

Naziv teme: Biblioteka



Članovi tima:

- Bektašević Kerim
- Ajkunić Harun
- Čukurića Džanan

1. Specifikacija projekta

1.1. Opis teme

Cilj ove aplikacije je da olakša način rada biblioteke na način da sve informacije budu dostupne na aplikaciji, kao i neke dodatne funkcionalnosti kao što su mogućnost filtriranja, sortiranja radi bolje preglednosti. Osim što korisnicima nudi lakšu saznavu informacija, ova aplikacija olakšava rad i bibliotekarima.

Nakon što neka osoba prvi put uđe u aplikaciju, može vršiti pregled knjiga i registraciju. Registracija obuhvata popunjavanje formulara(ime, prezime, broj telefona...) kao i choicebox na koliko mjeseci želi da se učlani, te način plaćanja.

Registrovani korisnik ima uvid u sve knjige biblioteke i njihovu trenutnu dostupnost, mogućnost pregleda najbolje ocijenjenih knjiga, bez da mora dolaziti lično u biblioteku kako bi dobio te informacije. Način na koji se vrši pozajmljivanje knjiga, tj. preko aplikacije je mnogo praktičniji nego što bi se uživo radilo. Nakon što odabere knjigu koju želi pozajmiti, dolazi u biblioteku radi preuzimanja. Nakon što korisnik pozajmi knjigu, dobit će obavijestjenje u aplikaciji do kada treba da vrati tu knjigu. Ukoliko želi, korisnik će moći poklanjati knjige biblioteci na način da popuni kratki formular o knjizi i ako bibliotekar odobri, onda odlazi u biblioteku da je dostavi.

Korisnici će moći ocijeniti knjigu, pa na taj način će drugi korisnici moći vidjeti koje knjige su najaktuelnije, odnosno najpopularnije trenutno.

Bibliotekar će imati uvid u sve pozajmljene knjige u biblioteci, te informacije o korisniku koji je ima. Moći će vršiti dodavanje, ažuriranje i brisanje knjiga, kao i korisnika.

Administrator će moći vršiti dodavanje novih ili brisanje ili editovanje postojećih biblioteka u sistem, kao i bibliotekara već postojećih biblioteka.

1.2. Zahtjevi

1.2.1. Funkcionalni zahtjevi

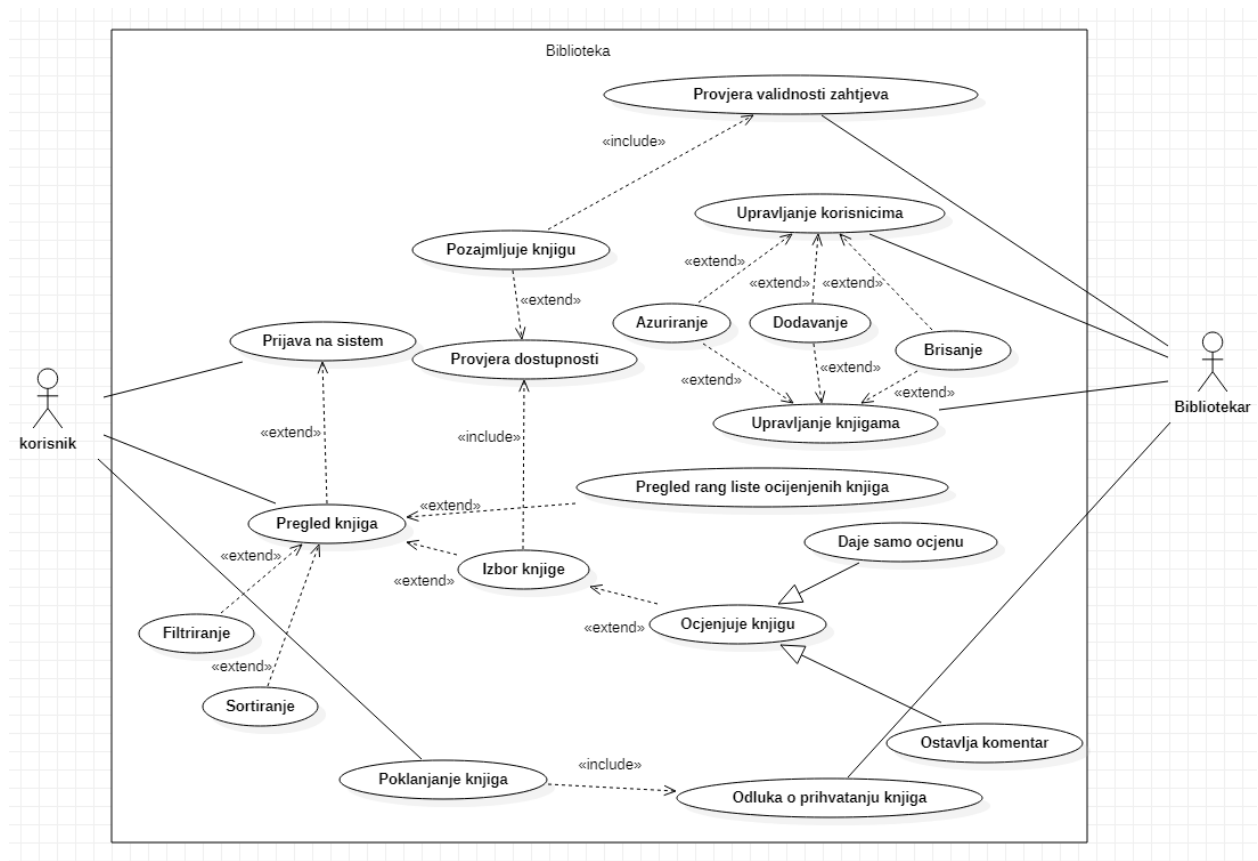
1. Pregled knjiga
2. Pozajmljivanje knjiga
3. Poklanjaju knjiga.
4. Prijava na sistem
5. Mogućnost ocjenjivanja i ostavljanja komentara za svaku knjigu.
6. Pregled dostupnosti knjiga

1.2.2. Nefunkcionalni zahtjevi

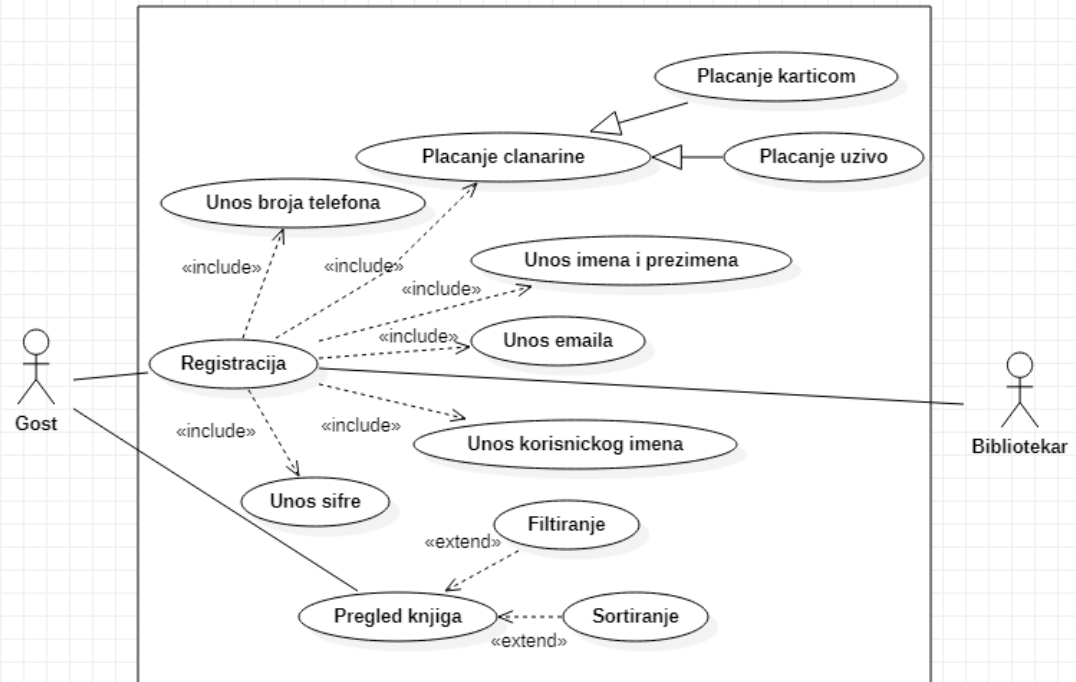
1. Prikaz osnovnih informacija o knjizi.
2. Privatnost na način da niko nema pristup informacijama drugih korisnika(osim njihovog korisničkog imena).
3. Pregled knjiga na sortiran ili filtiran način.
4. Mogućnost korištenja aplikacije na telefonu i na PC-u
5. Knjige i njihove kolicine su regularno ažurirane

2. Use Case diagram

Opći use case diagram



Use case diagram za gosta



3. Scenariji

3.1. Registracija korisnika

3. Registracija korisnika

Registracija podrazumijeva unos osnovnih informacija o korisniku (ime, prezime, e-mail...). Nakon unosa korisnik bira "paket" članarine odnosno da li će platiti mjesečnu ili godišnju članarinu, te na koji način će izvršiti uplatu iste. Ako se odluči za plaćanje preko kreditne kartice, dodavanje korisničkog računa u sistem se vrši automatski, dok ako želi uplatu izvršiti uživo korisnik se dodaje od strane bibliotekara nakon izvršene uplate.

