Univerzitet u Kragujevcu

Fakultet inženjerskih nauka



Seminarski rad iz predmeta

Softverski inženjering

Tema:

**Aplikaciju koja simulira rad sa bazom podataka preporuku literature za čitanje**

Student: Predmentni nastavnik:

Danica Stefanović 648/2018 prof. dr Nenad Filipović

Kragujevac 2021.

Sadržaj:

[1 Postavka zadatka i detaljan opis aplikacije 2](#_Toc80098330)

[1.1 Definisanje zadatka 2](#_Toc80098331)

[1.2 Opis korišćenja aplikacije 2](#_Toc80098332)

[2 Opis delova programa sa izvornim kodom 2](#_Toc80098333)

[2.1 Uvod 2](#_Toc80098334)

[2.2 Datoteka “KnjigaForma.java” 2](#_Toc80098335)

[2.3 Korišćenje aplikacije 8](#_Toc80098340)

[3 UML dijagrami 11](#_Toc80098341)

[3.1 Dijagram slučajeva korišćenja 11](#_Toc80098342)

[3.2 Dijagram klasa 12](#_Toc80098343)

[3.3 Dijagram sekvenci 13](#_Toc80098344)

[3.4 Dijagram aktivnosti 15](#_Toc80098345)

[3.5 Dijagram stanja 16](#_Toc80098346)

[4 Zaključak 17](#_Toc80098347)

[5 Literatura 18](#_Toc80098347)

# Postavka zadatka i detaljan opis aplikacije

## Definisanje zadatka

Osnovni zadatak je pravljenje korisničke aplikacije koja simulira čuvanje podataka o knjigama koje služe kao preporuka literature za čitanje.

## Opis korišćenja aplikacije

Korisnik kad pokrene aplikaciju,automatski se povezuje I na bazu podataka koja se nalazi u sql-u.Korisnik ima sledeće mogućnosti: selektuje određenu knjigu I pogleda infomacije,doda novu knjigu,da pretražuje,ažurira podatke ali I briše.

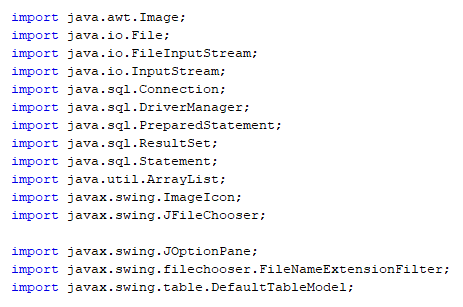
# Opis delova programa sa izvornim kodom

## Uvod

Za izradu ovog projektnog zadatka korišćeno je razvojno okruženje NetBeans IDE, programski jezik Java.

## Datoteka “KnjigaForma.java”

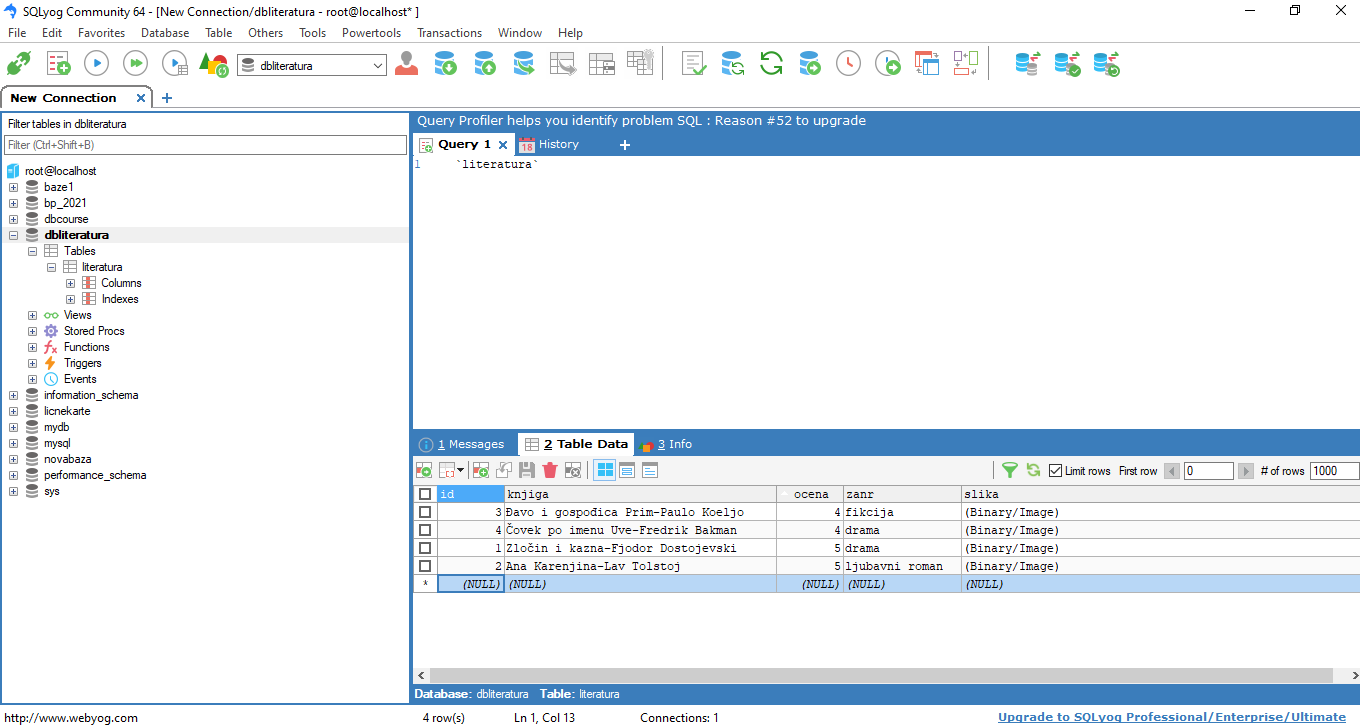
U ovoj aplikaciji upotrebljene su sledeće biblioteke.



