

Oblikovanje programske potpore

Ak. God. 2017./2018.

Upravljanje stambenom zgradom

Dokumentacija, Rev. 1

Grupa: *Eureka*

Voditelj: *Toni Martinčić*

Datum predaje:

Mentor: *Katarina Potkonjak*

Popis članova grupe i zadatci:

Toni Martinčić – voditelj projekta, koordinacija članova projekta, kontrola kvalitete rada, dizajn UML dijagrama, arhitektura sustava

Ivana Mršić – arhitektura sustava, UML dijagrami, baza podataka

Marko Ćurlin – arhitektura sustava, UML dijagrami, baza podataka

Marko Cavalli – arhitektura sustava, UML dijagrami, baza podataka

Luka Hrgović – arhitektura sustava, UML dijagrami, baza podataka

Ivan Biškup – arhitektura sustava, UML dijagrami, baza podataka

Sadržaj:

Sadržaj:	3
1. Dnevnik promjena dokumentacije	4
2. Opis projektnog zadatka	5
3. Pojmovnik	7
4. Funkcionalni zahtjevi	8
5. Ostali zahtjevi	33
6. Arhitektura i dizajn sustava	34
6.1. Svrha, opći prioriteti i skica sustava	34
6.2. Dijagram razreda s opisom	37
6.3. Dijagram objekata	46
7. Implementacija i korisničko sučelje	47
8. Zaključak i budući rad	48
9. Popis literature	49
Dodatak A: Indeks (slika, dijagrami, tablica, ispis koda)	50
Dodatak B: Dnevnik sastajanja	51
Dodatak C: Prikaz aktivnosti grupe	53
Dodatak D: Plan rada / Pregled rada i stanje ostvarenja	55

1. Dnevnik promjena dokumentacije

Rev.	Opis promjene/dodatka	Autor(i)	Datum
0.1	Napravljen predložak. Dodan opis projektnog zadatka, popis članova i zadataci.	Martinčić, Mršić	30.10.2017.
0.2	Dodan pojmovnik.	Cavalli	30.10.2017.
0.3	Dodani funkcionalni zahtjevi.	Mršić, Hrgović	30.10.2017.
0.31	Dodani ostali zahtjevi.	Biškup, Ćurlin	30.10.2017.
0.4	Dodan popis literature.	Martinčić	31.10.2017.
0.5	Opis obrazaca uporabe.	Biškup, Ćurlin	4.11.2017.
0.51	Dodan obrazac uporabe.	Ćurlin	11.11.2017.
0.6	Dodani sekvencijski dijagrami.	Biškup, Hrgović, Mršić	11.11.2017.
0.7	Svrha, opći prioriteti i skica sustava.	Biškup, Martinčić	11.11.2017.
0.8	Dijagram razreda s opisom.	Cavalli	11.11.2017.
0.9	Dijagram objekata.	Hrgović	12.11.2017.
1.0	Ispravljene pogreške kod sekvencijskih dijagrama i pravopisne pogreške.	Biškup, Martinčić	17.11.2017.

2. Opis projektnog zadatka

Cilj projekta je praktično primijeniti postupke oblikovanja programske potpore na rješavanje konkretnih problema upravljanja stambenim zgradama. Također izraditi projektnu dokumentaciju i implementaciju svih zadataka, te omogućiti preglednost i dostupnost svih funkcionalnosti i sadržaja.

Funkcionalnosti koje se žele postići su jednostavnija i lakša komunikacija između samih stanara i svih ostalih zainteresiranih stranaka. Također osigurati mogućnost objavljivanja svih stranaka na oglasnoj ploči. Same objave bi imale vrijeme nastanka i mogući istek objave, odnosno vrijeme kada bi se objava maknula sa oglasne ploče, ako korisnik tako naloži kod unosa. Isto tako korisnik bi mogao maknuti vlastitu objavu u bilo kojem trenutku. Druga funkcionalnost bi bila raspored čišćenja snijega gdje bi korisnik mogao vidjeti svoj termin čišćenja, i ako mu možda isti ne odgovara, mogao bi predložiti zamjenu stanaru čiji mu termin odgovara. Isto tako treba omogućiti generiranje naloga, te svim stanarima omogućiti pregled tijeka zajedničkog novca u bilo kojem trenutku, te pregled vlastitih uplata i svih plaćenih i ne plaćenih pričuva. Omogućiti svim stanarima prijavljivanje određene štete, gdje bi imali podjelu na hitne štete koje bi se automatski provodile i manje hitne gdje bi za to trebala suglasnost predstavnika suvlasnika. Još neke od funkcionalnosti bi bile dogovaranje sastanaka putem aplikacije i nekih zajedničkih druženja i sastanaka. Potrebno je napomenuti da će svaki korisnik moći obavljati samo one akcije unutar aplikacije koje će mu biti unaprijed predodređene privilegijama.

Predviđa se izrada web aplikacije koja bi se sastojala od početne stranice gdje bi se korisnici prijavljivali pomoću svojih korisničkih imena i lozinki. Nakon prijave bi se korisniku otvorio ekran sa oglasnom pločom gdje bi mogao vidjeti sve obavijesti koje su navedene u gornjem odlomku, te isto tako dodavati nove. Korisnik bi također mogao otvoriti stranicu sa nalogima gdje bi imao uvid u sve naloge zgrade i stanara. Treća stranica bi predstavljala troškove gdje bi svaki korisnik mogao pratiti tijek novca i troškove zgrade. Te bi zadnji ekran predstavljao mogućnost pregleda osobnih podataka korisnika koje bi mogao promijeniti, podatke zgrade u kojoj stanuje ili posjeduje stan, te bi imao mogućnost uključiti ili isključiti notifikacije.

Sve akcije korisnika, koje su nepredviđene i ne ispravne su pokrivene unutar sustava.

Sustav će osim stanara kao korisnika, imati još nekoliko različitih tipova korisnika:

1. *Predstavnik stanara* – osoba koja ima sve ovlasti kao i obični stanar, ali ima mogućnost određivanja prioriteta obavljanja određenih poslova u zgradi, kao i potvrđivanje naloga koji su unešeni u sustav, te bi imao mogućnost brisanja svih objava i naloga po potrebi.

2. *Upravitelj zgrade* - ovlaštena firma koja vodi brigu o financijama zgrade i slanju osoba za određene popravke na samoj zgradi, te o izvršavanju naloga, odnosno provođenju i plaćanju obavljenih naloga, te potvrđivanju uplaćenih pričuva.

3. *Administrator* – osoba koja se brine o samoj stranici i promjenama na njoj. Isto tako ima mogućnost dodavanja i brisanja novih korisnika i dodjeljivanje određenih privilegija određenim osobama, što je u ovom slučaju predstavnik stana i upravitelj zgrade.

3. Pojmovnik

Apache Tomcat: open source aplikacijski web poslužitelj za pokretanje i prikaz web stranica.

CSS (Cascading Style Sheets): stilski jezik koji se rabi za opis prezentacije dokumenta napisanog pomoću HTML jezika.

Git: distribuirani sustav za upravljanje izvornim kodom.

Hibernate: objektno/relacijska tehnologija koja povezuje objekte iz Jave s realacijskim tablicama baze podataka.

HTML (HyperText Markup Language): prezentacijski jezik za izradu web stranica.

Intellij IDEA: Java integrirano razvojno okruženje za razvoj računalnog softvera.

Java: objektno orijentirani programski jezik koji se može koristiti za izradu web aplikacija.

JavaScript: skriptni programski jezik, koji se izvršava u web pregledniku na strani korisnika.

React: biblioteka JavaScript programskog jezika koji služi za izradu korisničkog sučelja.

Redux: spremnik koji sadrži stanje JavaScript aplikacije.

Spring Boot: olakšava stvaranje samostalnih aplikacija koje se temelje na Spring proizvodima koje možete „samo pokrenuti“.

SQL: standardni programski jezik za manipuliranje, pohranu i preuzimanje podataka u bazama podataka.

UML (Unified Modeling Language): standardizirani jezik za slikovno predstavljanje i modeliranje objekata.

4. Funkcionalni zahtjevi

Dionici:

- Stanar
- Predstavnik stanara
- Upravitelj zgrade
- Administrator

Aktori i njihovi funkcionalni zahtjevi:

- Stanar – inicijator
 - Prijavljuje se lozinkom i korisničkim imenom
 - Prijavljuje štetu, kvar i izgubljene/pronađene stvari
 - Zatražuje promjenu termina čišćenja snijega po potrebi
 - Uvid u raspored čišćenja snijega, naloge, troškove i opće informacije o zgradi, kao i vlastite opće informacije
 - Objavljuje na oglasnoj ploči
 - Izmjenjuje vlastite obavijesti na oglasnoj ploči
 - Briše vlastite objave
- Predstavnik stanara – inicijator
 - Prijavljuje se lozinkom i korisničkim imenom
 - Prijavljuje štetu, kvar i izgubljene/pronađene stvari
 - Određuje važnost i redoslijed popravaka i radova
 - Ažurira i potvrđuje naloge
 - Zatražuje promjenu termina čišćenja snijega po potrebi
 - Objavljuje raspored čišćenja snijega
 - Uvid u raspored čišćenja snijega, naloge, troškove i opće informacije o zgradi, kao i vlastite opće informacije
 - Objavljuje na oglasnoj ploči
 - Uređuje vlastite objave na oglasnoj ploči
 - Briše objave po potrebi
- Upravitelj zgrade – inicijator
 - Prijavljuje se lozinkom i korisničkim imenom
 - Obrađuje i ažurira naloge
 - Dogovara posao sa izvođačima radova
- Administrator – inicijator
 - Upisuje i briše korisnike iz sustava
 - Upravlja ovlastima
 - Održava šifarnike

- Baza podataka – sudionik
 - Čuva podatke o svim stanarima, troškovima i nalogima
 - Čuva aktivnosti na oglasnoj ploči i raspored čišćenja snijega
 - Čuva podatke o svim zgradama i stanovima

Opis obrazaca uporabe:

- **UC1** – PrijavljivanjeKorisnika
 - **Glavni sudionik:** Stanar, Predstavnik, Upravitelj, Administrator.
 - **Cilj:** Prijaviti se u sustav kako bi mogao koristiti aplikaciju.
 - **Sudionici:** Baza podataka
 - **Preduvjeti:** Da ima korisnički račun.
 - **Rezultat:** Korisnik je prijavljen i ima pristup sadržaju i funkcijama stranice.
 - **Željeni scenarij:**
 - 1) Korisnik unosi svoje korisničko ime i lozinku.
 - 2) Sustav provodi provjeru podataka, te upućuje korisnika na početnu stranicu.
 - **Mogući drugi scenarij:**
 - 1) Korisnika nema u bazi podataka, te ga sustav vraća na stranicu za prijavu.
 - 2) Korisnik je unio krivu lozinku i korisničko ime, te ga sustav vraća na stranicu za prijavu.
- **UC2** – PogledajOglasnuPloču
 - **Glavni sudionik:** Stanar, predstavnik, upravitelj.
 - **Cilj:** Pogledati trenutno stanje na oglasnoj ploči.
 - **Sudionici:** Baza podataka.
 - **Preduvjeti:** Korisnik je prijavljen.
 - **Rezultat:** Prikaz svih objava na oglasnoj ploči.
 - **Željeni scenarij:**
 - 1) Korisnik odabire opciju za pregled oglasne ploče.
 - 2) Sustav prebacuje korisnika na oglasnu ploču, te mu prikazuje oglasnu ploču sa objavama.
 - **Mogući drugi scenarij:** -

- **UC3 – DodajNaOglasnuPloču**
 - **Glavni sudionik:** Stanar, predstavnik, upravitelj.
 - **Cilj:** Dodati nove objave na oglasnoj ploči.
 - **Sudionici:** Baza podataka.
 - **Preduvjeti:** Korisnik je ulogiran, te se nalazi na oglasnoj ploči.
 - **Rezultat:** Korisnik dodaje podatke na oglasnu ploču.
 - **Željeni scenarij:**
 - 1) Korisnik odabire opciju dodavanja objave.
 - 2) Sustav prikazuje formu za unos podataka objave.
 - 3) Korisnik unosi podatke i potvrđuje unos.
 - 4) Sustav provjerava unesene podatke, bilježi dodavanje te osvježava oglasnu ploču.
 - **Mogući drugi scenarij:**
 - 1) Korisnik odustaje od objavljivanja. Scenarij završava.
 - 2) Pogrešno su uneseni podatci, sustav odbija unos korisniku, te mu to javlja porukom.
- **UC4 – BrišiSaOglasnePloče**
 - **Glavni sudionik:** Stanar, predstavnik, upravitelj.
 - **Cilj:** Obrisati objave na oglasnoj ploči.
 - **Sudionici:** Baza podataka.
 - **Preduvjeti:** Korisnik je ulogiran, te se nalazi na oglasnoj ploči.
 - **Rezultat:** Korisnik briše podatke na oglasnoj ploči.
 - **Željeni scenarij:**
 - 1) Korisnik odabire opciju brisanja objave.
 - 2) Sustav bilježi brisanje te osvježava oglasnu ploču.
 - **Mogući drugi scenarij:** -
- **UC5 – PogledajRasporedČišćenjaSnijega**
 - **Glavni sudionik:** Stanar i predstavnik.
 - **Cilj:** Pogledati trenutno stanje na rasporedu čišćenja snijega.
 - **Sudionici:** Baza podataka.
 - **Preduvjeti:** Korisnik je prijavljen.
 - **Rezultat:** Otvara se stranica sa rasporedom čišćenja snijega.
 - **Željeni scenarij:**
 - 1) Korisnik odabire opciju za pregled rasporeda čišćenja snijega.
 - 2) Sustav prikazuje korisniku trenutno stanje rasporeda čišćenja snijega.
 - **Mogući drugi scenarij:** -

- **UC6 – ZatražioZamjenuTerminaRasporedaČišćenjaSnijega**
 - **Glavni sudionik:** Stanar i predstavnik.
 - **Cilj:** Zatražiti zamjenu termina u rasporedu čišćenja snijega.
 - **Sudionici:** Baza podataka.
 - **Preduvjeti:** Korisnik je prijavljen, te se nalazi na toj stranici.
 - **Rezultat:** Korisnik traži zamjenu termina.
 - **Željeni scenarij:**
 - 1) Korisnik odabire termin koji mu odgovara.
 - 2) Sustav bilježi zahtjev korisnika u bazi, te šalje poruku korisniku čiji je datum odabran.
 - **Mogući drugi scenarij:** -

- **UC7 – OdgovorNaZamjenuTerminaRasporedaČišćenjaSnijega**
 - **Glavni sudionik:** Stanar i predstavnik.
 - **Cilj:** Korisnik prihvaća ili odbija zamjenu u rasporedu čišćenja snijega.
 - **Sudionici:** Baza podataka.
 - **Preduvjeti:** Korisnik je prijavljen, te se nalazi na toj stranici.
 - **Rezultat:** Korisnik odgovara na zahtjev za zamjenu termina.
 - **Željeni scenarij:**
 - 1) Korisnik prihvaća zamjenu termina.
 - 2) Sustav šalje poruku o prihvaćanju zamjene korisniku koji je zatražio zamjenu.
 - 3) Sustav osvježava promjene u rasporedu čišćenja snijega.
 - **Mogući drugi scenarij:**
 - 1) Zamjena odbijena, te sustav javlja korisniku koji je zatražio izmjenu porukom o odbijanju zamjene.

