

# Oblikovanje programske potpore

Ak. god. 2018./2019.

## *Dostavljaona*

Dokumentacija, Rev. 2.0

Grupa: *BitSoftTechnologies*

Voditelj: *Luka Mrković*

Datum predaje: 17.1.19.

Nastavnik: *Hrvoje Nuić*

## Sadržaj

1. Dnevnik promjena dokumentacije .....	3
2. Opis projektnog zadatka .....	5
3. Pojmovnik .....	7
4. Funkcionalni zahtjevi.....	8
5. Ostali zahtjevi.....	32
6. Arhitektura i dizajn sustava .....	33
6.1. Svrha, opći prioriteti i skica sustava.....	33
6.2. Dijagram razreda s opisom.....	37
6.3. Dijagram objekata.....	45
6.4. Ostali UML dijagrami .....	45
7. Implementacija i korisničko sučelje .....	50
7.1. Dijagram razmještaja.....	50
7.2. Korištene tehnologije i alati.....	50
7.3. Isječak programskog koda vezan za temeljnu funkcionalnost sustava .....	53
7.4. Ispitivanje programskog rješenja .....	57
7.5. Upute za instalaciju .....	60
7.6. Korisničke upute .....	69
8. Zaključak i budući rad .....	74
9. Popis literature .....	75
Dodatak A: Indeks (slika, dijagrama, tablica, ispisa kôda) .....	76
Dodatak B: Dnevnik sastajanja.....	78
Dodatak C: Prikaz aktivnosti grupe.....	81
Dodatak D: Plan rada / Pregled rada i stanje ostvarenja .....	83

## 1. Dnevnik promjena dokumentacije

Rev.	Opis promjene/dodatka	Autor(i)	Datum
0.1	Dokumentacija spremna za popunjavanje	Mrković	13.10.18.
0.11	Dodani dionici i njihovi funkcionalni zahtjevi	Šterbić	15.10.18.
0.12	Rad na obrascima uporabe	Ramljak	19.10.18.
0.13	Rad na obrascima uporabe	Matešić	20.10.18.
0.14	Rad na obrascima uporabe	Barišić	20.10.18.
0.15	Dodani sekvencijski dijagrami	Šterbić	20.10.18.
0.2	Dodan opis, spojeni obrasci uporabe, formatiranje	Mrković	21.10.18.
0.3	Dorađeni obrasci uporabe	Ramljak	26.10.18.
0.31	Promjena sekvencijskih dijagrama	Šterbić	28.10.18.
0.32	Dorada obrazaca uporabe	Mrković	30.10.18.
0.33	Dodani i dorađeni sekvencijski dijagrami UC1-9	Šterbić	1.11.18.
0.34	Dodani ostali zahtjevi sustava	Žuglić	2.11.18.
0.5	Dodani UC dijagrami, dorada obrazaca uporabe, formatiranje cijelog dokumenta	Mrković	4.11.18.
0.51	Dorada UC dijagrama	Ramljak	4.11.18.
0.6	Svrha, opći prioriteti i skica sustava	Mrković	7.11.18.
0.61	Mali ispravci, dodani ispravni dijagrami	Šterbić	7.11.18.
0.62	Ispravak UC10 i UC11	Matešić	7.11.18.
0.63	Dorada sekvencijskih dijagrama	Barišić	7.11.18.
0.7	Formatiranje, kompozicija, dorada Svrhe, općih prioriteta i skice sustava	Mrković	10.11.18.
0.8	Dijagram razreda	Mrković	13.11.18.
0.81	Dodaci	Barišić	13.11.18.
0.82	Zaključak	Matešić	13.11.18.
0.83	Dijagram razreda – korisničko sučelje	Šterbić	13.11.18.
0.84	Dijagrami razreda - veze	Ramljak	14.11.18.
0.85	Baza podataka	Žuglić	15.11.18.

0.9	Kompozicija, sitna dotjerivanja	Mrković	16.11.18.
0.91	Sitni popravci	Mrković	24.11.18.
0.92	Dodan dijagram objekata	Šterbić	25.11.18.
0.93	Plan rada za drugi ciklus	Matešić	26.11.18.
0.94	Dorada popisa literature	Ramljak	26.11.18.
0.95	Dorada indeksa	Barišić	26.11.18.
0.96	Dorada baze podataka	Žuglić	26.11.18.
<b>1.0</b>		Mrković	27.11.18.
1.01	Popravljeni dijagrami obrazaca uporabe	Mrković	7.12.18.
1.1	Dijagram stanja i razmještaja	Matešić	4.1.19.
1.2	Dijagram aktivnosti i komunikacije	Barišić	4.1.19.
1.3	Dijagram komponenti	Ramljak	5.1.19.
1.4	Upute za instalaciju	Ramljak, Mrković, Žuglić	15.1.19.
1.5	Korisničke upute	Barišić, Matešić, Ramljak	16.1.19.
1.6	Ispitivanje programskog rješenja	Matešić, Barišić	16.1.19.
1.7	Isječak programskog koda	Ramljak	16.1.19.
1.8	Zaključak i dodatci	Matešić, Barišić	16.1.19.
2.0	Završne dorade	Mrković	17.1.19.

## 2. Opis projektnog zadatka

Cilj ovog projekta je razviti aplikaciju za organizaciju dostave hrane *Fina i brza hrana*. Aplikacija će korisniku omogućiti pregled i naručivanje hrane iz više različitih restorana. Restoranima će aplikacija pružiti platformu na kojoj će moći predstaviti i prodavati svoju ponudu.

Unutar aplikacije, ovisno o dodijeljenim dozvolama, razlikujemo sljedeće vrste korisnika:

- Neregistrirani korisnik
- Klijent
- Vlasnik restorana
- Dostavljač
- Dispečer
- Administrator

Neregistrirani korisnici mogu pregledavati ponudu i podatke o svim, u aplikaciji dostupnim, restoranima. Također, svi neregistrirani korisnici imaju opciju stvaranja korisničkog računa.

Jednom kada korisnik stvori korisnički račun, automatski postaje klijent. Kao klijent, može iz ponuda restorana odabrati željene proizvode koje tada može dodati u košaricu. U košarici, klijent ima mogućnost uklanjanja proizvoda od kojih je odustao, te promjene željene količine svakog proizvoda. Kada je zadovoljan odabirom, klijent finalizira narudžbu unosom lokacije dostave na karti. Također, klijent na karti može pratiti trenutnu lokaciju narudžbe (dostavljača). Klijent može predložiti svoj restoran. Nakon unosa svih potrebnih podataka o restoranu, i nakon odobrenja od strane administratora, klijent postaje Vlasnik restorana.

Svaki vlasnik restorana može unositi stavke (proizvode) u meni svog restorana. Također, može uređivati (mijenjati detalje i brisati) već postojeći meni restorana. Aplikacija vlasniku šalje e-mail sa detaljnim opisom narudžbe svaki put kada klijent finalizira narudžbu u vlasnikovom restoranu. Kao vlasnik restorana,

također ima dostupnu listu svih trenutno aktivnih narudžbi (u izradi i dostavi), kao i narudžbi dostavljenih u zadnja dva sata.

Dostavljač prilikom prijave postaje aktivan. U aplikaciji vidi popis zadataka koje mu je dispečer dodijelio. Ti zadaci ne moraju nužno biti iz istih restorana. Također, na karti vidi iscrtanu rutu, pozicije i redoslijed zadataka. Dostavljač mora, nakon završetka zadatka isti označiti gotovim.

Dispečer dodjeljuje zadatke trenutno aktivnim dostavljačima. U aplikaciji vidi listu svih trenutno neraspoređenih narudžbi. Na karti vidi početne i odredišne lokacije neraspoređenih narudžbi, trenutne pozicije svih aktivnih dostavljača i rute kojima oni već moraju proći. Svaka početna lokacija neraspoređene narudžbe pokraj sebe ima prikazano vrijeme preuzimanja (vrijeme kada će biti spremna).

Administrator ima pristup popisu svih aktivnih korisnika, te njihovim osobnim podacima. Mora provjeriti, i ukoliko je restoran valjan, potvrditi prijedloge restorana od strane korisnika. Može svakom korisniku promijeniti razinu pristupa aplikaciji (dozvole). Također, može mijenjati, unositi i brisati proizvode u ponudama svih restorana.

U ovom projektu, fokus nije razvoj API-ja za komuniciranje sa servisima koji će biti potrebni za ostvarenje zadatka, već implementacija objektne strukture koja će omogućiti jednostavno i sigurno funkcioniranje aplikacije. Sve dorade već postojećih open-source API-ja bit će izvršene samo u slučaju da trenutna rješenja ne nude funkcionalnosti potrebne za izradu aplikacije. Uz objektu strukturu u pozadini, značajan napor bit će usmjeren u izradu i dizajn korisničkog sučelja, te prikaz interaktivnih mapa unutar aplikacije.

Po završetku projekta, aplikacija bi trebala predstavljati primamljivo rješenje svim malim restoranima koji žele povećati vidljivost i promet.

### 3. Pojmovnik

- **OPEN-SOURCE SOFTWARE**

- Vrsta programske podrške čiji je izvorni kod izdan pod licencom koja korisnicima dopušta proučavanje, mijenjanje i distribuiranje bilo koje i ubilo koju svrhu

- **API (application programming interface)**

- Skup određenih pravila i specifikacija koje programeri slijede kako bi se mogli služiti uslugama ili resursima operacijskog sustava ili nekog drugog složenog programa kao standardne biblioteke rutina

## 4. Funkcionalni zahtjevi

### Dionici:

- Neregistrirani korisnik
- Klijent
- Vlasnik restorana
- Dostavljač
- Dispečer
- Administrator

### Aktori i njihovi funkcionalni zahtjevi:

- Neregistrirani korisnik, inicijator
  - Može pregledati proizvode svih prijavljenih restorana (ime proizvoda, opis, cijena i slika)
  - Može saznati općenite informacije o restoranu (ime restorana, lokacija na karti, kontakt telefon i slika)
  - Može kreirati novi korisnički račun
- Klijent, inicijator
  - Svaki korisnik koji koristi aplikaciju samo radi naručivanja hrane
  - Može u aplikaciji odabrati željene proizvode iz kataloga, spremati ih u košaricu te završiti kupovinu
  - Može preko aplikacije pratiti trenutnu poziciju narudžbe
  - Može predložiti vlastiti restoran u aplikaciju
- Vlasnik restorana, inicijator
  - Korisnik koji ima ista prava kao i klijent samo mu je dodatno omogućeno upravljanje nad informacijama vlastitog restorana
  - Može unositi, mijenjati i brisati proizvode iz menija vlastitog restorana



- Može pregledavati popis svih trenutno aktivnih narudžbi i narudžbi koje su se dostavile tijekom prošla 2h
- Dispečer, inicijator
  - Korisnik čiji je zadatak rasporediti zaprimljene narudžbe slobodnim dostavljačima i odrediti redoslijed kojim će obaviti dostavu
  - Ima pristup popisu svih neraspoređenih narudžbi, popisu svih aktivnih dostavljača i njima dodijeljenih zadataka.
  - Može dostavljaču dodijeliti zadatak uzmi narudžbu, ostavi narudžbu i idi na novu lokaciju
- Dostavljač, sudionik
  - Korisnik čija je uloga izvršavati zadatke dodijeljene sa strane dispečera
  - Ima pristup informacijama o pozicijama, rutama i redoslijedu dodijeljenih zadataka
  - Pri izvršenju zadatka mora ga u aplikaciji označiti kao gotov
- Administrator, inicijator
  - Registrirani korisnik s najvišim ovlastima
  - Ima neograničen pristup bazi podataka
  - Korisnicima može mijenjati razinu pristupa aplikaciji
  - Može dodavati, mijenjati i brisati proizvode u katalozima restorana
- Baza podataka, sudionik
  - Pohranjuje podatke o svim registriranim korisnicima i njihovim ovlastima
  - Pohranjuje sve podatke o restoranima, narudžbama i rutama

## Opis obrazaca uporabe:

- **UC1: RegistrirajSe**
  - **Glavni sudionik:** Anonimni korisnik
  - **Cilj:** Izrada korisničkog računa
  - **Sudionici:** Baza podataka, poslužitelj
  - **Preduvjet:** Valjani e-mail
  - **Rezultat:** Izrađen korisnički račun
  - **Željeni scenarij:**
    1. Korisnik odabire poveznicu za registraciju
    2. Korisnik unosi sve potrebne podatke i potvrđuje svoj unos
    3. Poslužitelj provjerava ispravnost unesenih podataka i njihovo postojanje u bazi podataka
    4. Poslužitelj prihvaća podatke i upisuje u bazu podataka podatke o novom korisniku
  - **Mogući drugi scenarij:**
    1. Odabrano već zauzeto korisničko ime ili neispravan e-mail
    2. Poslužitelj korisniku javlja da podaci nisu prihvaćeni

- **UC2: *PrijaviSe***

- **Glavni sudionik:** Anonimni korisnik
- **Cilj:** Prijava u aplikaciju
- **Sudionici:** Baza podataka, poslužitelj
- **Preduvjet:** Postoji korisnički račun
- **Rezultat:** Korisnik je prijavljen
- **Željeni scenarij:**
  1. Korisnik odabire opciju za prijavu u aplikaciju
  2. Korisnik upisuje podatke i lozinku
  3. Poslužitelj provjerava ispravnost unesenih podataka.
  4. Korisnik se prijavljuje u aplikaciju
- **Mogući drugi scenarij:**
  1. Korisnik je upisao krive podatke i/ili lozinku
  2. Poslužitelj korisniku javlja da su podaci neispravni

- **UC3: *PregledajPonude***

- **Glavni sudionik:** Anonimni korisnik, Klijent
- **Cilj:** Pregled ponude i podataka restorana
- **Sudionici:** Baza podataka, poslužitelj
- **Preduvjet:** Nema
- **Rezultat:** Korisnik pregledava ponude restorana
- **Željeni scenarij:**
  1. Korisnik u aplikaciji odabire restoran koji ga zanima
  2. Poslužitelj iz baze podataka dohvaća podatke
  3. Korisnik pregledava informacije o restoranu i njegovu ponudu

- **UC4:** *DodajProizvodUKošaricu*

- **Glavni sudionik:** Klijent
- **Cilj:** Dodavanje proizvoda iz kataloga u košaricu
- **Sudionici:** Poslužitelj, baza podataka
- **Preduvjeti:** Nema
- **Rezultat:** Proizvod je dodan u košaricu
- **Željeni scenarij:**
  1. Prijavljen klijent odabire opciju dodaj proizvod
  2. Unosi se promjena u bazi podataka
  3. U košarici se pojavljuje odabrani proizvod

- **UC5:** *UređivanjeKošarice*

- **Glavni sudionik:** Klijent
- **Cilj:** Uređivanje košarice
- **Sudionici:** Poslužitelj, baza podataka
- **Preduvjeti:** Nema
- **Rezultat:** Stanje u košarici je promijenjeno
- **Željeni scenarij:**
  1. Klijent odabire opciju uređivanja košarice
  2. Klijent mijenja količinu proizvoda i/ili briše proizvod iz košarice
  3. Promjene se spremaju u bazu podataka

- **UC6: *PotvrdiNarudžbu***

- **Glavni sudionik:** Klijent
- **Cilj:** Potvrditi narudžbu iz košarice
- **Sudionici:** Poslužitelj, baza podataka
- **Preduvjeti:** Košarica nije prazna
- **Rezultat:** Obavljena narudžba
- **Željeni scenarij:**
  1. Klijent odabire opciju potvrdi narudžbu
  2. Klijent odabire lokaciju na koju želi da se narudžba dostavi
  3. Narudžba se unosi u bazu podataka
- **Mogući drugi scenarij:**
  1. Klijent nema proizvoda u košarici
  2. Poslužitelj javlja korisniku da mu je košarica prazna

- **UC7: *PratiPoziciju***

- **Glavni sudionik:** Klijent
- **Cilj:** Pratiti poziciju dodijeljenog dostavljača
- **Sudionici:** Poslužitelj, baza podataka
- **Preduvjeti:** Klijent ima potvrđenu narudžbu
- **Rezultat:** Klijent na mapi prati trenutnu poziciju narudžbe
- **Željeni scenarij:**
  1. Klijent odabire opciju prati poziciju
  2. Dohvaća se trenutna lokacija i ostali podaci o narudžbi
  3. Klijent uspješno prati poziciju narudžbe

- **UC8: PredložiRestoran**

- **Glavni sudionik:** Klijent
- **Cilj:** Predložiti restoran
- **Sudionici:** Poslužitelj, baza podataka
- **Preduvjeti:** Nema
- **Rezultat:** Prijedlog restorana proslijeđen administratoru
- **Željeni scenarij:**
  1. Klijent odabire opciju predlaganja restorana
  2. Klijent unosi podatke o restoranu
  3. Prijedlog se unosi u bazu podataka
  4. Prijedlog čeka na odobrenje administratora
  5. Nakon odobrenja, klijent dobiva status vlasnika restorana
- **Mogući drugi scenarij:**
  1. Klijent nije unio podatke
    - I Obavijesti klijenta da nije unio potrebne podatke
  2. Prijedlog restorana je odbijen
    - I Poslužitelj briše prijedlog iz baze podataka
    - II Poslužitelj Klijentu javlja da je prijedlog odbijen

- **UC9: UređivanjeMenija**

- **Glavni sudionik:** Vlasnik restorana, Administrator
- **Cilj:** Unijeti promjene u meni
- **Sudionici:** Poslužitelj, baza podataka
- **Preduvjet:** Nema
- **Rezultat:** Meni je promijenjen
- **Željeni scenarij:**
  1. Vlasnik odabire opciju uređivanja menija
  2. Vlasnik unosi promjene u meni
  3. Promijenjeni meni se sprema u bazu podataka

- **UC10:** *PregledPopisaNarudžbi*

- **Glavni sudionik:** Vlasnik restorana
- **Cilj:** Vlasnik restorana pregledava sve trenutno aktivne narudžbe, te one narudžbe koje su bile aktivne u zadnja 2 sata
- **Sudionici:** Poslužitelj, baza podataka
- **Preduvjet:** Postoje narudžbe koje zadovoljavaju kriterije
- **Rezultat:** Omogućen pregled popisa svih trenutno aktivnih narudžbi i narudžbi koje su se dostavile tijekom prošla 2h
- **Željeni scenarij:**
  1. Vlasnik odabire opciju pregleda popisa s narudžbama i dostavljačima
  2. Narudžbe se dohvaćaju iz baze podataka
  3. Vlasnik pregledava narudžbe
- **Mogući drugi scenarij:**
  1. Nema narudžbi koje zadovoljavaju kriterije
  2. Poslužitelj vlasniku javlja da nema trenutno aktivnih narudžbi ni narudžbi aktivnih u zadnja 2 sata

- **UC11:** *PostaviOznakuGotov*

- **Glavni sudionik:** Dostavljač
- **Cilj:** Dostavljač nakon završetka nekog zadatka označava taj zadatak kao gotov
- **Sudionici:** Poslužitelj, baza podataka
- **Preduvjeti:** Dostavljač je obavio njemu dodijeljenu dostavu
- **Rezultat:** Dostavljač označio izvršenje zadatka te prelazi na iduću narudžbu
- **Željeni scenarij:**
  1. Dostavljač odabire opciju postavi oznaku gotovo
  2. Promjena se upisuje u bazu podataka

- **UC12: *PristupiKarti***

- **Glavni sudionik:** Dostavljač
- **Cilj:** Pristupiti karti i saznati pozicije zadanih zadataka
- **Sudionici:** Poslužitelj, baza podataka
- **Preduvjeti:** Dostavljač ima dodijeljene narudžbe
- **Rezultat:** Informacije o pozicijama, rutama i redoslijedu dodijeljenih zadataka se iscrtavaju na karti
- **Željeni scenarij:**
  1. Dostavljač odabire opciju pristupi karti
  2. Dostavljaču se prikazuje karta i informacije o narudžbama

- **UC13: *NeraspoređeneNarudžbe***

- **Glavni sudionik:** Dispečer
- **Cilj:** Prikazivanje neraspoređenih narudžbi, trenutne pozicije dostavljača i njihove buduće rute
- **Sudionici:** Poslužitelj, baza podataka
- **Preduvjet:** Nema
- **Rezultat:** Dispečeru prikazane neraspoređene narudžbe, pozicije dostavljača i njihove buduće rute
- **Željeni scenarij:**
  1. Dispečer odabire opciju prikaza neraspoređenih narudžbi
  2. Iz baze podataka dohvaćaju se potrebni podaci
  3. Dispečeru se u poslužitelju prikazuje lista aktivnih dostavljača, lista neraspoređenih narudžbi i karta s relevantnim informacijama



- **UC14: DodjelaZadatka**

- **Glavni sudionik:** Dispečer
- **Cilj:** Dodjeljivanje zadatka dostavljaču
- **Sudionici:** Poslužitelj, baza podataka
- **Preduvjet:** Postoje neraspoređene narudžbe i aktivni dostavljači
- **Rezultat:** Dostavljač dobio zadatak
- **Željeni scenarij:**
  1. Dispečer odabire opciju dodijele zadatka dostavljačima
  2. Dispečer odabire dostavljača i zadatak (preuzmi ili ostavi narudžbu, idi na novu lokaciju) za njega
  3. Dispečer potvrđuje svoj unos
  4. U bazu podataka se sprema zadani zadatak
  5. Dostavljač dobiva svoj zadatak
- **Mogući drugi scenarij:**
  1. Trenutno nema aktivnih dostavljača
  2. Aplikacija dispečeru javlja da trenutno nema aktivnih dostavljača

- **UC15: OdobriRestoran**

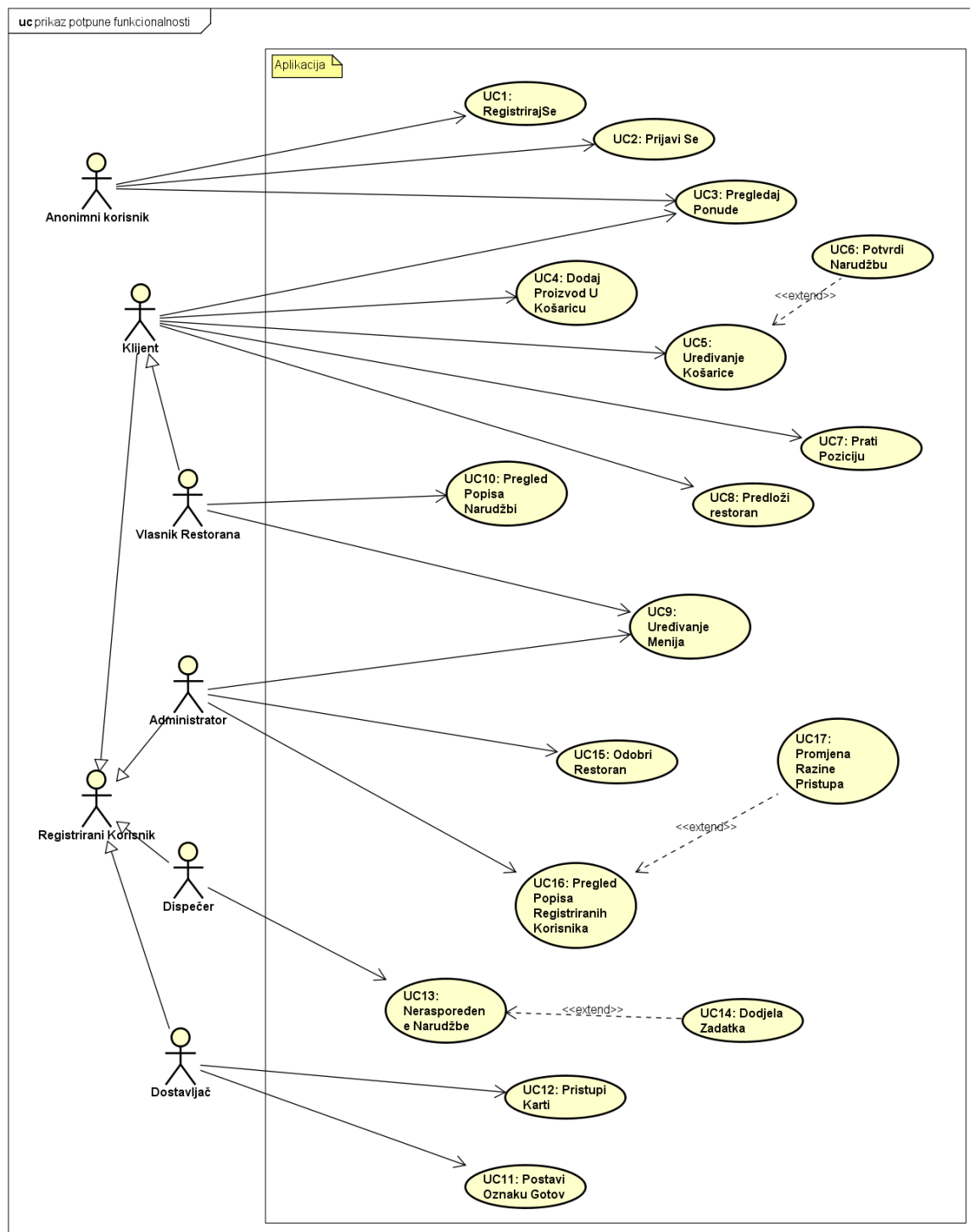
- **Glavni sudionik:** Administrator
- **Cilj:** Odobriti restoran kojeg je predložio klijent
- **Sudionici:** Poslužitelj, baza podataka
- **Preduvjet:** Klijent predložio vlastiti restoran
- **Rezultat:** Restoran odobren od administratora
- **Željeni scenarij:**
  1. Administrator pregledava prijavu
  2. Administrator odabire opciju prihvatanja prijedloga
  3. Odluka administratora sprema se u bazu podatak

