Željeni scenarij:**
 1. Korisnik odabere opciju prijave u sustav
 2. Korisnik upisuje adresu elektroničke pošte i lozinku
 3. Poslužitelj provjerava podatke pomoću baze podataka
 4. Korisnik je prijavljen u sustav, započinje sjednica
- **Mogući drugi scenarij:**
 1. Podaci su pogrešni
 2. Poslužitelj obavijesti korisnika o grešci i traži ga ponovni unos podataka

UC4 – Pregled i Uređivanje Podataka

- **Glavni sudionik:** korisnik
- **Sporedni sudionik:** poslužitelj, baza podataka
- **Preduvjeti:** korisnik je prijavljen u sustav
- **Cilj:** pregled/promjena podataka korisnika
- **Rezultat:** podaci korisnika su promijenjeni/pregledani
- **Željeni scenarij:**
 1. Korisnik odabire opciju pregleda svojih podataka
 2. Korisniku se za neke podatke nudi i mogućnost izmjene
 3. Poslužitelj provjerava jesu li izmijenjeni podaci u odgovarajućem formatu
 4. Novi se podaci upisuju u bazu podataka
- **Mogući drugi scenarij:**
 1. Dio podataka je u pogrešnom formatu te poslužitelj obavještava korisnika o grešci

UC5 – Uklanjanje Zaposlenika

- **Glavni sudionik:** administrator
- **Sporedni sudionik:** poslužitelj, baza podataka
- **Preduvjeti:** administratorske ovlasti, administrator se nalazi u pregledu podataka zaposlenika (UC4)
- **Cilj:** maknuti korisniku ulogu zaposlenika
- **Rezultat:** zaposlenik više nije u sustavu/bazi
- **Željeni scenarij:**
 1. Administrator odabere opciju uklanjanja uloge zaposlenika
 2. Korisnik gubi ulogu zaposlenika
- **Mogući drugi scenarij:**

1. Zaposlenik trenutno ima dodijeljenu potvrđenu uslugu
2. Sustav javlja poruku o grešci

UC6 – BrisanjeKorisnika

- **Glavni sudionik:** korisnik
- **Sporedni sudionik:** poslužitelj, baza podataka
- **Preduvjeti:** Klijent je prijavljen i nalazi se u pregledu vlastitih podataka (UC4)
- **Cilj:** obrisati korisnika iz sustava
- **Rezultat:** Korisnik više nije u sustavu/bazi
- **Željeni scenarij:**
 1. Korisnik odabere opciju brisanja sebe iz sustava
 2. Korisnik se obriše iz sustava
- **Mogući drugi scenarij**
 1. Korisnik ima ulogu zaposlenika te sustav javlja poruku o nemogućnosti brisanja dok se ona ne oduzme

UC7 – DodavanjeLjubimca

- **Glavni sudionik:** klijent
- **Sporedni sudionik:** poslužitelj, baza podataka
- **Preduvjeti:** klijent je prijavljen u sustav
- **Cilj:** dodati novog ljubimca klijentu
- **Rezultat:** klijent ima novog ljubimca u sustavu
- **Željeni scenarij:**
 1. Klijent odabire opciju dodavanja novog ljubimca
 2. Klijent unosi sve potrebne podatke o ljubimcu
 3. Podaci o ljubimcu upisuju se u bazu podataka
- **Mogući drugi scenarij:**
 1. Dio podataka je u pogrešnom formatu te poslužitelj obavještava korisnika o grešci

UC8 – BrisanjeLjubimca

- **Glavni sudionik:** klijent
- **Sporedni sudionik:** poslužitelj, baza podataka
- **Preduvjeti:** klijent je prijavljen u sustav
- **Cilj:** obrisati jednog od ljubimaca klijenta
- **Rezultat:** ljubimac je obrisao
- **Željeni scenarij:**
 1. Klijent odabire opciju brisanja ljubimca
 2. Sustav dohvaća sve ljubimce koji pripadaju klijentu
 3. Klijent odabire i briše ljubimca
 4. Ljubimac je obrisao

UC9 – StvaranjeRezervacije

- **Glavni sudionik:** klijent

- **Sporedni sudionik:** poslužitelj, baza podataka
- **Preduvjeti:** Klijent je prijavljen u sustav
- **Cilj:** stvoriti rezervaciju
- **Rezultat:** rezervacija stvorena
- **Željeni scenarij:**
 1. Klijent odabire opciju rezerviranja usluge
 2. Klijent ispunjava detalje rezervacije
 3. Rezervacija je stavljena u listu gdje je zaposlenici mogu preuzeti te postaje dostupna klijentu na praćenje
- **Mogući drugi scenarij:**
 1. Dio podataka je u pogrešnom formatu te poslužitelj obavještava korisnika o grešci

UC10 – UređivanjeRezervacije

- **Glavni sudionik:** klijent
- **Sporedni sudionik:** poslužitelj, baza podataka
- **Preduvjeti:** rezervacija još nije preuzeta, klijent se nalazi u pregledu za praćenje rezervacija (UC11)
- **Cilj:** promijeniti podatke rezervacije
- **Rezultat:** promijenjeni podaci rezervacije
- **Željeni scenarij:**
 1. Klijent odabire rezervaciju koju želi urediti
 2. Klijent uređuje rezervaciju
 3. Klijent sprema promjene
 4. Podaci su ažurirani
- **Mogući drugi scenarij:**
 1. Netko je u međuvremenu preuzeo rezervaciju ili dio podataka je u pogrešnom formatu
 2. Poslužitelj javlja grešku

UC11 – PraćenjeRezervacije

- **Glavni sudionik:** klijent
- **Sporedni sudionik:** poslužitelj, baza podataka
- **Preduvjeti:** postoje rezervacije za praćenje te je prijavljeni klijent njihov vlasnik
- **Cilj:** saznati stanje rezervacije
- **Rezultat:** Klijent je dobio informaciju o stanju rezervacije
- **Željeni scenarij:**
 1. Klijent ulazi u pregled za praćenje rezervacija
 2. Poslužitelj iz baze dohvaća popis svih rezervacija dotičnog klijenta te mu ih prikazuje
 3. Klijent dobiva informacije o stanjima svih svojih rezervacija

UC12 – PriprihvatanjeRezervacije

- **Glavni sudionik:** zaposlenik
- **Sporedni sudionik:** poslužitelj, baza podataka
- **Preduvjeti:** prijavljeni zaposlenik se nalazi u pregledu rezervacija (UC16), rezervacija se nalazi u listi nepreuzetih rezervacija

- **Cilj:** prihvatiti termin rezervacije (rezervacija se premješta iz liste nepreuzetih u zaposlenikovu listu preuzetih rezervacija)
- **Rezultat:** termin rezervacije prihvaćen/ rezervacija se premješta u listu preuzetih rezervacija zaposlenika
- **Željeni scenarij:**
 1. Zaposlenik pregledava listu nepreuzetih rezervacija
 2. Zaposlenik odabire rezervaciju koju želi prihvatiti
 3. Rezervacija se prebacuje u vlastitu listu preuzetih rezervacija dotičnog zaposlenika
 4. Korisniku se šalje poruka s ponudom
 5. Podaci se upisuju u bazu podataka
- **Mogući drugi scenarij:**
 1. Netko je drugi u međuvremenu prihvatio rezervaciju koju je zaposlenik htio preuzeti
 2. Poslužitelj javlja zaposleniku da je rezervacija već preuzeta

UC13 – PotvrdaPlaćeneRezervacije

- **Glavni sudionik:** administrator
- **Sporedni sudionik:** poslužitelj, baza podataka
- **Preduvjeti:** administratorske ovlasti, administrator se nalazi u pregledu rezervacija (UC16), postoji zaposlenik koji je rezervaciju preuzeo
- **Cilj:** potvrditi rezervaciju kako bi mogao započeti stvaran posao
- **Rezultat:** rezervacija potvrđena, posao započinje
- **Željeni scenarij:**
 1. Administrator odabire opciju potvrđivanja rezervacije te se ona prebacuje u listu potvrđenih rezervacija

UC14 – OtkazivanjeRezervacije

- **Glavni sudionik:** administrator
- **Sporedni sudionik:** poslužitelj, baza podataka
- **Preduvjeti:** administratorske ovlasti, administrator se nalazi u pregledu rezervacija (UC16)
- **Cilj:** otkazati rezervaciju neovisno o tome u kojoj se fazi / listi nalazi
- **Rezultat:** rezervacija je otkazana
- **Željeni scenarij:**
 1. Administrator odabere opciju otkazivanja neke od rezervacija
 2. Rezervacija je otkazana i miče se iz svih pregleda

UC15 – SlanjePodsjetnika

- **Glavni sudionik:** poslužitelj
- **Sporedni sudionik:** klijent, baza podataka
- **Preduvjeti:** -
- **Cilj:** podsjetiti Klijenta da ima zakazani termin
- **Rezultat:** Klijent je informiran
- **Željeni scenarij:**
 1. Poslužitelj periodički (jednom dnevno) provjerava postoje li potvrđene rezervacije unutar iduća 24 sata za koje je klijent odabrao opciju slanja podsjetnika

2. Za svaku rezervaciju koja ispunjava navedeni uvjet, poslužitelj šalje podsjetnik klijentu putem elektroničke pošte.

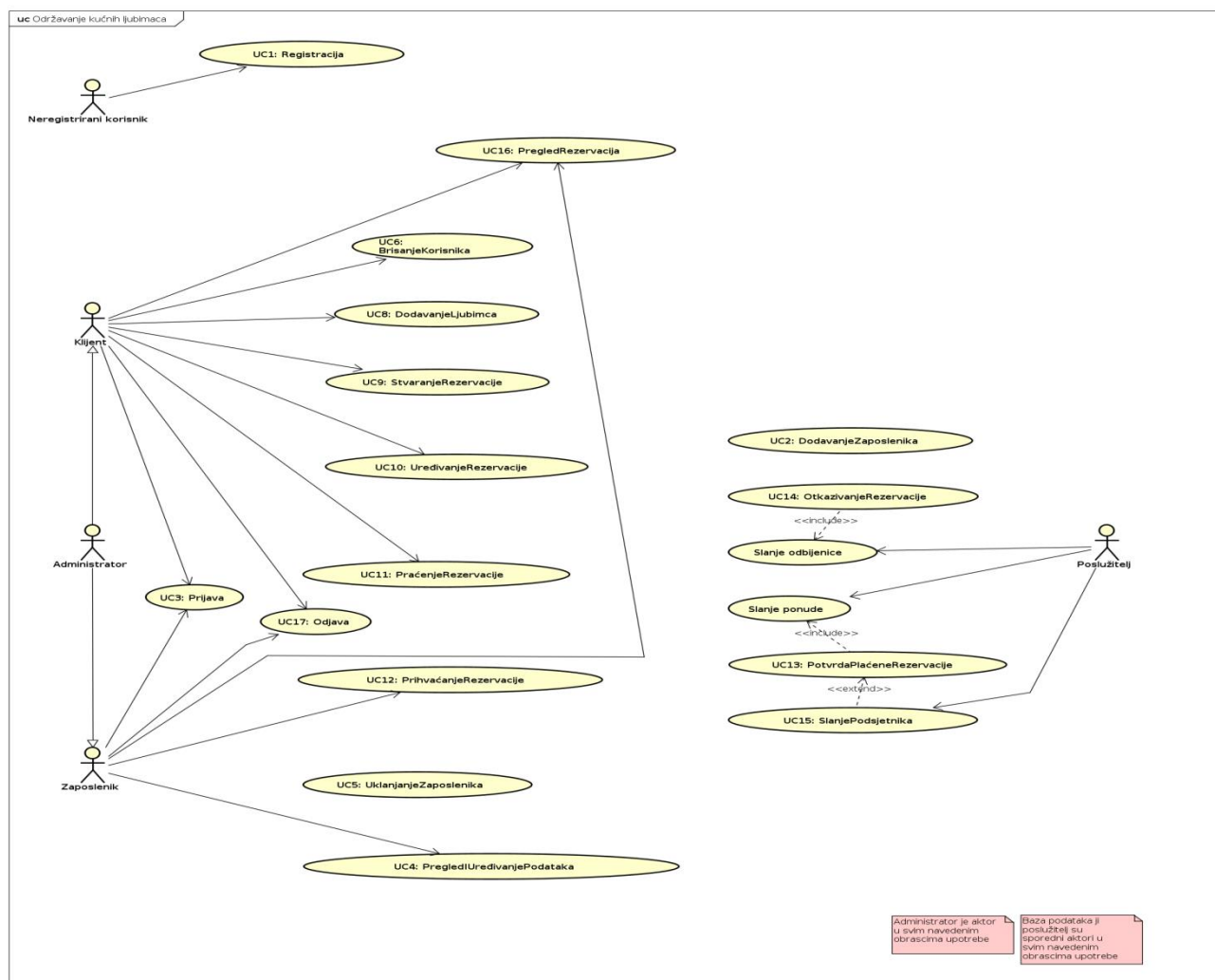
UC16 – PregledRezervacija

- **Glavni sudionik:** zaposlenik
 - **Sporedni sudionik:** poslužitelj, baza podataka
 - **Preduvjeti:** zaposlenik je prijavljen u sustav
 - **Cilj:** pregledati postojeće rezervacije
 - **Rezultat:** -
 - **Željeni scenarij:**
1. Zaposlenik odabere opciju pregleda postojećih rezervacija
 2. Zaposlenik vidi:
 - Sve nepreuzete rezervacije u vremenu u kojem je dotični zaposlenik dostupan
 - Sve rezervacije koje je on preuzeo, a još nisu plaćene
 - Sve rezervacije koje je on preuzeo i plaćene su

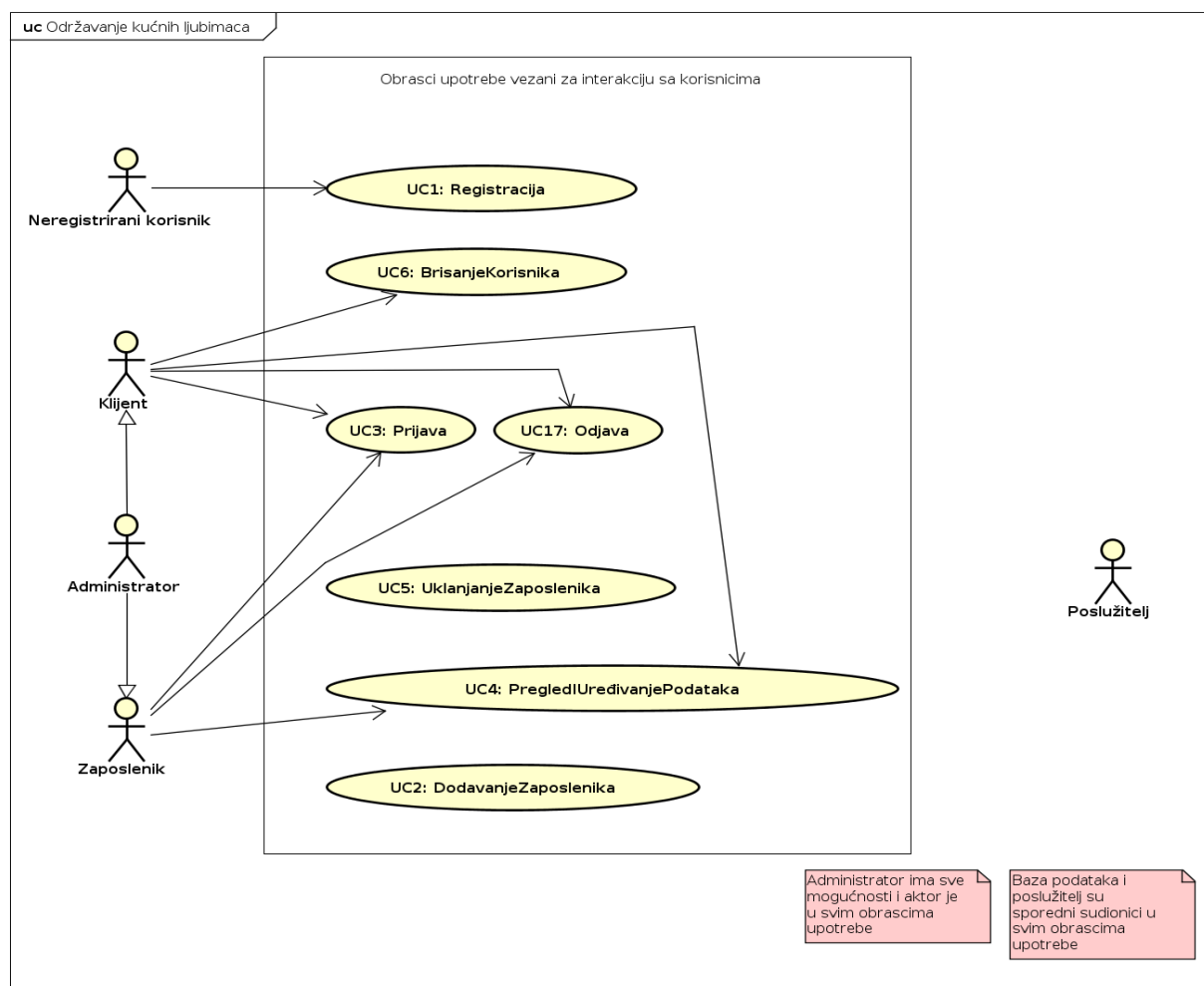
UC17 – Odjava

- **Glavni sudionik:** korisnik
 - **Sporedni sudionik:** poslužitelj
 - **Preduvjeti:** korisnik je prijavljen u sustav
 - **Cilj:** odjaviti se iz sustava
 - **Rezultat:** korisnik više nije prijavljen, sjednica je zatvorena
 - **Željeni scenarij:**
1. Korisnik odabere opciju odjave iz sustava
 2. Poslužitelj odbacuje postojeću sjednicu i odjavljuje korisnika