- **UC8 – PregledajOpćeInformacije**
 - **Glavni sudionik:** Stanar, predstavnik, upravitelj.
 - **Cilj:** Dobiti uvid u opće informacije korisnika i zgrade.
 - **Sudionici:** Baza podataka.
 - **Preduvjeti:** Korisnik mora bit prijavljen.
 - **Rezultat:** Otvara se stranica sa općim informacijama korisnika i zgrade.
 - **Željeni scenarij:**
 - 1) Korisnik odabire opciju za pregled općih informacija.
 - 2) Sustav prikazuje informacije o korisniku i zgradi.
 - **Mogući drugi scenarij:** -

- **UC9– IzmjeniOpćeInformacije**
 - **Glavni sudionik:** Stanar, predstavnik, upravitelj.
 - **Cilj:** Izmijeniti opće informacije korisnika ili zgrade.
 - **Sudionici:** Baza podataka.
 - **Preduvjeti:** Korisnik mora bit prijavljen, te se nalaziti na toj stranici.
 - **Rezultat:** Korisnik izmjenjuje informacije.
 - **Željeni scenarij:**
 - 1) Korisnik odabire opciju izmjene vlastitih podataka ili podataka o zgradi.
 - 2) Sustav prikazuje formu za izmjenu podataka.
 - 3) Korisnik unosi nove podatke.
 - 4) Sustav provodi provjeru podataka i ažurira unesene podatke.
 - **Mogući drugi scenarij:**
 - 1) Uneseni podatci su neispravni, te sustav odbija akciju uz poruku.
 - 2) Korisnik nema privilegiju za izmjenu odabranih podataka.
- **UC10 – PogledajNaloge**
 - **Glavni sudionik:** Stanar, predstavnik, upravitelj.
 - **Cilj:** Dobiti uvide u naloge.
 - **Sudionici:** Baza podataka
 - **Preduvjeti:** Korisnik mora biti prijavljen.
 - **Rezultat:** Otvara mu se stranica sa nalogima na njegov zahtjev.
 - **Željeni scenarij:**
 - 1) Korisnik odabire opciju za pregled naloga.
 - 2) Sustav prikazuje korisniku sve naloge.
 - **Mogući drugi scenarij:** -
- **UC11 – AžurirajNaloge**
 - **Glavni sudionik:** Upravitelj.
 - **Cilj:** Ažurirati postojeće naloge.
 - **Sudionici:** Baza podataka.
 - **Preduvjeti:** Korisnik mora bit ulogiran i nalaziti se na toj stranici.
 - **Rezultat:** Korisnik ažurira odabrane naloge.
 - **Željeni scenarij:**
 - 1) Korisnik označava plaćene pričuve i naloge.
 - 2) Sustav osvježava i sprema promjene.
 - **Mogući drugi scenarij:** -

- **UC12 – DodajNaloge**
 - **Glavni sudionik:** Upravitelj, predstavnik.
 - **Cilj:** Dodati nove naloge.
 - **Sudionici:** Baza podataka.
 - **Preduvjeti:** Korisnik mora bit ulogiran i nalaziti se na toj stranici.
 - **Rezultat:** Korisnik dodaje nove naloge.
 - **Željeni scenarij:**
 - 1) Korisnik odabire opciju za dodavanje novog naloga.
 - 2) Sustav prikazuje formu za unos naloga.
 - 3) Korisnik unosi podatke novog naloga i potvrđuje ih.
 - 4) Sustav provjerava i ažurira unesene podatke.
 - **Mogući drugi scenarij:**
 - 1) Uneseni podatci su neispravni, te sustav odbija akciju uz poruku.

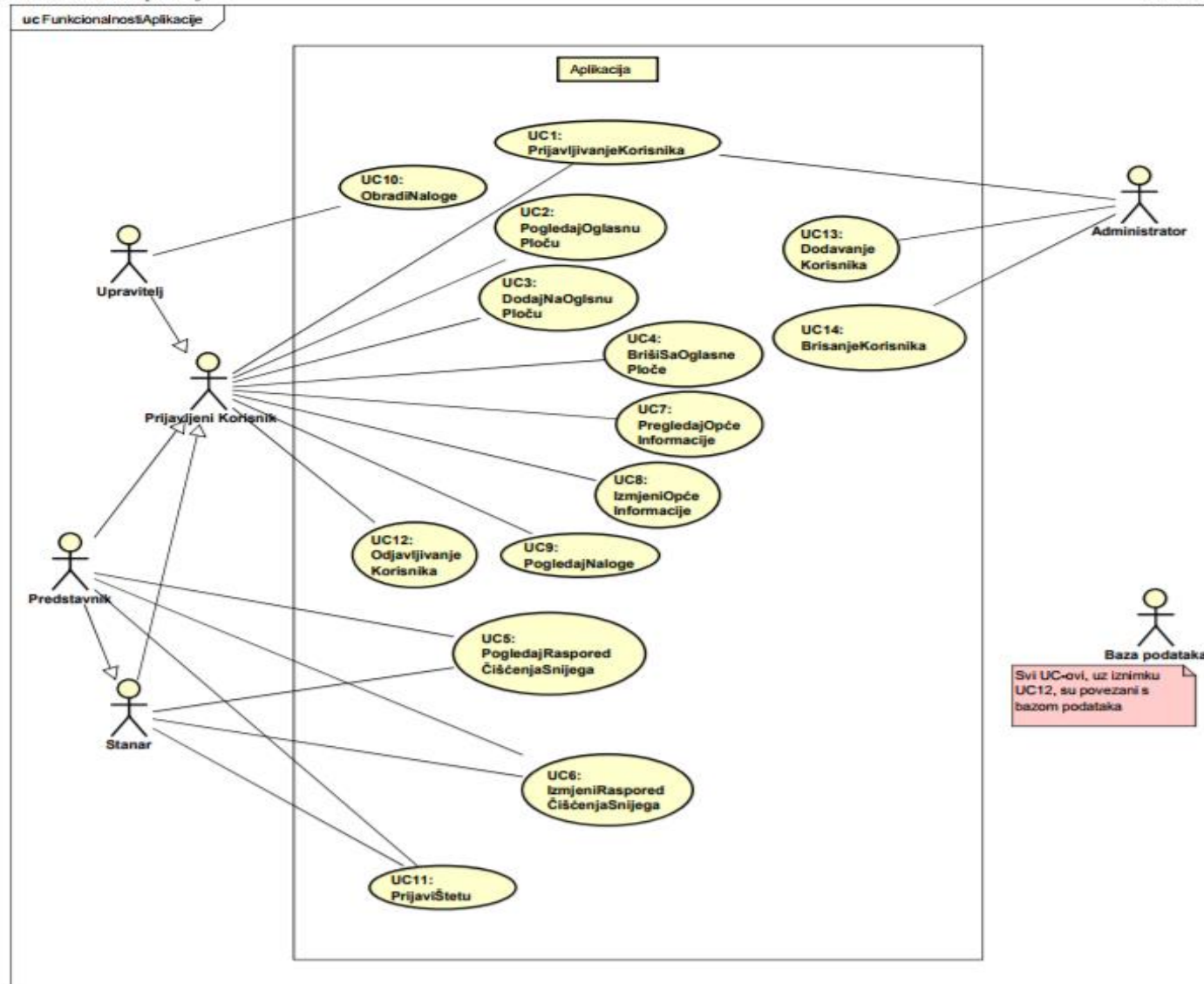
- **UC13 – PrijaviŠtetu**
 - **Glavni sudionik:** Stanar, predstavnik.
 - **Cilj:** Prijaviti nastalu štetu i kvar.
 - **Sudionici:** Baza podataka.
 - **Preduvjeti:** Korisnik je ulogiran.
 - **Rezultat:** Šteta je prijavljena.
 - **Željeni scenarij:**
 - 1) Korisnik odabire opciju za unos štete.
 - 2) Sustav prikazuje formu za unos štete.
 - 3) Korisnik unosi sve potrebne podatke za opis štete.
 - 4) Sustav provjerava ispravnost podataka, te ažurira unesene podatke.
 - **Mogući drugi scenarij:**
 - 1) Uneseni podatci su neispravni, te sustav odbija akciju uz poruku.

- **UC14 – OdjavljivanjeKorisnika**
 - **Glavni sudionik:** Stanar, predstavnik, upravitelj.
 - **Cilj:** Odjaviti korisnika iz aplikacije.
 - **Sudionici:** -
 - **Preduvjeti:** Korisnik je prijavljen.
 - **Rezultat:** Korisnik se odjavljuje, te se otvara stranica za prijavu korisnika.
 - **Željeni scenarij:**
 - 1) Korisnik odabire opciju za odjavu.
 - 2) Sustav odjavljuje korisnika, te ga vraća na stranicu za prijavu.
 - **Mogući drugi scenarij:** -

- **UC15 – DodavanjeKorisnika**
 - **Glavni sudionik:** Administrator.
 - **Cilj:** Dodati novog korisnika.
 - **Sudionici:** Baza podataka.
 - **Preduvjeti:** Korisnik nije dodan, administrator je prijavljen.
 - **Rezultat:** U bazu podataka se dodaje novi korisnik.
 - **Željeni scenarij:**
 - 1) Administrator odabire opciju za dodavanje korisnika.
 - 2) Sustav prikazuje formu za unos novog korisnika.
 - 3) Administrator unosi podatke i potvrđuje ih.
 - 4) Sustav provjerava podatke, te dodaje korisnika .
 - **Mogući drugi scenarij:**
 - 1) Uneseni su krivi podatci, te sustav javlja pogrešku.
 - 2) Postoji korisnik s istim privilegijama(predstavnik, upravitelj), te sustav javlja grešku.
- **UC16 – BrisanjeKorisnika**
 - **Glavni sudionik:** Administrator
 - **Cilj:** Obrisati postojećeg korisnika.
 - **Sudionici:** Baza podataka.
 - **Preduvjeti:** Korisnik već postoji, administrator je prijavljen.
 - **Rezultat:** Korisnik ne postoji više u bazi.
 - **Željeni scenarij:**
 - 1) Administrator odabire korisnika koji se briše.
 - 2) Sustav osvježava bazu podataka.
 - **Mogući drugi scenarij:** -
- **UC17 – UrediObjavuNaOglasnojPloči**
 - **Glavni sudionik:** Stanar, predstavnik, upravitelj, administrator.
 - **Cilj:** Urediti objave na oglasnoj ploči.
 - **Sudionici:** Baze podataka.
 - **Preduvjeti:** Korisnik je ulogiran.
 - **Rezultat:** Korisnik uređuje podatke na oglasnoj ploči.
 - **Željeni scenarij:**
 - 1) Korisnik odabire opciju uređivanja objave.
 - 2) Sustav prikazuje formu za uređivanje objave.
 - 3) Sustav provjerava unesene promjene, te ih ažurira.
 - **Mogući drugi scenarij:**
 - 1) Uneseni su krivi podatci tokom uređivanja, te sustav odbija uređivanje porukom.

Funkcionalnosti Aplikacije

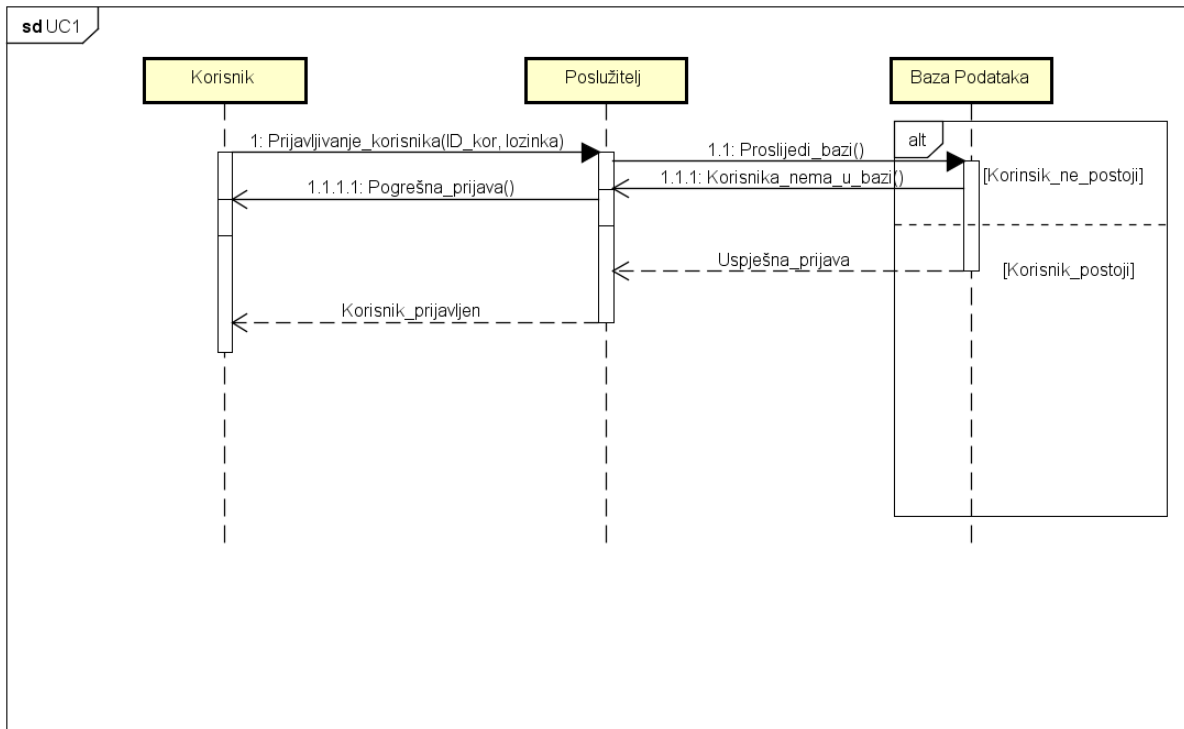
2017/11



Slika 1 Funkcionalnosti aplikacije

Sekvencijski dijagrami :**Obrazac uporabe UC1 (PrijavljivanjeKorisnika)**

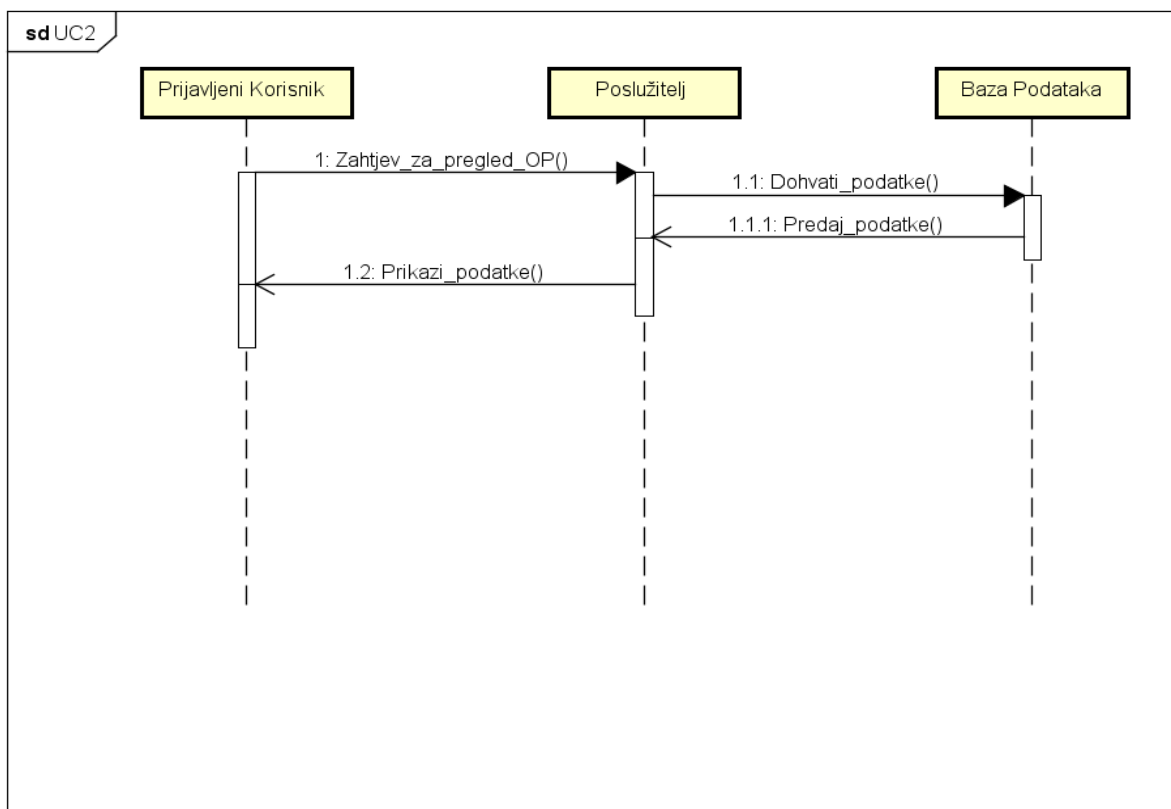
Korisnik se prijavljuje na sustav sa svojim ID-om i lozinkom. Poslužitelj prosljeđuje ID i lozinku bazi koja provjerava da li korisnik postoji u bazi. Ukoliko korisnik ne postoji, baza obavještava poslužitelja koji javlja korisniku da je pogrešna prijava. U suprotnom korisnik se uspješno prijavljuje u sustav.



