

**Školska godina 2021/2022**

*Sistem za biblioteke*

*IT350*

*Baze podataka*

*Projektna dokumentacija*

***Student****:*

Bojana Stajić 4596

***Asistent****:*

Milica Vlajkovic

Sadržaj

[Apstrakt 3](#_Toc30784438)

[Opis problema 3](#_Toc30784439)

[Zahtevani upiti 3](#_Toc30784440)

[Uvod 4](#_Toc30784441)

[Cilj 4](#_Toc30784442)

[Realizacija 4](#_Toc30784443)

[Projektno rešenje 5](#_Toc30784444)

[Konceptualni model 5](#_Toc30784445)

[Logički model 6](#_Toc30784446)

Fizički model…………………………………………………………………………………………………………………………………….7

[Kreiranje baze i normalizacija 8](#_Toc30784447)

[Generisanje Java koda 8](#_Toc30784448)

[Rešenja](#_Toc30784450) upita……………………………………………………………………………………………………………………………………9

[Aplikacija 10](#_Toc30784451)

Zaključak ………………………………………………………………………………………………………………………………………………14

Literatura ……………………………………………………………………………………………………………………………………………..14

# Apstrakt

## Opis sistema

## Državna uprava je odlučila da im je potreba aplikacija za potrebe svih državnih biblioteka.

## Aplikacija mora da čuva podatke o svim bibliotekama po opštinama u kojima se nalaze. Za svaku opštinu

## može se zaključiti kom regionu pripada i dobiti naziv regiona. Naziv opštine kojoj biblioteka pripada

## predstavlja deo kompletne adrese biblioteke. Svaka adresa ima i broj zgrade i ime ulice u kojoj se

## biblioteka nalazi.

## Postoji evidencija o zaposlenim u bibliotekama, svaki zaposleni ima ime, prezime, poziciju i biblioteku u

## kojoj radi. Moguće pozicije su bibliotekar, čistač i direktor.

## Postoji evidencija o knjigama. Svaka knjiga ima autore, ime, godinu izdanja i ISBN. Svaki autor ima ime,

## prezime i opis. Jedna knjiga može da ima više autora. Takođe jedan naslov knjige može da ima više

## primeraka.

## Postoje korisnici biblioteke. Svaki korisnik ima korisničko ime i šifru. Jedna knjiga može da bude

## iznajmljena od strane jednog korisnika, ali svaki korisnik može da ima i do dve iznajmljene knjige. Potrebno

## je voditi evidenciju koji korisnik je iznajmio koju knjigu.Zahtevani upiti

1. Prikazati sva korisnička imena i knjige koje je svaki od njih iznajmio.

2. Prikazati ukupan broj biblioteka za svaku opštinu. Rezultate sortirati opadajući.

3. Za svaku knjigu prikazati ukupan broj autora ukoliko knjiga ima više od jednog autora.

4. Prikazati spisak svih biblioteka zajedno sa svim informacijama o korisnicima koji su iz nje iznajmili knjige.

5. Prikazati sve bibliotekare zajedno sa informacijama o bibliteci u kojoj rade.

6. Prikazati biblioteku koja ima maksimalan broj izdatih knjiga. Ukoliko ima više takvih biblioteka prikazati ih sve..

# Uvod

## Cilj

Cilj projektnog zadatka je da se razvije jedna aplikacija koja komunicira sa bazom podataka biblioteke I korisniku prikazuje podatke o knjigama, autorima, I izdatim knjigama, kao I korisnikova mogućnost da iznajmi knjige, kao i realizacija CRUD operacija nad svim tabelama baze podataka vezane za ovaj projekat.

