

# Programsko inženjerstvo

Ak. god. 2020./2021.

## Papirni.ca

Dokumentacija, Rev. 2

Grupa: *fooBar*

Voditelj: *Fran Hrabar*

Datum predaje: *14.1.2021.*

Nastavnik: *Eugen Vušak*

# Sadržaj

<b>1 Dnevnik promjena dokumentacije</b>	<b>3</b>
<b>2 Opis projektnog zadatka</b>	<b>4</b>
<b>3 Specifikacija programske potpore</b>	<b>8</b>
3.1 Funkcionalni zahtjevi . . . . .	8
3.1.1 Obrasci uporabe . . . . .	10
3.1.2 Sekvencijski dijagrami . . . . .	22
3.2 Ostali zahtjevi . . . . .	26
<b>4 Arhitektura i dizajn sustava</b>	<b>27</b>
4.1 Baza podataka . . . . .	28
4.1.1 Opis tablica . . . . .	28
4.1.2 Dijagram baze podataka . . . . .	34
4.2 Dijagram razreda . . . . .	35
4.3 Dijagram stanja . . . . .	39
4.4 Dijagram aktivnosti . . . . .	40
4.5 Dijagram komponenti . . . . .	42
<b>5 Implementacija i korisničko sučelje</b>	<b>43</b>
5.1 Korištene tehnologije i alati . . . . .	43
5.2 Ispitivanje programskog rješenja . . . . .	45
5.2.1 Ispitivanje komponenti . . . . .	45
5.2.2 Ispitivanje sustava . . . . .	45
5.3 Dijagram razmještaja . . . . .	46
5.4 Upute za puštanje u pogon . . . . .	47
<b>6 Zaključak i budući rad</b>	<b>51</b>
<b>Popis literature</b>	<b>53</b>
<b>Indeks slika i dijagrama</b>	<b>55</b>

**Dodatak: Prikaz aktivnosti grupe**

**56**

# 1. Dnevnik promjena dokumentacije

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Napravljen predložak. Dodan opis zadatka.	Pećanić	19.10.2020.
0.2	Rad na poglavljima 3 i 4.	Hrabar	31.10.2020.
0.3	Rad na poglavljima 3 i 4. Dodani manji detalji u opis.	Pećanić	5.11.2020.
0.4	Rad na poglavlju 4.	Hrabar	7.11.2020.
0.5	Rad na poglavlju <i>Dodatak</i>	Hrabar, Mihovilović, Protulipac, Likakur, Pećanić	11.11.2020.
1.0	Verzija samo s bitnim dijelovima za 1. ciklus	Pećanić, Hrabar	13.11.2020.
1.01	Promjena manjih tekstualnih detalja u poglavlju 4	Pećanić	28.11.2020.
1.02	Promjena dijagrama obrazaca uporabe	Pećanić	28.12.2020.
1.1	Rad na poglavlju 5.1 i 6	Pećanić	9.1.2021.
1.2	Rad na poglavlju 5.4 i 4.1	Hrabar	10.1.2021.
1.8	Rad na dijagramu stanja i komponenti	Vujević	12.1.2021.
1.9	Rad na poglavljima 4.4, 5.2, 5.3	Pećanić	13.1.2021.
2.0	Konačni tekst predložka dokumentacije	Hrabar, Mihovilović, Protulipac, Likakur, Pećanić, Vujević, Kukešćak	14.1.2021.

## 2. Opis projektnog zadatka

Papirni.ca je aplikacija koja služi za poslovanje prodavaonice papira unutar čije ponude postoje razne vrste i veličine papira uz pojedine dodatne usluge. Usluge variraju ovisno o vrsti korisnika koji se njima služi, a podjela korisnika je sljedeća:

1. **neregistrirani korisnik**

2. **registrirani korisnik**

- (a) privatni

- (b) poslovni

3. **administrator**

Neregistrirani korisnici mogu mijenjati kontekst platforme iz privatnog u poslovni i obrnuto te im je dostupno isključivo pregledavanje usluga unutar istih, dok je za kupovinu papira i/ili pružanje usluga potrebna registracija.

Registrirani privatni korisnici za prijavu koriste ime, prezime i e-mail adresu. Jedina usluga koja se pruža ovim korisnicima jest prodaja raznih vrsta papira s obzirom na sljedeće faktore:

1. boja

2. vrsta

3. težina (gsm)

4. format (A0 - A10)

Registrirani poslovni korisnici za prijavu koriste ime i OIB firme, ime kontakt osobe iz firme te kontaktni broj telefona i e-mail adresu iste. Kao i privatnima, poslovnim korisnicima omogućena je prodaja papira koja, uz već navedene, ima dvije dodatne opcije - izbor nekonvencionalne veličine papira po narudžbi i mogućnost dodatka vodenog žiga na papire. Korisnik mora priložiti sliku koju želi upotrijebiti kao vodeni žig ako odabere tu opciju. Osim prodaje papira, poslovnim korisnicima je omogućeno slanje posebnog tekstualnog zahtjeva putem kojeg se može zatražiti usluga koja nije u standardnoj ponudi papirnice - npr. specifičan način pakiranja

narudžbe. Da bi dodatni zahtjev bio validan, korisnik mora unijeti kratki naziv usluge koju traži i kontakt podatke. Uz jednokratno korištenje usluga, poslovni korisnici mogu napraviti mjesečnu pretplatu na neki proizvod - time se odabrana kupovina obavlja automatski svaki mjesec. Ako korisnik u košarici ima proizvode koje kupuje jednokratno uz one koji su označeni za pretplatu, nakon potvrde kupnje svi će proizvodi biti naplaćeni s time da će se oni označeni za pretplatu naplaćivati mjesečno od tog trenutka do otkazivanja pretplate. Poslovni korisnik može pregledavati svoje pretplate i otkazati ih u bilo kojem trenutku.

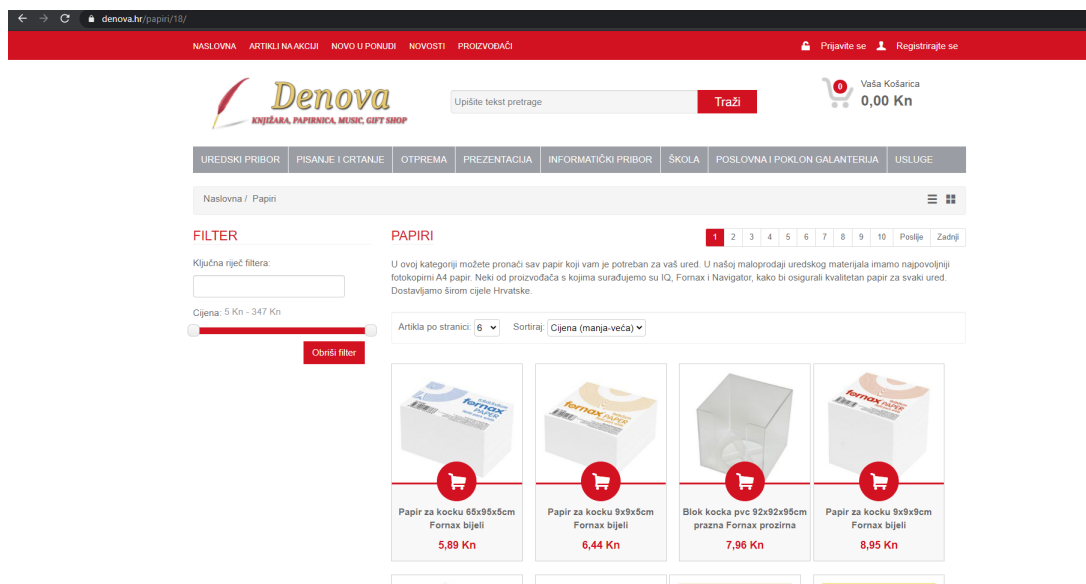
Administratorima je omogućeno sljedeće:

1. pregled povijesti kupljenih usluga za sve korisnike
2. pregled i otkazivanje pretplata poslovnih korisnika
3. pregled i mogućnost uklanjanja dodatnih uskuga dodanih putem tekstualnog zahtjeva

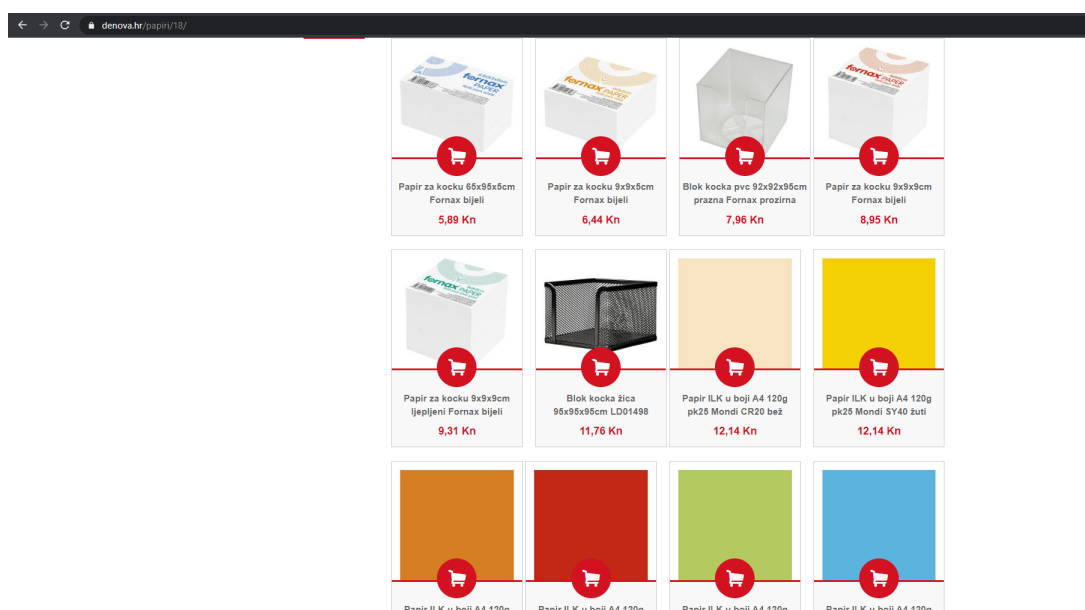
Kupovina funkcionira na sljedeći način: odabirom na proizvod on se dodaje u košaricu, a plaćanje se vrši putem servisa PayPal. Dodatna usluga slanja tekstualnog zahtjeva se ne dodaje u košaricu nego se za nju napisani zahtjev najprije mora odobriti i tek kada (i ako) je on odobren zahtjev će se dodati tom poslovnom korisniku. U slučaju problema sa zahtjevom, korisnika se obavještava putem kontakt podataka koji su uneseni prilikom predaje zahtjeva te se istim putem i rješavaju. Nakon rješavanja potencijalnih problema dodatna usluga se dodaje korisniku, a njega se obavještava o uspješnom dodavanju usluge (također putem prethodno navedenih kontakt podataka). Takva usluga vidljiva je isključivo tom korisniku (među ostalim uslugama).

Projekt kao što je Papirni.ca može koristiti bilo kojem manjem obrtu koji se bavi ovakvom djelatnošću, a uz proširenje bi se ovakva aplikacija mogla koristiti i za veće trgovine. Budući da se zbog pandemije koronavirusa naglašava smanjenje međuljudskog kontakta na minimum, online trgovine postaju sve prometnije tako da bi ovakvo rješenje bilo iznimno prikladno za papirnice koje posluju u ovo doba. Izuzevši trenutnu situaciju, ovakav pristup pogodan je udaljenim kupcima koji nisu u mogućnosti fizički posjetiti trgovinu i osobno napraviti narudžbu. Također, kupovina putem aplikacije ima određenu preglednost s izlistanim opcijama koju kupci možda ne bi dobili jednim pogledom u fizičkoj papirnici.

Primjer sličan ovome projektu je web-stranica knjižare i papirnice Denova (<https://www.denova.hr/>). Njihova ponuda nije ograničena samo na papire, no kao i kod Papirni.ca aplikacije, među papirima također postoji mnogo opcija i dosta širok izbor.



Slika 2.1: Snimka web-stranice papirnice Denova



Slika 2.2: Snimka raznolike ponude papira papirnice Denova

Papirni.ca namijenjena je svakome tko ima interes kupovati papir putem aplikacije. Za pretpostaviti je da bi se njome među privatnim korisnicima našlo dosta studenata i učenika budući da su mladi skloniji ovakvoj vrsti kupovine jer su tehnički pismeniji od starijih generacija. Dodatne usluge koje se pružaju poslovnim korisnicima zasigurno bi bile primamljive obrtnicima i manjim firmama koje koriste svoj logo na papirima koji su priloženi uz isporuku proizvoda.

Projektno rješenje moglo bi se prilagoditi na više načina. Proširenjem ponude proizvoda mogla bi se ostvariti aplikacija za rad knjižare ili kakve veće trgovine sa sličnim asortimanom. Međutim, samo promjenom vrste proizvoda i opcija koje se za njih mogu odabrati, ovakva aplikacija mogla bi se koristiti za razne trgovine i obrte kao što su slastičarnica, pekara, trgovina odjećom, trgovina igračkama i sl.

Projekt Papirni.ca od resursa ima na raspolaganju (ljudsku) radnu snagu članova tima fooBar, (besplatan) pristup mnogim razvojnim okruženjima i vrstama programske potpore zahvaljujući FER-u te vremenski okvir od 1.10.2020. do 14.1.2021. za isporuku. Finalna isporuka projekta jest aplikacija modernog i vizualno privlačnog dizajna koja sadrži funkcionalnosti navedene ranije u ovom poglavlju, no prije finalne isporuke postoji nekoliko kontrolnih točaka koje obuhvaćaju isporuku aplikacije u verziji koja je dotad napravljena. Rokovi vezani uz periodične (djelomične) isporuke aplikacije dani su u sljedećoj tablici:

Rok	Opis isporuke
13.11.2020.	Predaja projekta za prvi ciklus (v0.0)
7.1.-13.1.2021.	Demonstracija alfa inačice rada aplikacije
14.1.2021.	Predaja projekta za drugi ciklus (finalna isporuka)

Mogućih nadogradnji ovakvog rješenja ima mnogo. Neke od njih su: dodavanje tekstone papira kao još jedne opcije pri kupnji papira, omogućavanje neregistriranim korisnicima da bez registracije (uz jednokratno davanje svojih podataka) naprave narudžbu, dodavanje godišnjih pretplata za poslovne korisnike.



## 3. Specifikacija programske potpore

### 3.1 Funkcionalni zahtjevi

#### Dionici:

1. Vlasnik (naručitelj)
2. Klijenti
  - (a) Privatni
  - (b) Poslovni
3. Administrator
4. Razvojni tim

#### Aktori i njihovi funkcionalni zahtjevi:

1. Neregistrirani/neprijavljeni korisnik (inicijator) može:
  - (a) jednostavno mijenjati kontekst iz privatnog u poslovni i obrnuto
  - (b) pregledavati predefinirane izbore boja, vrsta, težina i formata papira
2. Privatni klijent (inicijator) može:
  - (a) odabrati predefinirane izbore boja, vrsta, težina i formata papira i platiti svoju narudžbu
3. Poslovni klijent (inicijator) može:
  - (a) odabrati predefinirane izbore boja, vrsta, težina i formata papira i platiti svoju narudžbu
  - (b) odabrati papir nekonvencionalnih veličina i dodati vodeni žig
  - (c) zatražiti dodatne usluge koje nisu na ponudi
  - (d) pregledavati i odabrati potvrđene dodatne usluge
  - (e) zakazati, pregledavati i otkazivati mjesečne pretplate
4. Administrator (inicijator) može:

- (a) pregledavati pojedine tekstualne zahtjeve poslovnih klijenata, te potvrditi ili odbiti i uklanjati iste
- (b) pregledavati i otkazivati mjesečne pretplate
- (c) pregledavati povijest kupljenih usluga za sve korisnike

5. Baza podataka (sudionik) može:

- (a) pohranjivati podatke o klijentima
- (b) pohranjivati podatke o artiklima na ponudi
- (c) pohranjivati narudžbe

### 3.1.1 Obrasci uporabe

#### UC1-Pregled preddefiniranih papira

- **Glavni sudionik:** Korisnik, klijent
- **Cilj:** Pregledavanje preddefiniranih artikala
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju za pregled papira u načinu za privatnog klijenta
  2. Korisnik pregledava postojeće opcije po padajućim izbornicima

#### UC2-Registracija privatnog klijenta

- **Glavni sudionik:** Korisnik
- **Cilj:** Stvoriti privatni korisnički račun
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju za registraciju privatnog klijenta
  2. Korisnik unosi potrebne korisničke podatke
  3. Korisnik prima obavijest o uspješnoj registraciji
- **Opis mogućih odstupanja:**
  - 2.a Odabir već zauzetog korisničkog imena i/ili e-maila, unos korisničkog podatka u nedozvoljenom formatu ili pružanje neispravnoga e-maila
    1. Sustav obavještava korisnika o neuspjelom upisu i vraća ga na stranicu za registraciju
    2. Korisnik mijenja potrebne podatke te završava unos ili odustaje od registracije

#### UC3-Registracija poslovnog klijenta

- **Glavni sudionik:** Korisnik
- **Cilj:** Stvoriti poslovni korisnički račun
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju za registraciju poslovnog klijenta
  2. Korisnik unosi potrebne korisničke podatke