Naziv	Registracija korisnika
Opis	Registracija novog korisnika na aplikaciju kako bi iskoristio pogodnosti naše aplikacije
Vezani zahtjevi	/
Preduvjeti	/
Posljedice - uspješan završetak	Uspješna registracija
Posljedice - neuspješan završetak	Neuspješna registracija
Primarni akteri	Korisnik, sistem
Ostali akteri	/
Glavni tok	Korisnik unosi potrebne podatke, bira paket članarine, šalje zahtjev
Proširenje/alternative	Korisnik unese pogrešne podatke (e-mail, korisničko ime, šifra ne zadovoljava kriterije)

Glavni tok

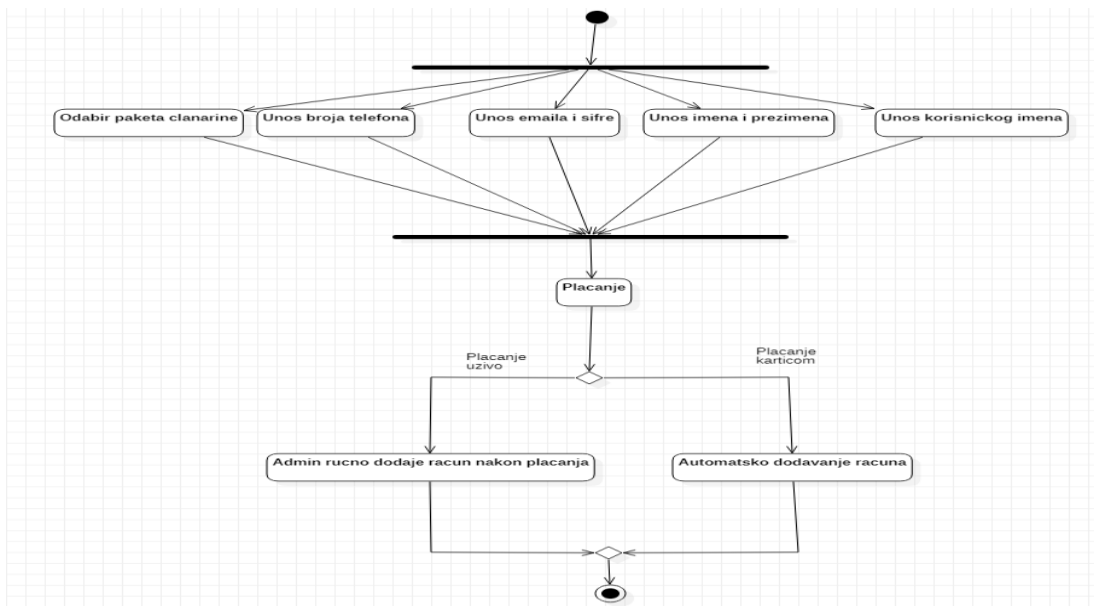
Korisnik	Sistem
1. Otvara formu za registraciju	2. Prikazivanje forme za registraciju
3. Unosi potrebne podatke	4. Validacija unesenih podataka
	5. Slanje povratne informacije o uspješnoj registraciji korisnika te prebacivanje na formu za prijavu
6. Završava interakciju sa sistemom/prijavljuje se na sistem	

Alternativni tok

Korisnik unese pogrešan/zaузet e-mail ili korisničko ime, unese šifru koja ne zadovoljava kriterije(ne sadrži više od 8 karaktera, veliko i malo slovo, specijalni znak itd.)

Korisnik	Sistem
	1. Zahtjev se ne prihvata, zbog pogrešno unesenih podataka
	2. Obavješćavanje korisnika o počinjenoj grešci
3. Nastavak na koraku 3. glavnog toka	

Dijagram aktivnosti



3.2. Poklanjanje knjige

2. Poklanjanje knjige

Svaki prijavljeni korisnik ima opciju poklanjanja knjige. Klikom na tu opciju otvara se formular u koji korisnik unosi podatke o knjizi. Nakon popunjavanja formulara korisnik šalje zahtjev i čeka na povratnu informaciju od strane bibliotekara koji treba da odobri ili odbije primljeni zahtjev. Bibliotekar preko sistema šalje povratnu informaciju korisniku, ukoliko je odobren zahtjev korisnik može doći u biblioteku da dostavi knjigu.

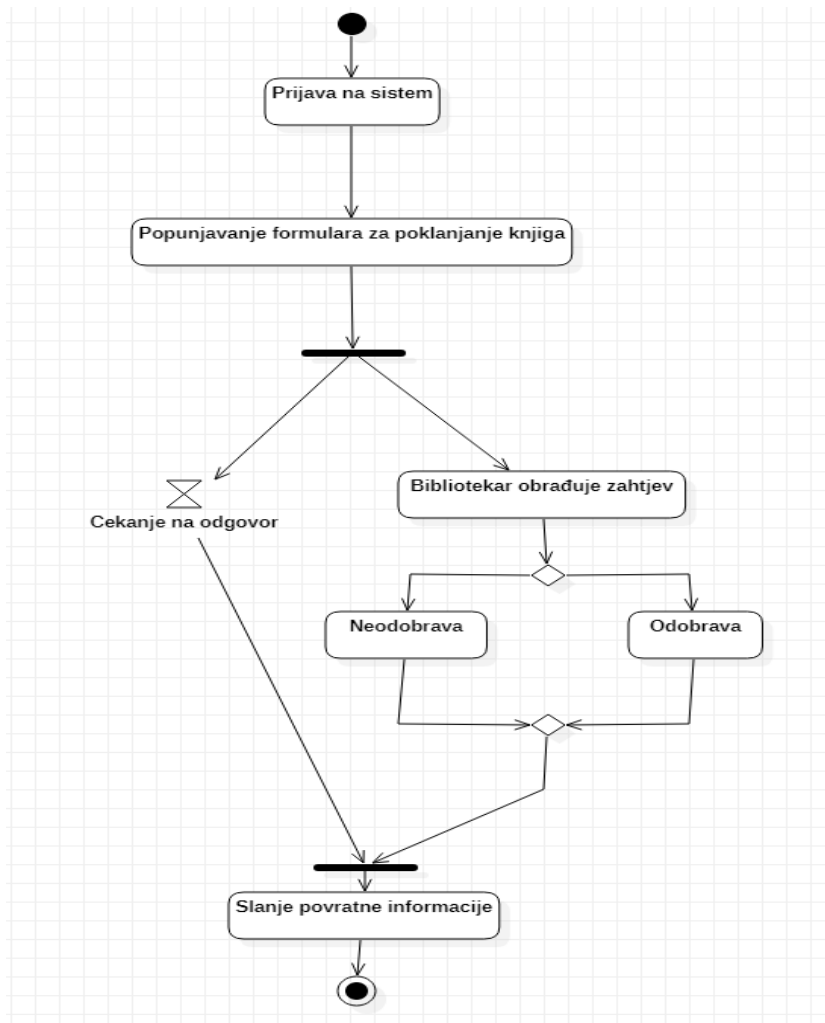
Naziv	Poklanjanje knjige
Opis	Registrovani korisnik poklanja knjigu biblioteci
Vezani zahtjevi	/
Preduvjeti	Korisnik mora biti registrovan
Posljedice - uspješan završetak	Korisnik uspješno poklanja knjigu
Posljedice - neuspješan završetak	Odbijen korisnički zahtjev za poklanjanje knjige
Primarni akteri	Registrovani korisnik, sistem
Ostali akteri	/
Glavni tok	Prijavljivanje na sistem, popunjavanje formulara za poklanjanje knjige, slanje zahtjeva
Proširenje/alternative	/

Registrovani korisnik	Sistem biblioteka
1. Prijava na sistem	2. Prikaz knjiga sa kojima raspolaže biblioteka
3. Popunjavanje formulara za poklanjanje knjige	4. Validacija zahtjeva
	5. Slanje povratne informacije korisniku da je zahtjev prihvaćen
6. Kraj interakcije sa sistemom, korisnik može pokloniti knjigu	

Tok događaja 1.2 - Neuspješno poklanjanje knjige

Registrovani korisnik	Sistem biblioteka
1. Prijava na sistem	2. Prikaz knjiga sa kojima raspolaže biblioteka
3. Popunjavanje formulara za poklanjanje knjige	4. Validacija zahtjeva
	5. Slanje povratne informacije korisniku da je zahtjev odbijen
6. Kraj interakcije sa sistemom.	