- **UC16:** *PregledPopisaRegistriranihKorisnika*

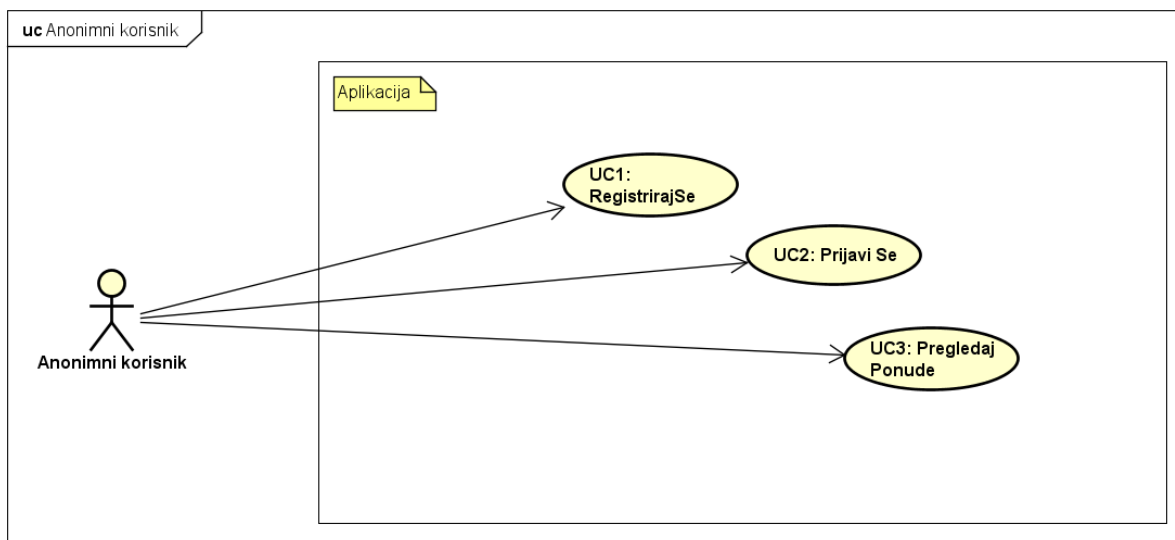
- **Glavni sudionik:** Administrator
- **Cilj:** Pregled popisa svih registriranih korisnika i njihovih osobnih podataka
- **Sudionici:** Poslužitelj, baza podataka
- **Preduvjet:** Nema
- **Rezultat:** Administrator vidi popis svih registriranih korisnika i njihove osobne podatke
- **Željeni scenarij:**
  1. Administrator odabire opciju pregleda popisa
  2. Administratoru se prikazuje popis registriranih korisnika

- **UC17:** *PromjenaRazinePristupa*

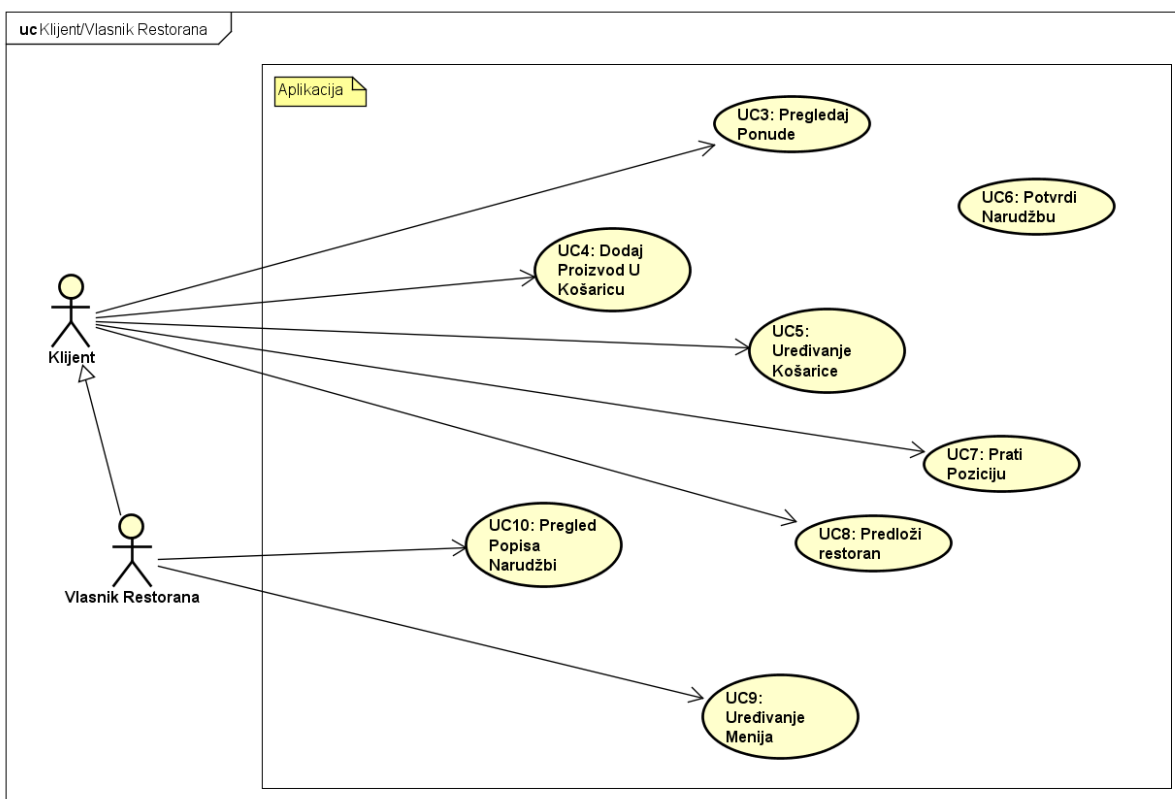
- **Glavni sudionik:** Administrator
- **Cilj:** Promijeniti razinu pristupa korisnika aplikaciji
- **Sudionici:** Poslužitelj, baza podataka
- **Preduvjet:** Postoji barem jedan korisnik osim administratora
- **Rezultat:** Razina pristupa korisnika promijenjena
- **Željeni scenarij:**
  1. Administrator odabire opciju promjene razine pristupa korisnika
  2. Administrator unosi promjenu
  3. Promjena se sprema u bazu podataka



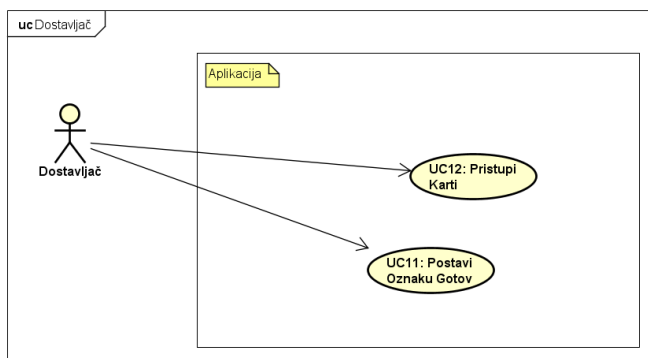
Slika 1: Dijagram obrazaca uporabe - cjeloviti pregled



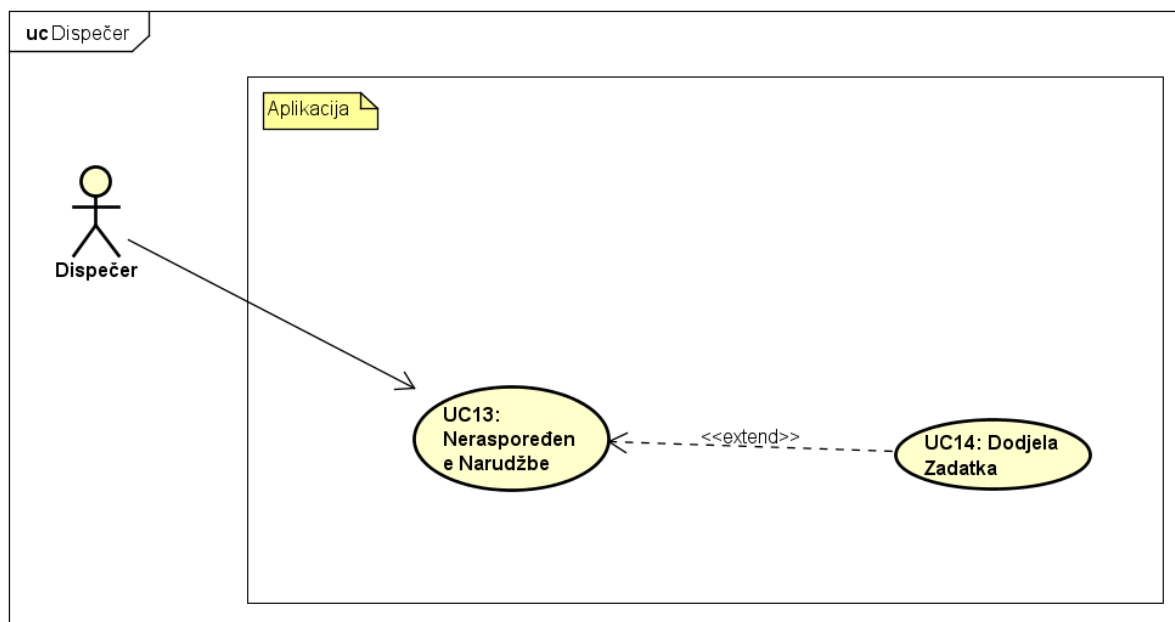
Slika 2: Ponašanje anonimnog korisnika



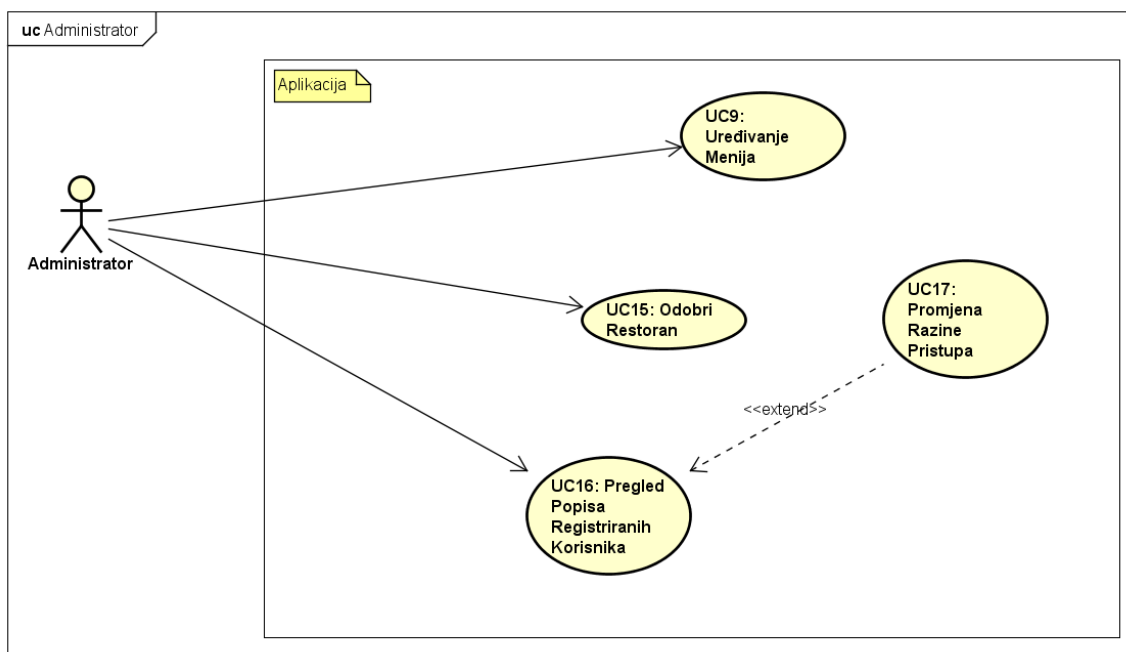
Slika 3: Ponašanje klijenta i vlasnika restorana



Slika 4: Ponašanje dostavljača



Slika 5: Ponašanje dispečera

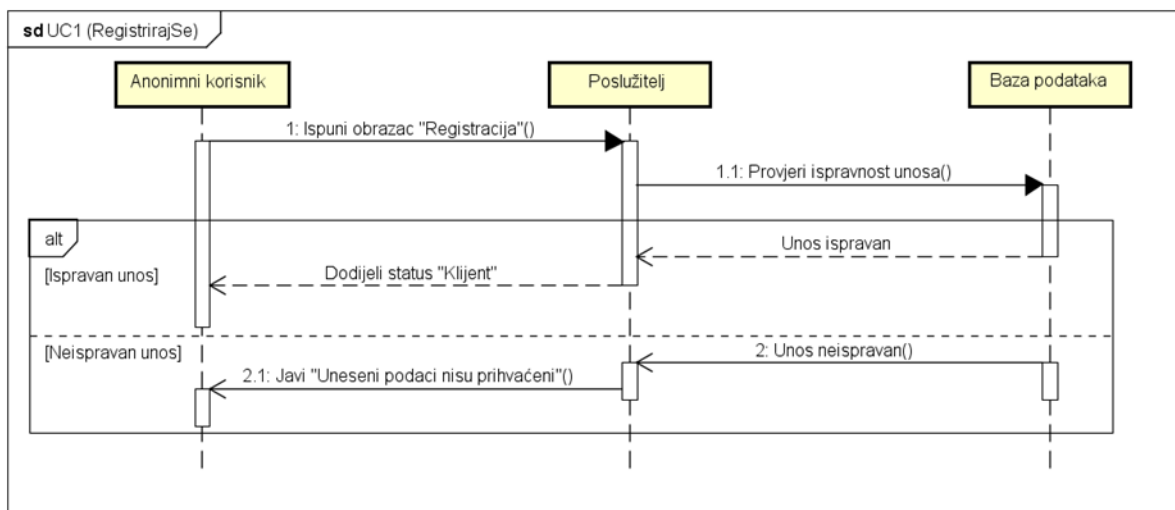


Slika 6: Ponašanje administratora

## Sekvencijski dijagrami:

### Obrazac uporabe UC1 (*RegistrirajSe*):

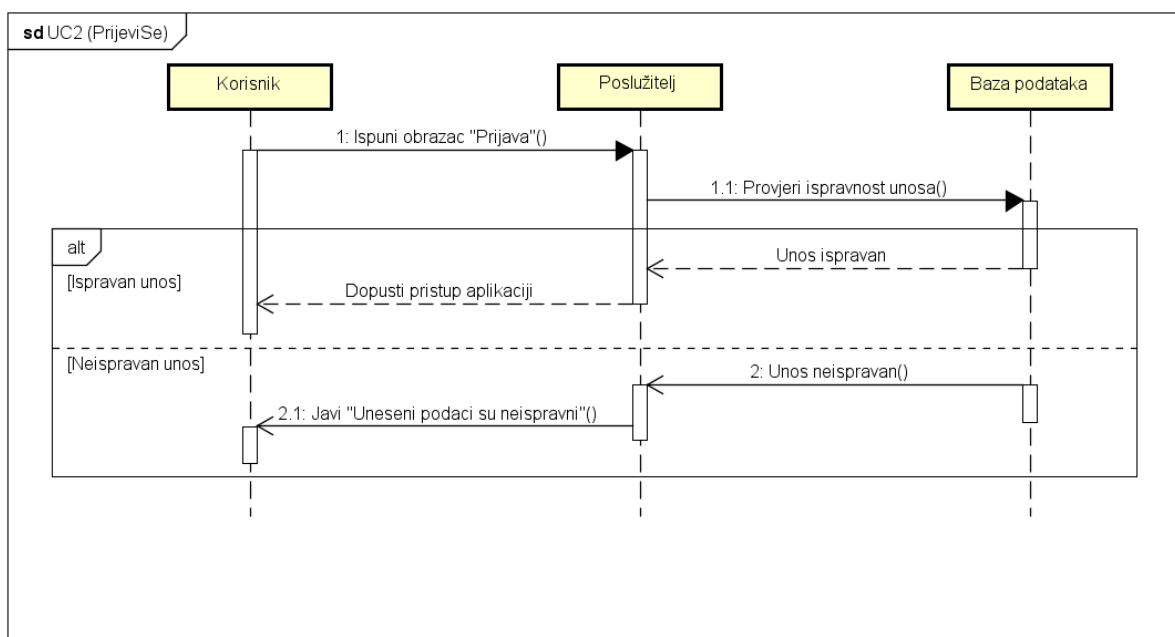
Anonimni korisnik ispunjava obrazac za registraciju. Ako je obrazac ispravno ispunjen, anonimni korisnik postaje klijent. Ako je obrazac nepravilno ispunjen, poslužitelj ispisuje prikladnu poruku anonimnom korisniku.



Slika 7: Sekvencijski dijagram - Registriraj se

### Obrazac uporabe UC2 (*PrijaviSe*):

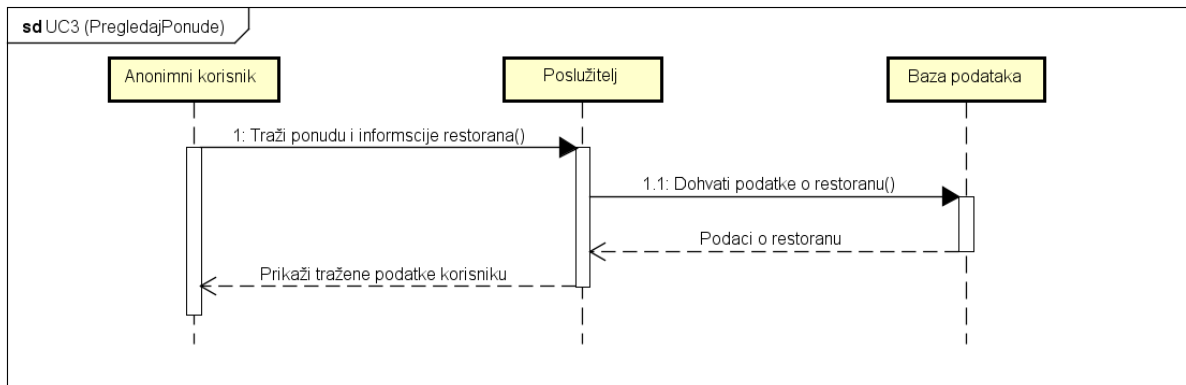
Korisnik ispunjava obrazac za prijavu. Ako su podaci dani u obrascu ispravni, korisniku je dan pristup aplikaciji. Ako podaci nisu ispravni, poslužitelj ispisuje prikladnu poruku.



Slika 8: Sekvencijski dijagram - Prijavi se

**Obrazac uporabe UC3 (PregledajPonude):**

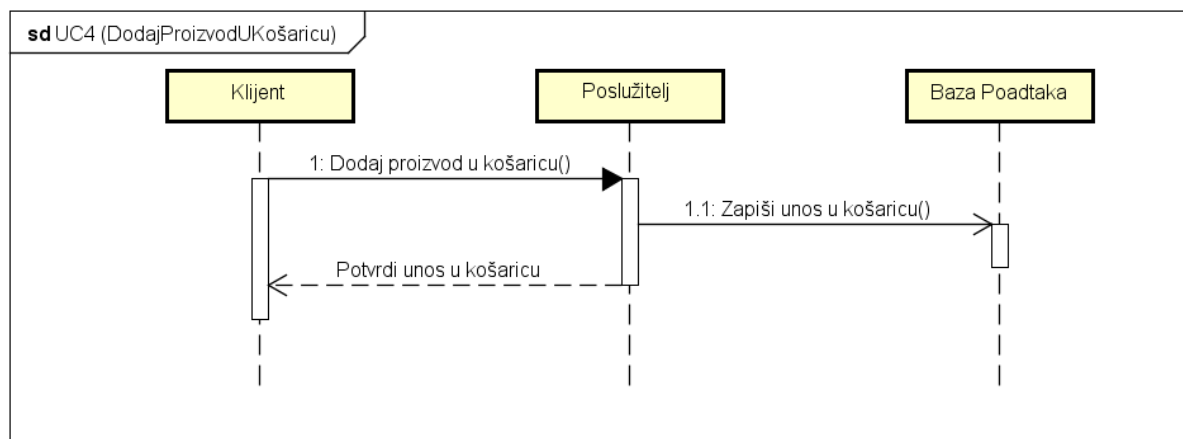
Anonimni korisnik može tražiti pregled informacija i ponude određenog restorana pri čemu se poslužitelj spaja na bazu podataka i prosljeđuje tražene informacije anonimnom korisniku.



Slika 9: Sekvencijski dijagram – Pregledaj ponude

**Obrazac uporabe UC4 (DodajProizvodUKošaricu):**

Korisnik šalje sučelju zahtjev za unos proizvoda u košaricu. Poslužitelj javi bazi podataka da zapiše unos u košaricu te pošalje korisniku potvrdu o uspješnom unosu u košaricu.

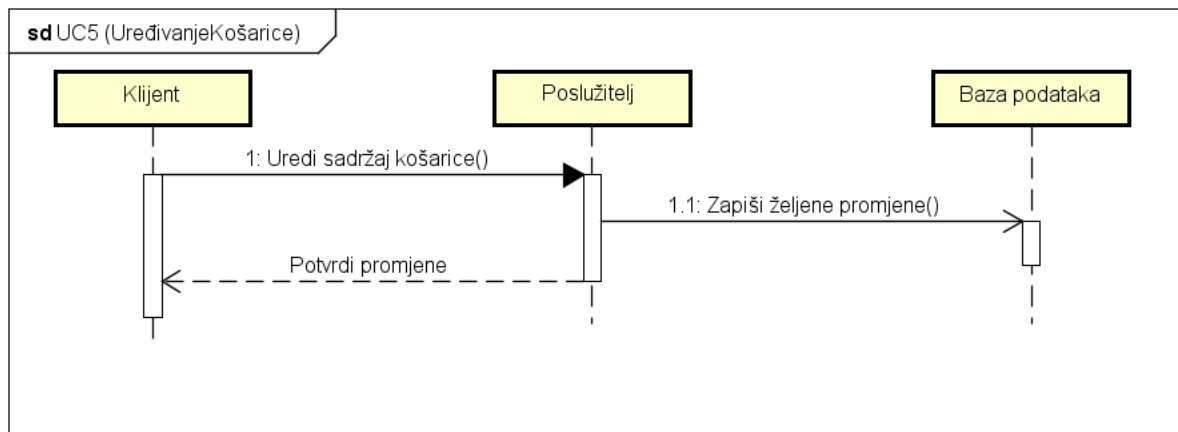


Slika 10: Sekvencijski dijagram – Dodaj proizvod u košaricu



**Obrazac uporabe UC5 (UređivanjeKošarice):**

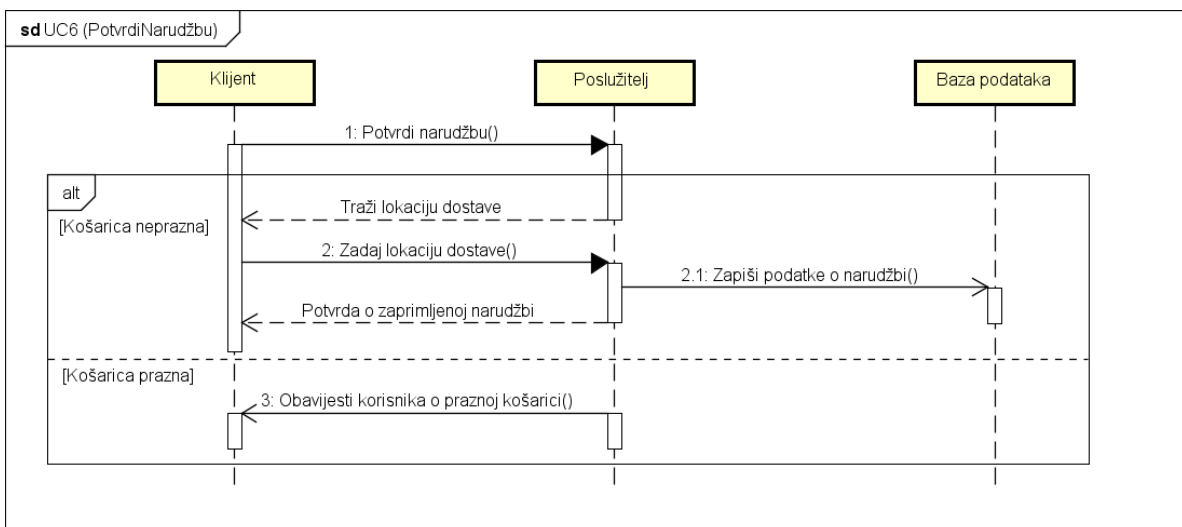
Korisnik uređuje košaricu (mijenja količinu proizvoda ili uklanja proizvod) preko poslužitelja. Tada poslužitelj zahtjeva od baze podataka da zapiše promjene te pošalje korisniku potvrdu o uspješnoj provedbi promjena.



Slika 11: Sekvencijski dijagram – Uređivanje košarice

**Obrazac uporabe UC6 (PotvrdiNarudžbu):**

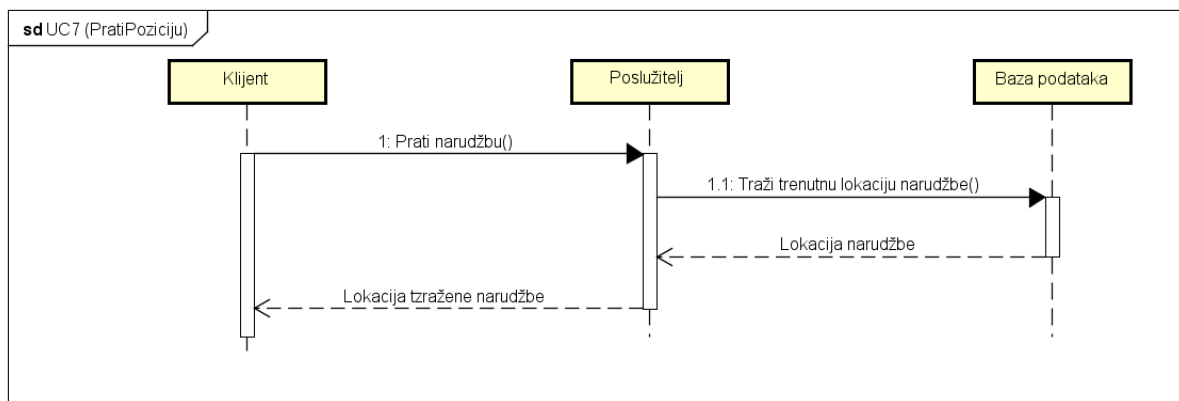
Klijent odabire opciju za potvrdu narudžbe, pri čemu poslužitelj traži od korisnika da unese lokaciju na koju želi da se narudžba dostavi. Kada klijent unese lokaciju poslužitelj zapisuje narudžbu u bazu podataka. Ako je košarica prazna pri pokušaju potvrde narudžbe poslužitelj šalje klijentu poruku da košarica ne smije biti prazna.