Slika 1-biblioteke

Povezivanje sa bazom podataka dbliteratura možemo videti na slici 2:

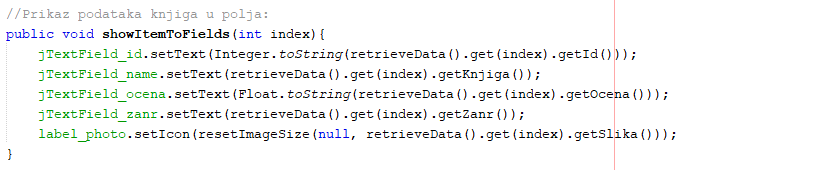
Slika 2-povezivanje baze podatka sa aplikacijom

Na slici 3 imamo prikaz kako baza izgleda u programu SQLyog Community.

Slika 3-izgled naše baze u SQL-u

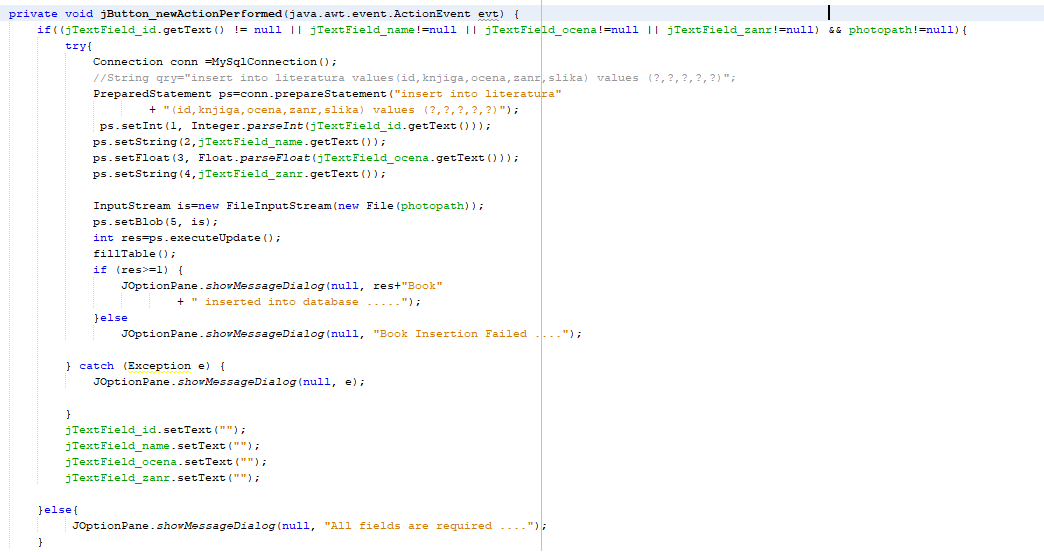
Slika 4 nam prikazuje smeštanje knjiga u arraylist I popunjavanje tabele.

