**VISOKA ŠKOLA STRUKOVNIH STUDIJA ZA INFORMACIONE TEHNOLOGIJE**

****

**Internet programerski alati**

Projekat

**Projektovanje i razvoj Java Web aplikacije za ketering službu**

Predmetni nastavnik: Student:

Prof. Dr. Svetlana Jevremović David Gudović 11/18

Datum predaje: x.06.2022

**Beograd**

**Jun, 2022**

Sadržaj

[1. UVOD 3](#_Toc108465576)

[2. LARMANOVA METODA 3](#_Toc108465577)

[3. USE CASE 3](#_Toc108465578)

[3.1 Akter Klijent 3](#_Toc108465579)

[3.2 Akter Menadžer 6](#_Toc108465580)

[3.3 Akter Administrator 8](#_Toc108465581)

[4. UGOVORI 10](#_Toc108465582)

[5. KONCEPTUALNI MODEL 24](#_Toc108465583)

[5.1 Dijagram klasa 24](#_Toc108465584)

[5.2 Relacioni model 24](#_Toc108465585)

[5.3 Dijagrami baze podataka 25](#_Toc108465586)

[6. DIJAGRAMI SEKVENCI 27](#_Toc108465587)

[6.1 Registracija 27](#_Toc108465588)

[6.2 Login 28](#_Toc108465589)

[7. MVC PATTERN 28](#_Toc108465590)

[8. REPOSITORY PATTERN 28](#_Toc108465591)

[9. ZAKLJUČAK 28](#_Toc108465592)

[10. REFERENCE 29](#_Toc108465593)

# 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 klasa 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. Kratko je objašnjen MVC pattern, njegov značaj i implementacija u projektu, kao i Repository pattern korišćen u implementaciji.

Projekat je rađen po sledećim zahtevima[1]:

“

Korisnici aplikacije su klijenti, administratori i menadžeri. Web aplikacija ima za cilj da omogući online posetiocima da se upoznaju sa ponudom ketering službe, odnosno da pregledaju menije slatke i slane hrane, tačne cene proizvoda, način i uslove dostave hrane, mogućnost ostvarenja određenih popusta i da poruče izabrane proizvode.

Korisnici se mogu registrovati i time ostvariti pravo na različite pogodnosti. Svakom rezervacijom slatke ili slane hrane u određenim količinama, registrovani korisnici dobijaju poene, koje mogu iskoristiti za svaku sledeću porudžbinu hrane sa određenim popustom.

Administratori bi imali potpunu kontrolu nad aplikacijom, odnosno vršili bi administriranje korisnika, kao i same ketering službe. Menadžeri bi imali mogućnost da pregledaju različite vrste izveštaja o poslovanju ketering službe (samo one u kojoj imaju ulogu menadžera), kao i da ažuriraju podatke u vezi menija (prikaz, dodavanje, izmena i brisanje pojedinih proizvoda iz menija slatke i slane hrane).

”

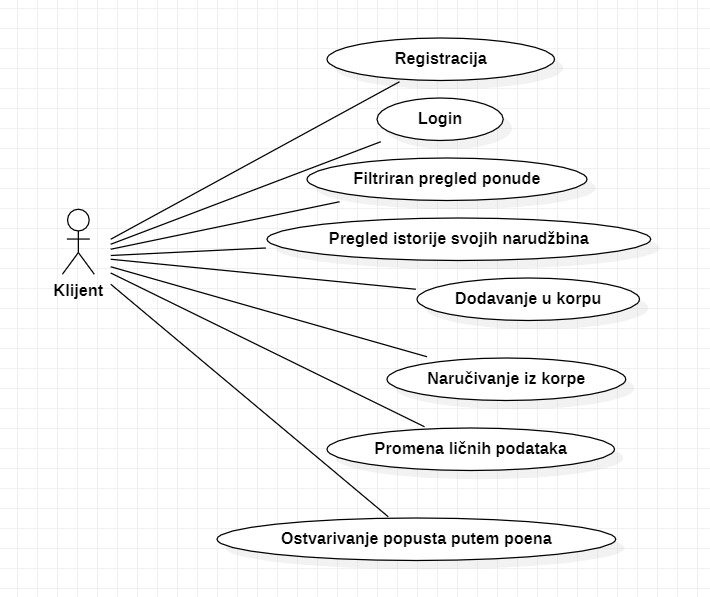
**Ključne reči:** Larmanova metoda, Java Web, Servlet, Java Bean, Java Server Pages, MVC, Repository pattern.

# LARMANOVA METODA

//tekst

# USE CASE

## 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 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: Filtriran pregled ponude

Aktor: Klijent

Preduslov: Nema

Osnovni scenario:

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

Alternativni scenario:

* 1. Nema rezultata pa sistem prikazuje poruku o greški (IA)

**SK 4**

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 poruku o grešci (IA)

**SK 5**

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 narudzbinu (SO)
4. Proizvod, količina i ukupna cena se dodaju u prikaz korpe (IA)

Alternativni scenario:

* 1. Proizvod je prvi artikal u narudžbini, pa sistem kreira narudžbinu u sesiji (SO)

**SK 6**

Naziv: Naručivanje iz korpe

Akter: Klijent

Preduslov: Klijent je dodao artikle u korpu

Osnovni scenario:

1. Klijent bira način plaćanja (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 (IA)

Alternativni scenario:

* 1. Klijent otkazuje narudžbu (APSO)
  2. Sistem miče narudžbu iz sesije (SO)
  3. Klijent se vraća na početnu stranu (IA)