Slika 12: Sekvencijski dijagram – Potvrdi narudžbu

**Obrazac uporabe UC7 (*PratiPoziciju*):**

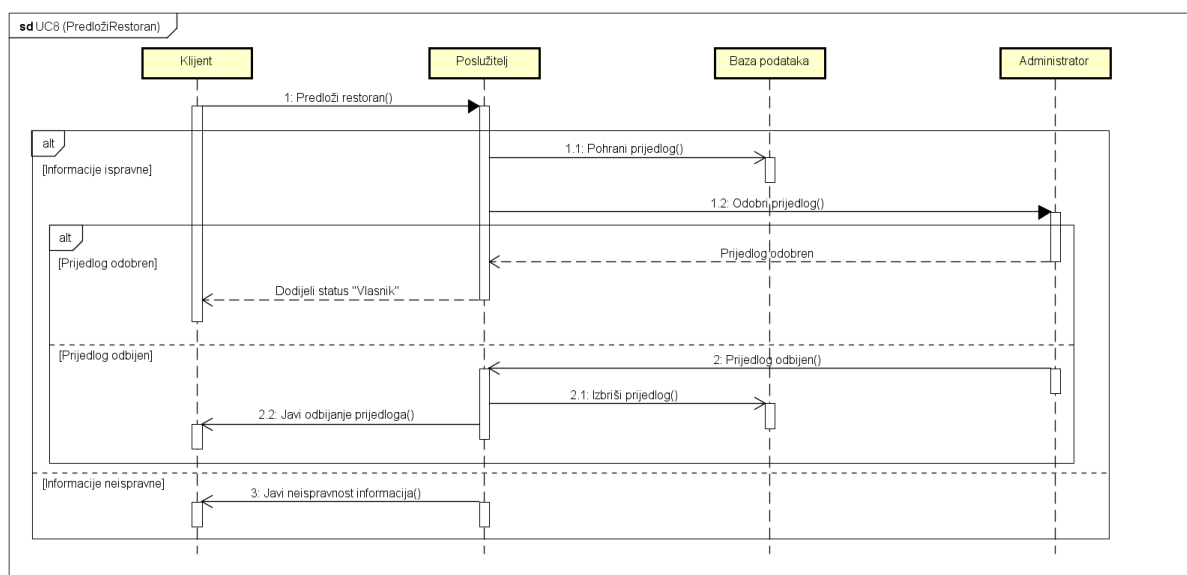
Klijent može potražiti poziciju vlastite narudžbe pri čemu se poslužitelj spaja na bazu podataka, pribavlja informaciju o poziciji narudžbe i prosljeđuje ju klijentu.



Slika 13: Sekvencijski dijagram – Prati poziciju

**Obrazac uporabe UC8 (*PredložiRestoran*):**

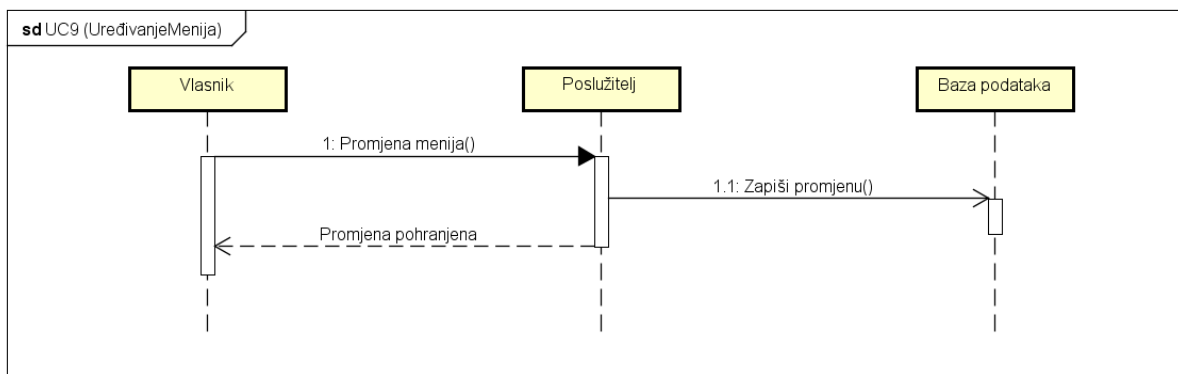
Klijent može predložiti restoran tako da poslužitelju proslijedi informacije o restoranu. Ako su informacije ispravne upisuju se u bazu podataka. Tada se čeka na administratora koji mora odobriti predložak. U slučaju odobrenja restoran postaje dostupan za naručivanje i klijent dobiva status vlasnika. Ako zahtjev bude odbijen tada se informacije izbrišu iz baze podataka i klijentu se ispiše prikladna poruka. Ako dane informacije samo po sebi nisu prikladne klijentu se samo ispiše poruka o pogrešci.



Slika 14: Sekvencijski dijagram – Predloži restoran

**Obrazac uporabe UC9 (UređivanjeMenija):**

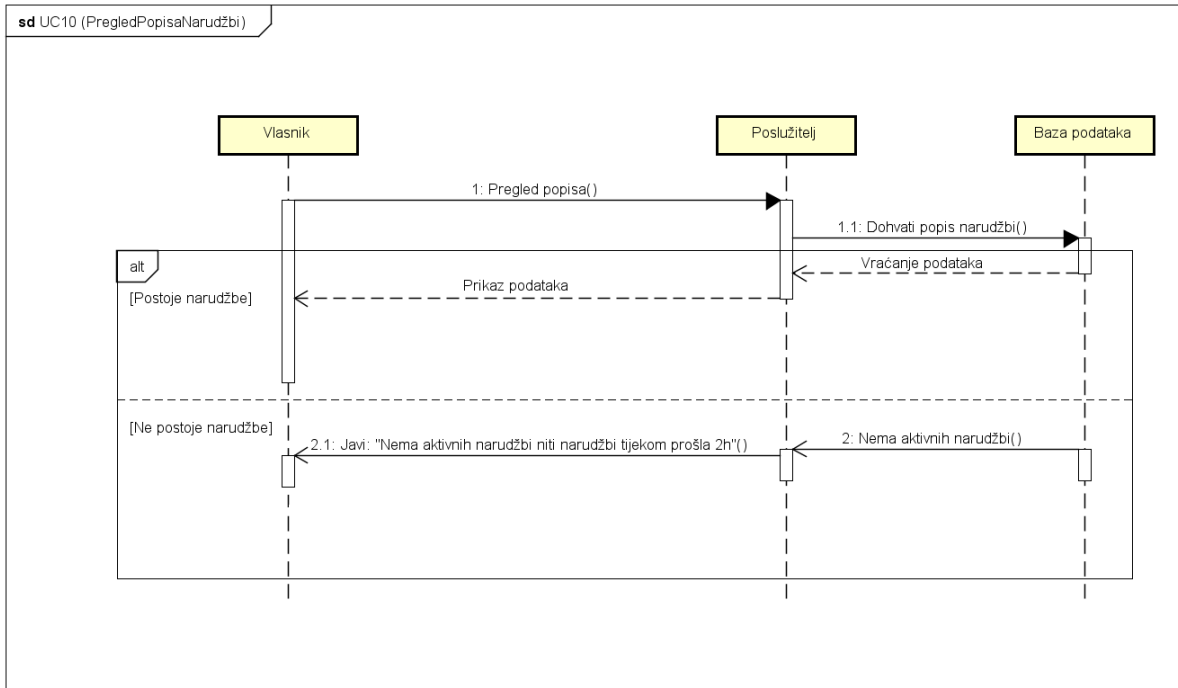
Vlasnik preko poslužitelja može modificirati sadržaj menija svog restorana. Kada vlasnik je zadovoljan promjenama, poslužitelj ih pohranjuje u bazu podataka.



Slika 15: Sekvencijski dijagram – Uređivanje menija

**Obrazac uporabe UC10 (PregledPopisaNarudžbi):**

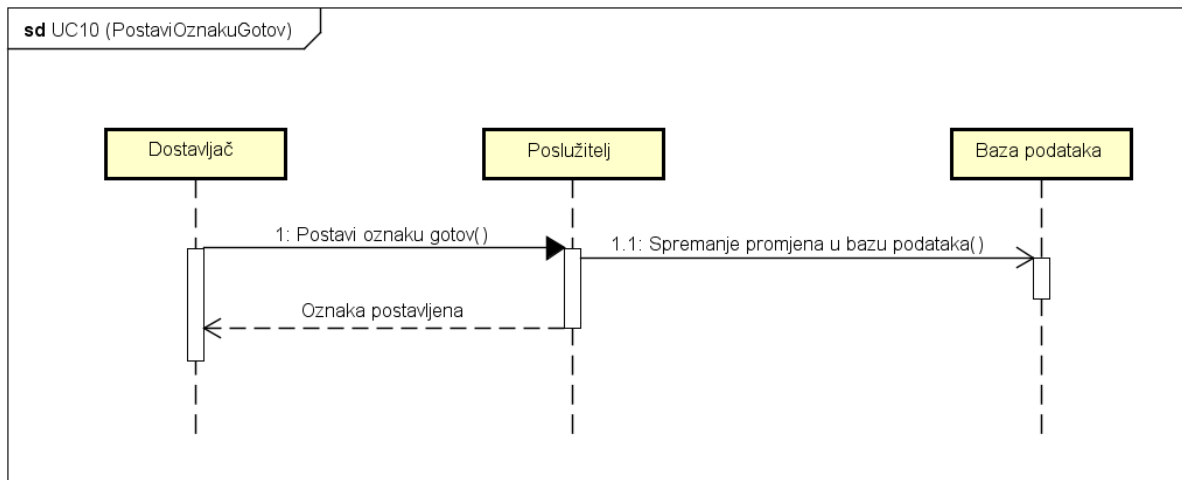
Korisnik odabire opciju pregleda popisa narudžbi na poslužitelju. Poslužitelj zahtjeva pristup popisu u bazi podataka. Baza podataka šalje popis poslužitelju, nakon čega se popis prikazuje korisniku



Slika 16: Sekvencijski dijagram – Pregled popisa narudžbi

**Obrazac uporabe UC11 (*PostaviOznakuGotov*):**

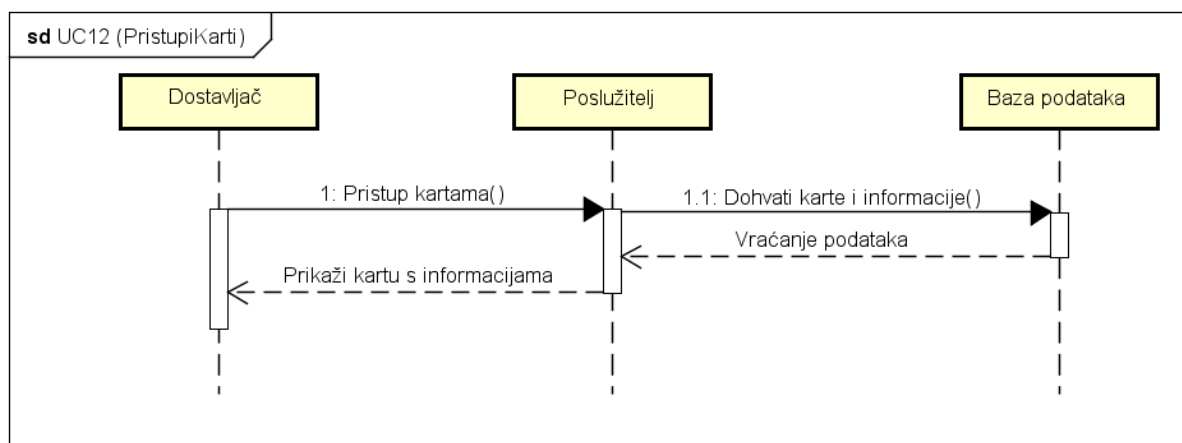
Dostavljač koji je prijavljen u aplikaciju, nakon izvršenja dodijeljenog zadatka, postavlja oznaku gotov tako što odabire opciju „gotov“ na poslužitelju. Njegova se promjena aktivnosti sprema u bazu podataka.



Slika 17: Sekvencijski dijagram – Postavi oznaku gotovo

**Obrazac uporabe UC12 (*PristupiKarti*):**

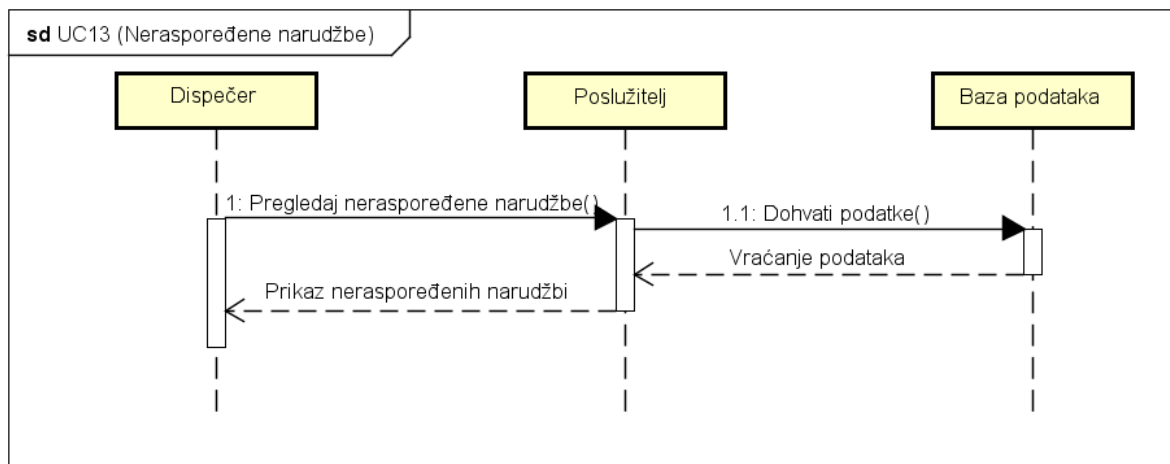
Dostavljač, nakon dodjele zadatka, može odabrati opciju „Pristupi karti“, kojom se informacije o pozicijama, rutama i redoslijedu dodijeljenih zadataka iscrtavaju na karti. Podaci se uzimaju iz baze podataka i preko poslužitelja prikazuju korisniku.



Slika 18: Sekvencijski dijagram – Pristupi karti

**Obrazac uporabe UC13 (Neraspoređene Narudžbe):**

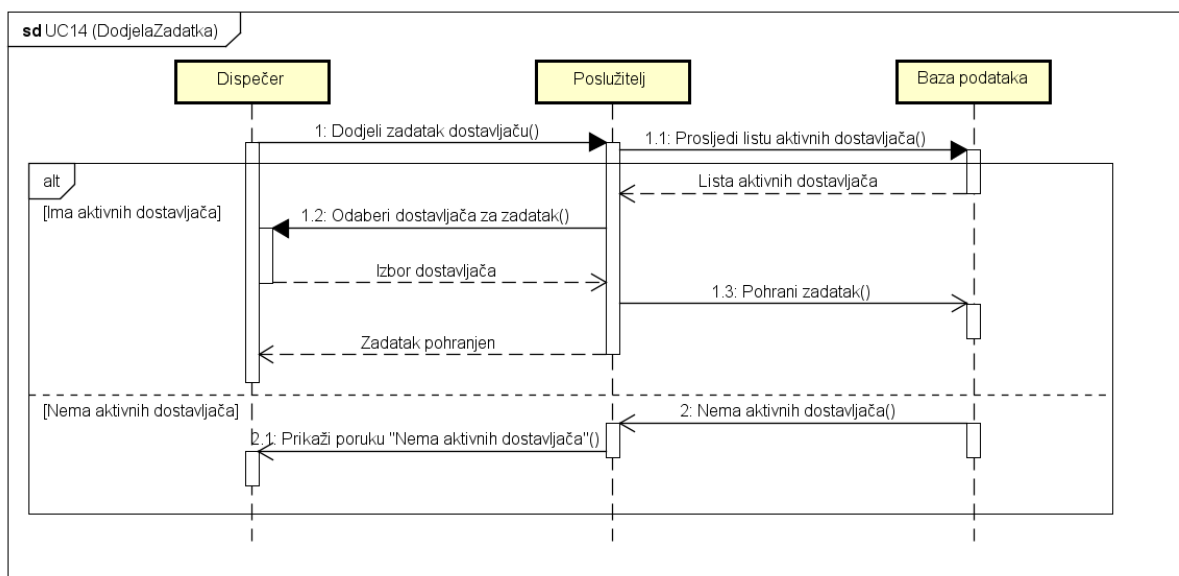
Dispečer ima mogućnost odabira opcije za prikaz neraspoređenih narudžbi. Odabirom te opcije dispečeru se prikazuje karta sa neraspoređenim narudžbama, trenutnim pozicijama dostavljača i njihovim budućim rutama, iz baze podataka.



Slika 19: Sekvencijski dijagram – Neraspoređene narudžbe

**Obrazac uporabe UC14 (DodjelaZadatka):**

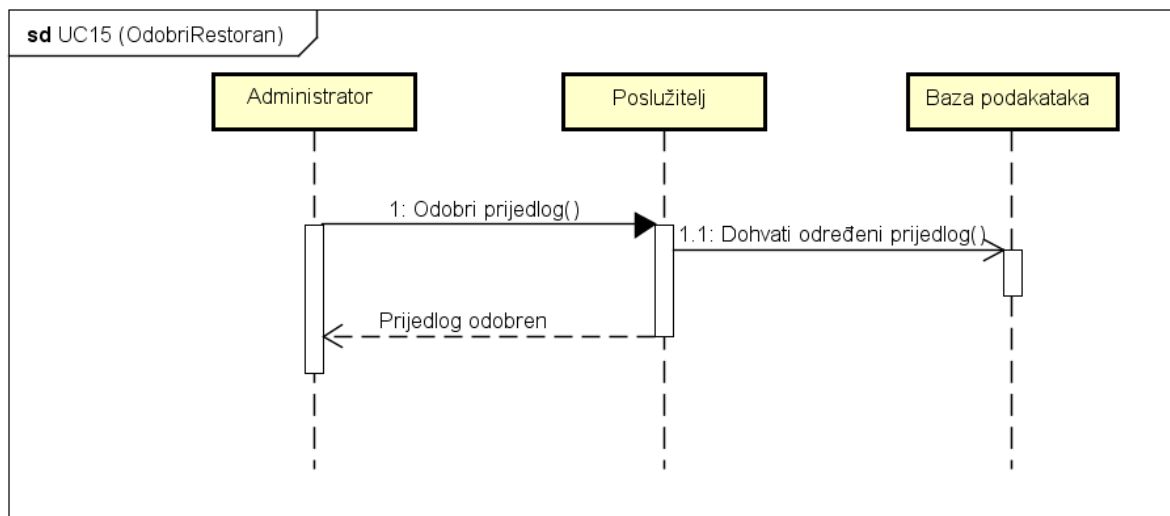
Zadatak dispečera je raspodijeliti zadatke. Poslužitelj se spaja na bazu podataka te uzima popis aktivnih dostavljača. U slučaju da nema aktivnih dostavljača, poslužitelj šalje poruku dispečeru. Nakon što dispečer odabere zadatak i dostavljača, poslužitelj javi bazi podataka da upiše zadatak te se pošalje poruka dispečeru o uspješno dodijeljenom zadatku.



Slika 20: Sekvencijski dijagram – Dodjela zadatka

**Obrazac uporabe UC15 (OdobriRestoran):**

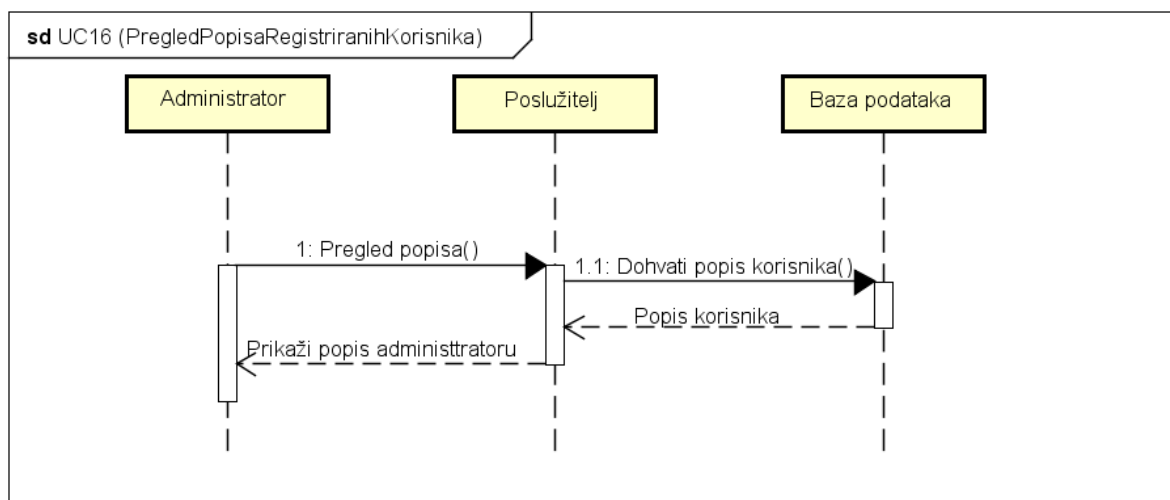
Administrator može odobriti restoran predložen od strane korisnika. Poslužitelj preko baze podataka dohvaća zadani prijedlog i šalje ga administratoru. Administrator odobrava prijedlog i on ostaje pohranjen u bazi podataka.



Slika 21: Sekvencijski dijagram – Odobri restoran

**Obrazac uporabe UC16 (PregledPopisaRegistriranihKorisnika):**

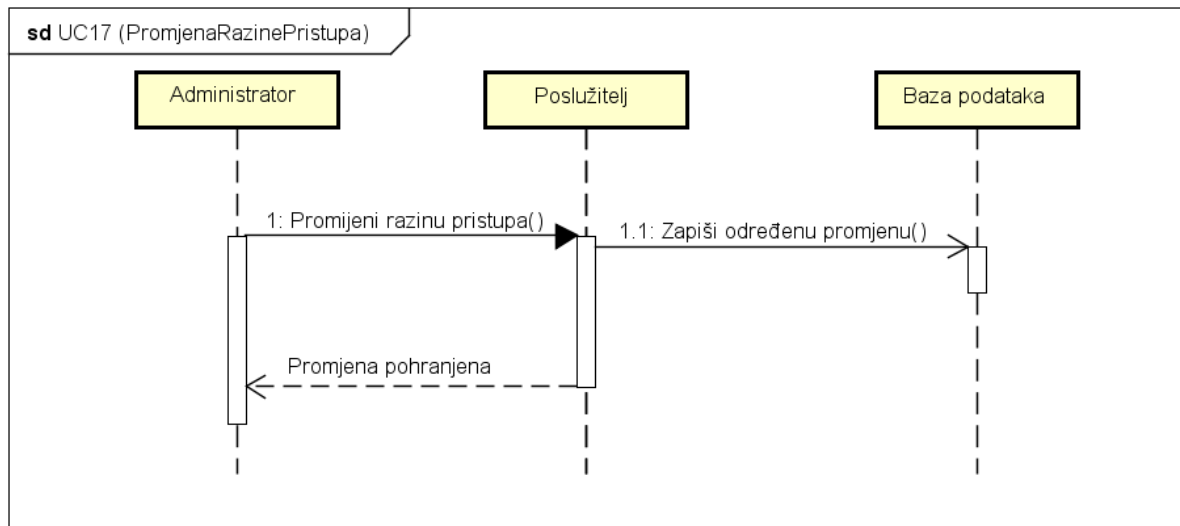
Ukoliko administrator želi pregledati popis registriranih korisnika, poslužitelju se šalje zahtjev za pregled popisa. Poslužitelj se spaja na bazu podataka, pribavlja zadani popis te ga šalje administratoru.



Slika 22: Sekvencijski dijagram – Pregled popisa registriranih korisnika

**Obrazac uporabe UC17 (PromjenaRazinePristupa):**

Administrator može mijenjati razinu pristupa korisnika. Nakon što unese odgovarajuće promjene, poslužitelj ih pohranjuje u bazu podataka.



Slika 23: Sekvencijski dijagram – Promjena razine pristupa

## 5. Ostali zahtjevi

- Korisnicima se ne smije dopustiti pristup aktivnostima za koje nemaju ovlasti
- Sustav mora biti neosjetljiv na pogreške izazvane nepravilnim ponašanjem korisnika
- Korisničko sučelje mora biti lako razumljivo
- Sustav mora izvršiti svaku aktivnost iniciranu od strane korisnika u trajanju do 15 sekundi
- Sustav mora pružati točne informacije pri svakom upitu
- Buduće nadogradnje sustava ne smiju ugroziti funkcionalnost sustava



## 6. Arhitektura i dizajn sustava

### 6.1. Svrha, opći prioriteti i skica sustava

Za ovaj projekt odabrali smo arhitekturu **desktop aplikacije**. Ta odluka donesena je iz više razloga:

Aplikacija mora omogućiti jednostavno i koordinirano korištenje od strane više korisnika. Obzirom na to da je bitan dio funkcionalnosti aplikacije vođenje i održavanje ponude i prometa restorana, odabiremo aplikaciju koja će biti pokretana na stolnim i prijenosnim računalima. Odluka je temeljena na višoj razini povjerenja koje ljudi imaju u desktop aplikacije u usporedbi s mobilnim aplikacijama kada je u pitanju baratanje osjetljivim podacima. U desktop okruženju imamo izbor između desktop i web aplikacije. Odabiremo desktop aplikaciju zbog iskustva svih članova tima u razvoju desktop aplikacija, nasuprot vrlo limitiranog iskustva članova tima u razvoju web aplikacija.