Dijagram aktivnosti



3.3. Registracija knjige

4. Registracija knjige

Bibliotekar provjera da li knjiga već postoji u sistemu, ako da samo ažurira količinu knjiga.

Ukoliko se knjiga ne nalazi u sistemu, unosimo podatke knjige(naziv, pisac, žanr, količina), i dodaje u bazu.

Nakon završenog dodavanja, prikaže se data knjiga u pregledniku.

Naziv	Registracija knjige
Opis	Registracija nove ili već postojeće knjige u bazu knjiga
Vezani zahtjevi	/
Preduvjeti	/
Posljedice - uspješan završetak	Uspješna registracija
Posljedice - neuspješan završetak	/
Primarni akteri	Bibliotekar, sistem
Ostali akteri	/
Glavni tok	Bibliotekar provjerava bazu registruje novu knjigu ukoliko ona ne postoji
Proširenje/alternative	Ukoliko knjiga već postoji ažurira količinu knjiga

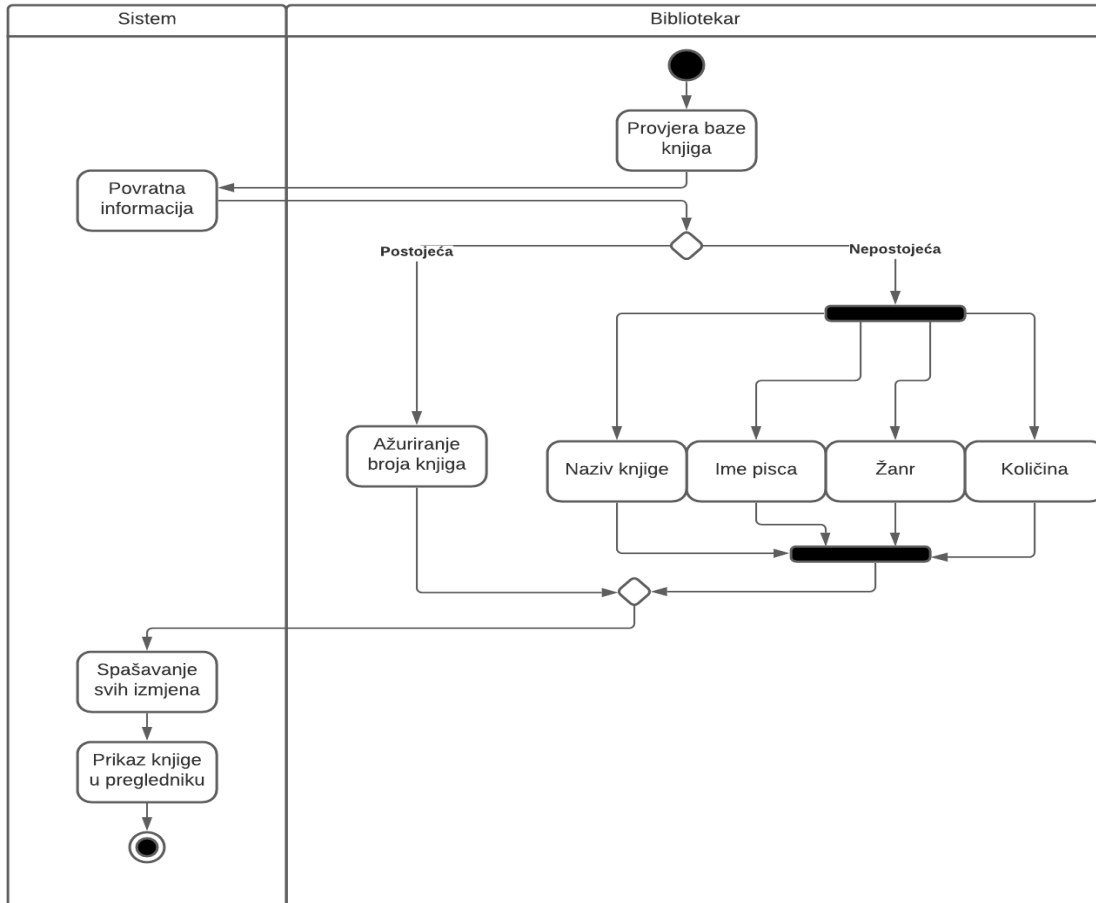
Glavni tok

Bibliotekar	Sistem
1. Provjerava bazu knjiga	2. Sistem vraća informaciju da li se knjiga već nalazi u bazi
3. Unosi podatke za knjigu(naziv, ime pisca, žanr, količina)	4. Spašavanje knjige/izmjena
	5. Prikaz knjige u pregledniku

Alternativni tok

Bibliotekar	Sistem
1. Provjerava bazu podataka	2. Sistem vraća informaciju da li se knjiga već nalazi u bazi
3. Ažurira broj knjiga	4. Spasava izmjene
	5. Prikaz knjige u pregledniku

Dijagram aktivnosti



3.4. Pozajmljivanje knjige

1. Pozajmljivanje knjige

Scenarij predstavlja način na koji korisnik može pozajmiti knjigu u našoj aplikaciji. Korisnik se prijavljuje na sistem, ako nema kreiran nalog potrebno je da isti kreira da bi mogao pristupiti procesu pozajmljivanja knjige. Nakon uspješne prijave/registracije korisnik vrši izbor knjige koju želi i potvrđuje pozajmljivanje.

Naziv	Pozajmljivanje knjige
Opis	Registrovani korisnik pozajmljuje odabranu knjigu
Vezani zahtjevi	/
Preduvjeti	Korisnik mora biti registrovan
Posljedice - uspješan završetak	Korisnik uspješno pozajmljuje knjigu
Posljedice - neuspješan završetak	Odbijen korisnički zahtjev za pozajmljivanje knjige
Primarni akteri	Registrovani korisnik, sistem
Ostali akteri	/
Glavni tok	Prijavljivanje na sistem, pretraživanje željene knjige, slanje zahtjeva za pozajmljivanje
Proširenje/alternative	/

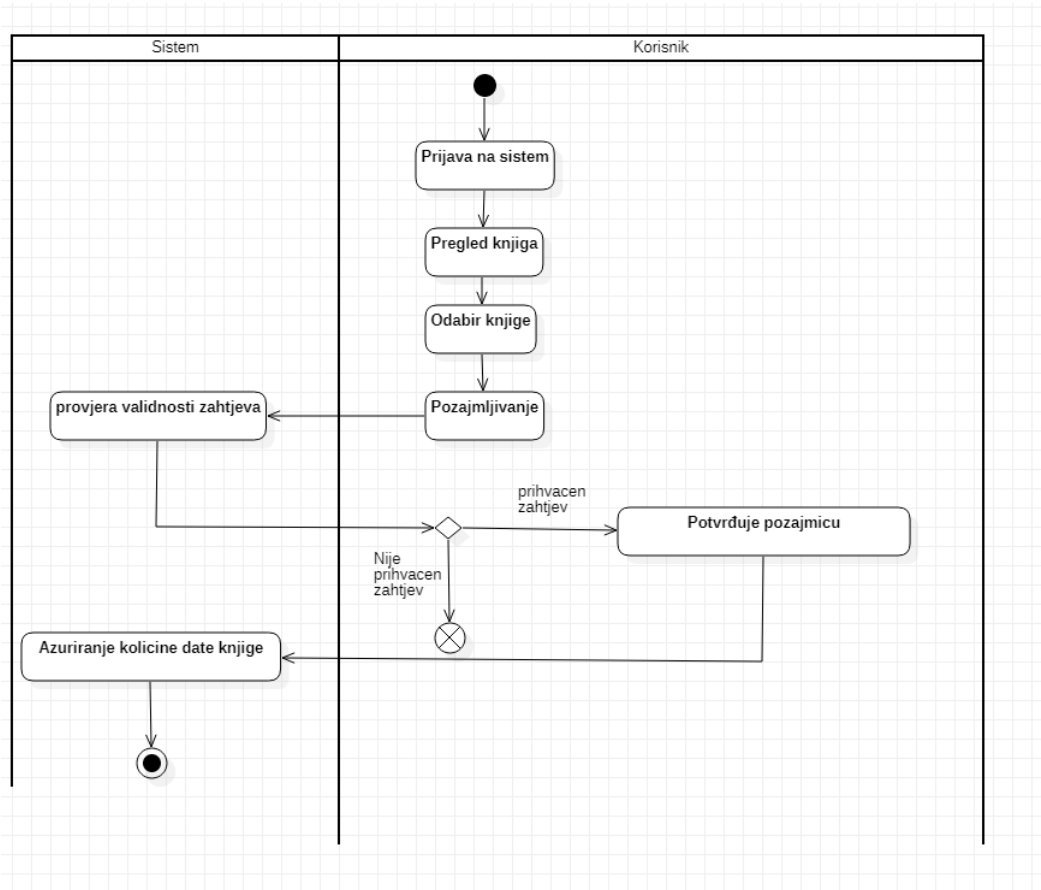
Tok događaja 1.1 - Uspješno pozajmljivanje knjige