**SK 7**

Naziv: Promena ličnih podataka

Akter: Klijent

Preduslov: Klijent je ulogovan

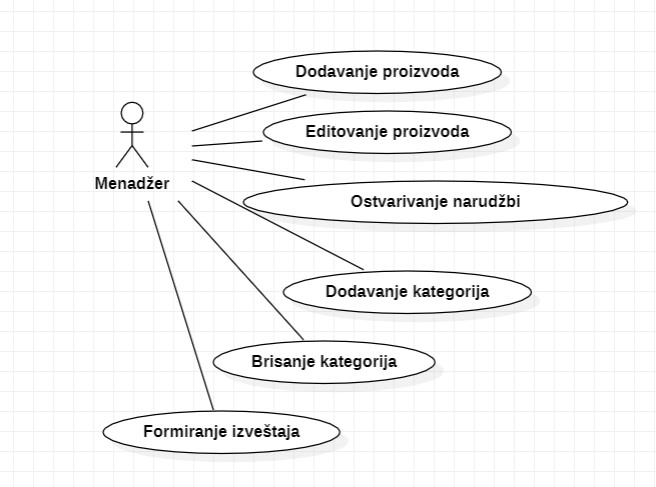
Osnovni scenario:

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)

## Akter Menadžer



(Slika 2) Use Case Menadžera

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

Korisnik je ulogovan sa rolom menadžer ili administrator.

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

**SK8**

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)

**SK9**

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)

**SK10**

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

**SK11**

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

**SK12**

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

**SK13**

Naziv: Kreiranje izveštaja

Akter: Menadžer

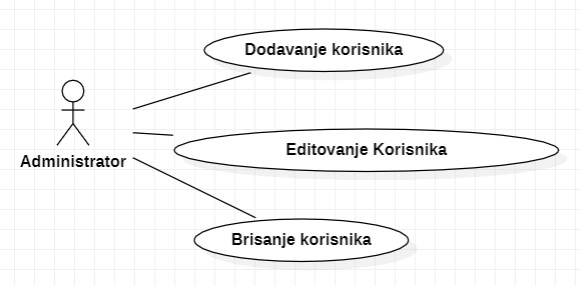
Preduslov: Podrazumevani

Osnovni scenario:

1. Korisnik unosi sve podatke izveštaja (APUSO)
2. Korisnik zatražuje čuvanje izveštaja (APSO)
3. Sistem generiše i čuva izveštaj (SO)
4. Sistem prikazuje izveštaj spreman za štampanje (IO)

Alternativni scenario: Nema

## Akter Administrator



(Slika 3) Use Case Administratora

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

Korisnik logovan sa rolom administrator.

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

**SK14**

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:

* 1. Korisnik se ne može uneti pa se korisniku prikazuje poruka o neuspehu (IA)

**SK15**

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 updejt podataka u bazi (SO)
4. Sistem prikazuje poruku o uspehu (IA)

Alternativni scenario:

* 1. Promena se ne može izvršiti pa sistem prikazuje poruku o grešci.

**SK16**

Naziv: Brisanje korisnika

Akter: Administrator

Preduslov: Korisnik je na stranici pregledkorisnika.jsp

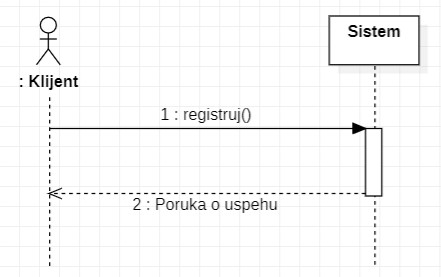
Osnovni scenario:

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

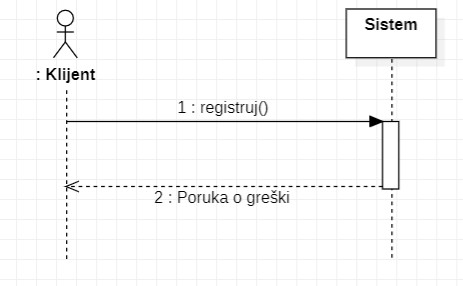
Alternativni scenario:

* 1. Sistem ne može ukolniti korisnika, pa prikazuje poruku o grešci (IA)

# UGOVORI



(Slika 4) Dijagram sekvenci SK1



(Slika 5) Dijagram sekvenci SK1 alternativni scenario

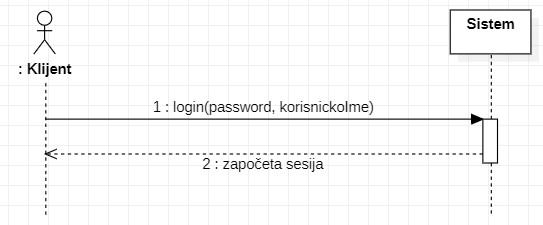
**Ugovor UG1:**

Operacija: registruj() : void

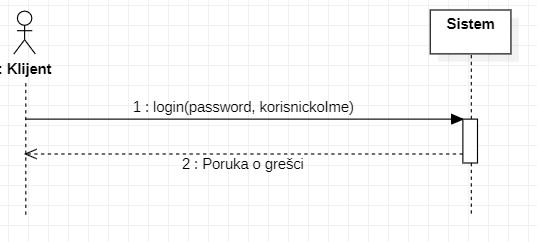
Veza sa SK: SK1

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:**

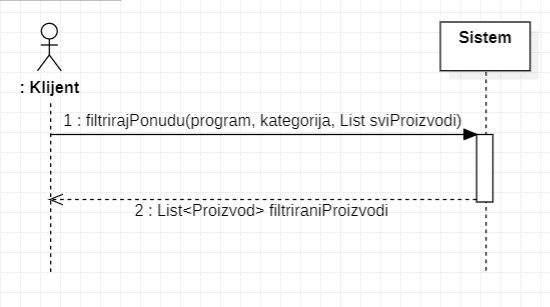
Operacija: login(Korisnik(password,korisnickoIme)): boolean

Veza sa SK: SK2

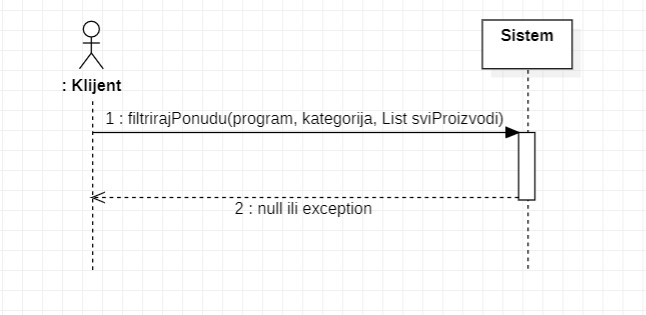
Preduslov: Controller je kreirao objekat sa zadatim parametrima

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

**\*Napomena:** Parametar login-a je objekat klase Korisnik kreiran konstruktorom sa samo passwordom i korisničkim imenom.