Slika 2 Sekvencijski dijagram obrasca PrijavljivanjeKorisnika

Obrazac uporabe UC2 (PogledajOglasnuPloču)

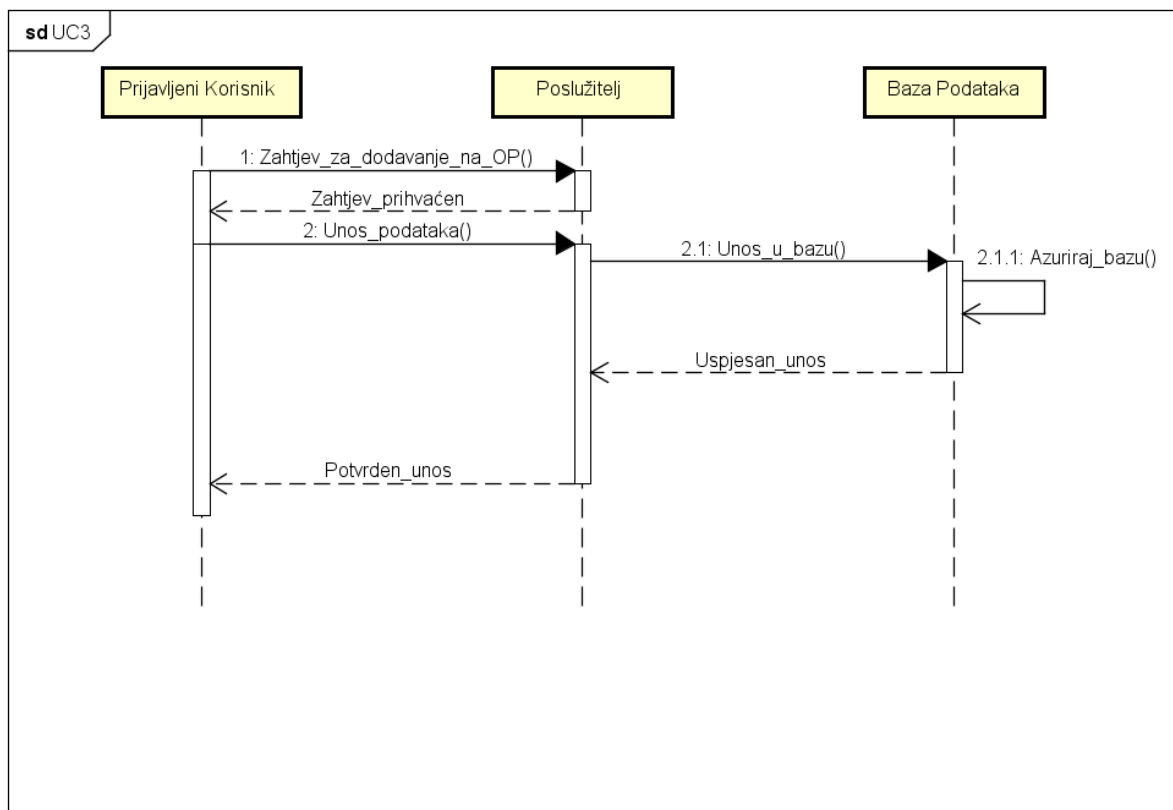
Prijavljeni korisnik zahtjeva pregled oglasne ploče. Poslužitelj dohvaća sve podatke iz baze podataka i daje ih na uvid korisniku.



Slika 3 Sekvencijski dijagram obrasca PogledajOglasnuPloču

Obrazac uporabe UC3 (DodajNaOglasnuPloču)

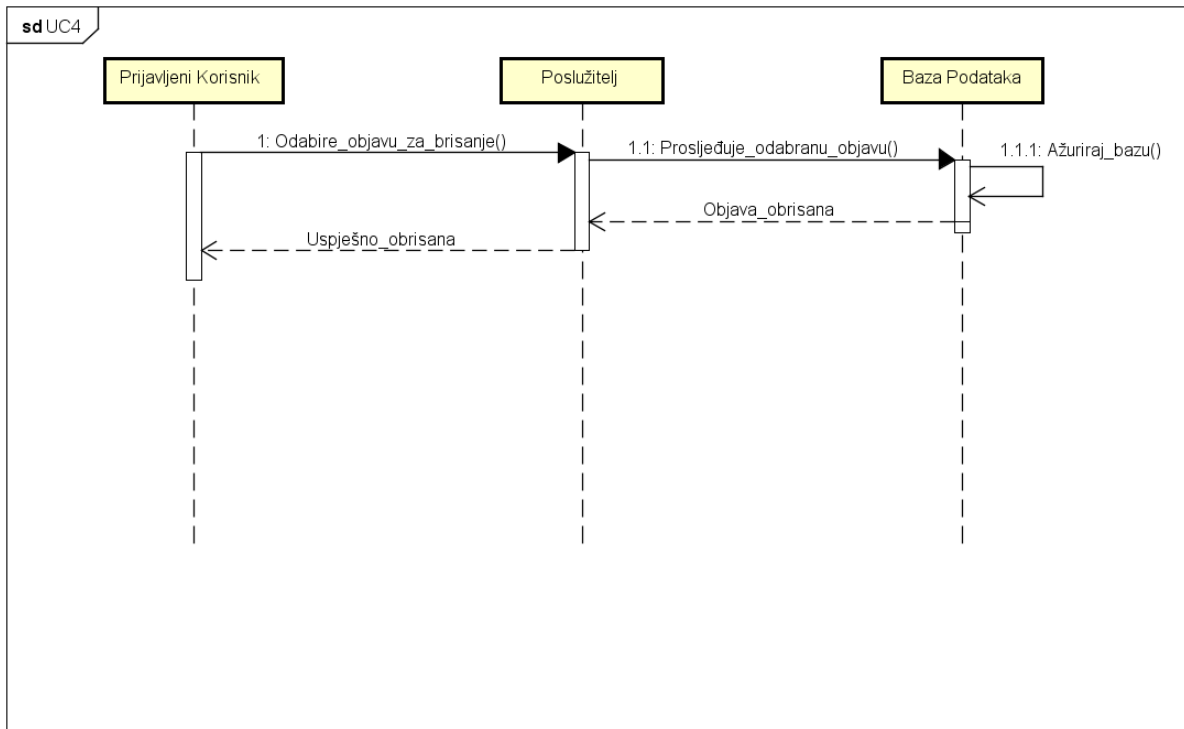
Korisnik zahtjeva dodavanje obavijesti na oglasnu ploču, što mu poslužitelj dozvoljava. Zatim korisnik unosi sve potrebne podatke koje želi dodati, poslužitelj ih prosljeđuje bazi podataka koja ih unosi i sprema. I na kraju korisnik dobiva obavijest da je njegov unos uspješno napravljen.



Slika 4 Sekvencijski dijagram obrasca DodajNaOglasnuPloču

Obrazac uporabe UC4 (BrišiSaOglasnePloče)

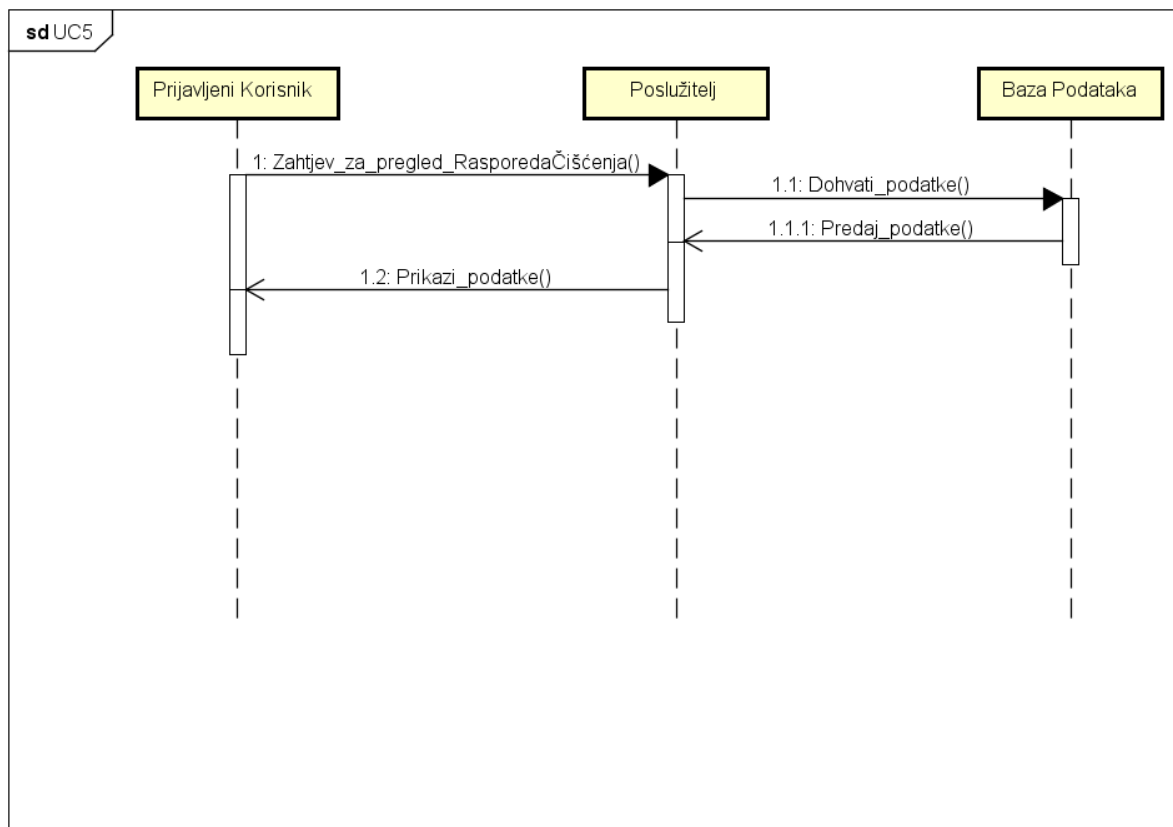
Korisnik odabire objavu koju želi obrisati. Poslužitelj proslijeđuje taj zahtjev bazi podataka koja obavlja brisanje i javlja poslužitelju da je objava obrisana. Na kraju poslužitelj vraća korisniku poruku o uspješno obavljenom brisanju objave.



Slika 5 Sekvencijski dijagram obrasca BrišiSaOglasnePloče

Obrazac uporabe UC5 (PogledajRasporedČišćenjaSnijega)

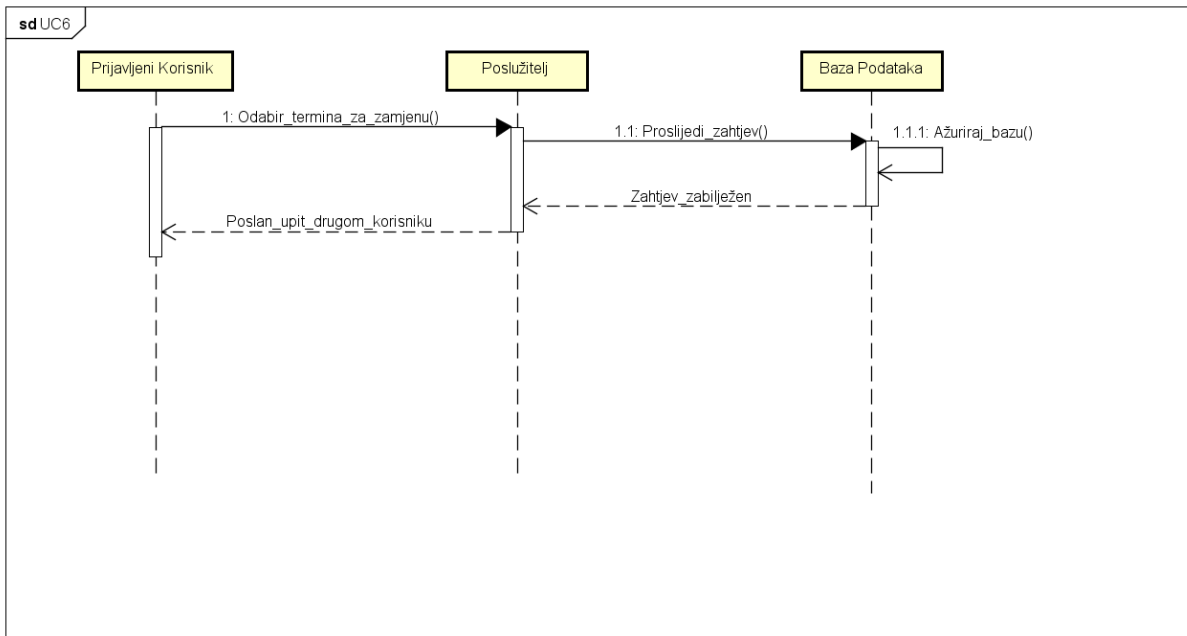
Prijavljeni korisnik zahtjeva pregled rasporeda čišćenja snijega. Poslužitelj dohvaća sve podatke iz baze podataka te ih daje na uvid korisniku.



Slika 6 Sekvencijski dijagram obrasca PregledajRasporedČišćenjaSnijega

Obrazac uporabe UC6 (ZatražioZamjenuTerminaRasporedaČišćenjaSnijega)

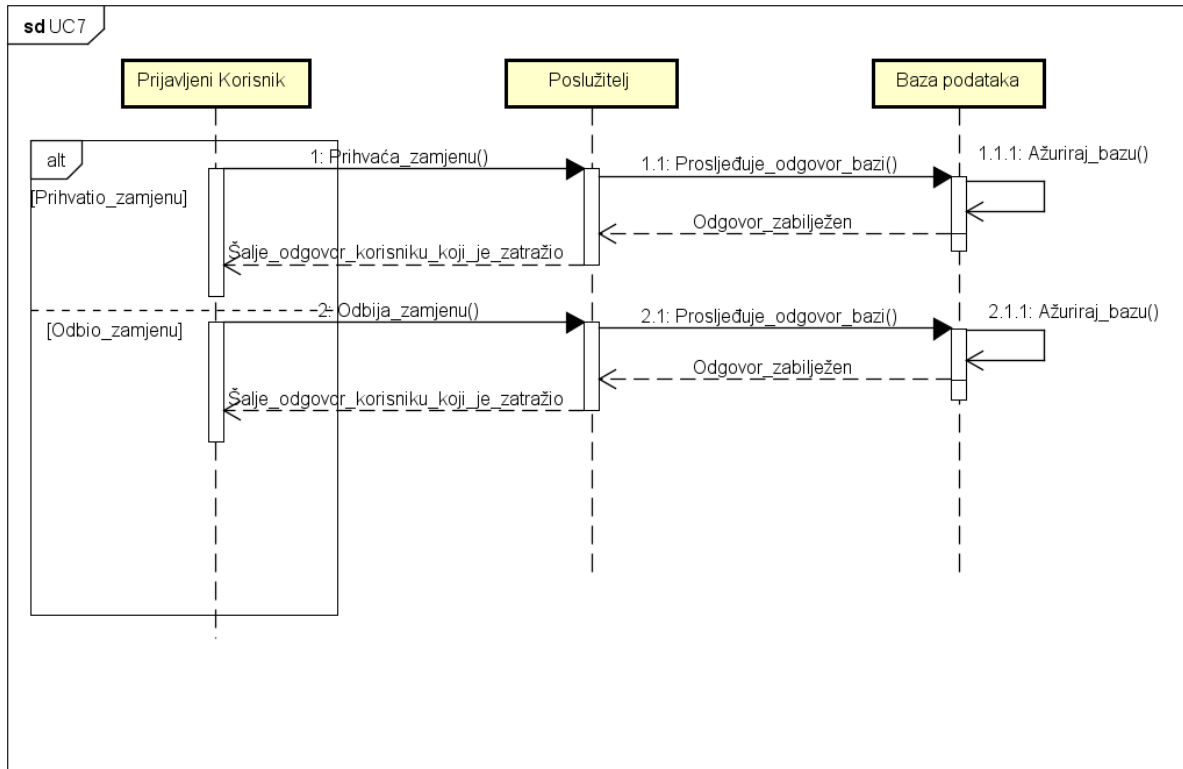
Korisnik odabire termin koji mu odgovara za zamjenu, te poslužitelj to prosljeđuje bazi koja to bilježi. Nakon toga poslužitelj obavještava porukom drugom korisnika čiji je datum za zamjenu zatražen.