3. Korisnik prima obavijest o uspješnoj registraciji
- **Opis mogućih odstupanja:**
  - 2.a Odabir već zauzetog ili neispravnog imena kompanije, OIB-a, kontakt osobe, telefona i/ili e-maila
    1. Sustav obavještava korisnika o neuspjelom upisu i vraća ga na stranicu za registraciju
    2. Korisnik mijenja potrebne podatke te završava unos ili odustaje od registracije

#### UC4-Prijava privatnog klijenta u sustav

- **Glavni sudionik:** Privatni klijent
- **Cilj:** Dobiti pristup korisničkom sučelju
- **Sudionici:** Baza podataka
- **Preduvjet:** Registracija
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju za registraciju privatnog klijenta
  2. Korisnik unosi potrebne korisničke podatke
  3. Korisnik prima obavijest o uspješnoj prijavi
  4. Korisnik gubi pristup poslovnim funkcijama
- **Opis mogućih odstupanja:**
  - 2.a Neispravno ime i/ili lozinka
    1. Sustav obavještava korisnika o neuspjelom upisu i vraća ga na stranicu za prijavu
    2. Korisnik ponovno unosi podatke za prijavu ili odustaje od prijave

#### UC5-Prijava poslovnog klijenta u sustav

- **Glavni sudionik:** Poslovni klijent
- **Cilj:** Dobiti pristup korisničkom sučelju
- **Sudionici:** Baza podataka
- **Preduvjet:** Registracija
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju za registraciju poslovnog klijenta
  2. Korisnik unosi potrebne korisničke podatke
  3. Korisnik prima obavijest o uspješnoj prijavi
- **Opis mogućih odstupanja:**
  - 2.a Neispravno ime i/ili lozinka

1. Sustav obavještava korisnika o neuspjelom upisu i vraća ga na stranicu za prijavu
2. Korisnik ponovno unosi podatke za prijavu ili odustaje od prijave

#### UC6-Dodavanje usluge u košaricu

- **Glavni sudionik:** Klijent
- **Cilj:** Dodati uslugu u košaricu
- **Sudionici:** -
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
  1. Klijent odabire preddefiniranu uslugu
  2. Klijent odabire opciju "Dodaj u košaricu"
- **Opis mogućih odstupanja:**
  - 2.a Nisu odabrani svi podaci o papiru
    1. Sustav obavještava klijenta o neuspjelom odabiru i pita ga da provjeri podatke
    2. Klijent odabire potrebne podatke te odabire opciju "Dodaj u košaricu" ili odustaje od dodavanja

#### UC7-Dodavanje papira posebne dimenzije i/ili sa žigom u košaricu

- **Glavni sudionik:** Poslovni klijent
- **Cilj:** Dodati papira posebne dimenzije i/ili sa žigom u košaricu
- **Sudionici:** -
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
  1. Klijent odabire dimenzije papira i/ili priloži sliku vodenog žiga
  2. Klijent odabire opciju "Dodaj u košaricu"
- **Opis mogućih odstupanja:**
  - 2.a Dimenzije izvan određenih granica ili u krivom formatu
    1. Sustav obavještava klijenta o neuspjelom upisu i pita ga da provjeri podatke
    2. Klijent mijenja potrebne podatke te odabire opciju "Dodaj u košaricu" ili odustaje od dodavanja
  - 2.b Vodeni žig u krivom formatu
    1. Sustav obavještava klijenta o neuspjelom prilogu žiga i obavještava ga da je potrebna slika žiga

2. klijent prilaže sliku te odabire opciju "Dodaj u košaricu" ili odustaje od dodavanja

### UC8-Slanje tekstualnog zahtjeva za dodatne usluge

- **Glavni sudionik:** Poslovni klijent
- **Cilj:** Zatražiti uslugu koja nije na ponudi
- **Sudionici:** Baza podataka, administrator
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
  1. Klijent odabire opciju za slanje tekstualnog zahtjeva
  2. Klijent unosi zahtjev, naziv usluge i kontakt podatke
  3. Klijent prima obavijest da njegov zahtjev čeka potvrdu
- **Opis mogućih odstupanja:**
  - 2.a Unesen zahtjev i opis su prazni
    1. Sustav obavještava klijenta da zahtjevi ne mogu biti prazni
    2. Klijent unosi potrebne podatke te završava unos ili odustaje od zahtjeva
  - 2.b Uneseni neispravni kontakt podaci
    1. Sustav obavještava klijenta o neispravnosti kontakt podataka
    2. Klijent mijenja kontakt podatke te završava unos ili odustaje od registracije

### UC9-Zakazivanje mjesečne pretplate

- **Glavni sudionik:** Poslovni klijent
- **Cilj:** Zakazivanje mjesečne pretplate za pojedine usluge
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
  1. Klijent odabire svoju košaricu
  2. Klijent u košarici odabire usluge za koje želi mjesečnu pretplatu i označava ih

### UC10-Pregled mjesečnih pretplata

- **Glavni sudionik:** Poslovni klijent
- **Cilj:** Pregledavanje aktivnih mjesečnih pretplata
- **Sudionici:** Baza podataka

- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
  1. Klijent odabire opciju za pregled mjesečnih pretplata

#### UC11-Otkazivanje mjesečnih pretplata

- **Glavni sudionik:** Poslovni klijent
- **Cilj:** Otkazivanje mjesečnih pretplata
- **Sudionici:** Baza podataka, administrator
- **Preduvjet:** Korisnik je prijavljen i postoje mjesečne pretplate
- **Opis osnovnog tijeka:**
  1. Klijent odabire opciju za pregled mjesečnih pretplata
  2. Klijent odabire mjesečnu pretplatu koju želi otkazati
  3. Klijent prima obavijest da je pretplata otkazana

#### UC12-Pregled sadržaja košarice

- **Glavni sudionik:** Klijent
- **Cilj:** Prikaz i pregled sadržaja košarice
- **Sudionici:** -
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
  1. Klijent odabire opciju za pregled košarice

#### UC13-Uređivanje sadržaja košarice

- **Glavni sudionik:** Klijent
- **Cilj:** Micanje usluga iz košarice, promjena kvantitete
- **Sudionici:** -
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
  1. Klijent odabire opciju za pregled košarice
  2. Klijent makne i/ili uređuje usluge u košarici

#### UC14-Naručivanje usluga

- **Glavni sudionik:** Klijent
- **Cilj:** Potvrditi narudžbu kako bi ju mogao platiti
- **Sudionici:** -
- **Preduvjet:** Korisnik je prijavljen

- **Opis osnovnog tijeka:**
  1. Klijent odabire opciju za pregled košarice
  2. Klijent odabire opciju za naručivanje
  3. Klijent se usmjeri na plaćanje usluge
- **Opis mogućih odstupanja:**
  - 2.a Košarica je prazna
    1. Klijenta sustav obavijesti da se narudžba provodi samo nad nepraznom košaricom i vraća ga se na pregled košarice

#### UC15-Plaćanje usluge

- **Glavni sudionik:** Klijent
- **Cilj:** Plaćanje usluge kako bi proces kupnje usluge bio završen
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
  1. Klijent za odabranu opciju plaćanja unosi svoje podatke
  2. Klijent potvrđuje plaćanje
  3. Košarica se isprazni i klijenta se vraća na pregled košarice
- **Opis mogućih odstupanja:**
  - 2.a Nedovoljan iznos na računu za plaćanje narudžbe
    1. Klijenta se obavještava da je iznos nedostatan za plaćanje narudžbe i vraća ga na pregled košarice

#### UC16-Prikaz povijesti kupljenih usluga za sve korisnike

- **Glavni sudionik:** Administrator
- **Cilj:** Prikaz povijesti kupljenih usluga za sve korisnike
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
  1. Administrator odabire opciju za prikaz povijesti kupljenih usluga

#### UC17-Prikaz vlastite povijesti kupljenih usluga

- **Glavni sudionik:** Klijent
- **Cilj:** Prikaz vlastite povijesti kupljenih usluga
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen



- **Opis osnovnog tijeka:**

1. Klijent odabire opciju za prikaz povijesti kupljenih usluga

### UC18-Potvrđivanje dodatnih usluga

- **Glavni sudionik:** Administrator, poslovni klijent
- **Cilj:** Potvrđivanje i dodavanje dodatnih usluga, poslanih tekstualnim zahtjevom, poslovnom klijentu
- **Sudionici:** Baza podataka
- **Preduvjet:** Administrator je prijavljen i tekstualni zahtjev postoji
- **Opis osnovnog tijeka:**
  1. Administrator odabire opciju za prikaz tekstualnih zahtjeva koji čekaju potvrdu
  2. Administrator odabire zahtjev koji želi potvrditi i potvrđuje ga
  3. Zahtjev se makne iz administratorovog prikaza tekstualnih zahtjeva koji čekaju potvrdu i prikazuje u opciji dodatnih usluga poslovnog klijenta čiji je zahtjev potvrđen

### UC19-Odbijanje dodatnih usluga

- **Glavni sudionik:** Administrator, poslovni klijent
- **Cilj:** Odbijanje dodatnih usluga, poslanih tekstualnim zahtjevom, poslovnom klijentu
- **Sudionici:** Baza podataka
- **Preduvjet:** Administrator je prijavljen i tekstualni zahtjev postoji
- **Opis osnovnog tijeka:**
  1. Administrator odabire opciju za prikaz tekstualnih zahtjeva koji čekaju potvrdu
  2. Administrator odabire zahtjev koji želi odbiti i odbija ga
  3. Zahtjev se makne iz administratorovog prikaza tekstualnih zahtjeva koji čekaju potvrdu i obavještava se poslovnog klijenta kojem je zahtjev odbijen

### UC20-Pregled potvrđenih dodatnih usluga

- **Glavni sudionik:** Administrator
- **Cilj:** Pregledavanje potvrđenih dodatnih usluga
- **Sudionici:** Baza podataka
- **Preduvjet:** Administrator je prijavljen

- **Opis osnovnog tijeka:**

1. Administrator odabire opciju za pregled potvrđenih dodatnih usluga
2. Administrator pregledava sve potvrđene dodatne usluge

#### UC21-Uklanjanje potvrđenih dodatnih usluga

- **Glavni sudionik:** Administrator, poslovni klijent
- **Cilj:** Uklanjanje potvrđenih dodatnih usluga iz ponude poslovnog klijenta
- **Sudionici:** Baza podataka
- **Preduvjet:** Administrator je prijavljen i potvrđene dodatne usluge postoje
- **Opis osnovnog tijeka:**
  1. Administrator odabire opciju za prikaz potvrđenih dodatnih usluga
  2. Administrator odabire zahtjev koji želi ukloniti i uklanja ga
  3. Zahtjev se makne iz administratorovog prikaza potvrđenih dodatnih usluga, obavještava se poslovnog klijenta i uklanja mu se dodatna usluga kao i postojeće pretplate sa tom uslugom

#### UC22-Pregled dodatnih usluga

- **Glavni sudionik:** Poslovni klijent
- **Cilj:** Pregled dodatnih usluga ovog poslovnog klijenta
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
  1. Klijent odabire opciju za prikaz dodatnih usluga

#### UC23-Dodavanje dodatnih usluga u košaricu

- **Glavni sudionik:** Poslovni klijent
- **Cilj:** Dodavanje dodatnih usluga u košaricu
- **Sudionici:** -
- **Preduvjet:** Klijent je prijavljen
- **Opis osnovnog tijeka:**
  1. Klijent odabire dodatnu uslugu među potvrđenim dodatnim uslugama i dodaje ju u košaricu

#### UC24-Pregled mjesečnih pretplata poslovnih klijenata

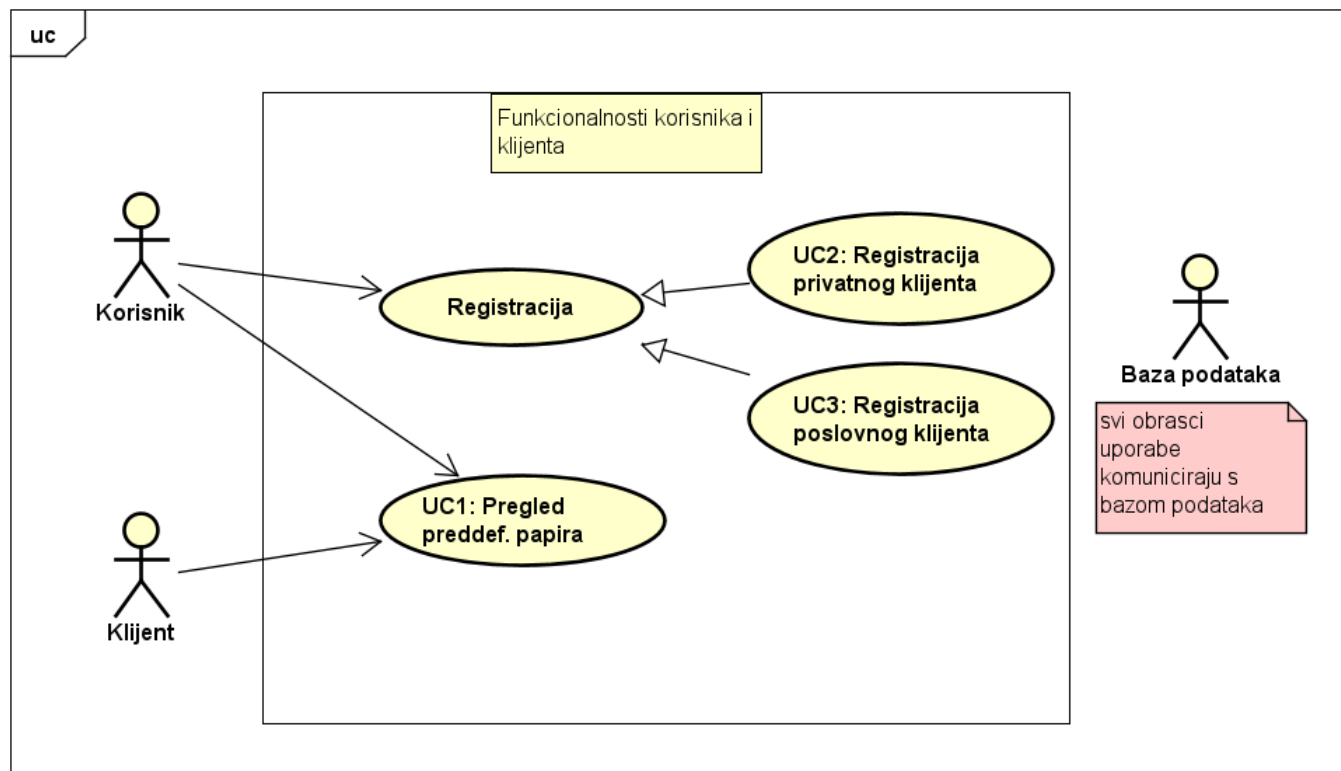
- **Glavni sudionik:** Administrator
- **Cilj:** Pregledavanje aktivnih mjesečnih pretplata

- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
  1. Administrator odabire pregled mjesečnih pretplata za poslovne korisnike

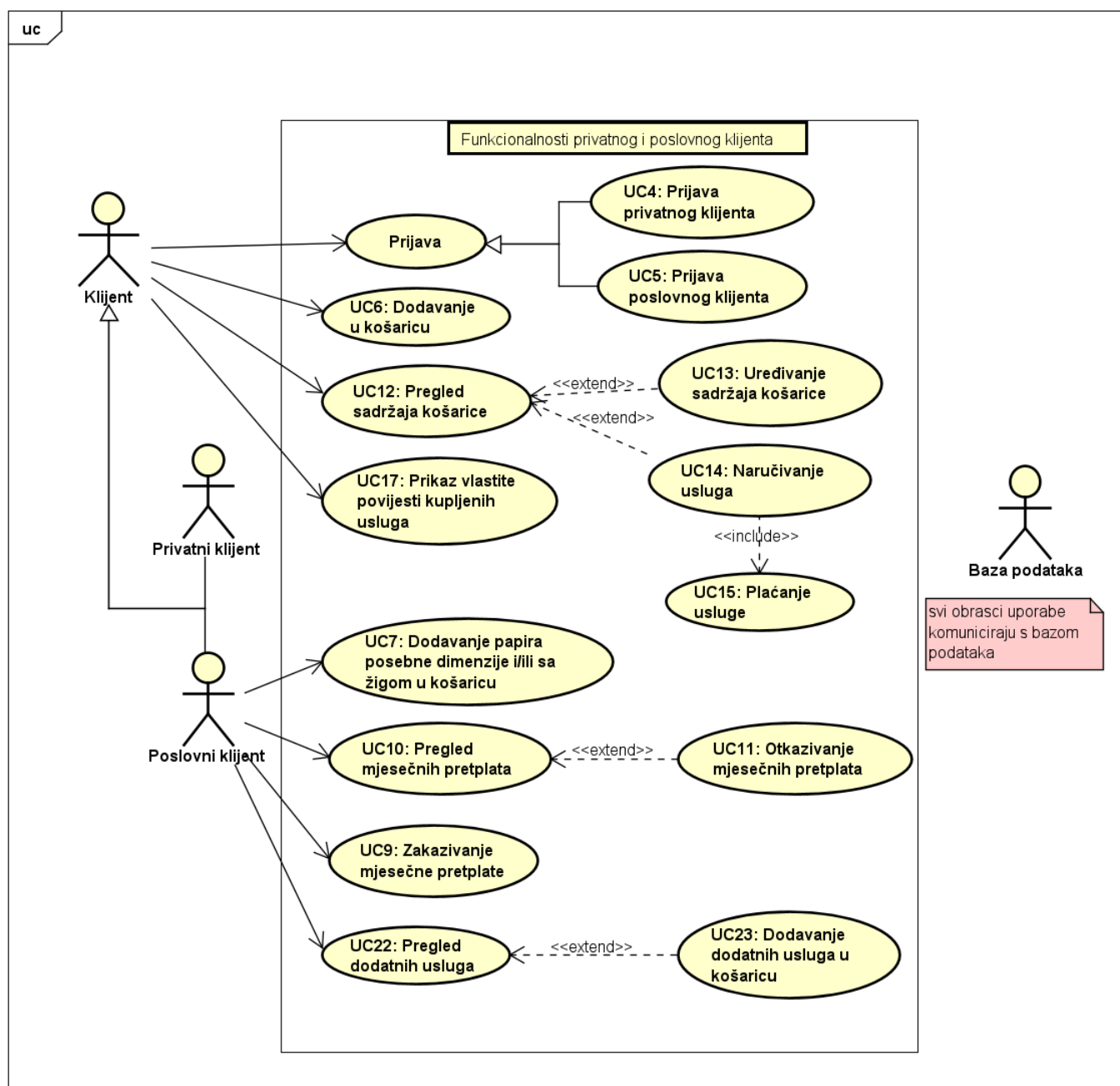
#### **UC25-Otkazivanje mjesečne pretplate poslovnih klijenata**