## Realizacija

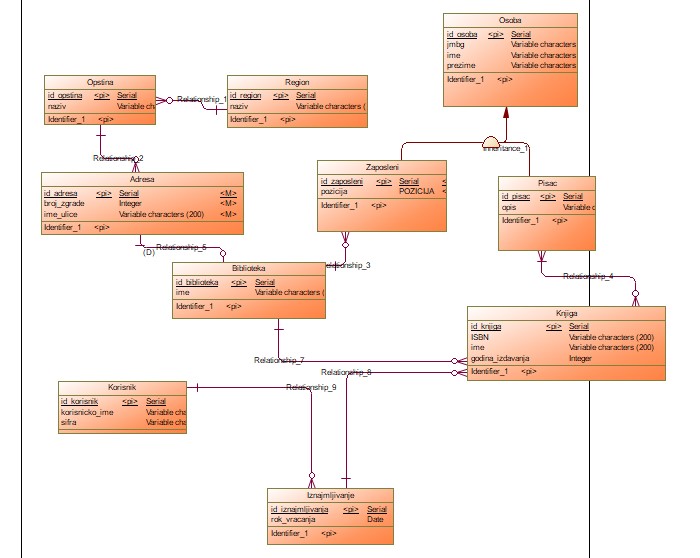
Baza podataka je MySQL tipa I sav rad sa bazom će biti realizovan preko phpmyadmin platforme. Dizajniranje baze će biti odrađeno preko **Powerdesigner**-a.

CRUD aplikacija će biti JavaFX aplikacija sa malim grafičko-korisničkim interfejsom. Aplikacija od korisnika zahteva jednu vrstu autentikacije pre pristupa samom sistemu za biblioteke, gde autentikacija takođe predstavlja komunikaciju sa bazom podataka. Bazu kreiramo na osnovu fizičkog modela baze podataka koju kasnije normalizujemo I dorađujemo po potrebama kako bi smanjili eventualne anomalije. Korisnik bira biblioteku I knjigu za iznajmljivanje a takođe postoji mogućnost brisanja, dodavanja I ažuriranja podataka iz baze.

# Projektno rešenje

Sledi opis projektnog rešenja od šeme baze do implementacije i primera korišćenja aplikacije.

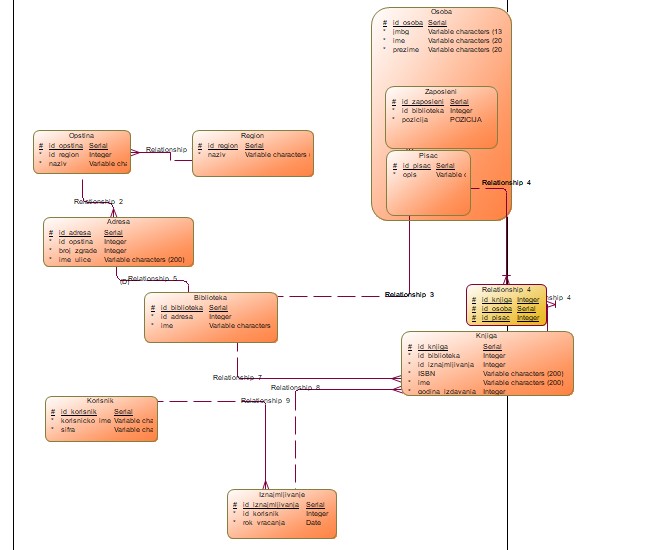
## Konceptualni model

Konceptualni model predstavlja prvi nivo apstrakcije problema dobijenog iz apstrakta.

Slika 1 Konceptualni model baze podataka

## Logički model

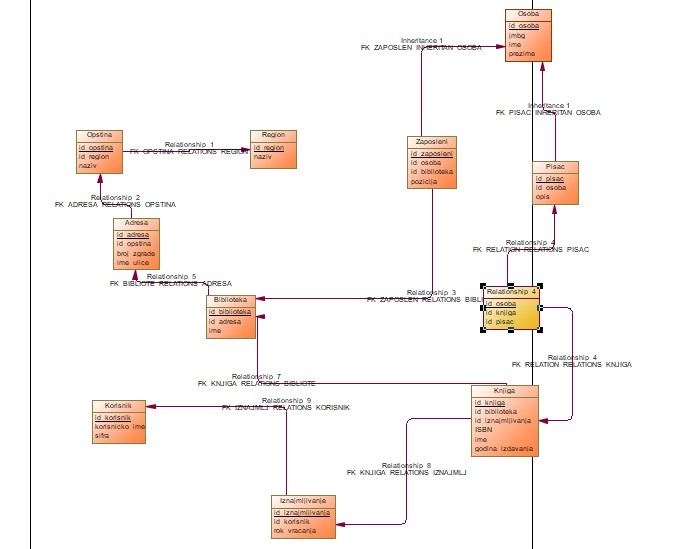
Na osnovu konceptualnog modela možemo da generišemo logički model koji je jedan korak bliži generisanju SQL DDL koda koji ćemo koristiti za kreairanje baze.