Slika 4-unos podataka knjiga u tabelu



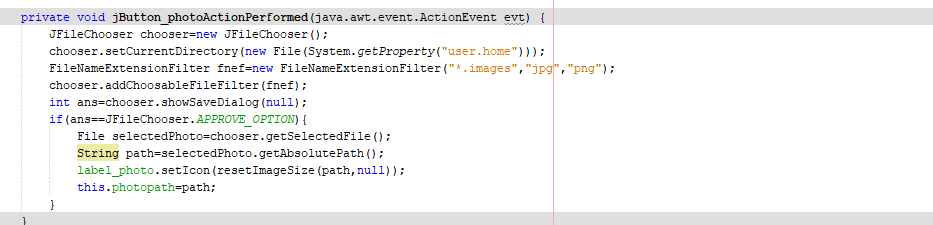
Slika 5-popunjavanje podataka u polja forme

Korisnik ima opciju dodavanja literature u bazu klikom na dugme I to možemo videti na slici 6:



Slika 6-dugme za dodavanje

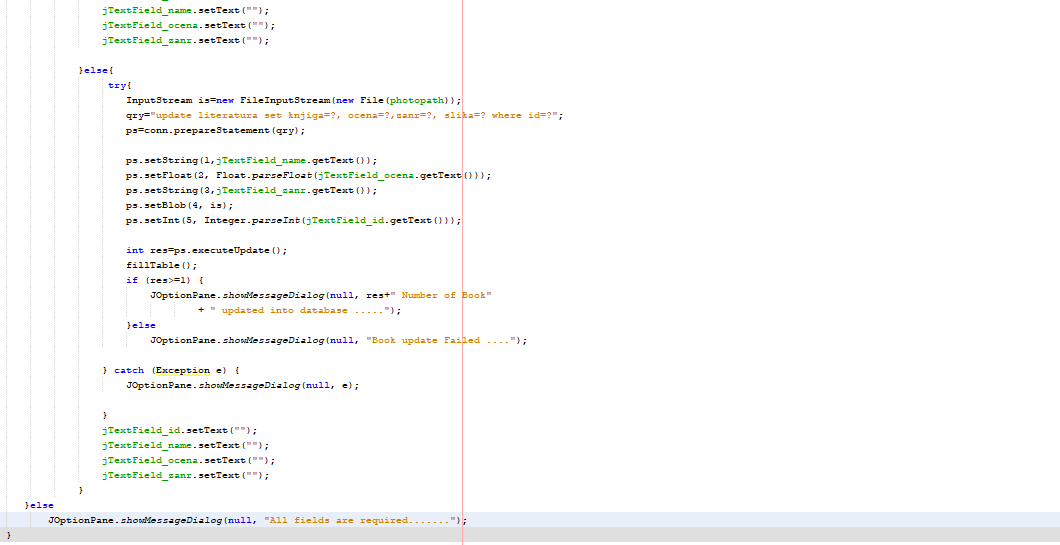
Slika 7 predstavlja unošenje slike u aplikaciju korišćenjem jButton\_photo koji nam omogućuje da izaberemo sliku koju imamo u našem računaru.



S1ika 7-dugme za unos slike

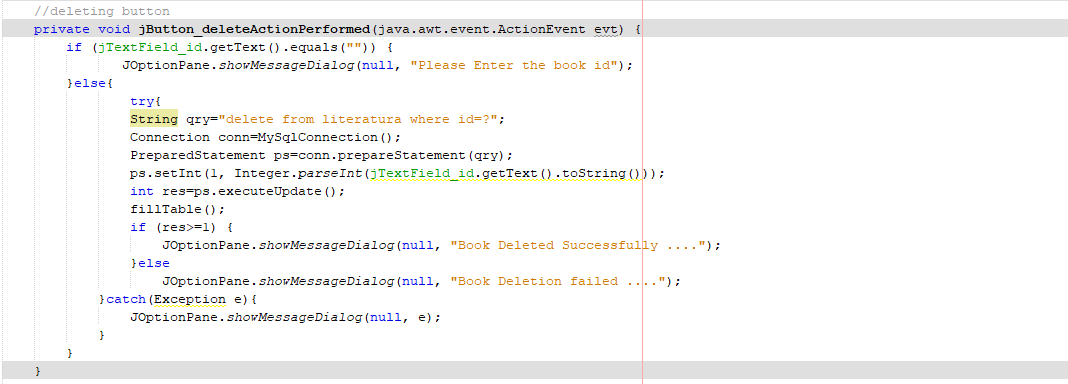
Slika 8 predstavlja update dugme uz pomoću kojeg korisnik ima opciju da izmeni knjigu koju on želi.