(Slika 8) Dijagram sekvenci SK3



(Slika 9) Dijagram sekvenci SK3 alternativni scenario

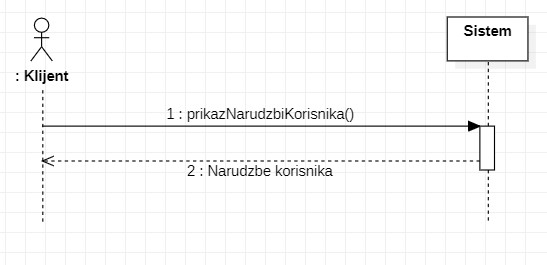
**Ugovor UG3:**

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

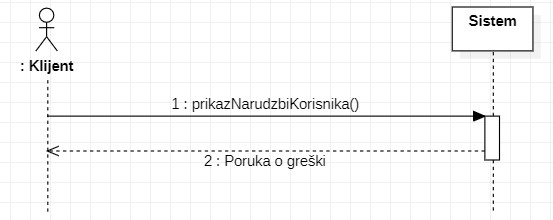
Veza sa SK: SK3

Preduslov: Korisnik je uneo validan program i/ili kategorije

Postuslov: prikazana je lista proizvoda



(Slika 10) Dijagram sekvenci SK4



(Slika 11) Dijagram sekvenci SK4 alternativni scenario

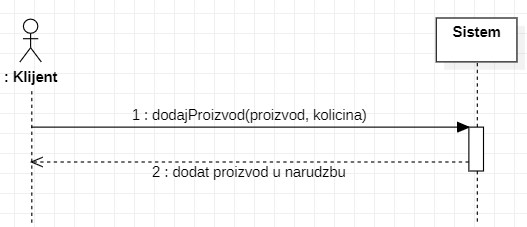
**Ugovor UG4:**

Operacija: prikazNarudzbiKorisnika() : void

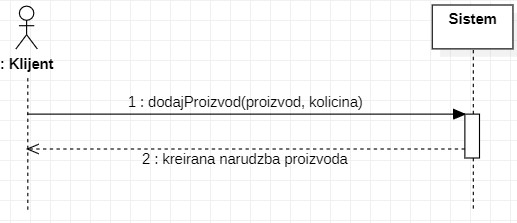
Veza sa SK: SK4

Preduslov: postoji sesija sa korisnikom

Postuslov: prikazane su sve narudzbe korisnika iz trenutne sesije



(Slika 12) Dijagram sekvenci SK5



(Slika 13) Dijagram sekvenci SK5 alternativni scenario

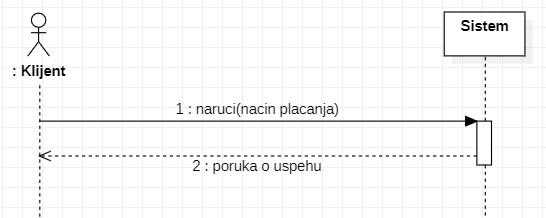
**Ugovor UG5:**

Operacija: dodajProizvod(proizvod, kolicina) : void

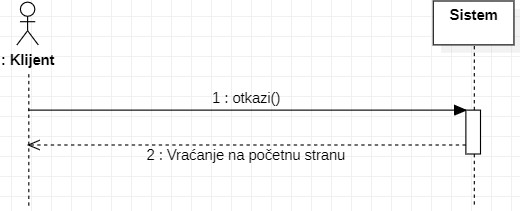
Veza sa SK: SK5

Preduslov: korisnik je uneo količinu u interfejsu

Postuslov: dodat je proizvod narudžbini



(Slika 14) Dijagram sekvenci SK6



(Slika 15) Dijagram sekvenci SK6 alternativni scenario

**Ugovor UG6:**

Operacija: naruci(nacinPlacanja) : void

Veza sa SK: SK6

Preduslov: korisnik je kreirao narudžbinu

Postuslov: narudžbina je dodata u bazu podataka

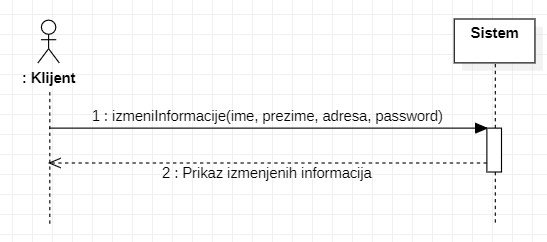
**Ugovor UG7:**

Operacija: otkazi() : void

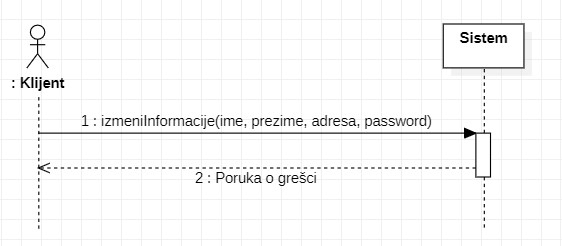
Veza sa SK: Sk7

Preduslov: korisnik je kreirao narudžbinu

Postuslov: Objekat narudžbine miče se iz sesije



(Slika 16) Dijagram sekvenci SK7



(Slika 17) Dijagram sekvenci SK7 alternativni scenario

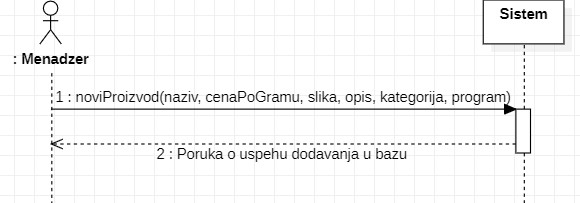
**Ugovor UG8:**

Operacija: izmeniInformacije(korisnickoIme, ime, prezime, adresa, password) : void