- **Glavni sudionik:** Administrator
- **Cilj:** Otkazivanje mjesečnih pretplata
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
  1. Administrator odabire pregled mjesečnih pretplata za poslovne korisnike
  2. Administrator otkazuje (deaktivira) odabranu mjesečnu pretplatu
  3. Administrator prima obavijest da je pretplata otkazana

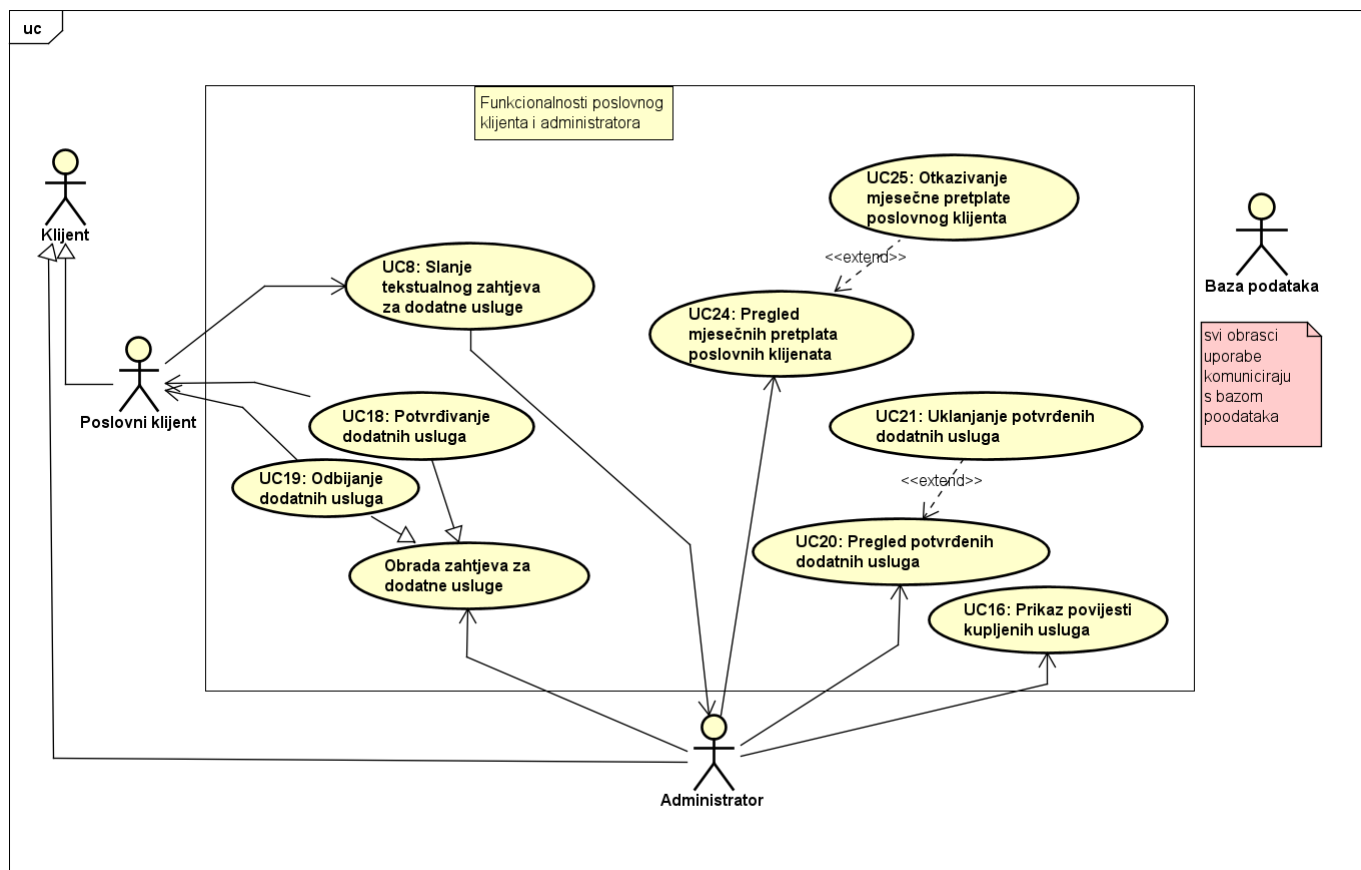
## Dijagrami obrazaca uporabe



Slika 3.1: Dijagram obrazaca uporabe za korisnika i klijenta



Slika 3.2: Dijagram obrazaca uporabe za privatne i poslovne klijente

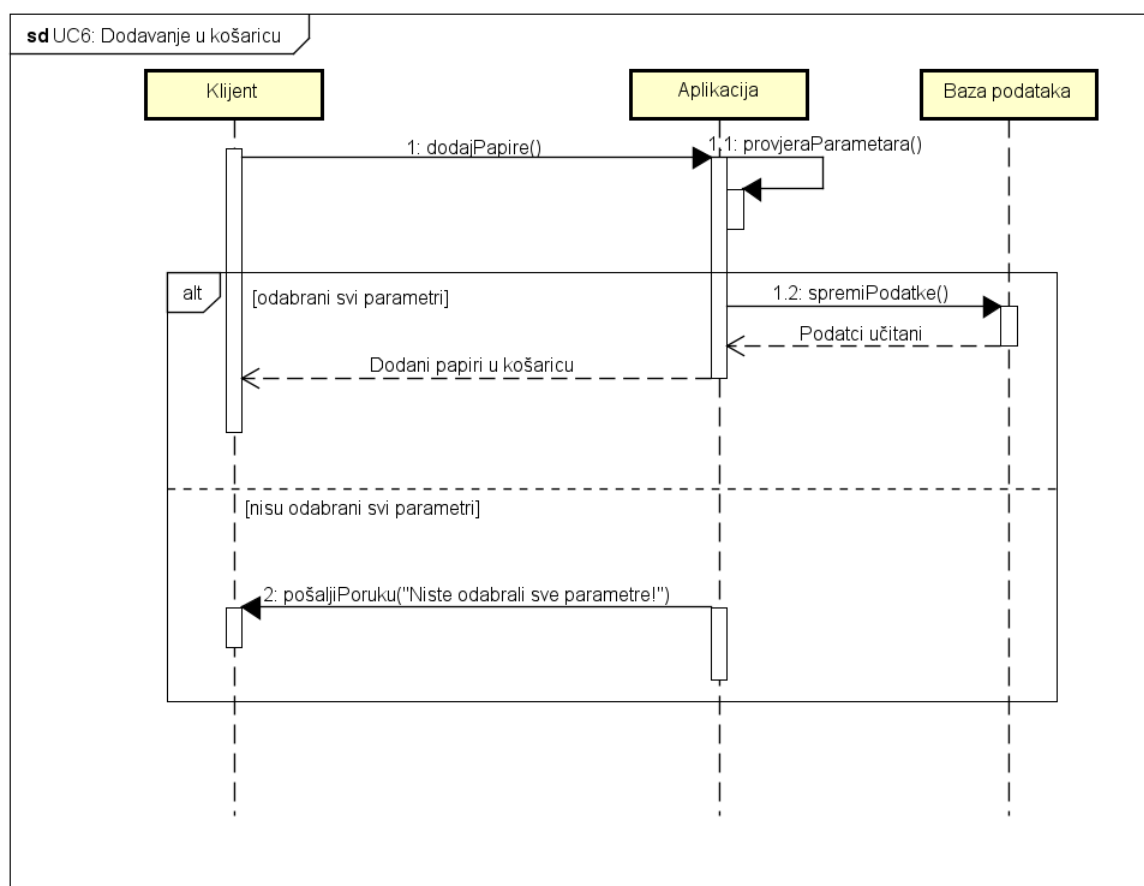


Slika 3.3: Dijagram obrazaca uporabe za poslovne klijente i administratora

### 3.1.2 Sekvencijski dijagrami

#### UC6: Dodavanje u košaricu

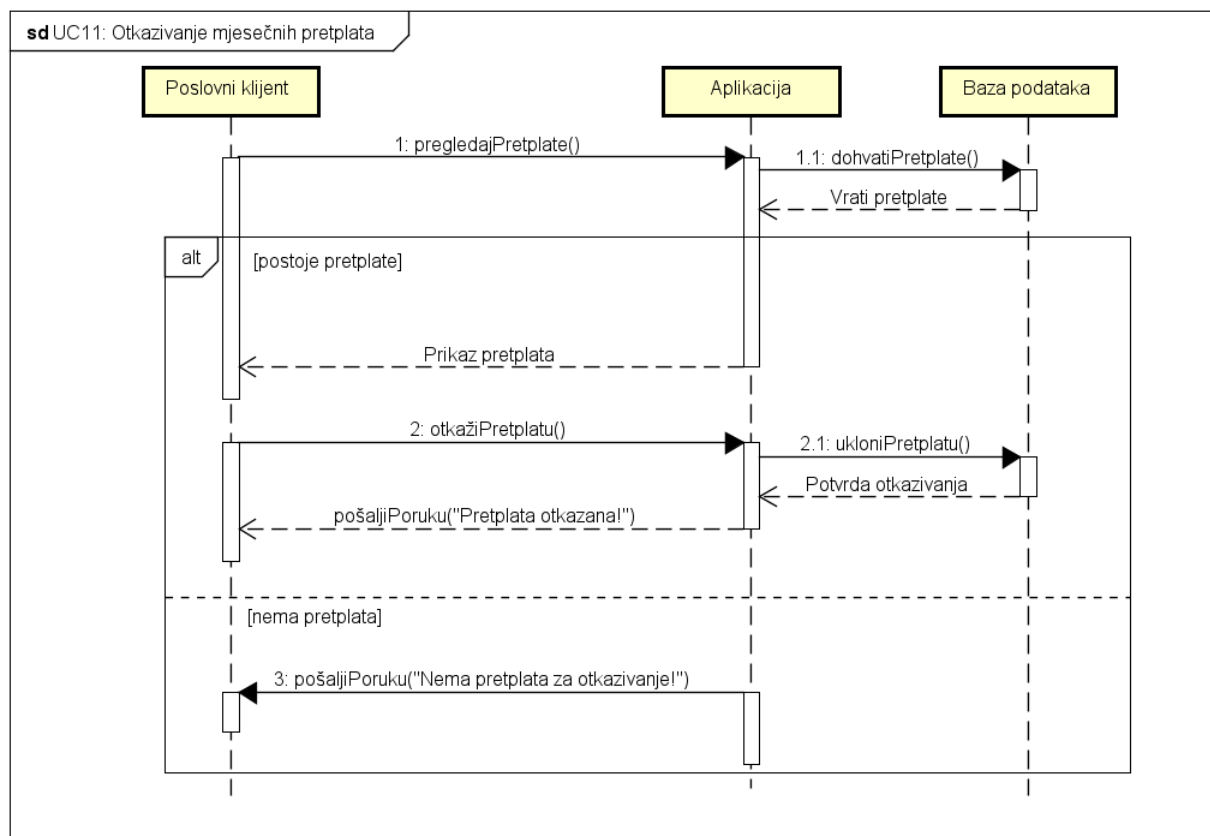
Klijent odabire parametre za željeni papir i stisne na gumb za dodavanje u košaricu. Ako su uneseni parametri ispravni (uneseni su svi i u pravilnom zapisu), onda se papir dodaje u košaricu. U suprotnom klijentu stiže poruka da nije odabrao sve parametre i košarica se ne mijenja.



Slika 3.4: Sekvencijski dijagram za UC6 - Dodavanje u košaricu

#### UC11: Otkazivanje mjesečnih pretplata

Poslovni klijent odabire pregled svojih pretplata. Ako pretplate postoje, one se prikazuju i klijent tada može otkazati bilo koju od njih. Kada se pretplata uspješno otkaže, klijentu dolazi poruka da je pretplata otkazana. U slučaju da klijent nema niti jednu pretplatu, dolazi mu poruka da nema pretplata i stoga ne može ništa otkazati.

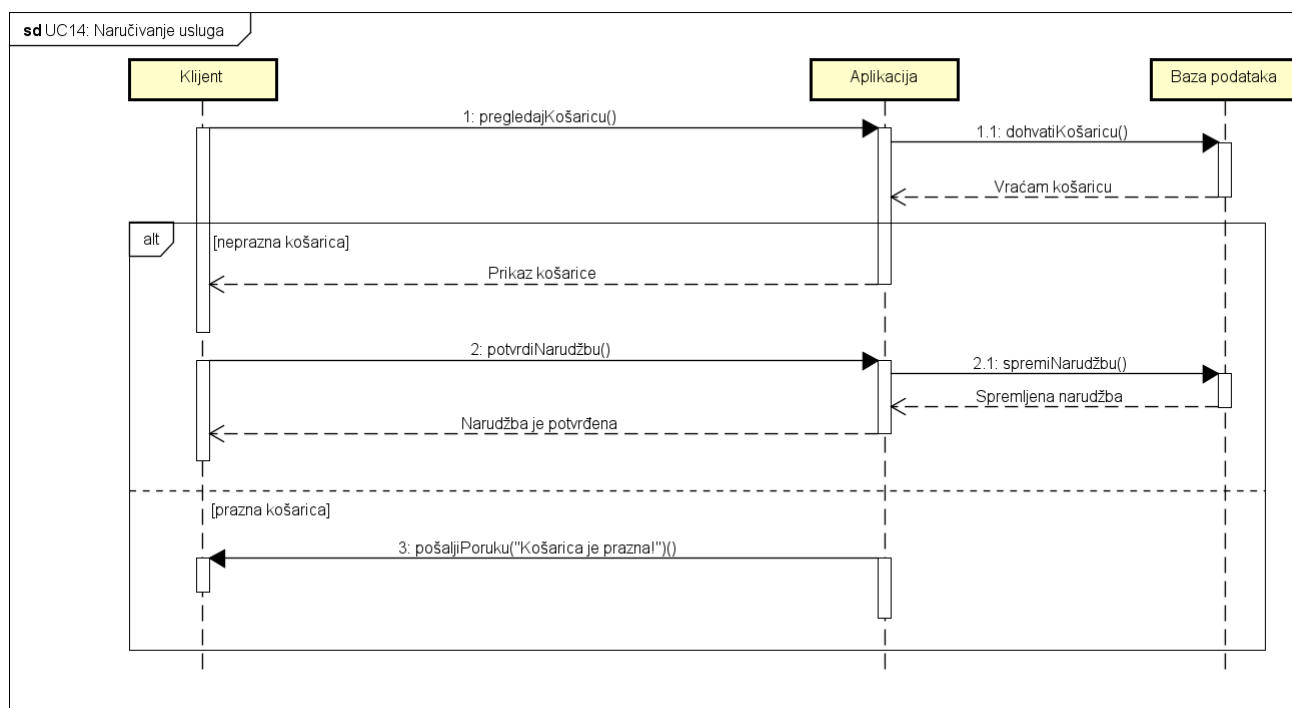


Slika 3.5: Sekvencijski dijagram za UC11 - Otkazivanje mjesečnih pretplata

**UC14: Naručivanje usluga**

Klijent odabire pregled košarice. Ako košarica nije prazna, prikazuje se njezin sadržaj i tada klijent može potvrditi narudžbu. U suprotnom, dolazi mu poruka da je košarica prazna (nema opciju potvrditi narudžbu.)