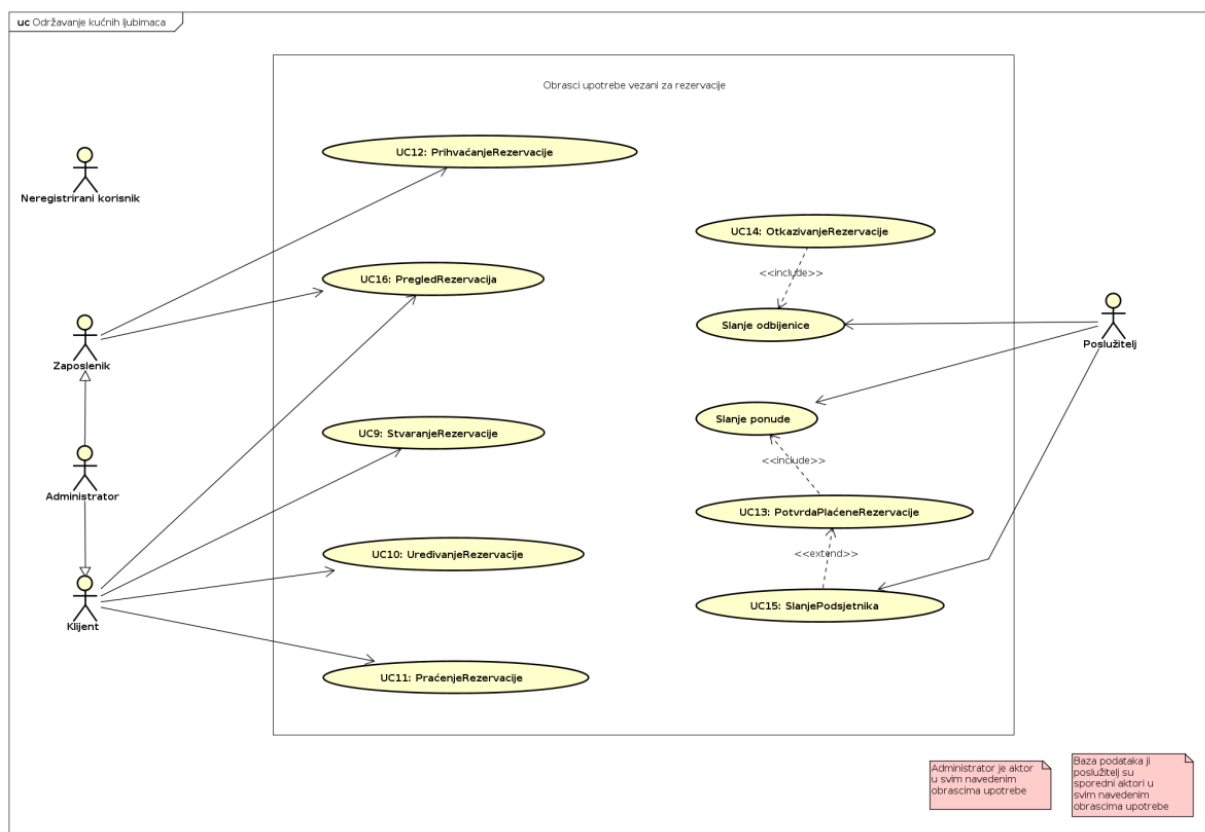
Dijagrami obrazaca uporabe



Slika 4.1 - Cjeloviti pregled svih obrazaca upotrebe



Slika 4.2 - Obrasci vezani za interakciju s korisnicima

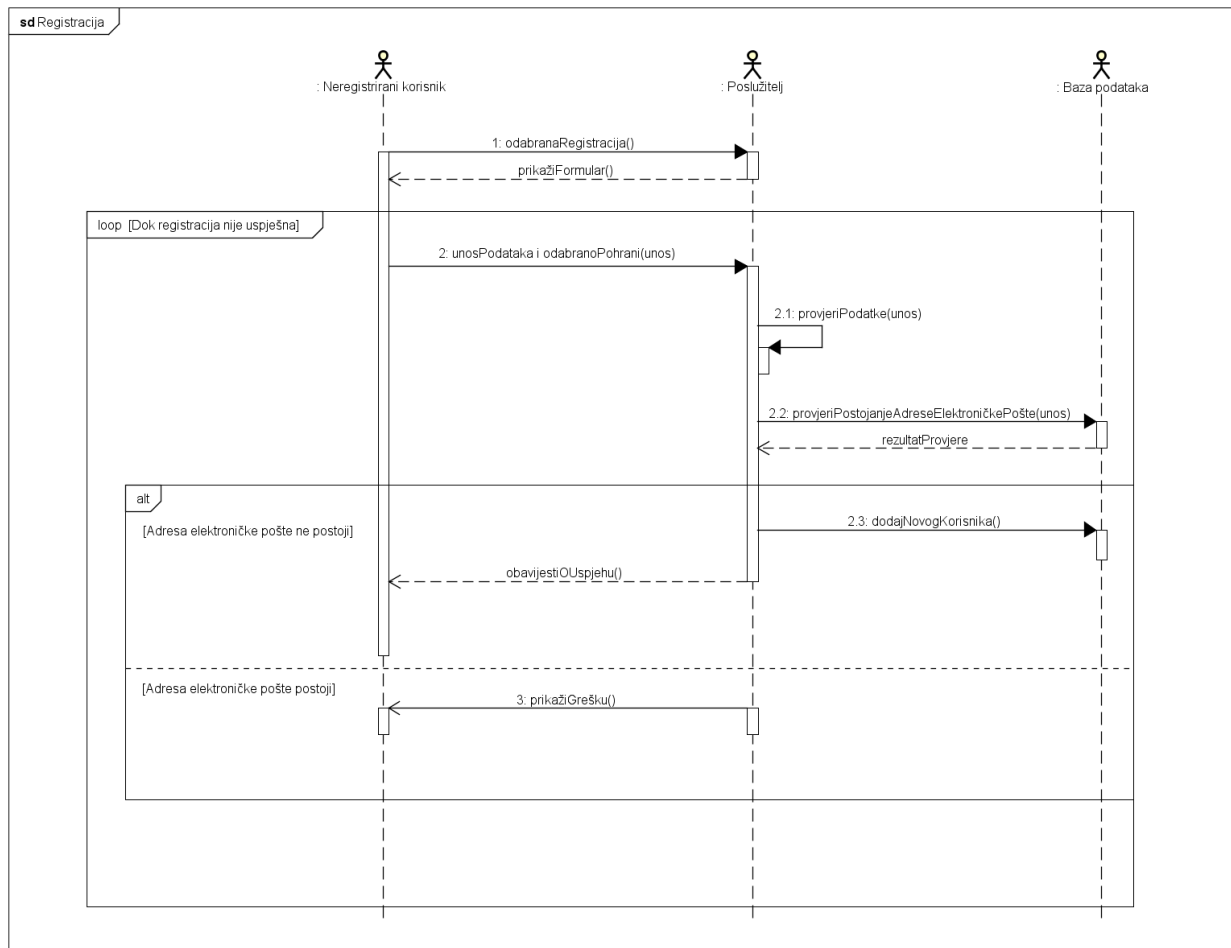


Slika 4.3 - Obrasci vezani za rezervacije

Sekvencijski dijagrami

Obrazac uporabe UC1 (Registracija)

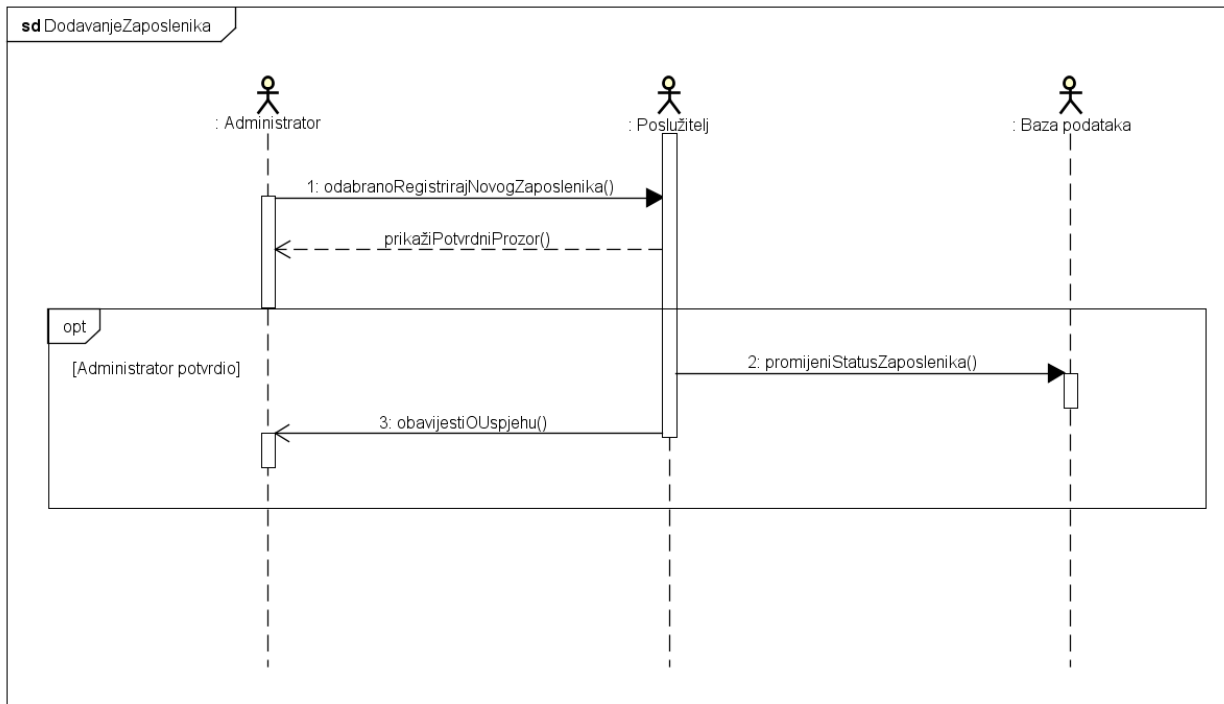
Neregistrirani korisnik odabire opciju registracije te mu poslužitelj prikazuje formular za registraciju. Sljedeći postupak se ponavlja dok neregistrirani korisnik uspješno ne dovrši registraciju. Neregistrirani korisnik unosi tražene podatke i odabire pohranu svojih podataka, nakon čega poslužitelj provjerava ispravnost podataka i provjerava postoji li klijent s danom adresom elektroničke pošte u bazi podataka. U slučaju da su podatci ispravno uneseni te ne postoji korisnik u bazi podataka, on će biti uspješno registriran i obaviješten o uspjehu registracije. U suprotnom, neregistriranom će korisniku biti prikazana poruka o grešci i omogućen ponovni unos podataka.



Slika 4.4 - Sekvencijski dijagram za UC1 (Registracija)

Obrazac uporabe UC2 (DodavanjeZaposlenika)

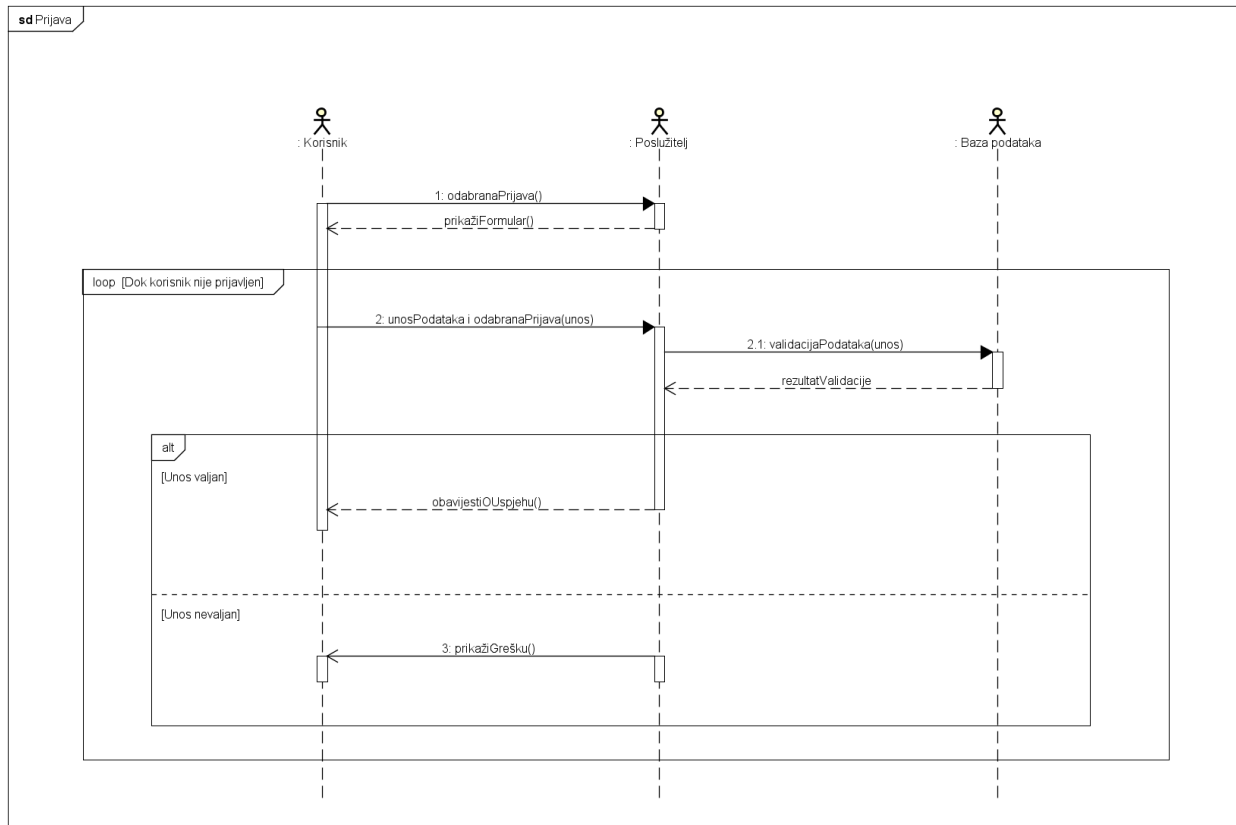
Administrator se nalazi na pregledu podataka korisnika. Ovdje može odabrati opciju dodavanja uloge zaposlenika korisniku, nakon čega mu poslužitelj prikazuje potvrdni prozor. Potvrdi li administrator navedenu opciju, korisniku se na ulogu klijenta dodaje i uloga zaposlenika.