Slika 7 Sekvencijski dijagram obrasca ZatražioZamjenuTerminaRasporedaČišćenjaSnijega

Obrazac uporabe UC7 (OdgovorNaZamjenuTerminaRasporedaČišćenjaSnijega)

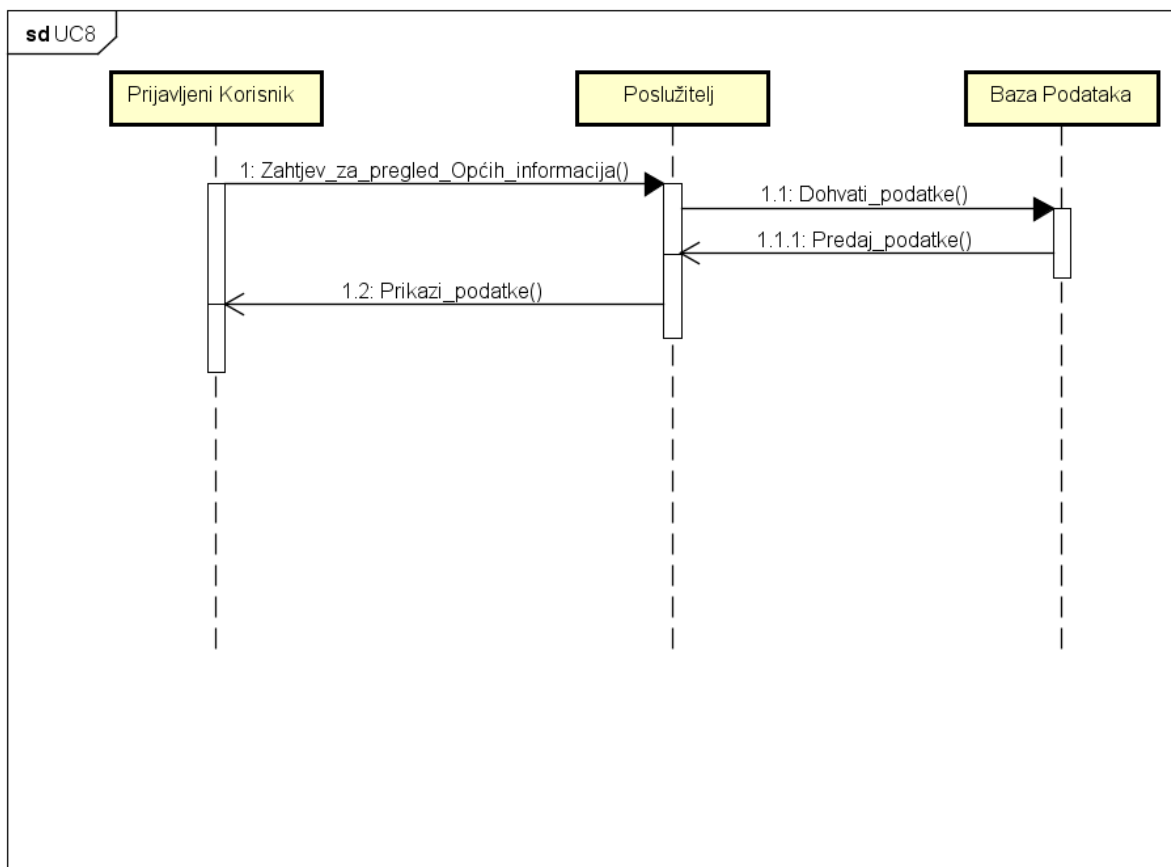
U ovom slučaju prijavljeni korisnik ima dvije opcije. Ako odabere prihvatanje zamjene, poslužitelj to proslijeđuje bazi koja to ažurira u svojim podacima, te nakon toga poslužitelj šalje obavijest korisniku da je poslana potvrdna poruka korisniku koji je zatražio zamjenu. U drugom slučaju, ako je korisnik odbio zamjenu, događa se isti scenarij osim što je poruka poslana korisniku koji je zatražio zamjenu u ovom slučaju negativna.



Slika 8 Sekvencijski dijagram obrasca *OdgovorNaZamjenuTerminaRasporedaČišćenjaSnijega*

Obrazac uporabe UC8 (PregledajOpćeInformacije)

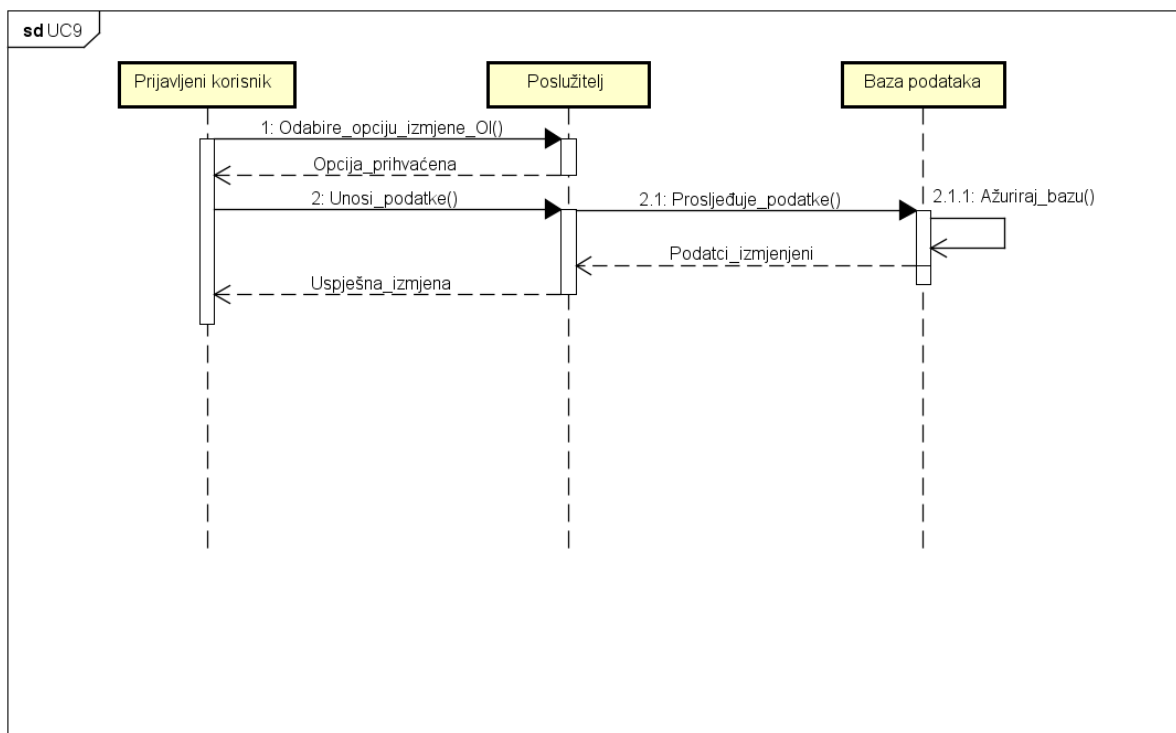
Prijavljeni korisnik zahtjeva pregled općih informacija. Poslužitelj dohvaća sve podatke iz baze podataka te ih daje na uvid korisniku.



Slika 9 Sekvencijski dijagram obrasca PregledajOpćeInformacije

Obrazac uporabe UC9 (IzmijeniOpćeInformacije)

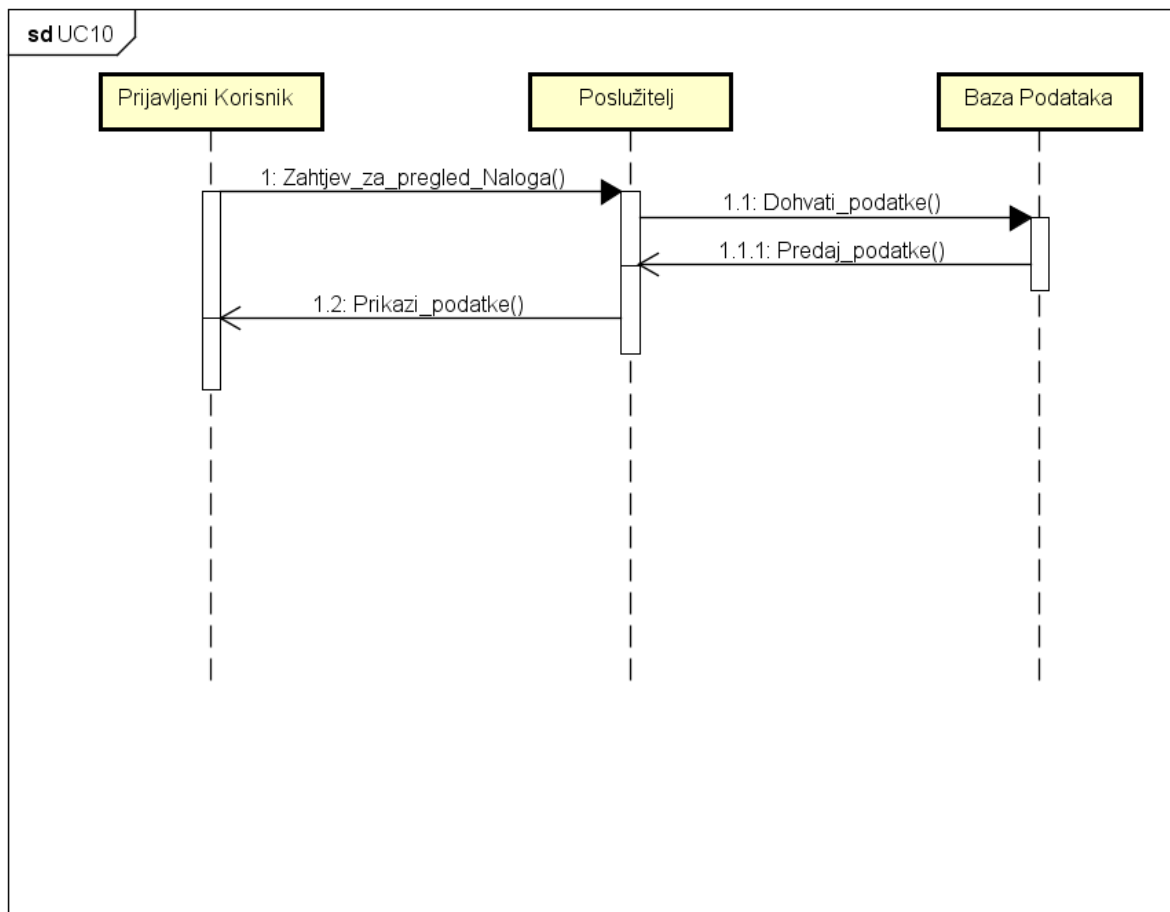
Prijavljeni korisnik odabire opciju izmjenu podataka u općim informacijama. Poslužitelj mu odobrava radnju, te korisnik unosi nove podatke. Poslužitelj te podatke proslijeđuje bazi koja ažurira iste, te potvrđuje izmjenu poslužitelju koji obavještava korisnika da je uspješna izmjena.



Slika 10 Sekvencijski dijagram obrasca IzmijeniOpćeInformacije

Obrazac uporabe UC10 (PogledajNaloge)

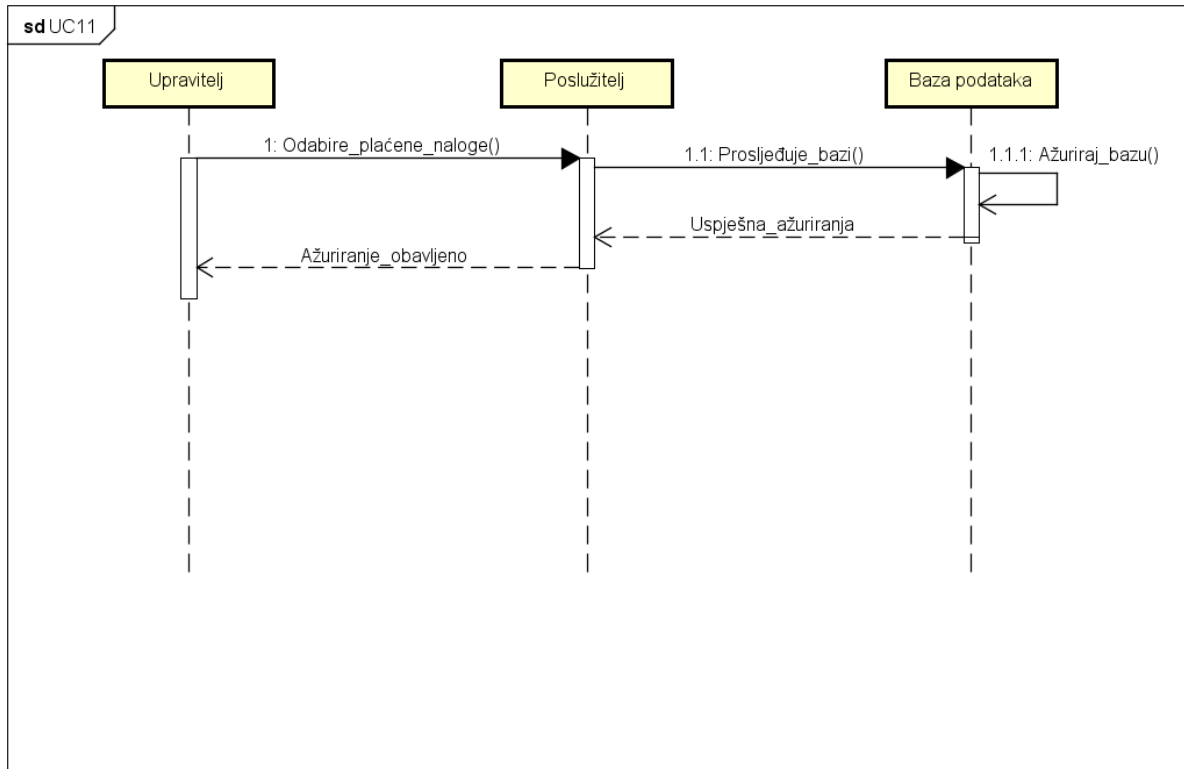
Prijavljeni korisnik zahtjeva pregled naloga. Poslužitelj dohvaća sve podatke iz baze podataka te ih daje na uvid korisniku.



Slika 11 Sekvencijski dijagram obrasca PogledajNaloge

Obrazac uporabe UC11 (AžurirajNaloge)

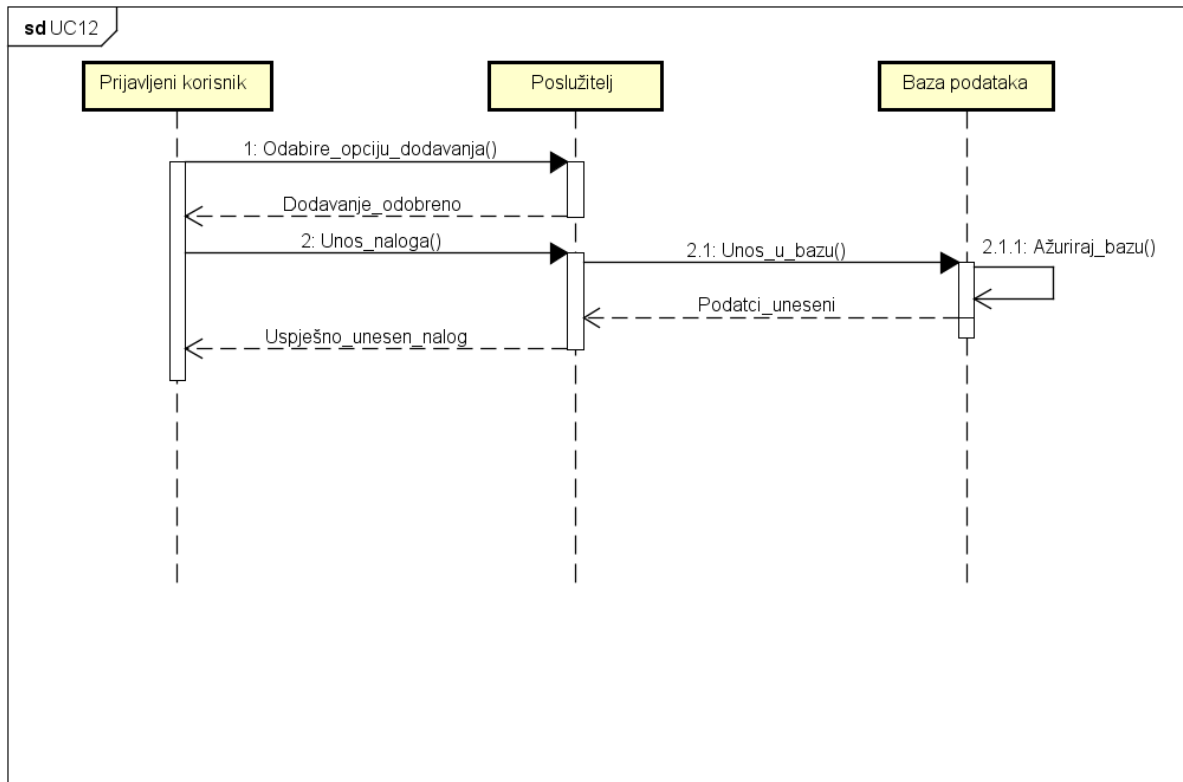
Upravitelj odabire naloge i pričuve koji su plaćeni, te ih poslužitelj prosljeđuje bazi koja ažurira označene naloge u svojoj bazi, te šalje povratnu poruku o uspješnosti, koju onda poslužitelj prosljeđuje upravitelju.