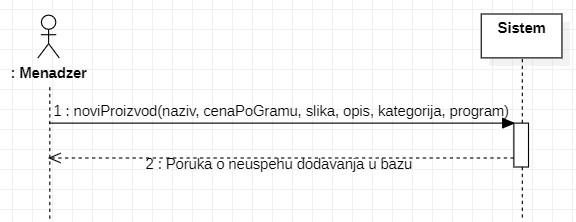
Veza sa SK: SK7

Preduslov: Korisnik je uneo validne nove podatke

Postuslov: Korisnikovi podaci su izmenjeni u bazi podataka



(Slika 18) Dijagram sekvenci SK8



(Slika 19) Dijagram sekvenci SK8 alternativni scenario

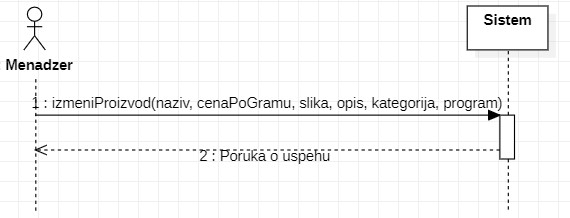
**Ugovor UG9:**

Operacija: noviProizvod(naziv, cenaPoGramu, slika, opis, kategorija, program) : void

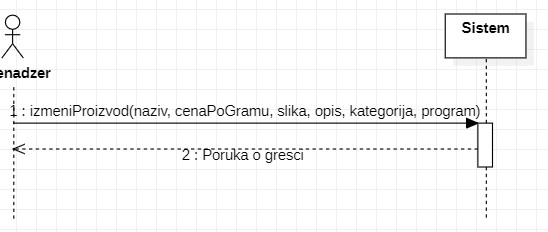
Veza sa SK: SK8

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

Postuslov: Dodat je nov proizvod u bazu podataka



(Slika 20) Dijagram sekvenci SK9



(Slika 21) Dijagram sekvenci SK9 alternativni scenario

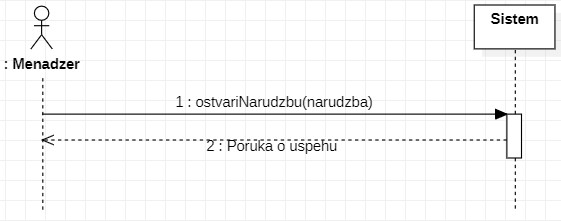
**Ugovor UG10:**

Operacija: izmeniProizvod(naziv, cenaPoGramu, slika, opis, kategorija, program) : void

Veza sa SK: SK9

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

Postuslov: Izmenjen je proizvod u bazi podataka.



(Slika 22) Dijagram sekvenci SK10

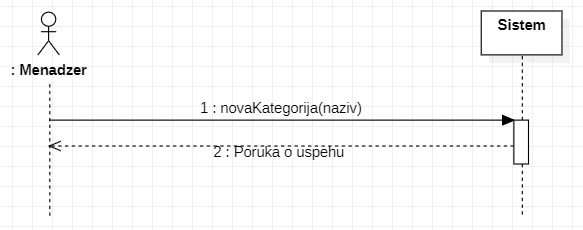
**Ugovor UG11:**

Operacija: ostvariNarudzbu(narudzba) : void

Veza sa SK: SK10

Preduslov: narudžba postoji

Postuslov: menja se vrednost ostvarena zadate narudžbine u bazi podataka



(Slika 23) Dijagram sekvenci SK11

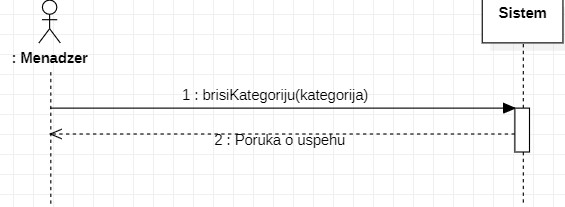
**Ugovor UG12:**

Operacija: novaKategorija(naziv, opis) : void

Veza sa SK: SK11

Preduslov:Ne postoji kategorija sa istim nazivom

Postuslov: Dodata je nova kategorija



(Slika 24) Dijagram sekvenci SK12

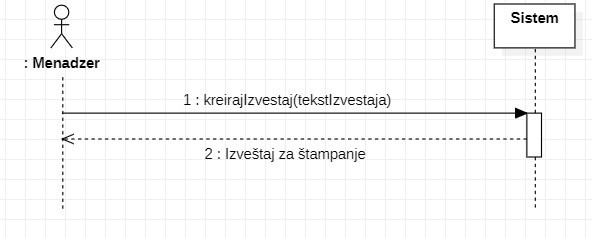
**Ugovor UG13:**

Operacija: brisiKategoriju(kategorija) : void

Veza sa SK: SK12

Preduslov: kategorija postoji

Postuslov: Zadata kategorija izbrisana je iz baze podataka



(Slika 25) Dijagram sekvenci SK13

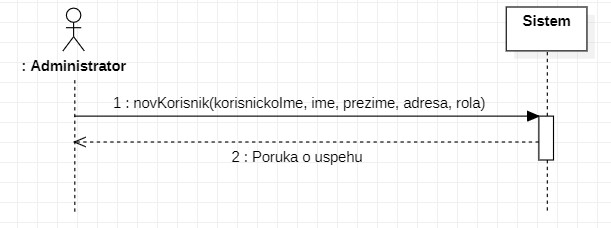
**Ugovor UG14:**

Operacija: kreirajIzvestaj(tekstIzveštaja) : void

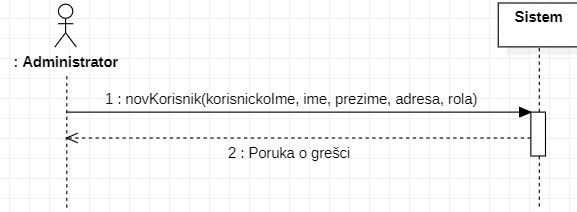
Veza sa SK: SK13

Preduslov: nema

Postuslov: Prikazana je strana za štampanje.