Slika 4.5 - Sekvencijski dijagram za UC2 (DodavanjeZaposlenika)

Obrazac uporabe UC3 (Prijava)

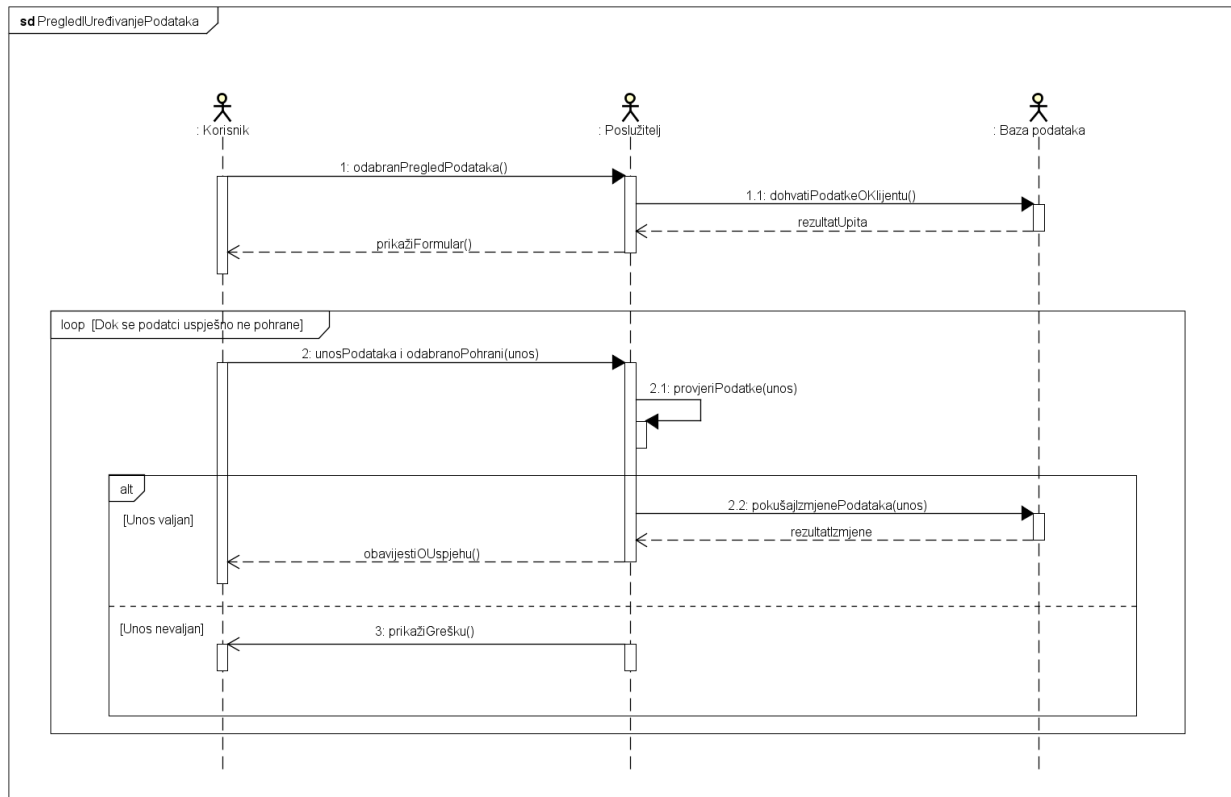
Korisnik odabire opciju prijave u sustav te mu poslužitelj prikazuje formular za prijavu. Sljedeći se postupak ponavlja dok korisnik uspješno ne dovrši prijavu. Korisnik unosi tražene podatke i odabire prijavu, nakon čega poslužitelj proslijeđuje podatke bazi podataka koja validira unos. Rezultat validacije vraća se poslužitelju koji u slučaju ispravnog unosa, obavještava korisnika o uspješnosti prijave u sustav. U suprotnome će korisniku biti prikazana poruka o grešci i omogućen ponovni upis podataka.



Slika 4.6 - Sekvencijski dijagram za UC3 (Prijava)

Obrazac uporabe UC4 (*PregledIUređivanjePodataka*)

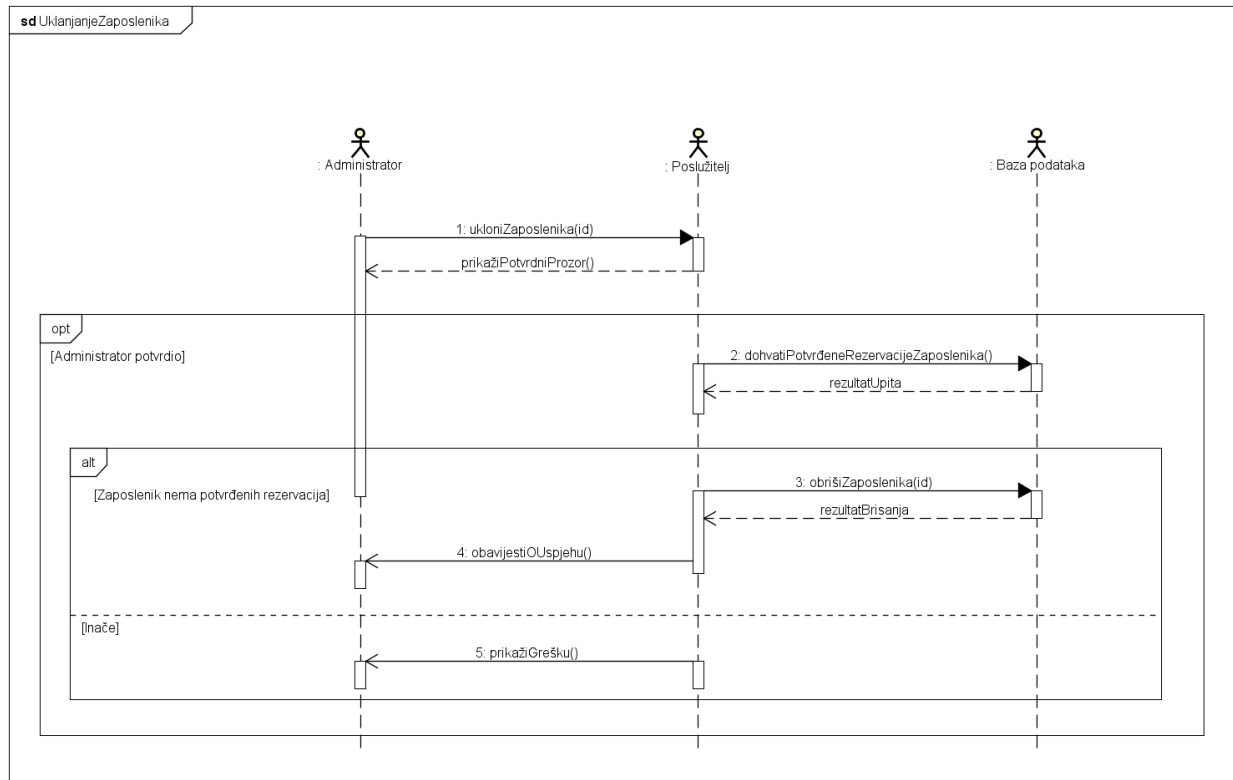
Korisnik odabire opciju pregleda svojih podataka nakon čega poslužitelj dohvaća relevantne podatke o korisniku iz baze podataka i prikazuje mu formular. Sljedeći se postupak ponavlja dok se željene promjene uspješno ne pohrane. Korisnik potom unosi podatke i odabire pohranu, nakon čega poslužitelj provjerava ispravnost unesenih podataka. U slučaju da su podatci valjani, poslužitelj pokušava izmijeniti podatke spremljene u bazi podataka. Baza podataka vraća rezultat pokušaja izmjene poslužitelju, koji ga prikazuje korisniku. U slučaju da je korisnikov unos bio nevaljan, poslužitelj korisniku prikazuje poruku o grešci i omogućuje ponovni upis podataka.



Slika 4.7 - Sekvencijski dijagram za UC4 (*PregledIUređivanjePodataka*)

Obrazac uporabe UC5 (UklanjanjeZaposlenika)

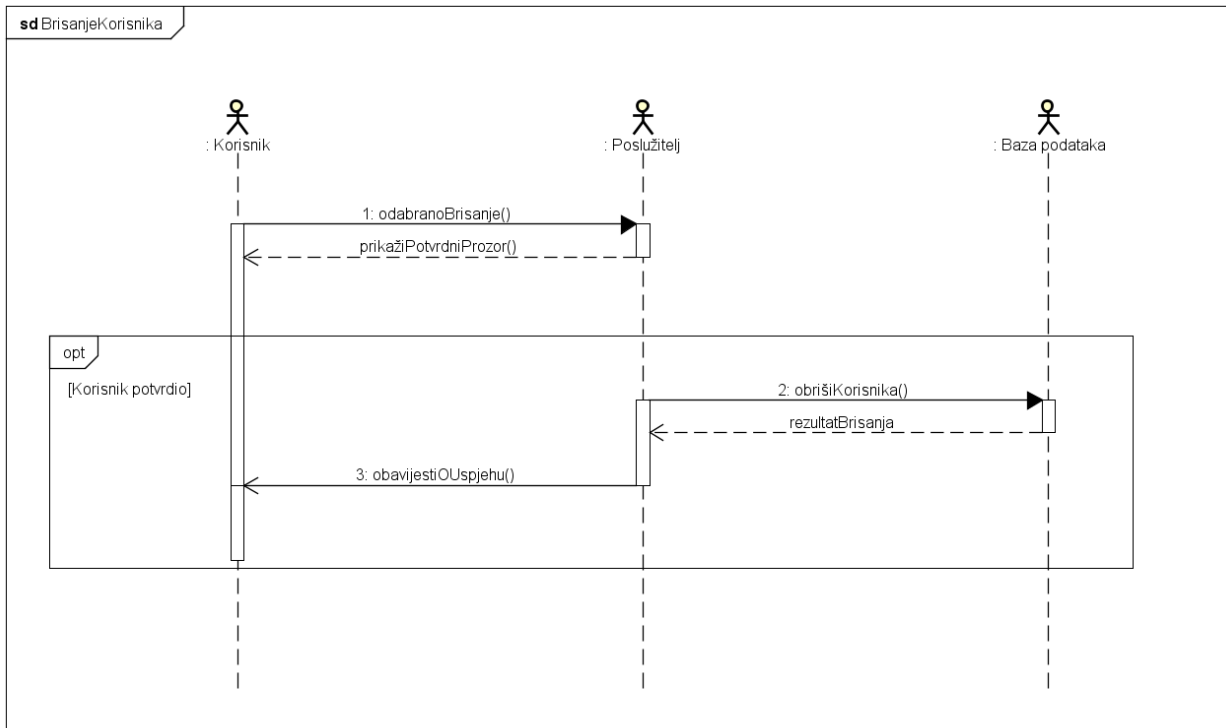
Administrator se nalazi na pregledu podataka zaposlenika i tamo odabire opciju uklanjanja zaposlenika. Poslužitelj mu nakon toga prikazuje potvrdni prozor. U slučaju da je administrator potvrdio i dotičnom zaposleniku trenutno nije dodijeljena ni jedna potvrđena rezervacija, miče mu se uloga zaposlenika. U suprotnom, sustav javlja grešku.



Slika 4.8 - Sekvencijski dijagram za UC5 (UklanjanjeZaposlenika)

Obrazac uporabe UC6 (*BrisanjeKorisnika*)

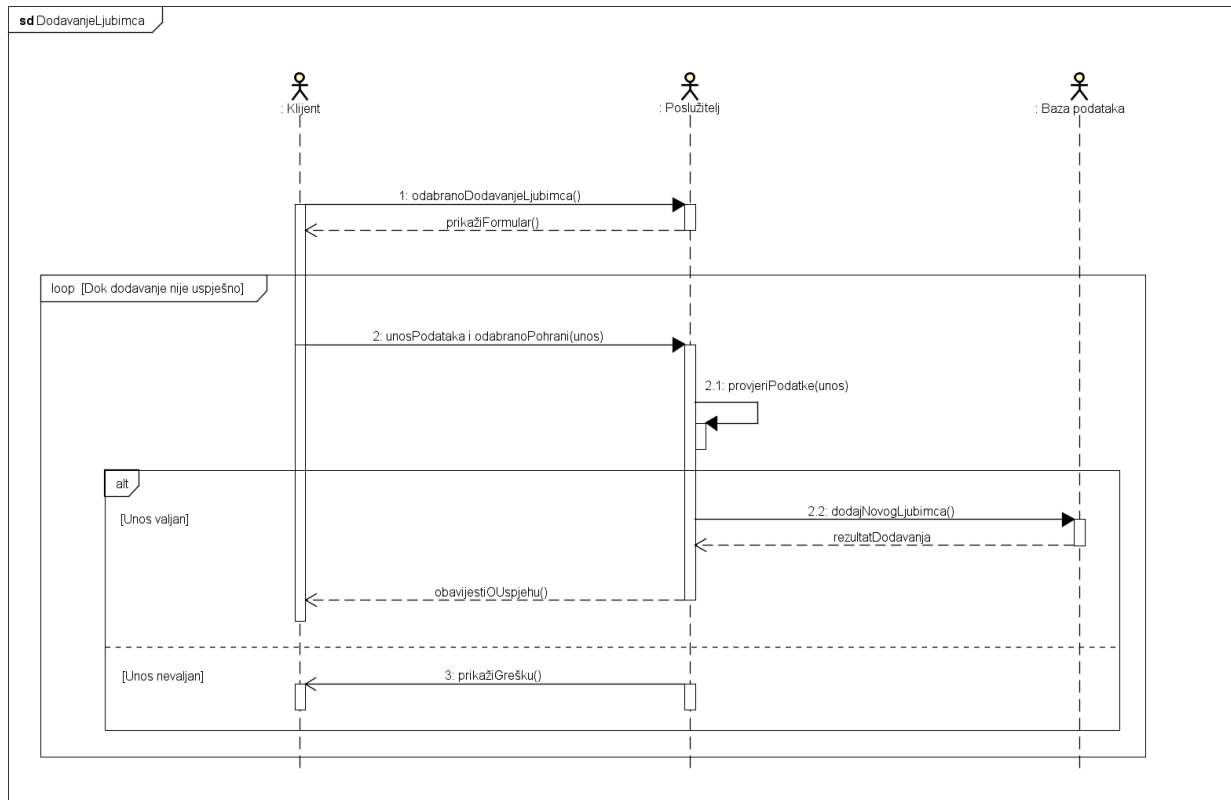
Korisnik odabire opciju brisanja svog korisničkog računa, nakon čega poslužitelj prikazuje potvrdni prozor. U slučaju da korisnik potvrdi brisanje, poslužitelj briše korisnika iz baze podataka i obavještava korisnika o uspješnosti brisanja iz sustava.