Slika 8-update dugme deo1.



Slika 8-update dugme deo2..

Korisnik ima mogućnost brisanja knjige iz baze podataka.Klasu koja to čini možemo videti na slici 9.



Slika 9-delete dugme

Korisnik ima mogućnost pretraživanja baze po nazivu knjige.Klasu za pretraživanje vidimo na slici 10.



Slika 10- search

Korišćenje aplikacije:

Nakon poktretanja aplikacije korisnik pred sobom ima prikaz jedne forme koja sadrži sledeće:

1. Lista knjiga koje su unete u bazu podataka
2. Mogućnost selektovanja knjige po izboru I čitanje njenih informacija
3. Mogućnost dodavanje nove knjige u bazu podataka
4. Mogućnost ažuriranje knjige u bazi podataka
5. Mogućnost brisanja knjige iz baze podataka
6. Pretraživanje određene knjige na osnovu njenog naziva

Izgled aplikacjie možemo videti na sledećim slikama:

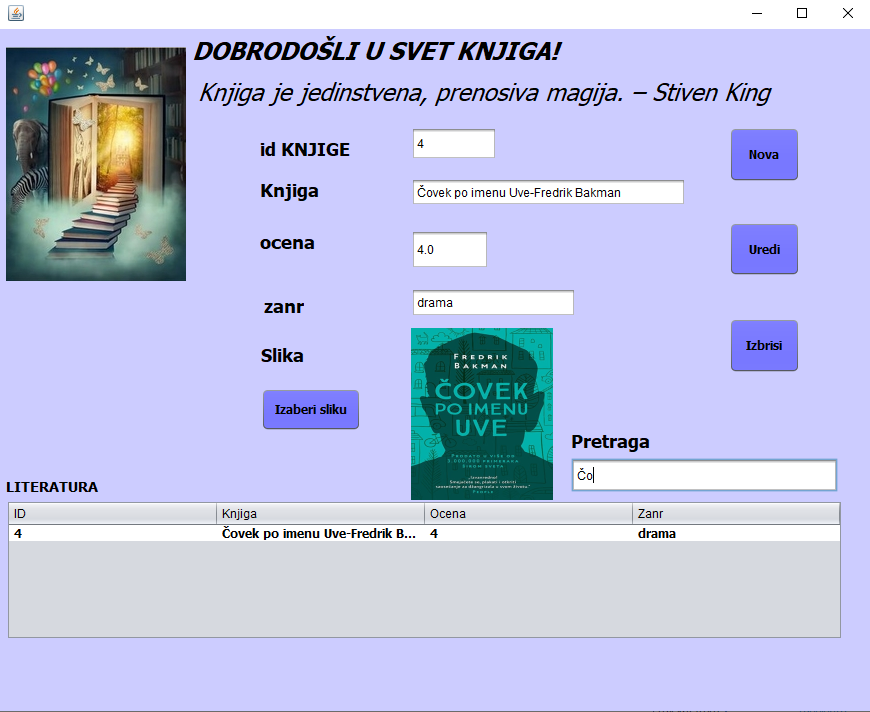
Slika 11 nam pokazuje šta vidi korisnik pri pokretanju aplikacije.

Slika 12 nam pokazuje kako se izvršava pretraživanje knjige.