(Slika 26) Dijagram sekvenci SK14



(Slika 27) Dijagram sekvenci SK14 alternativni scenario

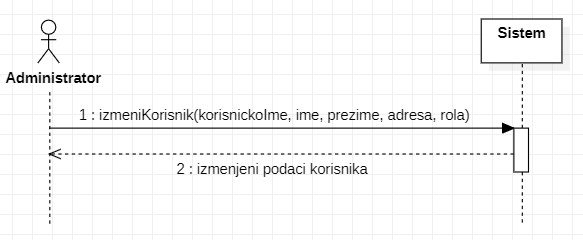
**Ugovor UG15:**

Operacija: novKorisnik(korisnickoIme, ime, prezime, adresa, rola, password) : void

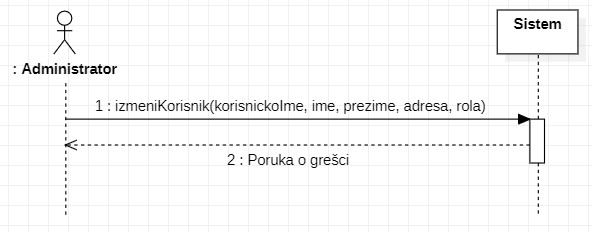
Veza sa SK: SK14

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

Postuslov: dodat je nov korisnik u bazi



(Slika 28) Dijagram sekvenci SK15



(Slika 29) Dijagram sekvenci SK15 alternativni scenario

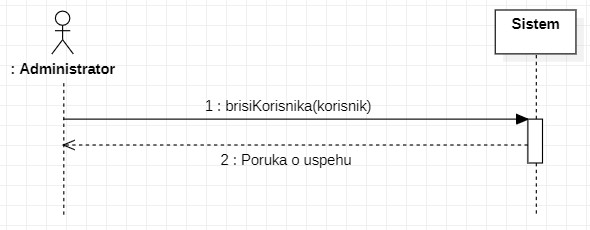
**Ugovor UG16:**

Operacija: izmeniKorisnika(korisnickoIme, ime, prezime, adresa, rola) : void

Veza sa SK: SK15

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 30) Dijagram sekvenci SK16

**Ugovor UG17:**

Operacija: brisiKorisnika(korisnik) : void

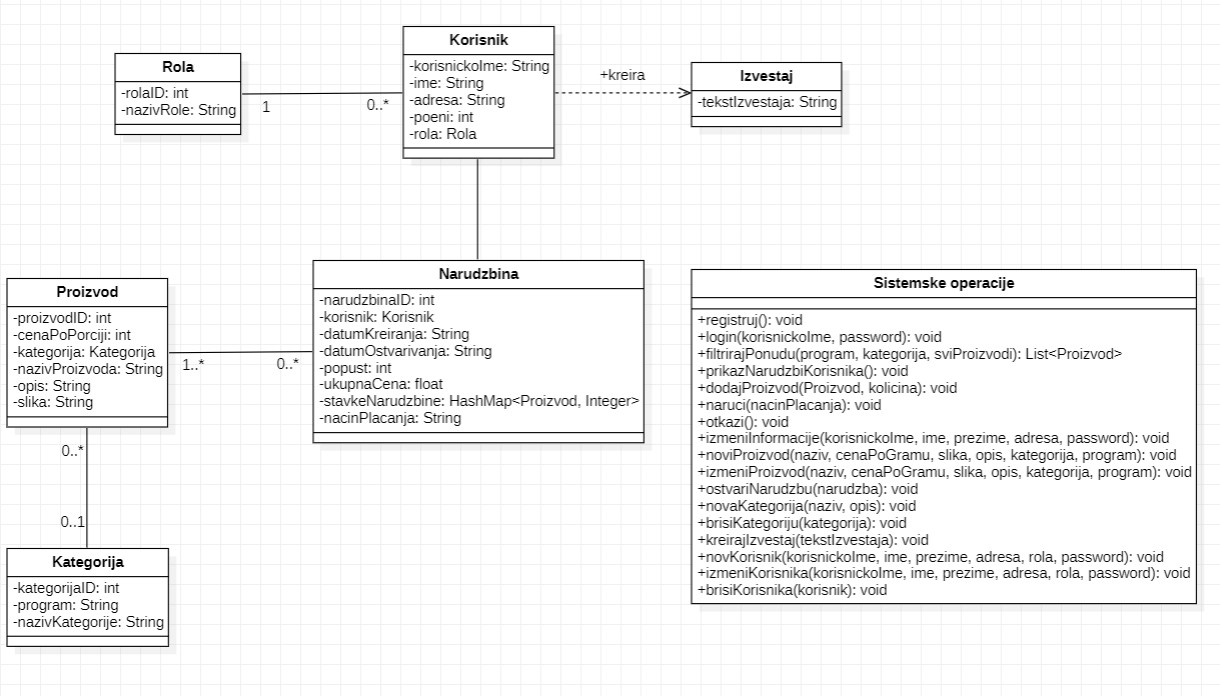
Veza sa SK: SK16

Preduslov: korisnik postoji

Postuslov: zadati korisnik je izbrisan iz baze podataka.