Slika 4.9 - Sekvencijski dijagram za UC6 (*BrisanjeKorisnika*)

Obrazac uporabe UC7 (DodavanjeLjubimca)

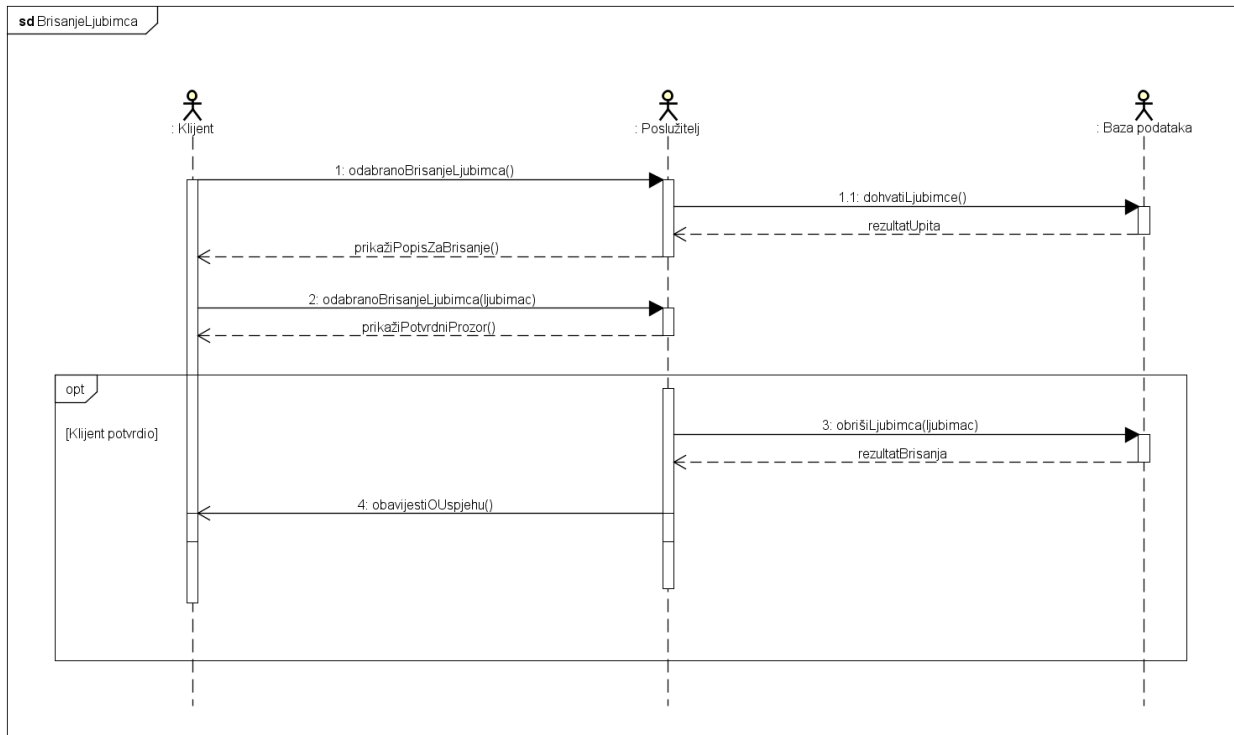
Klijent odabire opciju dodavanja novog ljubimca te mu poslužitelj prikazuje formular za dodavanje ljubimca. Sljedeći postupak se ponavlja dok dodavanje ljubimca nije uspješno. Klijent unosi podatke i odabire pohranu, nakon čega poslužitelj provjerava ispravnost podataka. U slučaju da su podatci ispravni, u bazu podataka se dodaje novi ljubimac i poslužitelj obavještava klijenta o uspjehu. U suprotnom, klijentu će biti prikazana poruka o grešci i omogućen ponovni upis podataka.



Slika 4.10 - Sekvencijski dijagram za UC7 (DodavanjeLjubimca)

Obrazac uporabe UC8 (BrisanjeLjubimca)

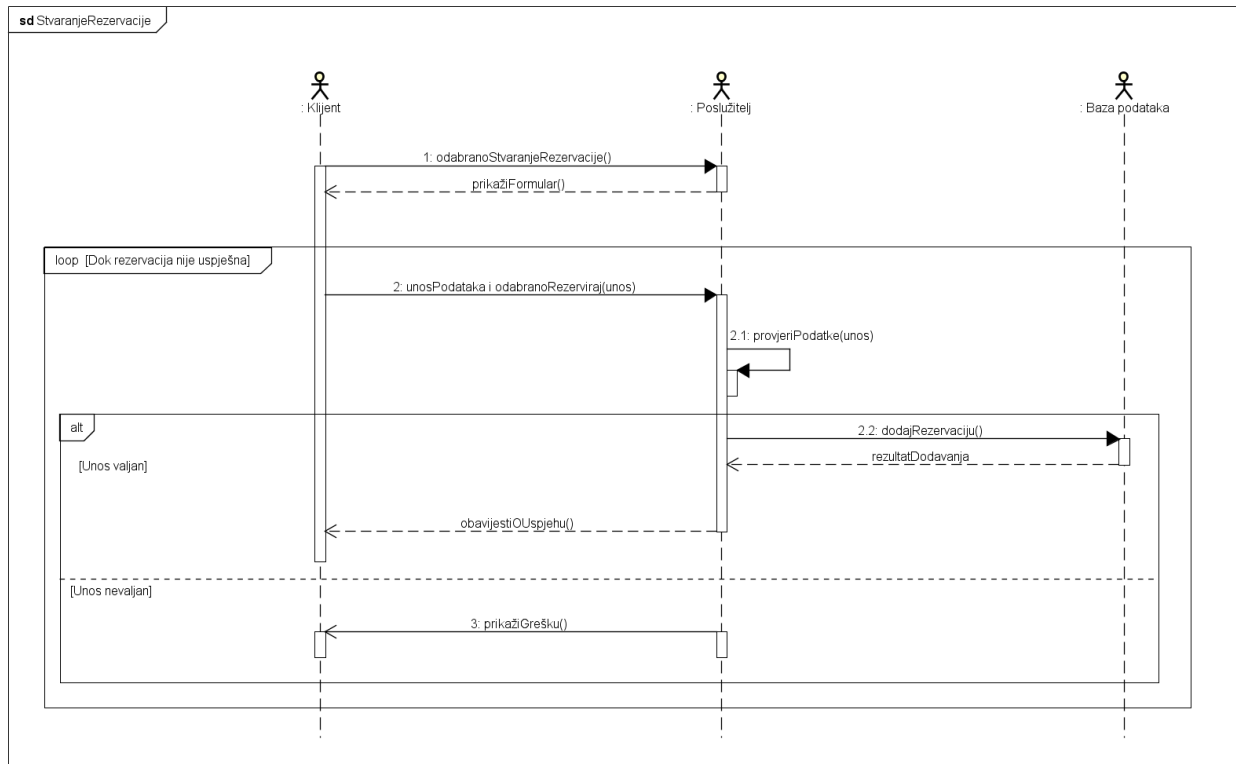
Klijent odabire opciju brisanja ljubimca nakon čega poslužitelj dohvaća sve klijentove ljubimce koje prikazuje klijentu. Klijent odabire jednog ljubimca iz liste za brisanje, te poslužitelj prikazuje potvrdni prozor. U slučaju da klijent potvrdi brisanje ljubimca, poslužitelj briše ljubimca iz baze podataka te prikazuje rezultat brisanja klijentu.



Slika 4.11 - Sekvencijski dijagram za UC8 (BrisanjeLjubimca)

Obrazac uporabe UC9 (StvaranjeRezervacije)

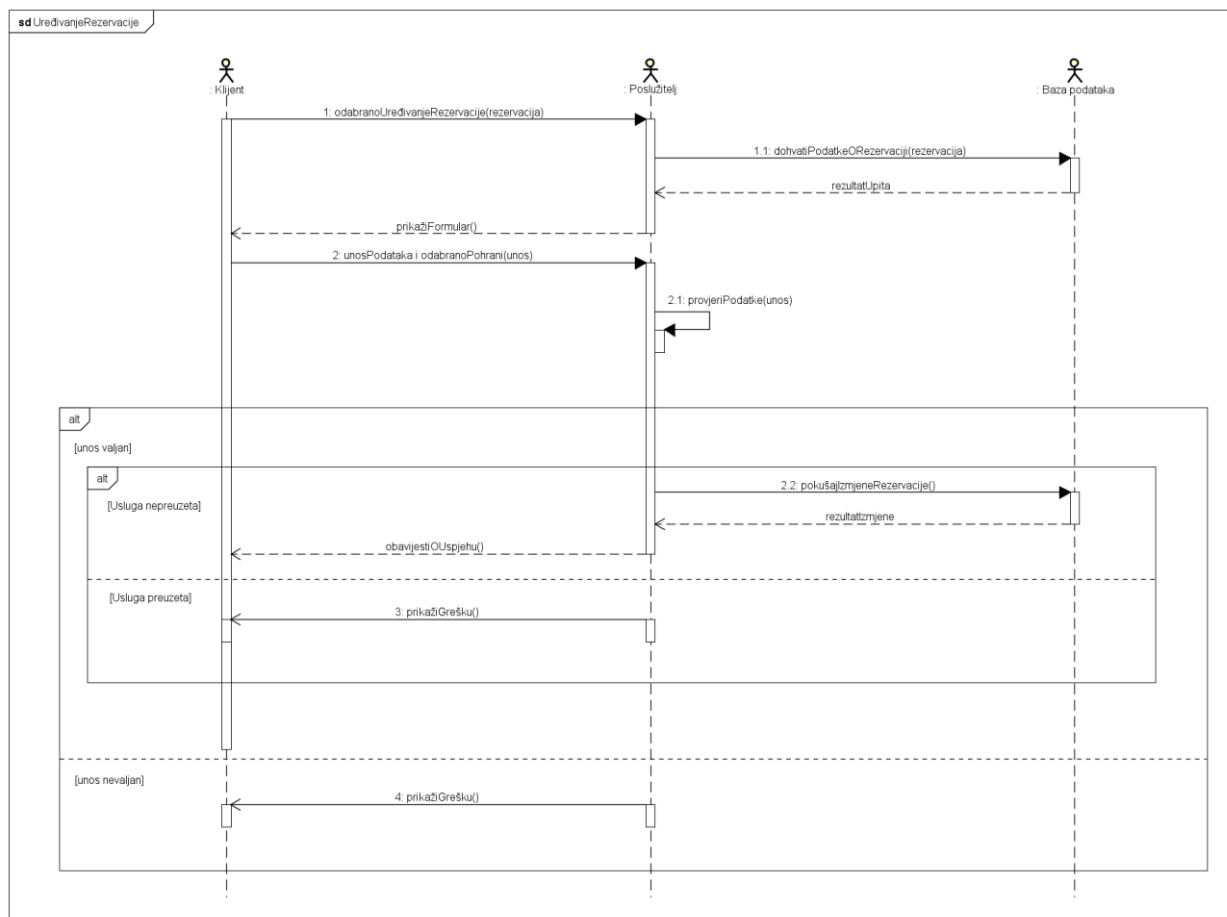
Klijent odabire opiju rezervacije nove usluge te mu poslužitelj prikazuje formular za rezervaciju usluge. Sljedeći postupak se odvija dok klijent ne rezervira novu uslugu. Klijent unosi tražene podatke i odabire pohranu podataka, nakon čega poslužitelj provjerava ispravnost podataka. U slučaju da je unos valjan, rezervacija se dodaje u bazu podataka i poslužitelj klijenta obavještava o uspješnosti rezervacije. U suprotnome, klijentu će biti prikazana poruka o grešci i omogućen ponovni upis podataka.



Slika 4.12 - Sekvencijski dijagram za UC9 (StvaranjeRezervacije)

Obrazac uporabe UC10 (UređivanjeRezervacije)

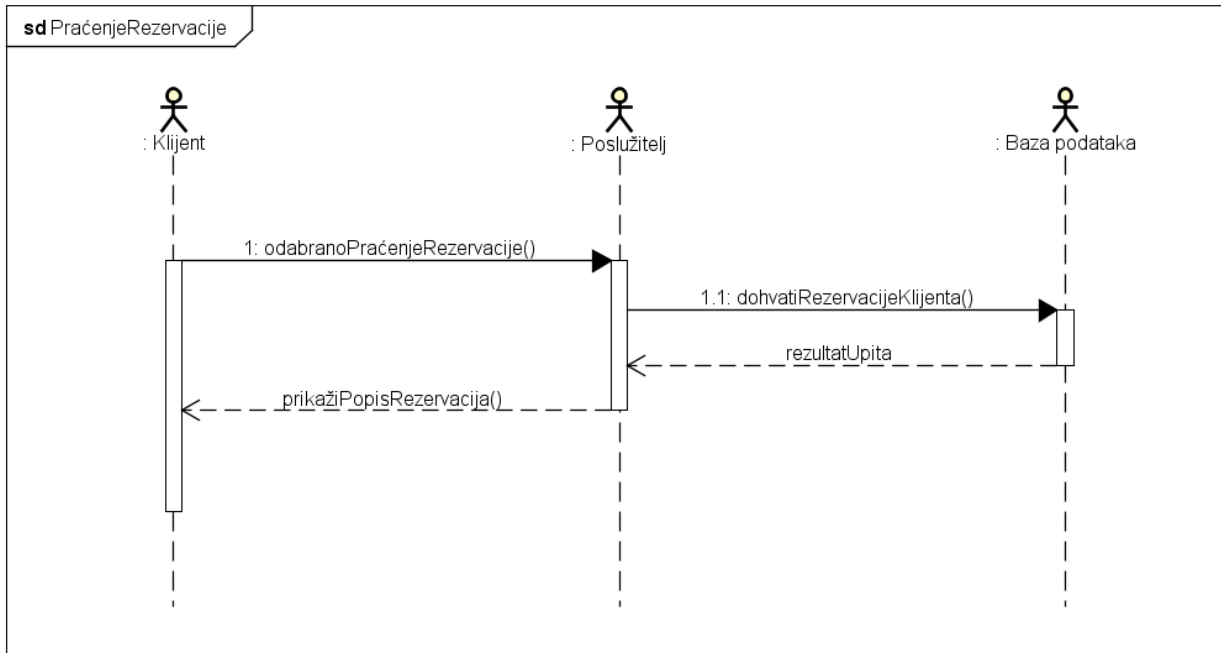
U pregledu rezervacija, korisniku se kraj rezervacija koje se još mogu uređivati nudi opcija “Uredi”. Nakon odabira opcije za uređivanje, korisniku se kroz obrazac za uređivanje prikazuju detalji rezervacije koje može mijenjati. Korisnik vrši željene promjene te podnosi zahtjev za izmjenom. Sve dok podaci koje je korisnik unio nisu valjani, prikazuje se obavijest o grešci i traže se ponovni unos i podnošenje zahtjeva. Tek kada korisnik unese valjane podatke, još jednom se provjerava stanje rezervacije. Ako je neki zaposlenik u međuvremenu rezervaciju preuzeo, korisnik dobiva obavijest o nemogućnosti izmjene rezervacije.



Slika 4.13 - Sekvencijski dijagram za UC10 (UređivanjeRezervacije)

Obrazac uporabe UC11 (*PraćenjeRezervacije*)

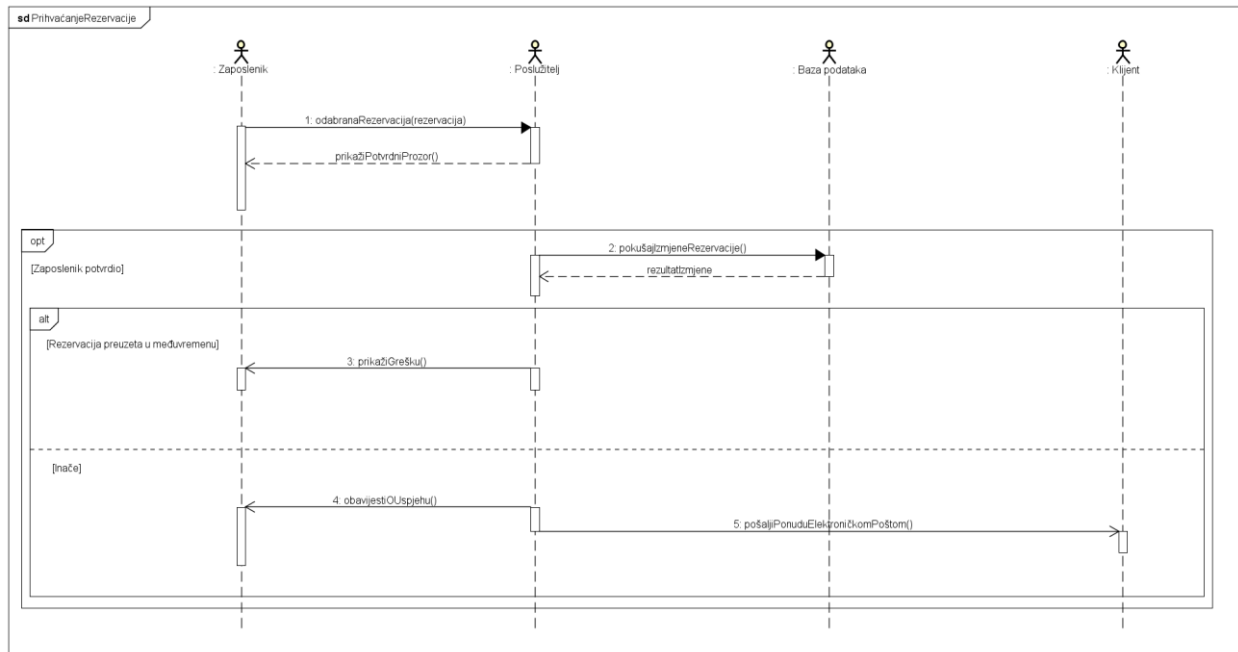
Klijent otvara pregled svih svojih rezervacija. Poslužitelj dohvaća usluge koje je rezervirao dotični klijent te prikazuje njihov popis kroz sučelje web aplikacije. Kraj svake su rezervacije vidljivi i njeni detalji, opcija otkazivanja rezervacije te (u nekim slučajevima) opcija za uređivanje rezervacija.