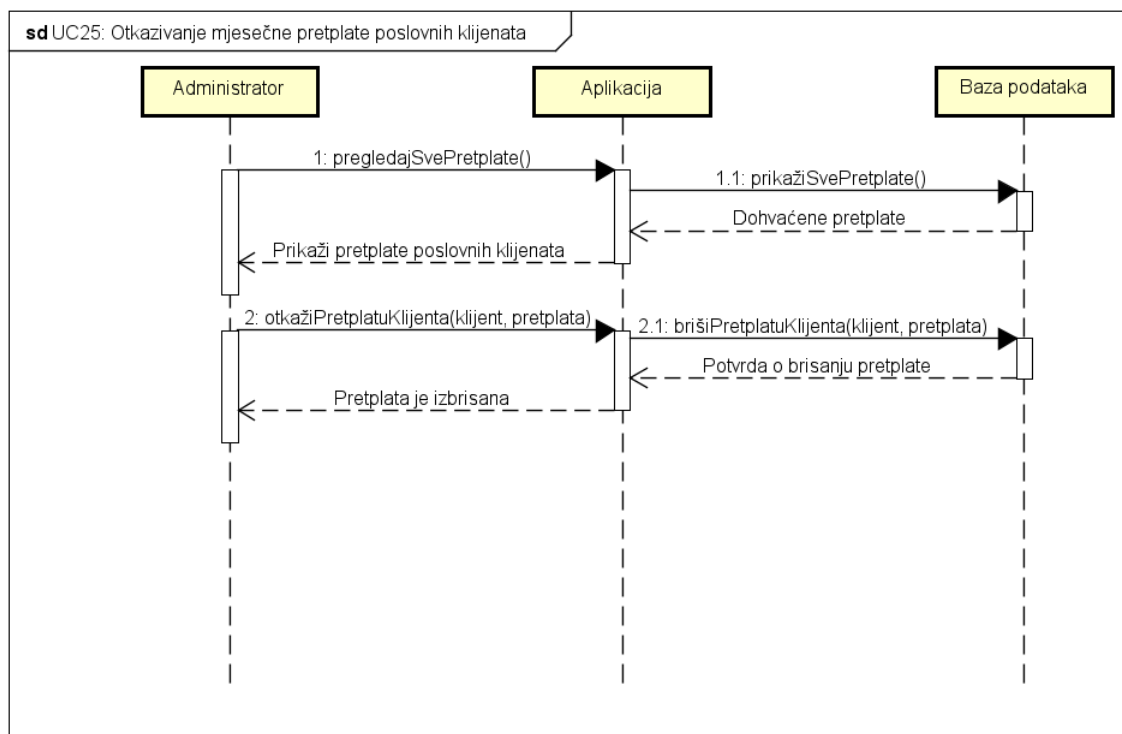




Slika 3.6: Sekvencijski dijagram za UC14 - Naručivanje usluga

**UC25: Otkazivanje mjesečne pretplate poslovnih klijenata**

Administrator odabire prikaz svih pretplata postojećih poslovnih klijenata. Među njima odabire neku pretplatu nekog od njih i otkazuje ju. Time se ta pretplata za tog klijenta briše.



Slika 3.7: Sekvencijski dijagram za UC25 - Otkazivanje mjesečne pretplate poslovnih klijenata

## 3.2 Ostali zahtjevi

1. Pristupanje bazi podataka ne smije trajati predugo (više od nekoliko sekundi)
2. Sustav treba omogućiti rad više korisnika u stvarnom vremenu
3. Nadogradnje sustava ne smiju narušavati funkcionalnosti sustava koje već postoje
4. Sustav će biti implementiran u obliku aplikacije koristeći objektno-orijentirane jezike
5. Aplikacija treba biti jednostavna i intuitivna za korištenje
6. Kao valutu aplikacija koristi hrvatsku kunu - HRK

## 4. Arhitektura i dizajn sustava

Odabrani stil je arhitektura zasnovana na događajima, točnije obrazac MVC (eng. Model-View-Controller) koji je podržan u Spring razvojnom okviru korištenjem anotacija (eng. annotations).

Arhitektura se može podijeliti na tri podsustava:

- Web poslužitelj
- Mobilna aplikacija
- Baza podataka

Mobilna aplikacija je program dizajniran za izvođenje specifičnih funkcija na mobilnom uređaju (npr. pametnom telefonu, tabletu, pametnom satu, itd.). Preko mobilne aplikacije se provodi interakcija sa korisnikom i slanje zatjeva web poslužitelju.

Web poslužitelj je program kojem je zadaća komunikacija sa mobilnom aplikacijom i obrada poslanih zahtjeva. Komunikacija između web poslužitelja i mobilne aplikacije se odvija na World Wide Webu koristeći HTTP (Hypertext Transfer Protocol) protokol za prijenos informacija.

Korišteni programski jezik za izradu poslužiteljske aplikacije je Java sa radnim okvirom (eng. framework) Spring, dok je za izradu mobilne aplikacije korišten Dart.

MVC obrazac je odabran zato što olakšava nezavisan razvoj, ispitivanje i nadogradnju dijelova aplikacije različitih namjena. MVC obrazac se sadrži:

- **Model** - Središnja komponenta sustava. Dinamička struktura podataka aplikacije, neovisna o korisničkom sučelju. Upravlja podacima, logikom i pravilima aplikacije.
- **View (Pogled)** - Bilo kakav prikaz podataka, kao što su graf, dijagram i tablica. Različiti prikazi iste informacije su mogući.
- **Controller (Nadglednik)** - Upravlja i rukuje korisničkom interakcijom sa modelom i pogledom (eng. view). Šalje naloge za korištenje podataka modelu, a pogledu šalje naloge za izmjenu prikaza.

## 4.1 Baza podataka

Koristit ćemo relacijsku bazu podataka koja svojom strukturom zadovoljava naručiteljeve zahtjeve i olakšava korištenje. Gradivna jedinka baze je relacija, odnosno tablica koja je definirana svojim imenom i skupom atributa. Zadaća baze podataka je brza i jednostavna pohrana, izmjena i dohvat podataka za daljnju obradu. Baza podataka ove aplikacije sastoji se od sljedećih entiteta:

- Korisnik
- Tvrtka
- Narudžba
- Redovna usluga
- Dodatna usluga
- Format papira
- Vrsta papira
- Boja papira
- Sadrži dodatnu
- Poruka pretplate
- Osnovna cijena

### 4.1.1 Opis tablica

*Podobljani atributi su primarni ključevi, dok su podcrtani vanjski ključevi.*

#### **Korisnik**

Ovaj entitet sadržava važne informacije o korisniku aplikacije. Sadrži attribute: `korisnik_id`, `email`, `lozinka`, `ime`, `prezime`, `razina_korisnika` i `tvrtka_id`. Ovaj entitet je u vezi 1:0..1 s entitetom `tvrtka` preko atributa `tvrtka_id` tvrtke, u vezi One-to-Many s entitetom `narudžba` preko atributa `korisnik_id` korisnika i u vezi One-to-Many s entitetom `dodatna_usluga` preko atributa `korisnik_id` korisnika.

korisnik		
<b>korisnik_id</b>	SERIAL	jedinstveni identifikator korisnika
email	VARCHAR	email koji se koristi pri prijavi
lozinka	VARCHAR	lozinka koje se koristi i s kojom se uspoređuje pri prijavi
ime	VARCHAR	ime korisnika
prezime	VARCHAR	prezime korisnika
razina_korisnika	INT	razina korisnika 0 - privatni 1 - poslovni 2 - admin
<u>tvrtka_id</u>	INT	id korisnikove tvrtke ako postoji, NULL inače

### Tvrtka

Ovaj entitet sadrži informacije o tvrtci za poslovne korisnike. Sadrži attribute: tvrtka\_id, tvrtka\_oib, broj\_telefona i korisnik\_id. Ovaj entitet je u vezi One-to-One sa entitetom korisnik preko korisnik\_id korisnika, te 0..1:1 sa entitetom korisnik preko tvrtka\_id tvrtke.

tvrtka		
<b>tvrtka_id</b>	SERIAL	jedinstveni identifikator tvrtke
tvrtka_oib	VARCHAR	OIB tvrtke
ime_tvrtke	VARCHAR	ime tvrtke
broj_telefona	VARCHAR	broj telefona kontakt osobe tvrtke
<u>korisnik_id</u>	INT	korisnički id tvrtke

### Narudžba

Ovaj entitet opisuje osnovne informacije o narudžbi korisnika. Sadrži attribute: narudzba\_id, korisnik\_id, cijena i datum. Ovaj entitet je u vezi Many-to-One s entitetom korisnik preko atributa korisnik\_id korisnika, u vezi One-to-Many s entitetom sadrziDodatnu preko atributa narudzba\_id i korisnik\_id narudzbe, u vezi One-to-Many s entitetom redovna\_usluga preko atributa narudzba\_id i korisnik\_id narudzbe i u vezi One-to-Many s entitetom porukaPretplate preko atributa narudzba\_id i korisnik\_id narudzbe.

narudzba		
<b>narudzba_id</b>	SERIAL	jedinstveni identifikator narudžbe
<b><u>korisnik_id</u></b>	INT	jedinstveni identifikator korisnika naručitelja
cijena	NUMERIC	cijena narudžbe
datum	TIMESTAMP	datum i vrijeme provedbe narudžbe

### Dodatna usluga

Ovaj entitet opisuje sve informacije o dodatnoj usluzi za pojedinog korisnika. Sadrži attribute: dodatna\_usluga\_id, korisnik\_id, dodatna\_naziv, dodatna\_opis, cijena, kontakt\_ime, kontakt\_broj, kontakt\_email, potvrdena i pretplata. Entitet je u vezi Many-to-One s entitetom korisnik preko atributa korisnik\_id korisnika i u vezi One-to-Many s entitetom dodatna preko atributa dodatna\_usluga\_id i korisnik\_id dodatne usluge.

dodatna_usluga		
<b>dodatna_usluga_id</b>	SERIAL	jedinstveni identifikator dodatne usluge
<b><u>korisnik_id</u></b>	INT	jedinstveni identifikator poslovnog korisnika kojem je usluga dodijeljena
dodatna_naziv	VARCHAR	kratki naziv dodatne usluge
dodatna_opis	VARCHAR	opis dodatne usluge
cijena	NUMERIC	cijena dodatne usluge
kontakt_ime	VARCHAR	ime kontakt osobe kompanije
kontakt_broj	VARCHAR	broj kontakt osobe kompanije
kontakt_email	VARCHAR	email kontakt osobe kompanije
potvrdena	BOOLEAN	označava je li zahtjev potvrđen ili nije, ako je true onda je prihvaćen, ako je false onda administrator još nije donio odluku. U slučaju da je odbijen onda entitet uklanja iz tablice
pretplata	BOOLEAN	označava je li narudžba mjesečna pretplata

### Sadrži dodatnu

Entitet dodatna sadrži informaciju koju/koje dodatne usluge sadrži narudžba.

Sadrži attribute: `narudzba_id`, `korisnik_id`, `dodatna_usluga_id` i `pretplata`. Entitet sadrži vezu Many-to-One s entitetom `narudzba` preko atributa `narudzba_id` i `korisnik_id` narudžbe i vezu Many-to-One s entitetom `dodatna_usluga` preko atributa `dodatna_usluga_id` i `korisnik_id` dodatne usluge.

sadrziDodatnu		
<u><code>narudzba_id</code></u>	INT	jedinstveni identifikator narudžbe
<u><code>korisnik_id</code></u>	INT	jedinstveni identifikator korisnika naručitelja
<u><code>dodatna_usluga_id</code></u>	INT	jedinstveni identifikator dodatne usluge
<code>pretplata</code>	BOOLEAN	označava je li narudžba mjesečna pretplata

### Redovna usluga

Ovaj entitet opisuje redovnu uslugu korisnika. Sadrži entitete: `narudzba_id`, `korisnik_id`, `visina`, `sirina`, `boja`, `tezina`, `vodeni_zig`, `pretplata`, `format_id`, `vrsta_id` i `boja_id`. Entitet je u vezi Many-to-One s entitetom `narudzba` preko atributa `narudzba_id` i `korisnik_id` narudžbe, u vezi Many-to-One sa `format_papira` preko atributa `format_id` formata papira, u vezi Many-to-One sa `vrsta_papira` preko atributa `vrsta_id` vrste papira, u vezi Many-to-One sa `boja_papira` preko atributa `boja_id` boje papira,

redovna_usluga		
<u><code>narudzba_id</code></u>	SERIAL	jedinstveni identifikator narudžbe
<u><code>korisnik_id</code></u>	INT	jedinstveni identifikator korisnika naručitelja
<code>visina</code>	INT	visina papira u mm
<code>sirina</code>	INT	širina papira u mm
<code>boja</code>	VARCHAR	heksadecimalni zapis boje papira
<code>tezina</code>	VARCHAR	težina papira u gsm
<code>vodeni_zig</code>	BLOB	slika vodenog žiga
<code>pretplata</code>	BOOLEAN	označava je li narudžba mjesečna pretplata
<u><code>format_id</code></u>	INT	jedinstveni identifikator formata papira, 0 je rezerviran za nekonvencionalne formate
<u><code>vrsta_id</code></u>	INT	jedinstveni identifikator vrste papira
<u><code>boja_id</code></u>	INT	jedinstveni identifikator boje papira



redovna_usluga		
		0 je rezerviran za proizvoljne boje

### Format papira

Ovaj entitet sadrži sve informacije o formatu papira. Sadrži attribute format\_id, format\_ime, visina, sirina i koeficijent\_cijena. Entitet je u vezi One-to-Many s entitetom redovna\_usluga preko atributa format\_id formata papira.

format_papira		
format_id	SERIAL	jedinstveni identifikator formata papira, 0 je rezerviran za nekonvencionalne formate
format_ime	VARCHAR	ime formata papira
visina	INT	visina formata papira u mm
širina	INT	širina formata papira u mm
koeficijent_cijena	NUMERIC	cijena narudžbe

### Vrste papira

Ovaj entitet sadrži sve informacije o vrsti papira. Sadrži attribute vrsta\_id, vrsta\_ime i koeficijent\_cijena. Entitet je u vezi One-to-Many s entitetom redovna\_usluga preko atributa vrsta\_id vrste papira.

vrste_papira		
vrsta_id	SERIAL	jedinstveni identifikator vrste papira
vrsta_ime	VARCHAR	ime vrste papira
koeficijent_cijena	NUMERIC	cijena narudžbe

### Boja papira

Ovaj entitet sadrži sve informacije o boji papira. Sadrži attribute boja\_id, boja\_ime, hexcode i koeficijent\_cijena. Entitet je u vezi One-to-Many s entitetom redovna\_usluga preko atributa boja\_id boje papira.

boja_papira		
boja_id	SERIAL	jedinstveni identifikator boje papira 0 je rezerviran za proizvoljne boje
ime_boja	VARCHAR	ime boje papira

boja_papira		
koeficijent_cijena	NUMERIC	cijena narudžbe
hexcode	VARCHAR	heksadecimalni zapis boje papira

### Poruka pretplate

Ovaj entitet sadrži osnovne informacije o porukama pretplate. Entitet se svakih 24h treba ažurirati i dodati unos za korisnika i narudžbu sa pretplatom koja je naručena na ovaj dan u mjesecu. Nakon što je korisnik primio redak on se treba obrisati, te nepojavljivati nadalje. Narudžba\_id je vanjski ključ na narudžba\_id narudžbe, kako bi znali da se pretplate, redovne\_usluge ili dodatne\_usluge nalaze u nekoj narudžbi. Korisnik\_id je vanjski ključ za korisnik\_id korisnika i preko njega jednostavno znamo kojemu korisniku treba biti prikazana poruka. Sadrži attribute: id, narudžba\_id i korisnik\_id. Entitet je u vezi Many-to-One s entitetom narudzba preko atributa narudzba\_id i korisnik\_id narudžbe.