# KONCEPTUALNI MODEL

## Dijagram klasa



(Slika 31) Dijagram klasa

## 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** (NarudzbinaID, KorisnickoIme, Popust, DatumKreiranja, DatumOstvarivanja, UkupnaCena, ostvarena)

FK KorisnikID referencira Korisnici(PK KorisnikID)

**StavkeNarudzbine** (ProizvodID, NarudzbinaID, KorisnickoIme, Kolicina)

FK Proizvod referencira Proizvodi(PK ProizvodID)

FK NarudzbinaID,KorisnickoIme referencira Narudzbine(PK NarudzbinaID, KorisnickoIme)

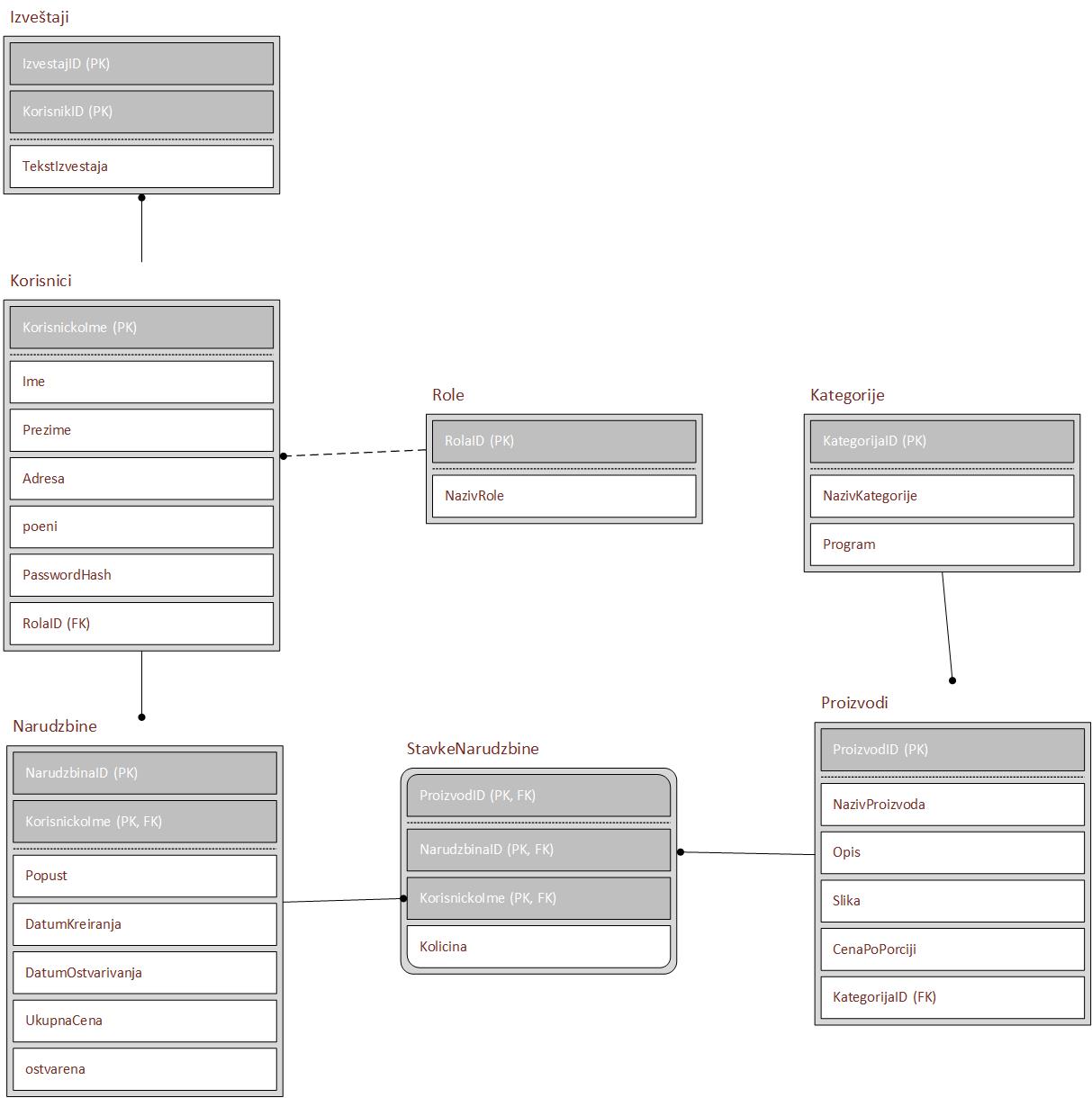
**Kategorije** (KategorijaID, NazivKategorije, Program)

**Role** (RolaID, NazivRole)

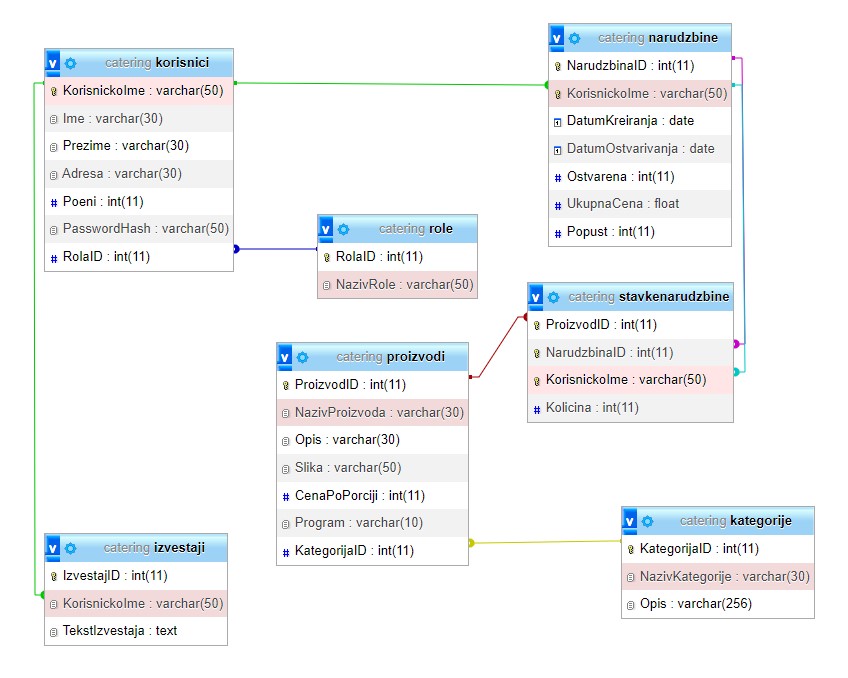
**Izvestaji** (IzvestajID, KorisnikID, TekstIzvestaja)