Registrovani korisnik	Sistem biblioteka
1. Prijava na sistem	2. Prikaz knjiga sa kojima raspolaže biblioteka
3. Odabir/pretraga željene knjige i slanje zahtjeva za pozajmljivanje	4. Validacija zahtjeva
	5. Slanje povratne informacije korisniku da je zahtjev prihvaćen
6. Kraj interakcije sa sistemom, korisnik može preuzeti knjigu	

Tok događaja 1.2 - Neuspješno pozajmljivanje knjige

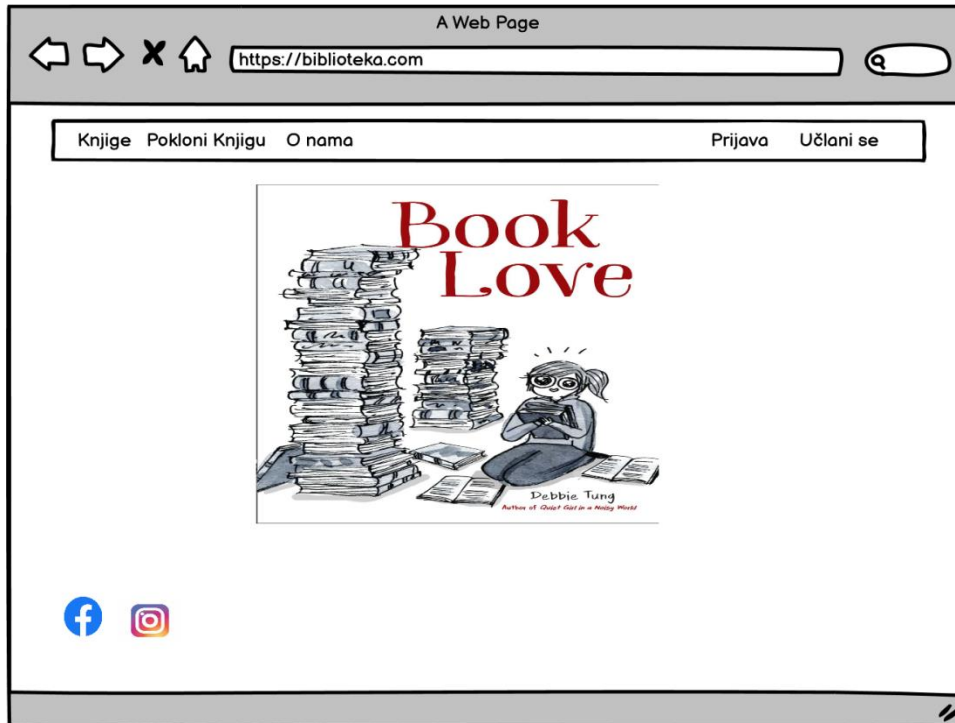
Registrovani korisnik	Sistem biblioteka
1. Prijava na sistem	2. Prikaz knjiga sa kojima raspolaže biblioteka
3. Odabir/pretraga željene knjige i slanje zahtjeva za pozajmljivanje	4. Validacija zahtjeva
	5. Slanje povratne informacije korisniku da je zahtjev odbijen
6. Kraj interakcije sa sistemom.	