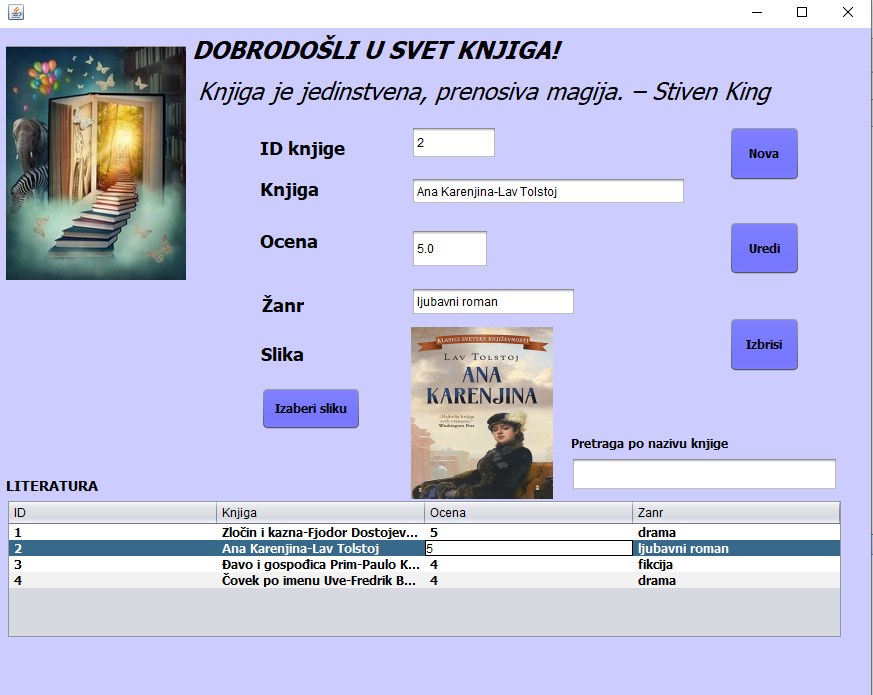
Slika 13 nam pokazuje primer jedne selektovane knjige.



Slika 11- prikaz aplikacije



Slika 12- prikaz aplikacije



Slika 13- prikaz aplikacije

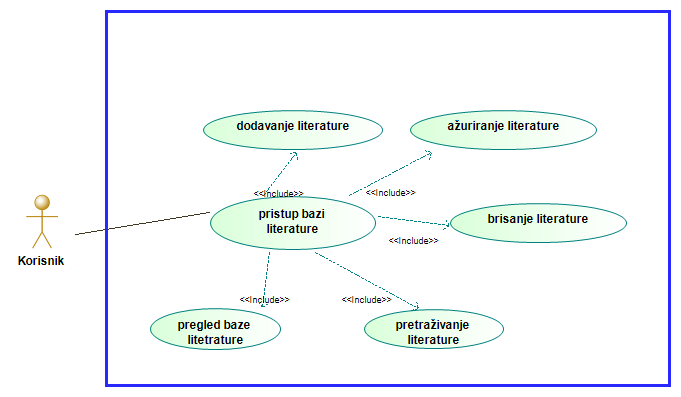
# UML dijagrami

The Unified Modeling Language ili skraćeno UML je standardni grafički jezik za modelovanje objektno-orjentisanog softvera.Unified Modeling Language (UML) je jezik za specifikaciju, vizuelizaciju, konstrukciju i dokumentovanje elemenata softverskog sistema, kao i poslovno modelovanje.Dijagrame koje obrađujemo su sledeći:

* dijagram slučajeva korišćenja;
* dijagram klasa;
* dijagram sekvenci;
* dijagram aktivnosti;
* dijagram stanja.

## Dijagram slučajeva korišćenja

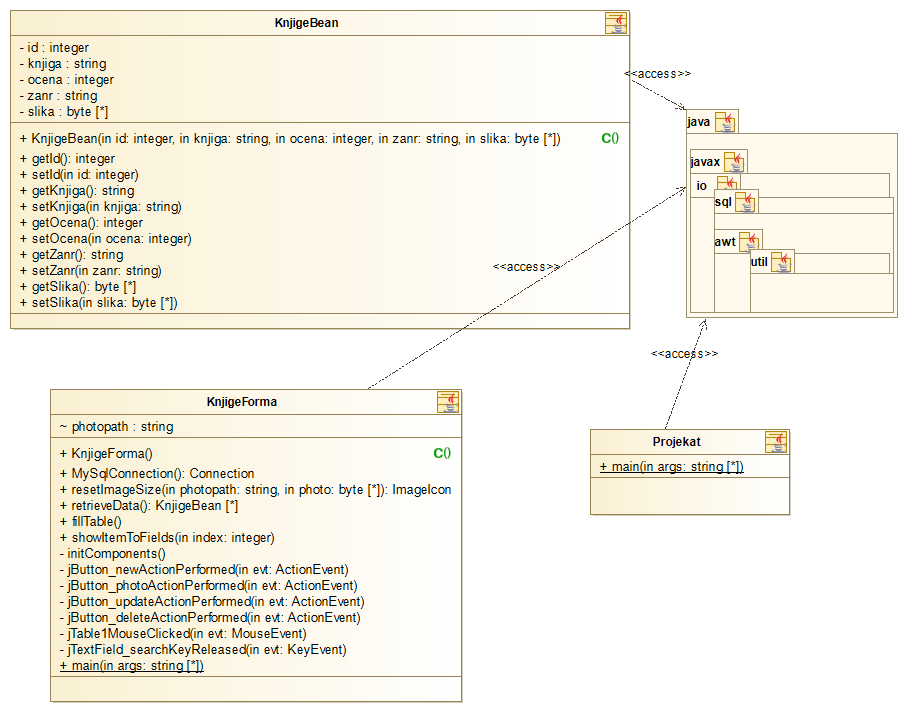
Dijagram slučajeva korišćenja (eng. use-case diagram) prikazuje skup slučajeva korišćenja i aktera.Slučaj korišćenja specificira šta subjekat radi,a ne kako radi.Slučajevi korišćenja predstavljeni su krugovima ili elipsama, a korisnici(akteri) predstavljeni su čovečuljcima.