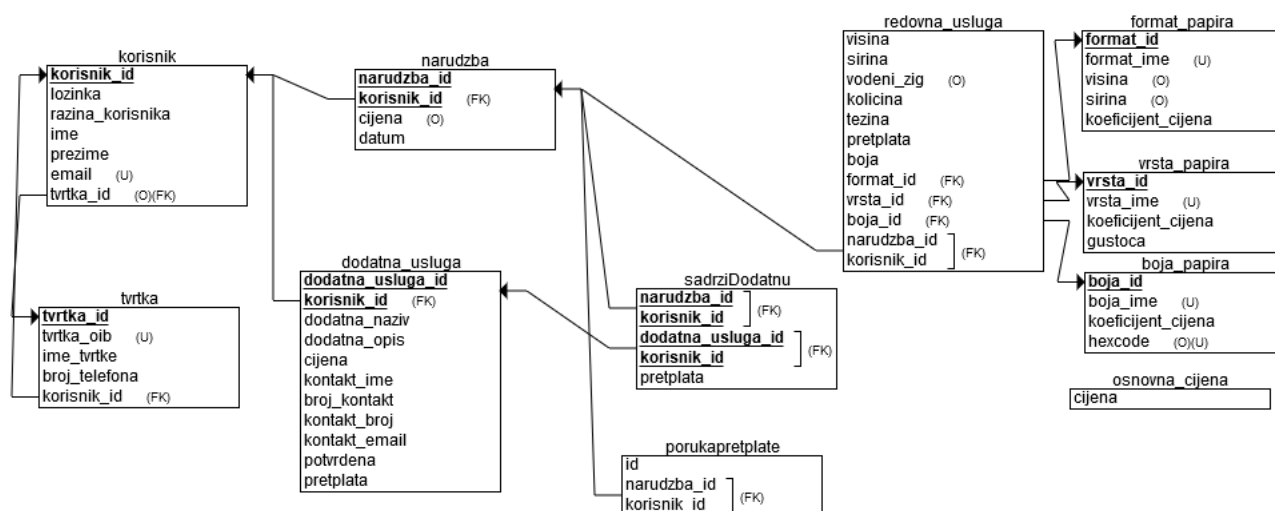
porukaPretplate		
<b>id</b>	SERIAL	jedinstveni identifikator mjesečne poruke o provedbi pretplate
<u>narudzba_id</u>	INT	jedinstveni identifikator narudžbe
<u>korisnik_id</u>	INT	jedinstveni identifikator korisnika

### Osnovna cijena

Ovaj entitet sadrži informaciju o osnovnoj cijeni proizvoda. Sadrži atribut: cijena. Tablica ima samo jedan unos.

osnovna_cijena		
cijena	NUMERIC	osnovna cijena proizvoda koja se množi sa koeficijentima cijena kako bi se dobila cijena neke redovne usluge

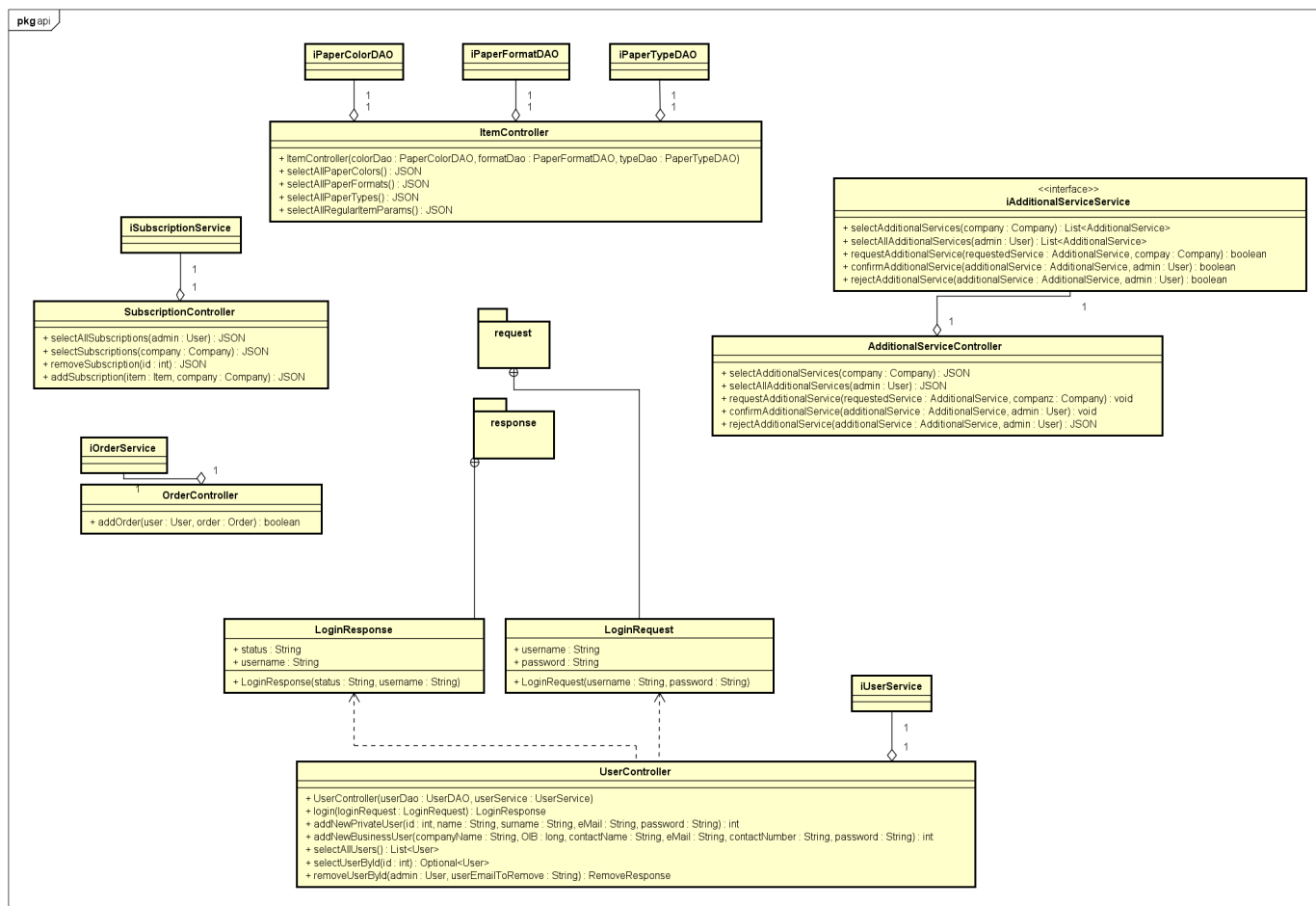
### 4.1.2 Dijagram baze podataka



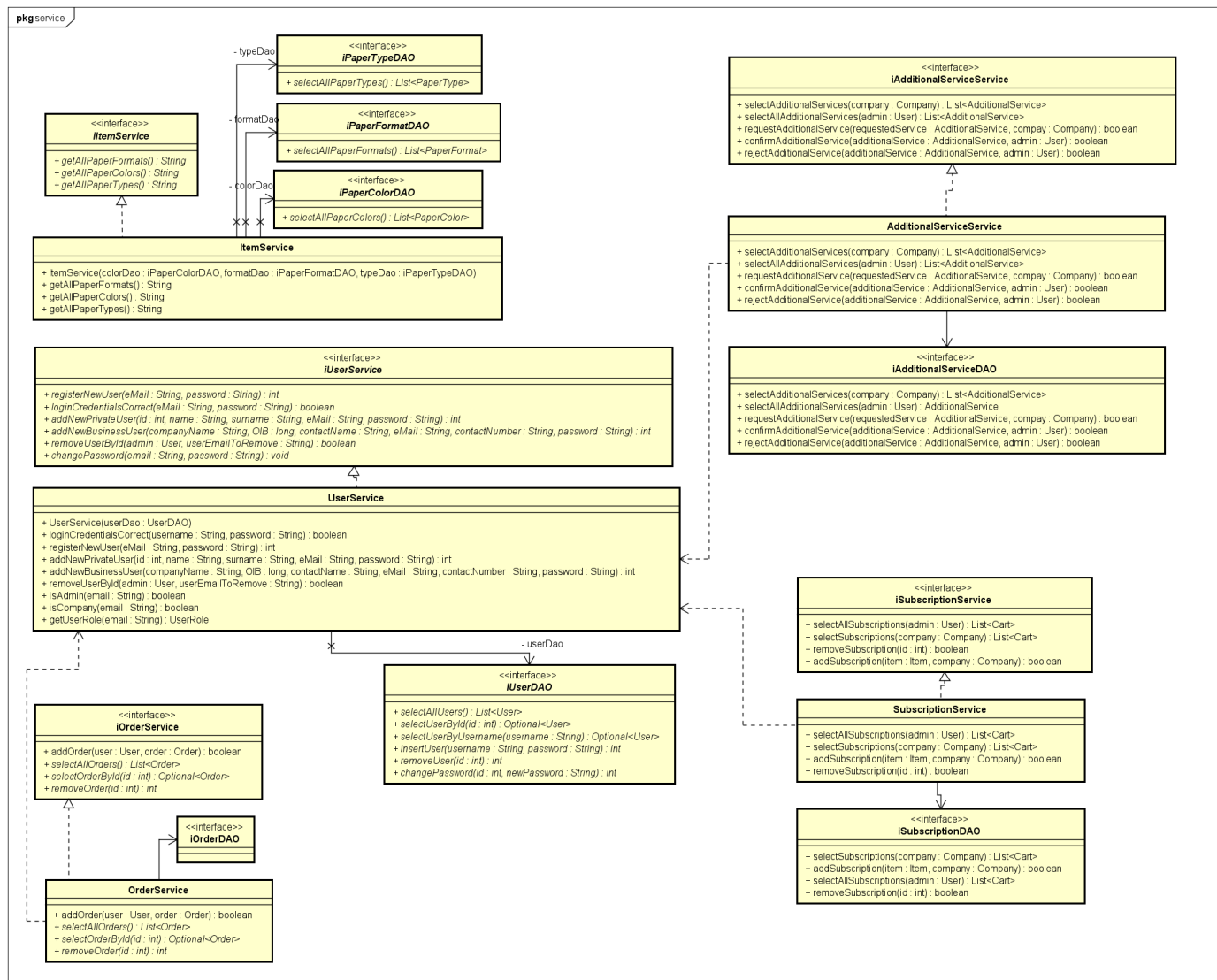
Slika 4.1: ER dijagram baze podataka

## 4.2 Dijagram razreda

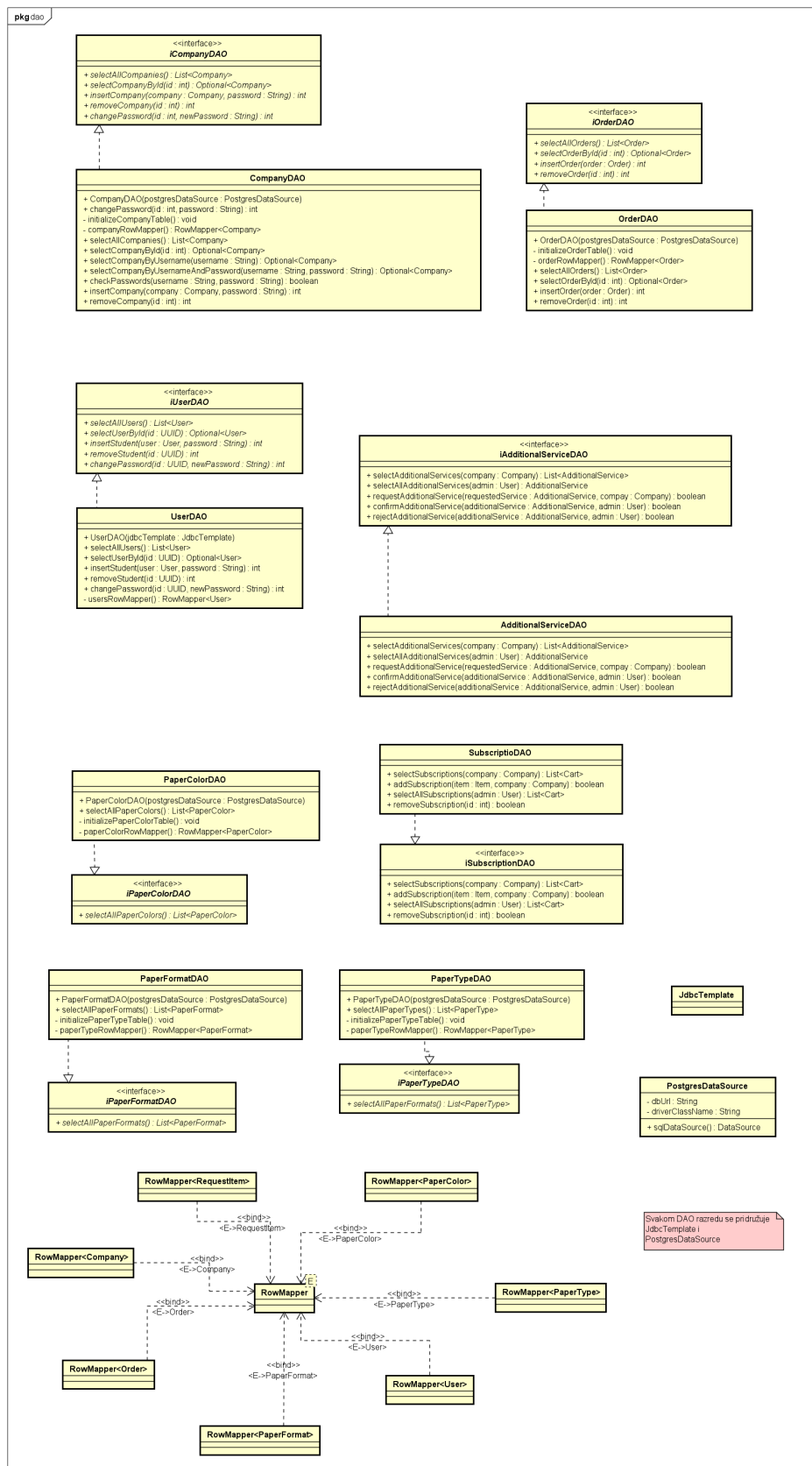
Prikazani razredi se odnose na backend MVC arhitekturu. Dijagram razreda je podijeljen po ulogama na 3 podskupine, controlleri, servisi i modeli. Controlleri primaju određene HTTP zahtjeve te pozivaju potrebne servise koji te zahtjeve dalje obrađuju koristeći potrebne DAO-e (eng. Data Access Object) za komunikaciju sa bazom podataka. Razredi koji se manipuliraju u programu se nalaze u modelu.



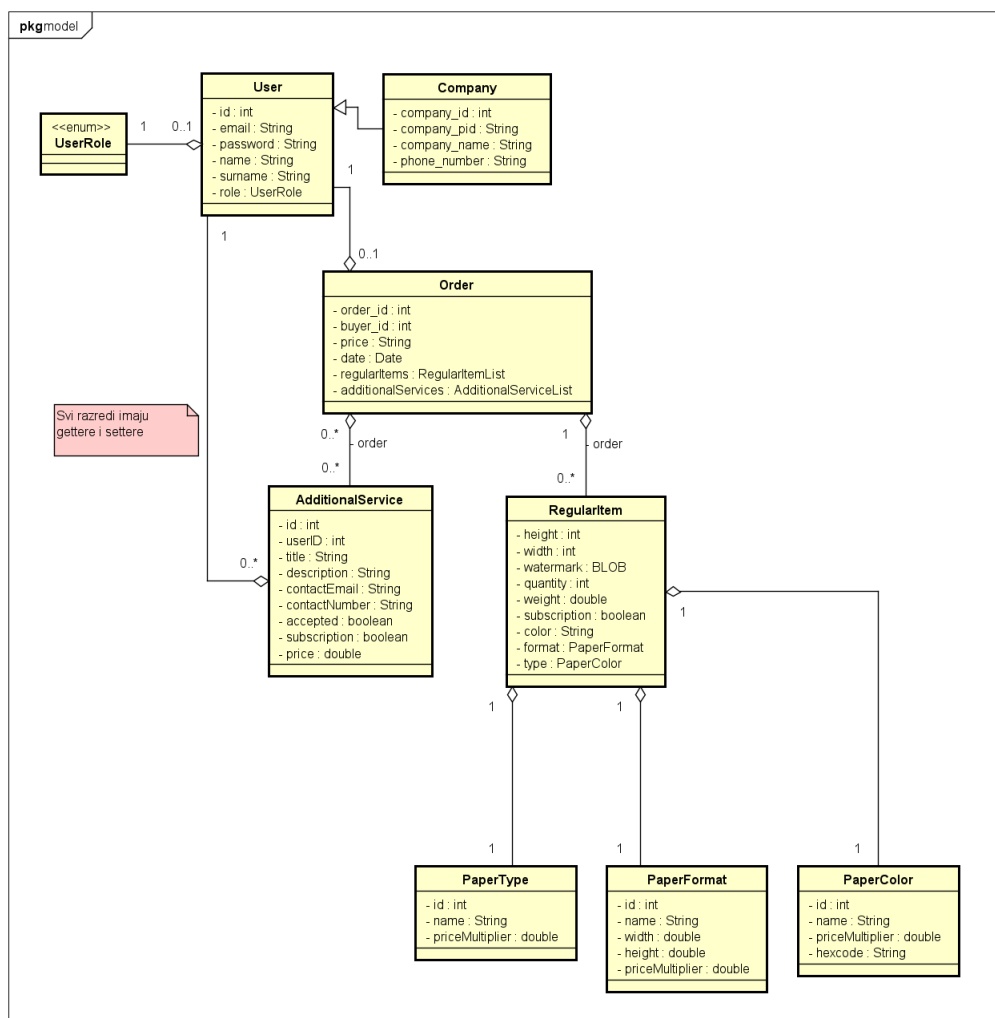
Slika 4.2: Dijagram razreda - controlleri



