

VISOKA ŠKOLA STRUKOVNIH STUDIJA ZA INFORMACIONE TEHNOLOGIJE



Internet programerski alati

Projekat

Projektovanje i razvoj Java Web aplikacije za ketering službu

Predmetni nastavnik:
Prof. Dr. Svetlana Jevremović

Student:
David Gudović 11/18
Datum predaje: 28.07.2022

Beograd
Jul, 2022

Sadržaj

1.	UVOD	3
2.	USE CASE.....	4
2.1	Akter Klijent	4
2.2	Akter Menadžer.....	8
2.3	Akter Administrator	10
3.	UGOVORI	12
4.	KONCEPTUALNI MODEL.....	29
4.1	Dijagram klasa.....	29
4.2	Relacioni model.....	29
4.3	IDEF1X dijagram baze podataka.....	30
4.5	Implementacija baze	31
5.	DIJAGRAMI SEKVENCI	32
5.1	Registracija	32
5.2	Login	33
5.3	Izmena informacija.....	34
5.4	Filter.....	35
5.5	Naručivanje	36
5.6	Ponavljanje narudžbi	37
5.7	Ostvarivanje narudžbi	38
5.8	Kreiranje izveštaja.....	39
5.9	Dodavanje korisnika	40
5.10	Editovanje korisnika	41
6.	KORISNIČKI INTERFEJS	42
6.1	Autentikacija	42
6.2	Pregled ponude	43
6.3	Korpa za kupovinu.....	44
6.4	Profil Korisnika.....	45
6.5	Menadžerski paneli	47
6.6	Administratorski panel	49
6.7	Poruke	50
7.	ZAKLJUČAK	52
8.	REFERENCE	52

1. UVOD

U ovom radu projektovana je i implementirana web aplikacija fiktivne ketering službe „Lorem Catering“. Web aplikacija je projektovana prateći inkrementalne faze Larmanove metode.

U nastavku date su slike UML use case dijagrama i tekstualni opisi svih njih, zatim su uvidom u opise slučajeva korišćenja kreirani sistemski ugovori i date fotografije njihovih sekvencijalnih dijagrama. Kreiran je i opisan konceptualni model aplikacije, dijagrami klase kao i relacioni model baze podataka, IDEF1X dijagram baze i slika Designer-a implementirane baze u phpmyadmin okruženju.

Zatim su nacrtani i prokomentarisani detaljni sekvencijalni dijagrami implementacije nekoliko važnijih sistemskih ugovora formiranih u prethodnom koraku projektovanja. Na kraju su dati snimci ekrana korisničkog interfejsa i kratak zaključak.

Funkcionalni zahtevi web aplikacije su bazirani na okvirnom opisu zadatom na studentskom portalu [1].

Funkcionalni zahtevi:

Korisnici aplikacije su **klijenti, menadžeri i administratori**.

Web aplikacija pruža mogućnost svim posetiocima da pregledaju ponudu, detaljne opise, cene i fotografije proizvoda, kao i da istu filtriraju po slanom ili slatkom programu, ili različitim kategorijama hrane.

Svi ulogovani korisnici ostvaruju pristup dodavanju proizvoda u korpu, editovanja, brisanja stavki iz korpe, ostvarivanja poena za popust kao i korišćenja poena za do 20% popusta na pojedinačnim narudžbama. Ulogovani korisnici takođe mogu pregledati istoriju svojih narudžbi, dodati stare narudžbe direktno u korpu, ili otkazati narudžbe koje još nisu ostvarene.

Svi korisnici imaju potpunu kontrolu nad svojim korisničkim profilima, mogu editovati sve podatke osim korisničko ime, ili izbrisati svoj profil.

Korisnici sa roлом menadžera pored navedenih mogućnosti mogu editovati, dodavati i brisati kategorije kao i pojedinačne proizvode. Takođe imaju pristup menadžerskom panelu gdje na listi narudžbi napravljenih u sistemu, mogu ostvarivati ili otkazivati narudžbe. Listu narudžbi mogu filtrirati po korisniku ili pregledati sve narudžbe u sistemu.

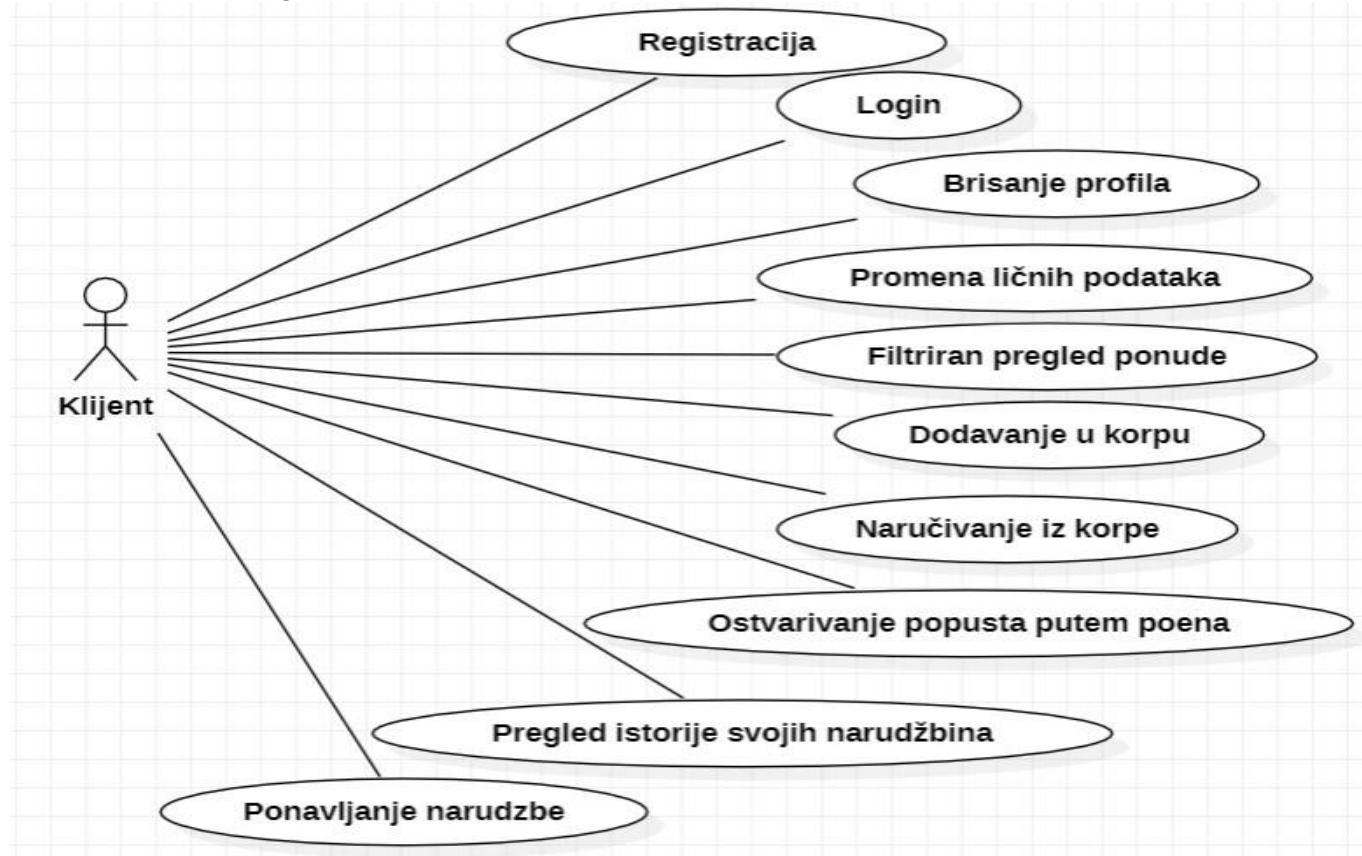
Menadžeri mogu generisati i u pdf formatu eksportovati mesečni izveštaj.

Korisnici sa roлом administratora, pored prethodno opisanih funkcionalnosti imaju pristup admin panelu odakle mogu brisati ili dodavati Korisnike, takođe mogu postavljati druge korisnike na rolu menadžera ili administratora, kao i oduzimati role.

Ključne reči: Larmanova metoda, Java Web, Servlet, Java Bean, Java Server Pages, MySQL, MVC, Repository pattern.

2. USE CASE

2.1 Akter Klijent



(Slika 1) Use case dijagram Klijenta

SK 1

Naziv: Registracija

Aktor: Klijent

Preduslov: Korisnik je na stranici registracija.jsp

Osnovni scenario:

1. Korisnik unosi Korisnicko Ime, Ime, Prezime, Adresu i Password (APUSO)
2. Korisnik traži od sistema da se registruje (APSO)
3. Sistem unosi novog korisnika u bazu podataka (SO)
4. Korisniku se prikazuje poruka o uspehu (IA)

Alternativni scenario:

- 3.1 Podaci nisu validni i korisniku se prikazuje poruka o grešci (IA)

SK 2

Naziv: Login

Aktor: Klijent

Preduslov: Korisnik je registrovan i nalazi se na stranici login.jsp

Osnovni scenario:

1. Korisnik unosi korisnicko ime i password (APUSO)
2. Korisnik traži od sistema da se loguje (APSO)
3. Sistem započinje sesiju korisnika (SO)
4. Korisniku je umesto opcije za Login i Registraciju prikazano dugme profila (IA)

Alternativni scenario:

- 3.1 Podaci nisu validni i korisniku se prikazuje poruka o greški (IA)

SK 3

Naziv: Brisanje profila

Aktor: Klijent

Preduslov: Korisnik je ulogovan

Osnovni scenario:

1. Korisnik unosi sifru u formu za brisanje (APUSO)
2. Korisnik traži od sistema da izbriše profil (APSO)
3. Sistem proverava sifru (SO)
4. Sistem briše korisnika iz baze i invalidira sesiju (SO)
5. Korisniku se prikazuje pocetna strana (IA)

Alternativni scenario:

- 3.1 Sifra nije tacna i prikazuje se poruka da pokusa ponovo(IA)

SK 4

Naziv: Promena ličnih podataka

Aktor: Klijent

Preduslov: Klijent je ulogovan

Osnovni scenario:

1. Klijent edituje željena polja na prikazu profila (APUSO)
2. Klijent zatražuje promjenu informacija (APSO)
3. Sistem updejtuje bazu novim informacijama (SO)
4. Sistem pokazuje poruku o uspjehu (IA)

Alternativni scenario:

- 3.1 Updejt baze ne uspije, Sistem prikazuje poruku o grešci (IA)

SK 5

Naziv: Filtriran pregled ponude

Aktor: Klijent

Preduslov: Nema

Osnovni scenario:

1. Korisnik bira program(slatki ili slani) ili kategorije koje želi prikazane (APSO)
2. Sistem vrši upit u bazi po zadatim kriterijumima nad tabelom proizvodi (SO)
3. Sistem prikazuje rezultat (IA)

Alternativni scenario:

- 3.1 Nema rezultata pa sistem prikazuje poruku o greški (IA)

SK 6

Naziv: Dodavanje u korpu

Akter: Klijent

Preduslov: Klijent je ulogovan

Osnovni scenario:

1. Klijent unosi količinu željenog proizvoda (APUSO)
2. Klijent traži dodavanje u korpu (APSO)
3. Sistem dodaje proizvod u narudžbinu (SO)
4. Proizvod, količina i ukupna cena se dodaju u prikaz korpe (IA)

Alternativni scenario:

- 1.1 Proizvod je prvi artikal u narudžbini, pa sistem kreira narudžbinu u sesiji (SO)
- 1.2 Brojac artikala korpe se prikazuje u navbaru (IA)

SK 7

Naziv: Naručivanje iz korpe

Akter: Klijent

Preduslov: Klijent je dodao artikle u korpu

Osnovni scenario:

1. Ako je potrebno klijent vrši izmene u korpu (APUSO)
2. Klijent naručuje (APSO)
3. Sistem čuva narudžbu u bazi kao nesotvarenu dok je menadžer ne ostvari (SO)
4. Klijentu se prikazuje poruka o uspešnosti i ostvarenim poenima(IA)

Alternativni scenario:

- 2.1 Klijent otkazuje narudžbu (APSO)
- 2.2 Sistem miče narudžbu iz sesije (SO)
- 2.3 Klijent se vraća na početnu stranu (IA)

SK 8

Naziv: Ostvarivanje popusta putem poena

Akter: Klijent

Preduslov: Klijent je dodao aritikle u narudžbu

Osnovni scenario:

1. Klijent bira količinu popusta (APSO)
2. Sistem računa total sa popustom(SO)
3. Sistem prikazuje total sa popustom (IA)
4. Klijent naručuje (APSO)
5. Sistem uračunava popust u cenu za bazu (SO)
6. U bazi i istoriji je prikazana snižena cena (IA)

Alternativni scenario:

- 1.1 Klijent nema poena pa ne može da unese popust veći od 0 (IA)

SK 9

Naziv: Pregled istorije svojih narudžbina

Aktor: Klijent

Preduslov: Korisnik je ulogovan

Osnovni scenario:

1. Korisnik traži od sistema prikaz svojih porudžbina (APSO)
2. Sistem pretražuje narudžbine sa korisničkim imenom Korisnika (SO)
3. Sistem prikazuje rezultat korisniku (IA)

Alternativni scenario:

- 3.1 Nema rezultata, sistem prikazuje link za ponudu ketering službe (IA)

SK 10

Naziv: Ponavljanje narudzbine

Aktor: Klijent

Preduslov: Korisnik je ulogovan i ima prethodnih ostvarenih ili otkazanih narudzbi

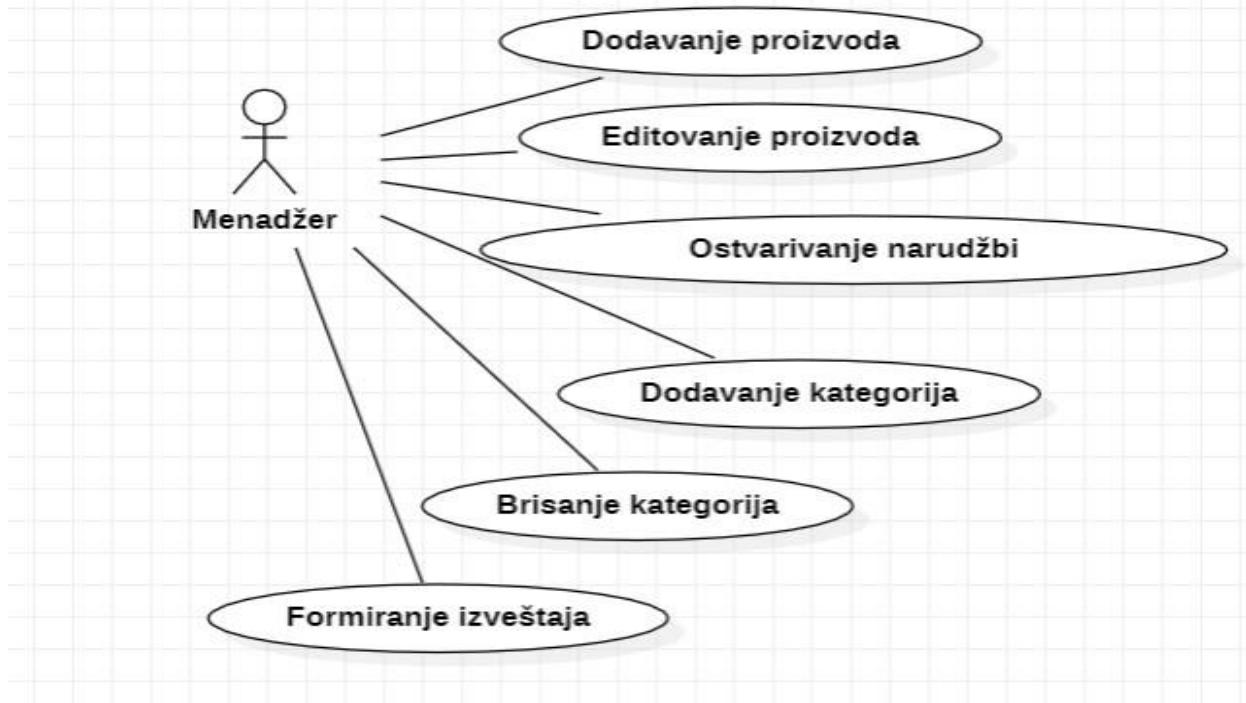
Osnovni scenario:

1. Korisnik traži ponavljanje narudzbe (APSO)
2. Sistem stavlja kopiju narudžbe u Korpu sesije (SO)
3. Sistem prikazuje stranicu korpe (IA)

Alternativni scenario:

- 3.1 Nema

2.2 Akter Menadžer



(Slika 2) Use Case Menadžera

***Napomena:** Podrazumevani preduslov svih SK menadžera je :

Korisnik je ulogovan sa rodom menadžer ili administrator.

****Napomena:** Menadžer nasleđuje sve use case-ove Klijenta

SK 11

Naziv: Dodavanje proizvoda

Akter: Menadžer

Preduslov: Podrazumevani.

Osnovni scenario:

1. Korisnik unosi naziv proizvoda, cenupogramu, sliku, opis, kategoriju i program (APUSO)
2. Korisnik zatražuje da se proizvod doda (APSO)
3. Sistem dodaje nov proizvod u bazu (SO)
4. Korisniku se prikazuje poruka o uspehu (IA)

Alternativni scenario:

- 4.1 Sistem ne može da doda proizvod, pa prikazuje poruku o grešci (IA)

SK 12

Naziv: Editovanje proizvoda

Akter: Menadžer

Preduslov: Podrazumevani.

Osnovni scenario:

1. Korisnik menja željene podatke na prikazu proizvoda (APUSO)
2. Korisnik zatražuje promenu od sistema (APSO)
3. Sistem vrši udejt proizvoda u bazi (SO)
4. Sistem prikazuje poruku o uspehu (IA)

Alternativni scenario:

- 4.1 Sistem ne može da updejtuje proizvod pa prikazuje poruku o greški (IA)

SK 13

Naziv: Ostvarivanje narudžbi

Akter: Menadžer

Preduslov: Korisnik je otvorio stranicu preglednarudzbi.jsp

Osnovni scenario:

1. Korisnik označi narudžbu kao ostvarenu (APSO)
2. Sistem updejtuje vrednost ostvarena narudžbine u bazi (SO)
3. Narudžba se prikazuje kao ostvarena (IA)

Alternativni scenario: Nema

SK 14

Naziv: Dodavanje kategorija

Akter: Menadžer

Preduslov: Podrazumevani.

Osnovni scenario:

1. Korisnik unosi naziv nove kategorije (APUSO)
2. Korisnik zatražuje dodavanje nove kategorije (APSO)
3. Sistem unosi novu kategoriju (SO)
4. Korisniku se prikazuje poruka o uspešnosti (IA)

Alternativni scenario: Nema

SK 15

Naziv: Brisanje kategorija

Akter: Menadžer

Preduslov: Podrazumevani.

Osnovni scenario:

1. Korisnik traži da se kategorija izbriše (APSO)
2. Sistem briše kategoriju (SO)

Alternativni scenario: Nema

SK 16

Naziv: Kreiranje izveštaja

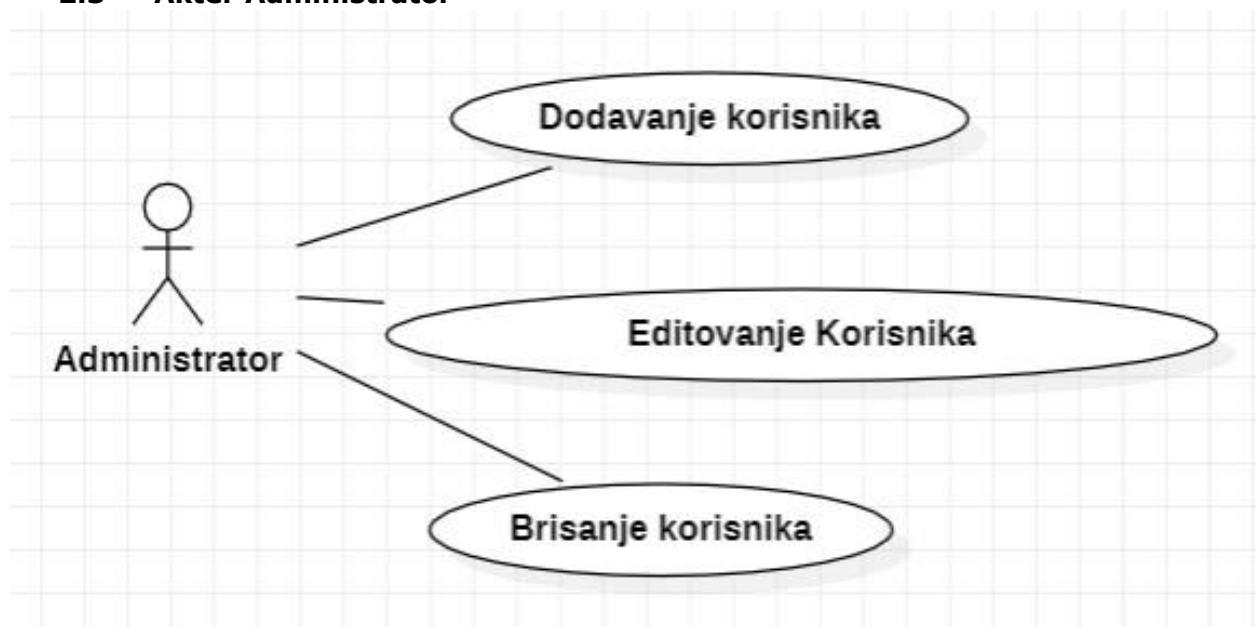
Akter: Menadžer

Preduslov: Podrazumevani

Osnovni scenario:

1. Korisnik zatražuje izveštaj (APSO)
2. Sistem generiše pdf izveštaja (SO)
3. Sistem download-uje pdf na korisnikov računar (IO)

Alternativni scenario: Nema

2.3 Akter Administrator

(Slika 3) Use Case Administratora

***Napomena:** Podrazumevani preduslov svih SK Administratora je:

Korisnik logovan sa roлом administrator.