Slika 12 Sekvencijski dijagram obrasca AžurirajNaloge

Obrazac uporabe UC12 (DodajNaloge)

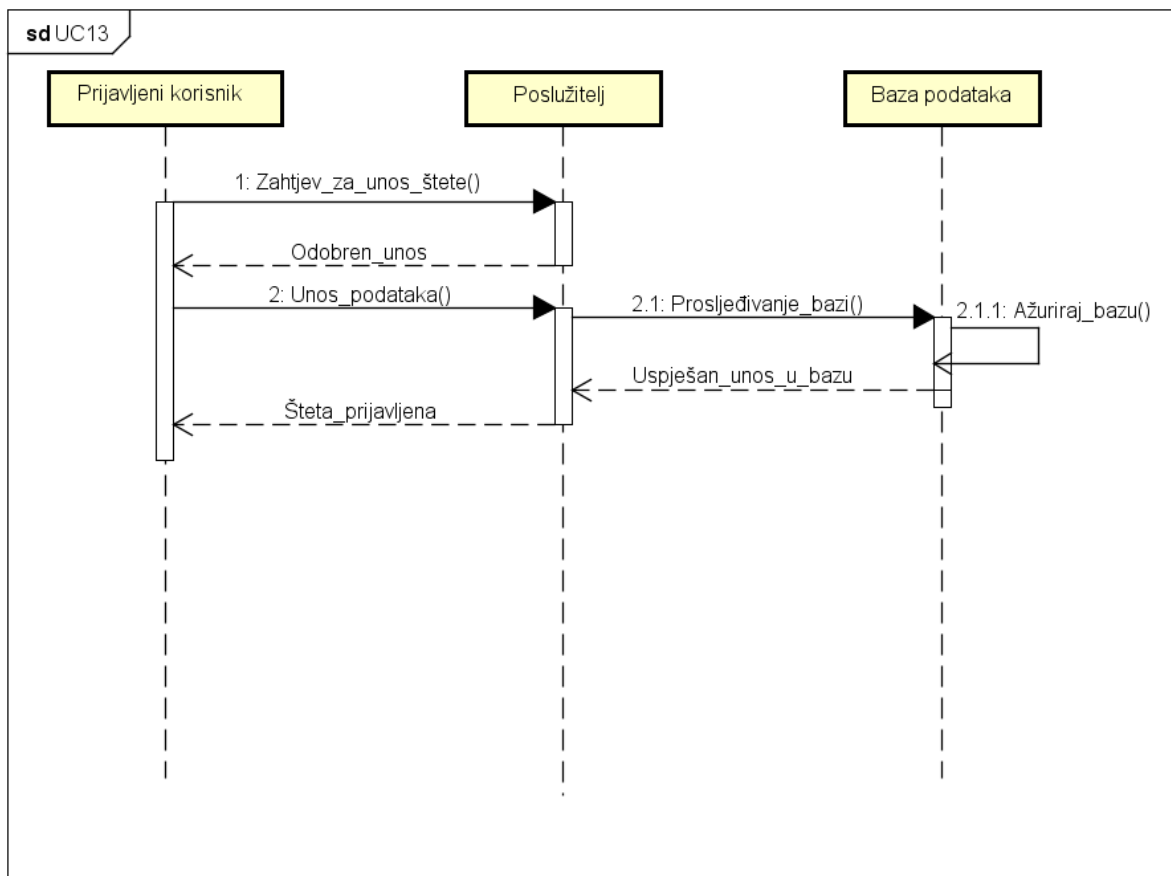
Korisnik, u ovom slučaju, upravitelj ili predstavnik, odabiru opciju dodavanja novih naloga, koje im poslužitelj odobrava, nakon čega oni unose sve potrebne podatke. Te podatke poslužitelj prosljeđuje bazi koja ih pohranjuje i vraća poruku o uspješnosti pohrane krajnjem korisniku.



Slika 13 Sekvencijski dijagram obrasca DodajNaloge

Obrazac uporabe UC13 (PrijaviŠtetu)

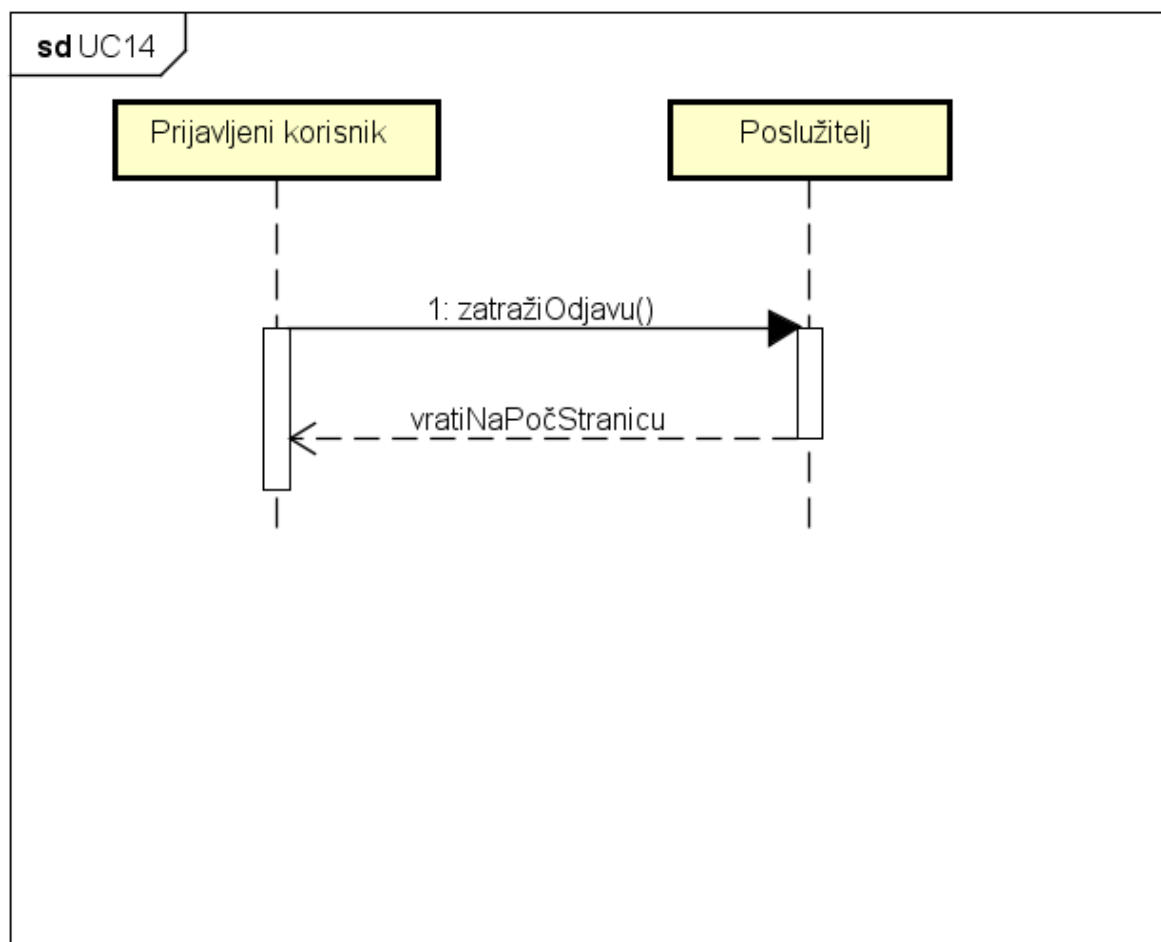
Prijavljeni korisnik zatražuje unos štete, što mu poslužitelj odobrava. Nakon toga korisnik unosi podatke koje poslužitelj prosljeđuje bazi koja ih pohranjuje i dojavljuje porukom uspješan unos, koji poslužitelj nakon toga prosljeđuje korisniku.



Slika 14 Sekvencijski dijagram obrasca PrijaviŠtetu

Obrazac uporabe UC14 (OdjavljivanjeKorisnika)

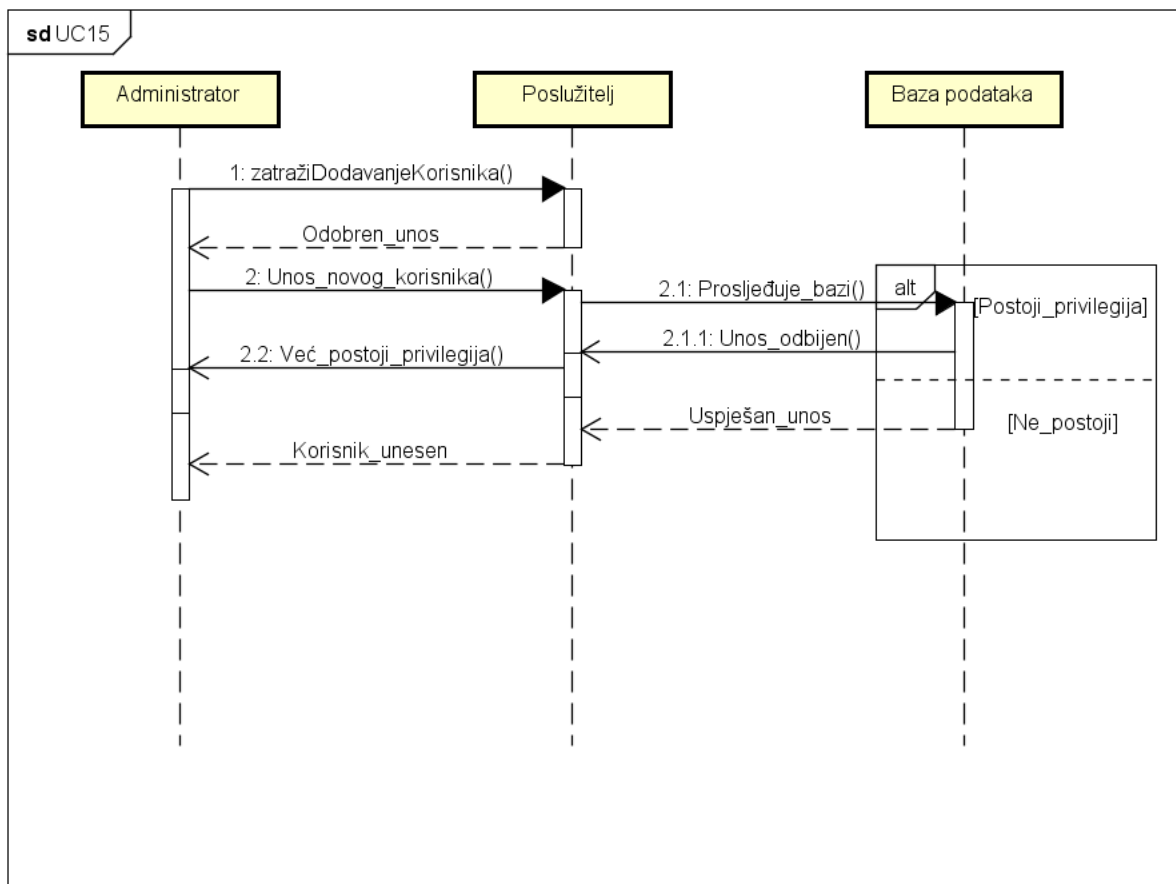
Prijavljeni korisnik zatražuje odjavu od poslužitelja, koji ga na zahtjev odjavljuje i vraća na početnu stranicu.



Slika 15 Sekvencijski dijagram obrasca OdjavljivanjeKorisnika

Obrazac uporabe UC15 (DodavanjeKorisnika)

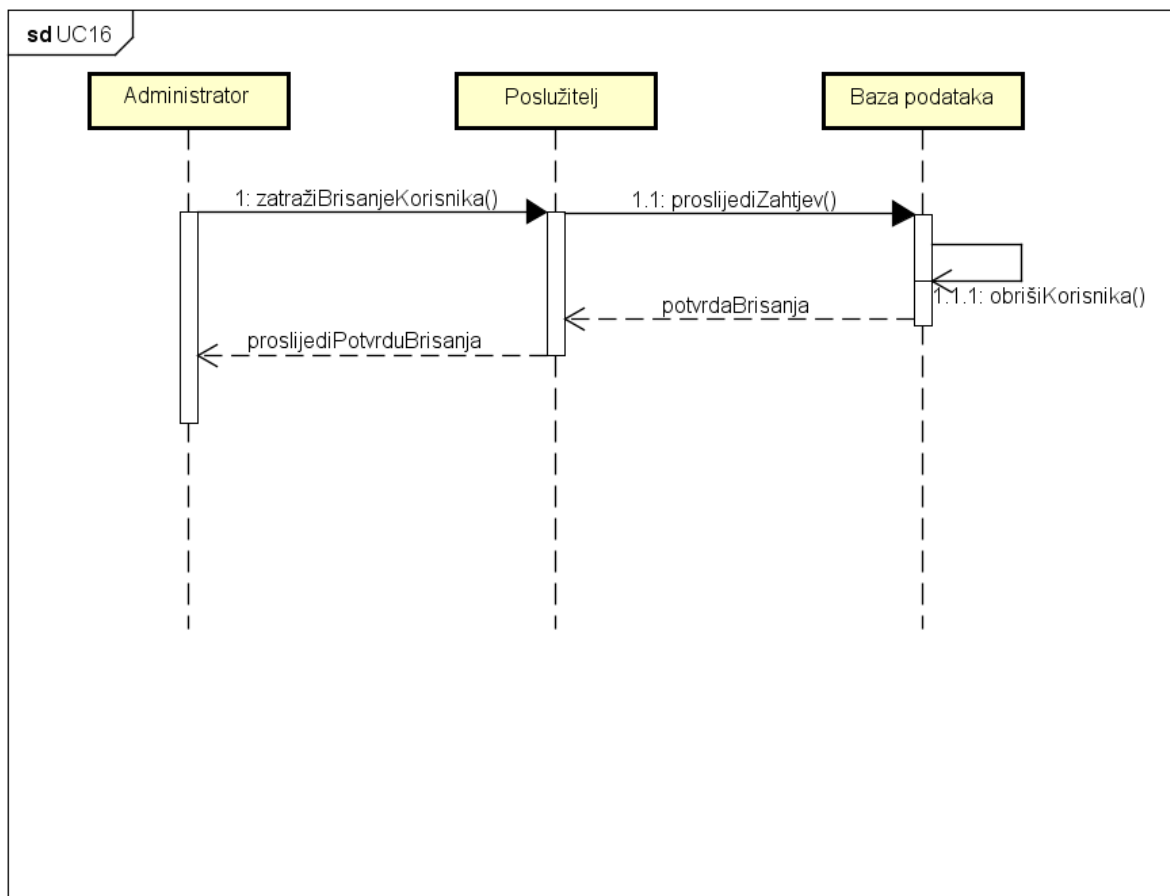
Administrator zatraži dodavanje korisnika koje mu poslužitelj odobrava. Zatim administrator unosi podatke koji se proslijeđuju bazi, te ako je unijet korisnik sa privilegijama koje već postoje onda mu se unos odbija sa odgovarajućom porukom. U suprotnom, baza pohranjuje unos korisnika, te unos potvrđuje porukom koja u konačnici dolazi do administratora.