FK KorisnikID referencira Korisnici(PK KorisnikID)

## Dijagrami baze podataka



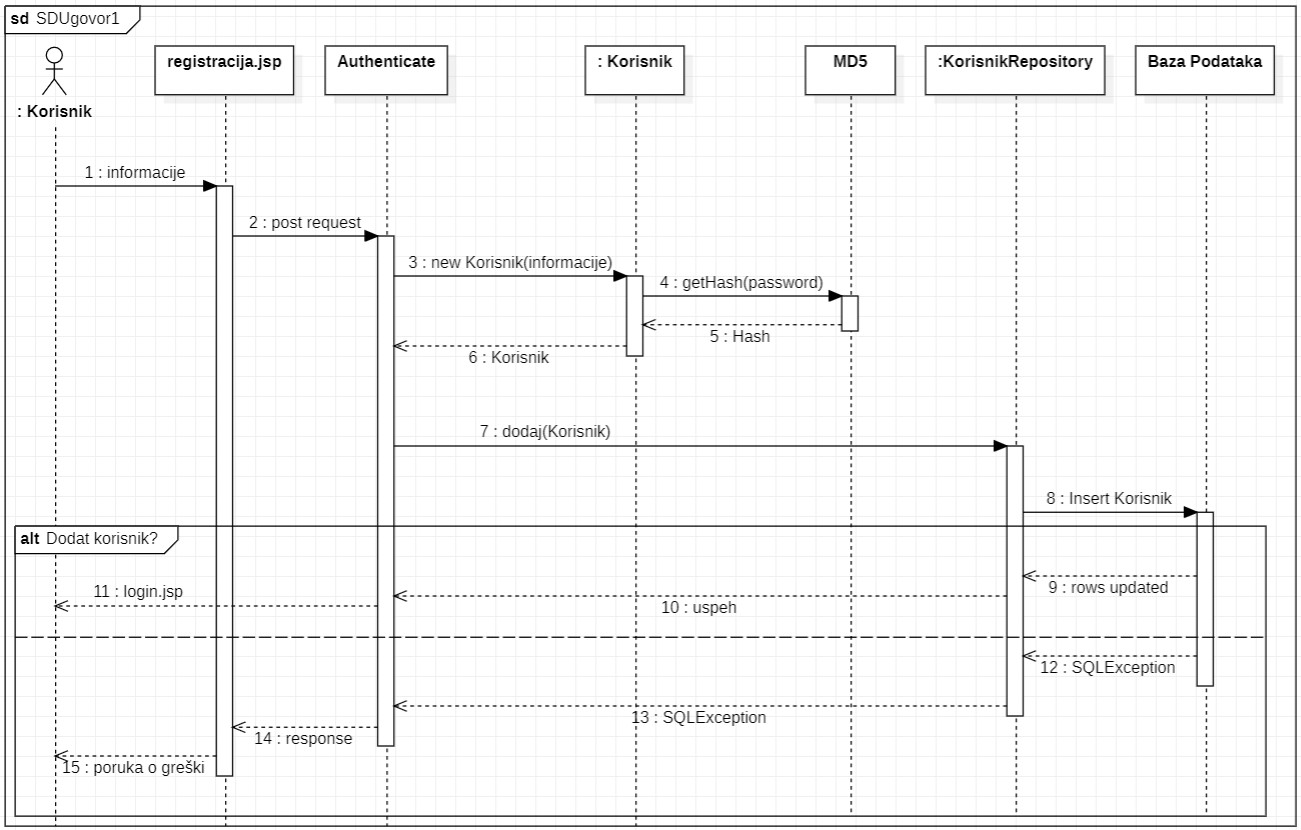
(Slika 32) IDEF1X model



(Slika 33) Tabele i relacije u bazi

# DIJAGRAMI SEKVENCI

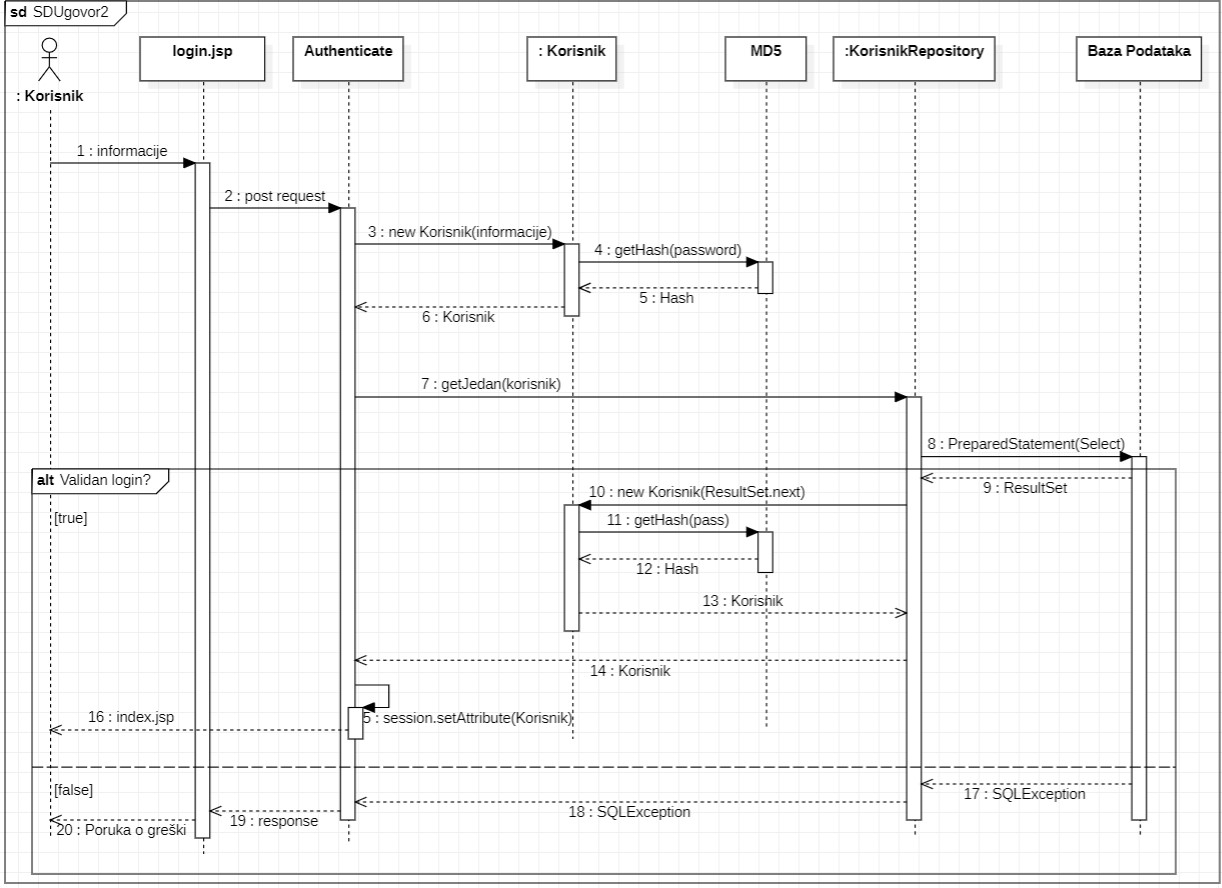
## 6.1 Registracija

****

(Slika 34) Dijagram sekvenci ugovora UG1