****Napomena:** Administrator nasleđuje sve use case-ove Klijenta i Menadžera

SK 17

Naziv: Dodavanje korisnika

Akter: Administrator

Preduslov: Podrazumevani.

Osnovni scenario:

1. Korisnik unosi podatke novog korisnika (APUSO)
2. Korisnik zatražuje unos novog korisnika (APSO)
3. Sistem unosi novog korisnika u bazu(SO)
4. Korisniku se prikazuje poruka o uspehu (IA)

Alternativni scenario:

- 4.1. Korisnik se ne može uneti pa se korisniku prikazuje poruka o neuspehu (IA)

SK 18

Naziv: Editovanje korisnika

Akter: Administrator

Preduslov: Korisnik je na stranici pregledkorisnika.jsp

Osnovni scenario:

1. Korisnik edituje željene podatke (APUSO)
2. Korisnik zatražuje promenu podataka (APSO)
3. Sistem vrši update podataka u bazi (SO)
4. Sistem prikazuje poruku o uspehu (IA)

Alternativni scenario:

- 4.1. Promena se ne može izvršiti pa sistem prikazuje poruku o grešci.

SK 19

Naziv: Brisanje korisnika

Akter: Administrator

Preduslov: Korisnik je na stranici pregledkorisnika.jsp

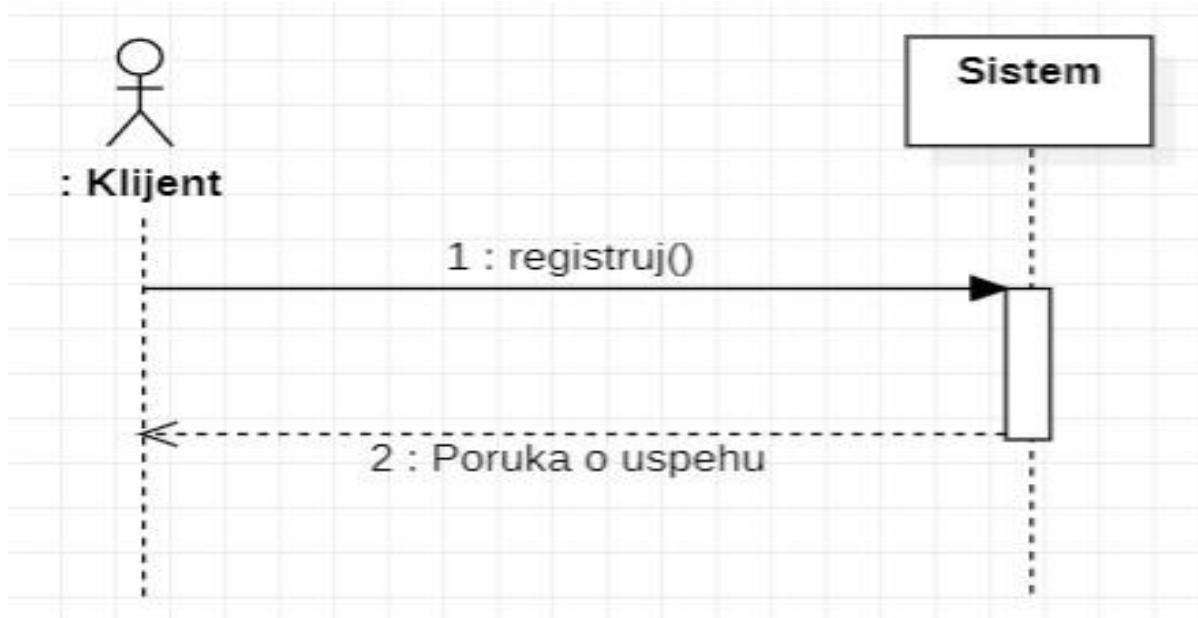
Osnovni scenario:

1. Korisnik označi korisnika za brisanje (APSO)
2. Sistem uklanja korisnika iz baze (SO)
3. Sistem prikazuje poruku o uspehu (IA)

Alternativni scenario:

- 3.1. Sistem ne može ukolniti korisnika, pa prikazuje poruku o grešci (IA)

3. UGOVORI



(Slika 4) Dijagram sekvenci SK1



(Slika 5) Dijagram sekvenci SK1 alternativni scenario

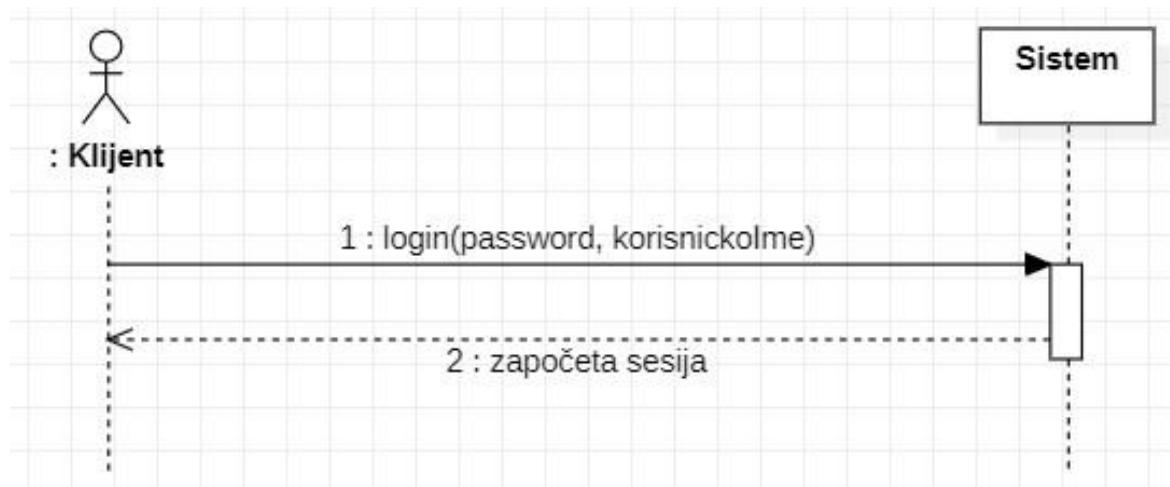
Ugovor UG1:

Operacija: registruj() : void

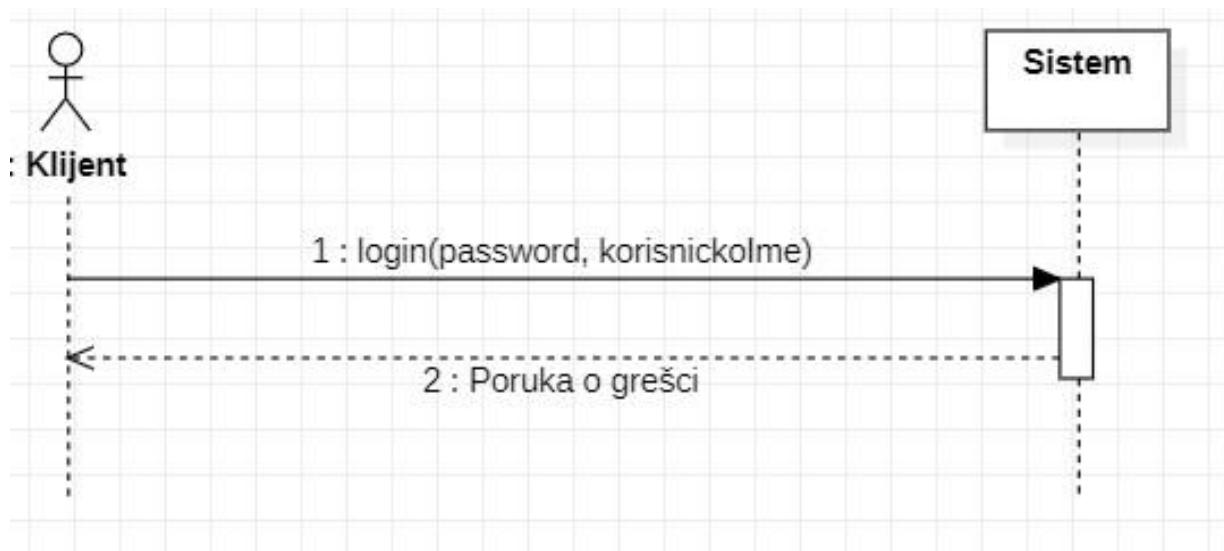
Veza sa SK: SK1

Preduslov: Controller je kreirao objekat nad kojim se poziva operacija

Postuslov: Kreiran je nov korisnik u bazi.



(Slika 6) Dijagram sekvenci SK2



(Slika 7) Dijagram sekvenci SK2 alternativni scenario

Ugovor UG2:

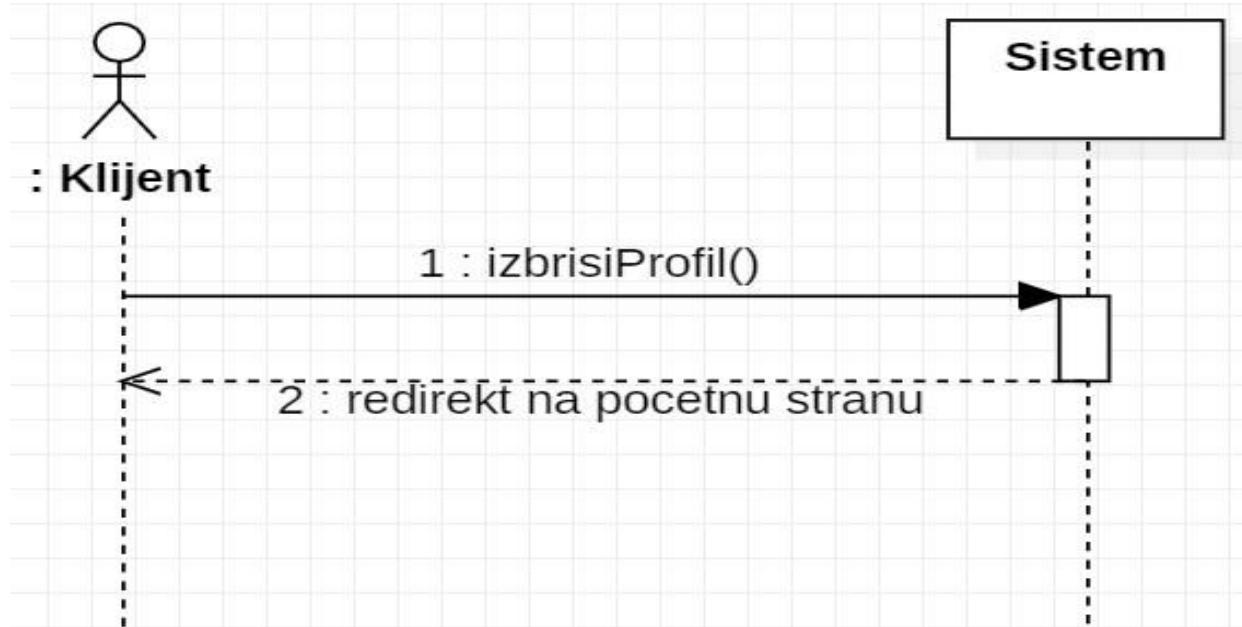
Operacija: login():boolean

Veza sa SK: SK2

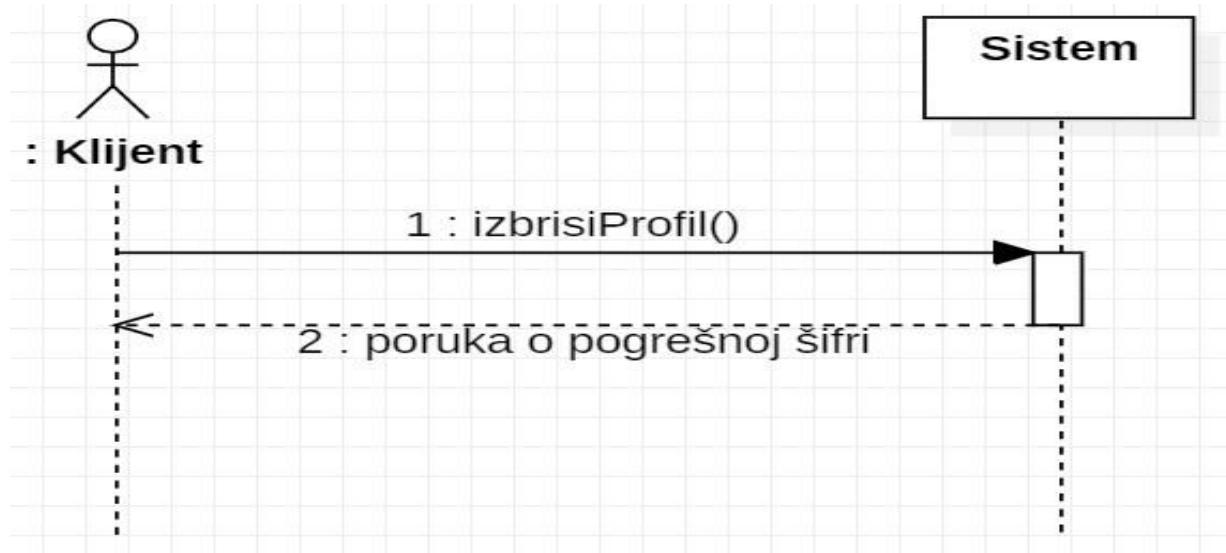
Preduslov: Controller je kreirao objekat sa zadatim parametrima

Postuslov: Započeta je sesija odgovarajućeg korisnika i postavljeni atributi korisnik i rola sesije.

***Napomena:** metoda login poziva se nad instancom Korisnika sa zadatim passwordom i korisničkim imenom, a sama metoda nema parametara.



(Slika 8) Dijagram sekvenci SK3



(Slika 9) Dijagram sekvenci SK3 alternativni scenario

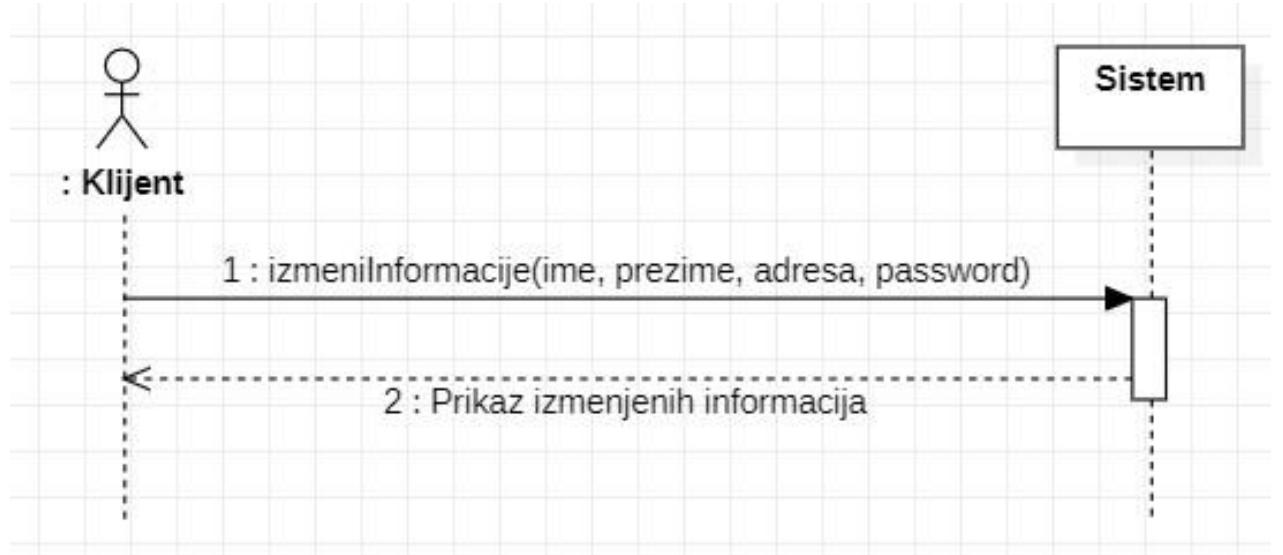
Ugovor UG3:

Operacija: izbrisiprofil() :boolean

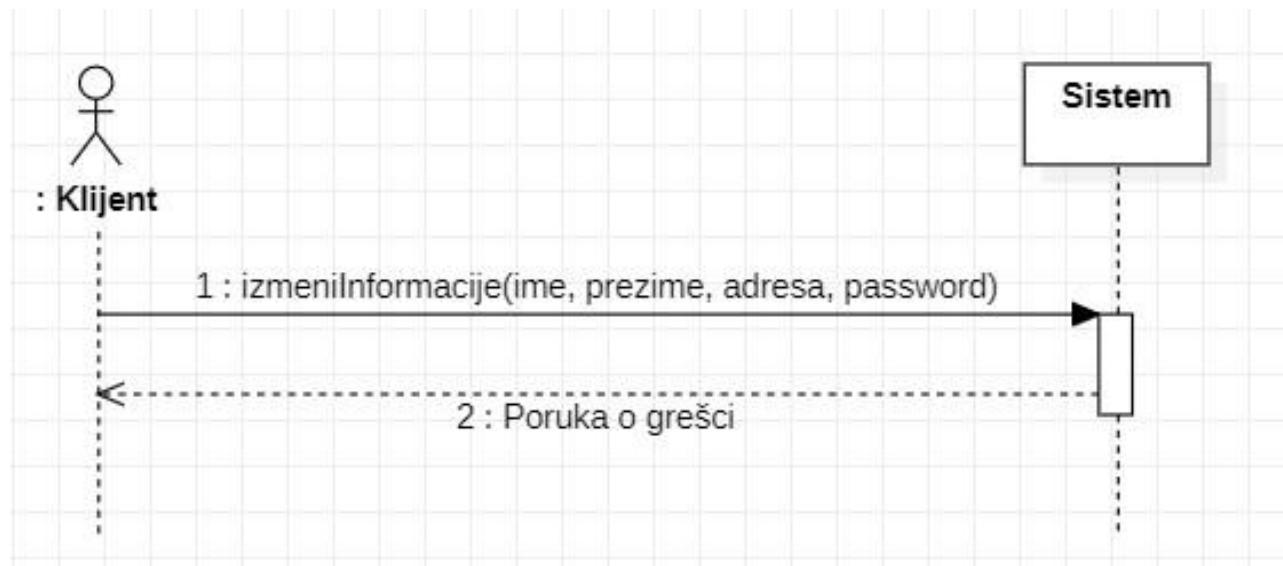
Veza sa SK: SK3

Preduslov: Korisnik je ulogovan i uneo je šifru

Postuslov: Korisnik je izbrisani iz baze , sesija invalidirana i korisnik je redirektovan na početnu stranu



(Slika 10) Dijagram sekvenci SK4



(Slika 11) Dijagram sekvenci SK4 alternativni scenario

Ugovor UG4:

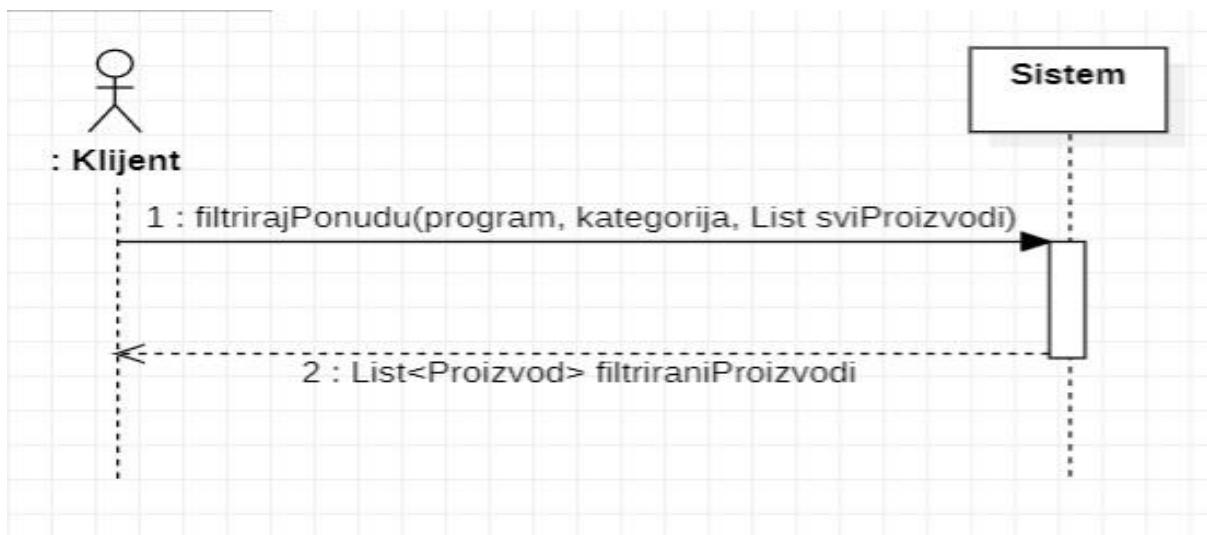
Operacija: izmeniInformacije(Korisnik(korisnickoIme, ime, prezime, adresa, password)) : bool

Veza sa SK: SK4

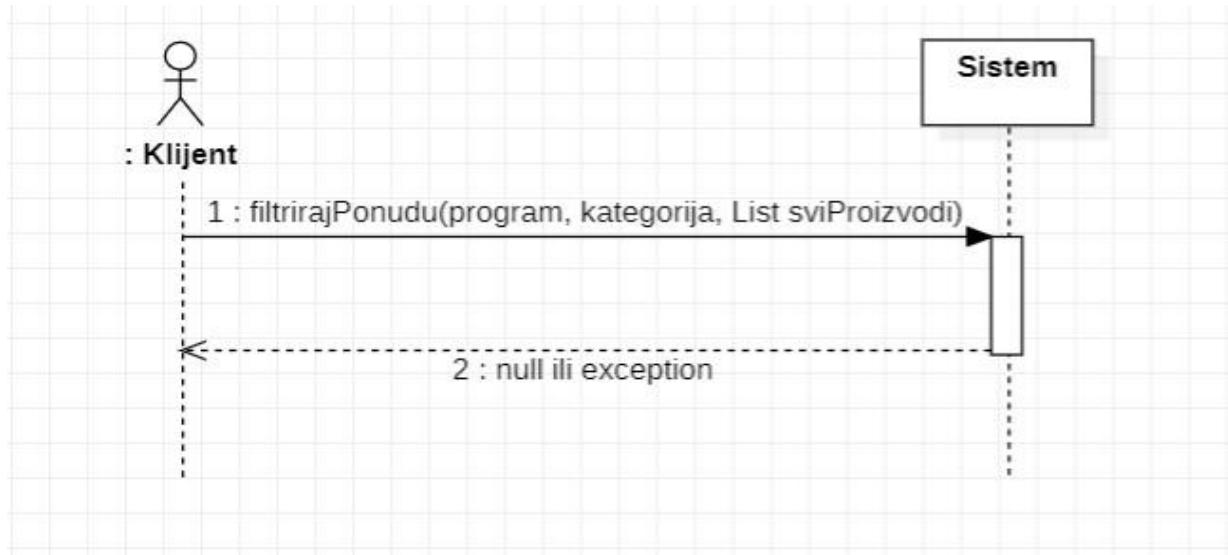
Preduslov: Korisnik je uneo validne nove podatke

Postuslov: Korisnikovi podaci su izmenjeni u bazi podataka

***Napomena:** Parametar metode je objekat klase Korisnik koji sadrzi potrebne informacije prikazane na dijagramu.



(Slika 12) Dijagram sekvenci SK5



(Slika 13) Dijagram sekvenci SK5 alternativni scenario

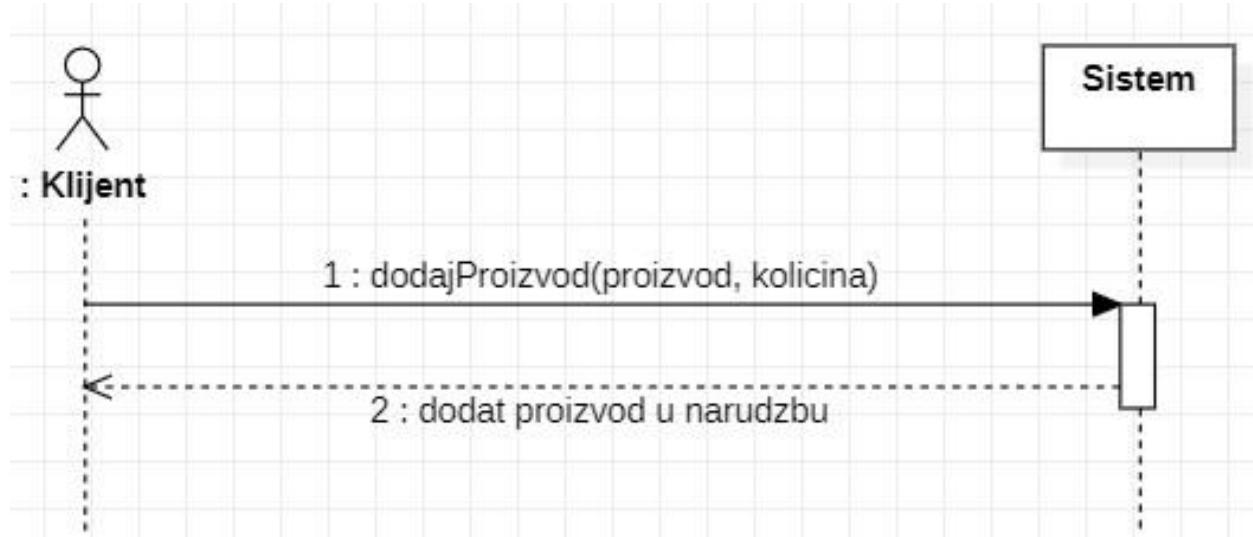
Ugovor UG5:

Operacija: static filtrirajPonudu(program, kategorija, List sviProizvodi) : List<Proizvod>

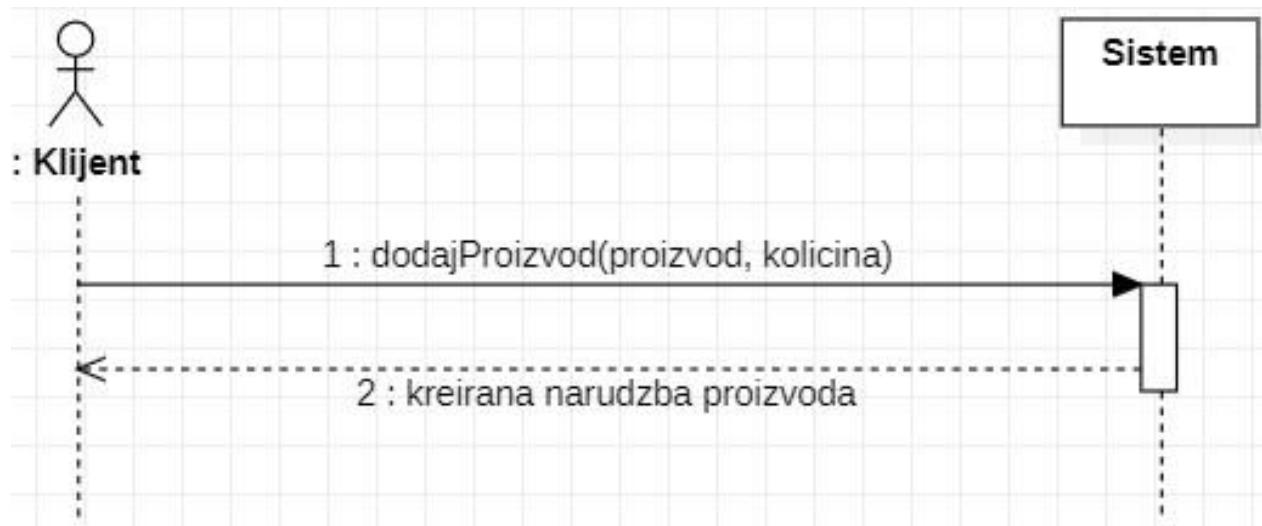
Veza sa SK: SK5

Preduslov: Korisnik je uneo validan program i/ili kategorije

Postuslov: prikazana je lista proizvoda



(Slika 14) Dijagram sekvenci SK6



(Slika 15) Dijagram sekvenci SK6 alternativni scenario

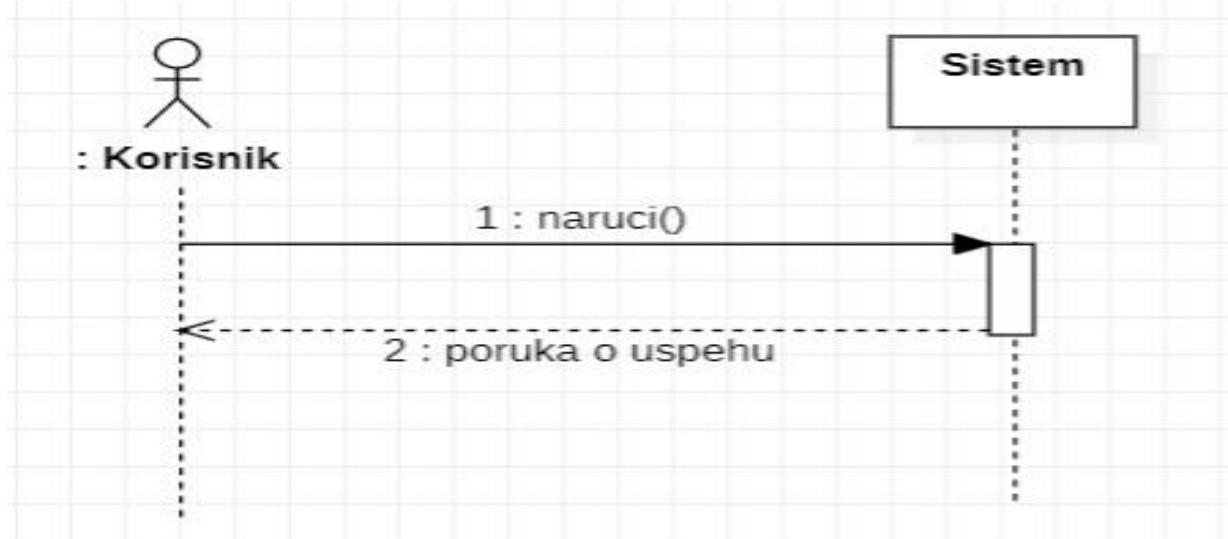
Ugovor UG6:

Operacija: **dodajProizvod(proizvod, kolicina) : void**

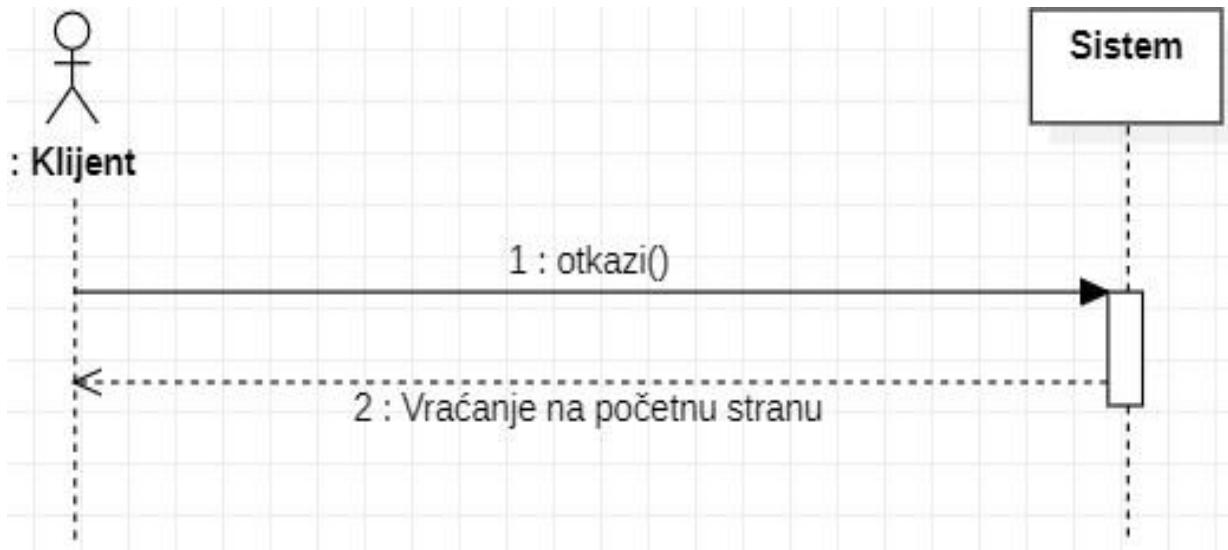
Veza sa SK: **SK6**

Preduslov: korisnik je uneo količinu u interfejsu

Postuslov: dodat je proizvod narudžbini



(Slika 16) Dijagram sekvenci SK7



(Slika 17) Dijagram sekvenci SK7 alternativni scenario

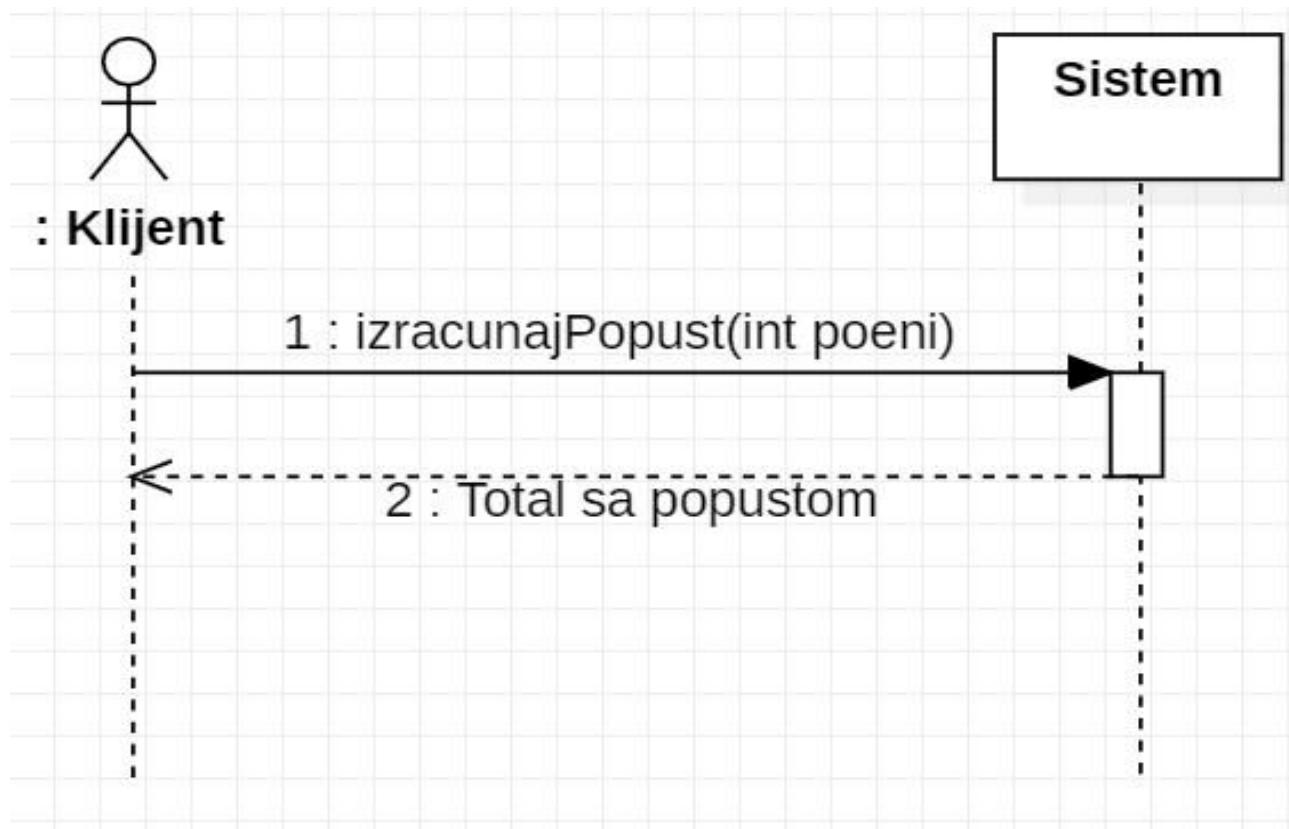
Ugovor UG7:

Operacija: naruci() : void

Veza sa SK: SK7

Preduslov: korisnik je kreirao narudžbinu

Postuslov: narudžbina je dodata u bazu podataka



(Slika 18) Dijagram sekvenci SK8

Alternativni scenario nema dijagram jer ne vrši interakciju sa sistemom, korisnik je sprečen da unese popust ako nema poena.