Slika 4.14 - Sekvencijski dijagram za UC11 (PraćenjeRezervacije)

Obrazac uporabe UC12 (PrihvaćanjeRezervacije)

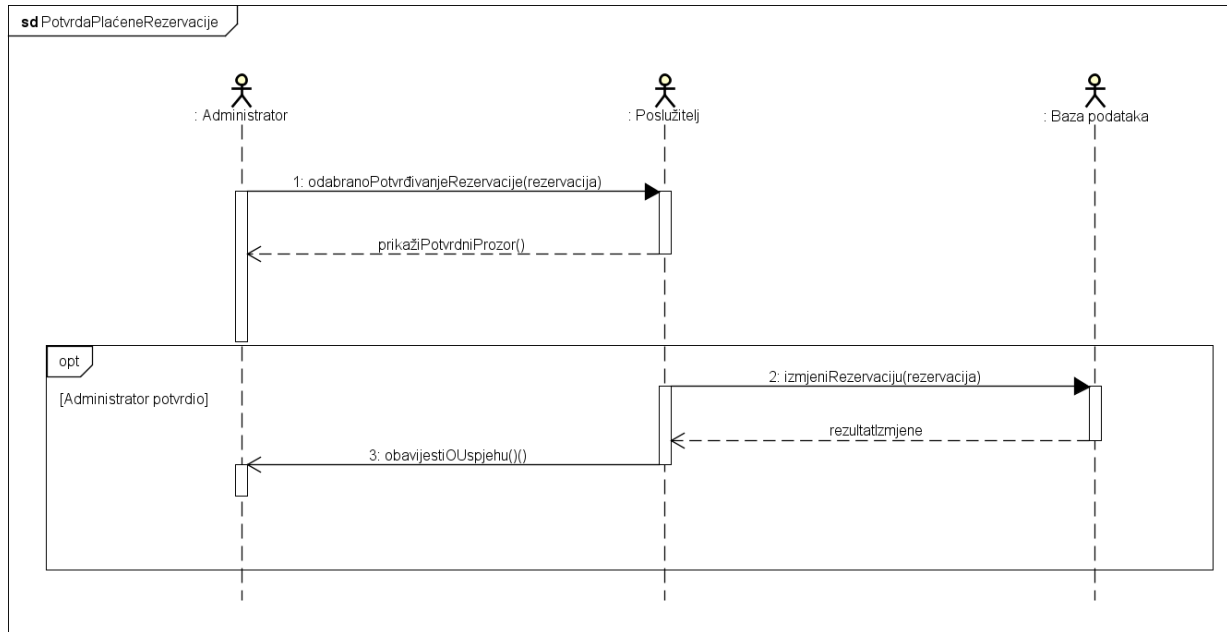
Zaposlenik se nalazi u pregledu svih rezervacija. Uz svaku se rezervaciju razine jedan (nepreuzetu rezervaciju) nudi opcija preuzimanja. Zaposlenik odabire preuzimanje neke od ponuđenih rezervacija. Ako u međuvremenu spomenutu uslugu nije preuzeo nitko drugi, ona se označava kao preuzeta. Poslužitelj joj ažurira stanje u bazi podataka te se ona time prebacuje iz liste nepreuzetih rezervacija u listu preuzetih rezervacija dotičnog zaposlenika te klijent dobiva poruku s ponudom.



Slika 4.15 - Sekvencijski dijagram za UC12 (PrihvaćanjeRezervacije)

Obrazac uporabe UC13 (PotvrdaPlaćeneRezervacije)

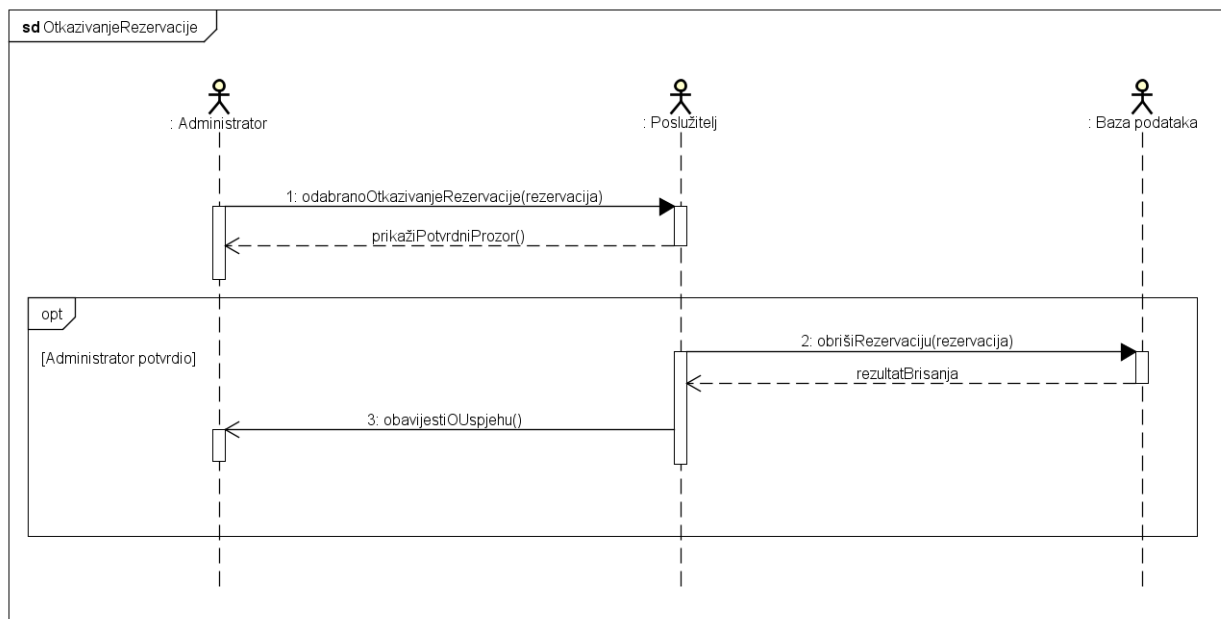
Administrator se nalazi u pregledu svih rezervacija. Svaka preuzeta usluga nudi opciju potvrde. Administrator može potvrditi bilo koju rezervaciju, tipično to radi nakon što je ona plaćena, ali sustav plaćanje ne regulira ni na koji način. Rezervacija se nakon potvrđivanja prebacuje u listu potvrđenih usluga te administrator dobiva obavijest o uspjehu akcije.



Slika 4.16 - Sekvencijski dijagram za UC13 (PotvrdaPlaćeneRegistracije)

Obrazac uporabe UC14 (OtkazivanjeRezervacije)

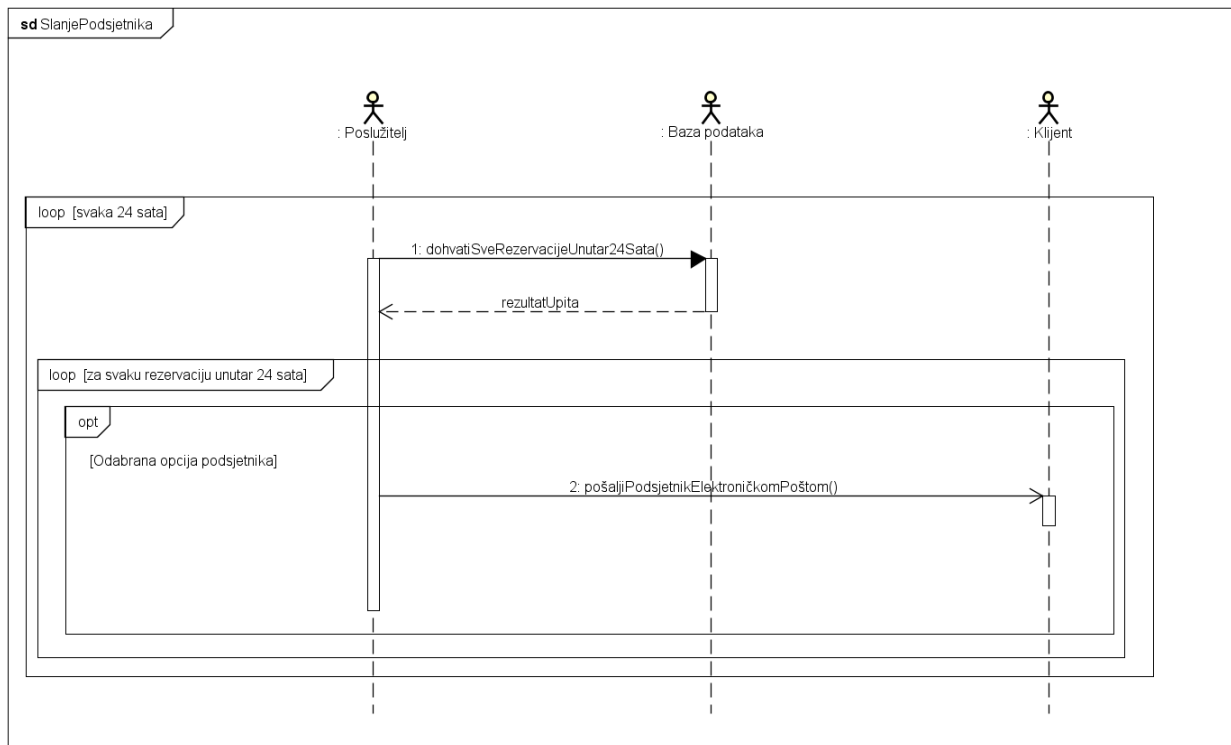
U pregledu rezervacija, korisniku se kraj svih rezervacija nudi opcija “Otkazi”. Ova je opcija dostupna uvijek, neovisno o razini rezervacije. Nakon što korisnik odabere akciju otkazivanja, poslužitelj mu prikazuje potvrdni prozor. Potvrdi li korisnik otkazivanje, rezervacija se miče s popisa aktivnih rezervacija.



Slika 4.17 - Sekvencijski dijagram za UC14 (OtkazivanjeRezervacije)

Obrazac uporabe UC15 (SlanjePodsjetnika)

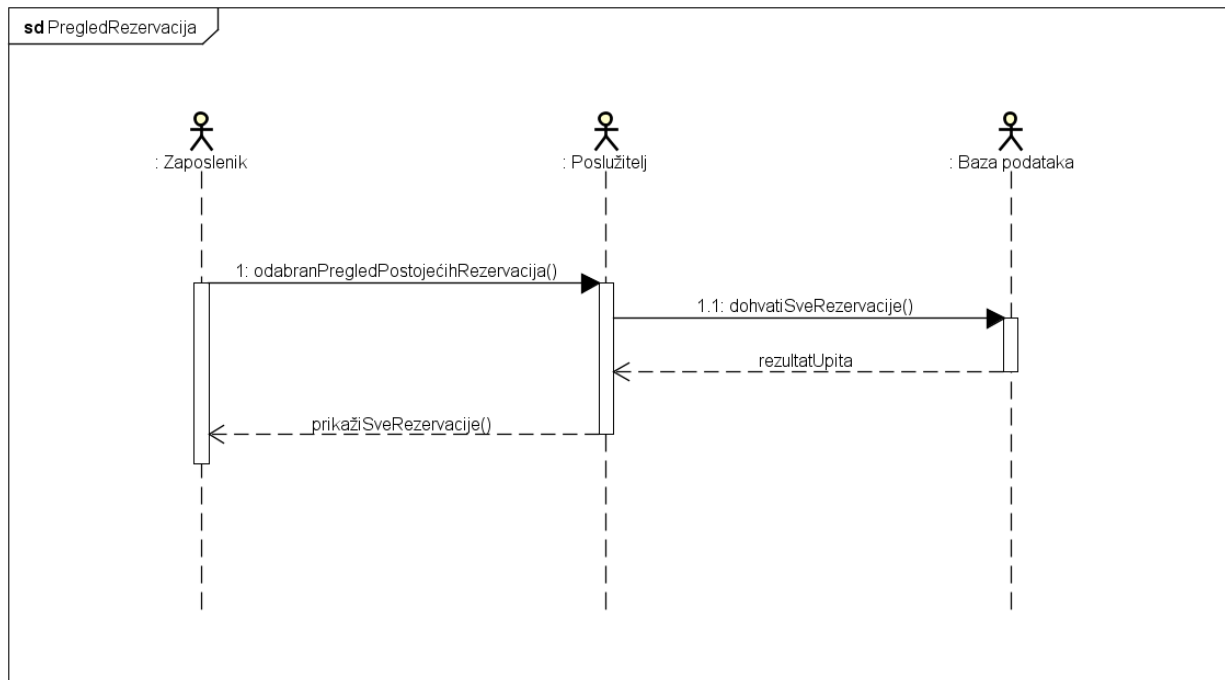
Poslužitelj svakoga dana u isto vrijeme prolazi kroz sve rezervacije u bazi podataka te traži one čiji su termini unutar iduća 24 sata. Za svaku takvu rezervaciju, u slučaju da je klijent odabrao opciju podsjetnika, poslužitelj mu na adresu elektroničke pošte šalje podsjetnik.



Slika 4.18 - Sekvencijski dijagram za UC15 (SlanjePodsjetnika)

Obrazac uporabe UC16 (PregledRezervacija)

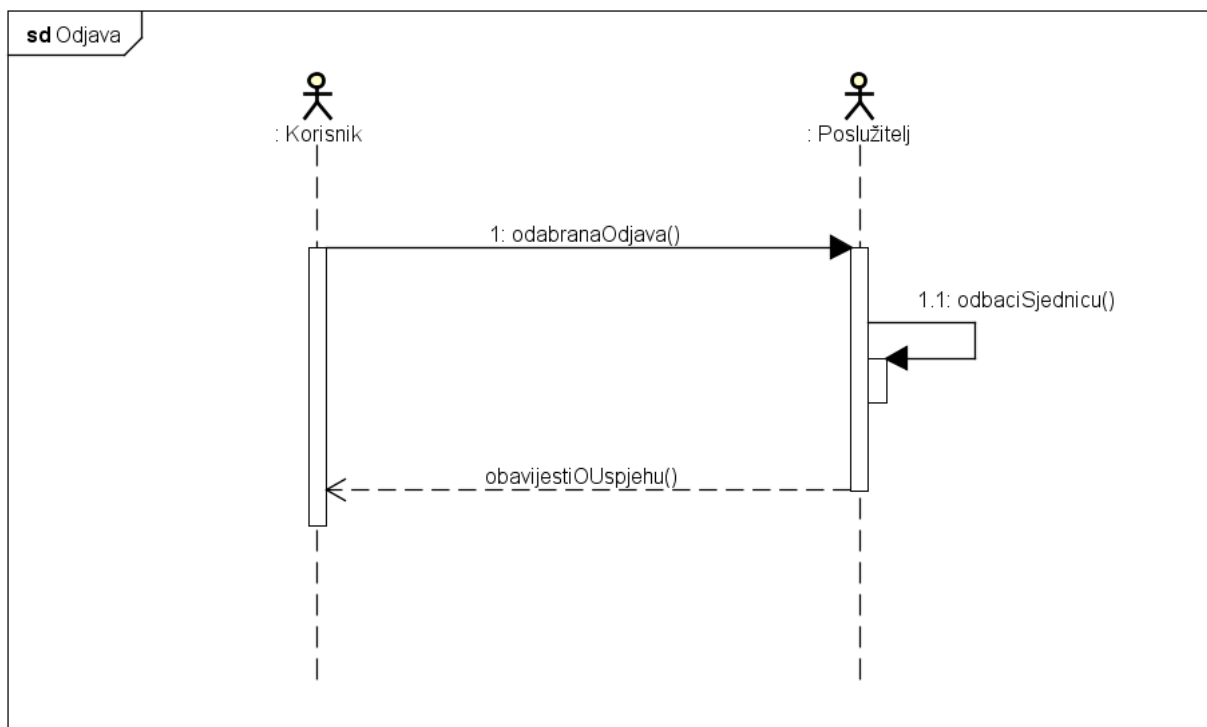
Zaposlenik otvara pregled rezervacija. Poslužitelj dohvaća sve aktivne rezervacije iz baze podataka i prikazuje ih zaposleniku. Rezervacije su podijeljene u tri kategorije (stupca). Prvu kategoriju čine sve rezervacije koje još nitko nije preuzeo. Druga su kategorija usluge koje je dotični zaposlenik preuzeo, ali nisu potvrđene od strane administratora. Treću kategoriju čine usluge koje je dotični zaposlenik preuzeo i potvrđene su od strane administratora.