Slika 2 Logički model baze podataka

## Fizički model

Poslednji korak je tranformacija logičkog u fizički model kako bismo napravili što efikasniji kod za generisanje baze podataka.



Slika 3 Fizički model baze podataka

## Kreiranje baze I normalizacija

Na kraju iz fizičkog modela možemo da generišemo SQL kod koji ćemo da koristimo za kreiranje same baze

Iz ovako generisanog SQL fajla mozemo veoma lako kreirati bazu i sve odgovarajuće tabele.

Tokom generisanja fizičkog modela za samu bazu podataka I tokom generisanja koda za bazu generisana je tabela relationship\_4 koja ne obavlja nikakvu funkciju I samim tim je višak, tako da je obrisana, kao I tabela osoba koja je kreirana da bi bila nasleđena, međutim narušila je logiku baze I stvorila nepotrebne podatke I nekonzistentost, samim tim polja vezana za tu tabelu više se ne pojavljuju u drugim tabelama. U tabelu iznajmljivanje dodata je kolona ID\_BIBLIOTEKA I ID\_KNJIGA koje korespondiraju tabelama biblioteka I knjiga I njihovim primarnim ključevima naglasivči povezanost tabele iznajmljivanje sa ove pomenute dve. U tabelu iznajmljivanje dodata je I kolona BROJ\_IZN\_KNJIGA koji pamti int vrednost koliko je primeraka knjige iznajmljeno u jednom iznajmljivanju. U tabelu korisnik dodata je kolona ID\_BIBLIOTEKA koja predstavlja povezanost između tabele korisnik I tabele biblioteka u smislu koje biblioteke je taj korisnik član. U tabli knjiga dodata je jedna kolona ID\_PISAC koja identifikuje pisca koji je napisao tu knjigu, odnosno povezanost između tabela pisac I knjiga. U tabelu pisac dodate su dve kolone – IME, odnosno ime pisca kao i kolona BROJ\_PISACA koja indikuje koliko je knjiga imala pisaca, jer se dešava da jedna knjiga ima više pisaca. U tabelu biblioteka dodate su kolone ID\_REGION, ID\_OPSTINA, BROJ\_ZGRADE, KAPACITET i kolona BROJ\_IZDANJA gde se ova tabela povezuje sa tabelama region, opstina I adresa. I na kraju, u tabelu zaposleni dodata je kolona IME kako nemamo više tabelu osoba potrebno je da IME bude u okviru ove tabele. Upravo ovakvom podelom I dodavanjem određenih kolona ostvarena je Boyce-Coddova normalna forma. Normalizacija u okviru ove baze podataka morala je biti izvršena kako bi se izbegle anomalije i redudantnosti podataka kao i njihovo nepotrebno dupliranje. Takođe, dodati su odgovarajući strani ključevi koji povezuju tabele u okviru pomenutih polja, a update i delete operacije su posatvljene na cascade kako ne bi imali problema da brišemo, ali imamo na umu da brisanje povezanih stavki povlači sa sobom kaskadno brisanje, tako da može da se obriše i više podatak iz više tabela kako ne bi došlo do redudantnosti ili null vrednosti, a isto tako crud operacije nisu moguće ukoliko nema korespondirajućih tabela.