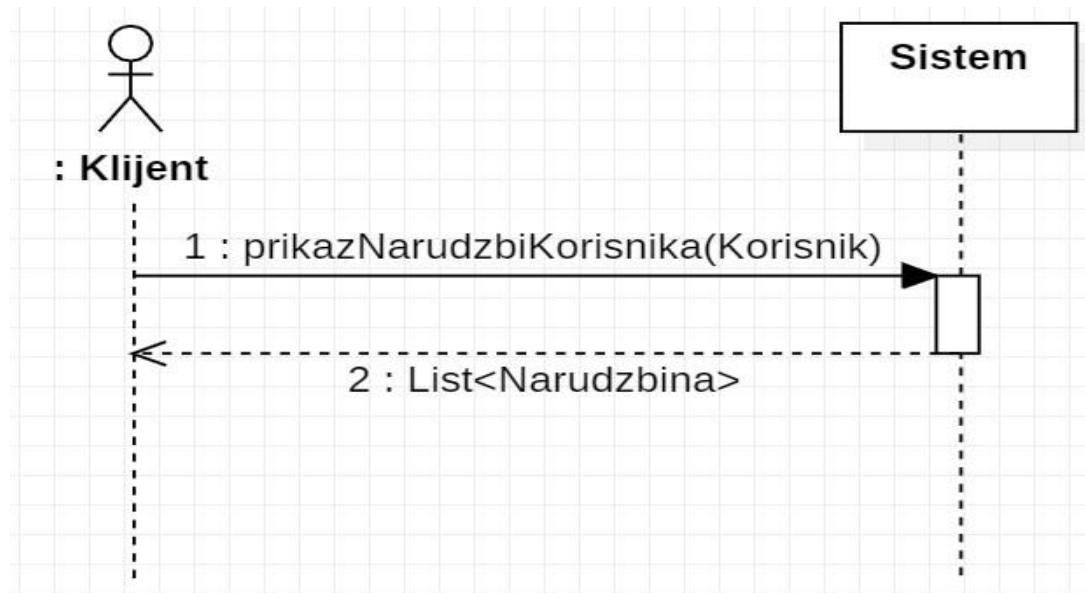
Ugovor UG8:

Operacija: izracunajPopust(int poeni) : void

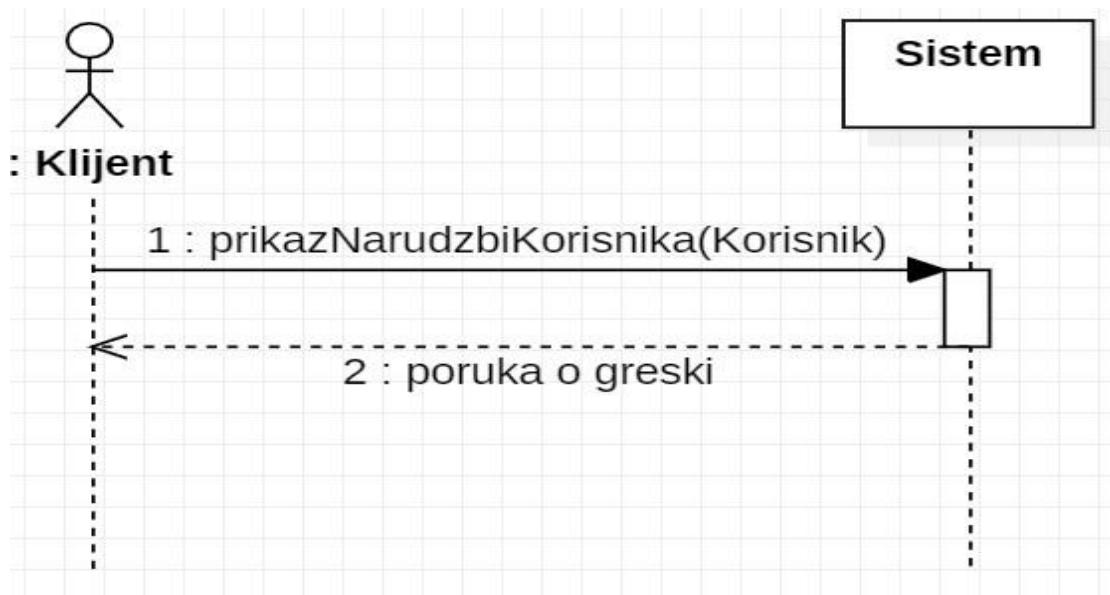
Veza sa SK: SK8

Preduslov: korisnik ima poena

Postuslov: Total je prikazan sa popustom



(Slika 19) Dijagram sekvenci SK9



(Slika 20) Dijagram sekvenci SK9 alternativni scenario

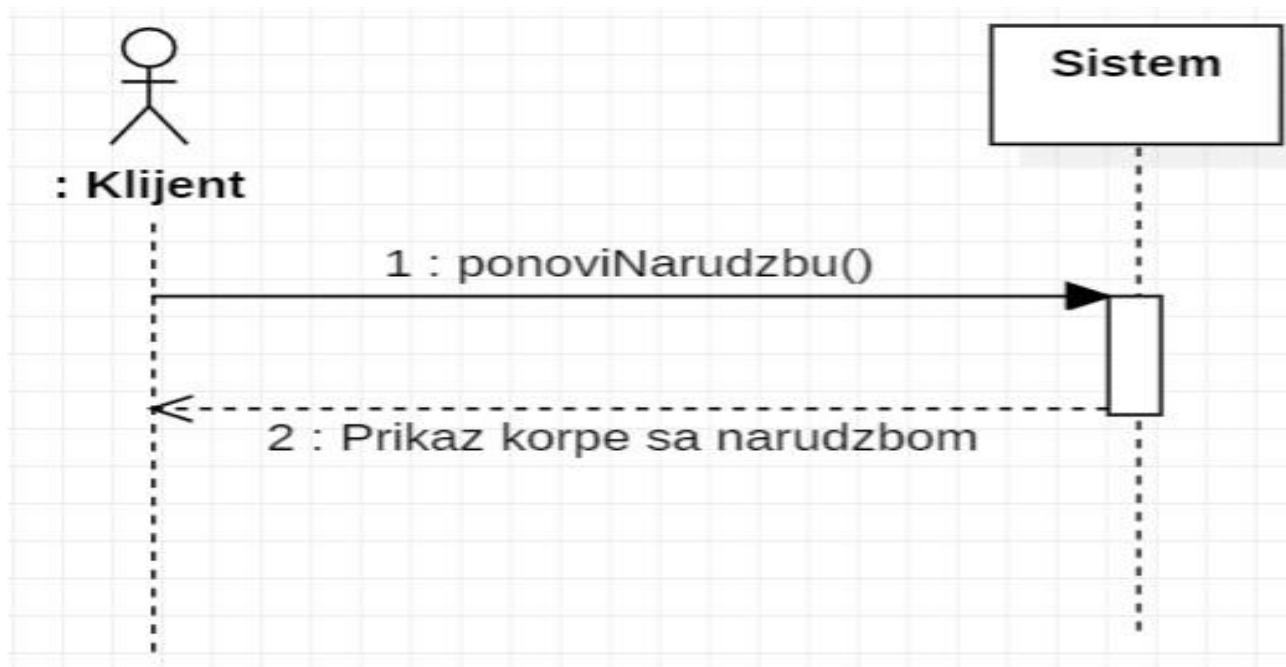
Ugovor UG9:

Operacija: static prikazNarudzbiKorisnika(Korisnik) : List<Narudzbina>

Veza sa SK: SK9

Preduslov: postoji sesija sa korisnikom

Postuslov: prikazane su sve narudzbe korisnika iz trenutne sesije



(Slika 21) Dijagram sekvenci SK10

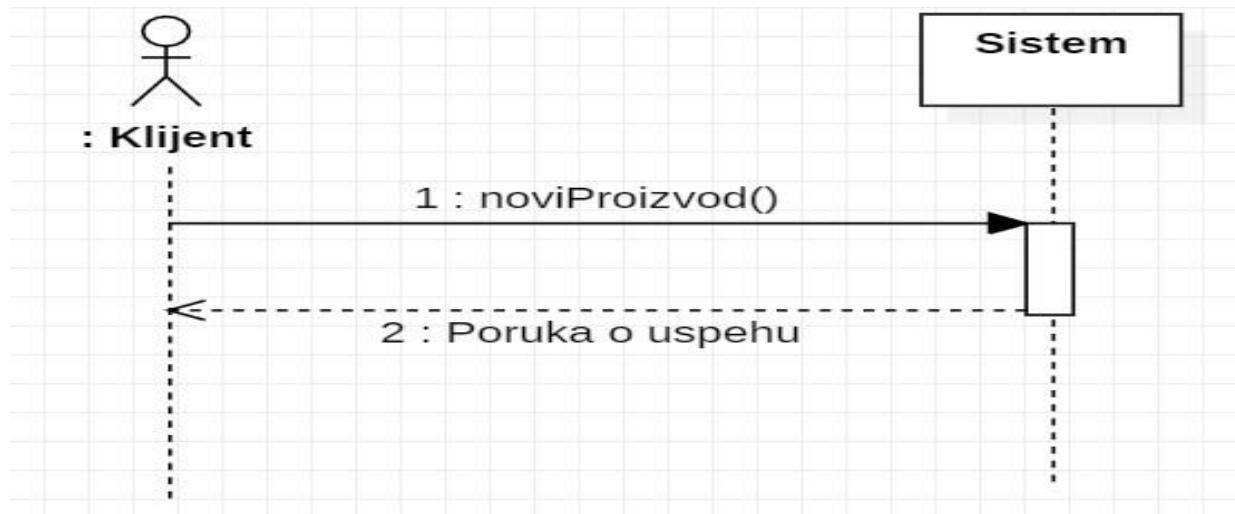
Ugovor UG10:

Operacija: ponoviNarudzbinu()

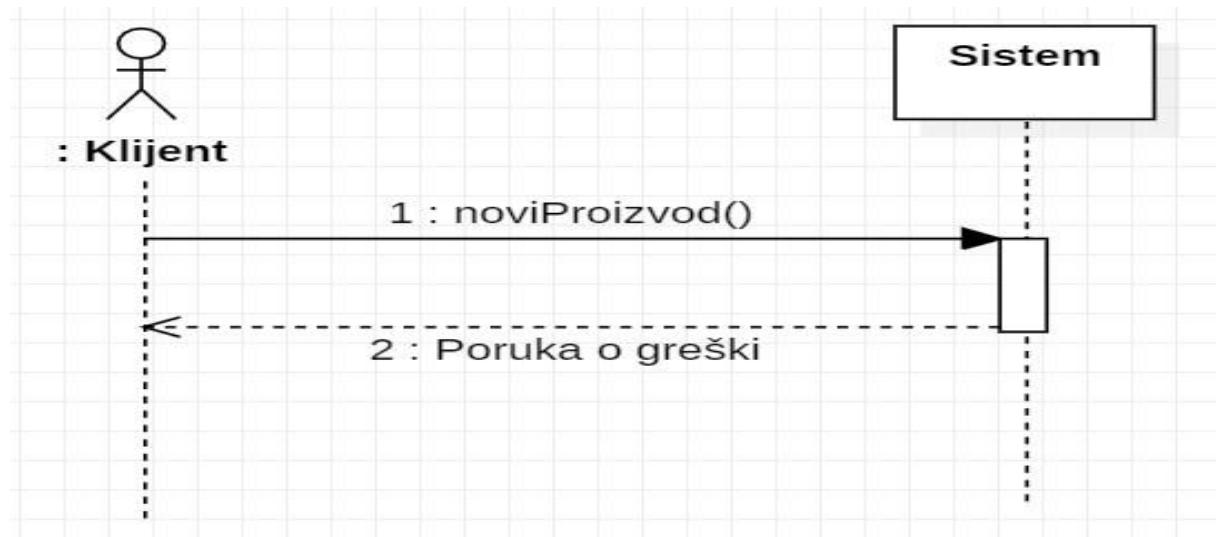
Veza sa SK: SK10

Preduslov: Korisnik je ulogovan

Postuslov: Kopija narudžbine dodata je u sesiju i prikazana u korpi



(Slika 22) Dijagram sekvenci SK11



(Slika 23) Dijagram sekvenci SK11 alternativni scenario

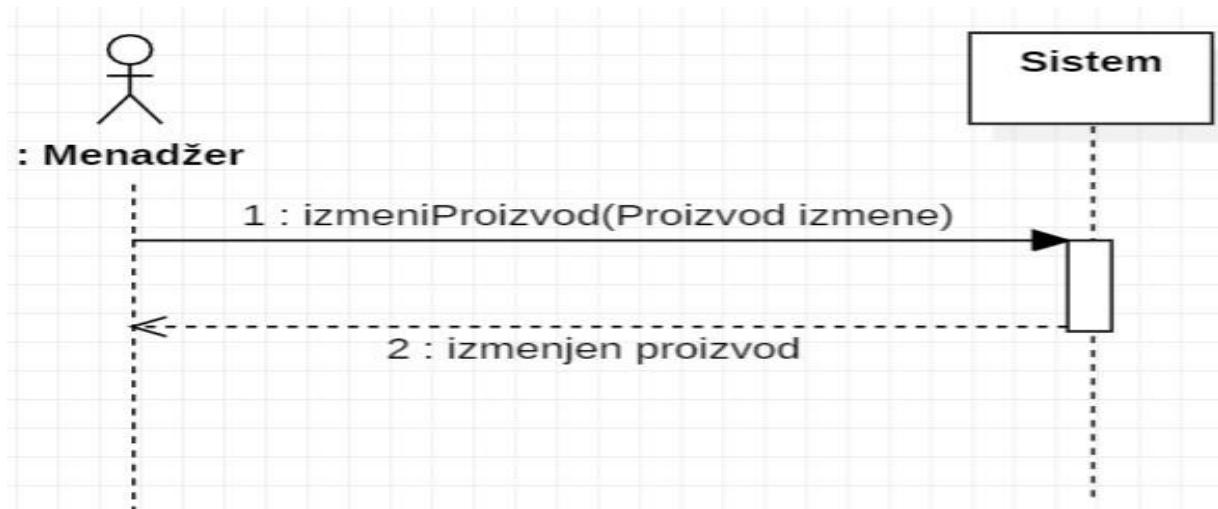
Ugovor UG11:

Operacija: noviProizvod() : void

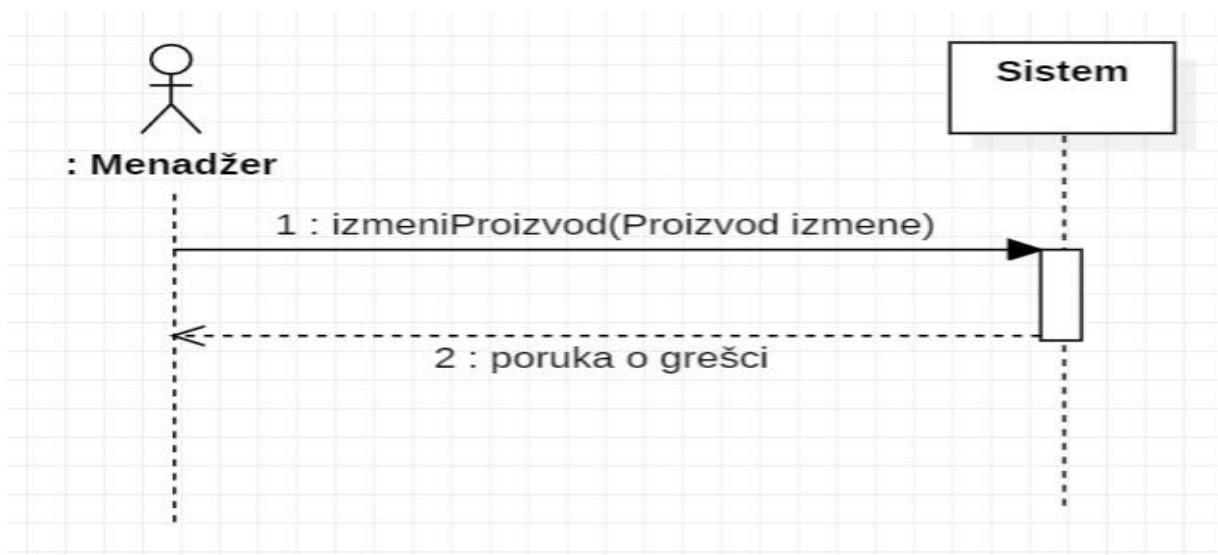
Veza sa SK: SK11

Preduslov: Korisnik je uneo validne podatke na formi za nov proizvod

Postuslov: Dodat je nov proizvod u bazu podataka



(Slika 24) Dijagram sekvenci SK12



(Slika 25) Dijagram sekvenci SK12 alternativni scenario

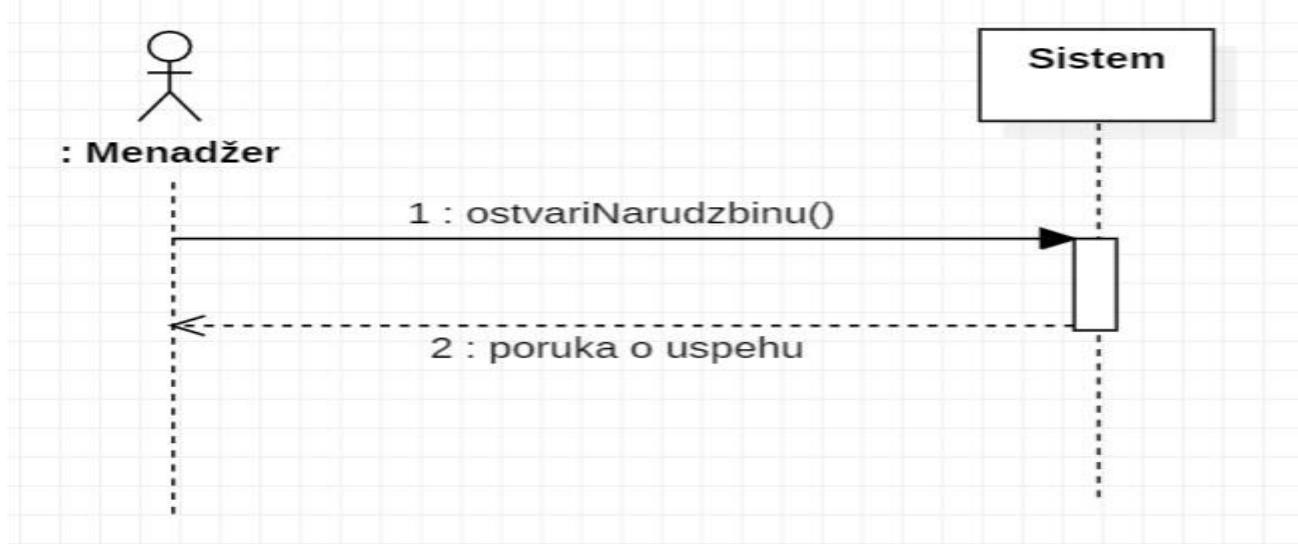
Ugovor UG12:

Operacija: izmeniProizvod(Proizvod izmene) : void

Veza sa SK: SK12

Preduslov: Korisnik je uneo validne podatke na formi za nov proizvod

Postuslov: Izmenjen je proizvod u bazi podataka.



(Slika 26) Dijagram sekvenci SK13

Ugovor UG13:Operacija: `ostvariNarudzbu()` : void

Veza sa SK: SK13

Preduslov: narudžba postoji

Postuslov: menja se vrednost ostvarena zadate narudžbine u bazi podataka



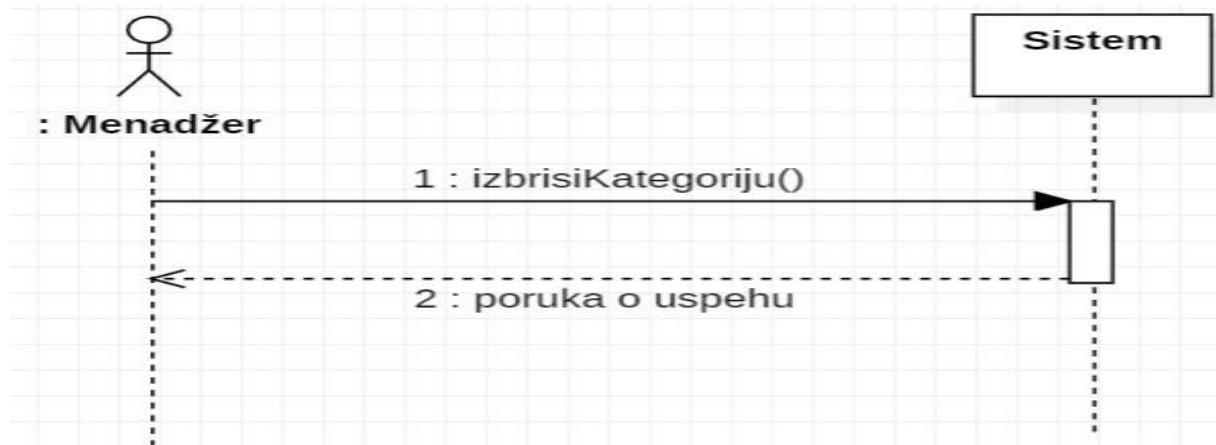
(Slika 27) Dijagram sekvenci SK14

Ugovor UG14:Operacija: `dodajKategoriju()` : void

Veza sa SK: SK14

Preduslov: Ne postoji kategorija sa istim nazivom

Postuslov: Dodata je nova kategorija



(Slika 28) Dijagram sekvenci SK15

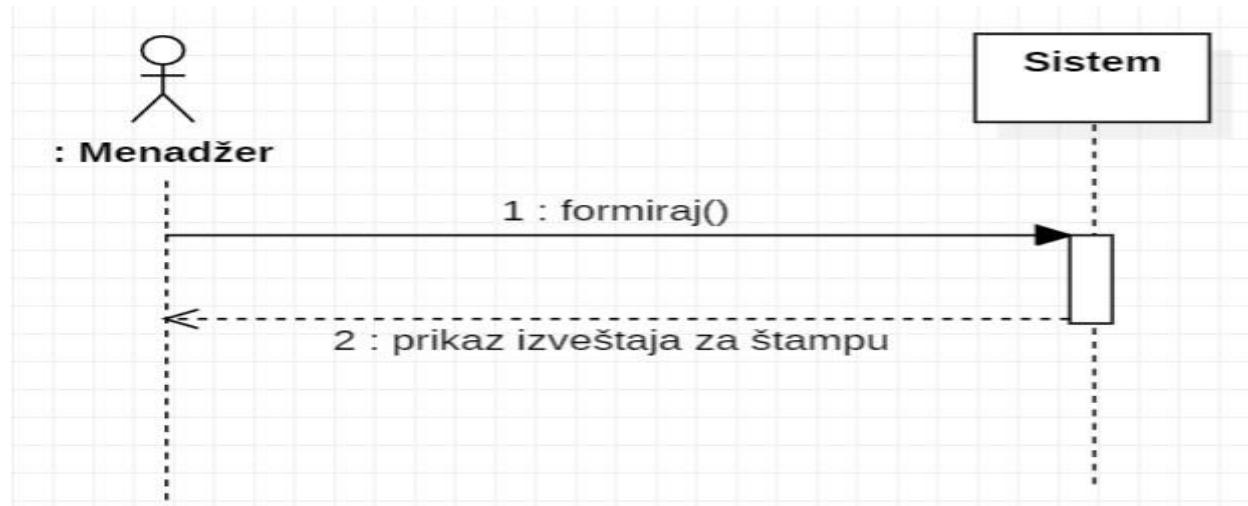
Ugovor UG15:

Operacija: izbrisikategoriju() : void

Veza sa SK: SK15

Preduslov: kategorija postoji

Postuslov: Zadata kategorija izbrisana je iz baze podataka



(Slika 29) Dijagram sekvenci SK16

Ugovor UG16:

Operacija: Izvestaj.formiraj() : void

Veza sa SK: SK16

Preduslov: nema

Postuslov: Prikazana je strana za štampanje.