Slika 16 Sekvencijski dijagram obrasca DodavanjeKorisnika

Obrazac uporabe UC16 (BrisanjeKorisnika)

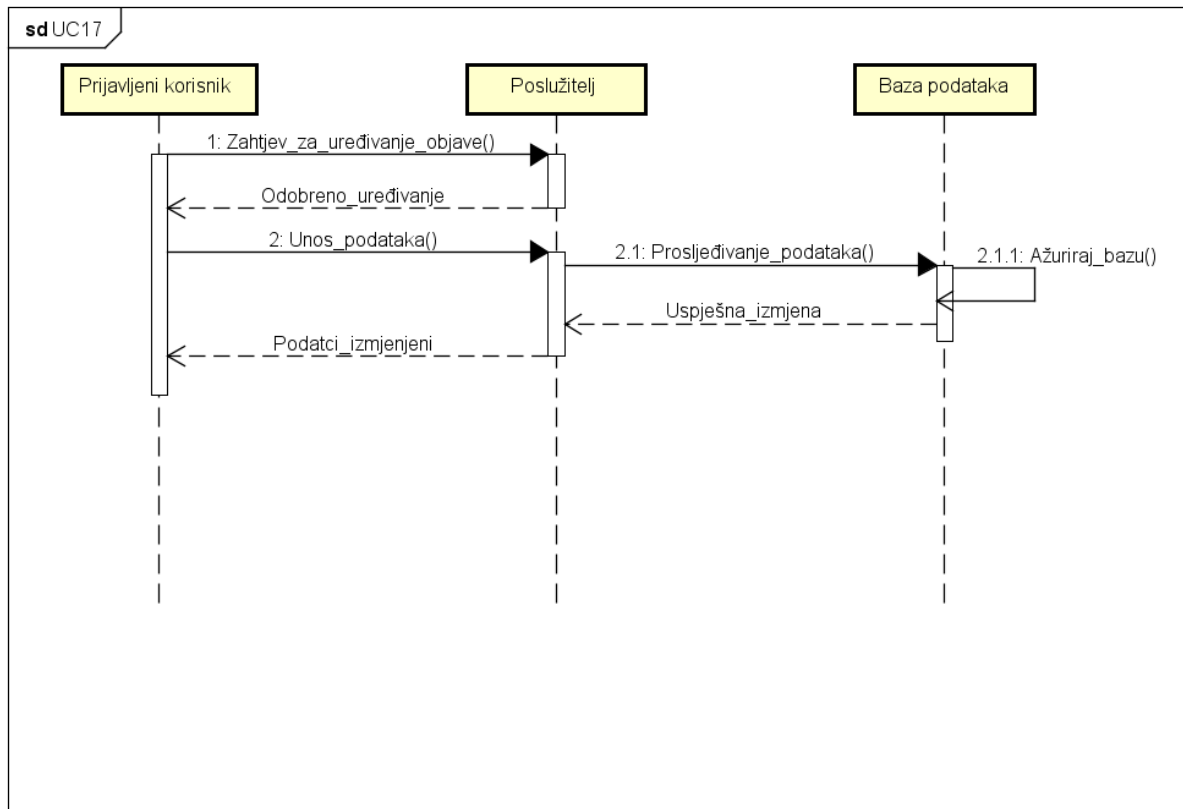
Administrator zatražuje brisanje korisnika. Poslužitelj zahtjev proslijeđuje bazi podataka, koja briše korisnika iz baze i to javlja porukom o potvrdi brisanja koju poslužitelj u konačnici proslijeđuje administratoru.



Slika 17 Sekvencijski dijagram obrasca *BrisanjeKorisnika*

Obrazac uporabe UC17 (UrediObjavuNaOglasnojPloči)

Prijavljeni korisnik zahtjeva uređivanje objave na oglasnoj ploči, što mu poslužitelj odobrava. Nakon toga korisnik unosi nove podatke, odnosno izmjenjuje već unesene podatke, koje onda poslužitelj prosljeđuje bazi, koja ih pohranjuje te dojavljuje uspješnu izmjenu poslužitelju koji u konačnici javlja to korisniku.



Slika 18 Sekvencijski dijagram obrasca UrediObjavuNaOglasnojPloči

5. Ostali zahtjevi

- Sustav mora podržavati paralelan rad više korisnika
- Sustav i korisničko sučelje moraju podržavati sve znakove hrvatske abecede, kao i interpunkcijske znakove
- Sustav mora biti napravljen tako da omogućava jednostavno korištenje korisniku bez detaljnih uputa
- Pretraga baze podataka ne smije trajati duže od 5 sekundi

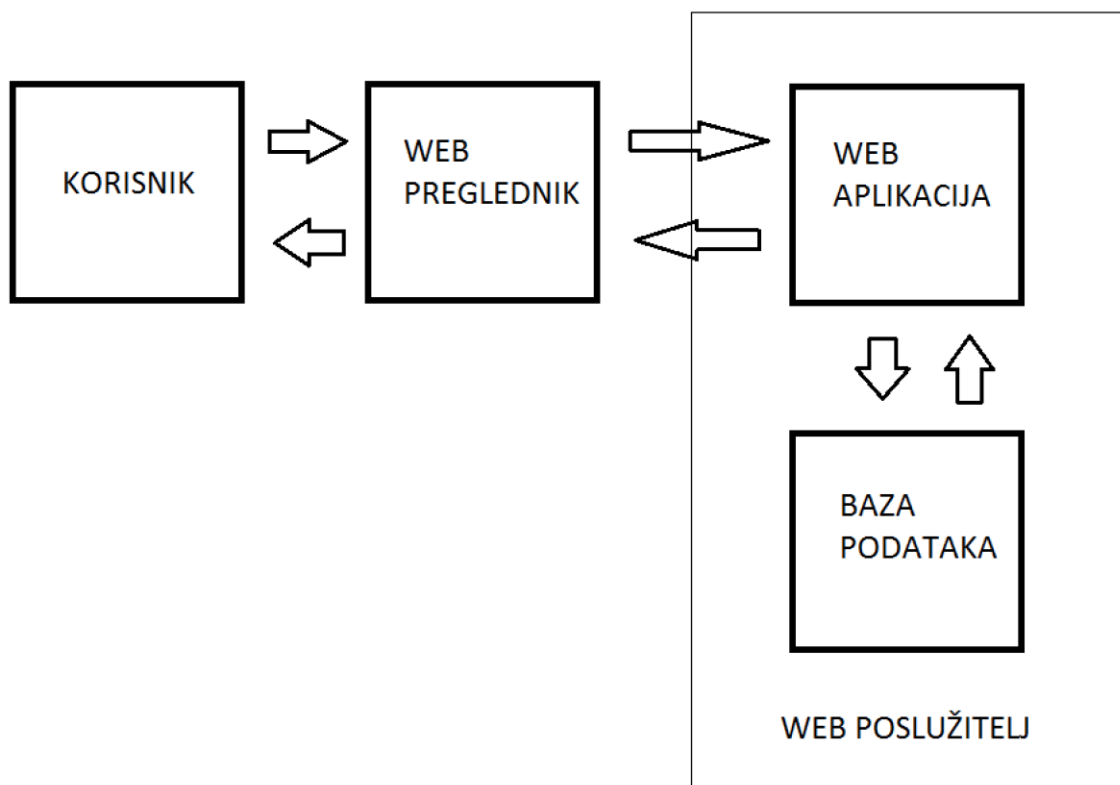
6. Arhitektura i dizajn sustava

6.1. Svrha, opći prioriteti i skica sustava

Arhitektura našeg sustava je web aplikacija iz razloga da korisnici ovog sustava mogu lakše komunicirati i lakše koristiti sam sustav.

Podsustavi arhitekture:

- Web poslužitelj
- Web aplikacija
- Baza podataka



Slika 19 Skica sustava

Web poslužitelj

Web poslužitelj obrađuje sve zahtjeve koji dolaze sa web aplikacije HTTP protokolom, koji nakon obrade šalje odgovarajući odgovor web aplikaciji, također HTTP protokolom.

Naša web aplikacija će se izvoditi *Apache Tomcat* web poslužitelju koji je opisan u pojmovniku ove dokumentacije.

Web aplikacija

Putem web aplikacije korisnik ima pristup svim funkcionalnostima sustava ovisno o pridjeljenim ovlastima. Preko web aplikacije šaljemo HTTP zahtjev na poslužitelj kojim dohvaćamo i šalje podatke bazi podataka.

Zadaća web aplikacije je isto tako da prikazuje vizualne informacije u obliku HTML dokumenta, dok će preglednik te dokumente prikazivati u obliku web stranica.

Baza podataka

Baza podataka sadrži pohranu svih podataka o korisnicima i svim potrebnim podacima za određene funkcionalnosti. Zadaća same baze je da brzo i po potrebi dohvaća podatke koje je korisnik zatražio, osvježava promjene koje su unešene od strane korisnika i sprječava mogući unos podataka koji se već nalaze u samoj bazi.

Entiteti baze podataka:

Zgrada – Ovaj entitet sadrži sve bitne informacije o zgradama. Atributi u ovom entitetu su: ID, Adresa, Fond, IDPredstavnika, IDUpravitelja. Entitet ima One-to-Many vezu na entitet **Stan** preko atributa ID prema atributu IDZgrade i dvije Many-to-One veze preko atributa IDPredstavnika i IDUpravitelj prema atributu ID u entitetu **Korisnik**.

Stan – Ovaj entitet sadrži sve podatke o pojedinom stanu. Atributi ovog entiteta su: ID, Površina, IDZgrade, IDVlasnika. Entitet ima vezu Many-to-One preko atributa IDZgrade na atribut ID u entitetu **Zgrada** i Many-to-One vezu prema entitetu **Korisnik** preko atributa IDVlasnika, prema atributu ID.

Korisnik – Entitet koji sadrži sve podatke o pojedinom korisniku, kao što su atributi: ID, Ime, Prezime, Mail, Lozinka, Privilegija, Podsjetnik. Entitet ima dvije vezu One-to-Many preko atributa ID prema atributima IDPrimatelja i IDPlatitelja u entitetu **Nalog**, jednu One-to-Many vezu preko atributa ID prema atributu IDVlasnika u entitetu **Stan**, dvije One-to-Many veze preko atributa ID prema atributima IDPredstavnik i IDUpravitelja u entitetu **Zgrada**, jednu Many-to-One vezu preko atributa ID prema atributu IDOsobe u entitetu **TerminČišćenjaSnijega**, jednu One-to-Many vezu preko atributa ID prema atributu IDKorisnika u entitetu **Notifikacija**, jednu One-to-Many vezu preko atributa ID prema atributu IDKreatora u entitetu **Objava**, te One-to-Many vezu preko atributa ID prema atributu IDKreatora u entitetu **Trošak**.

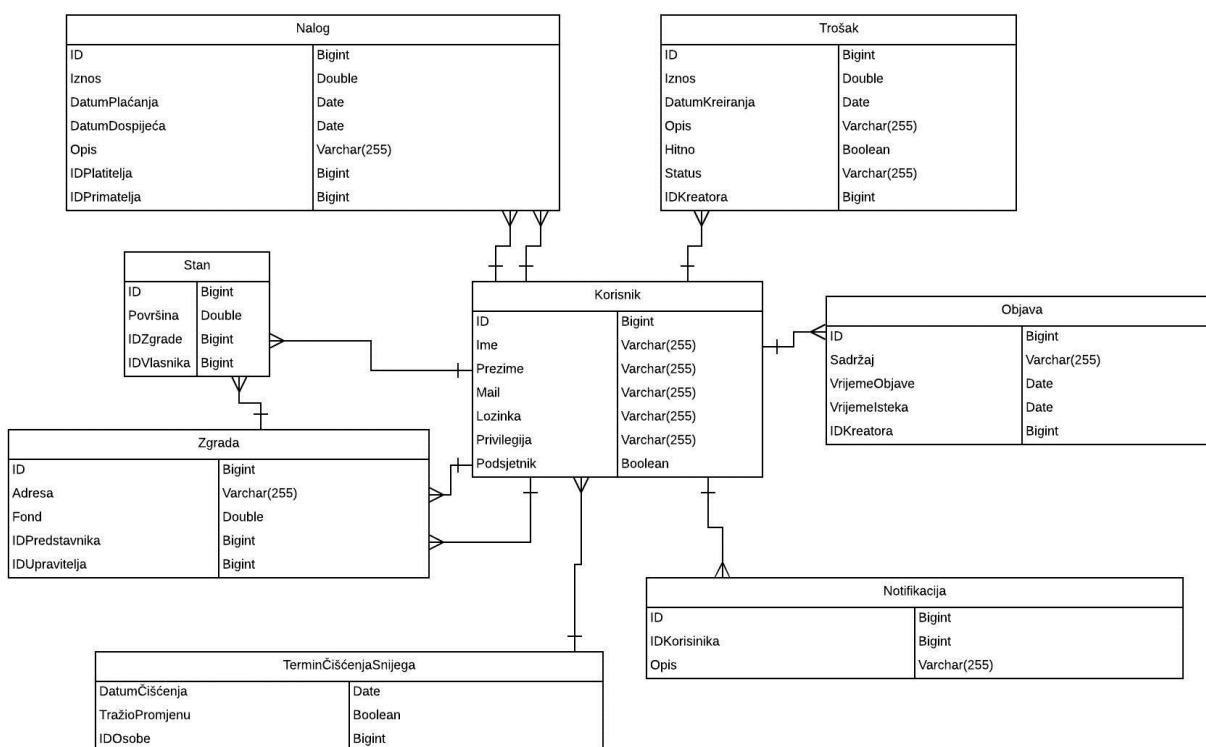
Trošak – Entitet koji sadrži sve podatke o troškovima zgrade sa atributima: ID, Iznos, DatumKreiranja, Opis, Hitno, Status, IDKreatora. Entitet sadrži vezu Many-to-One preko atributa IDKreatora prema atributu ID u entitetu **Korisnik**.

Nalog – Entitet koji sadrži podatke potrebne o nalogima zgrade i attribute kao što su: ID, Iznos, DatumPlaćanja, DatumDospijeca, Opis, IDPlatitelja i IDPrimatelja. Entitet sadrži dvije Many-to-One veze preko atributa IDPlatitelja i IDPrimatelja prema atributu ID u entitetu **Korisnik**.

TerminČišćenjaSnijega – Entitet koji pregled termina čišćenja za određenog korisnika i mogućnost da isti zatraži promjenu termina. Entitet sadrži attribute: DatumČišćenja, TražioPromjenu, IDOsobe. Sadrži jednu One-to-Many vezu preko atributa IDOsobe prema atributu ID u entitetu **Korisnik**.

Notifikacija – Entitet koji prikazuje obavijest i korisnika kome je obavijest namjenjena. Sadrži attribute: ID, IDKorisnika, Opis. Entitet sadrži jednu Many-to-One vezu preko IDKorisnika prema atributu ID u entitetu **Korisnik**.

Objava – Entitet koji sadrži objave, njihovo vrijeme objave, vrijeme isteka i sadržaj objave, koje je stvorio određeni korisnik. Sadrži attribute: ID, Sadržaj, VrijemeObjave, VrijemeIsteka i IDKreatora. Entitet sadrži jednu Many-to-One vezu preko atributa IDKreatora prema atributu ID u entitetu **Korisnik**.