Sustav se na najvišoj razini dijeli na dva podsustava:

1. **Desktop aplikacija**
2. **Baza podataka**

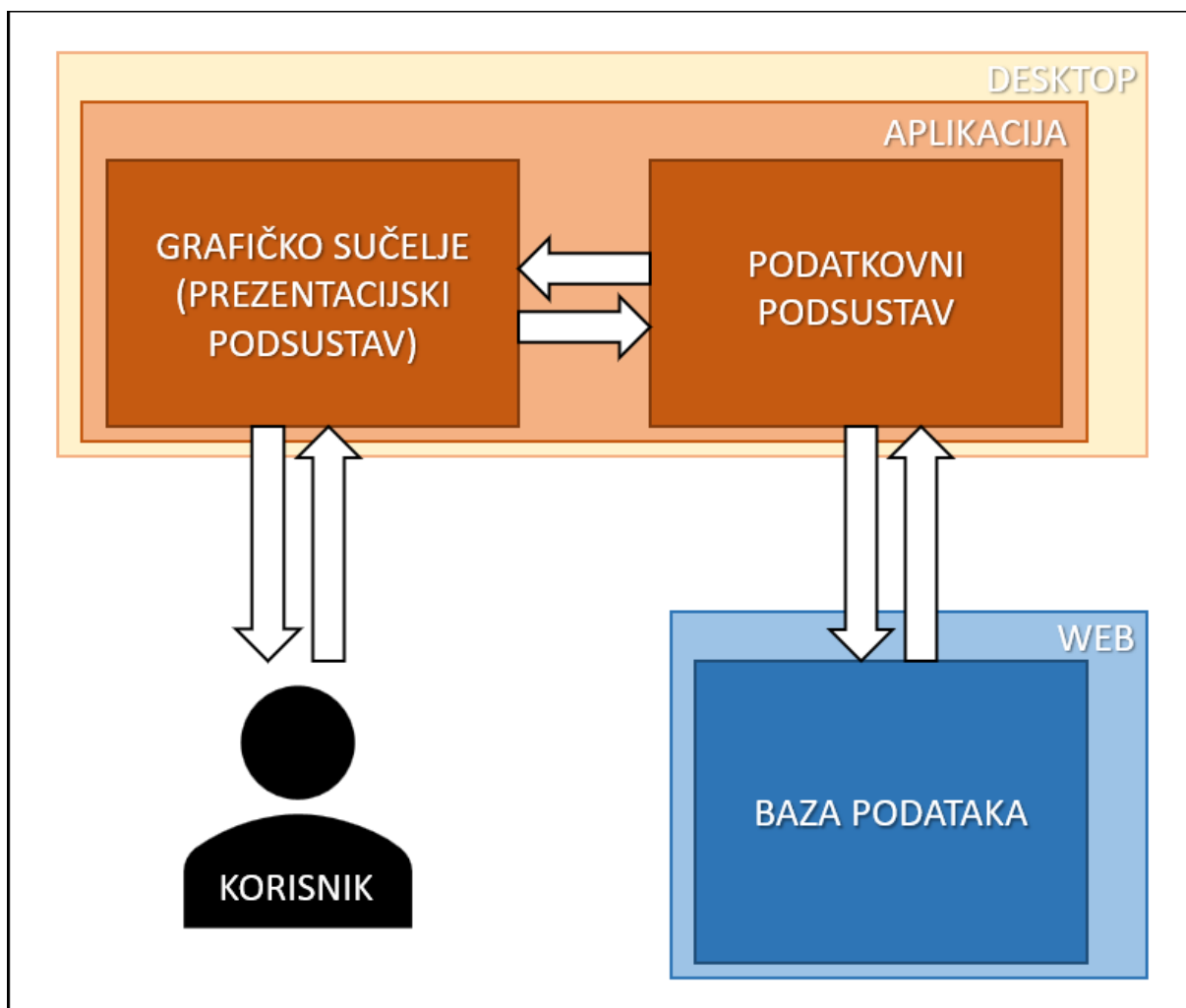
Desktop aplikacija će se dalje dijeliti na dva velika podsustava:

1. **Grafičko sučelje** (Prezentacijski podsustav)
2. **Podatkovni podsustav**

Rad s aplikacijom će prolaziti kroz ovu petlju:

1. Korisnik u grafičkom sučelju odabire akciju koju želi izvršiti (sve akcije su neki oblik baratanja podacima)
2. Podatkovni podsustav pohranjuje ili dohvaća željene podatke iz baze podataka (web)
3. Podatkovni podsustav obavlja operacije nad podacima (ako je potrebno)

4. Aplikacija korisniku kroz grafičko sučelje daje povratnu informaciju o izvršenoj akciji (ispis podataka, poruku o uspješnom unosu itd.)

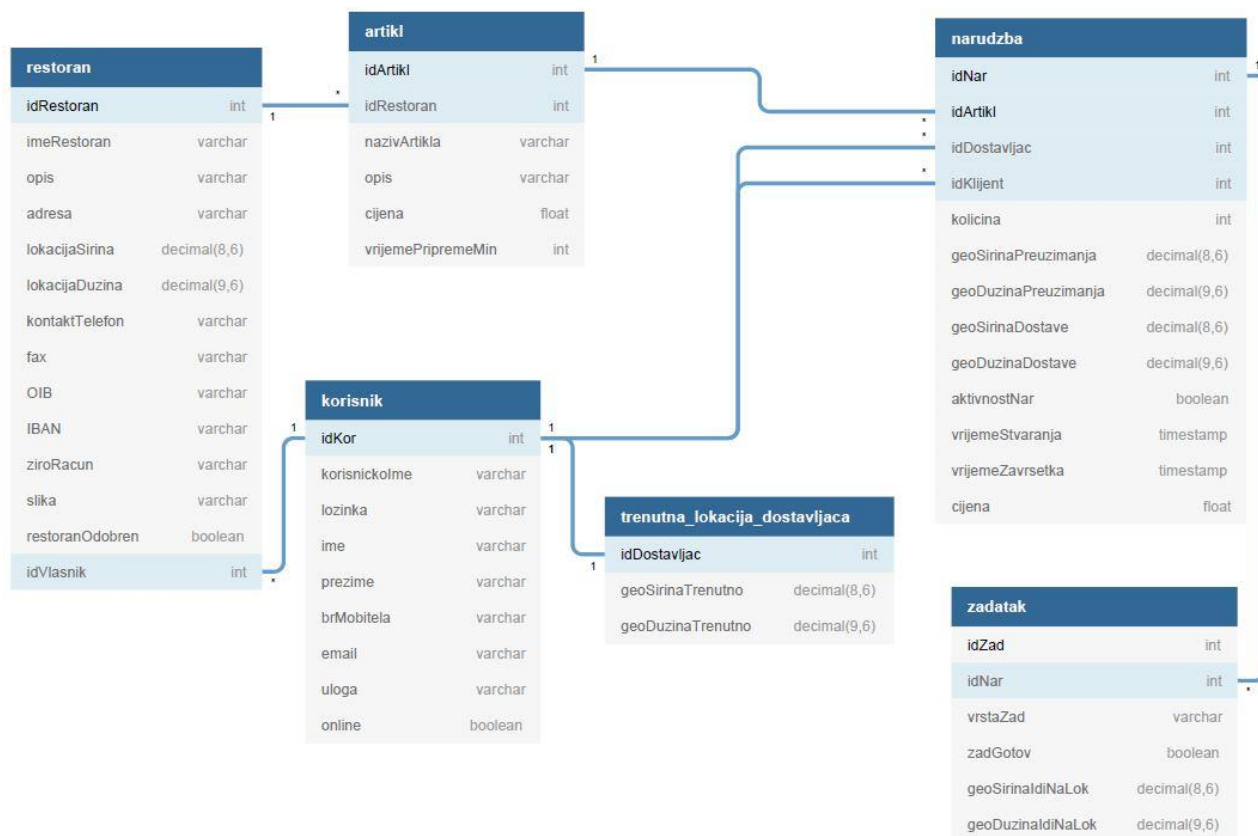


Slika 24: Nacrt sustava

## Baza podataka

Baza podataka je dio sustava zadužen za pohranu, izmjenu ili unos podataka. Baza podataka ovog sustava sastoji se od sljedećih entiteta:

- **Korisnik** – Sadrži podatke o korisniku koji su privatni ili javni. Također sadrži i ulogu svakog korisnika koja određuje koje su operacije nad bazom podataka i aplikacijom dostupne tom korisniku. Korisnik ovisno o ulozi može biti klijent, vlasnik, dostavljač, dispečer ili administrator i određen je identifikacijskim brojem.
- **Restoran** – Sadrži podatke o restoranu kojeg je predložio neki korisnik koji je ujedno i vlasnik tog restorana. Predlaganjem i prihvaćanjem prijedloga restorana, korisnik koji ga je predložio se upisuje u atribut *idVlasnik* i u atribut *restoranOdobren* se upiše true. Svaki restoran je određen jedinstvenim identifikacijskim brojem.
- **Artikl** – Sadrži podatke o artiklima u ponudi restorana. Svaki artikl je jednoznačno određen jedinstvenim identifikacijskim brojem.
- **Narudžba** – Sadrži podatke o svakoj narudžbi za koju se napravi zahtjev. Narudžba je određena jedinstvenim brojem svake narudžbe i brojem koji označava pripadnost istoj narudžbi. Ako jedna narudžba za jednog korisnika ima više artikala iz istog restorana, oni će dijeliti isti *idNar* a imat će različit atribut *idArtikl* ovisno koliko je različitih artikala naručeno iz tog restorana. Količina svakog artikla pohranjena je u atributu *kolicina*. U atributima vezanima za koordinate spremaju se geografske širine i dužine od kojih se one vezane za dostavu zadaju od strane klijenta naručitelja dok se u one vezane za preuzimanje upisuje lokacija restorana. Atribut *aktivnostNar* označava je li narudžba u postupku izrade i dostave(true) ili je neaktivna(false).
- **Trenutna\_lokacija\_dostavljača** – Sadrži zadnje očitane koordinate lokacije svih dostavljača.
- **Zadatak** – Sadrži podatak o vrsti zadatka za svaku narudžbu i svaki proizvod u taj narudžbi te podatak je li zadatak izvršen. Još dodatno sadrži koordinate koje se unese samo ako je vrsta zadatka *IDINANOVULOKACIJU*.



Slika 25: Baza podataka

Za komunikaciju s bazom podataka, koriste se razredi iz paketa **Database**. Razredi iz ovog paketa uspostavljaju vezu s bazom podataka, te upućuju **SQL zahtjeve** i obrađuju **odgovore** na njih.

## 6.2. *Dijagram razreda s opisom*

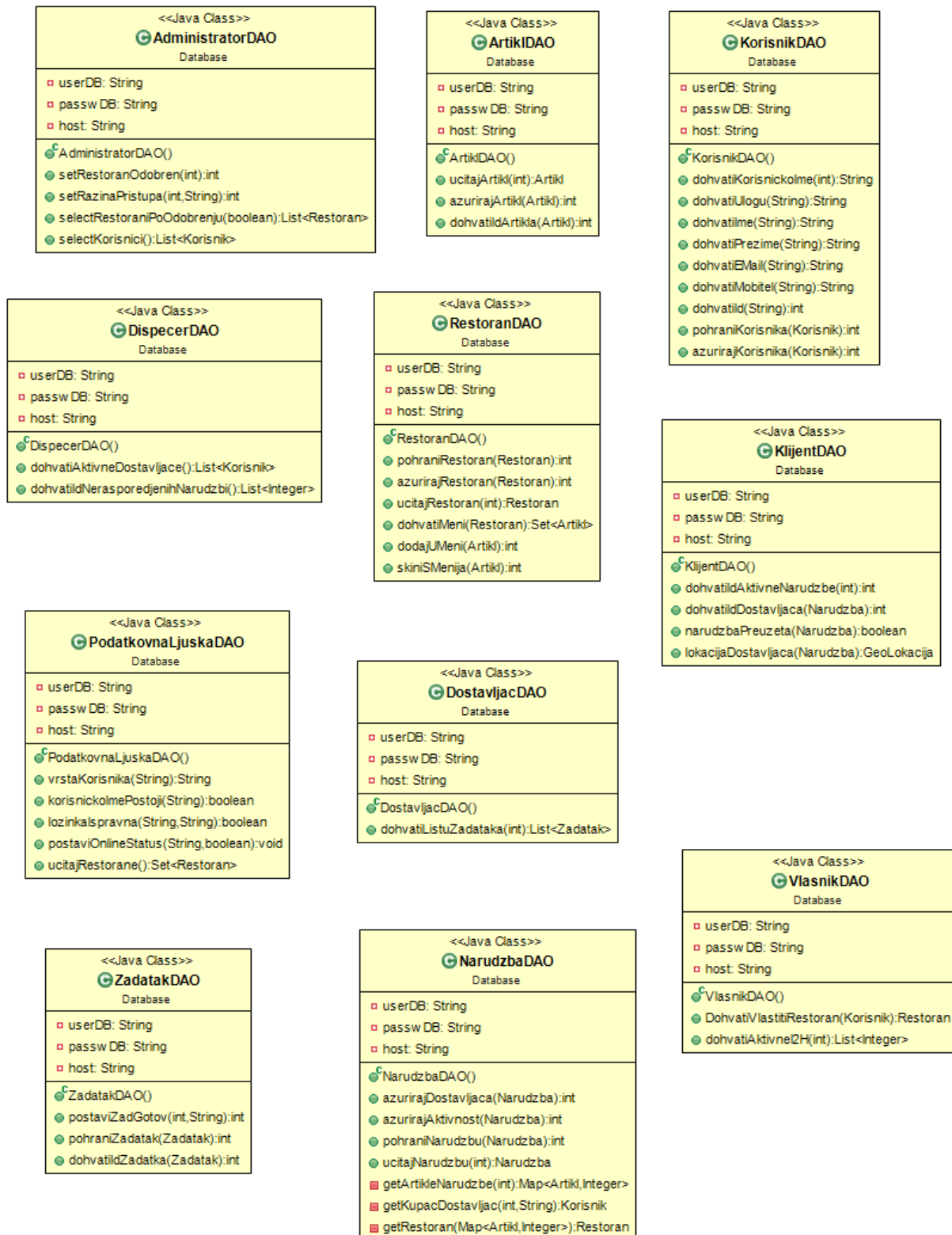
Projektna aplikacija bit će podijeljena u dva velika podsustava – podatkovni podsustav i grafičko sučelje. Zadaća podatkovnog podsustava je obrada i upravljanje informacijama potrebnim za funkcioniranje sustava te komunikacija s bazom podataka. Zadaća grafičkog sučelja je prikaz informacija i interakcija s korisnikom sustava.

Razredi se dijele na tri paketa:

- Entiteti
- Okviri
- Baza Podataka

### ***Baza Podataka***

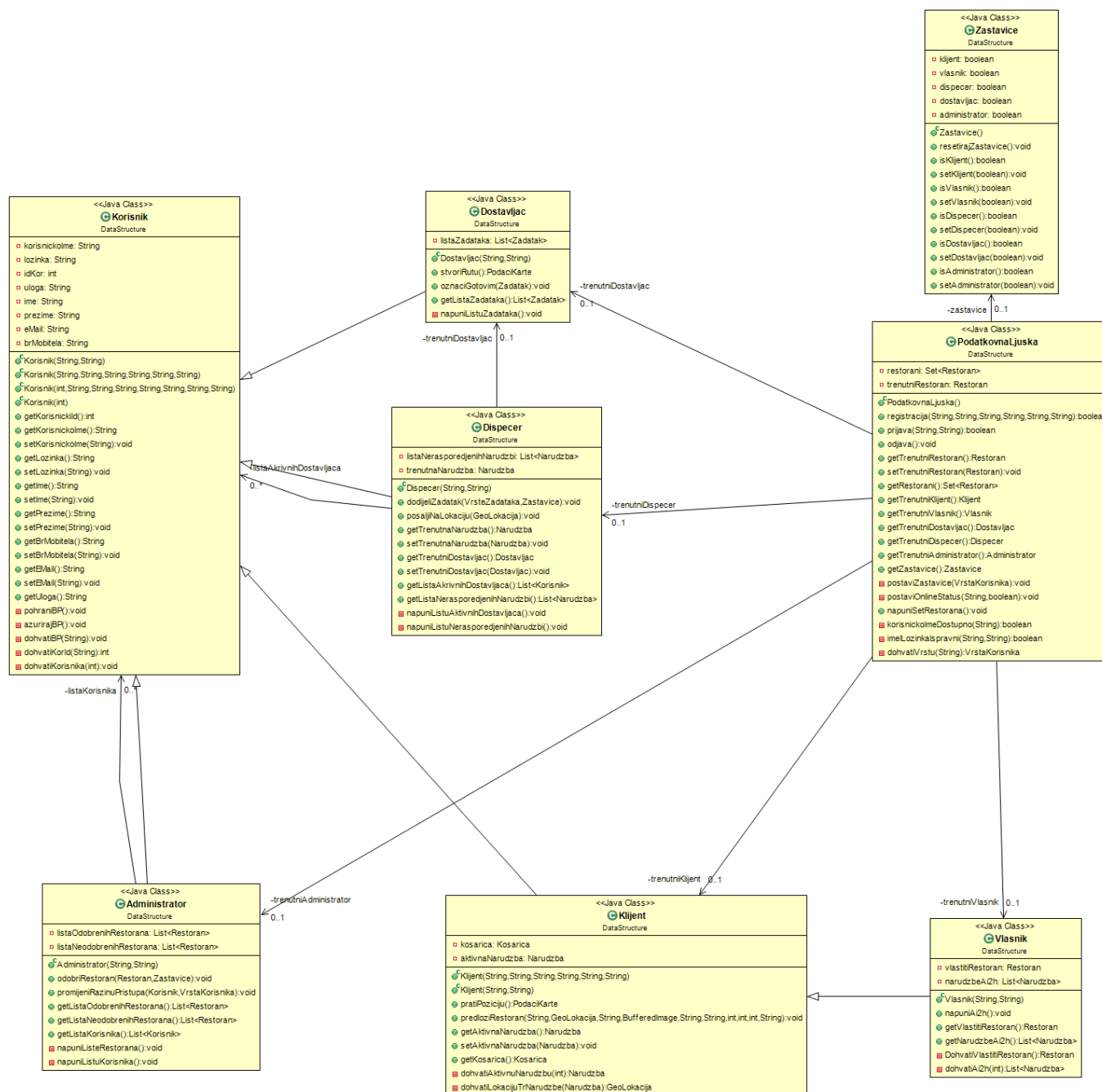
Za komunikaciju s bazom podataka, koriste se razredi iz paketa **Database**. Razredi iz ovog paketa uspostavljaju vezu s bazom podataka, te upućuju **SQL zahtjeve** i obrađuju **odgovore** na njih.



Slika 26: Razredi Baze Podataka

## Entiteti

U ovom paketu nalaze se svi razredi potrebni za izvođenje svih operacija koje sustav nudi te razredi u koje će se spremati lokalni podaci povučeni iz baze podataka. **PodatkovnaLjuska** je glavni razred iz kojeg se instanciraju i kontroliraju svi ostali razredi ovog paketa. **PodatkovnaLjuska** stvara razred **Korisnik** i o njemu neovisne instance razreda potrebne za pregledavanje ponuda. **Korisnik** je generalizacija grupe razreda (**Klijent**, **Vlasnik**, **Dostavljač**, **Dispečer**, **Administrator**) od kojih svaki sadrži dodatne varijable i metode potrebne za izvođenje operacija određene uloge u sustavu. Razredi **Košarica**, **Restoran**, **Artikl**, **Zadatak** i **Narudžba** modeliraju dotične objekte u stvarnom svijetu. **VrsteZadataka** je enumeracija koja označava vrstu zadatka dodijeljenu dostavljaču. **PodaciKarte** je razred koji sadrži sve informacije koje je potrebno prikazati na karti. Razred se instancira i puni svaki put kada je potrebno prikazati kartu. Razred **GeoLokacija** modelira Geo lokaciju. Razred **Zastavice** sadrži zastavice koje omogućavaju ili onemogućavaju izvođenje određenih metoda, ovisno o razini pristupa korisnika.

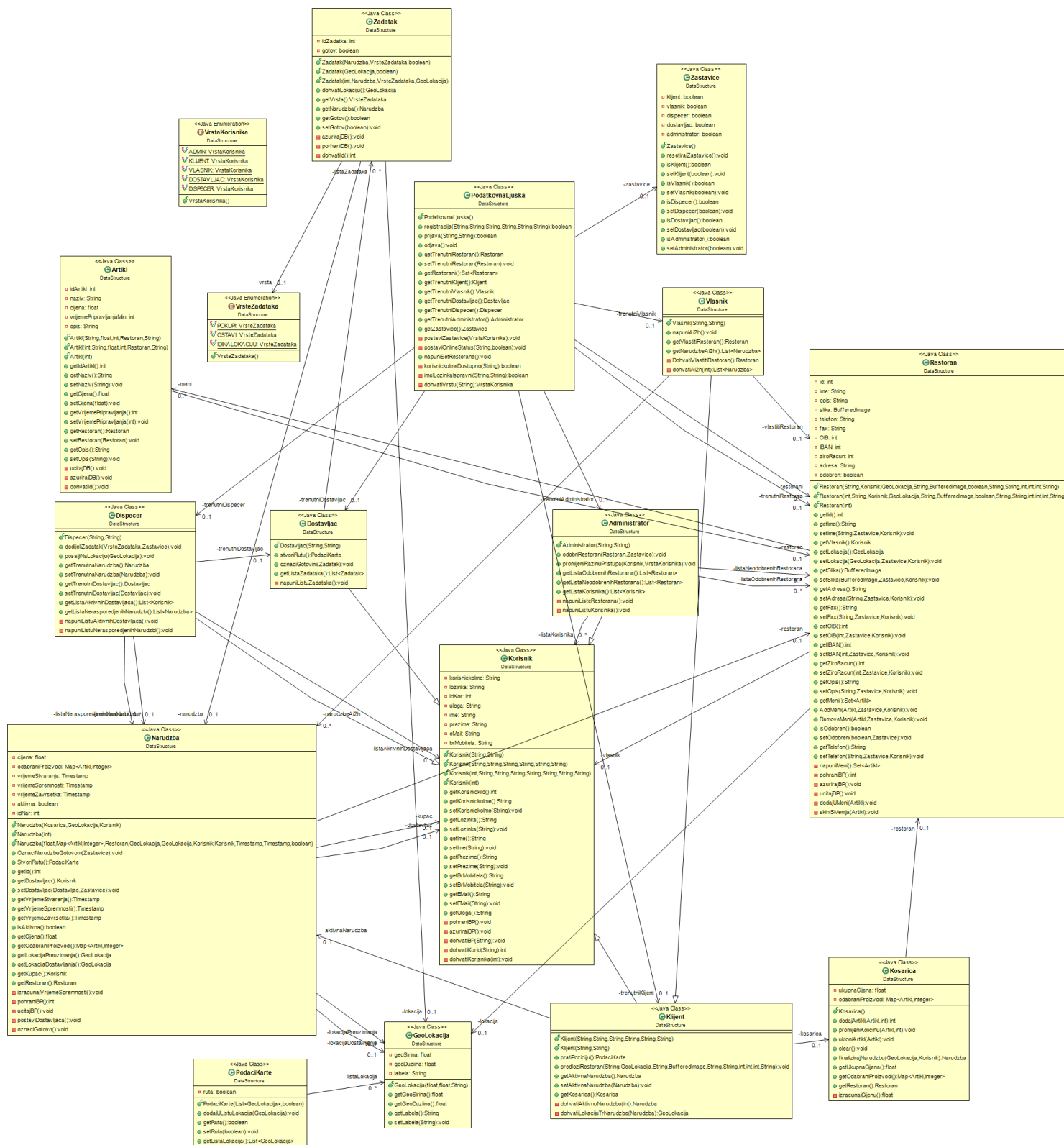


Slika 27: Podatkovni podsustav - prikaz grupe razreda koji omogućavaju funkcionalnost sustava





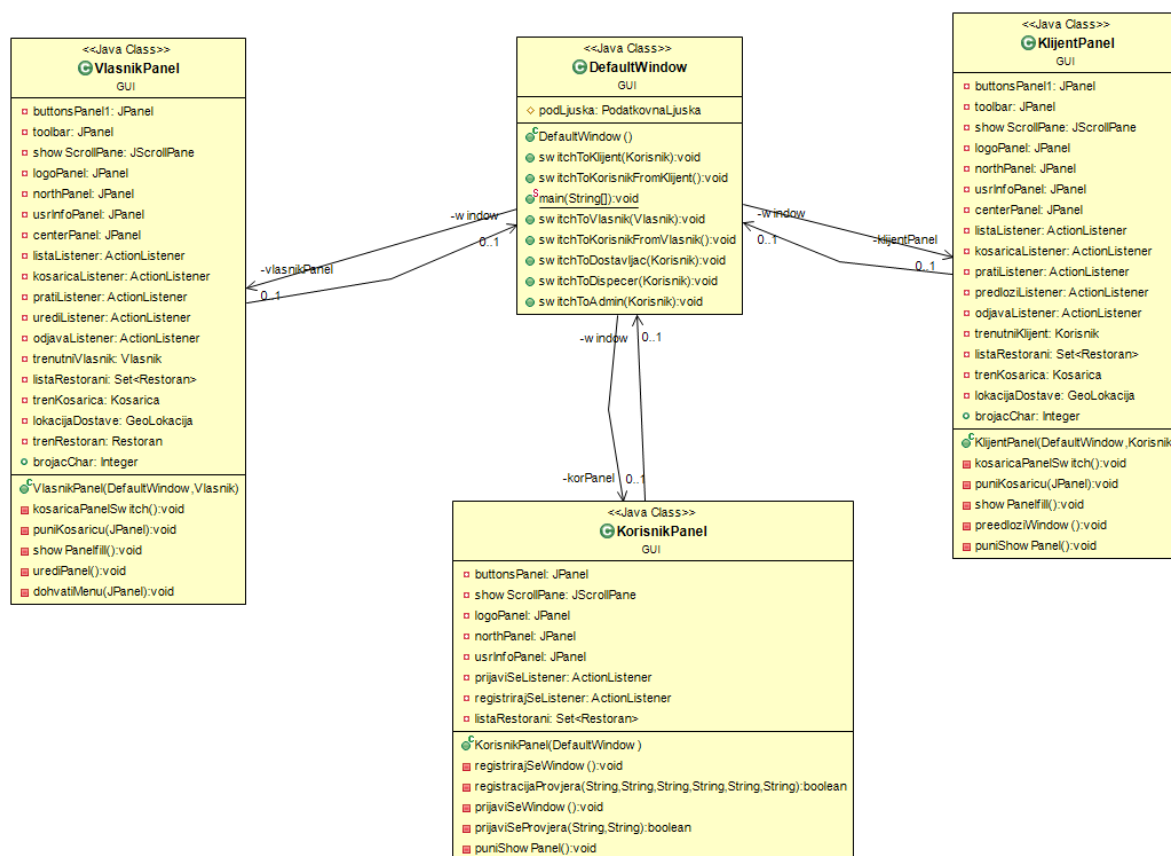
Slika 28: Podatkovni podsustav - prikaz razreda koji modeliraju entitete iz stvarnog svijeta



Slika 29: Prikaz dijagrama razreda podatkovnog podsustava i odnosa među njima

## Okviri

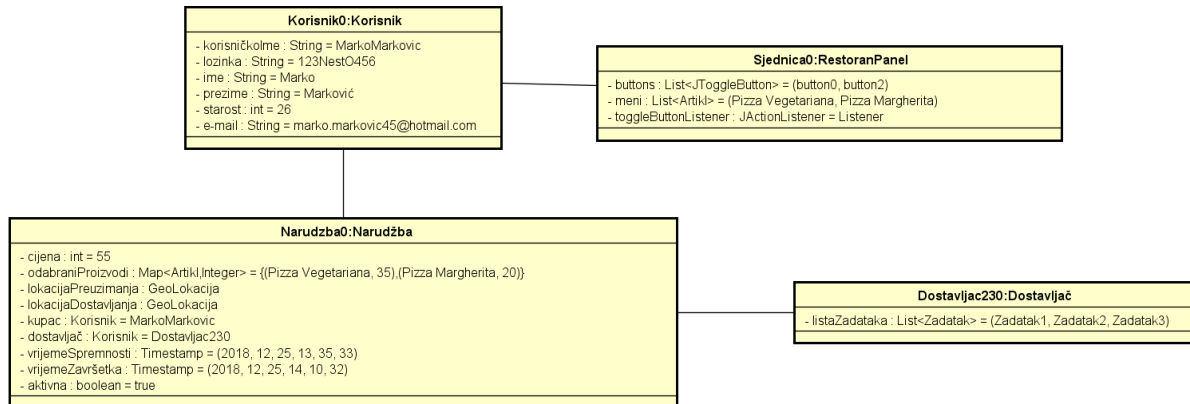
U ovom paketu se nalaze svi razredi potrebni za oblikovanje i prikaz korisničkog sučelja aplikacije. Glavni razred je **DefaultWindow**, namjena tog razreda je kreiranje prozora koji će se „puniti“ različitim panelima, ovisno o vrsti korisnika i o željenoj usluzi. Razredi **KorisnikPanel**, **KlijentPanel** i **VlasnikPanel** tvore jedno te isti panel s jedinom razlikom da pružaju drugačije usluge različitim korisnicima. Razred **PratiNarudzbuPanel** služi za prikazivanje stanja i lokacije aktivne narudžbe korisniku, bio on Klijent ili Vlasnik. Razred **RestoranPanel** generira prikaz informacija restorana i daje mogućnost dodavanja artikla iz menija u košaricu. Razred **UrediMeniPanel** daje istovjetan prikaz razredu RestoranPanel, ali pruža dodatnu mogućnost vlasnika restorana da upravlja sadržajem menija. Razred **TvojRestoran** kreira panel koji daje Vlasniku restorana pregled informacija o vlastitom restoranu i listu aktivnih narudžba restorana. Razred **KošaricaPanel** daje pregled sadržaja košarice prije finalizacije narudžbe, dodatno panel pruža mogućnost povećanja i smanjenja količine artikla koji su već u košaricu te potpuno uklanjanje artikla iz košarice. Razred **DispečerPanel** kreira grafički prikaz koji omogućava Dispečeru da odabire Dostavljača i narudžbu te da zada zadatak dostavljaču u vezi odabrane narudžbe. Razred **DostavljačPanel** mora dati Dostavljaču jasni prikaz njegove rute na karti i listu svih nedovršenih zadataka, te pri završetku zadatka Dostavljač mora moći označiti zadatak kao dovršen. Razred **AdminPanel** tvori poseban grafički prikaz pristupan samo Administratoru, panel sadrži listu Restorana koji traže odobrenje, listu odobrenih Restorana i listu svih korisnika.