Slika 4.19 - Sekvencijski dijagram za UC16 (PregledRezervacija)

Obrazac uporabe UC17 (Odjava)

Korisnik odabire odjavu nakon čega poslužitelj briše trenutnu sjednicu i odjavljuje korisnika te ga obavještava o tome.



Slika 4.20 - Sekvencijski dijagram za UC17 (Odjava)

5. Ostali zahtjevi

- Sustav treba omogućiti rad više korisnika u isto vrijeme
- Sustav treba podržavati znakove hrvatske abecede
- Sustav treba osiguravati vlastiti integritet i integritet baze podataka prilikom neispravnog korištenja

6. Arhitektura i dizajn sustava

Svrha, opći prioriteti i skica sustava

Uzevši u obzir način na koji bi poslovni subjekt trebao koristiti sustav za suradnju s krajnjim korisnicima, logičan je odabir arhitekture web aplikacije. Korisniku mora biti ponuđen jednostavan, brz i efikasan način dolaska do željene usluge. Potonji ne smije pred korisnika postaviti kompliciran postupak instalacije, a korištenje mora biti intuitivno. Osim korisničkog iskustva, važno je imati na umu da održavanje sustava mora biti jeftino, a nadogradnja jednostavna.

Cijeli sustav radi po principu arhitekture klijent-poslužitelj s većim naglaskom na poslu koji obavlja poslužitelj. Glavne podsustave arhitekture čine:

- **web aplikacija** - zaslužna za korisničko sučelje, logiku i pristup bazi podataka
- **baza podataka** - zadužena za pohranu informacija nužnih za ispravan rad sustava
- **web poslužitelj** - koristeći web aplikaciju obrađuje korisničke zahtjeve

Web poslužitelj

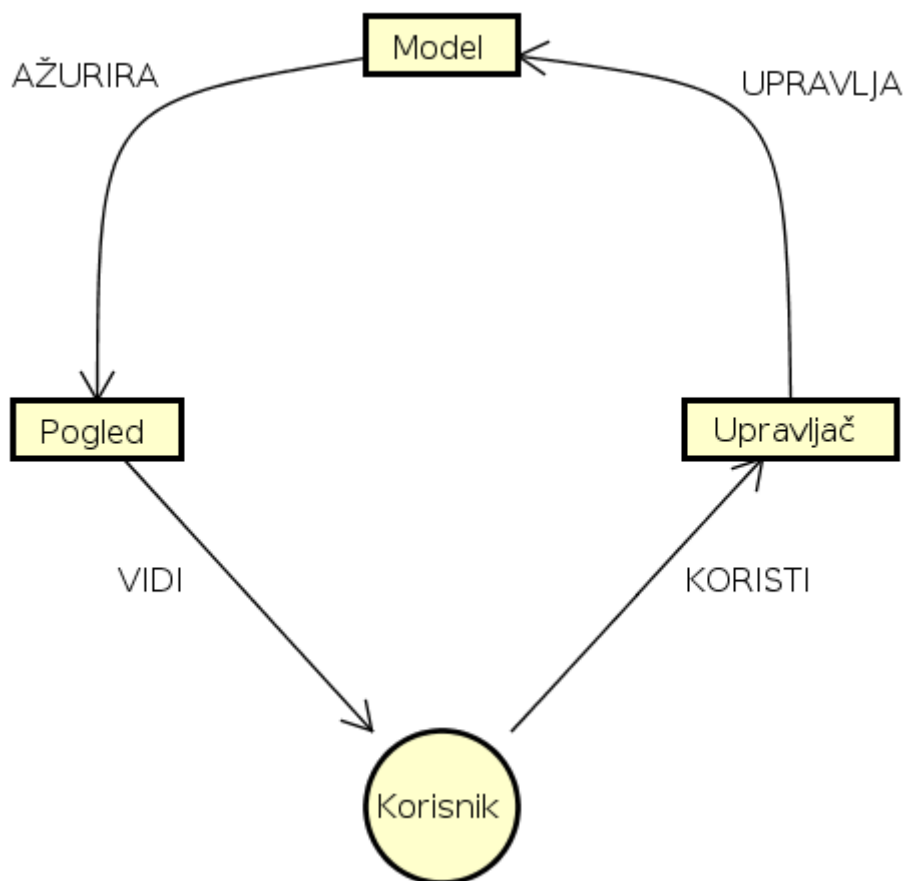
Web poslužitelj naziv je za računalo na kojem se sustav nalazi. Na njemu se nalazi baza podataka te je na njemu sama web aplikacija pokrenuta. Poslužitelj s klijentom komunicira protokolom *HTTP*. Krajnji korisnik s klijentskog računala šalje zahtjeve, a poslužitelj ih obrađuje i šalje odgovore.

Web aplikacija

Web aplikaciju čini program koji se vrti na poslužitelju, a zadužen je za obradu zahtjeva korisnika. Osim s korisnikom, web aplikacija komunicira i s bazom podataka. S obzirom na važnost kvalitete sloja aplikacije prezentiranog krajnjem korisniku aplikacije za arhitekturu aplikacije odabran je **obrazac MVC** (engl. *Model-View-Controller*) zahvaljujući kojem je izrada korisničkog sučelja jednostavna, testiranje i nadogradnja olakšane, a komunikacija između komponenata aplikacije brza. Obrazac MVC razdvaja dijelove aplikacije po dužnostima u tri kategorije:

- **Model** - dio aplikacije koji predstavlja podatkovne cjeline. Obuhvaća strukture koje razmjenjuju *upravitelj* i baza podataka, ali i strukture koje *upravitelj* prikazuje kroz *pogled*. Ova su dva modela razdvojena kako se *pogledima* ne bi slale redundantne informacije. Važno je napomenuti da *model* ni na koji način ne smije ovisiti o *pogledu* ili *upravitelju* već služi isključivo za reprezentaciju podataka.
- **Pogled** - dio aplikacije kojem korisnik neposredno pristupa, služi za prikaz poglednog modela i interakciju s korisnikom. Ima ulogu spojnice između korisnika i aplikacije, dijelovi *pogleda* s ostatkom aplikacije komuniciraju preko *upravitelja*
- **Upravitelj** - dio aplikacije koji interpretira akcije korisnika, poziva odgovarajuće *pogled*e te im prosljeđuje potrebne podatke koje moraju prikazati.

Glavna prednost MVC uzorka jest razdvajanje podatkovnog (*model*) i prezentacijskog (*pogled*) dijela umetanjem *upravitelja* između njih što povećava mogućnost ponovnog korištenja ranije napisanog koda te vrlo jasno razdvaja funkcionalnosti i dužnosti aplikacije (što je za razvoj u timu izuzetno bitno).



Slika 6.1 - način rada obrasca MVC

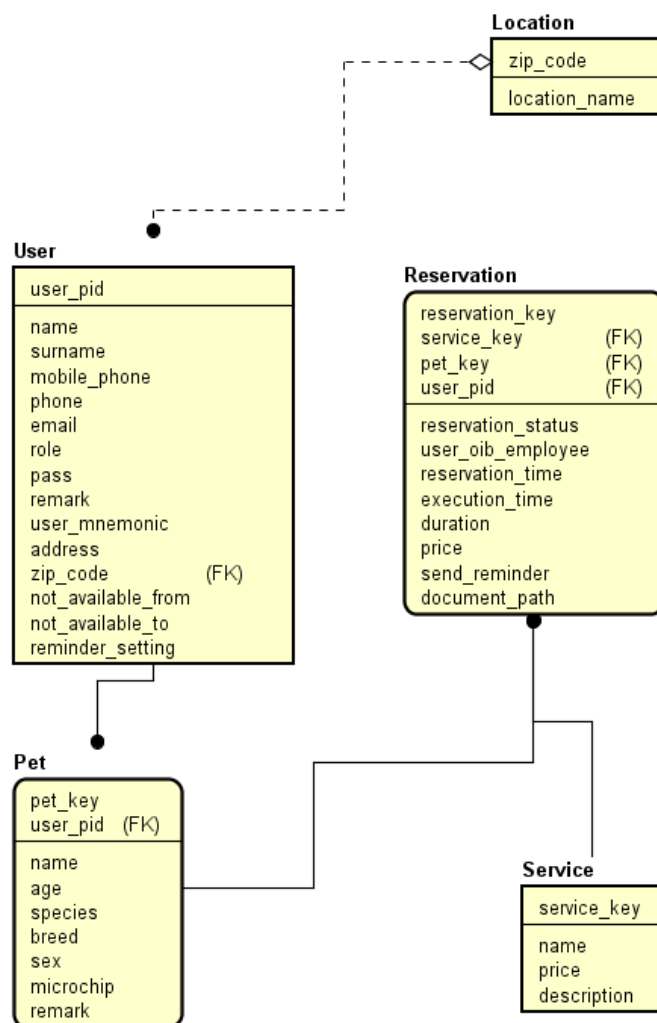
Baza podataka

Kako bi sustav uspješno funkcionirao nužna je pohrana svih podataka koje koristi web aplikacija te u tu svrhu koristimo relacijsku bazu podataka. Tablice od kojih se sastoji baza podataka ovog sustava su sljedeće:

- **User**
 - Ovaj entitet sadrži informacije o korisnicima aplikacije. Attribute ovog entiteta možemo promatrati kao tri cjeline. Prva grupa atributa odnosi se na osnovne informacije o fizičkoj osobi nužne za jednoznačnu identifikaciju. Tu spadaju osobni identifikacijski broj osobe, ime, prezime, adresa elektroničke pošte te lozinka, a važan dio ove cjeline jest i atribut uloga koji služi za upravljanje ulogama i dozvolama unutar sustava te određuje radi li se o klijentu, zaposleniku ili administratoru. Druga cjelina sastoji se od atributa nužnih za komunikaciju i vremensko-prostornu organizaciju podjele posla kao što su adresa stanovanja, poštanski broj, broj telefona, itd. U treću cjelinu spadaju atributi koji govore o preferencijama korisnika, to jest atributi napomena i postavke elektroničke pošte, atributi kojima su određeni početak i kraj radnog vremena zaposlenika te atribut kojim zaposlenik odabire hoće li elektroničkom poštom primati obavijesti za sve poslove, samo za poslove u kojima je naveden kao preferirani odabir ili nikada.
- **Pet**
 - Ovaj entitet sadrži informacije o kućnim ljubimcima te je usko vezan uz entitet User preko atributa osobni identifikacijski broj vlasnika. Riječ je o *jedan-na-mnogo* vezi, odnosno korisnik može imati više ljubimaca, a ljubimac nužno ima jednog vlasnika. Osim toga ovaj entitet sadrži i identifikacijske attribute kao što su

ime, vrsta, pasmina, dob i spol ljubimca te attribute mikročip i napomena koji daju dodatne informacije o ljubimcu.

- **Service**
 - Ovaj entitet predstavlja usluge koje poslovni subjekt nudi svojim klijentima. Osim atributa identifikacijski broj usluge sastoji se i od atributa tip usluge, cijena usluge i opis usluge koji daju sve informacije o ponuđenim uslugama. Ovaj entitet vezan je uz entitet *Reservation* *jedan-na-mnogo* vezom jer jedna rezervacija nužno ima jednu uslugu, a jedna usluga može biti vezana za više rezervacija.
- **Reservation**
 - Ovaj entitet predstavlja konkretnu uslugu za nekog klijenta, to jest njegovog ljubimca, a sastoji se od atributa poput statusa rezervacije, identifikacijskog broja ljubimca, osobnog identifikacijskog broja zaposlenika zaduženog za izvršenje ove usluge, vremena kada je rezervacija stvorena, vremena kada bi usluga trebala biti izvršena, vremena trajanja usluge te cijene. Iako je cijena usluge već određena u entitetu *Service*, nužno je postojanje ovog atributa kako bi bili pokriveni slučajevi individualnog popusta i sličnog.
- **Location**
 - Ovaj entitet sadrži attribute poštanski broj i naziv mjesta.



Slika 6.2 - ER model baze podataka

Dijagrami razreda

Podjela razreda u pakete napravljena je u skladu s MVC obrascem:

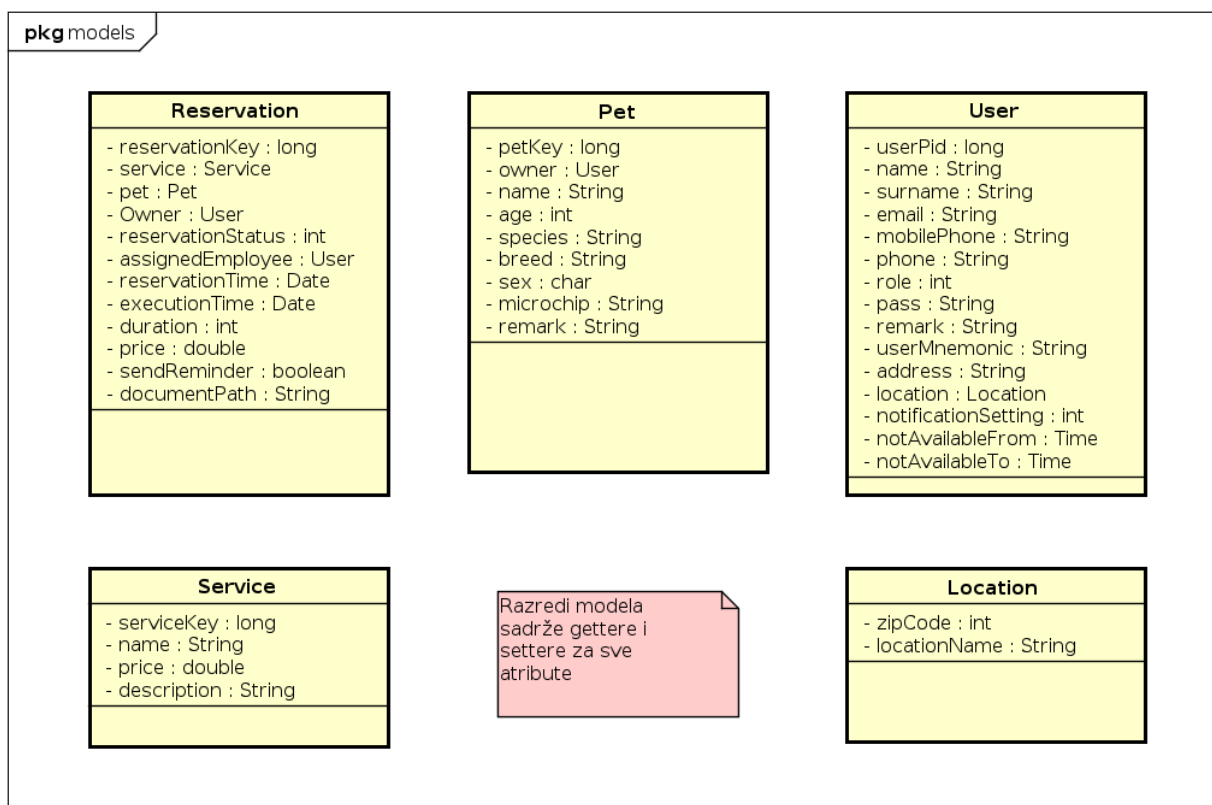
- **Razredi modela**
- **Razredi upravitelja** - obrađuju HTTP zahtjeve i rukuju pogledima
- **Repozitoriji** - vrše komunikaciju s bazom podataka
- **Pomoćni razredi** - koriste ih razredi upravitelja u obradi zahtjeva