## Generisanje Java koda

Kod je kreiran na osnovu gledanja u konceptualni model baze podataka vezane za projekat imajuću u vidu izvršene normalizacije nad bazom podataka

Najpre izdvajamo entite, pa zatim klase za upravljanje bazom podataka, GUI klase za predstavljanje podataka i povezujemo ih u jednu celinu.

Takođe, za svrhu realizacije iznajmljivanja knjige u aplikaciji, primenjujemo Builder pattern, odnosno šablon projektovanja koji nam olakšava proces pravljenja GUI-a za iznajmljivanje.

JavaFX aplikacija je podeljenja u pakete koji predstavljaju određene segmente aplikacije, kao na primer izuzeci, entiteti, baza i tako dalje.

# Rešenja zahtevanih upita

## Upit *1*

*Prikazati sva korisnička imena i knjige koje je svaki od njih iznajmio.*

select korisnik.KORISNICKO\_IME, iznajmljivanje.ID\_KNJIGA from korisnik right join iznajmljivanje on iznajmljivanje.ID\_KORISNIK = korisnik.ID\_KORISNIK group by korisnik.ID\_KORISNIK;

## Upit *2*

*Prikazati ukupan broj biblioteka za svaku opštinu. Rezultate sortirati opadajući.*

*select count(id\_biblioteka) as broj\_biblioteka, opstina.ID\_OPSTINA, opstina.ID\_REGION, opstina.NAZIV from biblioteka right join opstina on biblioteka.ID\_OPSTINA = opstina.ID\_OPSTINA group by opstina.ID\_OPSTINA order by broj\_biblioteka desc;*

## Upit *3*

Za svaku knjigu prikazati ukupan broj autora ukoliko knjiga ima više od jednog autora.

select count(pisac.ID\_PISAC) as broj\_pisaca,pisac.ID\_PISAC, knjiga.IME, knjiga.ID\_KNJIGA from pisac right join knjiga on knjiga.ID\_PISAC = pisac.ID\_PISAC where pisac.BROJ\_PISACA > 1 group by knjiga.ID\_KNJIGA;

## Upit *4*

*Prikazati spisak svih biblioteka zajedno sa svim informacijama o korisnicima koji su iz nje iznajmili knjige*

select biblioteka.IME as spisak\_biblioteka, korisnik.ID\_KORISNIK, korisnik.KORISNICKO\_IME, korisnik.SIFRA, korisnik.ID\_BIBLIOTEKA from biblioteka right join korisnik on korisnik.ID\_BIBLIOTEKA = biblioteka.ID\_BIBLIOTEKA group by korisnik.ID\_KORISNIK;

## Upit *5*

*Prikazati sve bibliotekare zajedno sa informacijama o bibliteci u kojoj rade.*

*select zaposleni.ID\_ZAPOSLENI, zaposleni.ID\_BIBLIOTEKA, zaposleni.POZICIJA, zaposleni.IME as ime\_zaposleni,biblioteka.ID\_BIBLIOTEKA, biblioteka.ID\_ADRESA as adresa\_biblioteka,biblioteka.IME as naziv\_biblioteke,biblioteka.ID\_OPSTINA as biblioteka\_opstina from zaposleni join biblioteka on biblioteka.ID\_BIBLIOTEKA = zaposleni.ID\_BIBLIOTEKA where POZICIJA = 'bibliotekar';*

## Upit *6*

*Prikazati biblioteku koja ima maksimalan broj izdatih knjiga. Ukoliko ima više takvih biblioteka*

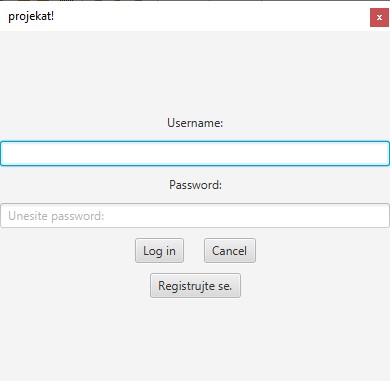
*prikazati ih sve.*

select biblioteka.ID\_BIBLIOTEKA, biblioteka.ID\_ADRESA, biblioteka.IME, biblioteka.ID\_OPSTINA,biblioteka.KAPACITET, biblioteka.BROJ\_IZDANJA from biblioteka join iznajmljivanje on iznajmljivanje.ID\_BIBLIOTEKA = biblioteka.ID\_BIBLIOTEKA where iznajmljivanje.BROJ\_IZN\_KNJIGA < biblioteka.KAPACITET;