Slika 4.3: Dijagram razreda - servisi



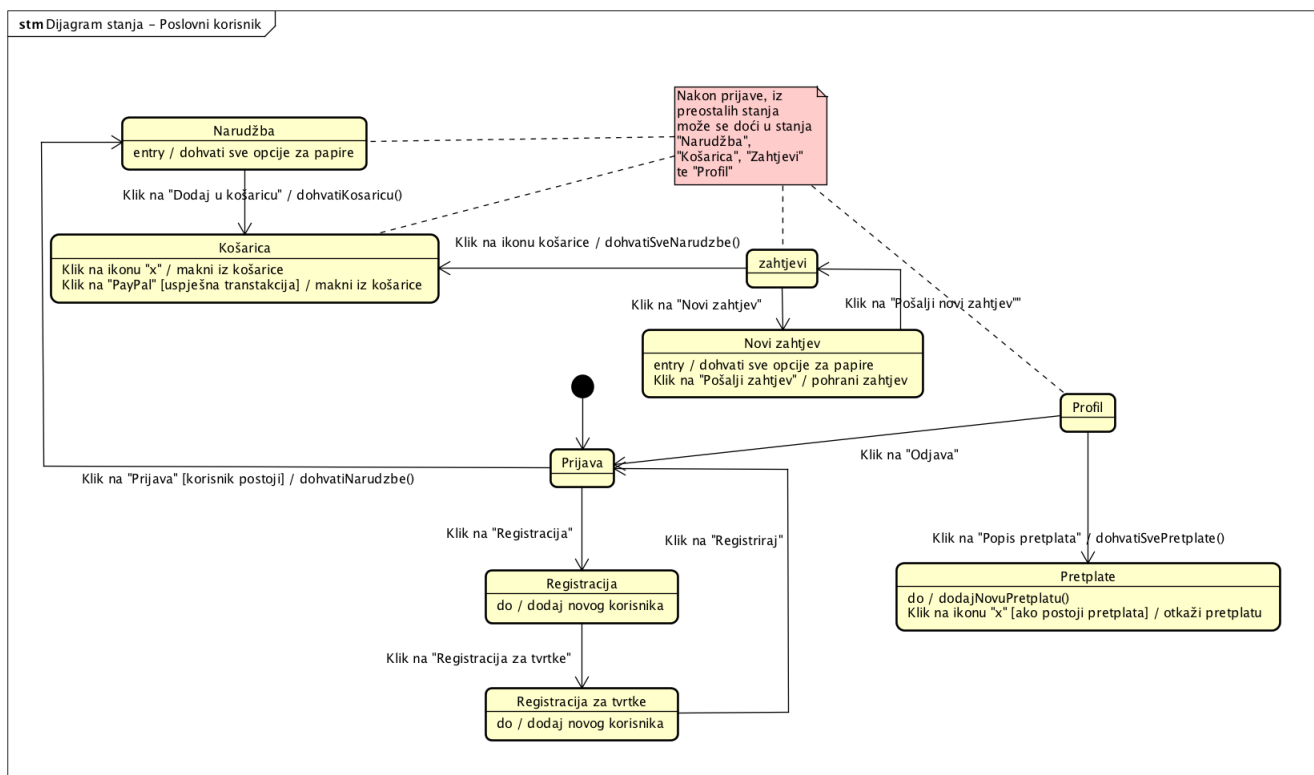
Slika 4.4: Dijagram razreda - DAO-i



Slika 4.5: Dijagram razreda - modeli

### 4.3 Dijagram stanja

Dijagram stanja prikazuje stanja objekta te prijelaze iz jednog stanja u drugo temeljene na događajima. Na slici je prikazan dijagram stanja za poslovnog korisnika, registriranog ili ne. Nakon prijave ili registracije, klijentu se odmah prikazuje stranica "Narudžbe" na kojoj može vidjeti svoje dosadašnje narudžbe, stvoriti nove te ih dodati u košaricu. U svakom trenutku klijent se može pozicionirati u svoj "Profil", u "Zahtjeve" ili u navedene "Narudžbe" i "Košaricu". Ukoliko se pozicionira u "Zahtjeve", tamo može pregledavati status postojećih zahtjeva ili stvoriti novi zahtjev klikom na "Novi zahtjev", te iste može dodati u košaricu klikom na ikonicu košarice što ga ponovo prebacuje na "Košaricu". Ukoliko se pozicionira u "Profil", od tamo može pristupiti svojim pretplatama klikom na "Popis pretplata", gdje ih zatim može dodavati i brisati. Iz "Profila" se također može odjaviti što ga prebacuje na početnu stranicu "Prijave".



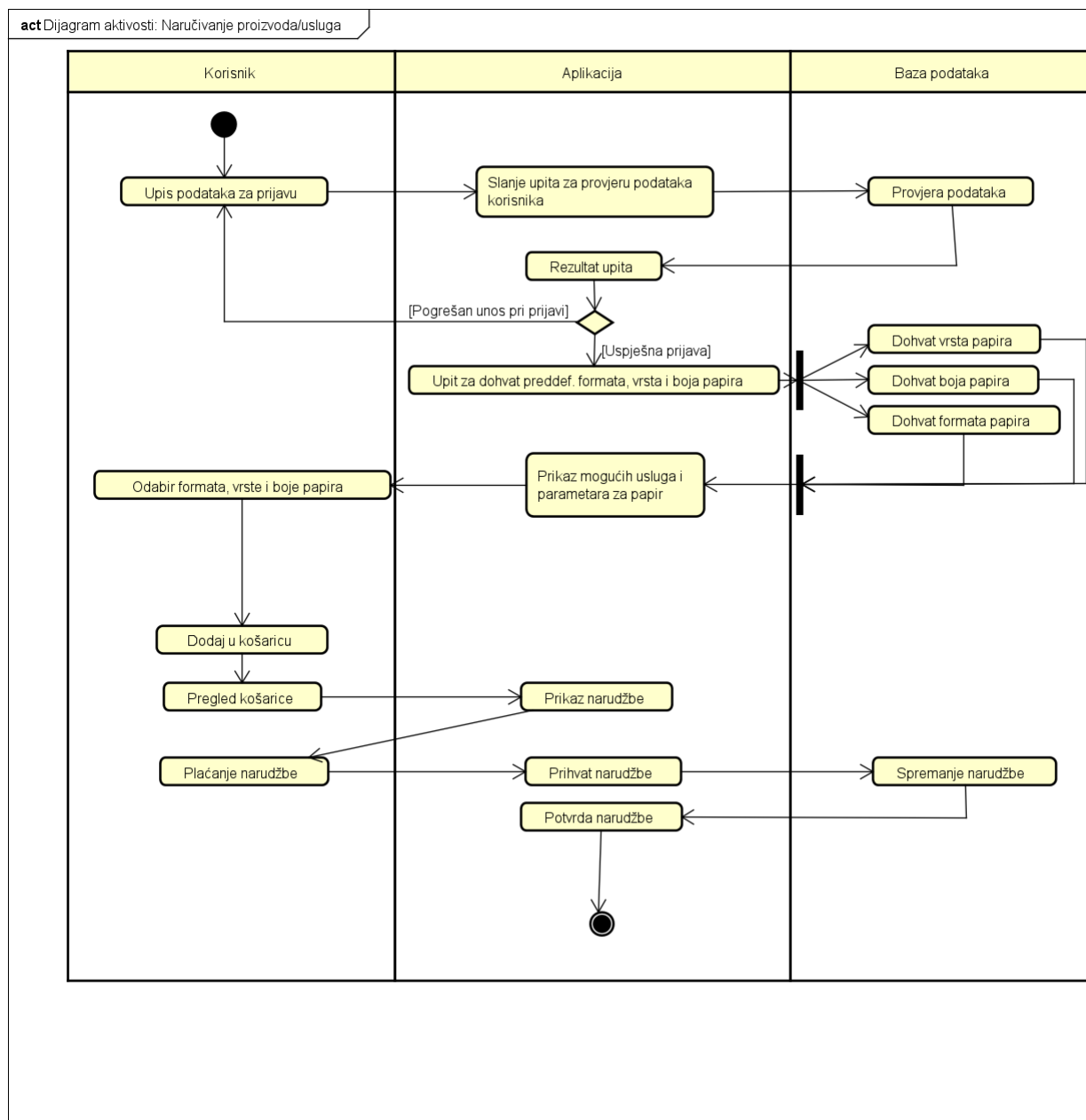
Slika 4.6: Dijagram stanja



## 4.4 Dijagram aktivnosti

Dijagram aktivnosti koristi se za prikaz poslovnih procesa te upravljačkog i podatkovnog toka. On je prigodan za opis sinkronizacije i konkurentnosti, no ne i za modeliranje događajima poticanog ponašanja.

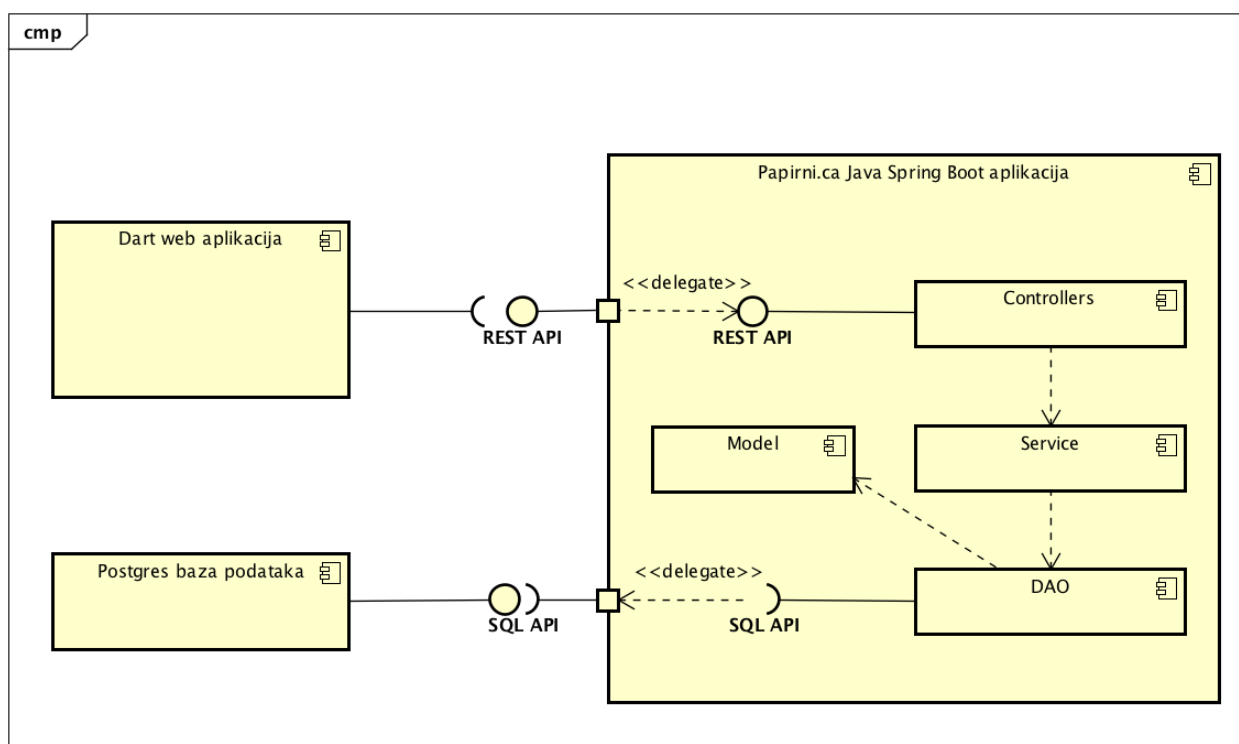
Sljedeći dijagram prikazuje proces kreiranja narudžbe u slučaju privatnog korisnika. Korisnik se prijavi u sustav, odabire što želi naručiti te potom, nakon što izabere sve što mu je potrebno, odabire pregled košarice i plaća narudžbu te se ona naplaćuje.



Slika 4.7: Dijagram aktivnosti

## 4.5 Dijagram komponenti

Dijagram komponenti prikazan na slici opisuje organizaciju i međuovisnost komponenti, interne strukture i odnose prema okolini. Sustavu se pristupa preko sučelja. Klijent pristupa web aplikaciji koristeći web preglednik i sučelje REST API. Web aplikacija organizirana je modularno prema pravilima radnog okvira Java Spring Boota. Modul DAO je zadužen za dohvat i obradu podatka iz baze preko sučelja SQL API te povezivanje istih s Modelom predstavlja klase sustava. Modul Service zatim te podatke obrađuje na aplikaciji svojstven način, a modul Controllere je zadužen za komunikaciju HTTPS protokolom sa Klijentom.



Slika 4.8: Dijagram komponenti

## 5. Implementacija i korisničko sučelje

### 5.1 Korištene tehnologije i alati

Komunikacija među članovima tima najvećim je dijelom ostvarena putem platforme Discord,<sup>1</sup> dok su pojedini kraći sastanci nakon konzultacija/demonstracija održani na platformi MS Teams.<sup>2</sup>

Za dizajn izgleda aplikacije korišten je software Adobe XD,<sup>3</sup> a za izradu UML dijagrama Astah.<sup>4</sup> Git<sup>5</sup> je služio kao sustav za upravljanje izvornim kodom na projektu čiji je udaljeni repozitorij smješten na web platformi GitLab.<sup>6</sup>

Uređivanje dokumentacije odvijalo se u okruženju TeXstudio,<sup>7</sup> a jedan od uređivača za pisanje programa za samu aplikaciju bio je Visual Studio Code.<sup>8</sup>

Za razvoj *frontenda* korišteni su Dart<sup>9</sup> i Flutter.<sup>10</sup> Dart je programski jezik optimiziran za izgradnju UI-a s posebnim dodacima kao što su *spread operator* i *collection if* koji omogućavaju lakšu prilagodbu UI-a nekoj platformi, pritom imajući poznatu sintaksu koju je lako svladati. Flutter je Google-ov UI alat za programiranje u Dartu koji se hvali mogućnošću brzog razvoja iznimno vizualno privlačnih aplikacija koje mogu biti mobilne, web ili desktop.

Za razvoj *backenda* korišten je Java Spring framework<sup>11</sup> (radni okvir). Spring je brz, fleksibilan i siguran okvir za rad s DAO, a poznat je po MVC obrascu korištenjem anotacija.

Baza podataka je relacijska baza podataka napravljena putem PostgreSQL<sup>12</sup>. To je sistem baze podataka s kojim smo upoznati tijekom studija na FER-u, a poznat je po dobrim performansama, pouzdanosti i robusnim značajkama.

---

<sup>1</sup><https://discord.com/>

<sup>2</sup><https://www.microsoft.com/hr-hr/microsoft-365/microsoft-teams/group-chat-software/>

<sup>3</sup><https://www.adobe.com/products/xd.html>

<sup>4</sup><https://astah.net/>

<sup>5</sup><https://git-scm.com/>

<sup>6</sup><https://about.gitlab.com/>

<sup>7</sup><https://www.textstudio.org/>

<sup>8</sup><https://code.visualstudio.com/>

<sup>9</sup><https://dart.dev/>

<sup>10</sup><https://flutter.dev/>

<sup>11</sup><https://spring.io/>

<sup>12</sup><https://www.postgresql.org/>

Heroku<sup>13</sup> je korišten za puštanje aplikacije u pogon. Heroku je *cloud Platform as a Service (PaaS)* koji se bazira na kontejnerima. Podržava mnoštvo programskih jezika i omogućuje integraciju s GitLabom. *PaaS* je platforma koja omogućuje razvoj, pokretanje i upravljanje aplikacijama uz zaobilaženje kompleksnosti koju donosi izrada infrastrukture tipično povezane s izgradnjom aplikacije. Time je povećana učinkovitost i olakšano održavanje i unaprjeđenje aplikacije.

---

<sup>13</sup><https://www.heroku.com/>

## 5.2 Ispitivanje programskog rješenja

*Nisu provedeni svi traženi testovi.*

### 5.2.1 Ispitivanje komponenti

Komponente su ispitane sljedećim unit testovima.

```
@SpringBootTest
class FooBarApplicationTests {

    PostgreSQLDataSource pgd = new PostgreSQLDataSource();

    @Test
    void testAdminUserRole() {
        UserDao userDao = new UserDao(pgd);
        User u = userDao.selectUserByEmail("admin@admin.hr").get();

        assertEquals(UserRole.ADMIN, u.getRole());
    }

    @Test
    void testPrivateUserCantBuyCustomOrders() {
        UserDao userDao = new UserDao(pgd);
        OrderDao oDao = new OrderDao(pgd);
        User u = userDao.selectUserByEmail("progi@fer.hr").get();

        List<Regular> reg = new ArrayList<>();
        reg.add(new Regular(10, 1, "poseban", 1, 10, 10, null, 10, false, "#010101", "CUSTOM", "paus", 42, 1.2, 2));

        assertEquals(UserRole.PRIVATE, u.getRole());
        assertEquals("fail", oDao.transformCart(new CartRequest(u.getId(), 10, reg, null)).message);
    }

    @Test
    void testPrivateUserCantAddAdditionalOrders() {
        UserDao userDao = new UserDao(pgd);
        OrderDao oDao = new OrderDao(pgd);
        User u = userDao.selectUserByEmail("progi@fer.hr").get();
        List<Additional> addit = new ArrayList<>();
        addit.add(new Additional(u.getId(), "dodatna", "desc", "kontakt", "mail", "123"));

        assertEquals(UserRole.PRIVATE, u.getRole());
        assertEquals("fail", oDao.transformCart(new CartRequest(u.getId(), 10, null, addit)).message);
    }

    @Test
    void testLoginSuccess() {
        UserService us = new UserService(new UserDao(pgd), new CompanyDao(pgd));
        UserController uc = new UserController(us, null, null);
        LoginResponse lr = uc.login(new LoginRequest("mail@mail", "pass"));

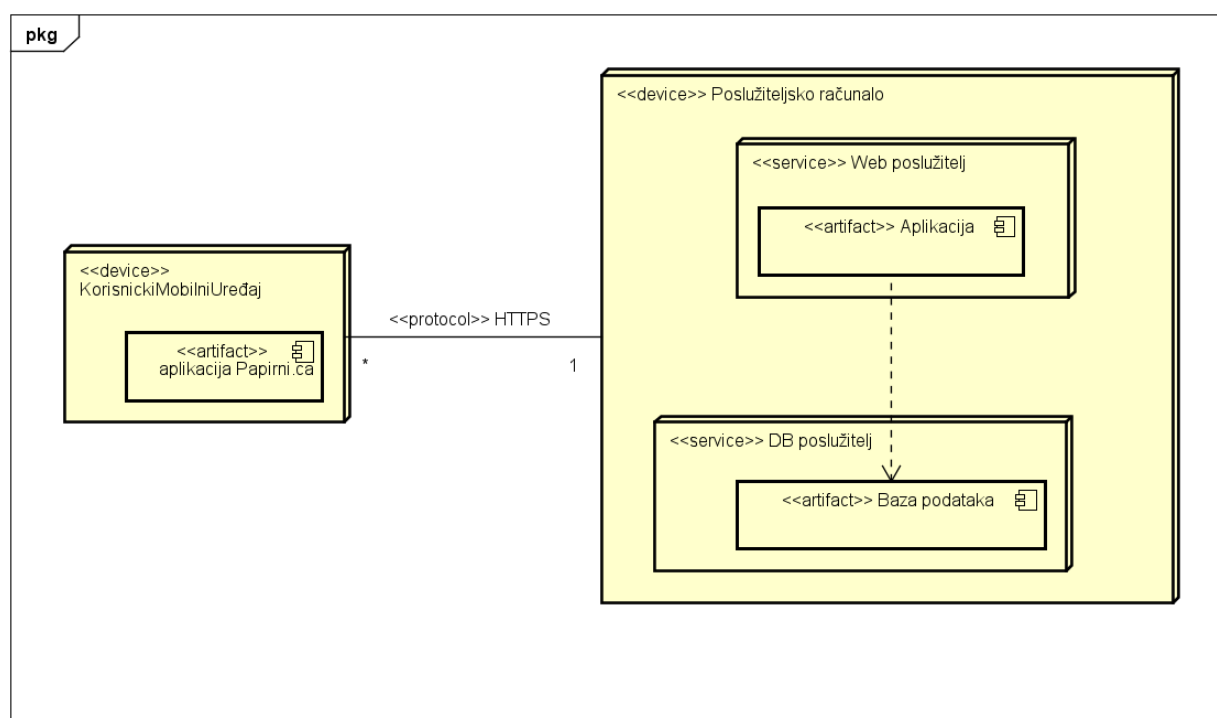
        assertEquals("Imenko", lr.firstName);
        assertEquals("Prezimenko", lr.lastName);
        assertEquals(UserRole.PRIVATE.toString(), lr.userId);
        assertEquals("mail@mail", lr.email);
        assertEquals("success", lr.status);
    }
}
```

Slika 5.1: Unit testovi

### 5.2.2 Ispitivanje sustava

## 5.3 Dijagram razmještaja

Dijagram razmještaja pruža uvid u topologiju sklopovlja i programsku potporu koja se koristi u implementaciji sustava. Klijenti koriste mobilnu aplikaciju kako bi pristupili samoj aplikaciji. Komunikacija između klijenta i poslužitelja bazira se na HTTPS protokolu dok se na poslužiteljskom računalu nalaze web poslužitelj i poslužitelj baze podataka.