Razredi modela

Dužnost razreda modela opisivanje je entiteta iz baze podataka (engl. *OR mapping*). U ovaj paket spadaju sljedeći razredi:

- **User** - predstavlja entitet *User*
- **Location** - predstavlja entitet *Location*
- **Pet** - predstavlja entitet *Pet*
- **Service** - predstavlja entitet *Service*
- **Reservation** - predstavlja entitet *Reservation*

Detaljan opis entiteta nalazi se u prethodnom poglavlju

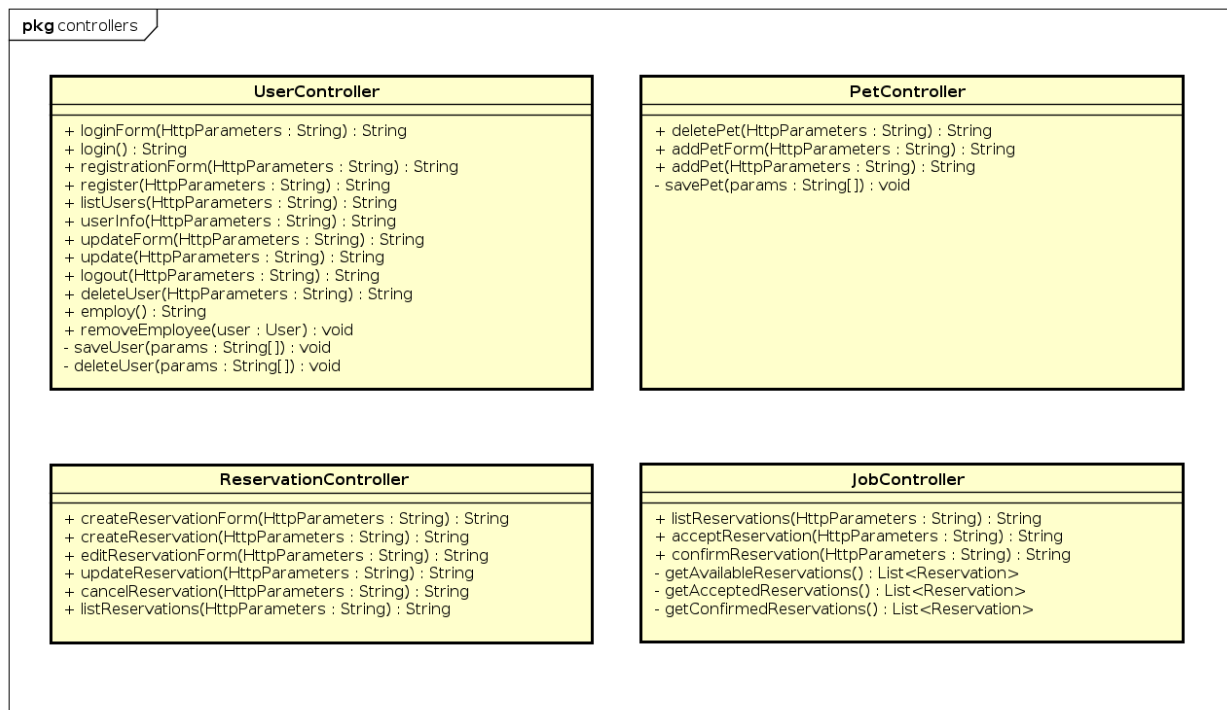


Slika 6.3 - Detaljan UML dijagram razreda modela

Razredi upravitelja

Razredi upravitelja obrađuju HTTP zahtjeve te šalju odgovore/rukuju pogledima. U ovaj paket spadaju sljedeći razredi:

- **UserController**
 - Rukuje svim funkcionalnostima koje su vezane s profilom korisnika: registracija, prijava, brisanje korisnika, zapošljavanje korisnika, uređivanje korisničkih podataka, odjava.
- **PetController**
 - Rukuje svim funkcionalnostima koje su vezane s kućnim ljubimcima: dodavanje ljubimaca, brisanje ljubimaca.
- **ReservationController**
 - Rukuje svim funkcionalnostima koje su vezane sa stvaranjem, uređivanjem i pregledom rezervacija iz perspektive klijenta.
- **JobController**
 - Rukuje svim funkcionalnostima vezanima uz obavljanje poslova. Drugim riječima, pruža pogled na rezervacija i mogućnosti rukavanja iz perspektive zaposlenika



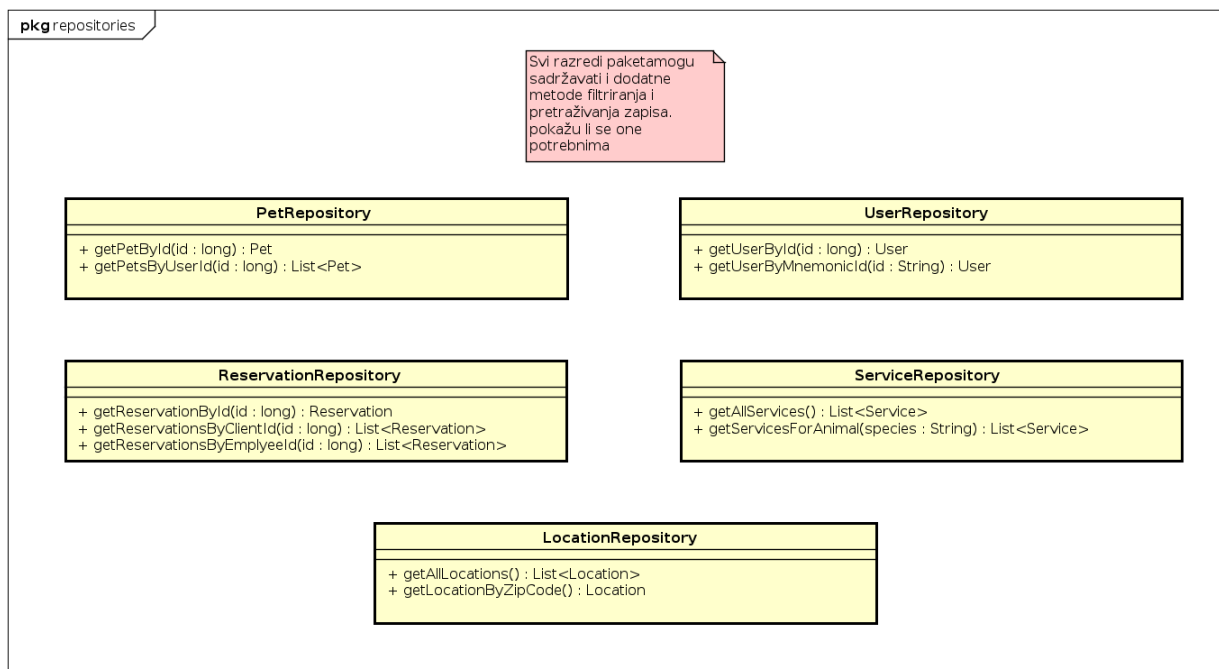
Slika 6.4 - Detaljan UML dijagram razreda upravitelja

Repozitoriji

Dužnost ovog paketa komunikacija je s bazom podataka

- **UserRepository** - rukuje entitetom *User*
- **LocationRepository** - rukuje entitetom *Location*
- **PetRepository** - rukuje entitetom *Pet*
- **ReservationRepository** - rukuje entitetom *Reservation*
- **ServiceRepository** - rukuje entitetom *Service*

Repozitoriji razredima upravitelja predstavljaju sučelje za komunikaciju s bazom podataka. Kao strukture podataka u toj komunikaciji koriste se razredi modela.

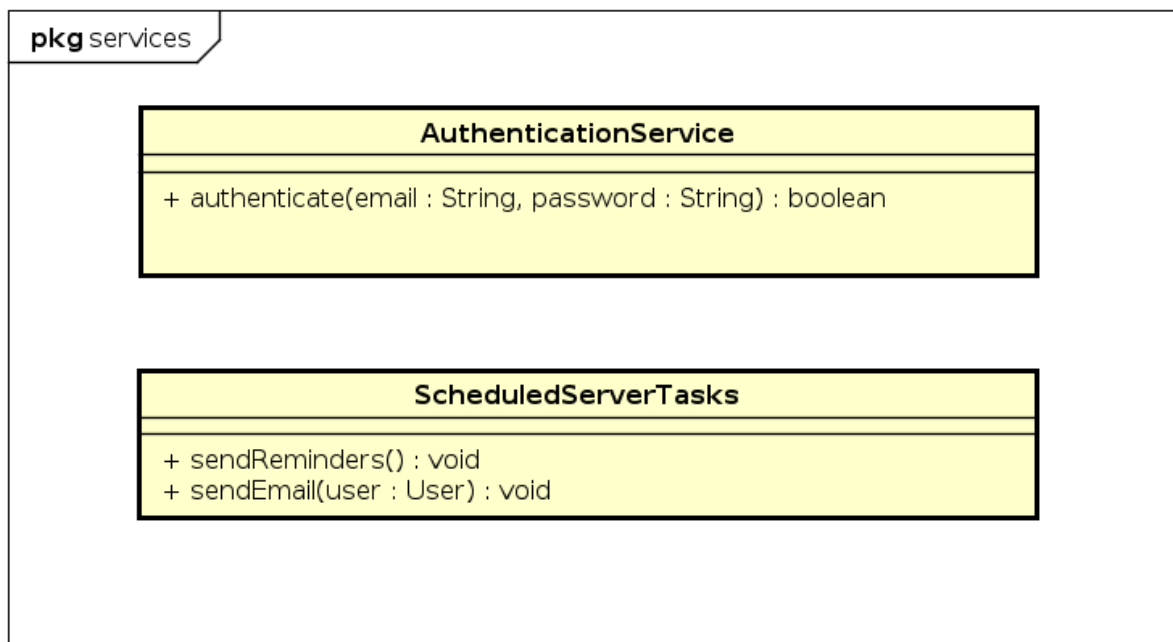


Slika 6.5 - Detaljan UML dijagram repozitorija

Pomoćni razredi

U ovom paketu nalaze se razredi koji implementiraju sve dodatne funkcionalnosti potrebne razredima upravitelja. Paket čine:

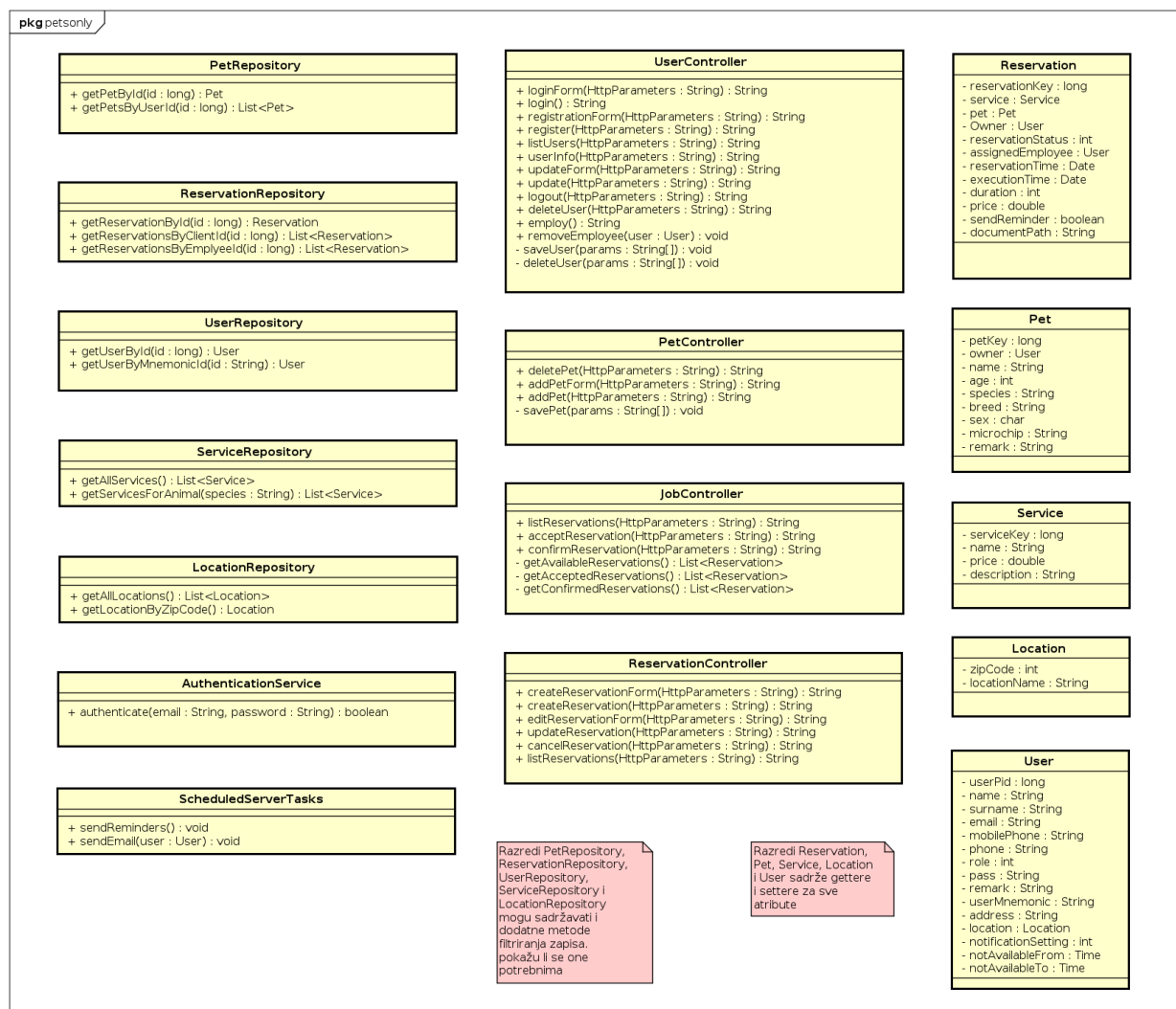
- ScheduledServerTasks - periodički obavlja zadatke poslužitelja
- AuthentiactionService - pruža usluge autentifikacije i autorizacije



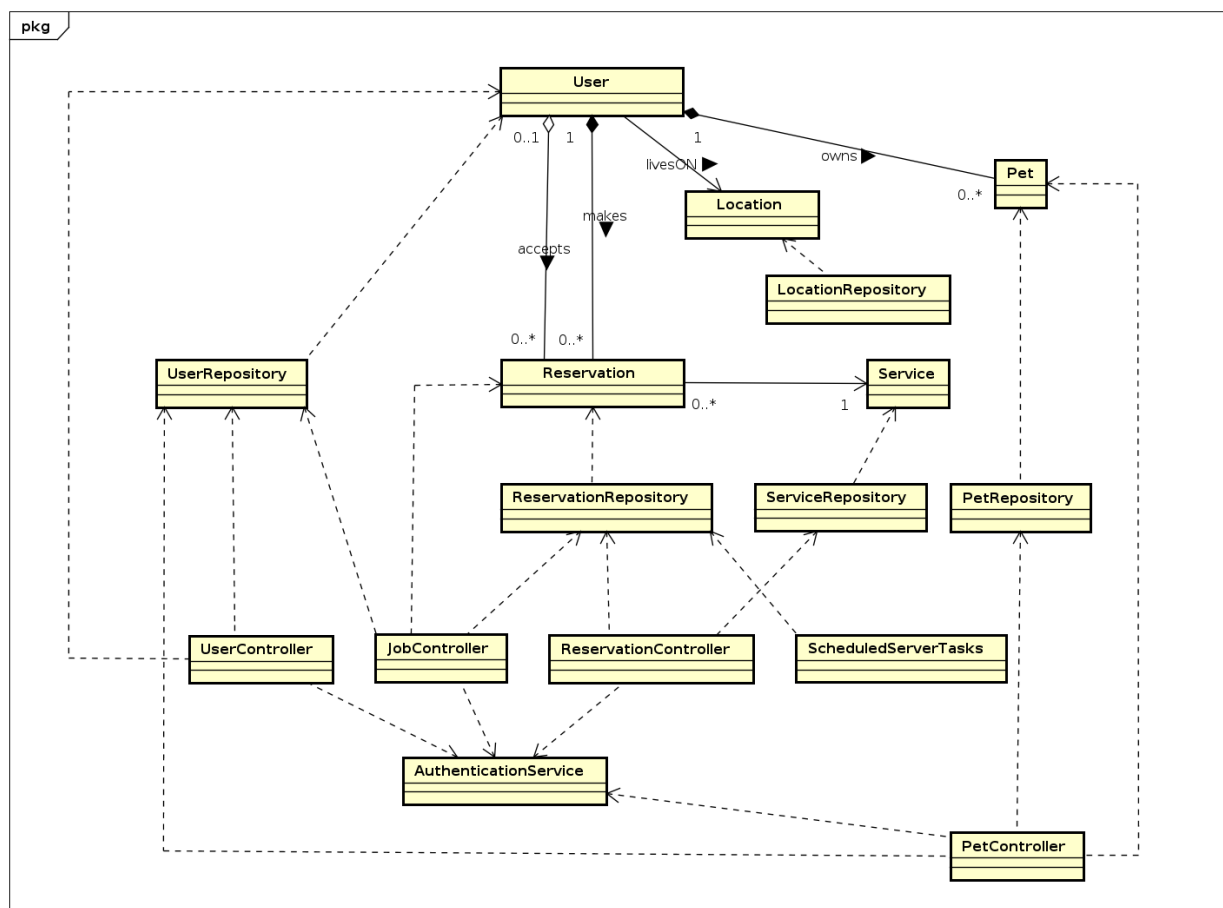
Slika 6.6 - Detaljan UML dijagram pomoćnih razreda

Pregled svih paketa

Uspješan rad web aplikacije temelji se na dobroj suradnji svih paketa i međusobnom dijeljenju ključnih funkcionalnosti.