(Slika 30) Dijagram sekvenci SK17



(Slika 31) Dijagram sekvenci SK17 alternativni scenario

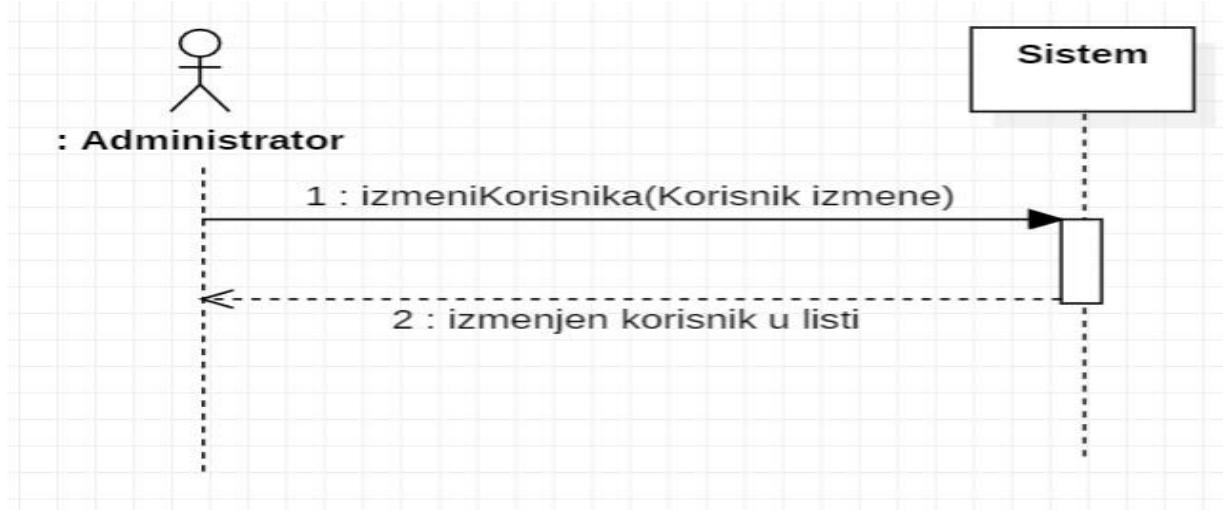
Ugovor UG17:

Operacija: korisnik.registruj() : void

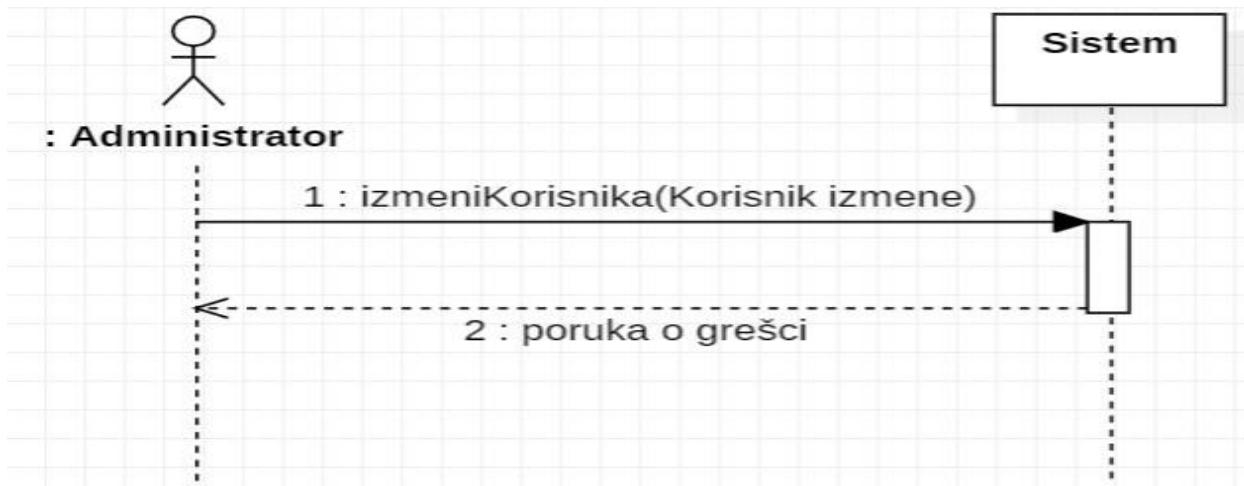
Veza sa SK: SK17

Preduslov: Administrator je uneo validne podatke u formu za novog korisnika

Postuslov: dodat je nov korisnik u bazi



(Slika 32) Dijagram sekvenci SK18



(Slika 32) Dijagram sekvenci SK18 alternativni scenario

Ugovor UG18:

Operacija: izmeniKorisnika(Korisnik izmene) : void

Veza sa SK: SK18

Preduslov: postoji korisnik sa zadatim korisnickim imenom i administrator je uneo validne podatke za izmenu

Postuslov: Izmenjeni su podaci zadatog korisnika u bazi podataka



(Slika 33) Dijagram sekvenci SK19

Ugovor UG19:

Operacija: izbrisikorisnika(korisnik) : void

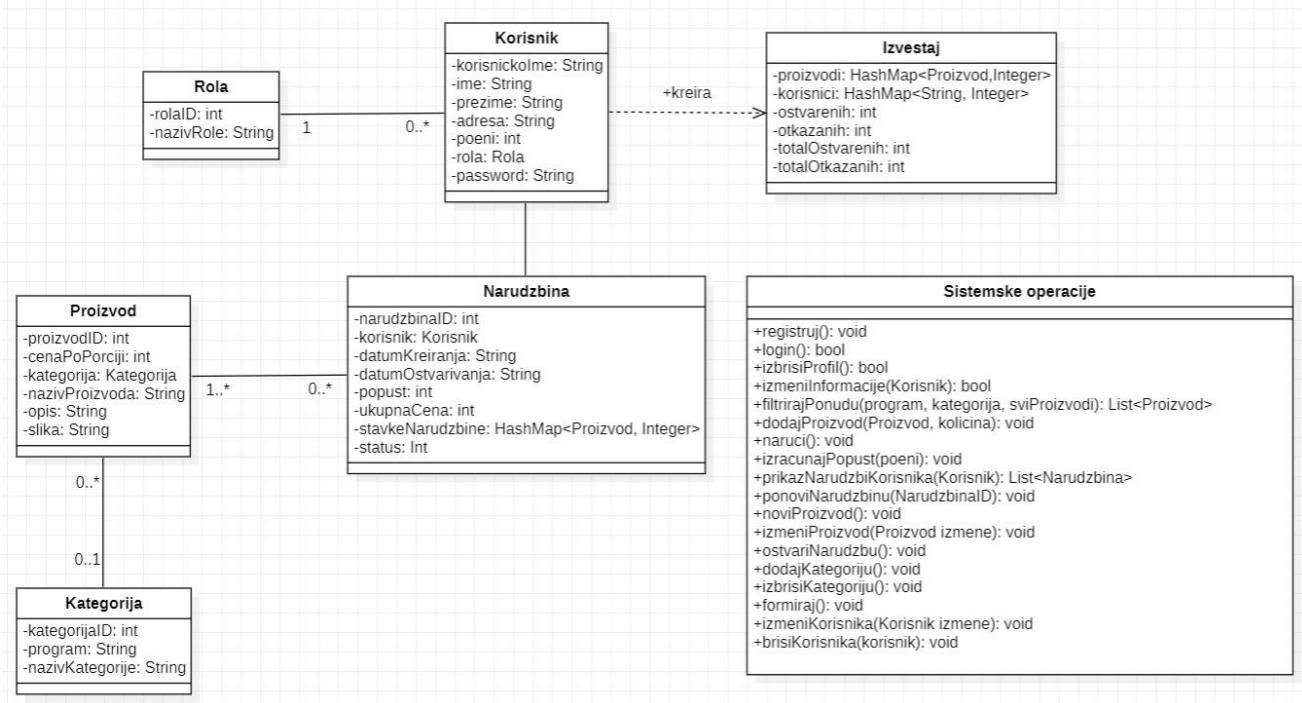
Veza sa SK: SK19

Preduslov: korisnik postoji

Postuslov: zadati korisnik je izbrisani iz baze podataka.

4. KONCEPTUALNI MODEL

4.1 Dijagram klasa



(Slika 34) Dijagram klasa

4.2 Relacioni model

Korisnici (KorisnickoIme, Ime, Prezime, Adresa, Poeni, PasswordHash, RolaID)

FK RolaID referencira Role(PK RolaId)

Proizvodi (ProizvodID, NazivProizvoda, Opis, Slika, CenaPoPorciji , KategorijaID)

FK KategorijaID referencira Kategorije(PK KategorijaID)

Narudzbine (NarudzbinID, KorisnickoIme, Popust, DatumKreiranja, DatumOstvarivanja, UkupnaCena, Status)

FK KorisnikID referencira Korisnici(PK KorisnikID)

StavkeNarudzbine (ProizvodID, NarudzbinID, Kolicina)

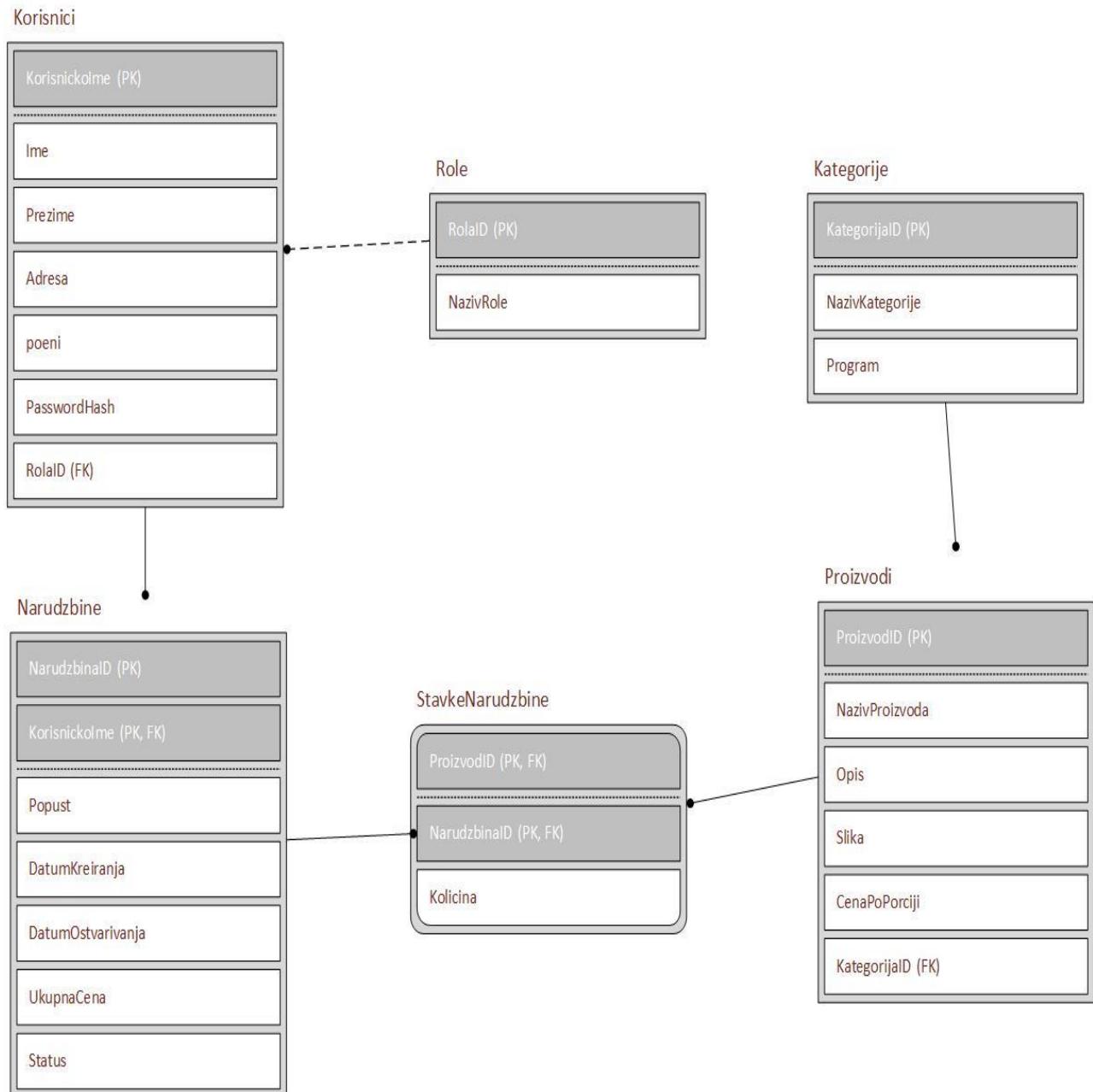
FK Proizvod referencira Proizvodi(PK ProizvodID)

FK NarudzbinID referencira Narudzbine(PK NarudzbinID)

Kategorije (KategorijaID, NazivKategorije, Program)

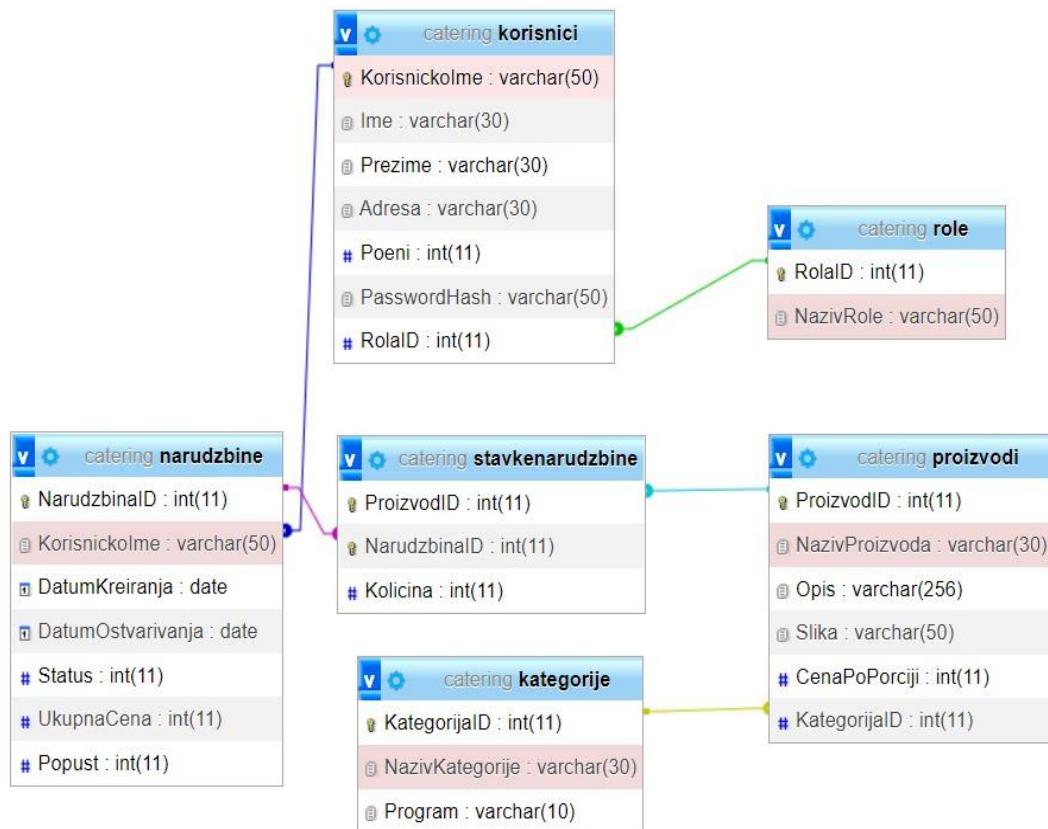
Role (RolaID, NazivRole)

4.3 IDEF1X dijagram baze podataka



(Slika 35) IDEF1X model

5.4 Implementacija baze



(Slika 36) Tabele i relacije u bazi

```

CREATE TRIGGER `Izbrisistavke` BEFORE DELETE ON `proizvodi`
FOR EACH ROW UPDATE stavkenarudzbine SET `ProizvodID` = 0 WHERE OLD.`ProizvodID` = stavkenarudzbine.ProizvodID

CREATE TRIGGER `Izbrisinarudzbe` BEFORE DELETE ON `korisnici`
FOR EACH ROW UPDATE narudzbine SET narudzbine.KorisnickoIme = 'izbrisani' WHERE narudzbine.KorisnickoIme = OLD.KorisnickoIme

CREATE TRIGGER `IzbrisikategorijeProizvoda` BEFORE DELETE ON `kategorije`
FOR EACH ROW UPDATE `proizvodi` SET `KategorijaID` = 0 WHERE OLD.`KategorijaID` = proizvodi.KategorijaID

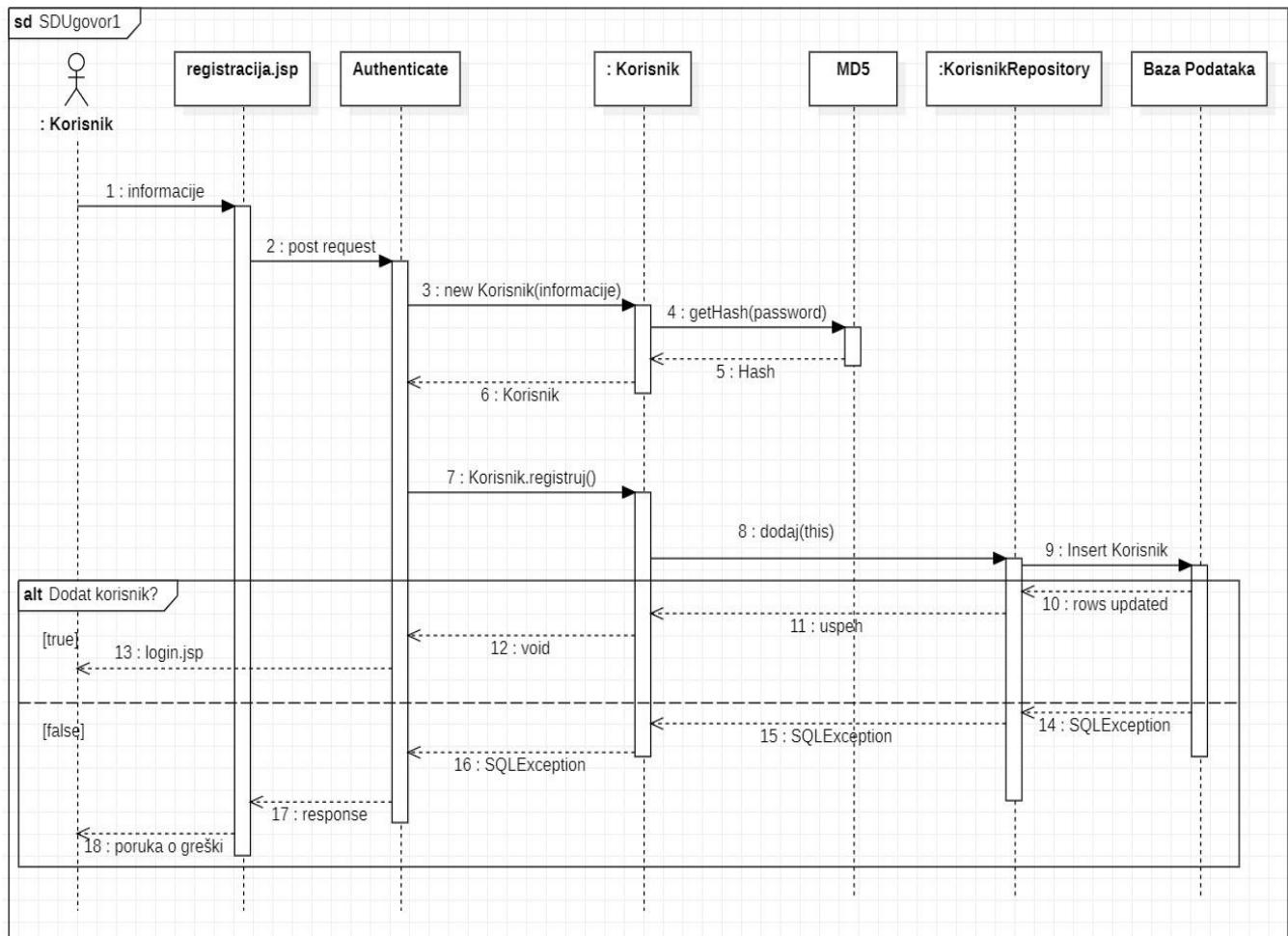
```

(Slika 37) Triggeri u bazi

5. DIJAGRAMI SEKVENCI

***Napomena:** Kreiranje objekta Connection ka bazi vrši se u static menadžeru konekcije koji poziva Repository. Ovo nije modelovano na predstojećim dijagramima zbog bolje preglednosti.