Dijagram aktivnosti

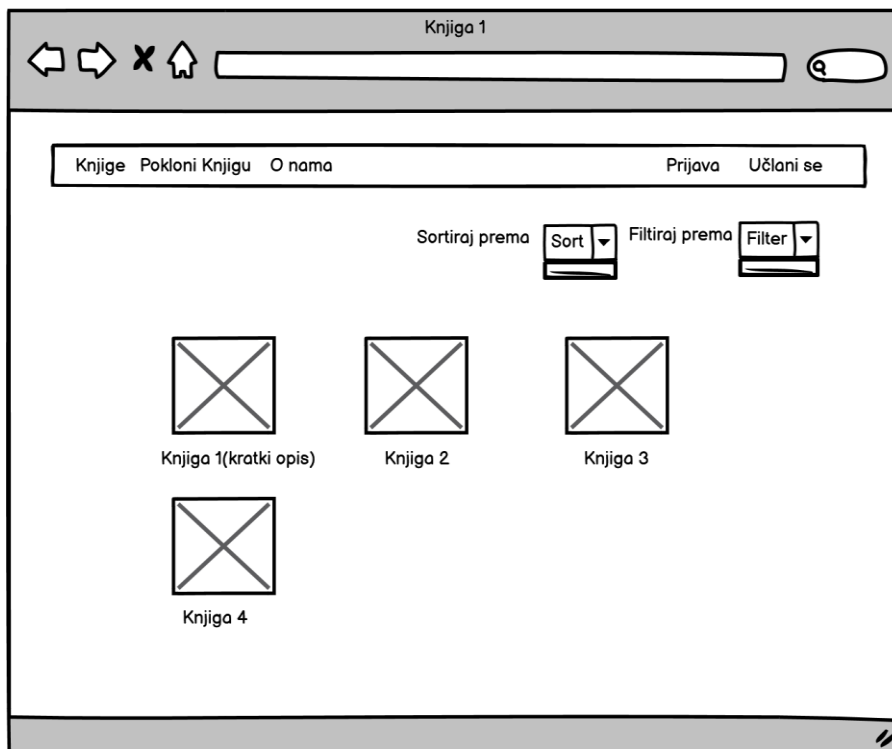


4. Prototip

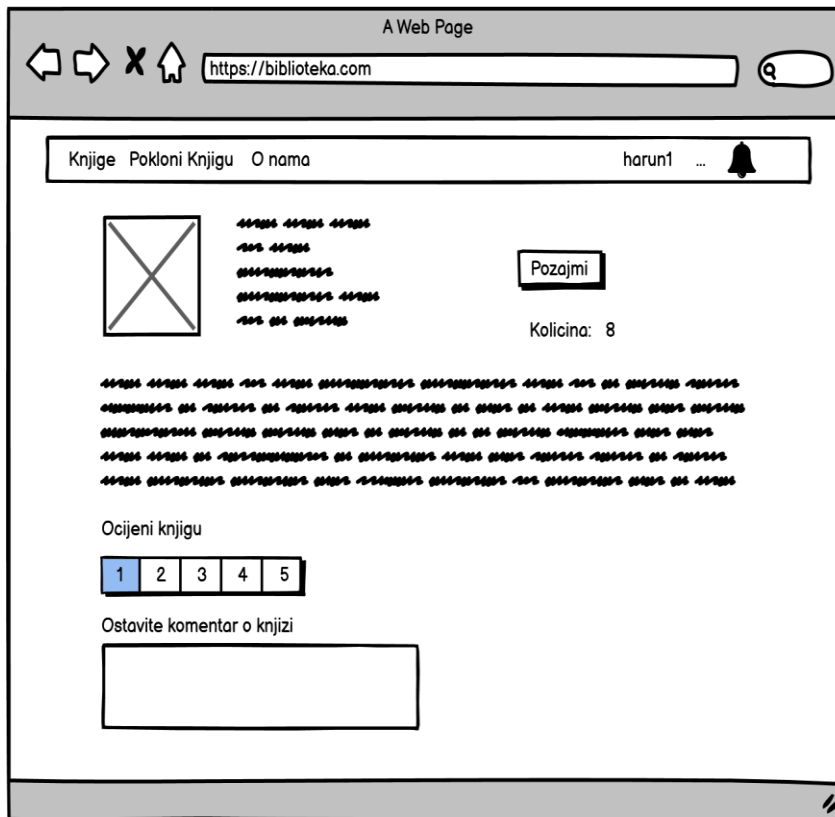
4.1. Glavna stranica



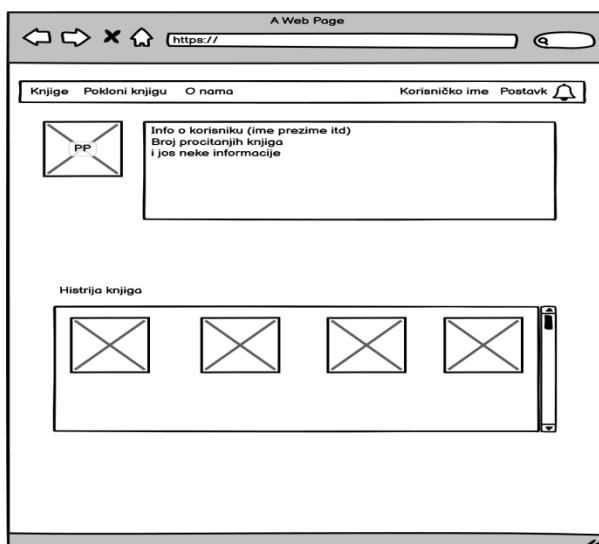
4.2. Pregled knjiga



4.3. Detalji knjige



4.4. Profil



4.5. Registracija

A Web Page

X

https://biblioteka.com

Q

Knjige Pokloni Knjigu O nama

Prijava Učlani se

Ime

Prezime

broj telefona

email

username

sifra

Dodaj sliku

f

Sljedece

A Web Page

https://biblioteka.com

Knjige Pokloni Knjigu O nama Prijava Učlani se

Paketi

Mjesec/10KM Odaberi

Godina/100KM Odaberi

Nacin placanja

Nacin placanja ▼

Karticom

Uzivo

Završi registraciju

f i

4.6. Poklanjanje knjige

A Web Page

https://

Knjige Pokloni knjigu O nama Korisničko ime Postavke

Naziv knjige

Pisac

žanr

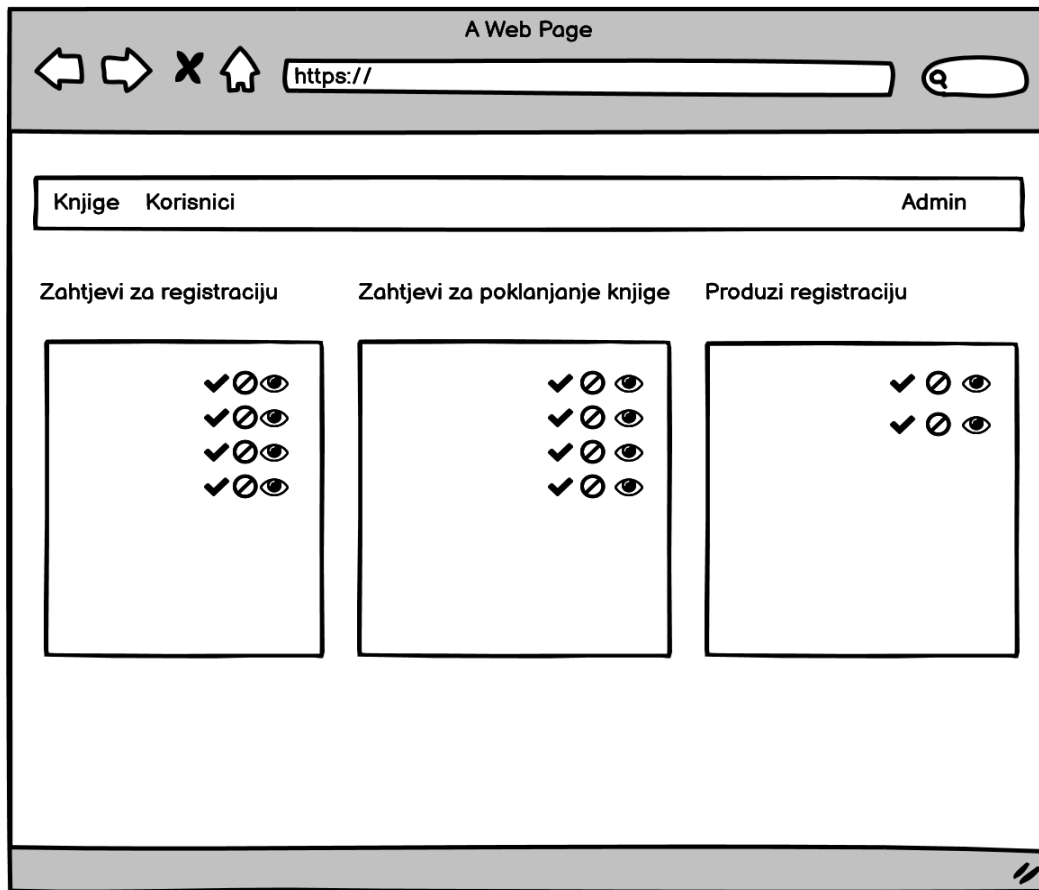
Kratki opis

Količina 0 ▼

Dodaj sliku Browse


Potvrdi

4.7. Admin interfejs




4.8. Registracija knjige

A Web Page



https://



Knjige

Korisnici

Admin

Registracija knjige

Naziv knjige

Pisac

žanr

Kratki opis

Količina

0

Dodaj sliku

Browse

Potvrdi

4.9. Zahtjev za poklanjanje

A Web Page

X

https://

Q

Knjige Korisnici Admin

Zahtjevi za poklanjanje knjige

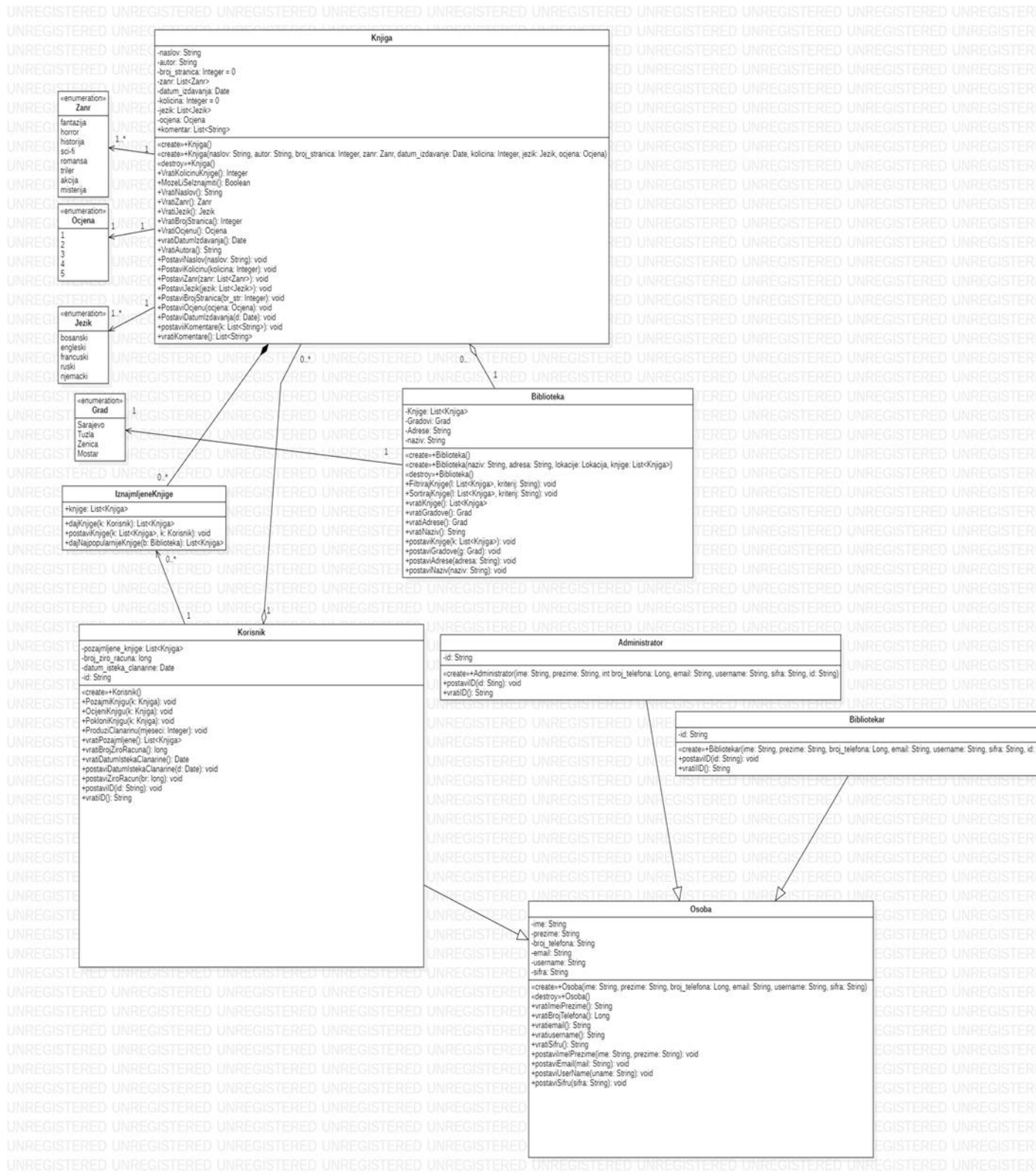
Pregled informacija o knjizi, slike knjige, stanje knjige itd.