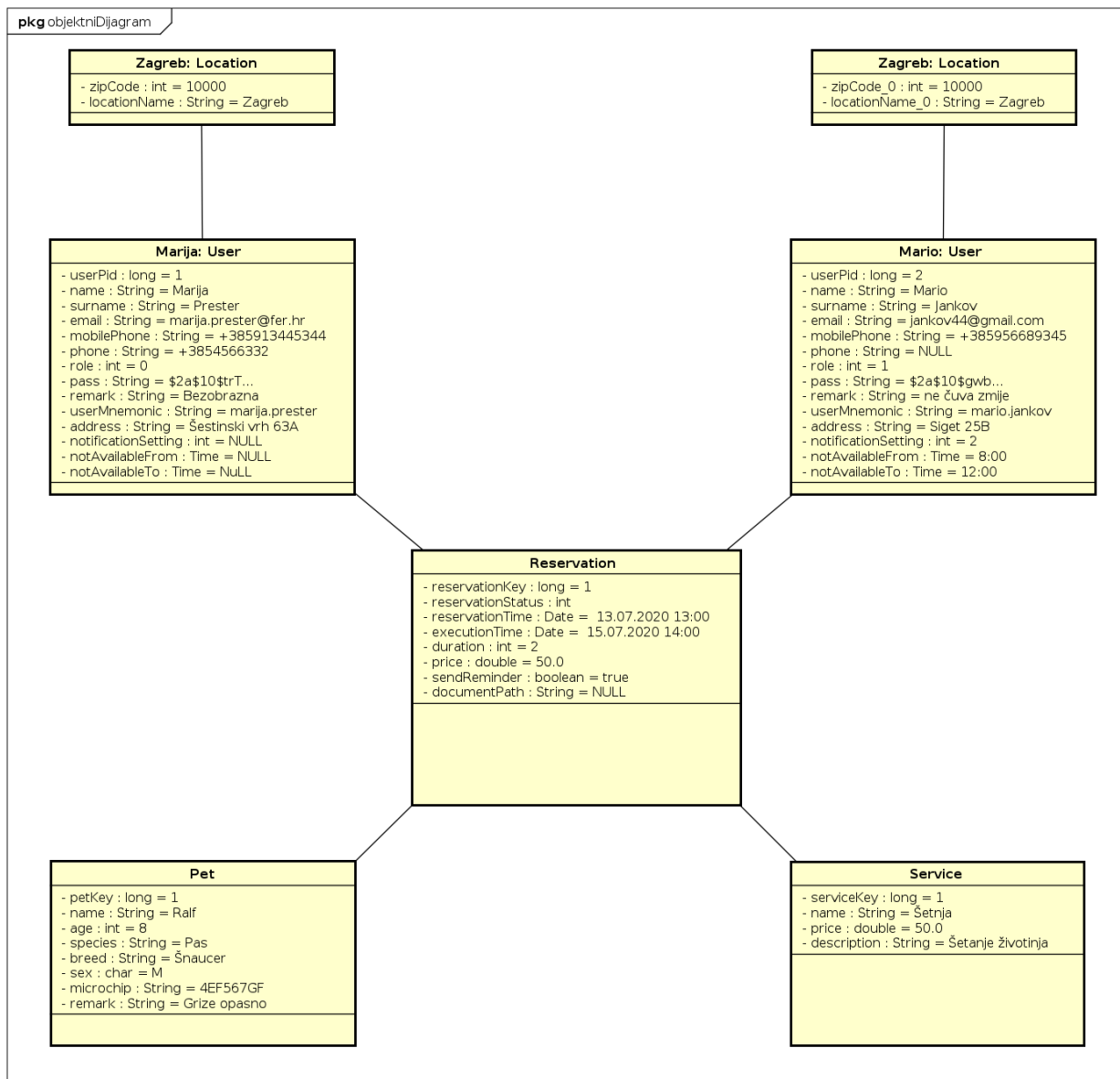
Slika 6.7 - Detaljan UML dijagram svih razreda



Slika 6.8 - UML dijagram odnosa između svih razreda

Dijagram objekata

Dijagram prikazuje odnos objekata u sustavu nakon što rezervaciju, koju je za svog psa Ralfa napravila klijentica Marija Prester, preuzme zaposlenik Mario Jankov.



Slika 6.9 - UML dijagram objekata

7. Implementacija i korisničko sučelje

8. Zaključak i budući rad

U sklopu izrađivanja prve verzije dokumentacije izradili smo opis projektnog zadatka, a sukladno njemu smo napravili i opise obrazaca upotrebe, dijagrame obrazaca upotrebe te sekvencijske dijagrame čime smo zaključili glavne funkcijske zahtjeve. U sklopu dizajna sustava i arhitekture odredili smo način oblikovanja web-poslužitelja, Model-View-Controller (MVC) model te smo njegov rad predložili dijagramom. Također, napravili smo i dijagrame razreda, koje smo radi preglednosti odvojili u više dijagrama te model baze podataka.

Poprilično smo dobro organizirani i rad je bio dobro raspodijeljen. Za praćenje verzija koristimo GitLab, a za komunikaciju Slack.

Budući rad se zasniva na implementaciji dosada definiranih modela, izradi korisničkog sučelja, ispravljanju nepredviđenih grešaka u modelu tijekom implementacije jer nismo još dovoljno iskusni kako bismo mogli predvidjeti tok sustava dovoljno detaljno, te testiranju sustava. Također, sljedeći je korak izrada ostalih UML dijagrama (dijagrami aktivnosti, stanja, komponenti, komunikacijski...).

Popis literature

- [1] Oblikovanje programske potpore, FER ZEMRIS, <http://www.fer.hr/predmet/opp>
- [2] Interna skripta "Procesi programskog inženjerstva",
http://www.fer.unizg.hr/predmet/opp/nastavni_plan/literatura
- [3] Spring MVC Tutorial for Beginners, <https://www.youtube.com/watch?v=BjNhGaZDroY>
- [4] Apache Tomcat 7, <http://tomcat.apache.org/tomcat-7.0-doc>

Dodatak A: Index

- Slika 4.1 - Cjeloviti pregled svih obrazaca upotrebe
- Slika 4.2 - Obrasci vezani za interakciju sa korisnicima
- Slika 4.3 - Obrasci vezani za rezervacije
- Slika 4.4 - Sekvencijski dijagram za UC1 (Registracija)
- Slika 4.5 - Sekvencijski dijagram za UC2 (DodavanjeZaposlenika)
- Slika 4.6 - Sekvencijski dijagram za UC3 (Prijava)
- Slika 4.7 - Sekvencijski dijagram za UC4 (PregledIUređivanjePodataka)
- Slika 4.8 - Sekvencijski dijagram za UC5 (UklanjanjeZaposlenika)
- Slika 4.9 - Sekvencijski dijagram za UC6 (BrisanjeKorisnika)
- Slika 4.10 - Sekvencijski dijagram za UC7 (DodavanjeLjubimca)
- Slika 4.11 - Sekvencijski dijagram za UC8 (BrisanjeLjubimca)
- Slika 4.12 - Sekvencijski dijagram za UC9 (StvaranjeRezervacije)
- Slika 4.13 - Sekvencijski dijagram za UC10 (UređivanjeRezervacije)
- Slika 4.14 - Sekvencijski dijagram za UC11 (PraćenjeRezervacije)
- Slika 4.15 - Sekvencijski dijagram za UC12 (PrihvatanjeRezervacije)
- Slika 4.16 - Sekvencijski dijagram za UC13 (PotvrdaPlaćeneRegistracije)
- Slika 4.17 - Sekvencijski dijagram za UC14 (OtkazivanjeRezervacije)
- Slika 4.18 - Sekvencijski dijagram za UC15 (SlanjePodsjetnika)
- Slika 4.19 - Sekvencijski dijagram za UC16 (PregledRezervacija)
- Slika 4.20 - Sekvencijski dijagram za UC17 (Odjava)
- Slika 6.1 - način rada obrasca MVC
- Slika 6.2 - ER model baze podataka
- Slika 6.3 - Detaljan UML dijagram razreda modela
- Slika 6.4 - Detaljan UML dijagram razreda upravitelja
- Slika 6.5 - Detaljan UML dijagram repozitorija
- Slika 6.6 - Detaljan UML dijagram pomoćnih razreda
- Slika 6.7 - Detaljan UML dijagram svih razreda
- Slika 6.8 - UML dijagram odnosa između svih razreda
- Slika 6.9 - UML dijagram objekata

Dodatak B: Dnevnik sastanaka

Datum	Sudionici	Sadržaj sastanka
7.10.2017.	Sodić Paulinović Modrušan Krmpotić-Đurđević Galić	Definiranje projektnog zadatka
13.10.2017.	Sodić Paulinović Modrušan Krmpotić-Đurđević Kravrščan	Osnovne funkcionalnosti aplikacije i use caseovi
22.10.2017.	Sodić Modrušan	Osnovne funkcionalnosti html-a, css-a i javascripta
23.10.2017	Sodić Kravrščan Galić	Izrada funkcionalnih zahtjeva (obraci uporabe definirani)
25.10.2017	Sodić Krmpotić-Đurđević Paulinović	Izrada sekvencijskih dijagrama i ER modela baze podataka
26.10.2017	Sodić	Konzultacije s profesorom

Datum	Sudionici	Sadržaj sastanka
29.10.2017	Modrušan Sodić Krmpotić-Đurđević Kravrščan Galić Paulinović	Dogovor oko završetka prve verzije dokumentacije
3.11.2017	Sodić	Konzultacije s profesorom
17.11.2017	Sodić Kravrščan Galić Paulinović	Konzultacije s profesorom i provjera finalne verzije dokumentacije
17.11.2017	Sodić Kravrščan Galić Paulinović	Završavanje i dorada prve verzije dokumentacije

Dodatak C: Prikaz aktivnosti grupe

Popis aktivnosti	Antun Modrušan	Filip Sodić	Maja Krmpotić-Đurđević	Mate Paulinović	Tomislav Kravaršćan	Vlado Galić
Dnevnik promjena dokumentacije	0%	0%	0%	0%	0%	100%
Upravljanje projektom	—	—	—	—	—	—
Opis projektnog zadatka	15%	45%	10%	8%	7%	10%
Rječnik pojmova	80%	3%	0%	0%	0%	17%
Opis funkcionalnih zahtjeva	0%	15%	0%	40%	35%	10%
Opis ostalih zahtjeva	0%	0%	0%	100%	0%	0%
Arhitektura i dizajn sustava						
Svrha, opći prioriteti i skica sustava	0%	10%	90%	0%	0%	0%
Dijagram razreda s opisom	0%	100%	0%	0%	0%	0%
Dijagram objekata	0%	100%	0%	0%	0%	0%
Ostali UML	—	—	—	—	—	—

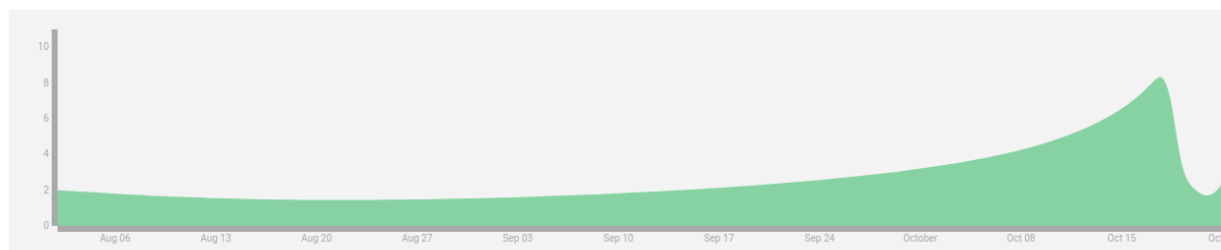
Popis aktivnosti	Antun Modrušan	Filip Sodić	Maja Krmpotić-Đurđević	Mate Paulinović	Tomislav Kravaršćan	Vlado Galić
dijagrami						
Implementacija i korisničko sučelje	—	—	—	—	—	—
Dijagram razmještaja	—	—	—	—	—	—
Korištene tehnologije i alati	—	—	—	—	—	—
Isječak programskog koda	—	—	—	—	—	—
Ispitivanje programskog rješenja	—	—	—	—	—	—
Upute za instalaciju	—	—	—	—	—	—
Korisničke upute	—	—	—	—	—	—
Zaključak i budući rad	0%	0%	0%	0%	100%	0%
Popis literature	100%	0%	0%	0%	0%	0%
Dodatci						
Indeks	0%	0%	80%	0%	0%	20%

Popis aktivnosti	Antun Modrušan	Filip Sodić	Maja Krmpotić-Đurđević	Mate Paulinović	Tomislav Kravaršćan	Vlado Galić
Dnevnik sastajanja	0%	0%	0%	100%	0%	0%
Prikaz aktivnosti grupe	0%	30%	0%	0%	70%	0%
Plan rada	0%	0%	0%	0%	0%	100%
Pregled rada i stanje ostvarenja	0%	0%	0%	0%	0%	100%

Grafički prikaz aktivnosti

August 2 2017 - November 17 2017

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



Dodatak D: Plan rada i ostvarenje

Ostvareno u rev. 1:

Dokumentacija:

- Opis projektnog zadatka
- Funkcionalni zahtjevi
- Ostali zahtjevi
- Dnevnik sastanaka
- Dnevnik promjene dokumentacije
- Arhitektura i dizajn sustava

Implementacija:

- Inicijalno stvaranje *Maven* projekta
- Povezivanje okoline s Gitlab-om

Plan za rev. 2:

Dokumentacija:

- Implementacija i korisničko sučelje
- Zaključak i budući rad
- Popis literature
- Pregled i konačna dorada cijele dokumentacije

Implementacija:

- Svih glavnih i sporednih dijelova sustava
- Konačno ispitivanje funkcionalnosti i ispravnosti sustava