Slika 14 Dijagram slučajeva korišćenja

## Dijagram klasa

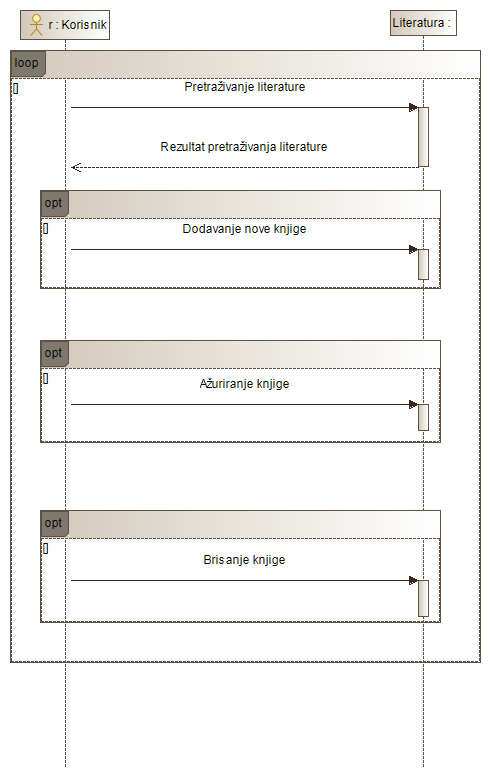
* Dijagram klasa prikazuje skup klasa, interfejsa, saradnji i drugih stvari strukture, povezanih relacijama. Simbol klase je pravougaonik podeljen horizontalnim linijama u odeljke(naziv klase, atributi, operacije, odgovornosti).
* Elementi dijagrama klasa su:
* stvari: klasa, interfejsi, tipovi, izuzeci, šabloni, saradnje, paketi
* relacije: zavisnosti, generalizacije, asocijacije, realizacije