*Ili ako se misli na najveci broj izdatih knjiga a ne iznamljenih, onda bi ovako izgledao upit:*

*select biblioteka.ID\_BIBLIOTEKA, biblioteka.ID\_ADRESA, biblioteka.IME, biblioteka.ID\_OPSTINA, biblioteka.KAPACITET, biblioteka.BROJ\_IZDANJA from biblioteka where biblioteka.BROJ\_IZDANJA = (select max(biblioteka.BROJ\_IZDANJA) FROM biblioteka);*

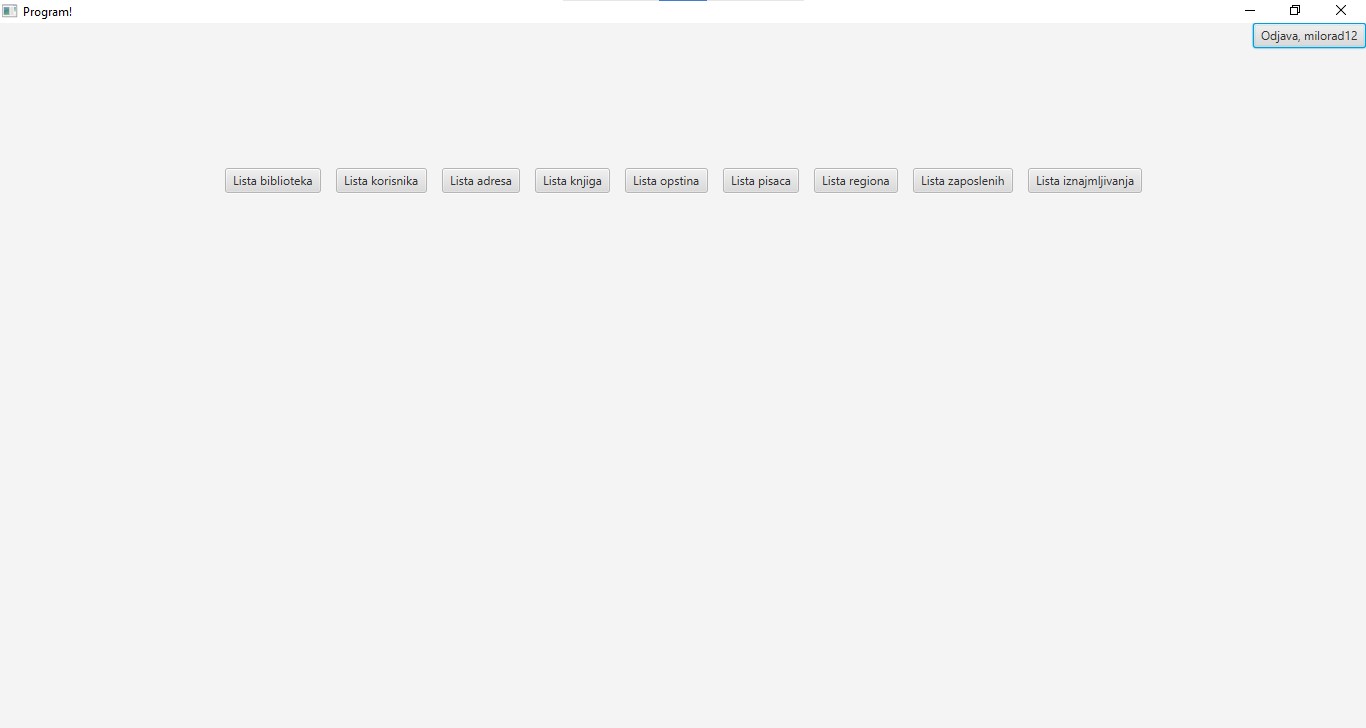
# Aplikacija

U ovom delu cemo napravili osnovni pregled izgelda i funkcionalnosti CRUD aplikacije. Aplikaciju pokrecemo na Run iz menija razvojnog okruženja, u ovom slučaju to će biti NetBeans IDE.