Slika 20 ER model baze podataka

6.2. Dijagram razreda s opisom

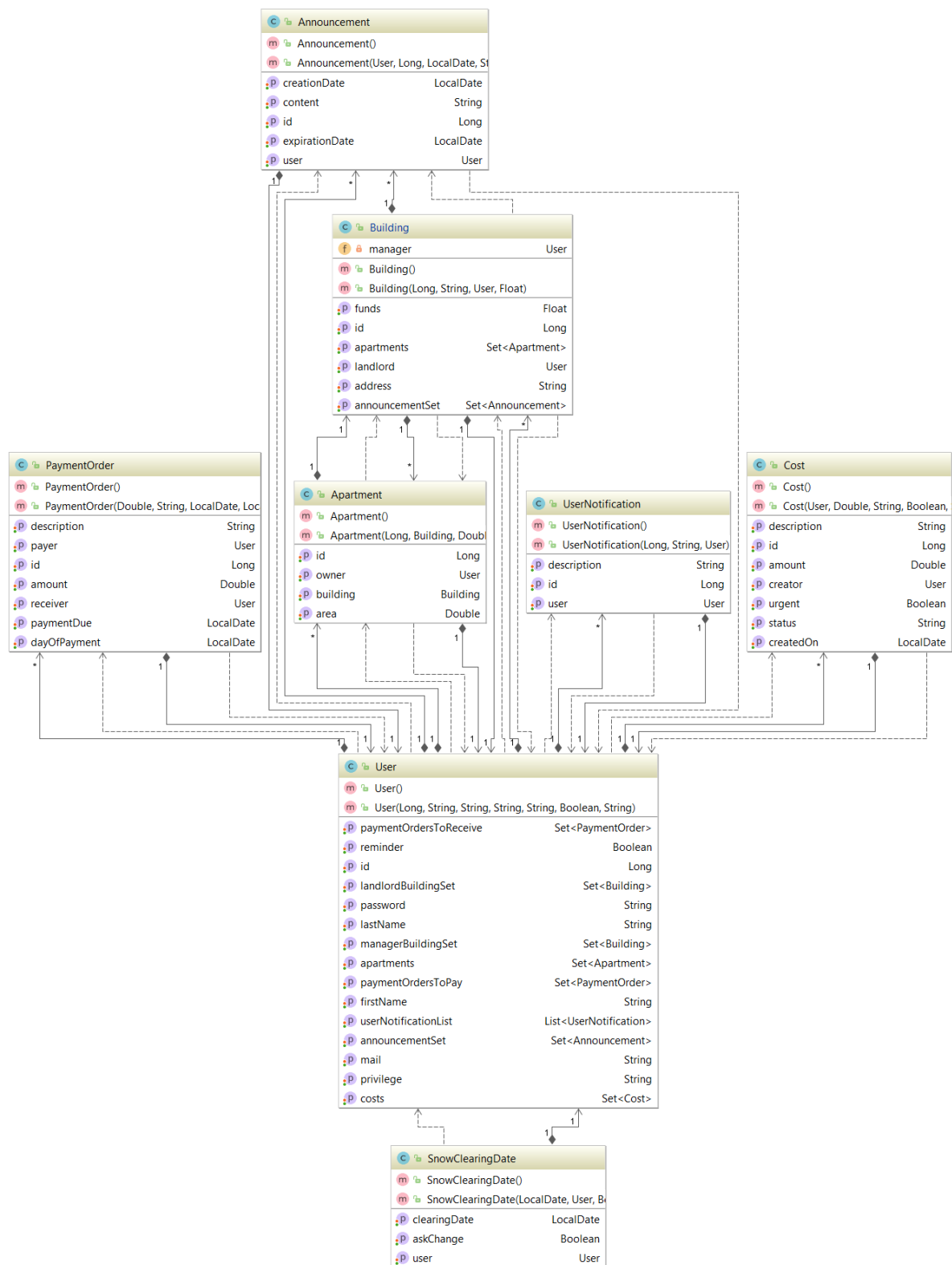
U ovom odlomku opisani su svi razredi te priložen pripadajući dijagrami. Također su navedene sve pripadajuće metode te vrste pristupa tim metodama. Iz obrazaca uporabe vidimo koje operacije trebamo implementirati za svaki razred. Razredi su podijeljeni u tri paketa: *domain*, *service* i *controller*. U paketu *domain* nalaze se svi potrebni modeli podataka koji se koriste u izradi aplikacije, u paketu *service* nalaze razredi zaduženi za komunikaciju s bazom podataka, a u paketu *controller* nalaze se svi razredi koji upravljaju korisničkim zahtjevima.

Razredi koji su u paketu *domain*:

- Obavijest korisniku
 - Razred koji oblikuje obavijesti koje se prikazuju korisniku u aplikaciji, njegovi atributi su: **identifikator**, **tekst obavijesti** i objekt **Korisnik** kojem se prikazuje obavijest
 - Postoje javne metode za postavljanje i dohvaćanje svih svojih atributa
- Korisnik
 - Razred koji predstavlja korisnika web-aplikacije, sadrži attribute: **identifikator**, **ime**, **prezime**, **adresu elektroničke pošte**, **privilegiju**, zastavicu **Podsjetnik** koja označava želi li korisnik primati obavijest ako nisu izvršili uplatu na vrijeme, **skup obavijesti**, **listu notifikacija**, **skup troškova**, **skup zgrada** u kojima ima stanove i **skup stanova** kojih je vlasnik
 - Sadrži javne metode za postavljanje i dohvaćanje svih svojih atributa
- Termin čišćenja snijega
 - Razred koji modelira termin čišćenja snijega sa atributima: **datum čišćenja snijega**, objektom **Korisnik** koji je zadužen za čišćenje i zastavicom **zatražiZamjenu** koja je aktivna ako je korisnik zatražio zamjenu
 - Metode su javne i omogućavanju dohvaćanje i postavljanje svih svojih atributa
- Nalog
 - Razred koji predstavlja naloge za plaćanje, sadrži attribute: **identifikator** koji jednoznačno određuje korisnika, **iznos** uplate, **opis** sadrži tekst kojim se pobliže objašnjava svrha plaćanja, **dospijeće** kao datum do kada je potrebno platiti nalog, **datumPlaćanja** označava datum kada je korisnik izvršio uplatu, **platitelj** predstavlja korisnika(dužnika) zaduženog za plaćanje, **primatelj** je korisnik kojem se sredstva uplaćuju
 - Metode su javne i omogućavaju dohvaćanje i postavljanje svih svojih atributa
- Trošak
 - Razred koji modelira troškove zgrade, razred sadrži attribute: **identifikator** troška, objekt **Korisnik** koji je stvorio trošak, **cijenu** troška, **opis** troška, **datumKreiranja**, zastavicu **hitnost** koja je aktivna ako je trošak hitan, **status** pobliže pojašnjava u kojoj je fazi realizacije taj trošak
 - Sadrži javne metode za postavljanje i dohvaćanje svih svojih atributa
- Zgrada
 - Razred koji oblikuje zgradu sa atributima: **identifikator** koji jednoznačno određuje zgradu, **adresa** zgrade, **predstavnikStanara** kao objekt razreda

Korisnik koji predstavlja sve stanare jedne zgrade , **sredstva** označavaju koliko je trenutno raspoloživo stanje na računu zgrade, **skup obavijesti** koje su trenutno na oglasnoj ploči, **skup stanova** koji pripadaju toj zgradi

- Sadrži javne metode za postavljanje i dohvaćanje svih svojih atributa
- Stan
 - Razred koji predstavlja stan u zgradi sa sljedećim atributima: **identifikator** koji jednoznačno određuje stan, **vlasnik** stana koji je objekt razreda Korisnik, **zgrada** objekt razreda Zgrada označava kojoj zgradi pripada, **površina** broj koji predstavlja površinu stana u metrima kvadratnim
 - Sadrži javne metode za postavljanje i dohvaćanje svih svojih atributa
- Obavijest
 - Razred koji modelira obavijesti koje se nalaze na oglasnoj ploči sa atributima: **identifikator** koji jednoznačno određuje obavijest, **datumKreiranja** je datum kada je korisnik kreirao obavijest, **datumIsteka** je datum kada obavijest ističe te se ona miče sa oglasne ploče, **sadržaj** ima tekst obavijesti i objekt **Korisnik** koji je stvorio obavijest
 - Sadrži javne metode za postavljanje i dohvaćanje svih svojih atributa



Powered by yFiles

Slika 21 Dijagram razreda paketa domain

Razredi koji su u paketu *service*:

- Servis obavijesti korisniku
 - Razred je zadužen za komunikaciju s bazom podataka i upravljanje obavijestima korisnika koje su prisutne u bazi. Njegovi atributi su repozitorij korisnika te repozitorij svih obavijesti
 - Sadrži javne metode za dohvaćanje obavijesti iz baze, pojedinačne i sve obavijesti i sve obavijesti pojedinog stanara. Postoje i javne metode za dodavanje novih te brisanje starih obavijesti
- Servis termina čišćenja snijega
 - Razred je zadužen za komunikaciju s bazom podataka i upravljanje terminima čišćenja koji su prisutni u bazi. Njegovi atributi su repozitorij korisnika te repozitorij svih termina čišćenja snijega
 - Sadrži javne metode za dohvaćanje termina čišćenja iz baze, pojedinačnih i svih termina i svih termina pojedinog stanara. Postoje i javne metode za dodavanje novih i brisanje starih termina te stvaranje rasporeda čišćenja snijega za sve stanare
- Servis korisnika
 - Razred je zadužen za komunikaciju s bazom podataka i upravljanje korisnicima koji su prisutni u bazi. Njegov atribut je repozitorij korisnika
 - Sadrži javne metode za dohvaćanje korisnika iz baze, pojedinačnih i svih korisnika. Postoje i javne metode za dodavanje i brisanje korisnika te metoda za provjeru postoji li korisnik u bazi podataka
- Servis naloga
 - Razred je zadužen za komunikaciju s bazom podataka i upravljanje nalogima koji su prisutni u bazi. Njegov atribut je repozitorij naloga
 - Sadrži javne metode za dohvaćanje naloga iz baze, pojedinačnih i svih naloga. Postoje i javne metode za dodavanje i brisanje naloga
- Servis troška
 - Razred je zadužen za komunikaciju s bazom podataka i upravljanje troškovima koji su prisutni u bazi. Njegov atribut je repozitorij troškova
 - Sadrži javne metode za dohvaćanje troškova iz baze, pojedinačnih i svih troškova. Postoje i javne metode za dodavanje i brisanje troškova
- Servis zgrade
 - Razred je zadužen za komunikaciju s bazom podataka i upravljanje zgradama koje su prisutne u bazi. Njegov atribut je repozitorij zgrada
 - Sadrži javne metode za dohvaćanje zgrada iz baze, pojedinačnih i svih zgrada. Postoje i javne metode za dodavanje i brisanje zgrada

- Servis stana
 - Razred je zadužen za komunikaciju s bazom podataka i upravljanje stanovima koji su prisutni u bazi. Njegov atribut je repozitorij zgrada
 - Sadrži javne metode za dohvaćanje zgrada iz baze, pojedinačnih i svih zgrada. Postoje i javne metode za dodavanje i brisanje zgrada
- Servis obavijesti
 - Razred je zadužen za komunikaciju s bazom podataka i upravljanje obavijestima koji su prisutni u bazi. Njegov atribut je repozitorij obavijesti
 - Sadrži javne metode za dohvaćanje obavijesti iz baze, pojedinačnih i svih obavijesti. Postoje i javne metode za dodavanje i brisanje obavijesti

SnowClearingDateServiceImpl		
f	snowClearingDateRepository	SnowClearingDateRepository
f	userRepository	UserRepository
m	SnowClearingDateServiceImpl(SnowClearingDateRepository, UserRepository)	
m	add(SnowClearingDate)	SnowClearingDate
m	createSchedule(LocalDate, LocalDate)	void
p	all	List<SnowClearingDate>

UserNotificationServiceImpl		
f	userNotificationRepository	UserNotificationRepository
f	userRepository	UserRepository
m	UserNotificationServiceImpl(UserNotificationRepository, UserRepository)	
m	getAllNotificationsForCurrentUser(Long)	List<UserNotification>
m	add(UserNotification)	UserNotification
p	all	List<UserNotification>

AnnouncementServiceImpl		
f	announcementRepository	AnnouncementRepository
m	AnnouncementServiceImpl(AnnouncementRepository)	
m	getById(Long)	Announcement
m	add(Announcement)	Announcement
m	deleteById(Long)	void
p	all	List<Announcement>

CostServiceImpl		
f	costRepository	CostRepository
m	CostServiceImpl(CostRepository)	
m	getById(Long)	Cost
m	add(Cost)	Cost
m	handleUrgentCosts()	void
p	all	List<Cost>

UserServiceImpl		
f	userRepository	UserRepository
m	UserServiceImpl(UserRepository)	
m	getById(Long)	User
m	validateUser(String, String)	User
m	add(User)	User
p	all	List<User>

PaymentOrderImpl		
f	paymentOrderRepository	PaymentOrderRepository
m	PaymentOrderImpl(PaymentOrderRepository)	
m	getById(Long)	PaymentOrder
m	add(PaymentOrder)	PaymentOrder
p	all	List<PaymentOrder>

ApartmentServiceImpl		
f	apartmentRepository	ApartmentRepository
m	ApartmentServiceImpl(ApartmentRepository)	
m	getById(Long)	Apartment
m	add(Apartment)	Apartment
p	all	List<Apartment>

BuildingServiceImpl		
f	buildingRepository	BuildingRepository
m	BuildingServiceImpl(BuildingRepository)	
m	getById(Long)	Building
m	add(Building)	Building
p	all	List<Building>

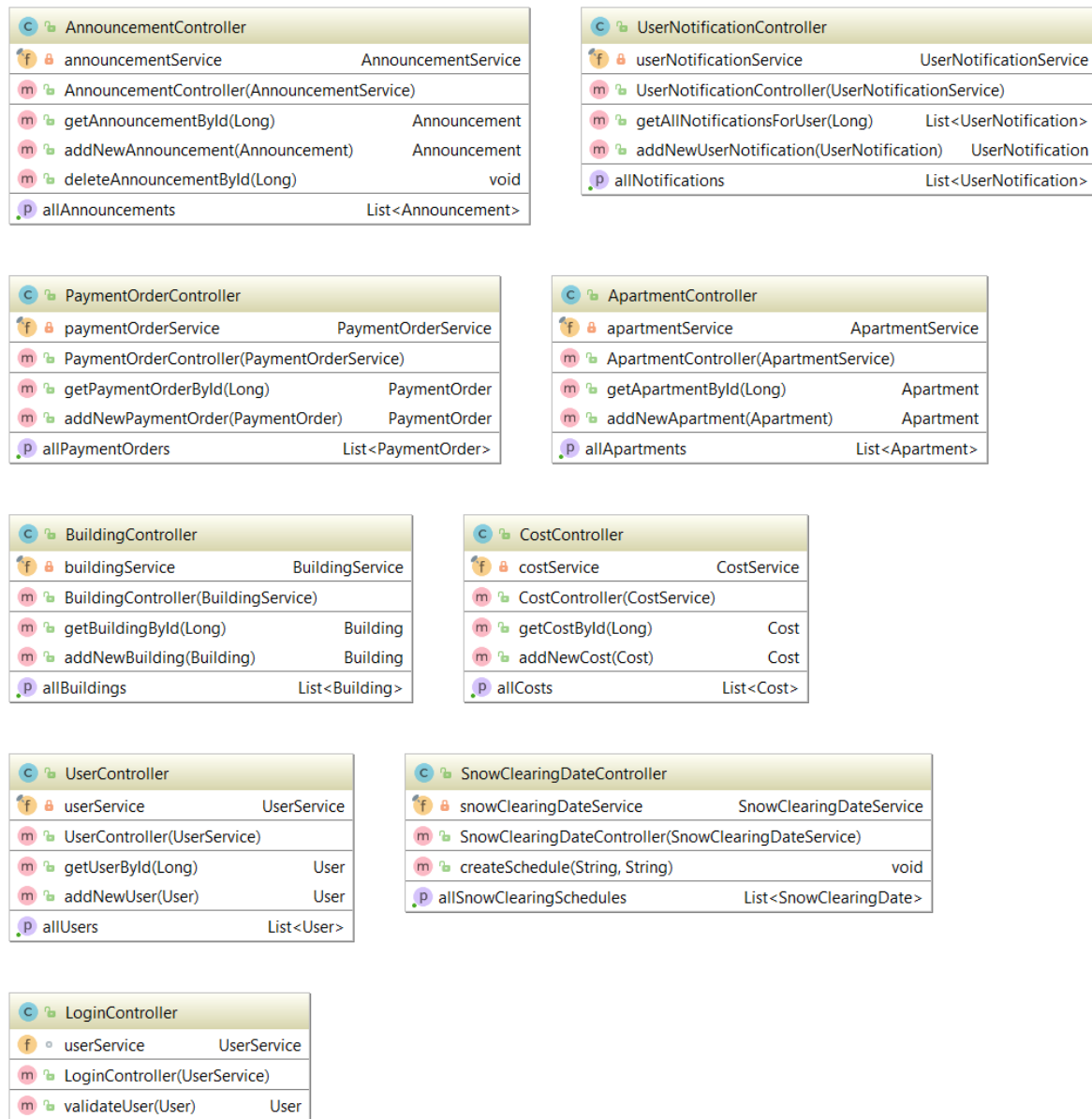
Powered by yf-ies

Slika 22 Dijagram razreda paketa service

Razredi koji su u paketu *controller*:

- Upravitelj obavijesti
 - Razred koji upravlja korisničkim zahtjevima za obradu podataka obavijesti. Njegov atribut je servis obavijesti
 - Sadrži javne metode koje komuniciraju sa servisom obavijesti za dohvaćanje pojedinačnih i svih obavijesti. Postoje i javne metode za dodavanje i brisanje obavijesti
- Upravitelj stana
 - Razred koji upravlja korisničkim zahtjevima za obradu podataka stanova. Njegov atribut je servis stanova
 - Sadrži javne metode koje komuniciraju sa servisom stanova za dohvaćanje pojedinačnih i svih podataka stanova. Postoje i javne metode za dodavanje i brisanje stanova
- Upravitelj zgrade
 - Razred koji upravlja korisničkim zahtjevima za obradu podataka zgrada. Njegov atribut je servis zgrada
 - Sadrži javne metode koje komuniciraju sa servisom zgrada za dohvaćanje pojedinačnih i svih podataka zgrada. Postoje i javne metode za dodavanje i brisanje zgrada
- Upravitelj troškova
 - Razred koji upravlja korisničkim zahtjevima za obradu podataka troškova. Njegov atribut je servis troškova
 - Sadrži javne metode koje komuniciraju sa servisom troškova za dohvaćanje pojedinačnih i svih podataka troškova. Postoje i javne metode za dodavanje i brisanje troškova
- Upravitelj prijave korisnika
 - Razred koji upravlja prijavom korisnika na aplikaciju. Njegov atribut je servis korisnika
 - Sadrži javne metode koje komuniciraju sa servisom korisnika za provjeru ispravnosti podataka koje je korisnik unio pri prijavi na sustav
- Upravitelj naloga
 - Razred koji upravlja korisničkim zahtjevima za obradu podataka naloga. Njegov atribut je servis naloga
 - Sadrži javne metode koje komuniciraju sa servisom naloga za dohvaćanje pojedinačnih i svih podataka zgrada. Postoje i javne metode za dodavanje i brisanje naloga

- Upravitelj termina čišćenja snijega
 - Razred koji upravlja korisničkim zahtjevima za obradu podataka termina čišćenja snijega. Njegov atribut je servis termina čišćenja snijega
 - Sadrži javne metode koje komuniciraju sa servisom termina čišćenja snijega za dohvaćanje pojedinačnih i svih podataka termina čišćenja snijega. Postoje i javne metode za dodavanje i brisanje termina čišćenja snijega te kreiranje rasporeda čišćenja snijega
- Upravitelj korisnika
 - Razred koji upravlja korisničkim zahtjevima za obradu podataka korisnika. Njegov atribut je servis korisnika
 - Sadrži javne metode koje komuniciraju sa servisom korisnika za dohvaćanje pojedinačnih i svih podataka korisnika. Postoje i javne metode za dodavanje i brisanje korisnika
- Upravitelj obavijesti korisnicima
 - Razred koji upravlja korisničkim zahtjevima za obradu podataka obavijesti korisnicima. Njegov atribut je servis obavijesti korisnicima
 - Sadrži javne metode koje komuniciraju sa servisom obavijesti korisnicima za dohvaćanje pojedinačnih i svih podataka obavijesti korisnicima. Postoje i javne metode za dodavanje i brisanje obavijesti korisnicima

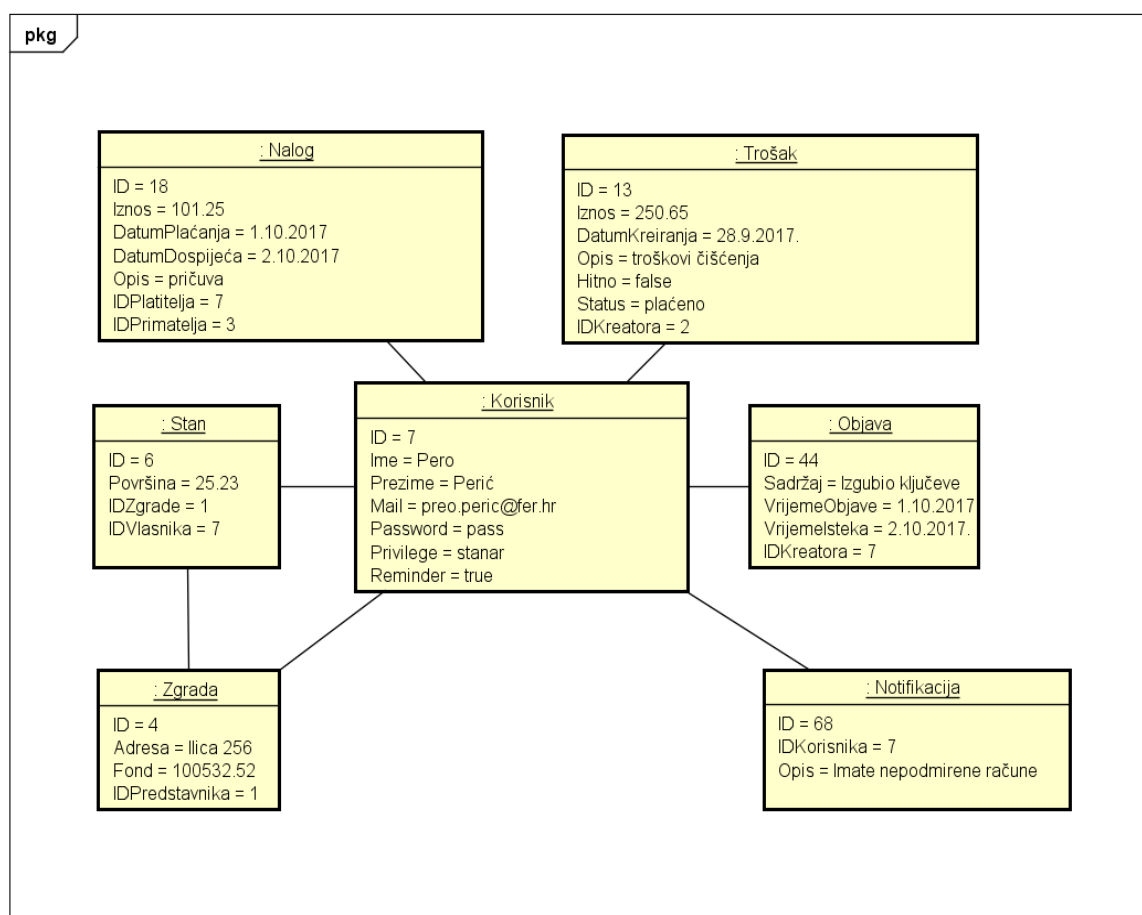


Powered by yfiles

Slika 23 Dijagram razreda paketa controller

6.3. Dijagram objekata

Pomoću dijagrama objekata prikazat ćemo stanje sustava u nekom trenutku. Odabrali smo trenutak kada je na web aplikaciji prijavljen korisnik Pero Perić. Pero je vlasnik stana površine 25.23 m² u Ilici 256. Također vidi svoj plaćeni nalog za 'pričuvu', te također i plaćeni trošak čišćenja koji iznosi 250.65. Stanar Pero je objavio objavu 'Izgubio ključeve', koja ističe dan nakon objavljivanja. Pero ima uključene podsjetnike, a primio je i notifikaciju o nepodmirenim računima.



Slika 24 Dijagram objekata stvarnog stanja sustava

7. Implementacija i korisničko sučelje

7.1. Dijagram razmještanja

7.2. Korištene tehnologije i alati

7.3. Isječak programskog koda vezan za temeljnu funkcionalnost

7.4. Ispitivanje programskog rješenja

7.5. Upute za instalaciju

7.6. Korisničke upute

8. Zaključak i budući rad

Do sada je, što se tiče dokumentacije, napravljena analiza i opis projektnog zadatka te su napravljeni obrasci uporabe, sekvencijski dijagrami, model sustava, dijagrami razreda, dijagram objekata i navedeni ostali zahtjevi koje sustav treba imati. Što se tiče implementacije napravljen je velik dio backenda i manji dio frontenda.

Budući rad na projektu bazirati će se na budućoj implementaciji, većinom na implementaciji frontenda, te pisanju ostatka dokumentacije.

9. Popis literature

1. Oblikovanje programske potpore, FER ZEMRIS,
2. Oblikovanje programske potpore, FER ZEMRIS,
3. I. Sommerville, „Software engineering“, 8th ed, Addison Wesley, 2007.
4. Spring Boot tutorijal,
5. Spring Boot tutorijal,
6. React tutorijal,
7. React-Bootstrap,
8. Redux,
9. Code School,
10. Stackoverflow,

Dodatak A: Indeks (slika, dijagrami, tablica, ispis koda)

S

Slika 1 Funkcionalnosti aplikacije	13
Slika 10 Sekvencijski dijagram obrasca PogledajNaloge	22
Slika 11 Sekvencijski dijagram obrasca PogledajNaloge	23
Slika 12 Sekvencijski dijagram obrasca PrijaviŠtetu	24
Slika 13 Sekvencijski dijagram obrasca OdjavljivanjeKorisnika	25
Slika 14 Sekvencijski dijagram obrasca DodavanjeKorisnika	26
Slika 15 Sekvencijski dijagram obrasca BrisanjeKorisnika	27
Slika 16 Sekvencijski dijagram obrasca UrediObjavuNaOglasnojPloči	28
Slika 17 ER model baze podataka	32
Slika 18 Dijagram razreda paketa domain	36
Slika 19 Dijagram razreda paketa service	38
Slika 2 Sekvencijski dijagram obrasca PrijavlivanjeKorisnika	14
Slika 20 Dijagram razreda paketa controller.....	41
Slika 21 Dijagram razreda paketa controller.....	42
Slika 3 Sekvencijski dijagram obrasca PogledajOglasnuPloču	15
Slika 4 Sekvencijski dijagram obrasca DodajNaOglasnuPloču	16
Slika 5 Sekvencijski dijagram obrasca BrišiSaOglasnePloče	17
Slika 6 Sekvencijski dijagram obrasca PregledajRasporedČišćenjaSnijega.....	18
Slika 7 Sekvencijski dijagram obrasca IzmjeniRasporedČišćenjaSnijega	19
Slika 8 Sekvencijski dijagram obrasca PregledajOpćeInformacije	20
Slika 9 Sekvencijski dijagram obrasca IzmjeniOpćeInforamcije	21

T

Tablica 1 Dnevnik promjena dokumentacije.....	4
Tablica 2 Članovi grupe (abecednim redom)	

Dodatak B: Dnevnik sastajanja

1.sastanak – 18.10.2017.

Prisutni: Ivan Biškup, Marko Cavalli, Marko Ćurlin, Luka Hrgović, Toni Martinčić, Ivana Mršić

Sažetak: Instaliranje svih potrebnih tehnologija za rad na projektu. Kloniranje projekta s Gitlaba u lokalni repozitorij.

2. sastanak – 21.10.2017.

Prisutni: Ivan Biškup, Marko Cavalli, Marko Ćurlin, Luka Hrgović, Toni Martinčić, Ivana Mršić

Sažetak: Učenje korištenja Gitlaba, Spring Boota. Rasprava i proučavanje projektnog zadatka. Početak rada na bazi podataka i backendu.

3. sastanak – 28.10.2017.

Prisutni: Ivan Biškup, Marko Cavalli, Marko Ćurlin, Luka Hrgović, Toni Martinčić, Ivana Mršić

Sažetak: Učenje Reacta i Reduxa. Rad na backendu, početak rada na frontendu i početak pisanja dokumentacije.

4. sastanak – 4.11.2017.

Prisutni: Ivan Biškup, Marko Cavalli, Marko Ćurlin, Luka Hrgović, Toni Martinčić, Ivana Mršić

Sažetak: Rad na dokumentaciji, obrascima uporaba i sekvencijskim dijagramima.

5. sastanak – 11.11.2017.

Prisutni: Ivan Biškup, Marko Cavalli, Marko Ćurlin, Luka Hrgović, Toni Martinčić, Ivana Mršić

Sažetak: Rad na dokumentaciji, ostali zahtjevi, svrha, opći prioriteti, skica sustava, dijagram razreda s opisom, dijagram objekata.

6. sastanak – 15.11.2017.

Prisutni: Ivan Biškup, Marko Cavalli, Marko Ćurlin, Luka Hrgović, Toni Martinčić, Ivana Mršić

Sažetak: Dopršenje izrade dokumentacije, dodavanje dodataka, te zaključak i budući rad na projektu.

Dodatak C: Prikaz aktivnosti grupe

Članovi grupe (abecednim redom)

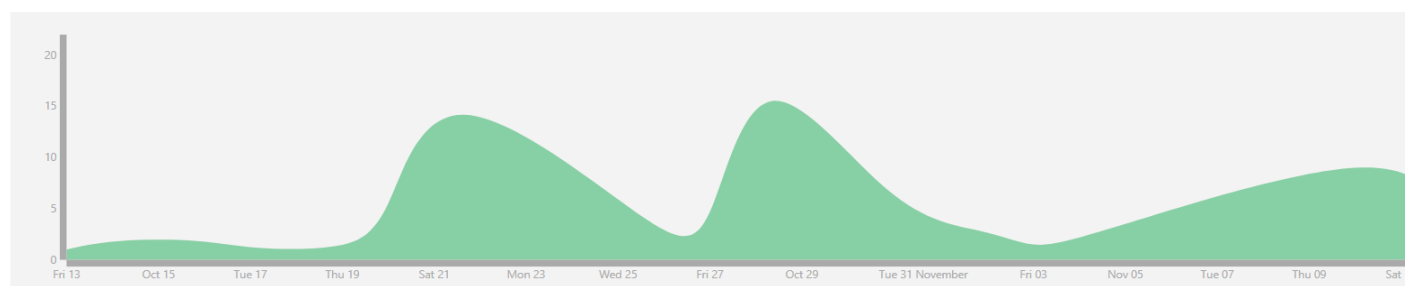
Popis aktivnosti	Toni Martinčić	Ivan Biškup	Marko Cavalli	Marko Ćurlin	Luka Hrgović	Ivana Mršić	
Upravljanje projektom	50%					50%	
Opis projektnog zadatka	50%					50%	
Rječnik pojmova	20%		80%				
Opis funkcionalnih zahtjeva					50%	50%	
Opis ostalih zahtjeva		50%		50%			
Arhitektura i dizajn sustava							
Svrha, opći prioriteti i skica sustava	50%	50%					
Dijagram razreda s opisom							
Dijagram objekata					100%		
Ostali UML dijagrami							
Implementacija i korisničko sučelje							
Dijagram razmještaja							
Korištene tehnologije i alati							
Isječak programskog kôda							
Ispitivanje programskog rješenja							
Upute za instalaciju							
Korisničke upute							
Plan rada							
Pregled rada i stanje	10%	50%	10%	10%	10%	10%	

ostvarenja							
Zaključak budući rad	16.66%	16.66%	16.66%	16.66%	16.66%	16.66%	
Popis literature	50%	10%	10%	10%	10%	10%	
Dodaci							
Indeks						100%	
Dnevnik sastajanja	50%	50%					

Tablica 1 Članovi grupe (abecednim redom)

October 13 2017 - November 12 2017

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



Dodatak D: Plan rada / Pregled rada i stanje ostvarenja

Plan rada za rev. 2 je napraviti dokumentaciju za korisničko sučelje i implementaciju. Isto tako i napraviti samu implementaciju do kraja, te nadopuniti postojeću dokumentaciju.

U rev. 1 smo ostvarili u dokumentaciji:

- opis projektnog zadatka,
- funkcionalne zahtjeve,
- ostale zahtjeve i
- arhitekturu i dizajn sustava.

Također je napravljen dio implementacije koji se odnosi većim dijelom na backend, a jednim manjim dijelom na frontend.