Slika 15 Dijagram klasa, prvi deo

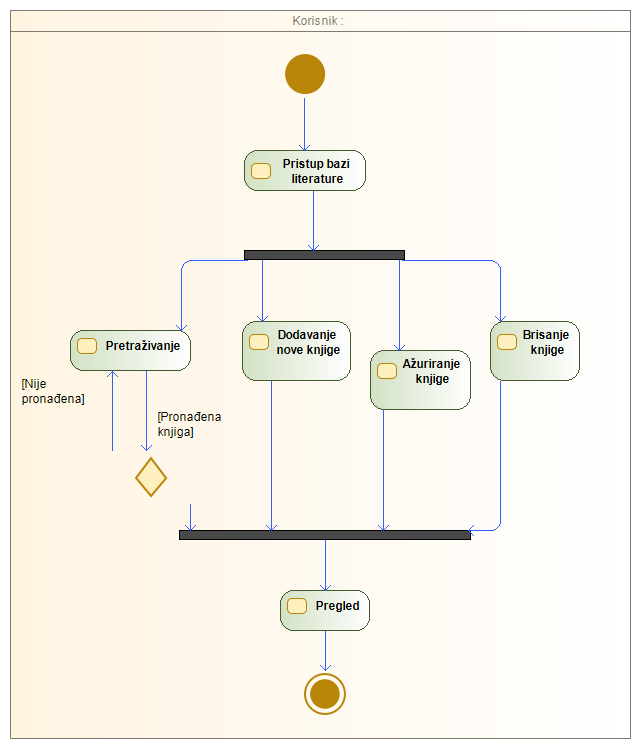
## Dijagram sekvenci

* Dijagram sekvenci(engl. sequence diagram) je UML dijagram interakcije.Dijagram sekvenci prikazuje komunikaciju između skupa objekata, koja se ostvaruje porukama koje objekti međusobno razmenjuju u cilju ostvarivanja očekivanog ponašanja. Dijagram sekvenci može da sadrži aktere, objekte i poruke. Dijagram sekvenci se može koristiti da bolje objasni slučajeve korišćenja sistema. Ovaj pristup je koristan kod analize zahteva jer može pomoći kod identifikacije klasa i asocijacija koje nisu uočene na početku.



Slika 16 Dijagram sekvenci

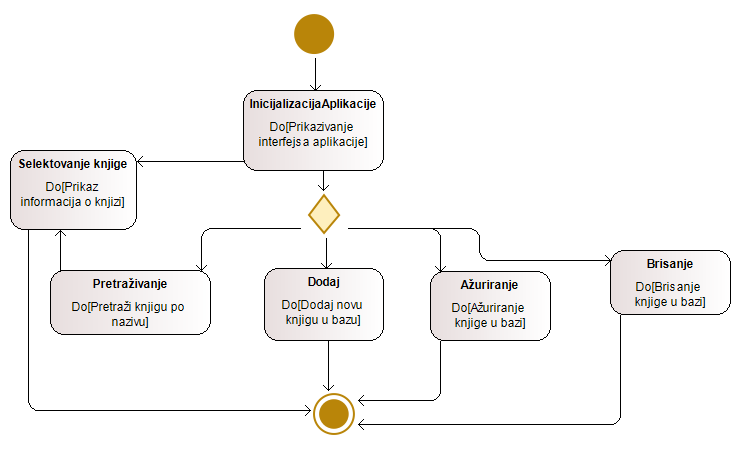
## Dijagram aktivnosti

* Dijagram aktivnosti je UML dijagram koji spada u grupu dijagrama posašanja, zajedno sa dijagramom stanja, dijagramom slučajeva upotrebe i dijagramima interakcije. On predstavlja tok u sistemu iz jedne aktivnosti u drugu. Slično kao ostali dijagrami ponašanja, dijagram aktivnosti se koristi za reprezentovanje dinamičkog ponašanja sistema. Njime se predstavljaju poslovni procesi višeg nivoa, tokovi podataka i složeni logički elementi sistema.
* Glavni elementi dijagrama aktivnosti • Akcije • Procesi • Tokovi podataka • Početne I završne tačke • Particije procesa • Fork • Join • Uslovne tačke • Merge • Objekt

Slika 17 Dijagram aktivnosti

## Dijagram stanja

* Automat stanja (state machine) – ponašanje koje specificira sekvence stanja kroz koja prolazi neki objekat ili interakcija, kao odgovor na događaje, proizvodeći akcije – modelira ponašanje nekog entiteta ili protokol interakcije.
* Stanje se označava pravougaonikom sa zaobljenim ivicama.
* On može da opiše moguća stanja objekta kako se događaji pojavljuju.
* Dijagram stanja se može upotrebiti da grafički predstavi automate konačnih stanja.



Slika 18 Dijagram stanja

# Zaključak

Tokom predavanja I vežbi ,kao I tokom izrade ovog projektnog zadatka imala sam prliku da se upoznam sa alatima I okvirima koji znatno olakšavaju razvoj softvera I omogućavaju brži razvoj aplikacije-Modeli nam pomažu da vizualiziramo sistem kakav jeste ili kakav želimo.Zanimljivo je što za svaku poslovnu ulogu može da se koristi poseban dijagram koji odgovara potrebama,bilo da se radi o klijentima,analitičarima ili programerima.To znači da se svaki dijagram radi sa različitim fokusima,do detalja.

# Literatura

1. 24.08.2021. <https://www.w3schools.com/java/default.asp>
2. 26.08.2021. <https://www.w3schools.com/asp/ado_connect.asp>
3. 26.08.2021 <https://www.tutorialspoint.com/jdbc/jdbc-db-connections.htm>
4. 24.08.2021. Literatura sa moodle portala,kurs- Softverski inženjering