Slika 30: Korisničko sučelje - prikaz razreda koji modeliraju korisničko sučelje

### 6.3. Dijagram objekata

Dijagram prikazuje stanje sustava u trenutku neposredno nakon potvrde narudžbe **Narudzba0** koju je postavio korisnik **Korisnik0**.



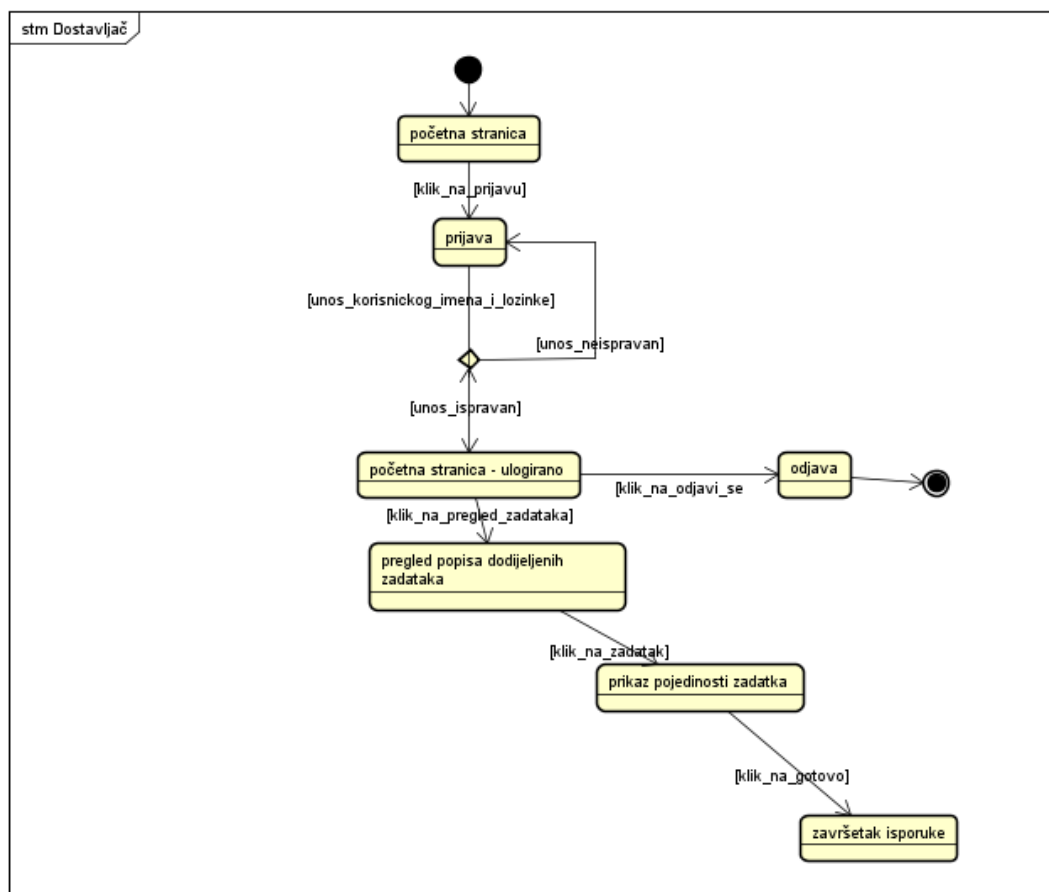
Slika 31: Dijagram objekata

## 6.4. Ostali UML dijagrami

### Dijagram stanja

Dijagram stanja služi za opis diskretnih stanja sustava i prijelaza između tih stanja. Težište mu je na unutarnjem djelovanju dijelova sustava i često prikazuje prijelaze između stanja u sustavu koji su poticani događajima.

Sljedeća slika prikazuje dijagram stanja dostavljača. Dostavljač se na početku prijavljuje u aplikaciju. Za prijavu je potreban unos korisničkog imena i lozinke. U slučaju neispravnog unosa nekog od podataka, javlja mu se odgovarajuća greška i mogućnost ponovne prijave. U suprotnom, dostavljač se uspješno prijavljuje u aplikaciju. Nakon prijave dostavljač postaje aktivan i ima mogućnost pregleda zadataka i odjave. Odabirom pregleda zadataka, može vidjeti detaljne informacije o zadatku. Nakon odrađenog zadatka mora isti označiti gotovim.



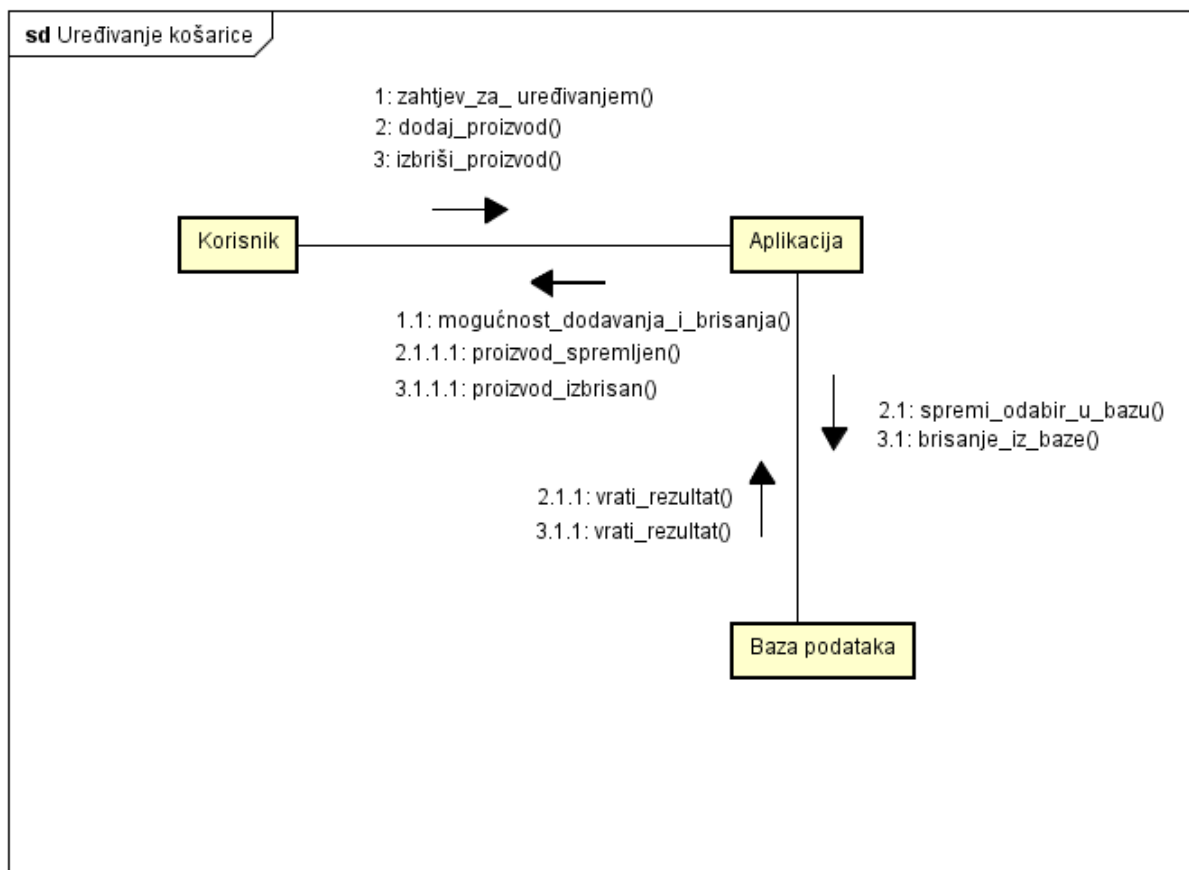
Slika 32: Dijagram stanja – Dostavljač

## Komunikacijski dijagram

Komunikacijski dijagram modelira upravljački tok. Specificira tijek komunikacije između instanci i pripada skupini ponašajnih i dinamičkih UML-dijagrama.

Dijagram na slici prikazuje slijed komunikacije korisnika, aplikacije i baze podataka.

Nakon odabira opcije „uredi košaricu“, korisnik ima mogućnost odabrati jednu od dvije opcije – „dodaj proizvod“ ili „obriši proizvod“. Ukoliko se odluči za dodavanje proizvoda u košaricu, proizvod se sprema u bazu podataka, a aplikacija javlja korisniku uspješnost dodavanja proizvoda. Ako želimo izbrisati proizvod, isti se briše iz baze te aplikacija javlja uspješnost brisanja proizvoda.

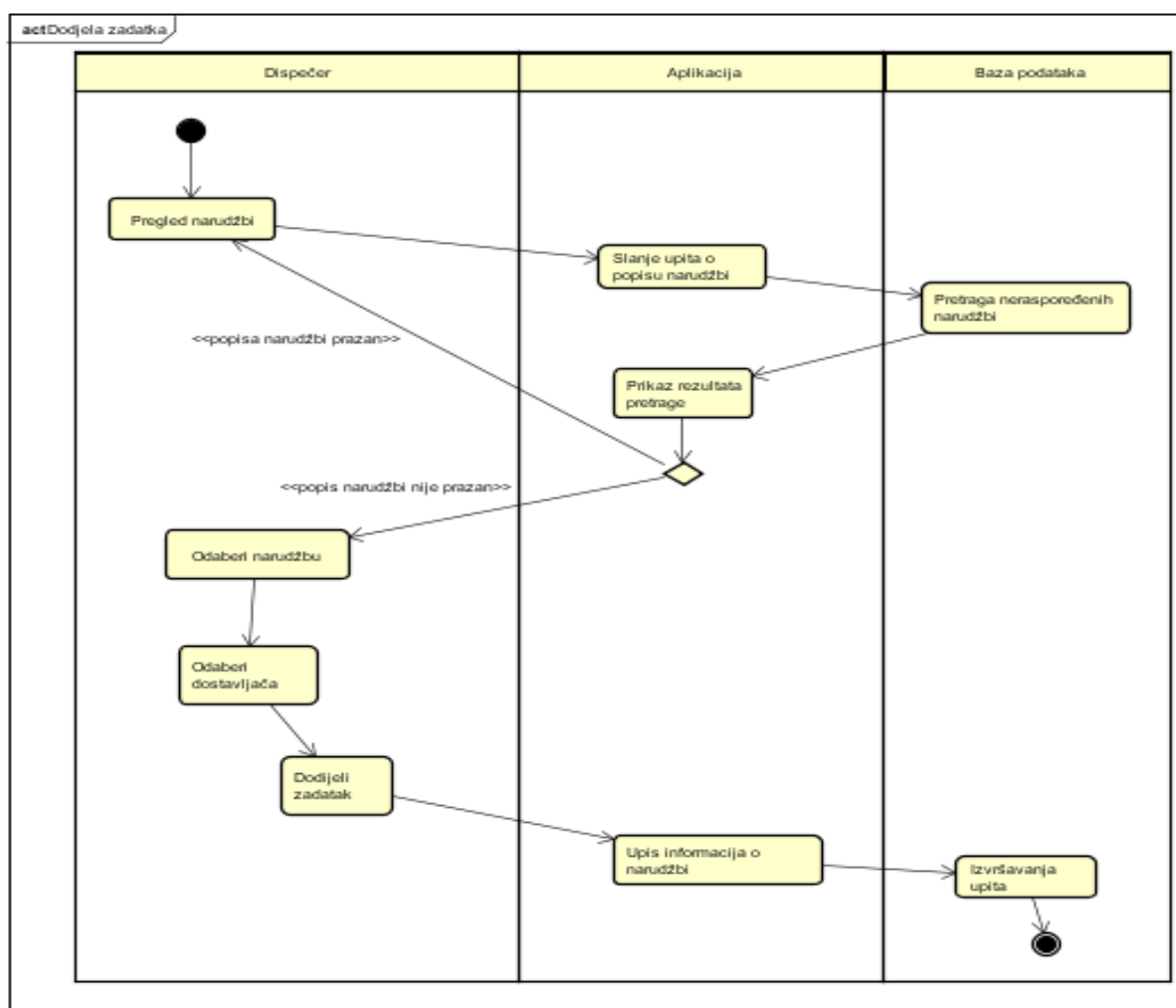


Slika 33: Komunikacijski dijagram – Uređivanje košarice

## Dijagram aktivnosti

Dijagram aktivnosti prikazuje radni tok aktivnosti koje se obavljaju u sustavu korak po korak. Stoga je kod dijagrama aktivnosti naglasak na jednostavnosti i poslovnim operacijama koji se uvijek odvijaju slijedno, jedna za drugom.

Dijagram aktivnosti prikazan na slici se sastoji od tri aktora: dispečer, aplikacija i baza podataka. Nakon odabira opcije pregleda narudžbi, baza vraća korisniku željeni popis. Ako postoje narudžbe, dispečer odabire jednu od njih, dodjeljuje joj dostavljača te bira zadatak za njega. Izabrani zadatak se upisuje u bazu podataka.



Slika 34: Dijagram aktivnosti – Dodjela zadatka

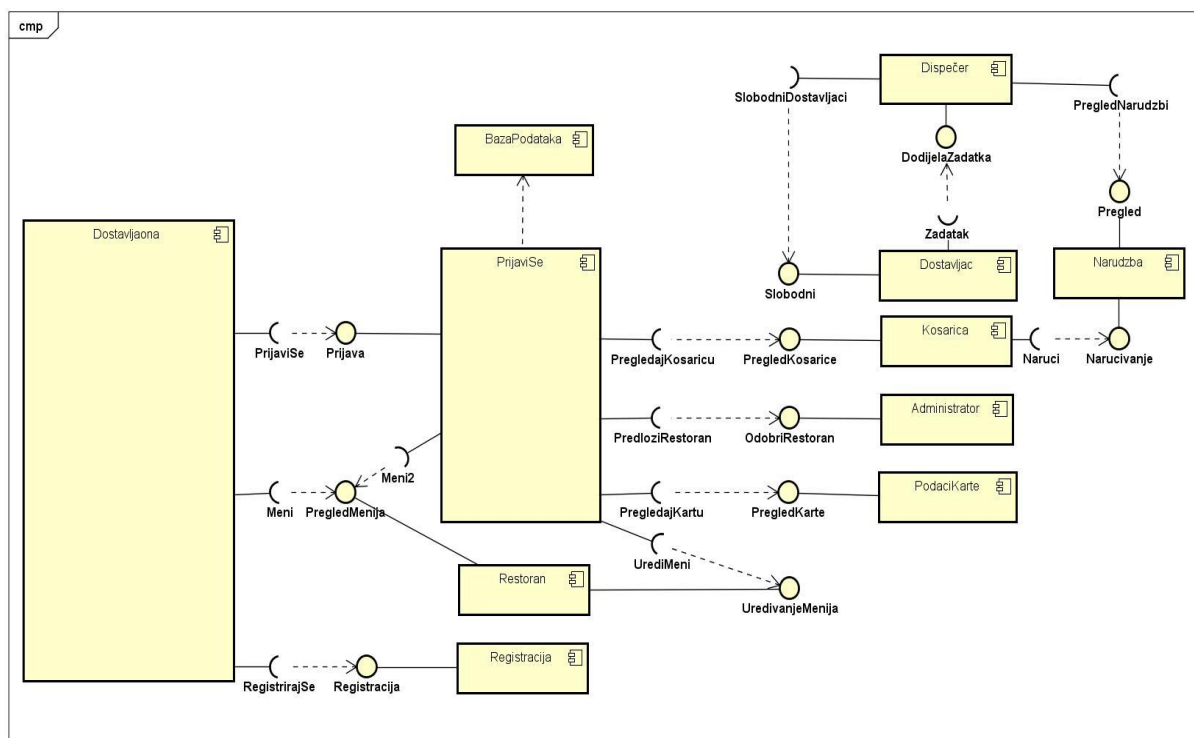


## Dijagram komponenti

Sustav se sastoji od komponenti od kojih je svaka zaslužna za jedan pogled.

Komponente su sljedeće:

- **Dostavljaona** prikazuje korisniku početnu stranicu
- **PrijaviSe** omogućuje korisniku prijavu u aplikaciju
- **Registracija** omogućuje korisniku registraciju u aplikaciju
- **BazaPodataka** omogućuje spremanje podataka
- **Restoran** omogućuje pregledavanje menija
- **PodaciKarte** prikazuju kartu korisniku
- **Administrator** obrađuje prijedloge restorana
- **Kosarica** prikazuje korisniku odabrane artikle
- **Narudzba** omogućuje korisniku naručivanje
- **Dostavljač** odrađuje dobijene zadatke
- **Dispečer** dodjeljuje zadatke dostavljačima



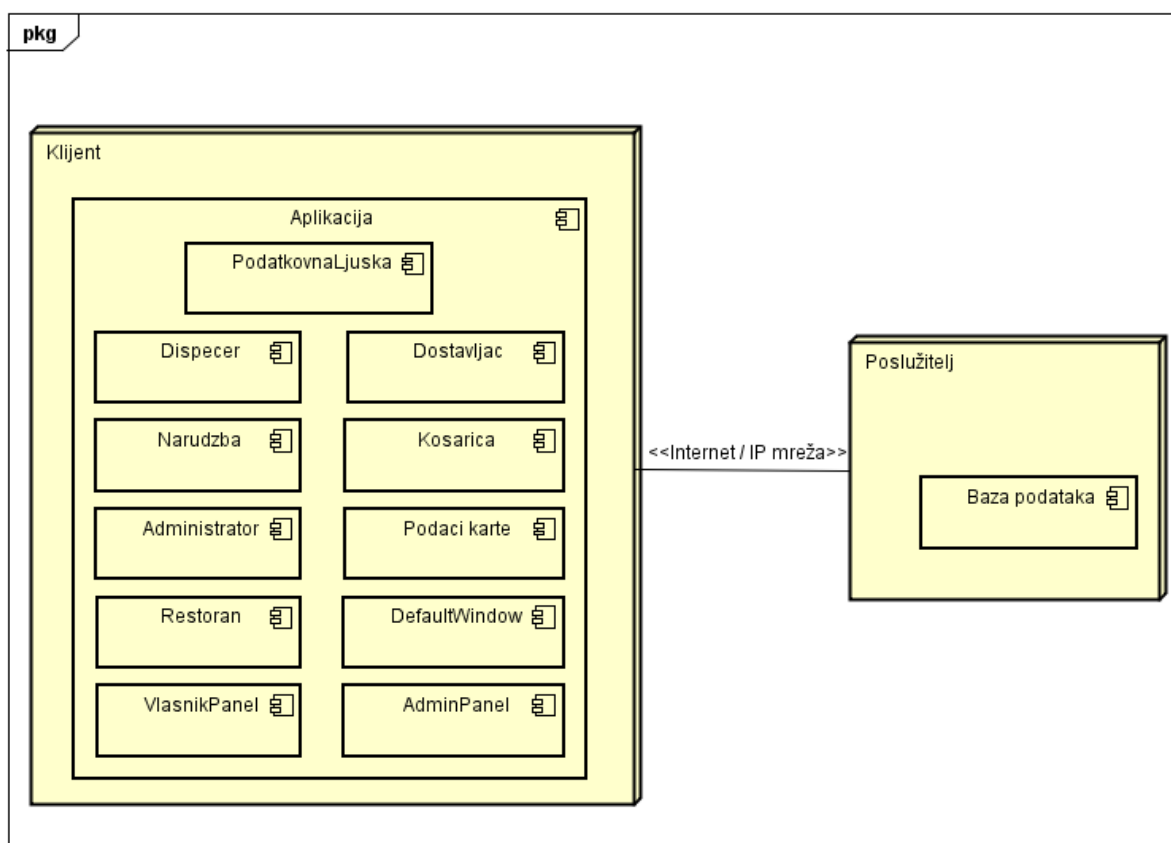
Slika 35: Dijagram komponenti

## 7. Implementacija i korisničko sučelje

### Dijagram razmještaja

Dijagrami razmještaja opisuju topologiju sklopovlja i programsku potporu koja se koristi u implementaciji sustava u njegovom radnom i produkcijskom okruženju.

Aplikacija je zamišljena kao desktop aplikacija, a sustav se temelji na komunikaciji „klijent – poslužitelj“. Na klijentskom računalu se pokreće aplikacija, a veza s bazom podataka, koja se nalazi na poslužitelju, ostvarena je Internet/IP mrežom.



Slika 36: Dijagram razmještaja

## ***Korištene tehnologije i alati***

### **GitLab**

Ovaj sustav služi kao repozitorij datoteka. Omogućuje praćenje izmjena i čuva cijelu povijest (kako su se verzije mijenjale, tko je kriv za neki bug...). Također omogućava timski rad na projektu, zajedničko pisanje po istom codebase-u, ujedinjavanje usporednih promjena te služi kao središnji poslužitelj odakle svi dobivaju istu verziju.

### **Astah UML**

Služi za modeliranje sustava i razvoj programske potpore korištenjem UML-a. Prednosti uporabe ovog alata su pohrana sadržaja na jednom mjestu. Podržava normu UML 2.0+ (2005.) i to sljedeće vrste dijagrama: strukturni i ponašajni. Jedan je od brojnih CASE alata.

### **Eclipse**

Eclipse je programski okvir otvorenog koda, baziran je na programskom jeziku Java i neovisan je o operacijskom sustavu. Pruža proširivu razvojnu platformu dizajniranu za razvoj integriranih razvojnih okruženja IDE, raznih alata, aplikacijskih okruženja, ali i samostalnih aplikacija.

Eclipse platforma dolazi s dodacima koji olakšavaju Java razvoj, koji se nalaze u paketu Java Development Tools (JDT).

Pružaju osnovni skup servisa koji kontroliraju dodatke kako bi oni međusobno mogli funkcionirati kao cjelina.

## **MySQL**

MySQL je jedan od popularnijih sustav otvorenog koda za upravljanje bazama podataka. Proizvod je tvrtke MySQL AB iz Švedske.

MySQL:

- Sustav za upravljanje bazama podataka
- Sustav za upravljanje relacijskim bazama podataka
- Open source software
- Brz, pouzdan i jednostavan za korištenje

## **Java**

Java je moderan programski jezik koji se koristi u praksi i jedan je od najpopularnijih jezika na svijetu. Baziran je na klasama, odnosno objektima pa ga možemo svrstati u objektno-orijentirane programske jezike. Velika prednost u odnosu na većinu dosadašnjih programskih jezika je to što se programi pisani u Javi mogu izvoditi bez preinaka na svim operacijskim sustavima za koje postoji JVM (Java Virtual Machine).