Slika 5.2: Dijagram razmještaja

## 5.4 Upute za puštanje u pogon

### Izrada Heroku profila

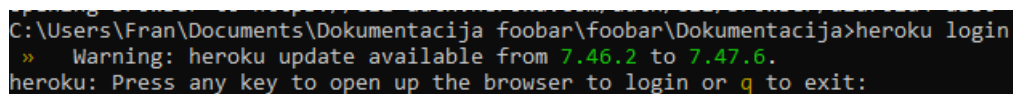
Potrebno je napraviti profil na službenim stranicama Herokua.

### Instalacija Heroku CLI-a

Sljedeći korak je sa službene stranice Herokua skinuti Heroku CLI i provesti klasičnu instalaciju.

### Namještanje i stvaranje Heroku aplikacije

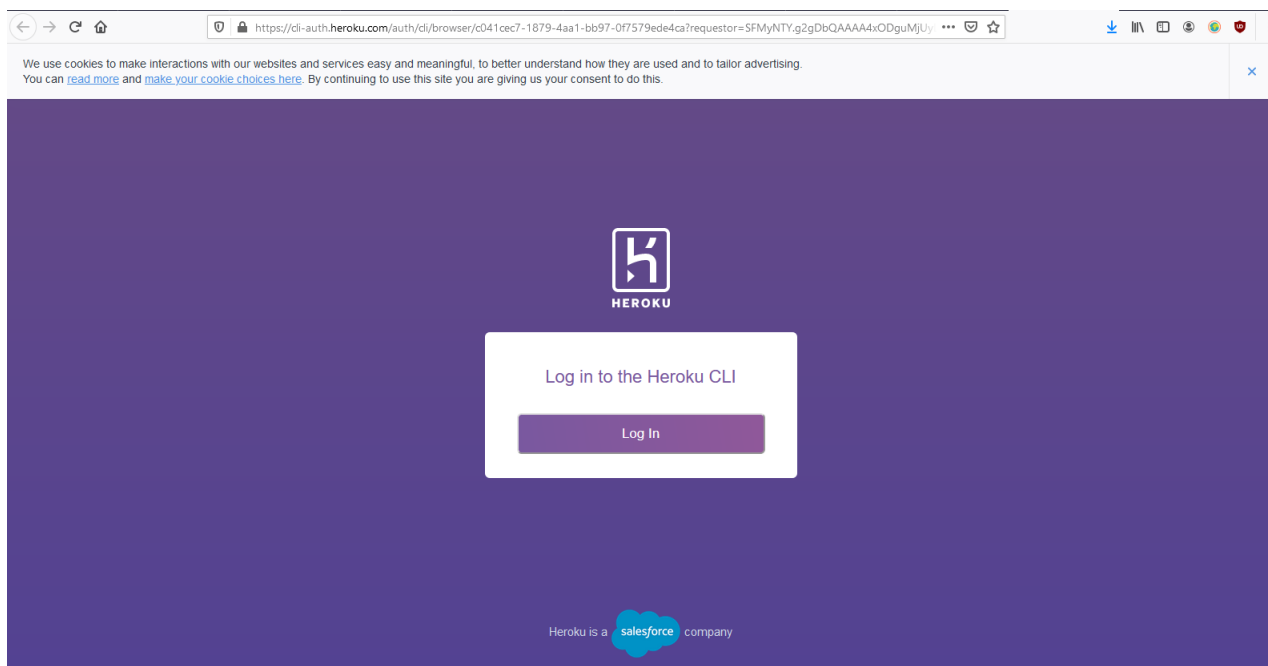
Zatim je potrebno otvoriti komandnu liniju i pozicionirati se u direktorij u kojemu će se nalaziti željeni projekt. Pokrenuti naredbu "heroku login" te na upit za otvaranje preglednika kako bi se proveo login stisnuti "q" ako ne želite provesti registraciju, a ako želite stisnite bilo koju preostalu tipku na tipkovnicu. U potvrdom slučaju se otvara preglednik te je nužno stisnuti "Log In" i prijaviti se. Kako bi se aplikacija stvorila potrebno je u komandnu liniju upisati "git init" pa "heroku create" te se nova aplikacija vašeg profila stvara i ispisuje njeno ime.



```
C:\Users\Fran\Documents\Dokumentacija foobar\foobar\Dokumentacija>heroku login
» Warning: heroku update available from 7.46.2 to 7.47.6.
heroku: Press any key to open up the browser to login or q to exit:
```

Slika 5.3: Upis naredbe za prijavu u Heroku CLI te upit





Slika 5.4: Otvoreni preglednik za prijavu na Heroku

```
C:\Users\Fran\Desktop\WIM>heroku create
» Warning: heroku update available from 7.46.2 to 7.47.6.
Creating app... done, ☑ boiling-scrubland-63334
https://boiling-scrubland-63334.herokuapp.com/ | https://git.heroku.com/boiling-scrubland-63334.git
```

Slika 5.5: Naredba za stvaranje nove Heroku aplikacije te ispis dodjeljenog imena

### Namještanje i stvaranje baze podataka na Herokuu

Kako bi se stvorila baza podataka u komandnu liniju je potrebno upisati "heroku addons:create heroku-postgresql -a <APLIKACIJA>" gdje je <APLIKACIJA> ime vaše aplikacije.

```
C:\Users\Fran\Desktop\WIM>heroku addons:create heroku-postgresql -a boiling-scrubland-63334
» Warning: heroku update available from 7.46.2 to 7.47.6.
Creating heroku-postgresql on ☑ boiling-scrubland-63334... free
Database has been created and is available
! This database is empty. If upgrading, you can transfer
! data from another database with pg:copy
Created postgresql-silhouetted-05544 as DATABASE_URL
Use heroku addons:docs heroku-postgresql to view documentation
```

Slika 5.6: Naredba za stvaranje PostgreSQL baze podataka za Heroku aplikaciju te ispis dodjeljenog imena

## Instalacija PostgreSQL-a

Sa službene stranice je potrebno skinuti i instalirati PostgreSQL i potom staviti PostgreSQL u sistemske varijable.

## Stvaranje i punjenje baze podataka

Kako bismo imali funkcionalnu bazu podataka potrebno je izmijeniti database\_and\_inserts.sql sa repozitorija tako da promijenimo vrijednosti INSERT naredbe u kojoj se dodaje administrator. Zatim u komandnoj liniji treba pokrenuti "heroku pg:psql -a <APLIKACIJA>" što nas spaja na konzolu baze podataka. Zatim treba jednostavno kopirati i zalijepiti izmijenjeni database\_and\_inserts.sql.

```
132 INSERT INTO korisnik(  
133     lozinka,  
134     razina_korisnika,  
135     ime,  
136     prezime,  
137     email)  
138     VALUES ('admin', 3, 'admin', 'admin', 'admin@admin');
```

Slika 5.7: Dio SQL upita koji se treba promijeniti u lozinka, 3, ime, prezime i email željenog administratora

```
C:\Users\Fran\Desktop\WIM>heroku pg:psql -a boiling-scrubland-63334  
» Warning: heroku update available from 7.46.2 to 7.47.6.  
--> Connecting to postgresql-silhouetted-05544  
psql (12.2, server 12.5 (Ubuntu 12.5-1.pgdg20.04+1))  
WARNING: Console code page (437) differs from Windows code page (1252)  
8-bit characters might not work correctly. See psql reference  
page "Notes for Windows users" for details.  
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, bits: 256, compression: off)  
Type "help" for help.  
boiling-scrubland-63334::DATABASE=>
```

Slika 5.8: Naredba za spajanje na stvorenu bazu podataka

## Stavljanje backenda na Heroku

Potrebno je skinuti izvorni kod iz direktorija fooBarServer sa repozitorija i staviti ga u direktorij u kojemu smo stvorili Heroku aplikaciju. Zatim treba, pomoću redom "git add .", "git commit -m "<VASA\_PORUKA>" i "git push origin master", aplikaciju postaviti na server. To pokreće izgradnju aplikacije te je nakon toga aplikacija spremna za korištenje.

```
remote: [INFO] Copying spring-test-5.2.9.RELEASE.jar to /tmp/build_4ffeedcb/target/dependency/spring-test-5.2.9.RELEASE.jar
remote: [INFO] Copying xmlunit-core-2.7.0.jar to /tmp/build_4ffeedcb/target/dependency/xmlunit-core-2.7.0.jar
remote: [INFO] Copying jakarta.persistence-api-2.2.3.jar to /tmp/build_4ffeedcb/target/dependency/jakarta.persistence-api-2.2.3.jar
remote: [INFO] Copying spring-boot-autoconfigure-2.3.4.RELEASE.jar to /tmp/build_4ffeedcb/target/dependency/spring-boot-autoconfigure-2.3.4.RELEASE.jar
remote: [INFO] Copying spring-boot-2.3.4.RELEASE.jar to /tmp/build_4ffeedcb/target/dependency/spring-boot-2.3.4.RELEASE.jar
remote: [INFO] Copying gson-2.8.6.jar to /tmp/build_4ffeedcb/target/dependency/gson-2.8.6.jar
remote: [INFO] --- maven-install-plugin:2.5.2:install (default-install) @ fooBar ---
remote: [INFO] Installing /tmp/build_4ffeedcb/target/fooBar-0.0.1-SNAPSHOT.jar to /tmp/codon/tmp/cache/.m2/repository/com/example/fooBar/0.0.1-SNAPSHOT/fooBar-0.0.1-SNAPSHOT.jar
remote: [INFO] Installing /tmp/build_4ffeedcb/pom.xml to /tmp/codon/tmp/cache/.m2/repository/com/example/fooBar/0.0.1-SNAPSHOT/fooBar-0.0.1-SNAPSHOT.pom
remote: [INFO] -----
remote: [INFO] BUILD SUCCESS
remote: [INFO] -----
remote: [INFO] Total time: 10.029 s
remote: [INFO] Finished at: 2021-01-14T17:04:21Z
remote: [INFO] -----
remote: -----> Discovering process types
remote: Procfile declares types      -> (none)
remote: Default types for buildpack -> web
remote:
remote: -----> Compressing...
remote: Done: 107.6M
remote: -----> Launching...
remote: Released v267
remote: https://murmuring-coast-83056.herokuapp.com/ deployed to Heroku
remote:
remote: Verifying deploy... done.
remote: To https://git.heroku.com/murmuring-coast-83056.git
remote: e587dce..5562d04 master -> master
```

Slika 5.9: Poruka da se baze uspješno postavila na Heroku

## Pokretanje mobilne aplikacije

Potrebno je iz direktorija "Izvršni kod/apk" skinuti aplikacije za mobitel te instalirati onu za Vaš mobilni uređaj. Klikom na ikonu aplikacije Papirni.ca moguće je započeti njeno korištenje.

## 6. Zaključak i budući rad

Projekt *Papirni.ca* bio je u izradi 4.10.2020. - 14.1.2021. Tijekom tih 14 tjedana tim je izradio aplikaciju s (gotovo svim) traženim funkcionalnostima. Zadatak je bio dizajnirati, programirati i isporučiti aplikaciju koja predstavlja papirnicu tj. trgovinu papirom i nekim vezanim uslugama. Korisnici (kupci) podijeljeni su na 2 vrste - privatni i poslovni te ovisno o vrsti kupca mogli su birati između kupovanja papira preddefiniranih i posebnih stilova te pretplaćivati se na proizvode iz papirnice. Plaćanje se trebalo odvijati putem usluga *PayPal* no zbog manjka vremena za taj dio i problema pri implementaciji te funkcionalnosti on nije napravljen (umjesto mjesečnog naplaćivanja, pretplatnike se obavještava o izvršenoj pretplati svaki mjesec).

U prvoj fazi projekta radilo se na okupljanju tima, podijeli posla, dizajnu aplikacije i izradi osnovnih funkcionalnosti. tim je podijeljen na 3 glavne skupine: frontend, backend i miješani poddtim (manji zadaci na frontendu i backendu, dokumentacija). Najprije su osmišljeni izvedba baze podataka i izgled aplikacije. U dokumentaciji su napravljeni prvi osnovni koraci i izrađeni početni dijagrami koji su pomogli u kasnijoj implementaciji funkcionalnosti - obrasci uporabe, sekvencijski dijagrami, model baze podataka i dijagram razreda. Tijekom prve faze tim se povremeno sastajao kako bi se razmijenile daljnje ideje i definirali pojedinačni zadaci. Najveći izazovi u ovoj fazi bili su postavljanje servera i baze podataka na Heroku te uspostavljanje komunikacije između frontenda i backenda koji su riješeni uz pomoć literature, dokumentacije i već stečenih znanja.

U drugoj fazi projekta radilo se na implementaciji ostatka funkcionalnosti i dovršavanju dokumentacije. Ažurirani su dijagrami iz prethodnih faza (dijagram baze podataka, dijagram razred) uslijed manjih promjena u bazi i nekim razredima. Najveći izazovi u ovoj fazi bili su izrada košarice (cart) i (kasnije) *PayPal*. Backend i frontend timovi imali su regularne tjedne sastanke u ovom periodu na kojemu su osim rasprava zajedno radili na programskom kodu i testirali dobivena rješenja.

Moguće proširenje aplikacije bilo bi uspješno dodavanje usluge *PayPal* i proširenje asortimana papirnice.

Rad na projektu je bilo vrijedno iskustvo i lekcija o zajedničkom radu, organizaciji i usklađenju poddtimova te komunikacije među istima. Proširili smo obzore i stekli nova znanja vezana uz programiranje i rad u sustavu Git koji su zasigurno veliki bonusi za buduće radno okruženje (za koje postoji velika mogućnost da će biti organizirano na sličan način).