5.1 Registracija



(Slika 38) Dijagram sekvenci ugovora UG1

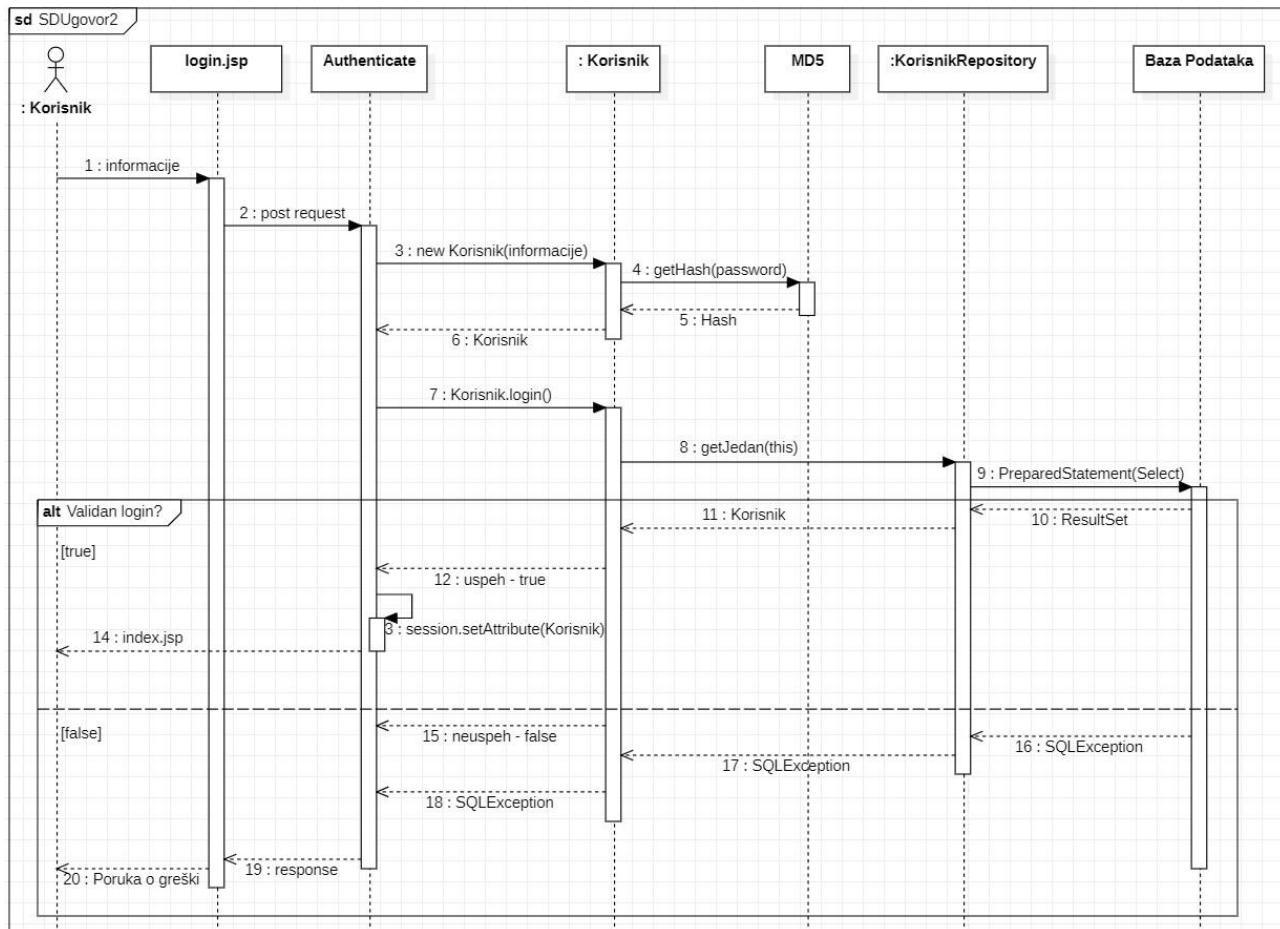
Proces registracije predstavlja planiranu implementaciju UG1 i use case-a Registracija.

Korisnik unosi podatke na View-u registracija.jsp koji se post requestom prenose Controlleru, u ovom slučaju Authenticate.java. Controller kreira novog korisnika i poziva metodu registruj() nad njim, time se sama logika implementacije prenosi na Model, odnosno klasu Korisnik.

Klasa Korisnik komunikaciju sa bazom delegira KorisnikRepository objektu, za detaljniji opis pogledati poglavlje 8.

Na dijagramu su predstavljena dva alternativna toka nakon interakcije sa bazom, jedan gde je dodavanje uspesno pa Controller poziva prikaz login.jsp stranice i jedan gde se baca SQLException koji se dodaje unazad na call stack-u do Controllera (Authenticate.java) koji pakuje grešku u request i vraća View-u (registracija.jsp) za prikazivanje.

5.2 Login



(Slika 39) Dijagram sekvenci ugovora UG2

Proces login predstavlja planiranu implementaciju UG2 i use case Login-a.

Korisnik unosi svoje informacije u View-u login.jsp koji ih pakuje u post request i predaje Authenticate.java Controller-u. Authenticate.java kreira novog korisnika i poziva metodu login() nad njim.

Korisnik zatim poziva Repository pozivom metode getJedan() predajući sebe kao parametra, rezervisanom reči "this". Repository vrši query baze i vraća rezultat klasi Korisnik.

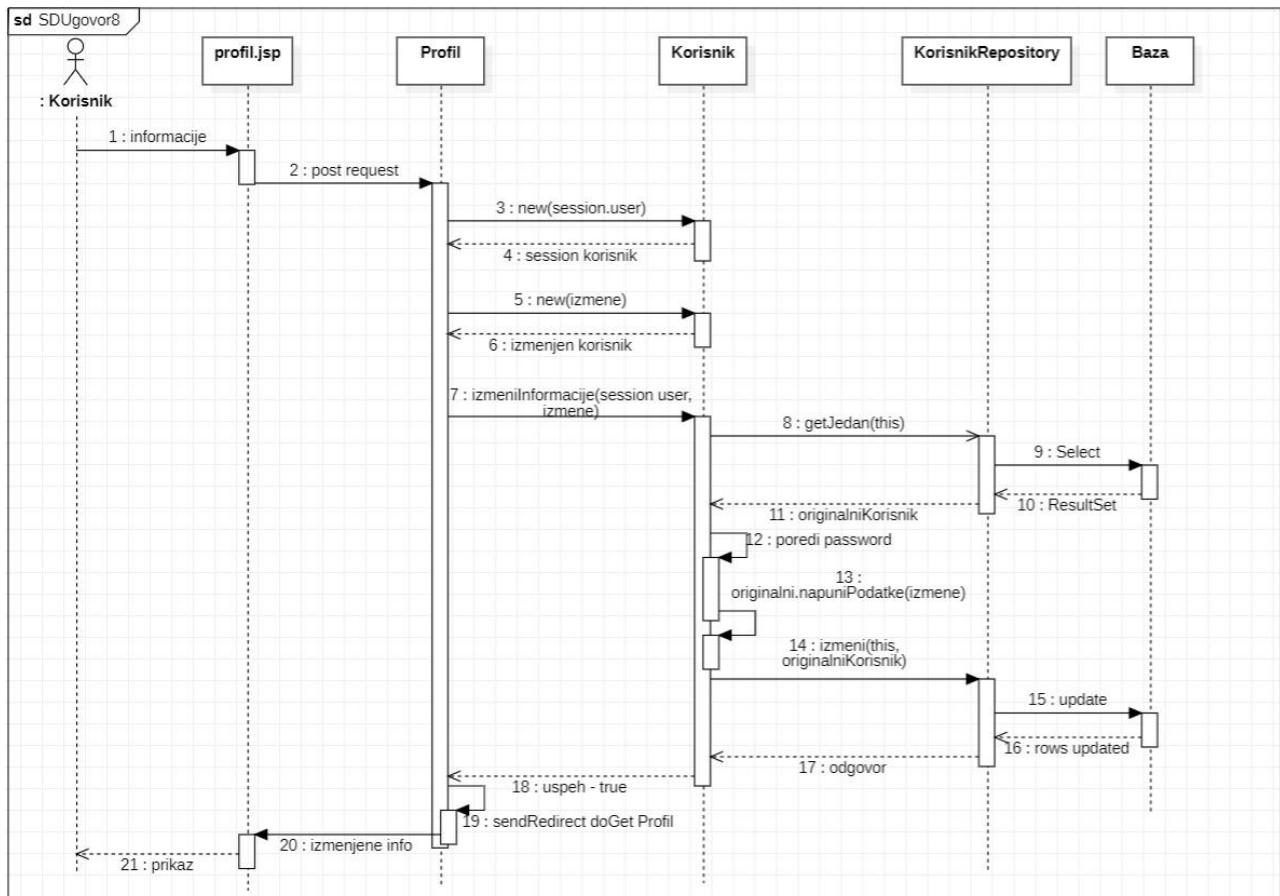
Zatim se poređi password hash instance Korisnika koji je kreirao kontroler sa inputom korisnika(unesenim korisničkim imenom i šifrom) i password hash-om instance Korisnika koji je vratio repository iz baze.

Ukoliko se hash-ovi poklapaju, login metoda pakuje informacije korisnika dobijene iz baze u svoju instancu i vraća true.

Controller zatim pokreće sesiju i pakuje u nju instancu Korisnika nad kojom je pozvao login(), a koja sada ima sve relevantne podatke. Korisnik je ulogovan.

Ukoliko se hash-ovi ne poklapaju, login metoda vraća false i kontroler u request pakuje poruku o greški i šalje login.jsp view-u na prikaz. Takođe može doći do SQLExceptiona koji se vraća na call stack-u do kontrolera i pakuje kao poruka o greški. Na dijagramu ova dva slučaja su oba prikazana u istom alt – false prostoru radi preglednosti.

5.3 Izmena informacija



(Slika 40) Dijagram sekvenci use case-a Izmene informacija

Proces izmena informacija predstavlja planiranu implementaciju UG4 iliti use case-a Izmene Informacija.

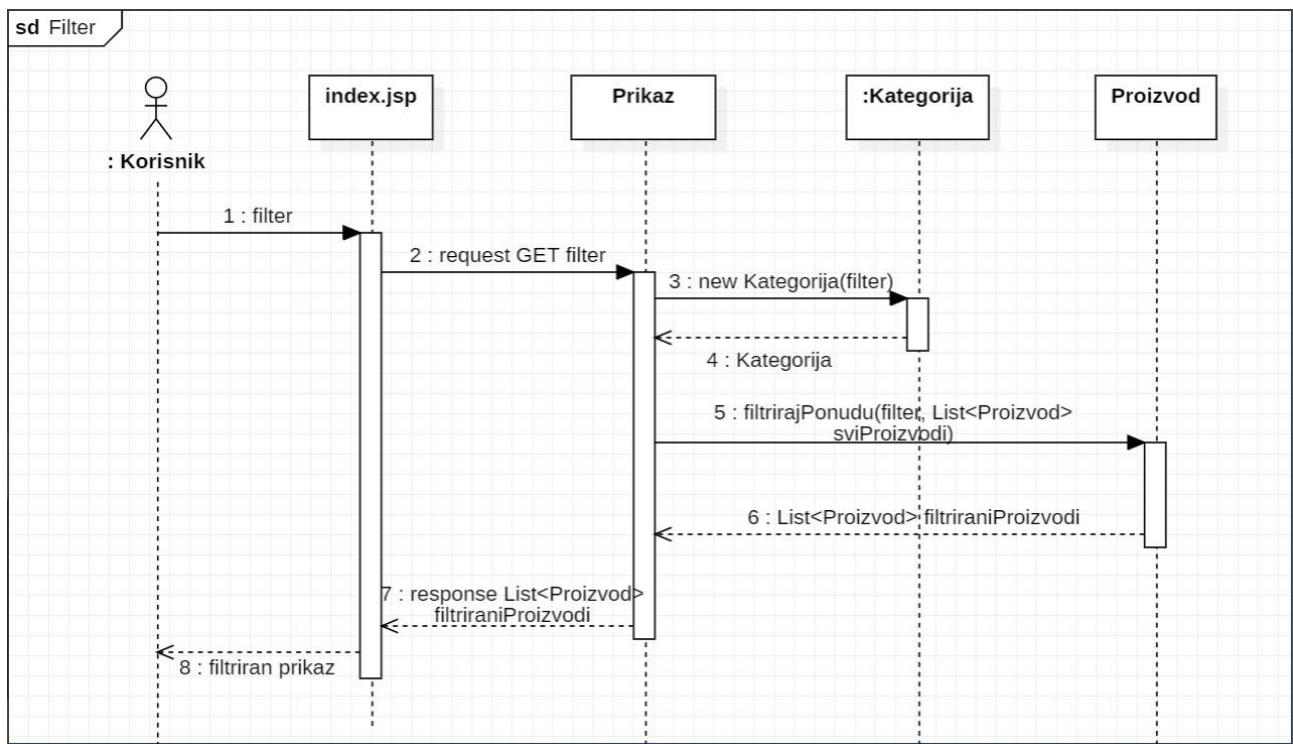
Korisnik menja podatke koje želi da izmeni u formi za prikaz profila. Profil.jsp šalje post request Profil Controller-u u koji su spakovani podaci sa forme. Controller kreira dve instance klase Korisnik, jednu sa usernameom ulogovanog korisnika(iz sesije), a jednu sa podacima iz forme za izmenu (request parametara), zatim predaje te dve instance Modelu na obradu pozivom metode izmeniInformacije nad instancom iz sesije.

Instanca korisnika iz sesije poziva Repository metodu getJedan i kao parametar predaje sebe, da bi dobili password hash i ostale informacije vezane za korisnika iz baze. Zatim se password hash poredi sa hashom passworda koji je korisnik uneo u formi za izmenu.

Ukoliko se hash-ovi ne poklapaju interakcija se prekida i korisniku se prikazuje greška, takođe se greška prikazuje ukoliko dodje do SQLException-a u interakciji sa repositorijem. Ovo ponašanje na slici 40 nije modelovano, već je modelovan samo uspešna interakcija, radi čitljivosti dijagrama.

Ako se hash-ovi poklapaju instanca Korisnika dobijena iz baze puni se izmenjenim podacima I predaje reperotoriju koji vrši updejt baze. Ukoliko ne dodje do exceptiona neke vrste, Model Controlleru vraća true. Profil controller zatim poziva svoju get metodu sendRedirect-om da bi prikazao formu profila sa izmenjenim podacima, takođe šalje i poruku o uspehu koja se prikazuje ispod forme.

5.4 Filter



(Slika 41) Dijagram sekvenci use case-a filtriranja

Proces filter prestavlja planiranu implementaciju UG5 ili use case-a filtriranja ponude.

Korisnik bira kategoriju ili program(slatki ili slani) koji želi da pregleda na View-u index.jsp koji služi kao početna strana, a i strana za prikaz proizvoda.

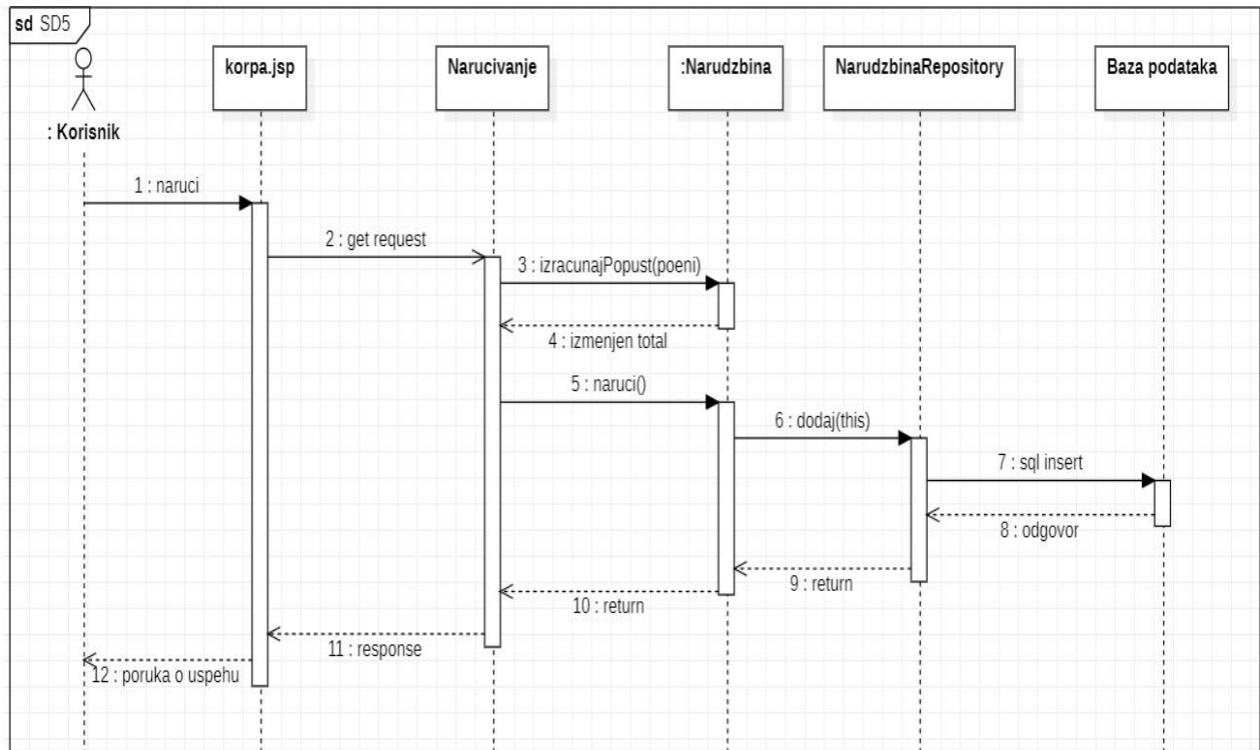
View šalje get request sa odgovarajućom kategorijom ili programom u url-u, Controller-u Prikaz.java koji ako je kriterijum filtera kategorija, kreira novu kategoriju sa tim nazivom I pakuje je zajedno sa listom svih proizvoda kao parametre za poziv staticke metode filtrirajPonudu klase Model-a Proizvod.

Proizvod filtrira datu listu po zadatom kriterijumu i vraća listu filtriraniProizvodi kontroleru. Kontroler je pakuje u response i vraća index.jsp-u za prikaz.

Proces ukoliko je kriterijum program je identičan osim što kontrol ne kreira instancu klase modela Kategorija već prosleđuje null.

Lista proizvoda koju controller Prikaz predaje Proizvodu kao argument metoda filtrirajPonudu Controller posjeduje jer poziva Repository getSve() u procesu koji nije modelovan ovim dijagramom.