Potvrdi

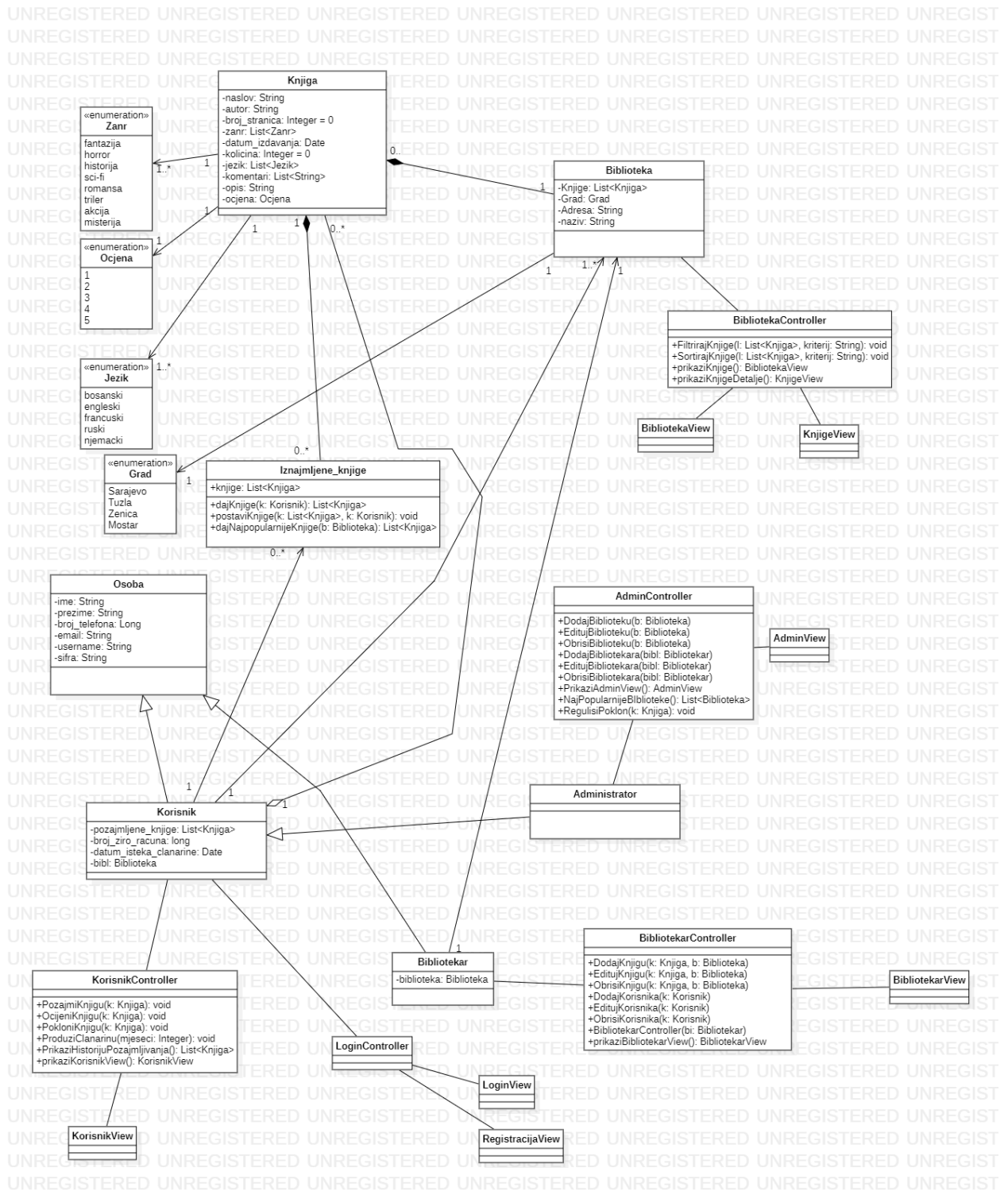
Odbij

< 1/N >

5. Dijagram klasa



5.1. MVC dijagram klasa



6. Solid principi

Single-responsibility princip zahtijeva da svaka klasa ima svoju odgovornost.

Naše klase sadržavaju konstruktore,gettere,settere i atributi se mogu samo mijenjati unutar te date klase tako da princip je zadovoljen.

Open closed principle kaže da klase trebaju biti otvorene za nadogradnju bez modifikovanja klase.

U našem slučaju klase su jednostavne. Ako promijenimo jednu klasu, ne može doći do promjene druge klase, tako da princip je zadovoljen.

Liskov princip kaže da izvedene klase možemo zamijeniti baznim, a da se rezultat ne promijeni.

Ovaj princip nije zadovoljen. Imamo baznu klasu Osoba i tri izvedene, i korisnik ima svoje metode koje klasa "osoba" nema, tako da objekat tipa 'Osoba' ne može zamijeniti ni jednu izvedenu.

Princip izoliranja interfejsa

Ovaj princip se koristi kada imamo ogromne klase, pa dosta metoda nisu upotrebljive i u tom slučaju pravi se interfejs.

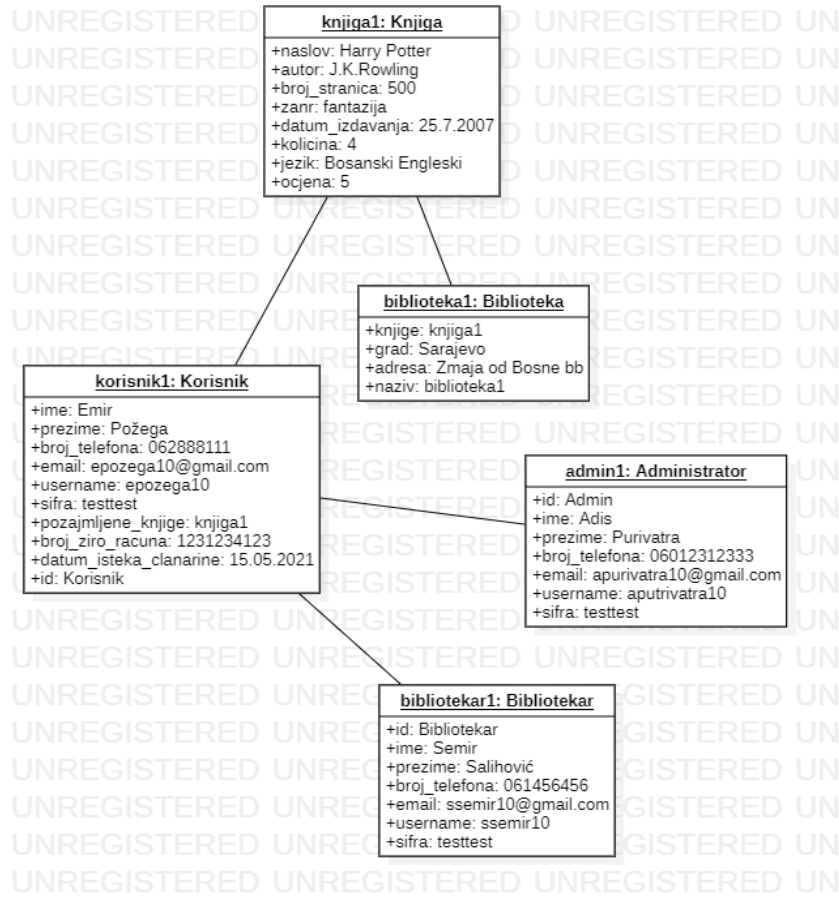
Naša aplikacija je jednostavna i nema ni jednog interfejsa i klase nisu kompleksne.

Nema nepotrebnih metoda ni u jednoj klasi.

Princip inverzije ovisnosti

Ovaj princip nalaze da ne treba ovisiti o izvedenim tj. konkretnim klasama, već o apstrakcijama jer se one rijetko mijenjaju. Princip je zadovoljen, jer sve izvedene klase zavise od bazne, a nema ni jedan slučaj gdje postoji zavisnost od izvedenih, tj. konkretnih klasa.

7. Dijagram objekata



8. Strukturalni paterni

1. Adapter pattern

Osnovna namjena Adapter paterna je da omogući širu upotrebu već postojećih klasa. Jedna od osnovnih funkcionalnosti naše aplikacije jeste prikaz knjiga date biblioteke. Elementi klase Knjiga se trebaju prikazati na stranci i da bi se to izvelo neophodno je napraviti adapter preko kojeg će se ta lista prikazivati, pri čemu se knjige dohvaćaju iz baze podataka. Još jedan način gdje možemo iskoristiti adapter pattern je prilikom dodavanja slika na sistem, bilo od strane bibliotekara pri dodavanju knjiga ili korisnika tokom registracije na način da se mogu dodavati .jpg i .png slike. 2.