// Tekst

## 6.2 Login

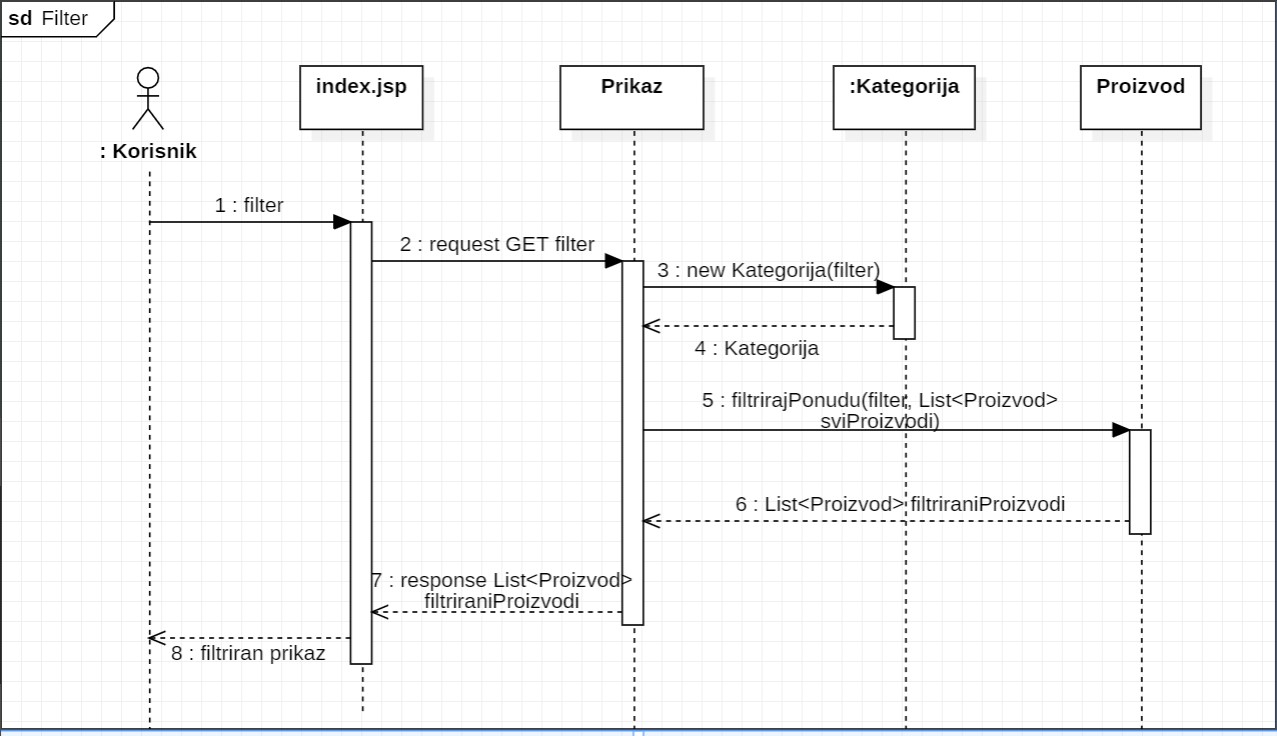


(Slika 35) Dijagram sekvenci ugovora UG2

// Tekst

## 6.3 Filter

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



(Slika 36) Dijagram sekvenci use case-a filtriranja

# MVC PATTERN

# REPOSITORY PATTERN

# ZAKLJUČAK

# REFERENCE