5.5 Naručivanje



(Slika 42) Dijagram sekvenci use case-a naručivanja

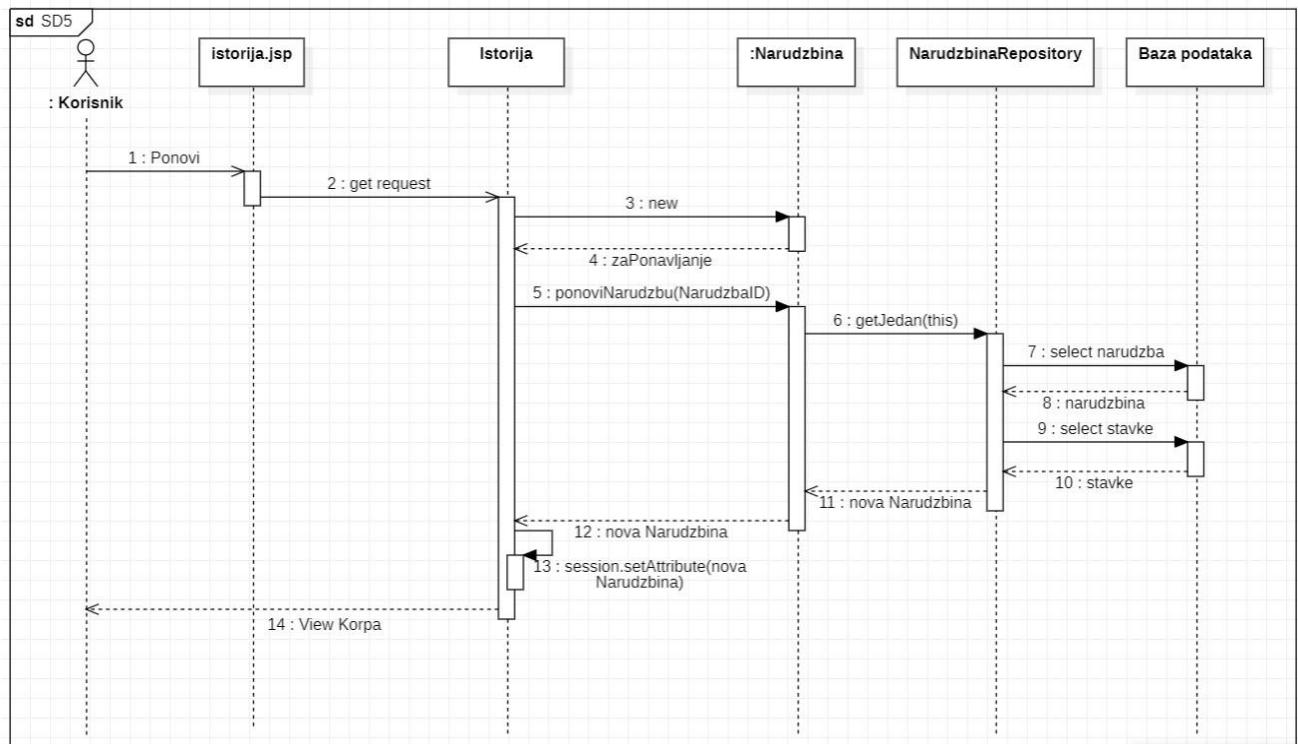
Proces naručivanja predstavlja planiranu implementaciju UG7 ili use case-a naručivanja.

Korisnik poziva naručivanje iz prikaza korpe na korpa.jsp-u. View šalje get request Controlleru Narucivanje, koji pakuje narudžbinu iz sesije u instancu klase Narudzbina i nad njom poziva metodu izracunajPopust i predaje broj poena iz parametra get request-a.

Narudžbina računa total sa popustom i setuje ukupnaCena i popust polja, nad takvom instancom Controller poziva metodu naruci() koja predaje instancu nad kojom je pozvana NarudzbinaRepository-ju za dodavanje u bazu.

Nakon što baza obavi insert, return-a se do Controllera Narucivanje koji poziva korpa.jsp da prikaže poruku zahvalnice i ostvarenih poena.

5.6 Ponavljanje narudžbi



(Slika 43) Dijagram sekvenci use case-a ponavljanja narudžbina

Proces ponavljanja narudžbi predstavlja planiranu implementaciju UG10 ili use case-a ponavljanja naručivanja.

Korisnik na View-u za prikazivanje istorije svojih narudžbi klikće na dugme ponovi na jednoj od narudžbi sa ostvarenim ili otkazanim statusom (Narudžbe koje su još uvek u pripremi mogu da se otkažu, umesto da se ponovo naruče).

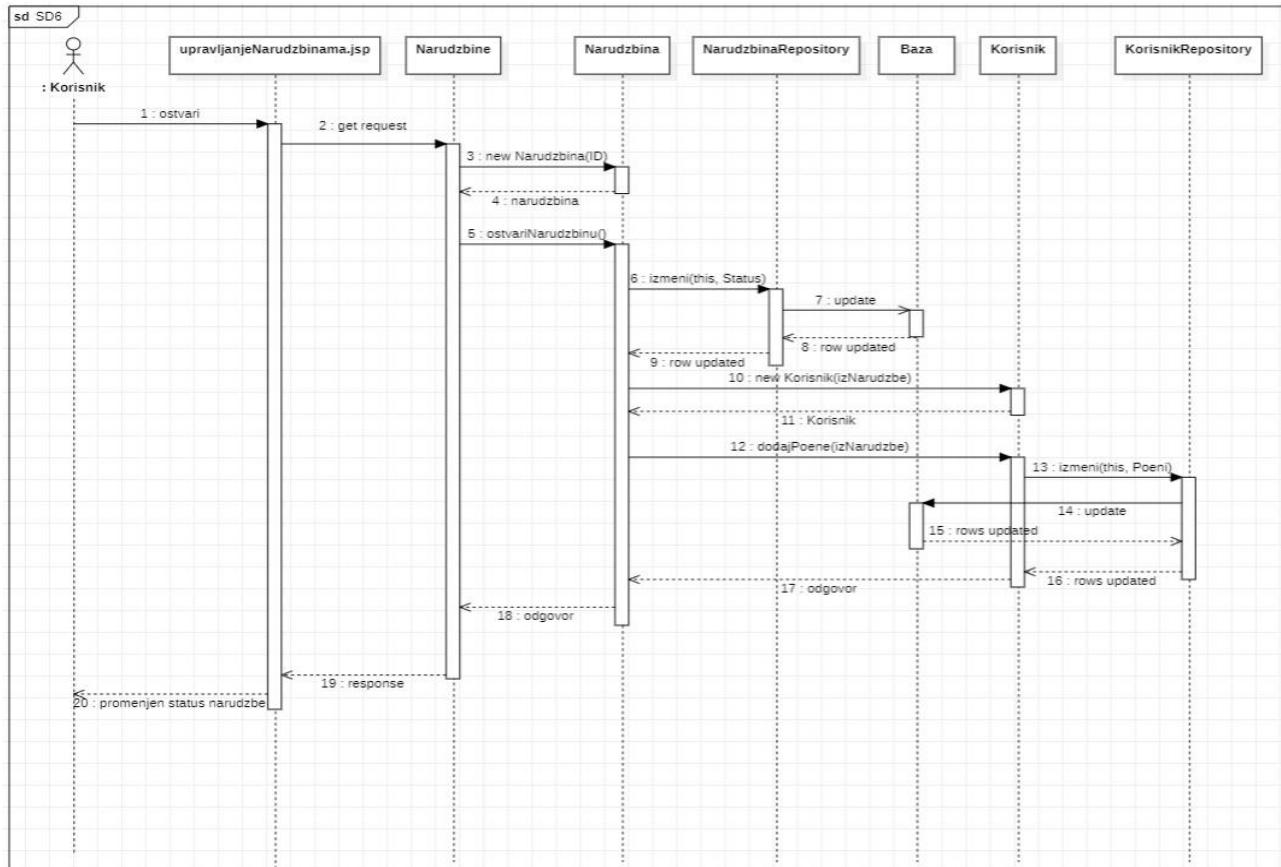
View istorija.jsp Controller-u Istorija šalje get request sa ID-em narudžbe koju korisnik želi da ponovi.

Controller kreira instancu Model-a Narudzbina i nad njom poziva metodu ponoviNarudzbinu, sa parametrom ID-a iz request-a. Narudzbina predaje sebe kao parametar metodi getJedan NarudzbinaRepository-a.

Repository prvo vrš select upit nad bazom da dobije red iz narudzbine tabele, a zatim vrši još jedan select nad Join-ovanom tabelom stavkenarudzbina i proizvodi, da bi napunio HashMap-u stavkeNarudzbine.

Ovakav objekat Narudzbine Repository vraća nazad, instanca Narudzbine preuzima potrebne podatke rezultata i vraća se u Controller. Istorija servlet pakuje tu instancu u atribut "narudzbina" sesije i poziva prikaz korpe.jsp gde se prikazuje kopija narudžbe koju je korisnik zatražio za ponavljanje. Korisnik je može dalje modifikovati, ili naručiti identičnu.

5.7 Ostvarivanje narudžbi



(Slika 44) Dijagram sekvenci use case-a ostvarivanja narudzbi

Proces ostvarivanja narudžbe predstavlja planiranu implementaciju UG13 ili use case-a ostvarivanja narudžbi.

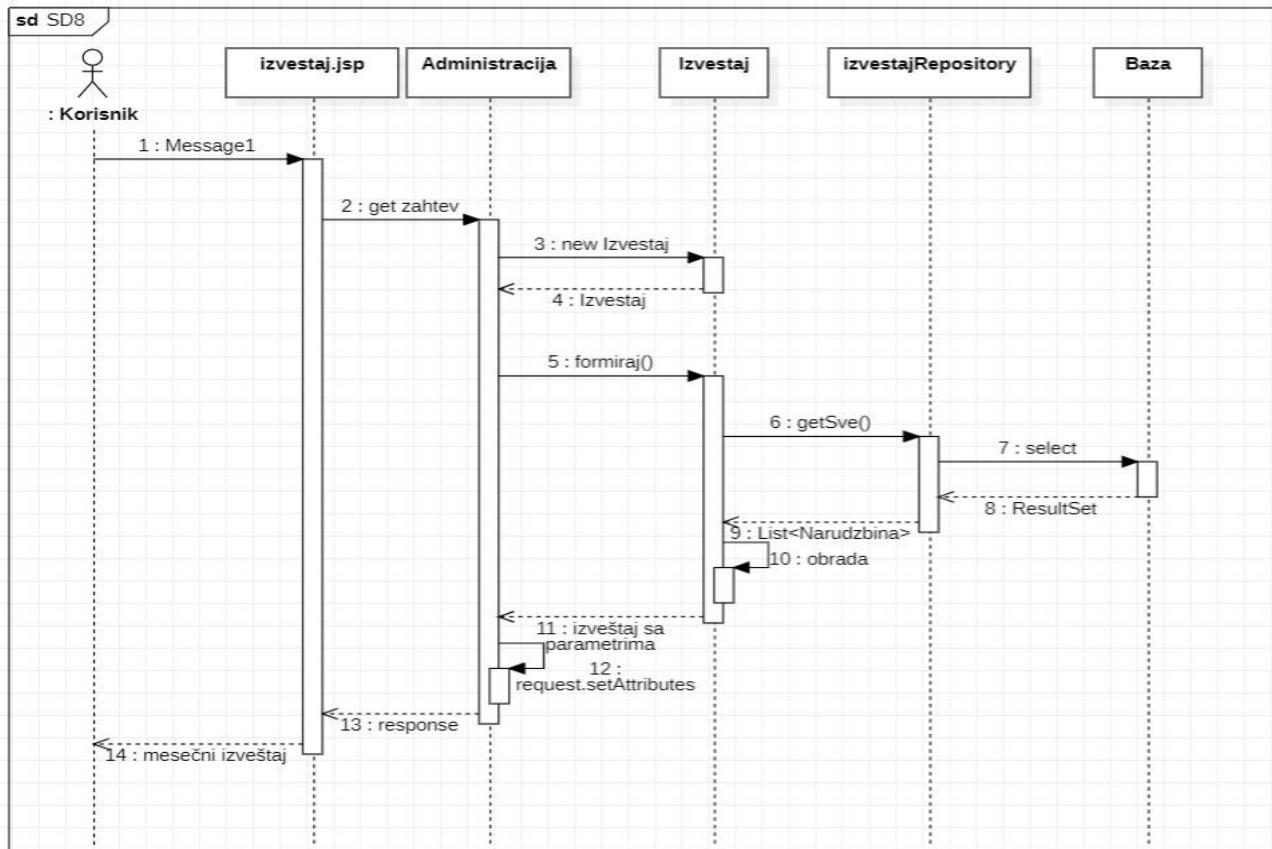
Korisnik(Menadžer ili Administrator) na view-u za upravljanje narudžbinama bira narudžbu za ostvarivanje, View šalje get request Controller-u Narudžbine, koji kreira objekat Narudzrina-e sa zadatim ID-em i nad njim poziva metodu ostvariNarudzbu().

Ova metoda kreira još jedan objekat narudžbe, menja mu status u 1(ostvarena) i predaje NarudzrinaRepository-ju kao drugi parametar metode izmeni(stari, novi).

Repository vrši update baze, metoda Model-a Narudzrina nastavlja sa izvršavanjem i kreira objekat Korisnik-a, i predaje mu broj poena ostvarenih narudžbinom metodom dodajPoene(poeni). Korisnik tada poziva svoj Repository i kao drugi parametar metode izmeni(stari, novi) predaje kopiju sebe sa dodatim ostvarenim poenima iz narudžbine.

KorisnikRepository vrši update baze, i izvršavanje svih metoda se završava i vraća nazad do Narudžbine Controllera, koji refreshuje view upravljanjeNarudzbinama.jsp i korisniku se prikazuje narudžbina sa novim statusom (1 – ostvarena).

5.8 Kreiranje izveštaja



(Slika 45) Dijagram sekevenci use case-a kreiranja izveštaja

Proces kreiranja izveštaja predstavlja planiranu implementaciju UG16 ili use case-a kreiranja izveštaja.

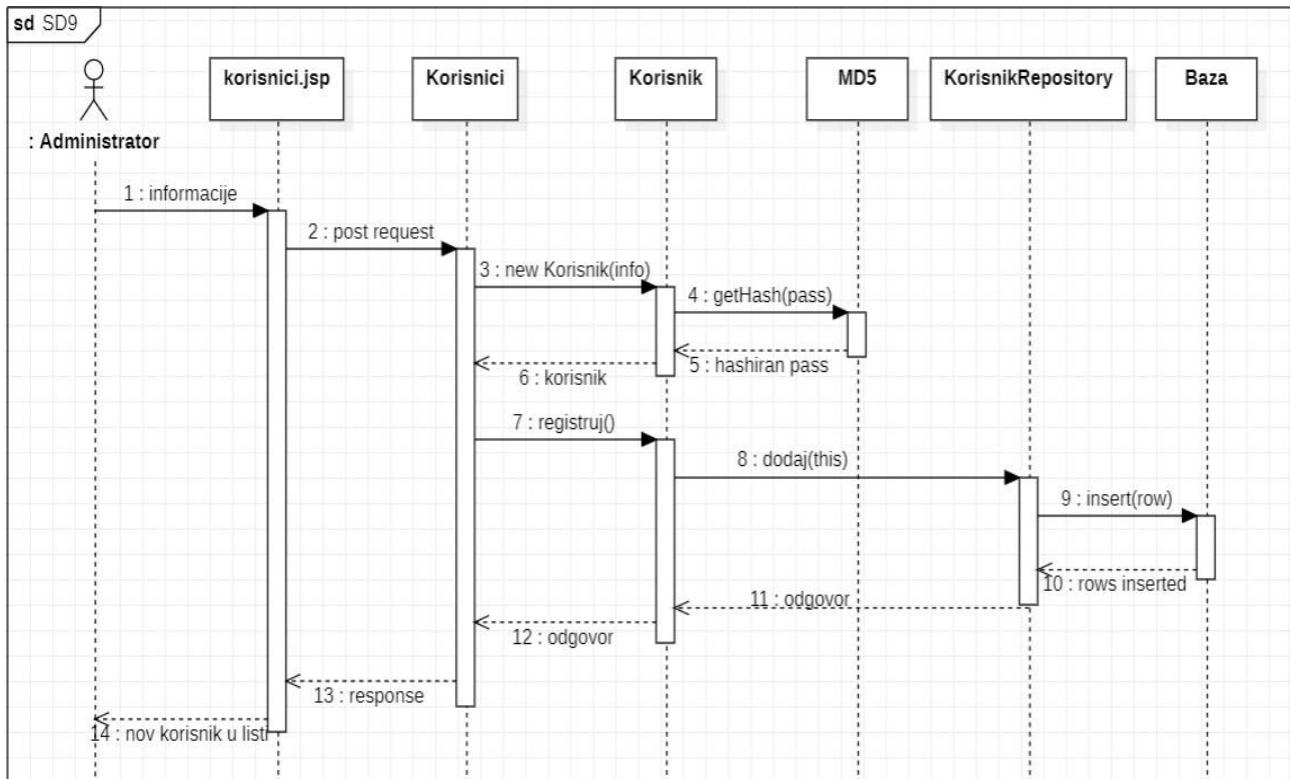
Kada korisnik otvori izvestaj.jsp View, šalje se get request Administracija Controller-u koji kreira nov objekat Izvestaj-a i nad njim poziva metodu formiraj().

Koristeći IzvestajRepository metodu getSve() Izvestaj dobija listu svih narudžbi i njihovih stavki iz trenutnog meseca. Izvestaj zatim prolazi kroz sve narudžbe i njihove stavke i puni dve HashMap-e, jednu koja čuva količinu pojedinačnih proizvoda prodatih u mesecu i drugu koja čuva ukupnu potrošnju svih korisnika koji imaju potrošnju u mesecu. Takođe puni 4 integer vrednosti za praćenje broja i totala otkazanih i ostvarenih narudžbi.

Nakon što se završi izvršavanje formiraj() metode, nastavlja se izvršavanje servleta koji HashMap-e i integer vrednosti pakuje kao request Parametre i predaje nazad izvestaj.jsp stranici za prikaz.

Korisniku se prikazuje mesečni izveštaj i mogućnost da isti download-uje u pdf formatu.

5.9 Dodavanje korisnika



(Slika 46) Dijagram sekvenci use case-a dodavanja korisnika

Proces dodavanja korisnika predstavlja planiranu implementaciju UG17 ili use case-a dodavanja korisnika.

Administrator na view-u korisnici.jsp unosi podatke novog korisnika i zatražuje da se korisnik doda u sistem. View šalje post request Korisnici Controlleru sa relevantnim podacima iz forme.

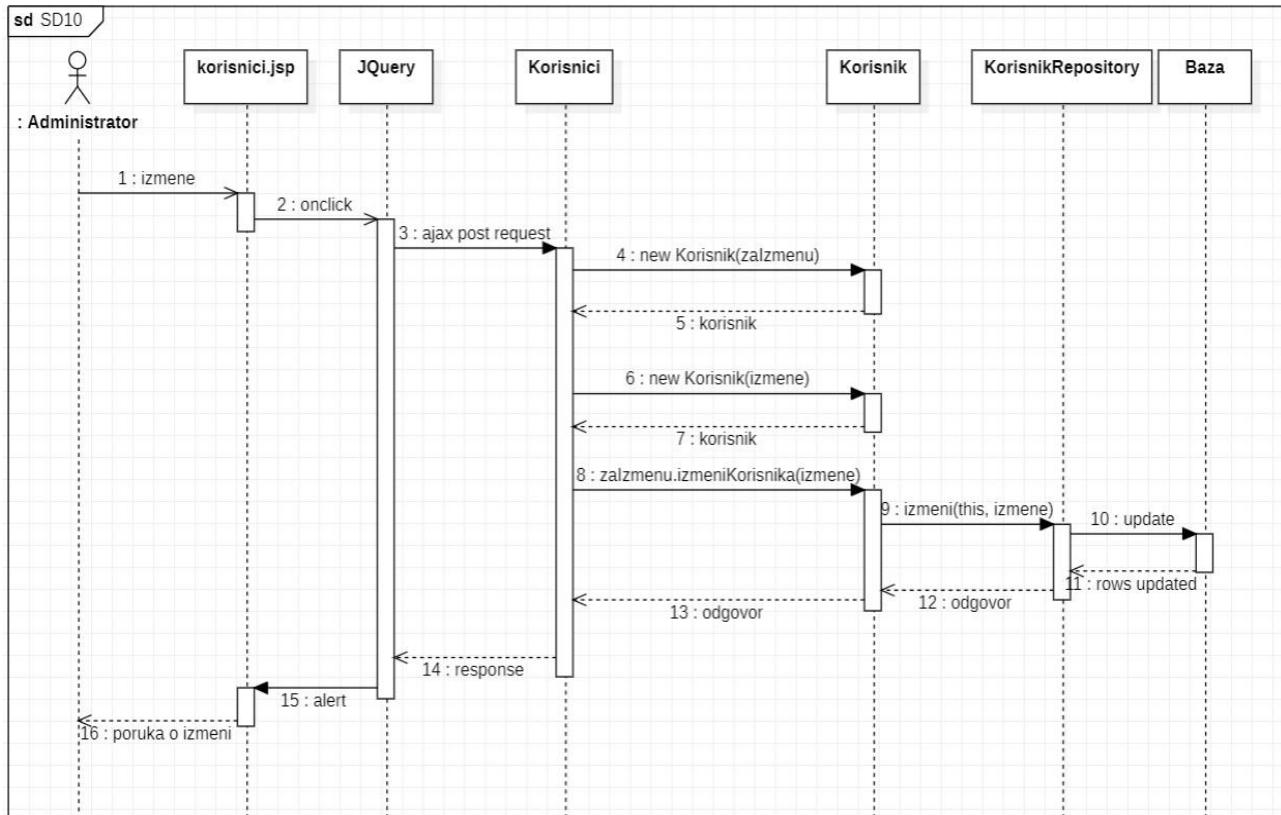
Controller kreira nov objekat Korisnik-a, koji u toku kreiranja heshi-ra zadati password pozivom static metode getHash(pass) klase MD5.