## Početak(Login i registracija)

U ovom delu, odnosno, početnom delu aplikacije(slika iznad), imamo pop-up window koji od korisnika zahteva da se uloguje sa svojim postojećim podacima, a ukoliko korisnik nema nalog, ima mogućnost registracije istog. Ovaj deo aplikacije komunicira sa bazom u smislu provere podataka i njihove ispravnosti kao i dodavanja novog korisnika prilikom registracije.

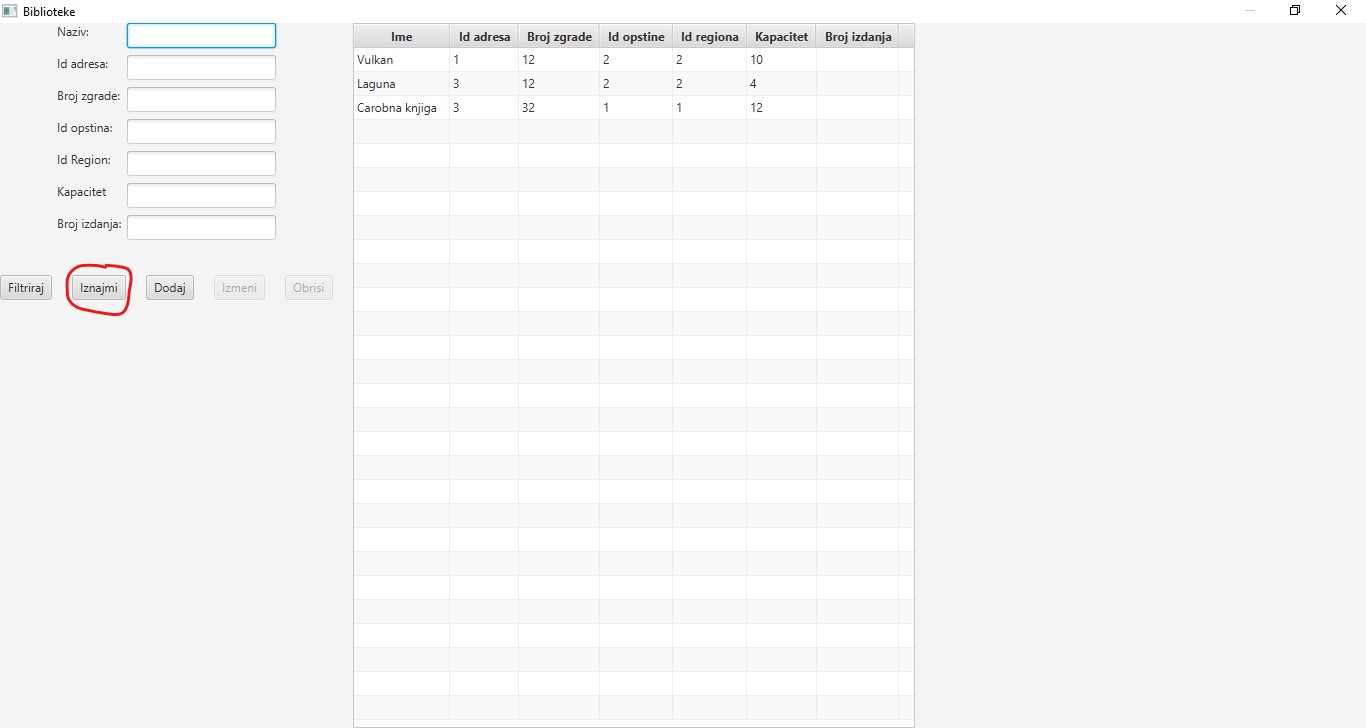
## Glavni CRUD Meni

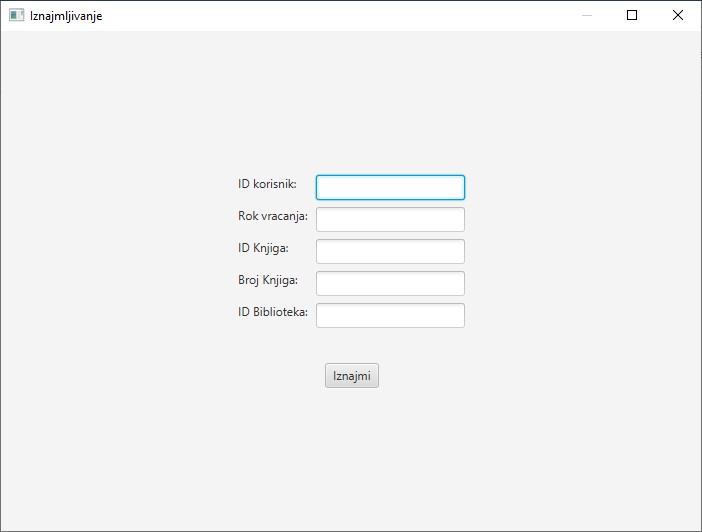


Nakon uspešne registracije/logina, korisnik dobija pristup glavnom CRUD meniju aplikacije, koji ima liste, odnosno dugmad za upravljanje(izmena,dodavanje,brisanje) iz svih tabela baze podataka kao i prikaz tabele iz istih. U gornjem desnom uglu, korisnik može i da se odjavi i izađe iz aplikacije u bilo kom trenutku.

## Lista iz tabele i iznajmljivanje

Na slici je prikazan primer liste biblioteka iz menija, ne stavljam screenshotove ostalih, jer se ponavljaju i imaju iste funkcije sem jedne, zato i ostavljam slike liste biblioteke. Iz ove liste, sem CRUD operacija moguće je i iznajmljivanje knjige nakon što korisnik obeleži u tabeli željenu biblioteku zatim odabere opciju Iznajmljivanja i popuni formu za iznajmljivanje, ovim procesom je ujedno i odrađena operacija dodavanja nad tabelom iznajmljivanje tako da u njenom zasebnom delu sa listom ostavljamo samo opcije izmeni i obriši.