## Isječak programskog koda vezan za temeljnu funkcionalnost sustava

```
44 public boolean prijava (String korisnickoIme, String lozinka) {
45
46     boolean valjan = false;
47     VrstaKorisnika vrsta = null;
48
49     valjan = this.imeILOzinkaIspravn(korisnickoIme, lozinka);
50
51
52     vrsta = dohvatiVrstu(korisnickoIme);
53
54     if (valjan) {
55
56         if (vrsta == VrstaKorisnika.KLIJENT) {
57             this.trenutniKorisnik = new Klijent(korisnickoIme, lozinka);
58         }
59         else if (vrsta == VrstaKorisnika.VLASNIK) {
60             this.trenutniKorisnik = new Vlasnik(korisnickoIme, lozinka);
61         }
62         else if (vrsta == VrstaKorisnika.DISPECER) {
63             this.trenutniKorisnik = new Dispecer(korisnickoIme, lozinka);
64         }
65         else if (vrsta == VrstaKorisnika.DOSTAVLJAC) {
66             this.trenutniKorisnik = new Dostavljac(korisnickoIme, lozinka);
67         }
68         else if (vrsta == VrstaKorisnika.ADMIN) {
69             this.trenutniKorisnik = new Administrator(korisnickoIme, lozinka);
70         }
71
72         this.postaviZastavice(vrsta);
73         this.postaviOnlineStatus(trenutniKorisnik.getKorisnickoIme(), true);
74
75         return true;
76     }
77     else {
78
79         return false;
80     }
81 }
82
83
84 public void odjava () {
85
86     this.postaviOnlineStatus(trenutniKorisnik.getKorisnickoIme(), false);
87     this.zastavice.resetirajZastavice();
88     this.trenutniKorisnik = null;
89 }
```

Slika 37: Isječak programskog koda – prijava i odjava

Prikazane su metode prijava i odjava koje korisnik može odabrati. Prvo se provjeravaju uneseni podatci te ako su točni korisnik se prijavljuje u sustav i prikazuje mu se odgovarajući prozor ovisno o njegovoj razini pristupa.

```
27 public int setRestoranOdobren(int idRestoran) {
28
29     String sql = "UPDATE restoran SET restoranOdobren = ? WHERE idRestoran = ?";
30     int result = 2;
31
32     try(Connection con = DriverManager.getConnection(host, userDB, passwDB);
33         PreparedStatement prepSt = con.prepareStatement(sql)) {
34
35         prepSt.setBoolean(1, true);
36         prepSt.setInt(2, idRestoran);
37         result = prepSt.executeUpdate();
38
39     } catch (SQLException sqlExc) {
40
41         System.out.println(sqlExc.getMessage());
42     }
43
44     return result;
45 }
46
47 public int setRazinaPristupa(int idKor, String novaRazinaPristupa) {
48
49     String sql = "UPDATE korisnik SET uloga = ? WHERE idKor = ?";
50     int result = 2;
51
52     try(Connection con = DriverManager.getConnection(host, userDB, passwDB);
53         PreparedStatement prepSt = con.prepareStatement(sql)) {
54
55         prepSt.setString(1, novaRazinaPristupa);
56         prepSt.setInt(2, idKor);
57         result = prepSt.executeUpdate();
58
59     } catch (SQLException sqlExc) {
60
61         System.out.println(sqlExc.getMessage());
62     }
63
64     return result;
65 }
```

Slika 38: Isječak programskog koda – *setRestoranOdobren* i *setRazinaPristupa*

Prikazane su metode *setRestoranOdobren* i *setRazinaPristupa* kojima administrator odobrava predloženi restoran i mijenja razinu pristupa korisnicima te se sprema u bazu podataka.

```
26 public List<Korisnik> dohvatiAktivneDostavljace () {
27
28     List<Korisnik> aktivniDostavljaci = new ArrayList<Korisnik>();
29     String sql = "SELECT korisnik.* FROM korisnik WHERE uloga = 'DOSTAVLJAC' AND online = true";
30
31     try(Connection con = DriverManager.getConnection(host, userDB, passwDB);
32         PreparedStatement prepSt = con.prepareStatement(sql)) {
33
34         ResultSet rs = prepSt.executeQuery();
35
36         while(rs.next()) {
37
38             int idKorisnika = rs.getInt(1);
39             String korisnickoIme = rs.getString(2);
40             String lozinka = null;
41             String ime = rs.getString(4);
42             String prezime = rs.getString(5);
43             String brMobitela = rs.getString(6);
44             String email = rs.getString(7);
45             String uloga = rs.getString(8);
46
47             Korisnik trenDostavljac = new Korisnik(idKorisnika, korisnickoIme, lozinka, ime, prezime, brMobitela, email, uloga);
48             aktivniDostavljaci.add(trenDostavljac);
49         }
50     } catch (SQLException sqlExc) {
51
52         System.out.println(sqlExc.getMessage());
53     }
54
55     return aktivniDostavljaci;
56 }
57
58
59 public List<Integer> dohvatiIdNerasporedjenihNarudzbi () {
60
61     List<Integer> idNerasporedjenihNarudzbi = new ArrayList<Integer>();
62     String sql = "SELECT UNIQUE idNar FROM narudzba WHERE idDostavljac = null";
63
64     try(Connection con = DriverManager.getConnection(host, userDB, passwDB);
65         PreparedStatement prepSt = con.prepareStatement(sql)) {
66
67         ResultSet rs = prepSt.executeQuery();
68
69         while(rs.next()) {
70
71             int idNar = rs.getInt(1);
72             idNerasporedjenihNarudzbi.add(idNar);
73         }
74     } catch (SQLException sqlExc) {
75
76         System.out.println(sqlExc.getMessage());
77     }
78
79     return idNerasporedjenihNarudzbi;
80 }
81
82 }
```

Slika 39: Isječak programskog koda – dohvatiAktivneDostavljace i dohvatiIdNerasporedjenihNarudzbi

Prikazane su metode `dohvatiAktivneDostavljace` i `dohvatiIdNerasporedjenihNarudzbi`. Prva metoda vraća listu aktivnih dostavljača koji su potrebni dispečeru kako bi im dodijelio zadatke, druga metoda dohvaća neraspoređene narudžbe iz baze podataka.

```

64 JButton PrijaviSe = new JButton("   Prijavi se   ");
65 PrijaviSe.setMaximumSize(new Dimension(120, 120));
66 PrijaviSe.addActionListener(prijaviSeListener);
67 ImageIcon image1 = new ImageIcon(getClass().getResource("/images/PrijaviSeLogoMini.png"));
68 PrijaviSe.setIcon(image1);
69
70
71 JButton RegistrirajSe = new JButton("Registriraj se");
72 RegistrirajSe.addActionListener(registrirajSeListener);
73 RegistrirajSe.setMaximumSize(new Dimension (120, 120));
74 ImageIcon image2 = new ImageIcon(getClass().getResource("/images/RegistrirajSELogoMini.png"));
75 RegistrirajSe.setIcon(image2);
76
77
78 buttonsPanel = new JPanel();
79 buttonsPanel.setLayout(new BoxLayout(buttonsPanel, BoxLayout.Y_AXIS));
80 buttonsPanel.add(PrijaviSe);
81 buttonsPanel.add(RegistrirajSe);
82 buttonsPanel.setBorder(BorderFactory.createLineBorder(new Color(0, 153, 255), 2));
83 buttonsPanel.setBackground(Color.white);
84 buttonsPanel.setPreferredSize(new Dimension(120, 500));
85 add(buttonsPanel, BorderLayout.EAST);
86
87
88 northPanel = new JPanel();
89 northPanel.setLayout(new BorderLayout());
90 add(northPanel, BorderLayout.NORTH);
91
92
93 logoPanel = new JPanel();
94 logoPanel.setBackground(Color.white);
95 logoPanel.setBorder(BorderFactory.createLineBorder(new Color(0, 153, 255), 3));
96 logoPanel.setPreferredSize(new Dimension(1080, 75));
97 ImageIcon imageLogo = new ImageIcon(getClass().getResource("/images/GlavniLogo.png"));
98 logoPanel.add(new JLabel(imageLogo));
99 northPanel.add(logoPanel, BorderLayout.CENTER);
100
101
102 usrInfoPanel = new JPanel();
103 usrInfoPanel.setLayout(new BorderLayout());
104 JTextArea usrName = new JTextArea();
105 usrName.setForeground(new Color(0, 153, 255));
106 usrName.setEditable(false);
107 usrName.setFont(new Font("Arial", 0, 14));
108 usrName.setText("\n Prijavljen kao:\n Anonimni korisnik");
109 usrInfoPanel.add(usrName, BorderLayout.CENTER);
110 usrInfoPanel.setPreferredSize(new Dimension(120, 75));
111 usrInfoPanel.setBackground(Color.white);
112 usrInfoPanel.setBorder(BorderFactory.createLineBorder(new Color(0, 153, 255), 2));
113 northPanel.add(usrInfoPanel, BorderLayout.EAST);

```

Slika 40: Isječak programskog koda – Panel

Prikazano je stvaranje panela koji će korisnik vidjeti kada se prijavi i gumbi koje će korisnik moći pritisnuti.



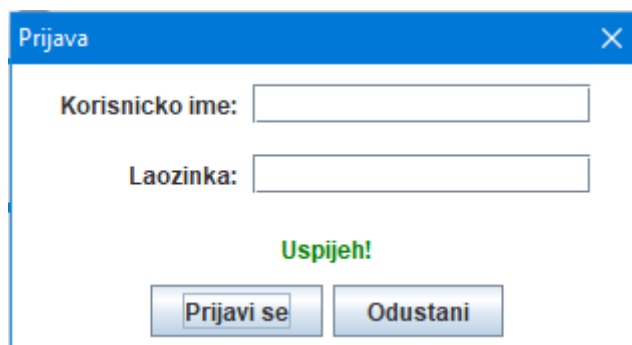
## Ispitivanje programskog rješenja

### Slučaj 1

**Opis:** Registrirani korisnik se prijavljuje u sustav.

**Očekivani rezultat:** Ispravnim unosom podataka, korisnik je prijavljen u sustav.

**Dobiveni rezultat:** Ukoliko je korisnik unio ispravne podatke, ispisuje mu se poruka o uspješnoj prijavi. U slučaju krivog upisa podataka, ispisuje mu se poruka o neuspješnoj prijavi.



The screenshot shows a window titled "Prijava" (Login) with a blue header bar. Inside the window, there are two text input fields: "Korisnicko ime:" (Username) and "Lozinka:" (Password). Below the fields, the text "Uspjeh!" (Success!) is displayed in green. At the bottom, there are two buttons: "Prijavi se" (Login) and "Odustani" (Cancel).

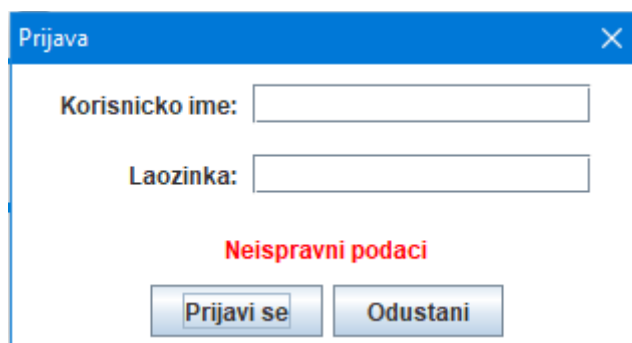
Slika 41: Ispitivanje programskog rješenja – Prijava

### Slučaj 2

**Opis:** Neregistrirani korisnik se pokušava prijaviti.

**Očekivani rezultat:** S obzirom da korisnik nije u bazi podataka javlja mu se poruka o neuspješnoj prijavi

**Dobiveni rezultat:** Korisniku se ispisuje odgovarajuća poruka i ne dopušta prijava u sustav.



The screenshot shows the same "Prijava" (Login) window. The input fields for "Korisnicko ime:" and "Lozinka:" are empty. Below the fields, the text "Neispravni podaci" (Invalid data) is displayed in red. The buttons "Prijavi se" and "Odustani" are still present at the bottom.

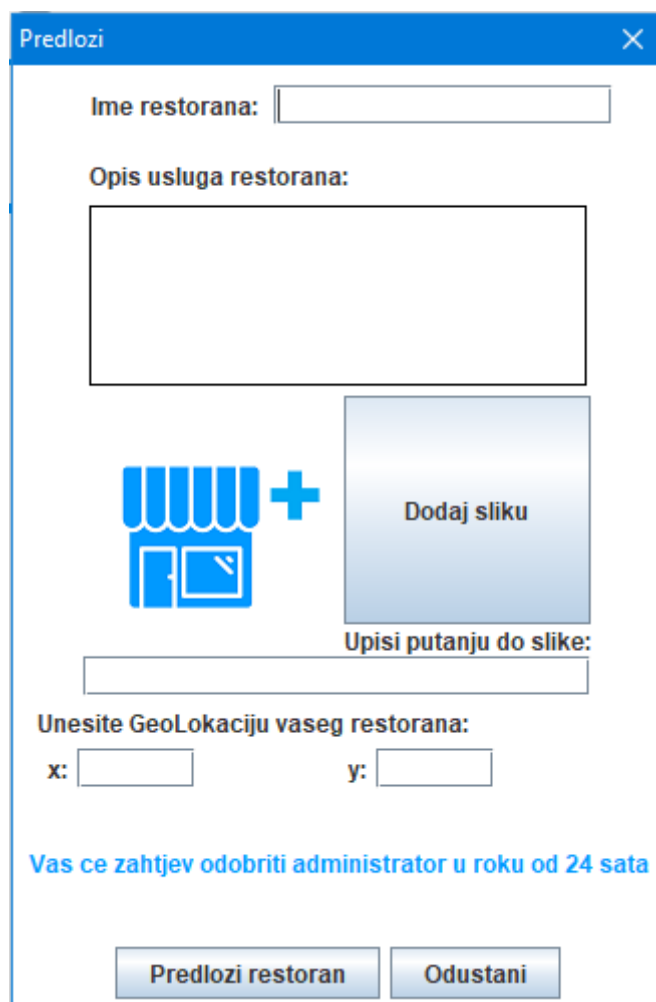
Slika 42: Ispitivanje programskog rješenja - Prijava

### Slučaj 3

**Opis:** Korisnik predlaže restoran.

**Očekivani rezultat:** Unosom svih potrebnih podataka, prijedlog se šalje administratoru na odobravanje.

**Dobiveni rezultat:** Ako korisnik unese sve potrebne podatke, prijedlog se šalje administratoru. U suprotnom, ispisuje se poruka o nepotpunom unosu podataka.



The image shows a Windows-style dialog box titled "Predlozi" (Proposals) with a close button (X) in the top right corner. The dialog contains the following elements:

- A text input field labeled "Ime restorana:" (Restaurant name:).
- A larger text area labeled "Opis usluga restorana:" (Description of restaurant services:).
- A blue icon representing a restaurant (a building with a flag) followed by a plus sign (+).
- A button labeled "Dodaj sliku" (Add picture).
- A text input field labeled "Upisi putanju do slike:" (Enter the path to the picture:).
- Two text input fields labeled "Unesite GeoLokaciju vasesg restorana:" (Enter the GeoLocation of your restaurant:), with sub-labels "x:" and "y:".
- A blue text message: "Vas ce zahtjev odobriti administrator u roku od 24 sata" (Your request will be approved by the administrator within 24 hours).
- Two buttons at the bottom: "Predlozi restoran" (Propose restaurant) and "Odustani" (Cancel).

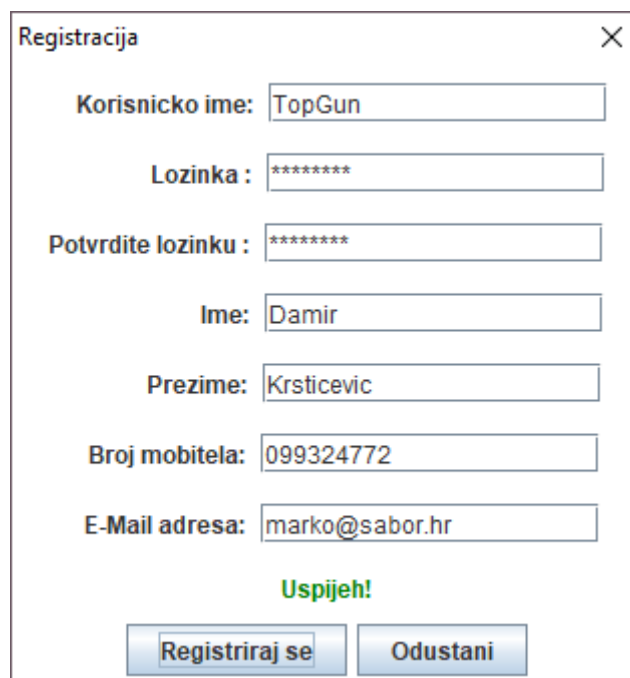
Slika 43: Ispitivanje programskog rješenja – Prijedlog restorana

## Slučaj 4

**Opis:** Registracija korisnika.

**Očekivani rezultat:** Nakon klika na gumb za registraciju, otvara mu se panel u kojem unosi podatke potrebne za registraciju.

**Dobiveni rezultat:** U slučaju unosa svih podataka javlja mu se poruka o uspješnosti registracije.



The screenshot shows a registration window titled "Registracija" with a close button (X) in the top right corner. The window contains several input fields for user registration:

- Korisnicko ime: TopGun
- Lozinka : \*\*\*\*\*
- Potvrdite lozinku : \*\*\*\*\*
- Ime: Damir
- Prezime: Krsticevic
- Broj mobitela: 099324772
- E-Mail adresa: marko@sabor.hr

Below the input fields, the text "Uspijeh!" is displayed in green, indicating a successful registration. At the bottom, there are two buttons: "Registriraj se" and "Odustani".

Slika 44: Ispitivanje programskog rješenja – Registracija

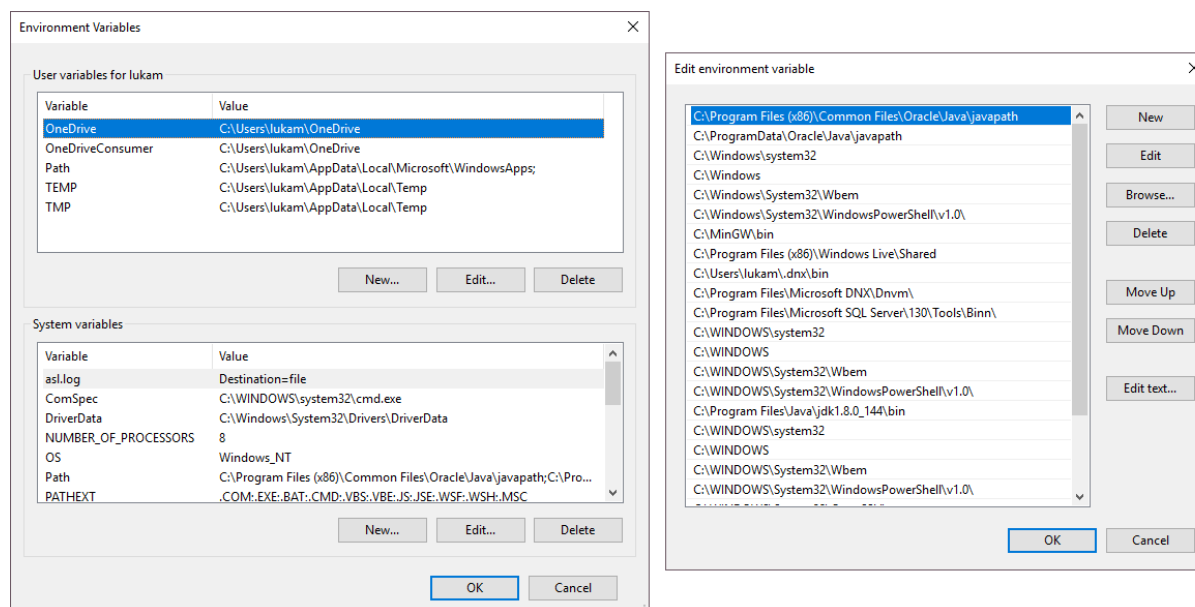
## Upute za instalaciju

### 1. Java SE Development Kit 8 – Windows Editino

Java Development Kit potreban je za pokretanje prototipne verzije aplikacije razvijene u projektu. JDK potrebno preuzeti sa službenih stranica Oracle-a. To možete napraviti na ovom linku:

<https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>. Preporučamo korištenje JDK verzije 8 zato što je projekt razvijen u toj verziji JDK. Ovisno o verziji Windows-a instaliranoj na vašem računalu odaberite 32-bitnu ili 64-bitnu verziju.

Po završetku instalacije potrebno je postaviti varijable okruženja PATH, to se može napraviti na sljedeći način : Control Panel > System > Advanced System Settings > Advanced > Environment variables. Unutar izbornika Environment variables, potrebno je urediti sistemsku varijablu Path. U nju potrebno je dodati apsolutnu destinaciju direktorija unutar kojega je instaliran JDK (npr.: C:\Program Files (x86)\Common Files\Oracle\Java\javapath). To se postiže klikom na varijablu path, pa klikom na gumb Edit. Kada ste ušli u izbornik za uređivanje varijable Path, pritisnite gumb New i dodajte gore navedenu apsolutnu destinaciju. Po završetku, izađite iz izbornika.



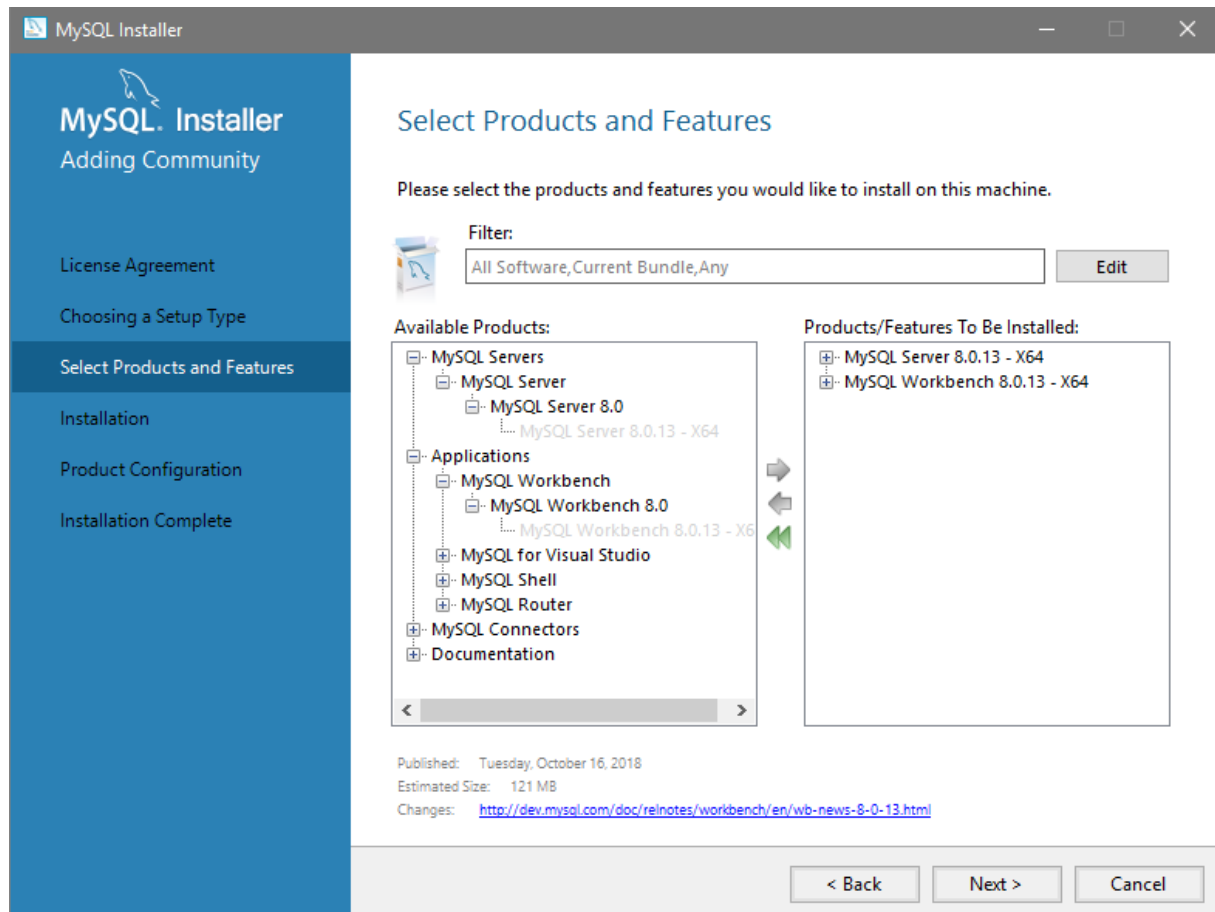
Slika 45: Izbornik Environment Variables i izbornik Edit environment variable

### 2. MySQL Installer 8.0.13

MySQL Installer je potreban kako bi se instalirao MySQL server baze podataka te MySQL Workbench koji služi za lakše i razumljivije pregledavanje i definiranje početne baze podataka. MySQL Installer se skine s sljedećeg linka:

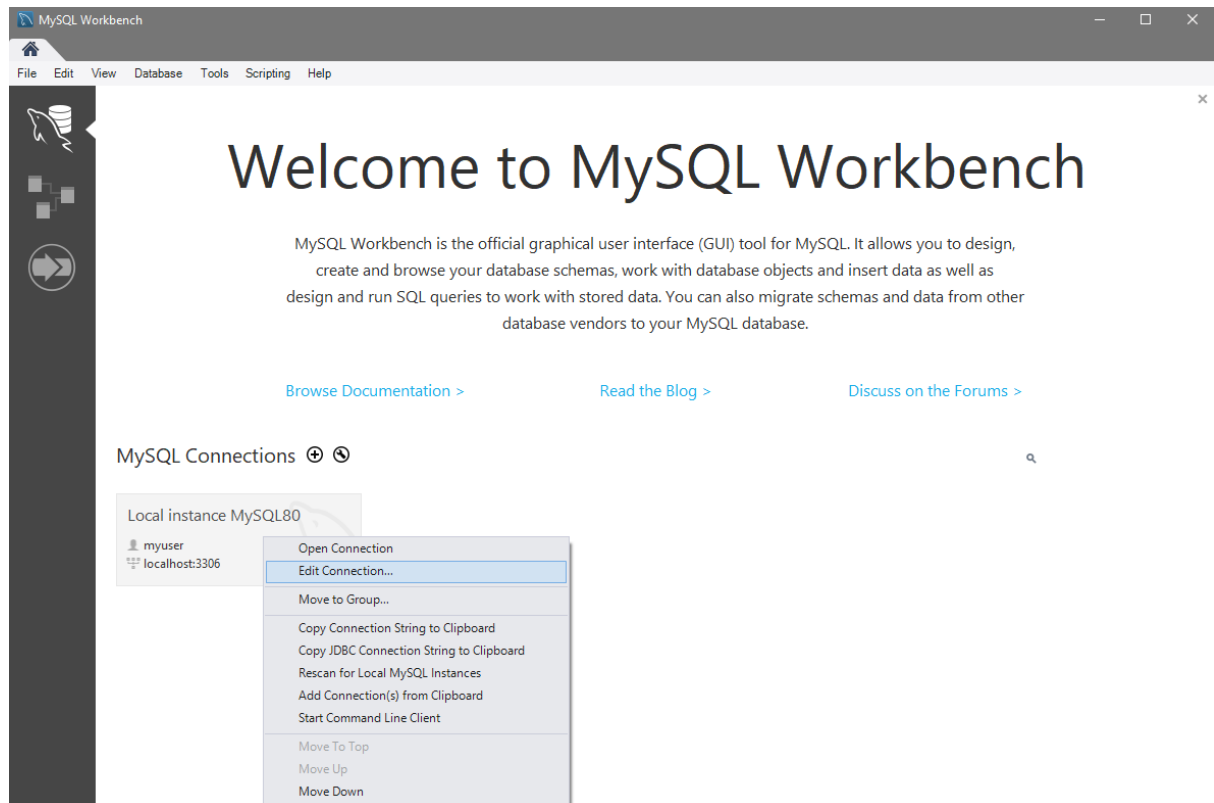
<https://dev.mysql.com/downloads/file/?id=480824>. Pri dnu je opcija “No thanks, just start my download” koju treba stisnuti.