Zatim nad dobijenim objektom poziva metodu registruj(). Istu metodu koja se poziva pri registrovanju novog korisnika od strane tog korisnika.

Korisnik objekat predaje sebe kao parametar dodaj(korisnik) metode KorisnikRepository-a koji vrši insert nad bazom.

Nakon dodavanja u bazu izvršavanje se vraća do Controller-a koji refrešuje korisnici.jsp View i administratoru se prikazuje nov korisnik u listi korisnika.

5.10 Editovanje korisnika



(Slika 47) Dijagram sekvenci use case-a editovanja korisnika

Proces editovanja korisnika predstavlja planiranu implementaciju UG18 ili use case-a editovanja korisnika.

Administrator unosi željene izmene direktno u listi korisnika nad row-om koji želi da izmeni. Na onclick događaju dugmeta za izmenu poziva se jquery funkcija u scripts.js fajlu.

Jquery funkcija pakuje izmenjene podatke u ajax post request parametre i šalje ih Korisnici Controlleru.

Korisnici Controller kreira dve instance Korisnik-a, jednu zaIzmenu i jednu izmenjenu, zatim nad instancom zaIzmenu poziva metodu izmeniKorisnika i kao parametar predaje izmenjenu instancu. Korisnik Model zatim predaje sebe i izmenjenu kopiju KorisnikRepository-ju metodom izmeni(stari, novi).

Repository vrši updejt, izvršavanje se vraća do Controllera koji pri uspehu vraća response sa status-om 200, a pri grešci (SQLException) statusom 500.

Jquery skripta prikazuje alert o grešci ili uspehu zavisno od response status-a.

Korisniku ostaje izmenjena verzija u listi, a stranica se zbog ajax poziva ne refrešuje.

6. KORISNIČKI INTERFEJS

6.1 Autentikacija



The registration form is titled "REGISTRACIJA". It contains five input fields: "Korisničko ime", "Ime", "Prezime", "Password", and "Adresa". Below the fields are two buttons: a blue "Registruj se" button and a grey "Resetuj" button.

Copyright @LoremCatering 2022 David Gudovic 1118

(Slika 48) Forma za registraciju



The login form is titled "LOGIN". It has two input fields: "Korisnicko ime" and "Password". Below the fields is a large blue "Uloguj se!" button.

Copyright @LoremCatering 2022 David Gudovic 1118

(Slika 49) Forma za login

6.2 Pregled ponude

The screenshot shows a web application interface for a catering service named "Lorem Catering". The top navigation bar includes links for "POČETNA", "PONUDA", "Administracija", and a user profile with "Poeni: 15" and "dgudovicM". On the left, a sidebar lists categories: "Sve", "Slani program" (Pizze, Burita, Salate, Mesa), and "Slatki program" (Pite, Kolači, Torte). The main content area displays six food items in a grid:

- Cesar Salata**: Salata od pileceg belog mesa, zelene salate, paradaiza, preliva...
Cena: 200 RSD
- Burito Madre**: Najpopularniji lokalni burito sa piletinom, zapečenim pasuljem i...
Cena: 300 RSD
- Capricciosa**: Klasična italijanska pica sa tankim testom, pelatom, šunkom i sirom
Cena: 400 RSD
- Popek**: Pohovano mleveno juneće meso, sa kajmakom i pršutom, pomfrit i salata
Cena: 500 RSD
- Lenja pita**: Tradicionalna pita crnogorske kuhinje sa dve kore i filom od jabuka
Cena: 400 RSD
- Quattro Fromagi**: Poznata italijanska pica sa cetiri vrste sira. Mozarella, Gauda, Trapist,...
Cena: 500 RSD

(Slika 50) Prikaz cele ponude

This screenshot shows the same web application interface, but the sidebar categories are different, reflecting a filtered view of the "Slatki program": "Sve", "Slani program" (Pizze, Burita, Salate, Mesa), and "Slatki program" (Pite, Kolači, Torte). The main content area displays three dessert items in a grid:

- Lenja pita**: Tradicionalna pita crnogorske kuhinje sa dve kore i filom od jabuka
Cena: 400 RSD
- Čokoladna torta**: Torta sa prelivom od crne čokolade i filom od bele čokolade
Cena: 400 RSD
- Mafin s' jagodom**: Ukusni mafin sa ukrasnim omotom, filom od vanile i jagodama
Cena: 300 RSD

(Slika 51) Prikaz ponude filtrirane na slatki program

6.3 Korpa za kupovinu

The screenshot shows a user interface for a catering service. At the top, there is a navigation bar with the logo "Lorem Catering", "POČETNA", "PONUDA", "Administracija", "Poeni: 46", a shopping cart icon with "5", and the user "dgudovicA". On the left, a sidebar has icons for "Informacije", "Korpa", "Istorija", and "Izbrisati profil". The main content area displays a message: "Vaša korpa je prazna!" with a button "Pogledajte ponudu!". At the bottom, it says "Copyright @LoremCatering 2022" and "David Gudovic 1118".

(Slika 52) Prikaz prazne korpe

The screenshot shows a user interface for a catering service. At the top, there is a navigation bar with the logo "Lorem Catering", "POČETNA", "PONUDA", "Administracija", "Poeni: 46", a shopping cart icon with "5", and the user "dgudovicA". On the left, a sidebar has icons for "Informacije", "Korpa", "Istorija", and "Izbrisati profil". The main content area displays a message: "Vaša narudžbina" followed by a list of items: "Lenja pita" (2 units, price 800 RSD), "Capricciosa" (3 units, price 1200 RSD), and "Čokoladna torta" (3 units, price 1200 RSD). At the bottom, it shows "Sub Total: 5000 RSD" and "Total: 4500 RSD". At the very bottom, it says "Copyright @LoremCatering 2022" and "David Gudovic 1118".

(Slika 53) Prikaz korpe sa 5 stavki

6.4 Profil Korisnika

Copyright @LoremCatering 2022

David Gudovic 1118

(Slika 54) Forma za prikaz i izmenu korisničkih informacija

Copyright @LoremCatering 2022

David Gudovic 1118

(Slika 55) Forma za brisanje profila

Lorem Catering POČETNA PONUDA Administracija Poeni: 46 dgudovicA

Informacije

Korpa

Istorija

Izbriši profil

Narucena: 28/07/2022
Status: Priprema se
Stavke
Popek: 500 RSD x 4
Mafin s' jagodom: 300 RSD x 2
Popust: 20 % **Otkazi**
Total: 2080 RSD

Narucena: 27/07/2022
Status: Otkazana
Stavke
Burito Madre: 300 RSD x 4
Mafin s' jagodom: 300 RSD x 2
Popust: 0 % **Ponovi!**
Total: 3800 RSD

Narucena: 25/07/2022
Status: Otkazana
Stavke
Mafin s' jagodom: 300 RSD x 8
Cezar Salata: 200 RSD x 7
Popust: 20 % **Ponovi!**
Total: 3040 RSD

Narucena: 22/07/2022
Status: Ostvarena
Stavke
Burito Madre: 300 RSD x 4
Popek: 500 RSD x 4
Popust: 20 % **Ponovi!**

Narucena: 20/07/2022
Status: Otkazana
Stavke
Popek: 500 RSD x 4
Cezar Salata: 200 RSD x 3
Popust: 0 % **Ponovi!**

Narucena: 20/07/2022
Status: Otkazana
Stavke
Cezar Salata: 200 RSD x 3
Popek: 500 RSD x 4
Popust: 0 % **Ponovi!**

(Slika 56) Pregled istorije narudžbi korisnika

6.5 Menadžerski paneli

The screenshot shows a web application interface for managing orders. On the left sidebar, there are menu items: 'Menadžer' (highlighted in red), 'Narudžbine', 'Proizvodi', and 'Izveštaji'. The main content area has a header: 'Ovde možete upravljati neostvarenim narudžbinama!'. Below this, four order cards are displayed:

- Narudžba 1:** Narucena: 28/07/2022, Naručio: Oker. Stavke: Quattro Fromagi: 500 RSD x 7. Total: 3500 RSD. Buttons: Ostvari, Otkazi.
- Narudžba 2:** Narucena: 28/07/2022, Naručio: Mirko. Stavke: Mafin s' jagodom: 300 RSD x 3; Cezar Salata: 200 RSD x 4. Total: 2600 RSD. Buttons: Ostvari, Otkazi.
- Narudžba 3:** Narucena: 28/07/2022, Naručio: dgudovicA. Stavke: Popek: 500 RSD x 4. Total: 2080 RSD. Buttons: Ostvari, Otkazi.
- Narudžba 4:** Narucena: 20/07/2022, Naručio: dgudovicM. Stavke: Burito Madre: 300 RSD x 4. Total: 1200 RSD. Buttons: Ostvari, Otkazi.

(Slika 57) Stranica za ostvarivanje narudžbina

The screenshot shows a 'Mesečni izveštaj' (Monthly Report) page. On the left sidebar, there are menu items: 'Menadžer' (highlighted in red), 'Narudžbine', 'Proizvodi', and 'Izveštaji'. The main content area has a header: 'Mesečni izveštaj'. It contains two large boxes:

- Prodaja proizvoda:**
 - 25 x Cezar Salata : 5000
 - 31 x Burito Madre : 9300
 - 6 x Capricciosa : 2400
 - 34 x Popek : 17000
- Potrošnja korisnika:**
 - dgudovicA je potrošio 23280 RSD
 - dgudovicM je potrošio 7000 RSD
 - dgudovicKI je potrošio 3700 RSD

At the bottom, there are summary statistics: '9 Ostvarenih narudžbi: 33980 RSD' and '10 Otkazanih narudžbi: 36500 RSD'. A 'PDF' button is also present.

(Slika 58) Stranica za prikaz mesečnog izveštaja

Menadžer

Narudžbine

Proizvodi

Izveštaji

Proizvod	Slika	Opis	RSD	Kategorija	Akcije
Cezar Salata	/img/cezar.jpg	Salata od pilecog belog mesa, zelene salate, paraidaiza, preliva...	200	Salate	X Izmeni
Burito Madre	/img/burito.jpg	Najpopularniji lokalni burito sa piletinom, zapečenim pasuljem i zelenom salatom	300	Burita	X Izmeni
Capricciosa	/img/pica.jpg	Klasična italijanska pica sa tankim testom, pelatom, šunkom i sirom	400	Pizze	X Izmeni
Popek	/img/popek.jpg	Pohovano mleveno juneće meso, sa kajmakom i pršutom, pomfrit i salata	500	Mesa	X Izmeni
Lenja pita	/img/lenjaPita.jpg	Tradicionalna pita crnogorske kuhinje sa dve kore i filom od jabuka	400	Pite	X Izmeni
Quattro Fromagi	/img/quattroS.jpg	Poznata italijanska pica sa cetiri vrste sira. Mozarella, Gauda, Trapist, Parmegiano.	500	Pizze	X Izmeni
Čokoladna	/img/cezar.jpg	Torta sa prelivom od crne čokolade i filom od	400		X Izmeni

Kategorije

Slane		Slatke	
Naziv kategorije	Dodaj	Naziv kategorije	Dodaj
Pizze	slani	Pite	slatki
Burita	slani	Kolači	slatki
Salate	slani	Torte	slatki
Mesa	slani		

Izaberite sliku

 No file chosen

Naziv

Opis

Kategorija

RSD

Dodaj u ponudu

Copyright @LoremCatering 2022

David Gudovic 1118

(Slika 59) Stranica za dodavanje, brisanje i edit proizvoda i kategorija

6.6 Administratorski panel

The screenshot shows the 'Admin panel' section of a web application. On the left sidebar, there are links for 'Korisnici', 'Narudžbine', 'Proizvodi', and 'Izveštaji'. The main area displays a table of users with columns: Username, Ime, Prezime, Adresa, Poeni, Rola, Password, and Akcije. A form at the top allows for adding a new user. The table data is as follows:

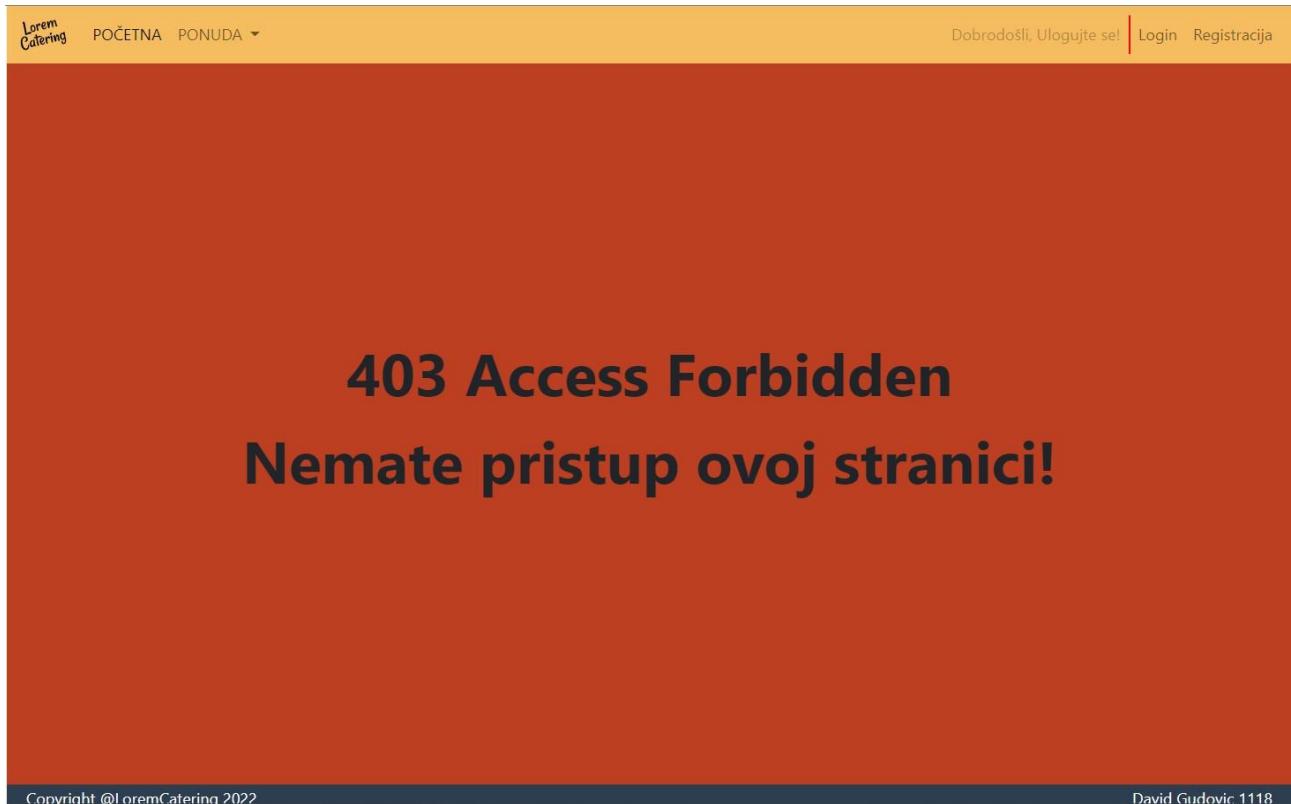
Username	Ime	Prezime	Adresa	Poeni	Rola	password	Akcije
Korisnicko ime	Ime	Prezime	Adresa	0	Klijent	<input type="button" value="password"/>	<input type="button" value="Dodaj"/>
dgudovicA	David	Gudov	IX Crnogorske Brigade 106	44	Administrator	Neizmenjen	<input type="button" value="X"/> <input type="button" value="Izmeni"/>
dgudovicK	David	Gudd	Adresaa	0	Klijent	Neizmenjen	<input type="button" value="X"/> <input type="button" value="Izmeni"/>
dgudovicKI	David	Gudovic	awdwa	3	Klijent	Neizmenjen	<input type="button" value="X"/> <input type="button" value="Izmeni"/>
dgudovicM	David	Gudovic	Obrenovacka 23	15	Menadzer	Neizmenjen	<input type="button" value="X"/> <input type="button" value="Izmeni"/>
Mirko	Milan	Markovic	Pozega bb	0	Klijent	Neizmenjen	<input type="button" value="X"/> <input type="button" value="Izmeni"/>
Oker	Boja	Bojanje	Adresa neka	0	Klijent	Neizmenjen	<input type="button" value="X"/> <input type="button" value="Izmeni"/>

Copyright @LoremCatering 2022

(Slika 60) Stranica za dodavanje, brisanje i edit korisnika

6.7 Poruke

Korisniku se u toku interakcije sa prethodno navedenim stranicama povremeno prikazuju poruke o uspešnosti zatraženih radnji ili greškama, primeri nekoliko takvih poruka dati su u nastavku.



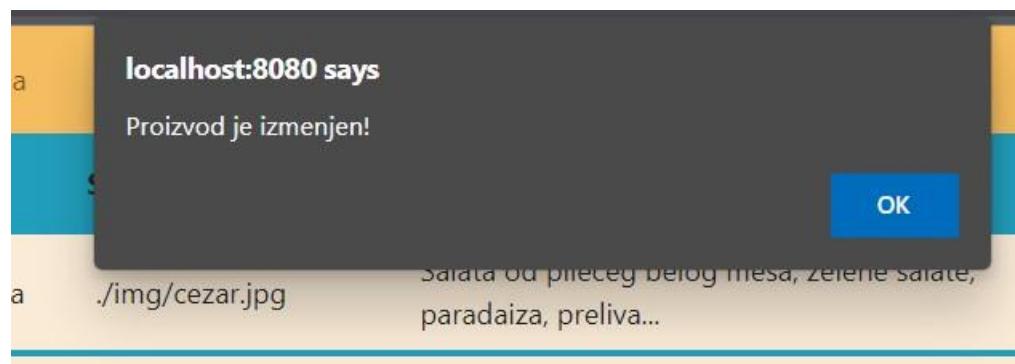
(Slika 61) Poruka o neautorizovanom pristupu nekoj stranici

A screenshot of a registration form titled "REGISTRACIJA". The form consists of several input fields: "Korisničko ime" (username), "Ime" (name), "Prezime" (surname), "Password" (password), and "Adresa" (address). Below the address field are two buttons: "Registruj se" (Register) and "Resetuj" (Reset). A red error message "Korisničko ime već postoji!" (The username already exists!) is displayed below the buttons.

(Slika 62) Poruka o grešci pri registraciji



(Slika 63) Poruka o uspešnom registrovanju



(Slika 64) Alert o izmeni proizvoda

7. ZAKLJUČAK

Implementacijom aplikacije obezbeđene su sve funkcionalnosti obuhvaćene verbalnim opisom zahteva sistema opisanom u uvodu ovog rada. Time su pokriveni svi osnovni use case-ovi ketering službe za čije bi poslovanje bila korišćena jedna ovakva aplikacija.

Primenom MVC i Repository pattern-a, pokušato je poboljšavanje kvaliteta aplikacije u vidu lakše čitljivosti i održavanja koda, kao i, u budućnosti, unapređivanja funkcionalnosti aplikacije. Mogućnosti za dodatne funkcionalnosti su brojne, aplikacija se može proširiti novim funkcionalnostima kao i davanjem većeg nivoa kontrole administratorima i menadžerima i boljem uvidu u statistiku i poslovanje aplikacije.

To se može postići proširivanjem parametara u mesečnim izveštajima, dodavanjem mogućnosti pregleda izveštaja iz prethodnih meseci kao i poređenjem istih.

Takođe administratorima se može implementirati mogućnost pregleda istorije zasebnih korisnika, a menadžerima mogućnost brisanja narudžbina.

Aplikacija se takođe može proširiti dodavanjem funkcionalnosti za reklamaciju narudžbina, gdje korisnici ostavljaju poruke i zahteve menadžerima koji u panelu za reklamacije mogu odgovoriti korisnicima porukom, kao i preuzeti mere poput povraćaja novca ili dodavanja poena za popust za buduće narudžbe.

Deo interfejsa predviđen za korišćenje od strane klijenata dizajniran je da bude u određenoj meri responsive, admin paneli su predviđeni za korišćenje sa računara pa njihov dizajn nije responzivan.

8. REFERENCE

[1] Informacije o predmetu Internet Programerski Alati na its.edu.rs/onlineservise