Takođe, izmena I dodavanje podataka su za sve tabele dati u okviru njihovih individaulnih menija sa listama, I nalaze se kao male forme pored tabele, korisnik najpre obeleži tabelu koju menja, zatim popuni formu I pritisne dugme izmeni, a što se tiče dodavanja korisnik takođe popunjava formu I klikće na dugme Dodaj. Korisnik će biti upitan da potvrdi sve ove promene gde on pritiska yes ili no I time realizuje promene. Još jedna funkcionalnost koja je karakteristična samo za listu biblioteka je mogućnost filtriranja, koje služi za efikasniju pretragu tabele, odnosno podataka, popunjavanjem forme koje vrednosti onda tretira kao parametre pretrage.

# Zaključak

Projekat realizuje CRUD operacije nad bazom podataka.Poseban naglasak je na automatskom generisanju koda kao i parsiranju DDL naredbi u cilju unapređenja znanja iz ovog predmeta kao i znanja o samim bazama podataka . Baza je osmišljena sa ciljem da se što više smanji redudantnost i anomalije pri brisanju, ažuriranju i dodavanju. Ostvareno je napredno upravljanje i organizacija baze s tim što aplikacija sama po sebi ostaje malo trivijalna, stavljajući operacije sa bazom podataka u prvi plan svojih mnogobrojnih funkcionalnosti. Nadam se da ću u budućnosti moći da pravim još složenije aplikacije i sisteme.

# Literatura

1. C. J. Date, An introduction to Database Systems, Addison-Wesley Publishing Company, 1990
2. <https://www.w3schools.com/SQl/sql_join.asp>
3. David M. Kroenke, Database Processing – fundamentals, design and implementation, Prentice Hall, 2004.