2. Dekorativni pattern

Osnovna namjena Dekorativnog paterna je da omogući dinamičko dodavanje novih elemenata i ponašanja (funkcionalnosti) postojećim objektima. U našem projektu korisnik ima mogućnost pregleda i promjene atributa na svom profilu, kao što su ime, prezime, sifru, email, broj telefona, korisničko ime, te mogućnost promjene slike. Dekorativni pattern se može još iskoristiti prilikom ocjenjivanja knjiga, gdje korisnik može dati ocjenu i ostaviti komentar za svaku knjigu u biblioteci.

3. Fasadni pattern

Glavna namjena fasadnog patterna je za prikaz pogleda bez potrebnog objašnjenja kako je implementirano. U našem projektu ga možemo iskoristiti za prikaz filtriranih knjiga.

4. Flyweight pattern

Prilikom kreiranja instance klase Knjiga i nakon što inicijalizujemo sve njene attribute, neki atributi se mogu promijeniti naknadno kao što su jezik i datum izdavanja, dok drugi ne mogu. Mogu se kreirati podtipovi knjiga na ovaj način. 5. Proxy pattern Prilikom prikaza liste knjiga biće prikazana ista slika za sve knjige, međutim kada korisnik klikne na određenu knjigu pokazat će se slika te date knjige.

6. Bridge pattern

Bridge pattern bi mogli iskoristiti prilikom plaćanja članarine za korisnike. Ako korisnik odluči da plaća karticom biće regulisan popust na cijenu članarine, a ako odluči da plaća uzivo onda neće biti popusta.

7. Composite pattern

Sa obzirom da u našem sistemu postoje korisnici sa popustom clanarine i korisnici bez popusta, i sada ako admin npr. zeli da isprinta izvještaj sa cijenama clanarine, potrebno je računati cijenu clanarine posebno za korisnike sa popustom i posebno za korisnike bez popusta.

9. Kreacijski paterni

1. Factory Method patern

Uloga Factory Method paterna je da omogući kreiranje objekata na način da podklase odluče koju klasu instancirati. Ovaj patern se može primjeniti pri iznajmljivanju knjiga. Konkretno na primjer is predavanja možemo upotrijebiti analogiju. Primjer: Trgovački centar nabavlja avokado uvijek preko istog dobavljača-servisa. U ovisnosti od godišnjegdoba dobavljač nabavlja avokadoiz različitih zemalja. U ovom primjeru trgovački centar predstavlja korisnika biblioteke, avokado predstavlja knjigu, dobavljač je bibliotekar, a različite zemlje predstavljaju različite biblioteke.

2. Singleton patern

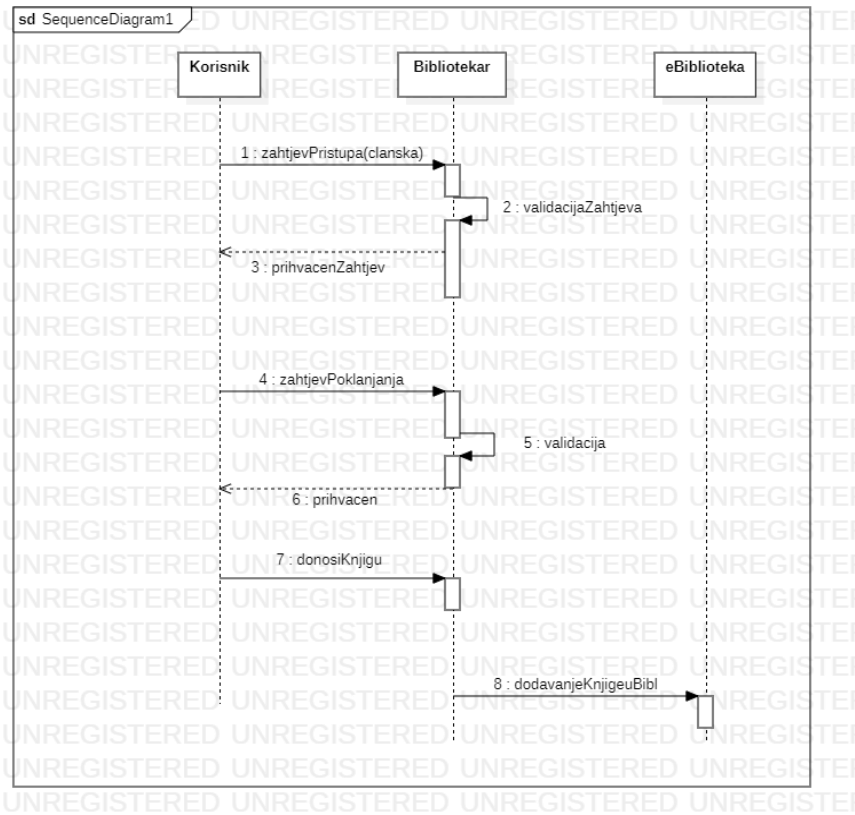
Uloga Singleton paterna je da osigura da se klasa može instancirati samo jednom i da osigura globalni pristup kreiranoj instanciklase. Ovaj patern se u našoj aplikaciji može primjeniti za kreiranje klase Administrator. Potrebno je da postoji samo jedna instanca i da ona upravlja kompletnim sistemom(globalno).

3. Prototype patern

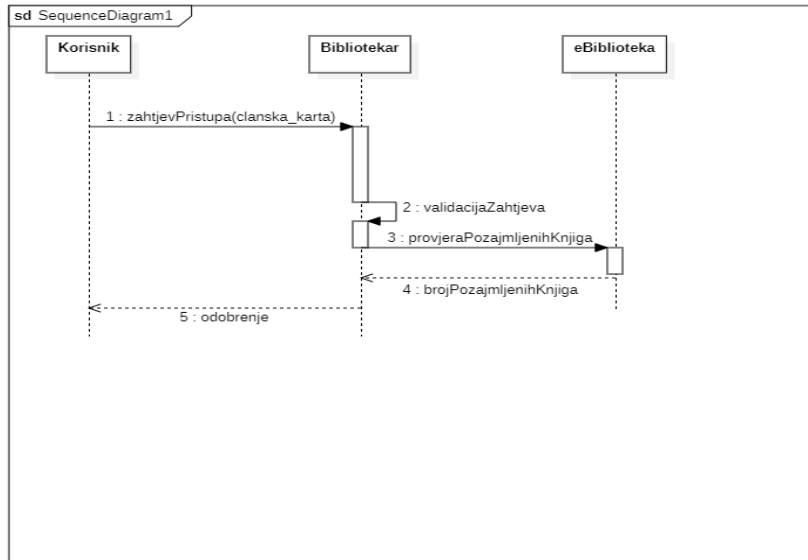
Uloga Prototype paterna je da kreira nove objekte klonirajući jednu od postojećih prototip instanci(postojeći objekat). Ovaj patern se može primjeniti pri kreiranju novih instanci knjiga, pa čak i korisnika. Kloniranjem već postojećeg prototipa za knjige i korisnike, i izmjena specifičnih podataka ćemo postići bolje memorijske performanse.