Dvoklikom na datoteku pokrećemo instalaciju i stisnemo “next” dok ne dođemo do “Setup type” u kojem odaberemo “Custom” opciju. Nakon klika za dalje, odaberu se “MySQL Server 8.0.13 -X64” i “MySQL Workbench 8.0.13 -X64” i klikne na zelenu strelicu usmjerenu prema desno.



Slika 46: Upute za instalaciju - MySQL

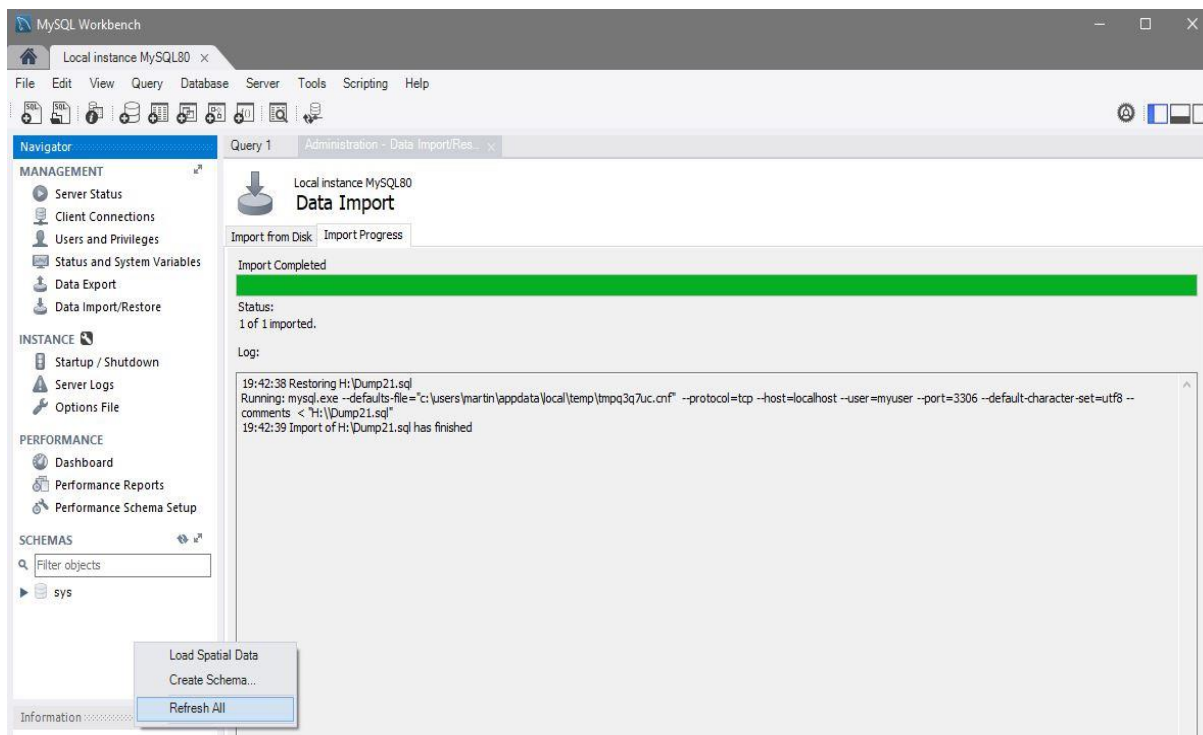
Potom se klikne dalje te execute. Nakon što se instalacija obavi, odabere se opcija “Standalone MySQL Server/Classic MySQL Replication”. U izborniku “Type and Networking” se ostavljaju pretpostavljene vrijednosti, a u izboru autentikacije se odabere prva opcija. Kada se dođe do prozora u kojem se traži root password, upiše se “abcd” te se klikne na gumb “add user” i pod *user name* se stavi “myuser” a pod *password* se stavi „abcd“ i klikne na OK. U *Windows Service* dijelu se samo klikne za dalje te se onda klikne na execute. Nakon izvršenja operacije klikne se na “finish” te se izađe iz instalacijskog programa. Zatim treba pokrenuti *Workbench* i na stranici dobrodošlice se može vidjeti prepoznata veza uspostavljenog servera imena *Local Instance MySQL80*. Desnim klikom miša na konekciju odabere se “Edit connection”.



Slika 47: Upute za instalaciju - MySQL

U novom prozoru upiše se proizvoljno ime veze i pod *username* se stavi "myuser". Nakon toga klikom na "Test connection" program nas traži lozinku koja glasi „abcd“ kako je navedeno u instalaciji. Kada program vrati uspješan rezultat, zatvori se prozor te se lijevim klikom stisne na vezu kako bi se uspostavila.

Nakon što uspostavimo vezu prema bazi podataka, moramo je zatim učitati za rad. To se radi odabirom "Data import/Restore" opcije u *Management* izborniku s lijeve strane. Zatim se odabere opcija "Import from Self-Contained File" te se upiše put do *Dump21.sql* koja se nalazi u mapi zajedno s instalacijom aplikacije te se klikne na "Start import".



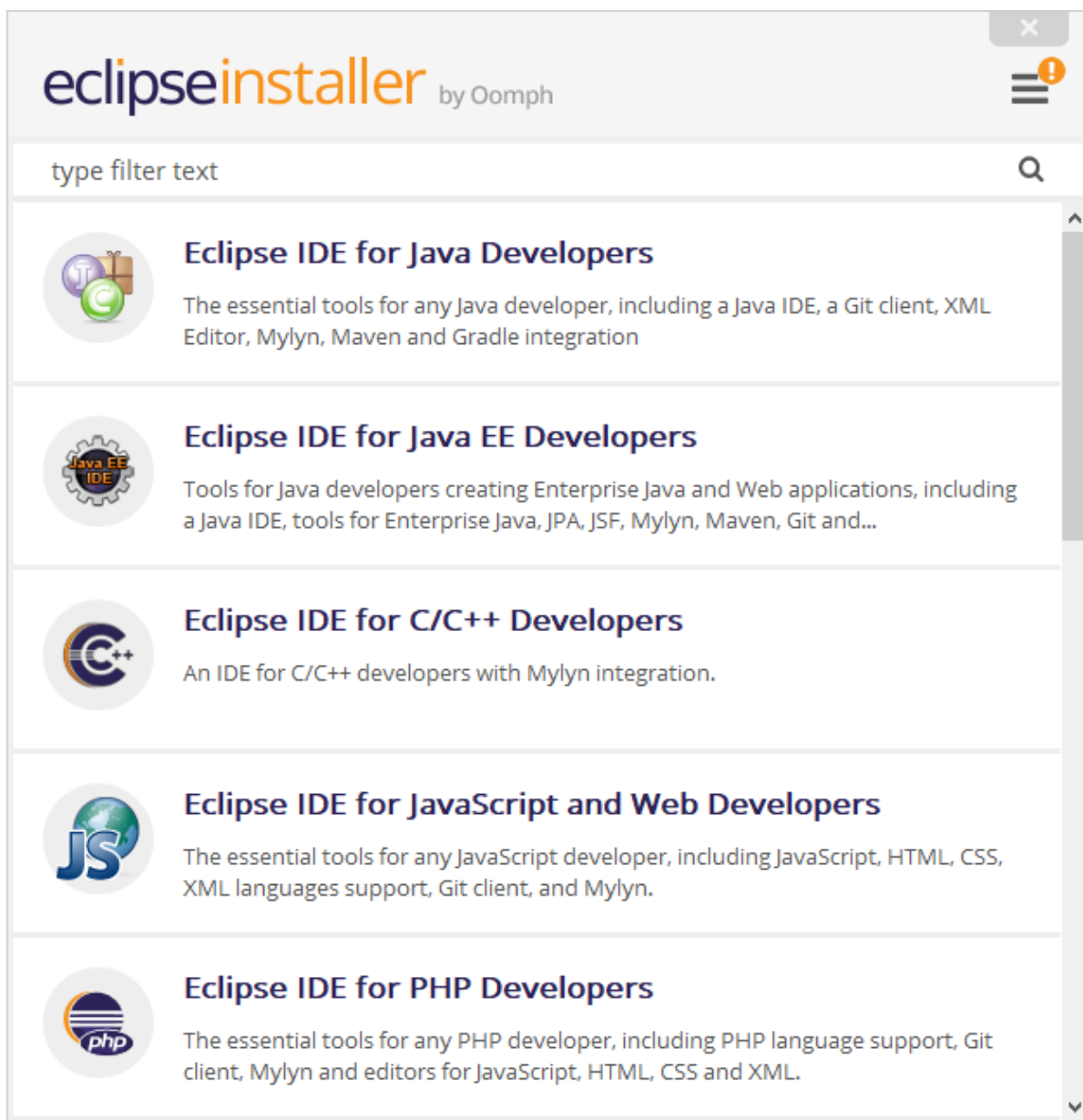
Slika 48: Upute za instalaciju - MySQL

Nakon toga se lijevim klikom u prazni prozor izbornika *Schemas* s lijeve strane klikne na “refresh all” te se odabere novostvorena shema “dostavljaona” i desnim klikom na nju odabere se “set as default schema”. Na kraju se još samo u “Query 1” upiše sljedeći SQL upit: “SET GLOBAL time\_zone = ‘+1:00’;” te se po želji može ugasiti Workbench jer on nije potreban za daljnji rad aplikacije.

### 3. Eclipse IDE 2018-12

Instalacija Eclipse-a je potrebna za pokretanje same aplikacije. Eclipse IDE 2018-12 možete skinuti na ovom linku:

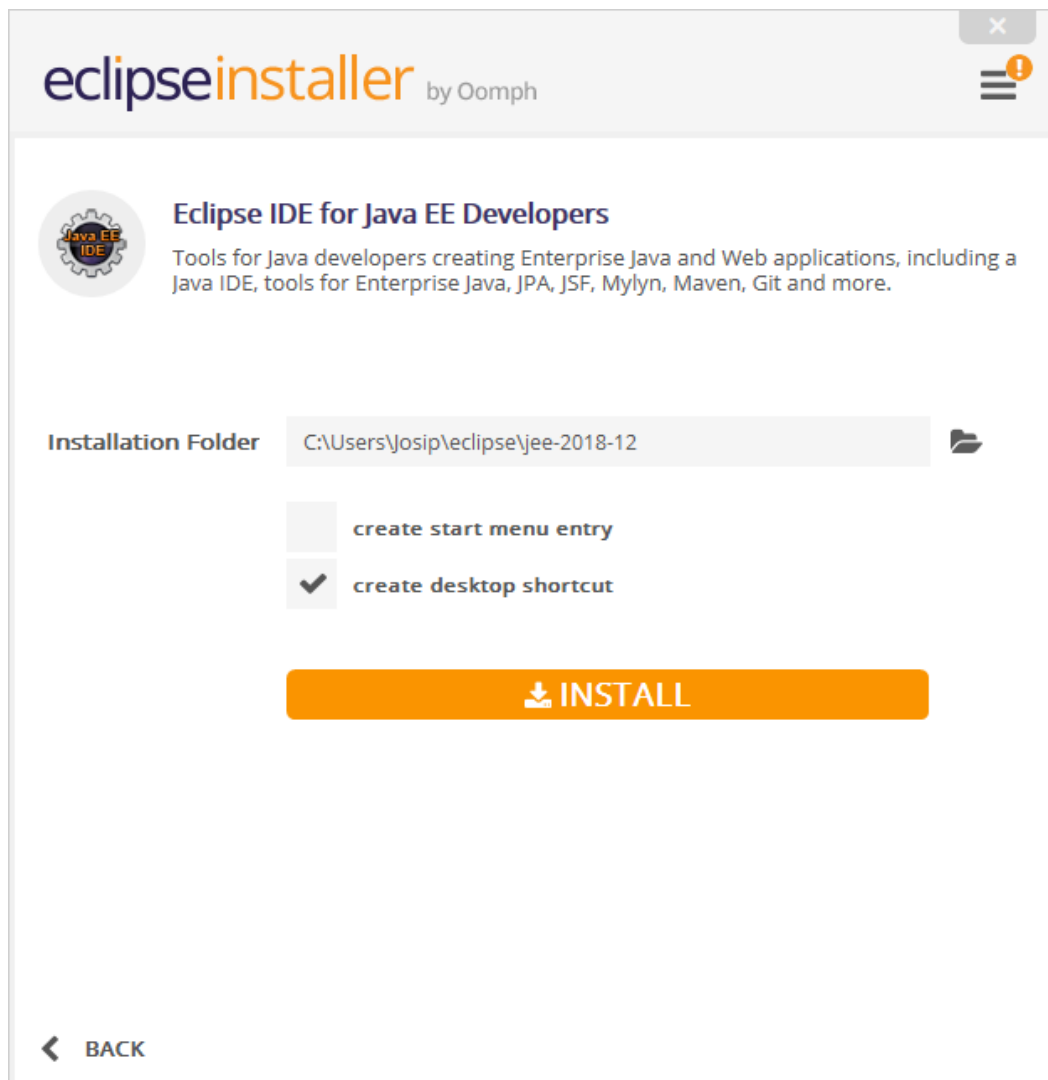
<https://www.eclipse.org/downloads/download.php?file=/oomph/epp/2018-12/R/eclipse-inst-win64.exe>. Dvoklikom na ikonu pokrećemo instalaciju, zatim se otvara prozor u kojem treba stisnuti na „Eclipse IDE for Java EE Developers“.



Slika 49: Upute za instalaciju - Eclipse

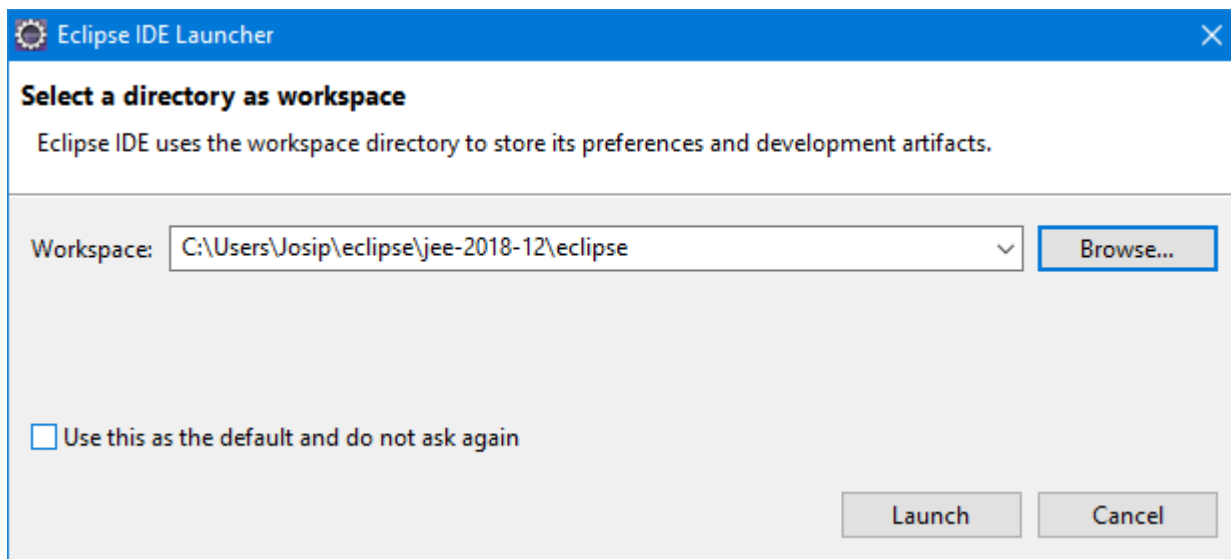


Nakon odabira se otvara prozor u kojem treba upisati destinaciju na koju želimo instalirati Eclipse (npr: C:\Users) i potom kliknuti na gumb "INSTALL".



Slika 50: Upute za instalaciju - Eclipse

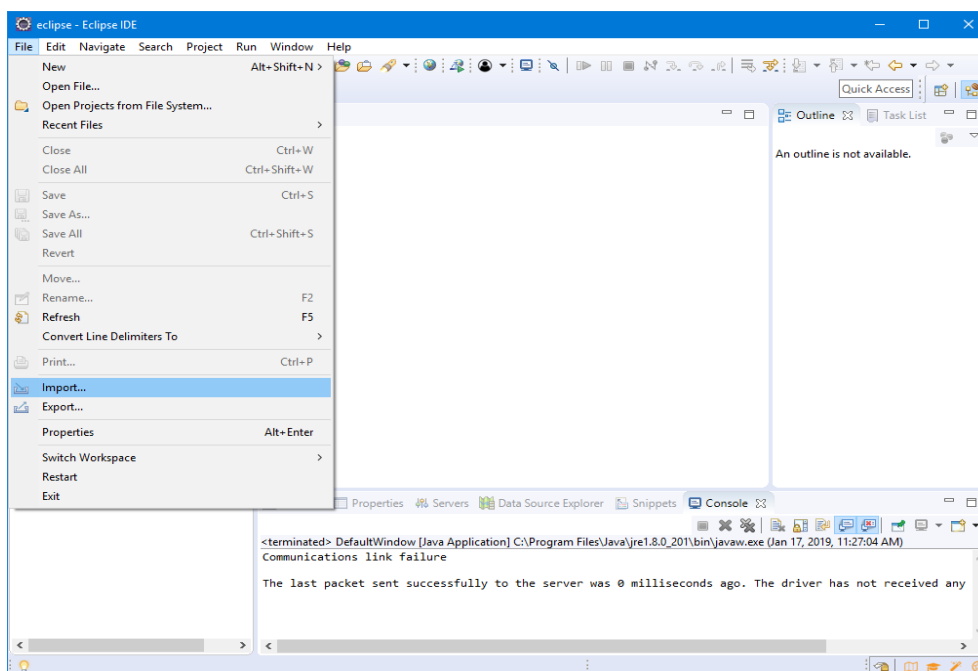
Tokom instalacije se pojavljuje prozor u kojem treba stisnuti „Accept“. Nakon toga se otvara prozor „Certificates“ u kojem treba označiti „Eclipse Foundation“ i kliknuti na „Accept selected“. Po završetku instalacije kliknuti na „Launch“ i time se pokreće Eclipse. Zatim treba odabrati direktorij u koji će se spremati razvojni artefakti i kliknuti na „Launch“.



Slika 51: Upute za instalaciju - Eclipse

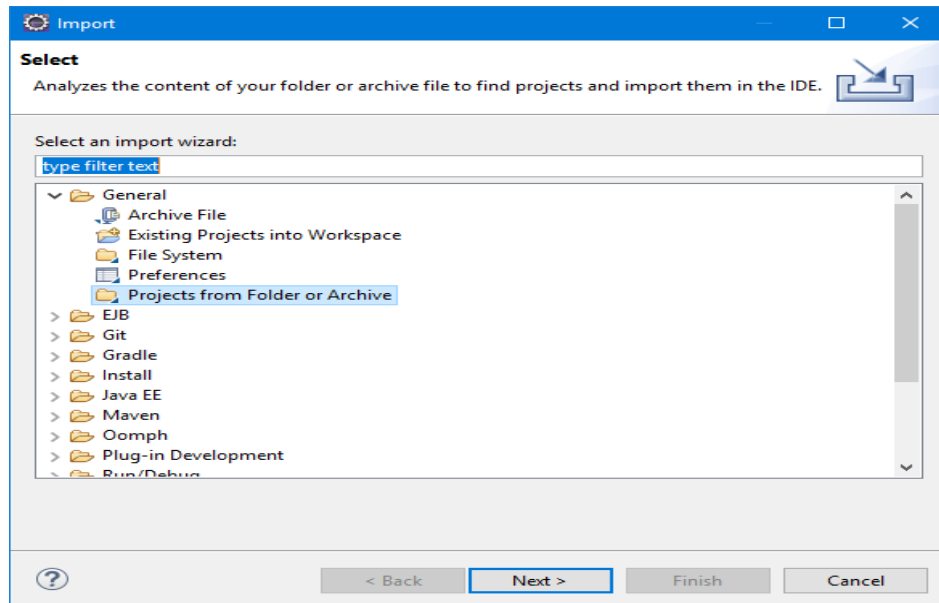
Nakon što se Eclipse pokrenuo, još je potrebno umetnuti projekt u Eclipse. To se radi na ovaj način:

1. Klik u gornjem lijevom kutu na File
2. U padajućem izborniku odabrati Import...



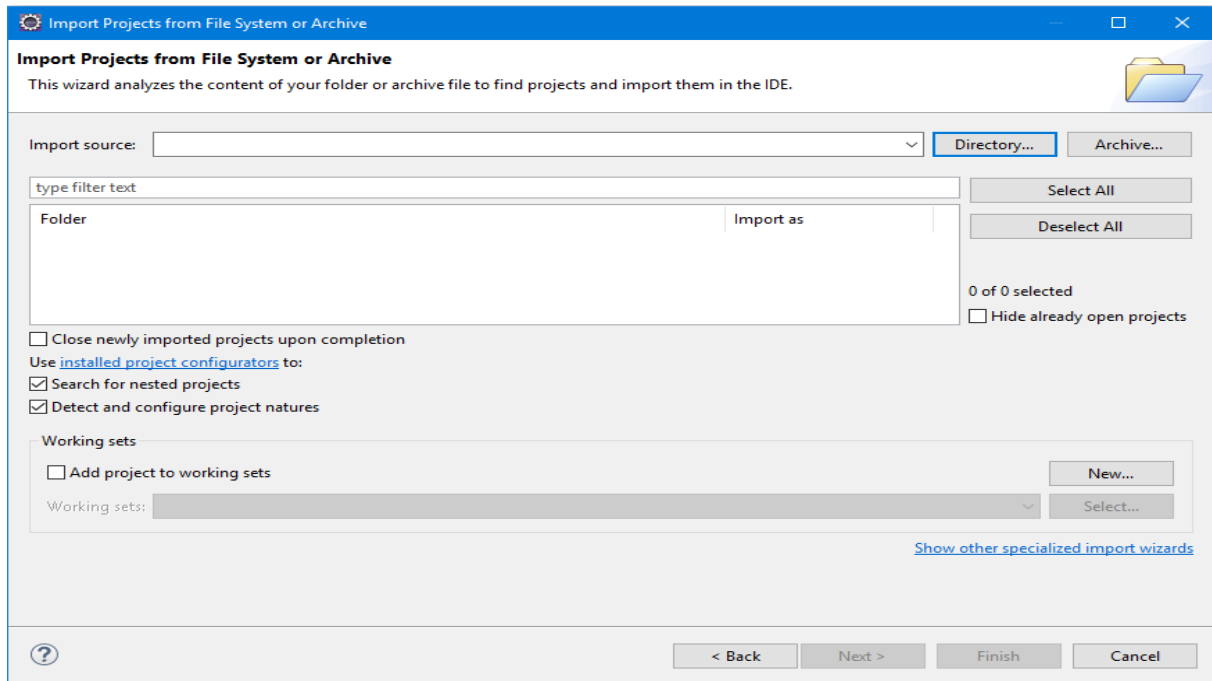
Slika 52: Upute za instalaciju - Eclipse

### 3. Odabrati General zatim Projects from Folder or Archive i kliknuti Next



Slika 53: Upute za instalaciju - Eclipse

### 4. Odabrati Directory... i pronaći direktorij BitSoftTechnologies i kliknuti na Select Folder



Slika 54: Upute za instalaciju - Eclipse

## 5. Klik na Finish za umetanje projekta.

Nakon što je projekt umetnut potrebno ga je pokrenuti. Pritiskom desne tipke miša na folder BitSoftTechnologies i odabirom Run as > Java Application otvara se prozor u kojem treba odabrati DefaultWindow – GUI i klik na OK i aplikacija je pokrenuta.

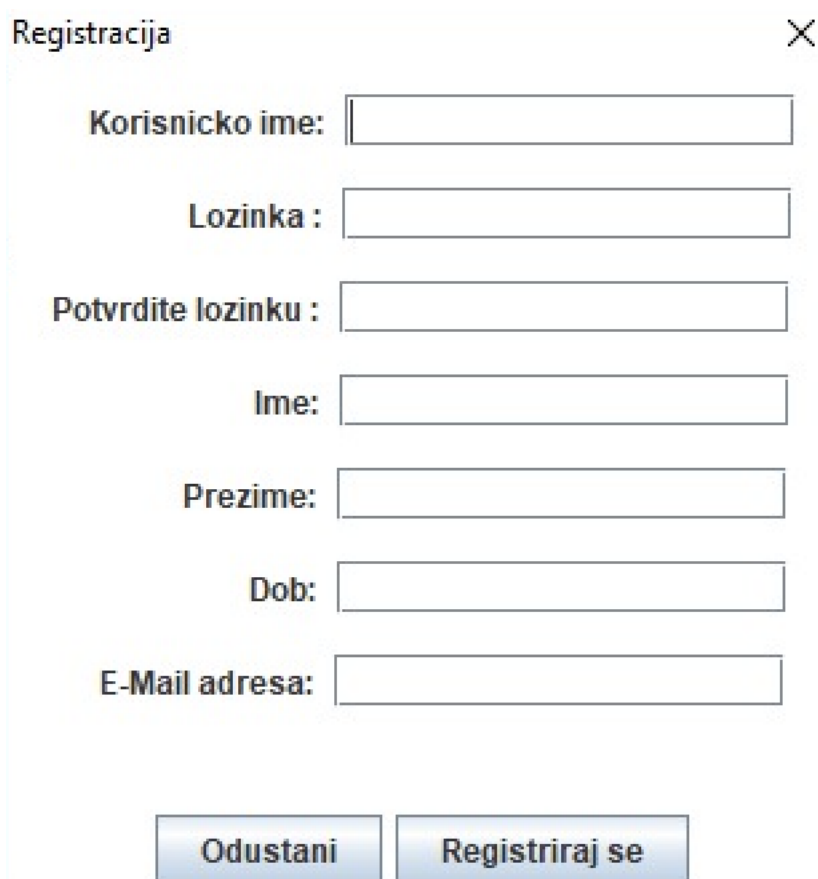
## Korisničke upute

Dolaskom na početnu stranicu aplikacije korisnik će imati sljedeće mogućnosti:

- 1) Pregledavati ponude restorana
- 2) Registrirati se u sustav
- 3) Prijaviti se u sustav

Ukoliko se korisnik ne želi prijaviti u sustav, stranicu može pregledavati kao anonimni korisnik uz određene mogućnosti, što znači da će samo moći vidjeti popis restorana i njihovu ponudu.

Da bi koristio aplikaciju i prijavio se, korisnik se treba registrirati. Odabirom opcije registriraj se otvori mu se prozor koji sadrži polja koje korisnik mora ispuniti svojim podacima.