# Popis literature

1. Programsko inženjerstvo, FER ZEMRIS, <http://www.fer.hr/predmet/proinz>
2. I. Sommerville, "Software engineering", 8th ed, Addison Wesley, 2007.
3. T.C.Lethbridge, R.Langaniere, "Object-Oriented Software Engineering", 2nd ed. McGraw-Hill, 2005.
4. I. Marsic, Software engineering book", Department of Electrical and Computer Engineering, Rutgers University, <http://www.ece.rutgers.edu/~marsic/books/SE>
5. The Unified Modeling Language, <https://www.uml-diagrams.org/>
6. Astah Community, <http://astah.net/editions/uml-new>

# Indeks slika i dijagrama

2.1	Snimka web-stranice papirnice Denova . . . . .	6
2.2	Snimka raznolike ponude papira papirnice Denova . . . . .	6
3.1	Dijagram obrazaca uporabe za korisnika i klijenta . . . . .	19
3.2	Dijagram obrazaca uporabe za privatne i poslovne klijente . . . . .	20
3.3	Dijagram obrazaca uporabe za poslovne klijente i administratora . .	21
3.4	Sekvencijski dijagram za UC6 - Dodavanje u košaricu . . . . .	22
3.5	Sekvencijski dijagram za UC11 - Otkazivanje mjesečnih pretplata .	23
3.6	Sekvencijski dijagram za UC14 - Naručivanje usluga . . . . .	24
3.7	Sekvencijski dijagram za UC25 - Otkazivanje mjesečne pretplate poslovnih klijenata . . . . .	25
4.1	ER dijagram baze podataka . . . . .	34
4.2	Dijagram razreda - controlleri . . . . .	35
4.3	Dijagram razreda - servisi . . . . .	36
4.4	Dijagram razreda - DAO-i . . . . .	37
4.5	Dijagram razreda - modeli . . . . .	38
4.6	Dijagram stanja . . . . .	39
4.7	Dijagram aktivnosti . . . . .	41
4.8	Dijagram komponenti . . . . .	42
5.1	Unit testovi . . . . .	45
5.2	Dijagram razmještaja . . . . .	46
5.3	Upis naredbe za prijavu u Heroku CLI te upit . . . . .	47
5.4	Otvoreni preglednik za prijavu na Heroku . . . . .	48
5.5	Naredba za stvaranje nove Heroku aplikacije te ispis dodjeljenog imena . . . . .	48
5.6	Naredba za stvaranje PostgreSQL baze podataka za Heroku aplikaciju te ispis dodjeljenog imena . . . . .	48
5.7	Dio SQL upita koji se treba promijeniti u lozinka, 3, ime, prezime i email željenog administratora . . . . .	49

5.8	Naredba za spajanje na stvorenu bazu podataka . . . . .	49
5.9	Poruka da se baze uspješno postavila na Heroku . . . . .	50
6.1	Statistika za granu master . . . . .	61
6.2	Statistika za granu master - nastavak . . . . .	62
6.3	Statistika za granu devdoc . . . . .	62
6.4	Statistika za granu client . . . . .	63
6.5	Statistika za granu database . . . . .	63



# Dodatak: Prikaz aktivnosti grupe

## Dnevnik sastajanja

### *Kontinuirano osvježavanje*

#### 1. sastanak

- Datum: 4. listopada 2020.
- Prisustvovali: F. Hrabar, I. Mihovilović, Z. Protulipac, Z. Pećanić, T. Likakur, J. Vujević, M. Kukelščak
- Teme sastanka:
  - organizacija grupe
  - raspodjela posla

#### 2. sastanak

- Datum: 15. listopada 2020.
- Prisustvovali: F. Hrabar, I. Mihovilović, Z. Protulipac, Z. Pećanić, T. Likakur, J. Vujević, M. Kukelščak
- Teme sastanka:
  - odabir programske potpore koja će se koristiti
  - izgled i dizajn aplikacije

#### 3. sastanak

- Datum: 2. 11. 2020.
- Prisustvovali: F. Hrabar, I. Mihovilović, Z. Protulipac, Z. Pećanić, T. Likakur, J. Vujević, M. Kukelščak
- Teme sastanka:
  - izvedba backenda i povezivanje s frontendom (json)
  - povezivanje git - Heroku

#### 4. sastanak

- Datum: 8. 11. 2020.
- Prisustvovali: F. Hrabar, I. Mihovilović, Z. Protulipac, Z. Pećanić, T. Likakur, J. Vujević, M. Kukelščak

- Teme sastanka:
  - izvedba frontenda
  - json i povezivanje backend-frontend

#### 5. sastanak

- Datum: 20. 11. 2020.
- Prisustvovali: F. Hrabar, I. Mihovilović, Z. Protulipac, Z. Pećanić, T. Likakur, J. Vujević, M. Kukelščak
- Teme sastanka:
  - daljnja podjela posla
  - backend

#### 6. sastanak

- Datum: 28. 11. 2020.
- Prisustvovali: F. Hrabar, I. Mihovilović, Z. Protulipac, Z. Pećanić, T. Likakur, J. Vujević, M. Kukelščak
- Teme sastanka:
  - frontend i json
  - cart
  - dodatci u bazi

#### 7. sastanak

- Datum: 6. 12. 2020.
- Prisustvovali: F. Hrabar, I. Mihovilović, Z. Protulipac, Z. Pećanić, T. Likakur, J. Vujević, M. Kukelščak
- Teme sastanka:
  - cart
  - backend
  - podjela posla tijekom praznika

#### 8. sastanak

- Datum: 12. 12. 2020.
- Prisustvovali: F. Hrabar, I. Mihovilović, Z. Protulipac, Z. Pećanić, T. Likakur, J. Vujević, M. Kukelščak
- Teme sastanka:
  - uloge admina

#### 9. sastanak

- Datum: 18. 12. 2020.

- Prisustvovali: F. Hrabar, I. Mihovilović, Z. Protulipac, Z. Pećanić, T. Likakur, J. Vujević, M. Kukelščak
- Teme sastanka:
  - podjela posla do kraja
  - dokumentacija - dijagrami u 2. ciklusu

#### 10. sastanak

- Datum: 27. 12. 2020.
- Prisustvovali: F. Hrabar, I. Mihovilović, Z. Protulipac, Z. Pećanić, T. Likakur, J. Vujević, M. Kukelščak
- Teme sastanka:
  - backend
  - vodeni žig

#### 11. sastanak

- Datum: 5. 1. 2021.
- Prisustvovali: F. Hrabar, I. Mihovilović, Z. Protulipac, Z. Pećanić, T. Likakur, J. Vujević, M. Kukelščak
- Teme sastanka:
  - završne funkcije
  - PayPal
  - demo alfa inačice - što i kako

#### 12. sastanak

- Datum: 12. 1. 2021.
- Prisustvovali: F. Hrabar, I. Mihovilović, Z. Protulipac, Z. Pećanić, T. Likakur, J. Vujević, M. Kukelščak
- Teme sastanka:
  - završavanje i dogovor oko termina kolokviranja
  - dogovori oko prezentacije

## Tablica aktivnosti

	Fran Hrabar	Ivan Mihovilović	Zvonimir Protulipac	Tomislav Likakur	Zrinka Pećanić	Josipa Vujević	Mihael Kukulščak
Upravljanje projektom	10	0	0	0	0	1	0
Opis projektnog zadatka	0	0	0	0	3	0	0
Funkcionalni zahtjevi	2	0	0	0	2	0	0
Opis pojedinih obrazaca	5	0	0	0	1	0	1
Dijagram obrazaca	0	0	0	0	3	0	0
Sekvencijski dijagrami	0	0	0	0	2	0	0
Opis ostalih zahtjeva	0	0	0	0	1	0	0
Arhitektura i dizajn sustava	1	0	0	0	0	8	1
Baza podataka	3	0	2	0	0	1	0
Dijagram razreda	3	0	0	0	2	0	0
Dijagram stanja	0	0	0	0	0	3	0
Dijagram aktivnosti	0	0	0	0	4	0	0
Dijagram komponenti	0	0	0	0	0	3	0
Korištene tehnologije i alati	0	0	0	0	2	0	0
Ispitivanje programskog rješenja	0	0	2	0	0	0	0
Dijagram razmještaja	0	0	0	0	2	0	0
Upute za puštanje u pogon	2	0	0	0	0	0	0
Dnevnik sastajanja	0	0	0	0	1	0	1
Zaključak i budući rad	0	0	0	0	2	0	0
Popis literature	0	0	0	0	1	0	0
Izrada početne stranice	0	0	0	0	0	1	0
Izrada baze podataka	3	0	8	0	0	0	0
Spajanje s bazom podataka	15	0	15	0	0	0	0
Back end	15	0	20	0	3	5	15
Front end	0	35	0	25	0	1	0

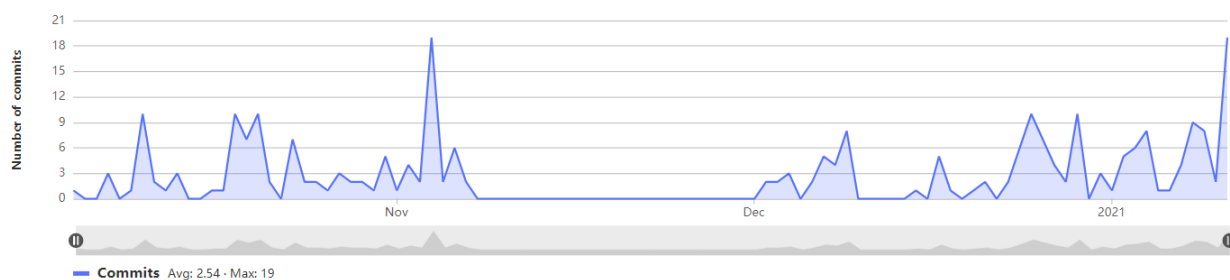
	<b>Fran Hrabar</b>	<b>Ivan Mihovilović</b>	<b>Zvonimir Protulipac</b>	<b>Tomislav Likakur</b>	<b>Zrinka Pećanić</b>	<b>Josipa Vujević</b>	<b>Mihael Kukulščak</b>
Osmišljavanje dizajna	0	2	0	5	0	1	0

## Dijagrami pregleda promjena

Slike stat1 i stat2 prikazuju statistiku za granu master, stat3 za granu devdoc, stat4 za client i stat5 za database.

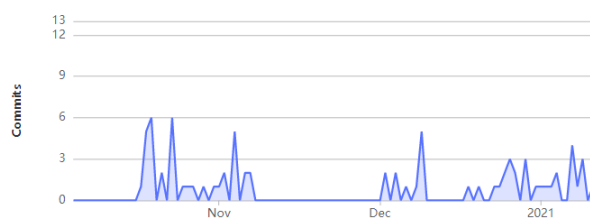
### Commits to master

Excluding merge commits. Limited to 6,000 commits.



### unknown

77 commits (ivan.mihovilovic@fer.hr)



### tlikakur

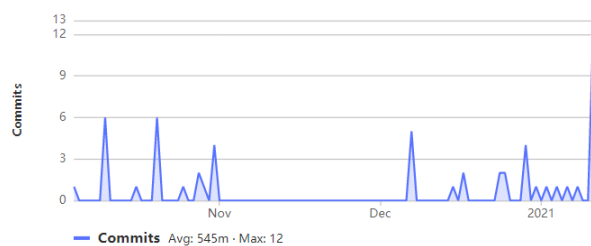
72 commits (tomislav.likokur@fer.hr)



Slika 6.1: Statistika za granu master

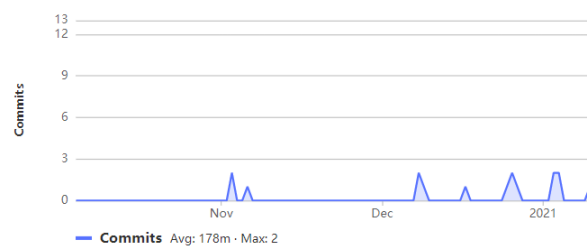
## Fran Hrabar

55 commits (fran.hrabar@fer.hr)



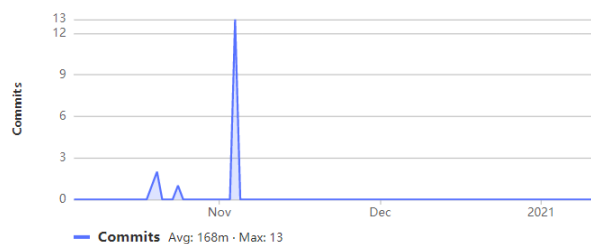
## Zvonimir Protulipac

18 commits (zp51561@fer.hr)



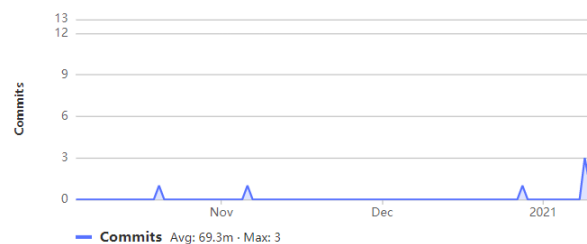
## Josipa Vujevic

17 commits (josipa.vujevic@fer.hr)



## Kookie123

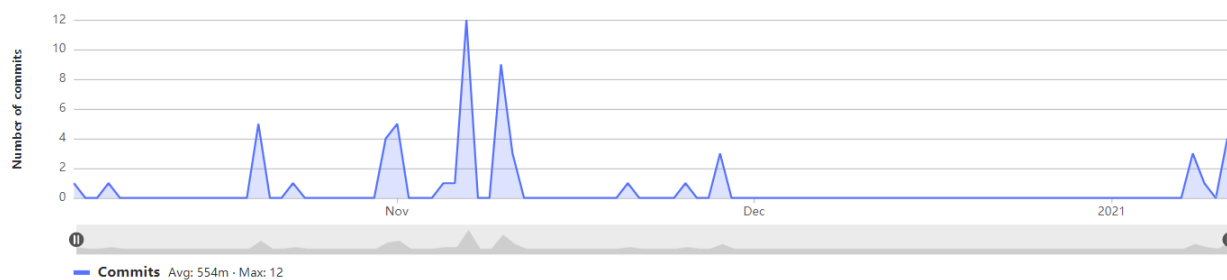
7 commits (mihael.kukelscak@fer.hr)



Slika 6.2: Statistika za granu master - nastavak

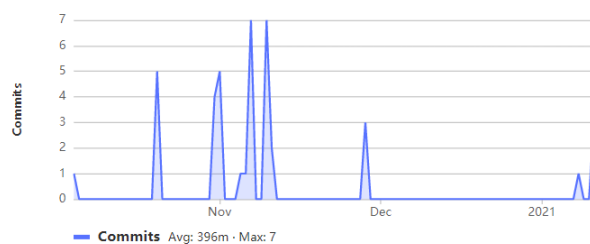
## Commits to devdoc

Excluding merge commits. Limited to 6,000 commits.



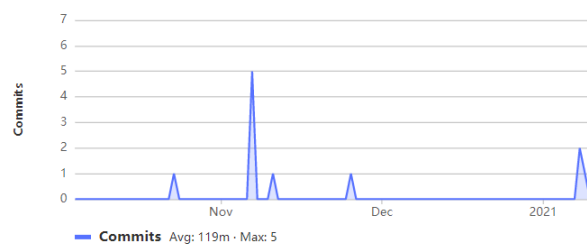
## Fran Hrabar

40 commits (fran.hrabar@fer.hr)



## Zrinka Pecanic

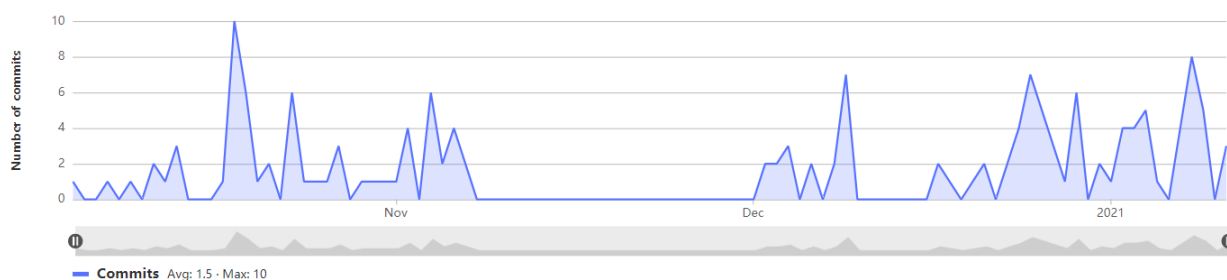
12 commits (zrinka.pecanic@fer.hr)



Slika 6.3: Statistika za granu devdoc

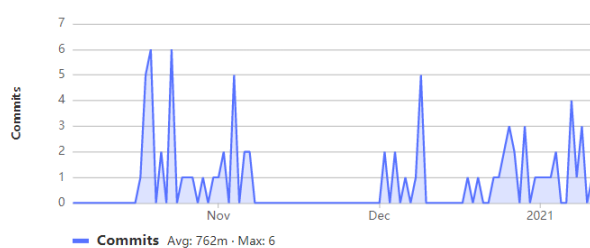
## Commits to client

Excluding merge commits. Limited to 6,000 commits.



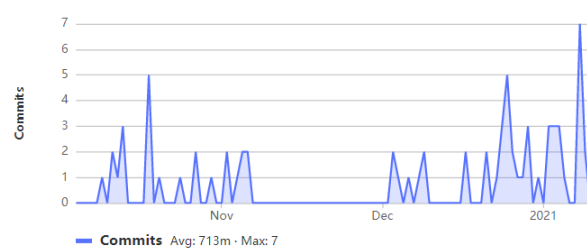
## unknown

77 commits (ivan.mihovilovic@fer.hr)



## tlikakur

72 commits (tomislav.likokur@fer.hr)



Slika 6.4: Statistika za granu client

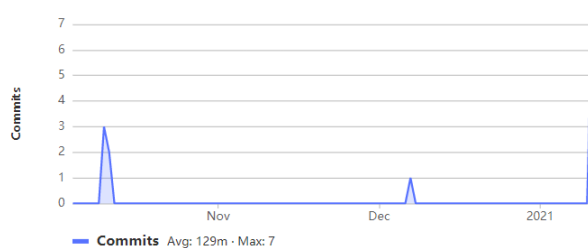
## Commits to database

Excluding merge commits. Limited to 6,000 commits.



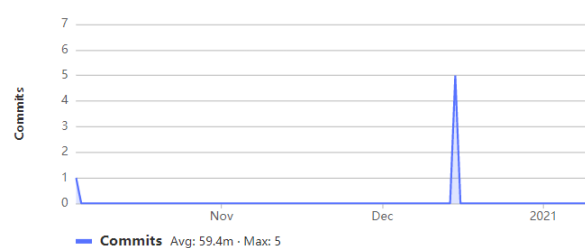
## Zvonimir Protulipac

13 commits (zp51561@fer.hr)



## Fran Hrabar

6 commits (fran.hrabar@fer.hr)



Slika 6.5: Statistika za granu database