10. Dijagram sekvenci

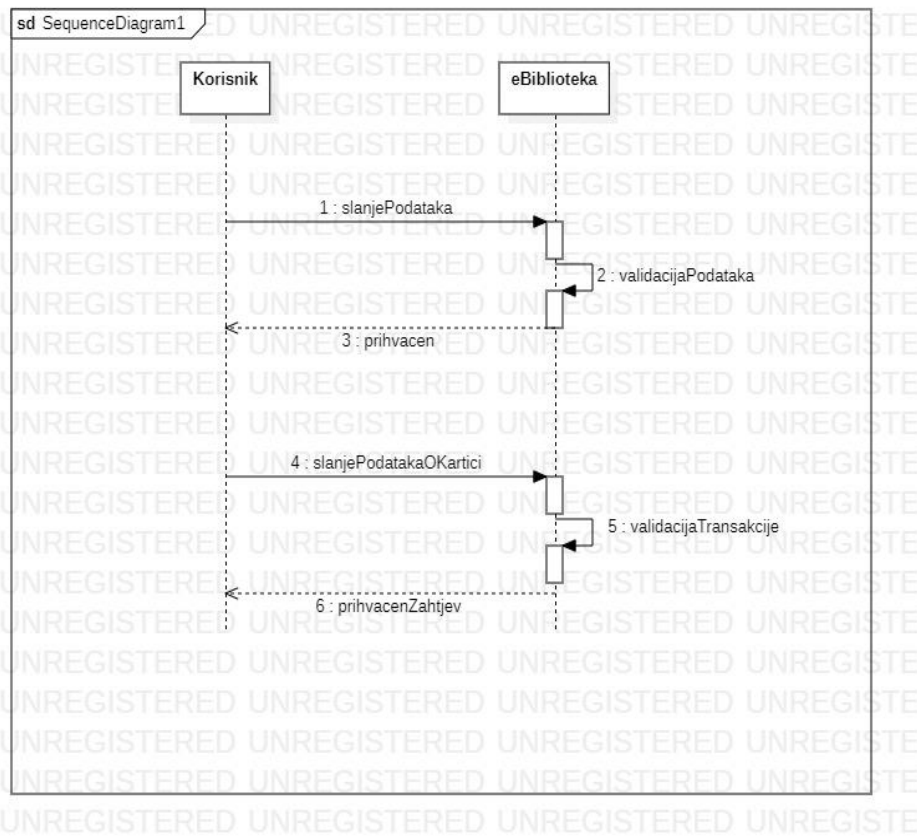
10.1. Poklanjanje knjige



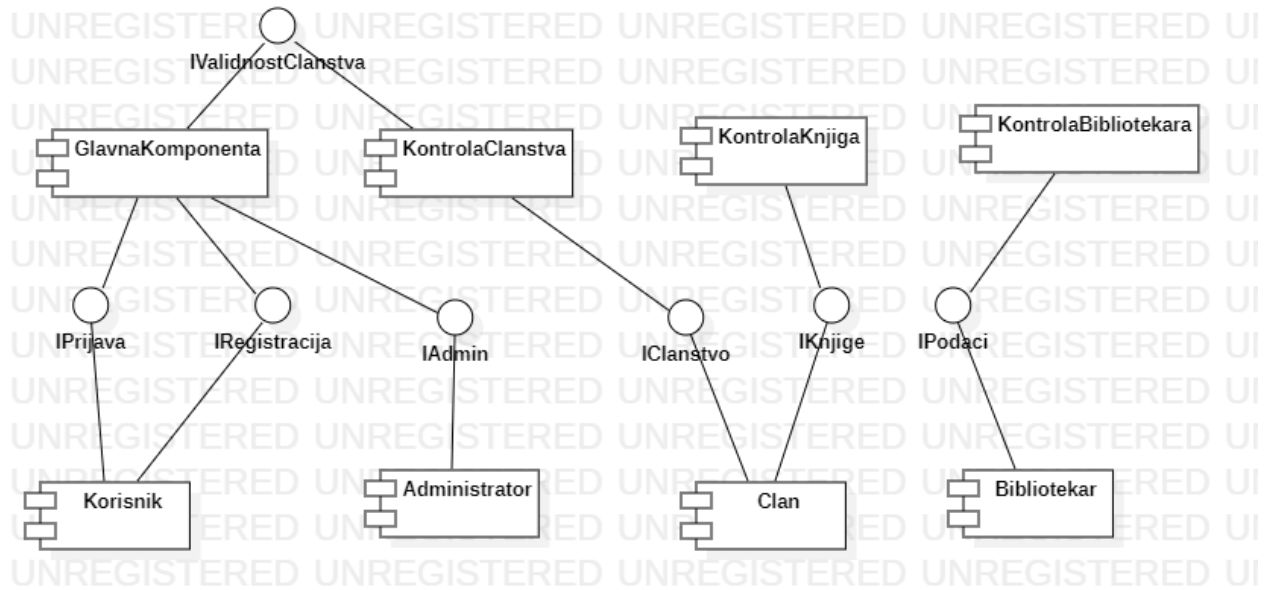
10.2. Pozajma knjige



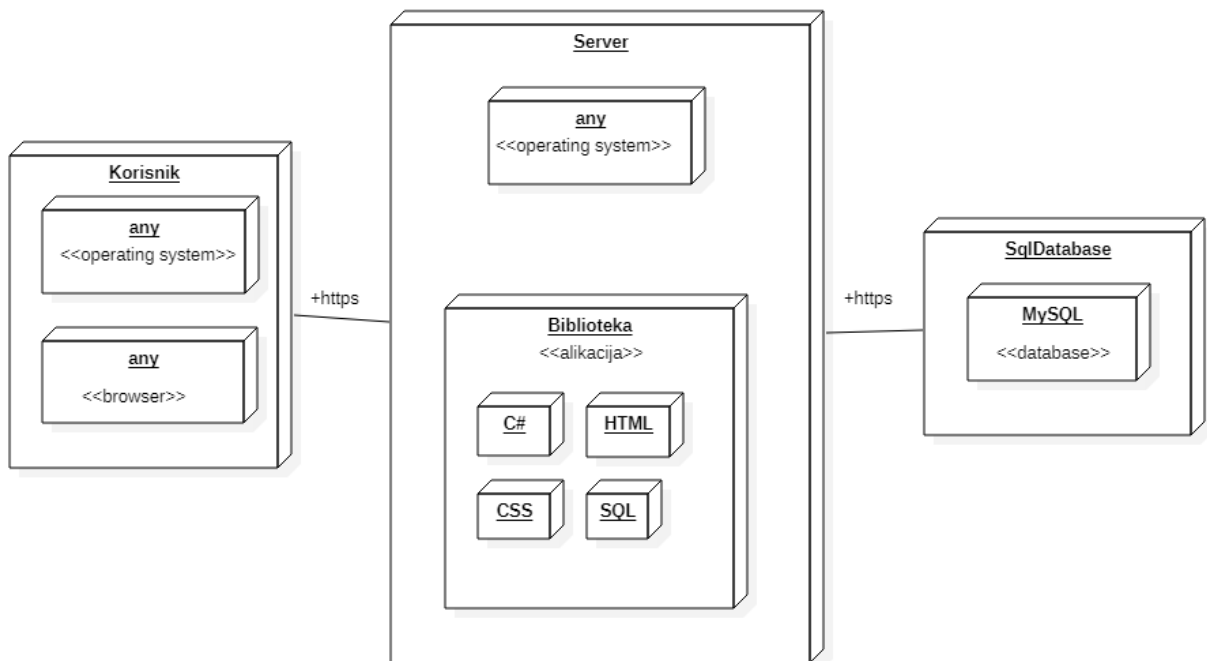
10.3. Registracija knjige



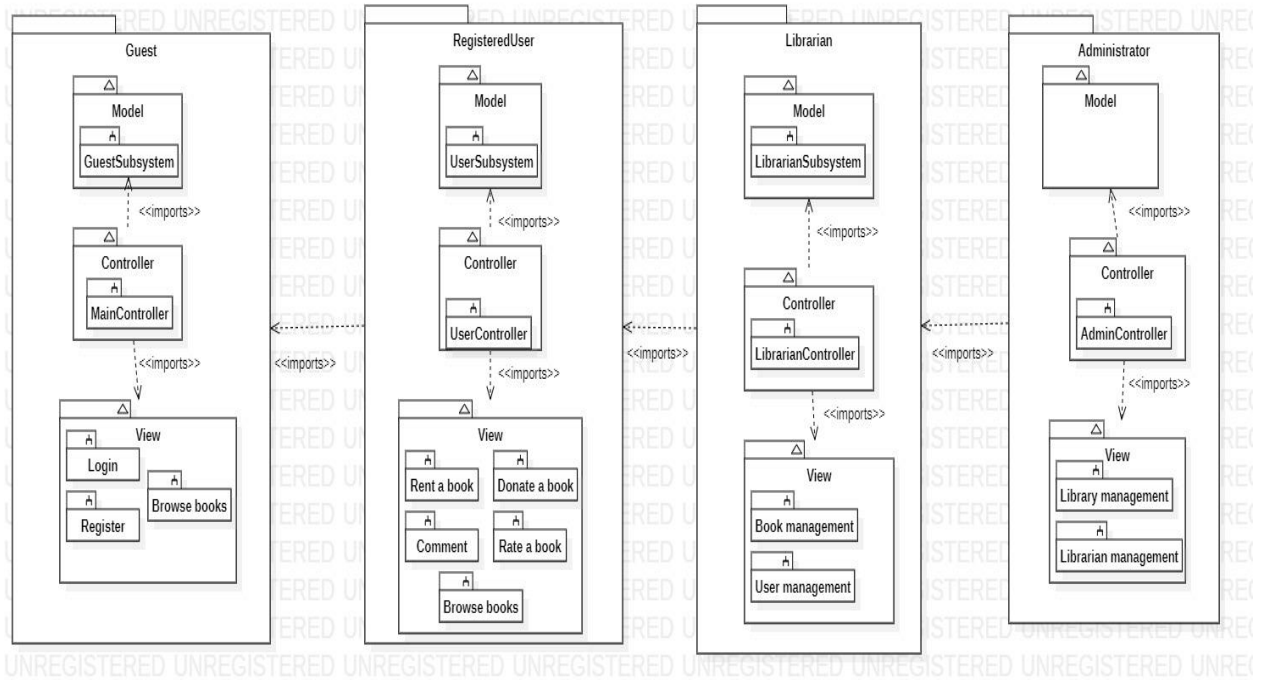
11. Dijagram paketa, komponenti i raspoređivanja



Dijagram komponenti



Dijagram raspoređivanja



Dijagram paketa

12. ERD