The image shows a registration window titled "Registracija" with a close button (X) in the top right corner. The form contains the following fields and labels:

- Korisnicko ime: [text input]
- Lozinka : [text input]
- Potvrdite lozinku : [text input]
- Ime: [text input]
- Prezime: [text input]
- Dob: [text input]
- E-Mail adresa: [text input]

At the bottom of the form are two buttons: "Odustani" (Cancel) and "Registriraj se" (Register).

Slika 55: Korisničke upute - Registracija

U slučaju uspješne registracije, korisnik se može prijaviti unošenjem ispravnog korisničkog imena i lozinke u predviđena polja.

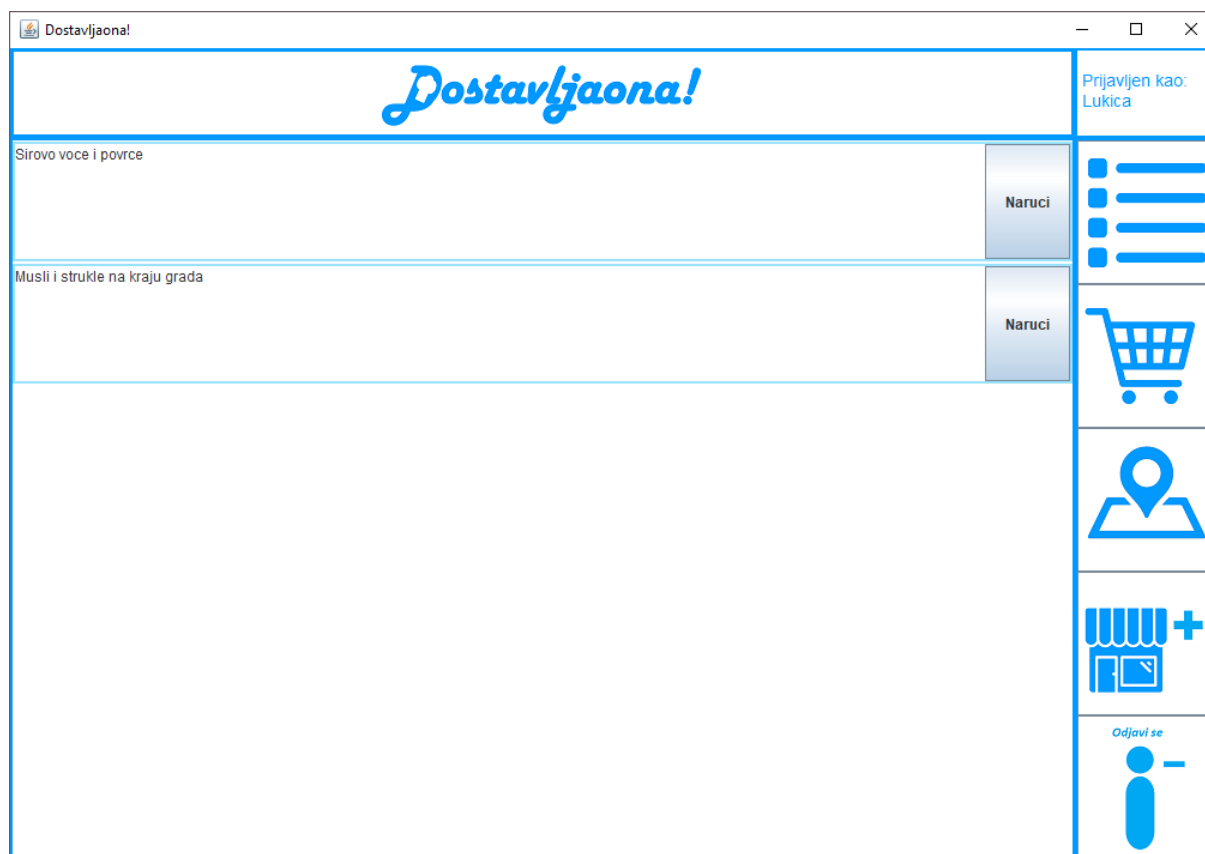
Prijava ×

Korisnicko ime:

Lozinka:

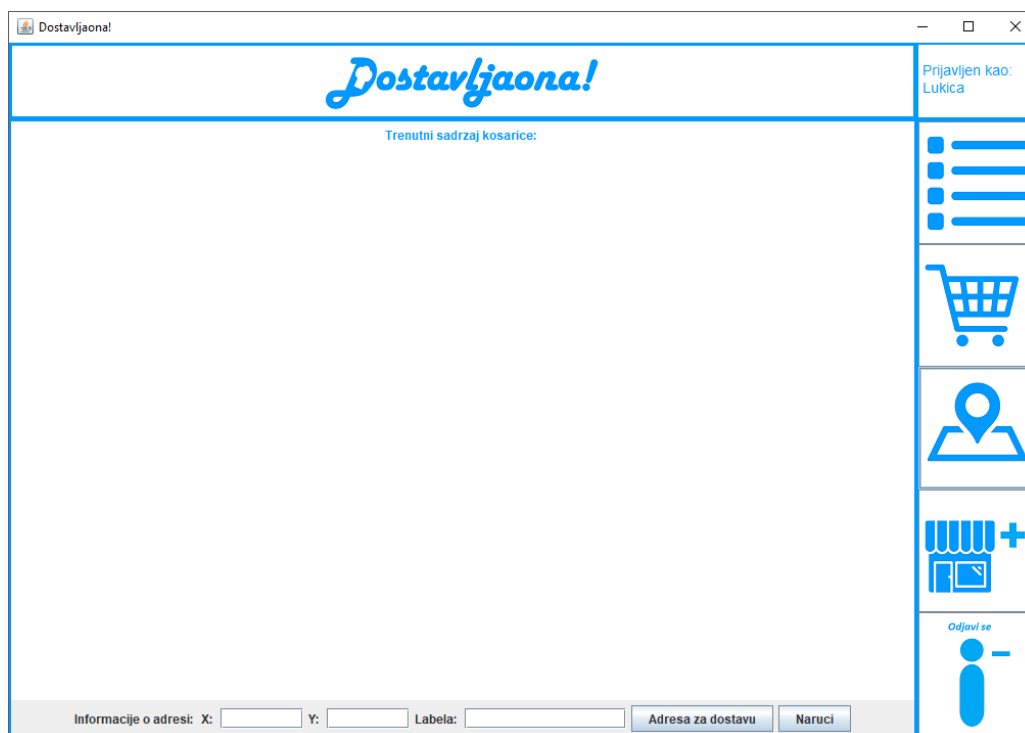
Slika 56: Korisničke upute - Prijava

Nakon prijave korisniku se otvara početni panel na kojem može iz ponuda restorana odabrati željene proizvode koji se dodaju u košaricu.



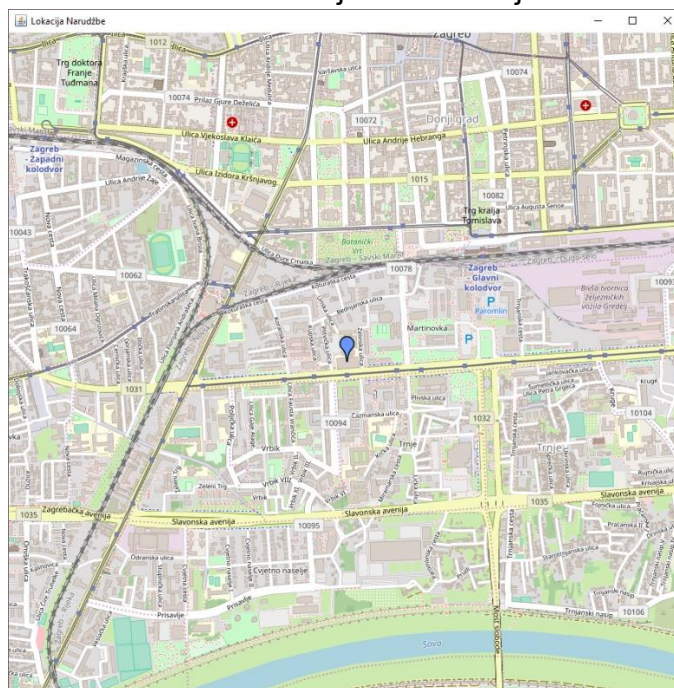
Slika 57: Korisničke upute – Korisnik panel

Također ima mogućnost uređivanja košarice, odnosno brisanja i dodavanja proizvoda.



Slika 58 : Korisničke upute – Uređivanje košarice

Nakon što je gotov sa odabirom proizvoda, klijent finalizira narudžbu unosom lokacije dostave. U svakom trenutku može pratiti trenutnu lokaciju dostavljača. Klikom na kartu, otvara mu se karta sa željenom lokacijom.



Slika 59: Korisničke upute - Karta

Klijent može predložiti restoran. Unosom potrebnih podataka, prijedlog se šalje administratoru, od kojeg se čeka odobrenje prijedloga.

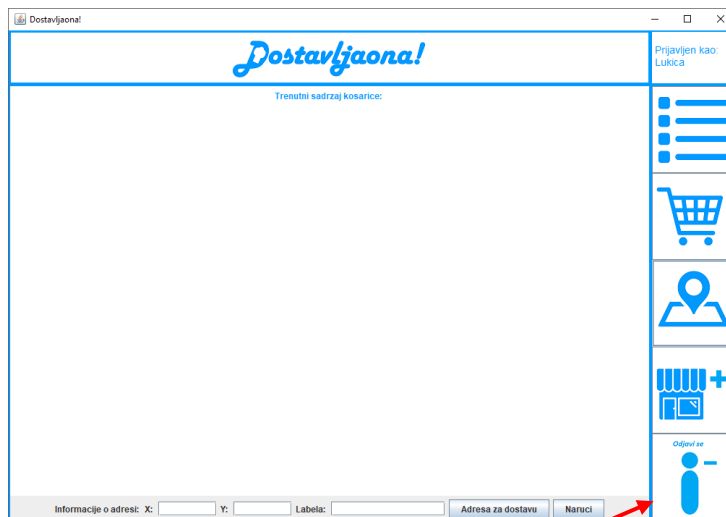


The screenshot shows a window titled "Predlozi" with a close button (X) in the top right corner. The window contains the following elements:

- A text input field labeled "Ime restorana:".
- A text area labeled "Opis usluga restorana:".
- A blue icon representing a restaurant building with a plus sign next to it.
- A button labeled "Dodaj sliku" (Add picture).
- A text input field labeled "Upisi putanju do slike:" (Enter the path to the picture).
- A text label "Unesite GeoLokaciju vasesg restorana:" (Enter the GeoLocation of your restaurant).
- Two text input fields labeled "x:" and "y:" for coordinates.
- A blue text message: "Vas ce zahtjev odobriti administrator u roku od 24 sata" (Your request will be approved by the administrator within 24 hours).
- Two buttons at the bottom: "Predlozi restoran" (Propose restaurant) and "Odustani" (Cancel).

Slika 60: Korisničke upute – Prijedlog restorana

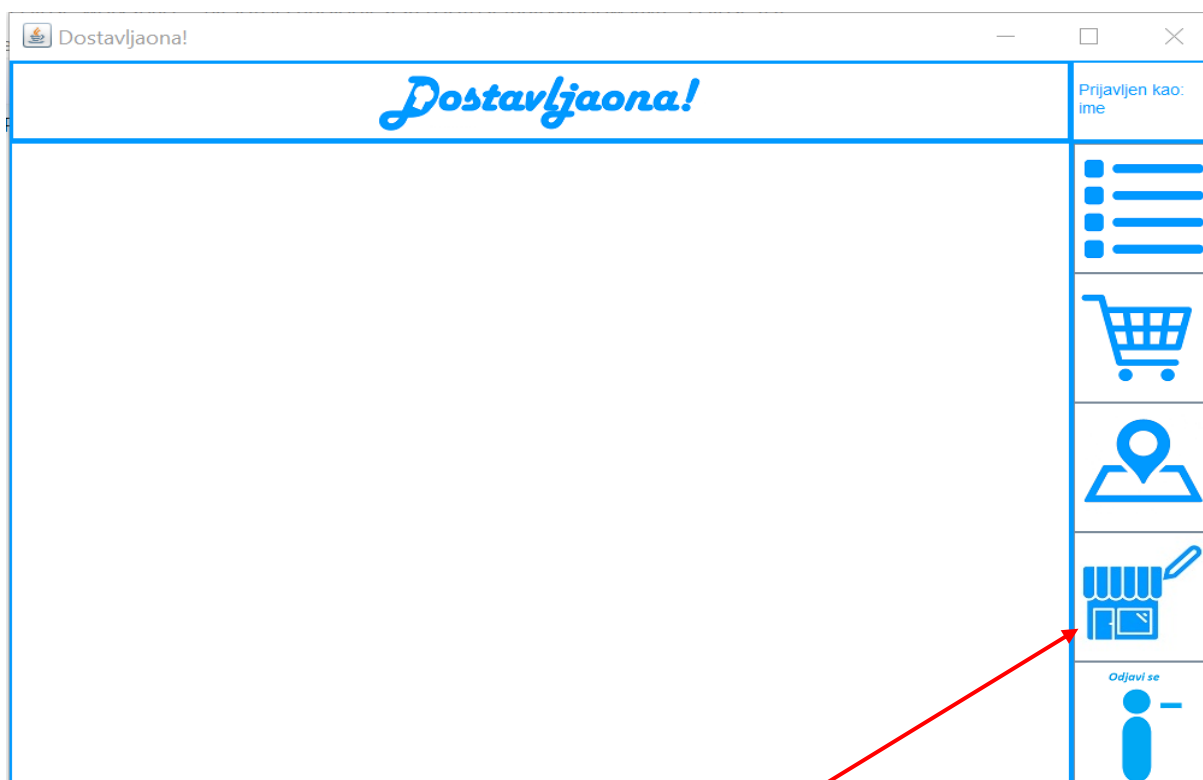




Slika 61: Korisničke upute - Odjava

Rad u aplikaciji korisnik završava odjavom.

Prijavom vlasnika restorana otvara se vlasnički panel.



Slika 62: Korisničke upute – Vlasnik panel

Vlasnik restorana ima opciju uređivanja menija.

## 8. Zaključak i budući rad

Rad u grupi je protekao vrlo dobro i prema planu. Sastanci su se održavali redovito, svaki tjedan jednom, a po potrebi i češće (pod timovi zaduženi za određene zadatke). Trudili smo se posao podijeliti ravnomjerno svim članovima.

U prvom ciklusu na projektu napravili smo opis projektnog zadatka, napisali pojmovnik i dnevnik promjena dokumentacije. Definirali smo i oblikovali sve funkcionalne zahtjeve te napravili obrasce uporabe, dijagrame obrazaca uporabe, a zatim i sekvencijske dijagrame. Napravili smo opis arhitekture sustava i njegovu svrhu i opće prioritete, a zatim i skicu sustava. Na kraju smo napravili model baze podataka i dijagrame razreda i objekata.

U drugom dijelu započeli smo s ostalim UML dijagramima, uvrštavanjem dodatna četiri dijagrama koji prikazuju arhitekturu sustava. Radili smo na implementaciji projekta i naveli odabir tehnologija i alata koje smo koristili za razvoj sustava. Napravili smo dijagrame razmještaja, prikazali isječke bitnih dijelova koda i opisali provedbu ispitivanja implementiranih funkcionalnosti s prikazom odabira ispitnih slučajeva. Na kraju smo dodali upute za instalaciju te korisničke upute.

Zahvaljujući radu u timu stekli smo nove vještine koje ćemo koristiti u daljnjem školovanju. Upoznali smo se s raznim alatima i tehnologijama. Sve u svemu jedno poučno i zanimljivo iskustvo i dobra priprema za rad na projektima u budućnosti.

## 9. Popis literature

- <sup>1</sup> Oblikovanje programske potpore, FER ZEMRIS, <http://www.fer.hr/predmet/opp>
- <sup>2</sup> Oblikovanje programske potpore, FER ZEMRIS, <http://www.zemris.fer.hr/predmeti/opp>
- <sup>3</sup> I. Sommerville, „Software engineering“, 8th ed, Addison Wesley, 2007.
- <sup>4</sup> T.C.Lethbridge, R.Langaniere, „Object-Oriented Software Engineering“, 2nd ed. McGraw-Hill, 2005.
- <sup>5</sup> Software engineering ,Rutgers University, <http://www.ece.rutgers.edu/~marsic/Teaching/SE>
- <sup>6</sup> I. Marsic, „Software engineering book“, Department of Electrical and Computer Engineering, Rutgers University, <http://www.ece.rutgers.edu/~marsic/books/SE>
- <sup>7</sup> Concepts: Requirements, [http://www.upedu.org/upedu/process/gcncpt/co\\_req.htm](http://www.upedu.org/upedu/process/gcncpt/co_req.htm)
- <sup>8</sup> UML 2 Class Diagram Guidelines, <http://www.agilemodeling.com/style/classDiagram.htm>
- <sup>9</sup> Domain Class Diagram Modeling Standards and Guidelines, <http://www.bced.gov.bc.ca/imb/downloads/classdiagramstandards.pdf>
- <sup>10</sup> Astah Community, <http://astah.net/editions/community/>
- <sup>11</sup> OSM API, <https://github.com/westnordost/osmapi/blob/master/README.md>
- <sup>12</sup> GIT, <https://git-scm.com/book/en/v2/Getting-Started-Git-Basics>

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

• Slika 1 Dijagram obrazaca uporabe - cjeloviti pregled.....	19
• Slika 2 Ponašanje anonimnog korisnika.....	20
• Slika 3 Ponašanje klijenta i vlasnika restorana.....	20
• Slika 4 Ponašanje dostavljača.....	21
• Slika 5 Ponašanje dispečera.....	21
• Slika 6 Ponašanje administratora.....	22
• Slika 7 Sekvencijski dijagram – Registriraj se.....	23
• Slika 8 Sekvencijski dijagram – Prijavi se.....	23
• Slika 9 Sekvencijski dijagram – Pregledaj ponude.....	24
• Slika 10 Sekvencijski dijagram – Dodaj proizvod u košaricu.....	24
• Slika 11 Sekvencijski dijagram – Uređivanje košarice.....	25
• Slika 12 Sekvencijski dijagram – Potvrdi narudžbu.....	25
• Slika 13 Sekvencijski dijagram – Prati poziciju.....	26
• Slika 14 Sekvencijski dijagram – Predloži restoran.....	26
• Slika 15 Sekvencijski dijagram – Uređivanje menija.....	27
• Slika 16 Sekvencijski dijagram – Pregled popisa narudžbi.....	27
• Slika 17 Sekvencijski dijagram – Postavi oznaku gotov.....	28
• Slika 18 Sekvencijski dijagram – Pristupi karti.....	28
• Slika 19 Sekvencijski dijagram – Neraspoređene narudžbe.....	29
• Slika 20 Sekvencijski dijagram – Dodjela zadatka.....	29
• Slika 21 Sekvencijski dijagram – Odobri restoran.....	30
• Slika 22 Sekvencijski dijagram – Pregled popisa registriranih korisnika.....	30
• Slika 23 Sekvencijski dijagram – Promjena razine pristupa.....	31
• Slika 24 Nacrt sustava.....	34
• Slika 25 Baza podataka.....	37
• Slika 26 Baza podataka – prikaz grupe razreda komunikacije s bazom podataka.....	38
• Slika 27 Podatkovni podsustav – prikaz grupe razreda koji omogućavaju funkcionalnost sustava.....	40
• Slika 28 Podatkovni podsustav – prikaz razreda koji modeliraju entitete iz stvarnog svijeta.....	41
• Slika 29 Prikaz dijagrama razreda podatkovnog podsustava i odnosa među njima.....	42
• Slika 30 Korisničko sučelje – prikaz razreda koji modeliraju korisničko sučelje.....	44
• Slika 31 Dijagram objekata – prikaz sustava u jednom trenutku.....	45
• Slika 32 Dijagram stanja – Dostavljač.....	46
• Slika 33 Komunikacijski dijagram – Uređivanje košarice.....	47
• Slika 34 Dijagram aktivnosti – Dodjela zadatka.....	48
• Slika 35 Dijagram komponenti.....	49
• Slika 36 Dijagram razmjestaja.....	50
• Slika 37 Isječak programskog koda – prijava i odjava.....	53
• Slika 38 Isječak programskog koda – setRestoranOdobren i setRazinaPristupa.....	54

• Slika 39 Isječak programskog koda-dohvatiAktivneDostavljače i DohvatildNeraspoređenihNarudžbi.	55
• Slika 40 Isječak programskog koda – Panel.....	56
• Slika 41 Ispitivanje programskog rješenja – Prijava.....	57
• Slika 42 Ispitivanje programskog rješenja – Prijava .....	57
• Slika 43 Ispitivanje programskog rješenja – Prijedlog restorana.....	58
• Slika 44 Ispitivanje programskog rješenja – Registracija.....	59
• Slika 45 Izbornik Environment Variables i izbornik Edit environment variable.....	60
• Slika 46 Upute za instalaciju – MySQL.....	61
• Slika 47 Upute za instalaciju – MySQL.....	62
• Slika 48 Upute za instalaciju – MySQL.....	63
• Slika 49 Upute za instalaciju – Eclipse.....	64
• Slika 50 Upute za instalaciju – Eclipse.....	65
• Slika 51 Upute za instalaciju – Eclipse.....	66
• Slika 52 Upute za instalaciju – Eclipse.....	66
• Slika 53 Upute za instalaciju – Eclipse.....	67
• Slika 54 Upute za instalaciju – Eclipse.....	68
• Slika 55 Korisničke upute – Registracija.....	69
• Slika 56 Korisničke upute – Prijava.....	70
• Slika 57 Korisničke upute – Korisnik Panel.....	70
• Slika 58 Korisničke upute – Uređivanje Košarice.....	71
• Slika 59 Korisničke upute – Karta.....	71
• Slika 60 Korisničke upute – Prijedlog restorana.....	72
• Slika 61 Korisničke upute – Odjava.....	73
• Slika 62 Korisničke upute – Vlasnik Panel.....	73

## **Dodatak B: Dnevnik sastajanja**

### **1. Sastanak:**

10.10.2018.

U sklopu uvodne laboratorijske vježbe, prvi službeni sastanak cijele grupe.

### **2. sastanak:**

16.10.2018.

U sklopu laboratorijske vježbe.

Prisutni svi članovi tima.

Razjasnjena neka pitanja vezana uz problematiku zadatka.

Dogovorena raspodjela posla za 3. tjedan.

Maja, Lucia, Josip - obrasci uporabe

Ivan, Luka - tehnologije

Luka - opis zadatka

Toni - dijagrami

### **3. sastanak:**

23.10.2018.

Tim se sastao na sat vremena u svrhu pregleda i provjere do tada odrađenog posla.

Prisutni svi članovi tima.

Detaljno pregledana cijela, do tada napravljena, dokumentacija; fokus na provjeri obrazaca uporabe.

Postignut dogovor oko dorade obrazaca uporabe prije daljnje izrade dijagrama.

### **4. sastanak:**

31.10.2018.

U sklopu laboratorijske vježbe.

Prisutni svi članovi tima osim Lucie.

Svrha sastanka bila je dobiti povratne informacije o do sada napravljenom poslu od asistenta i demonstratora.

## 5. sastanak:

12.10.2018.

Sastanak održan nakon nastave.

Prisutni svi članovi tima.

Sastanak u svrhu dogovora oko dijagrama razreda, započeto modeliranje.

## 6.sastanak

15.10.2018.

U sklopu laboratorijske vježbe.

Prisutni svi članovi tima.

Svrha sastanka bila je dobivanje povratnih informacija za finalizaciju dokumentacije verzije 1.0

## 7.sastanak

5.12.2018.

Sastanak održan nakon nastave.

Prisutni svi članovi tima.

Svrha sastanka bila je raspodjela zadataka, dogovor o implementaciji.

## 8.sastanak

17.12.2018.

Sastanak održan prije nastave.

Prisutni svi članovi tima.

Svrha sastanka bila je pregled odrađenog posla.

## 9.sastanak

11.01.2019.

U sklopu laboratorijske vježbe.

Prisutni svi članovi tima.

Održati demonstraciju alfa inačice naše projektne aplikacije

## 10.sastanak

16.01.2019.

Sastanak održan nakon nastave.

Prisutni svi članovi tima.

Finalizacija projekta te dorada određenih dijelova dokumentacije.



## Dodatak C: Prikaz aktivnosti grupe

Popis aktivnosti	Članovi grupe (abecednim redom)					
	Luka Mrković	Lucia Barišić	Maja Matešić	Josip Ramljak	Toni Šterbić	Ivan Žuglić
Upravljanje projektom	95%	1%	1%	1%	1%	1%
Opis projektnog zadatka	100%	0%	0%	0%	0%	0%
Rječnik pojmova	0%	50%	50%	0%	0%	0%
Opis funkcionalnih zahtjeva	10%	25%	25%	25%	15%	0%
Opis ostalih zahtjeva	0%	0%	0%	0%	0%	100%
Arhitektura i dizajn sustava						
Svrha, opći prioriteti i skica sustava	30%	0%	0%	0%	0%	70%
Dijagram razreda s opisom	45%	0%	0%	25%	30%	0%
Dijagram objekata	0%	0%	0%	0%	80%	20%
Ostali UML dijagrami	0%	34%	33%	33%	0%	0%
Implementacija i korisničko sučelje						
Dijagram razmještaja	0%	100%	0%	0%	0%	0%
Korištene tehnologije i alati	0%	50%	50%	0%	0%	0%
Isječak programskog kôda	0%	0%	0%	100%	0%	0%
Ispitivanje programskog rješenja	0%	33%	33%	33%	0%	0%
Upute za instalaciju	33%	0%	0%	33%	0%	33%
Korisničke upute	0%	33%	33%	33%	0%	0%
Plan rada	0%	0%	100%	0%	0%	0%

<b>Pregled rada i stanje ostvarenja</b>	0%	50%	50%	0%	0%	0%
<b>Zaključak i budući rad</b>	100%	0%	0%	0%	0%	0%
<b>Popis literature</b>	0%	0%	0%	100%	0%	0%
<b>Dodaci</b>						
<b>Indeks</b>	0%	100%	0%	0%	0%	0%
<b>Dnevnik sastajanja</b>	40%	60%	0%	0%	0%	0%

## **Dodatak D: Plan rada / Pregled rada i stanje ostvarenja**

Plan rada za drugi ciklus projekta je da do kraja popunimo i napišemo potrebnu dokumentaciju, implementiramo programsko rješenje zadatka i pri tome ravnomjerno raspodijelimo posao između članova tima.

U drugom ciklusu projekta implementirane su sve komponente programske podrške. Integracija i testiranje komponenti nisu u potpunosti provedeni – rad na tome se nastavlja do završne demonstracije projekta. Želja tima je ostvariti što veću razinu integracije.