Izrada aplikacije župne knjižnice

Cindrić, Marko

Master's thesis / Diplomski rad

2023

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:804404

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-07-14**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Sveučilišni studij

Izrada aplikacije župne knjižnice

Diplomski rad

Marko Cindrić

Osijek, 2023

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

Obrazac D1: Obrazac za imenovanje Povjerenstva za diplomski ispit

Osijek, 21.09.2023.

Odboru za završne i diplomske ispite

Imenovanje Povjerenstva za diplomski ispit

Ime i prezime Pristupnika:	Marko Cindrić
Studij, smjer:	Diplomski sveučilišni studij Računarstvo
Mat. br. Pristupnika, godina upisa:	D-1191R, 08.10.2021.
OIB studenta:	67112801420
Mentor:	izv. prof. dr. sc. Alfonzo Baumgartner
Sumentor:	,
Sumentor iz tvrtke:	
Predsjednik Povjerenstva:	doc. dr. sc. Tomislav Galba
Član Povjerenstva 1:	izv. prof. dr. sc. Alfonzo Baumgartner
Član Povjerenstva 2:	izv. prof. dr. sc. Tomislav Keser
Naslov diplomskog rada:	Izrada aplikacije župne knjižnice
Znanstvena grana diplomskog rada:	**Programsko inženjerstvo (zn. polje računarstvo)**
Zadatak diplomskog rada:	[Rezervirano: Marko Cindrić] Izraditi aplikaciju za vođenje evidencije o knjigama za potrebe župne knjižnice. Omogućiti pregled, dodavanje i brisanje knjiga. Također omogućiti evidenciju korisnika, te unos stavki posuđivanja i vraćanja knjiga. Izraditi odgovarajuće ispise posuđenih knjiga, popis svih knjiga po autorima, naslovima i vrstama.
Prijedlog ocjene pismenog dijela ispita (diplomskog rada):	Izvrstan (5)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 2 bod/boda Jasnoća pismenog izražavanja: 2 bod/boda Razina samostalnosti: 3 razina
Datum prijedloga ocjene od strane mentora:	21.09.2023.

Potvrda mentora o predaji konačne verzije rada:	*Mentor elektronički potpisao predaju konačne verzije.*
	Datum:

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

IZJAVA O ORIGINALNOSTI RADA

Osijek, 05.10.2023.

Ime i prezime studenta:	Marko Cindrić
Studij:	Diplomski sveučilišni studij Računarstvo
Mat. br. studenta, godina upisa:	D-1191R, 08.10.2021.
Turnitin podudaranje [%]:	6

Ovom izjavom izjavljujem da je rad pod nazivom**: Izrada aplikacije župne knjižnice**

izrađen pod vodstvom mentora izv. prof. dr. sc. Alfonzo Baumgartner

i sumentora ,

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

Zahvale

Hvala mom mentoru izv.prof.dr.sc. Alfonzu Baumgartneru na pristupačnosti i svim savjetima koje mi je dao, ne samo kod izrade kako završnog, tako i diplomskog rada, nego i na svim predavanjima i konzultacijama.

Hvala mojim stričevima čiči Bijelom i čiči Toni na prijevozu i riječima podrške svih ovih godina.

Hvala baki Jeli i didi Vinku.

Hvala prijatelju Luki, čiji je smisao za humor uspješno razbijao nervozu pred ispit.

Hvala mom cimeru Mateju na podršci, iako smo bili cimeri samo kratko vrijeme, stekli smo prijateljstvo za cijeli život.

Hvala Davidu, mom prijatelju od malih nogu.

Posebno hvala baki Kati koja je često molila za prolazak ispita.

Hvala Leonu, osobi za koju mogu reći da je baš pravi prijatelj i velika podrška.

Hvala Slavici, osobi koja je od prijateljice postala sestra, na svim zgodama i nezgodama svih ovih godina.

Hvala mojoj seki Kristini, njenom mužu Domagoju, mami Ivanki i mom uzoru, tati Zvonku na velikoj žrtvi kako bi mi omogućili školovanje, stoga ovaj uspjeh nikako nije samo moj, već naš zajednički!

I na kraju, najveća hvala ide dragom Bogu, jer znam da bez Njegove velike pomoći sigurno ne bih završio fakultet.

Sadržaj

1. UVOD 1
1.1. Zadatak diplomskog rada 1
2. Pregled sličnih aplikacija 2
2.1. Solibry 2
2.2. Personal library 2
2.3. CodeAchi Library Management System 3
2.4. Private Books Library 4
2.5. Book Collector DW 5
3. Korištene tehnologije 7
3.1. Java Programski jezik 7
3.2. MySQL 8
4. IZRADA APLIKACIJE 9
4.1. Izrada baze podataka 9
4.2. DBConnection.java 10
4.3. HomePage.java 10
4.4. IssueBook.java 10
4.5. IssueBookDetails.java 11
4.6. ManageBooks.java 11
4.7. ManageUsers.java 11
4.8. ReturnBook.java 12
4.9. SearchBooks.java 12
4.10. ViewAllRecords.java 12
4.11. SignupPage.java 12
4.12. LoginPage.java 13
4.13. Izrada korisničkog sučelja aplikacije 13
5. Prikaz rada aplikacije 14
5.1. Registracija 14
5.2. Prijava 15
5.3. Početna stranica 17
5.4. Upravljanje knjigama 18
5.5. Upravljanje korisnicima 20
5.6. Posuđivanje knjiga korisnicima 22
5.7. Vraćanje knjige 25
5.8. Zapisi o posudbama knjiga 27

5.9. Posuđene knjige 28
5.10. Popis dužnika 28
5.11. Pretraživanje 29
5.12. Odjava 30
6. Zaključak 31
Literatura 32
Sažetak 33
Abstract 34
Životopis 35
Prilozi 36

1. UVOD

Tema ovog diplomskog rada je izraditi aplikaciju župne knjižnice. Prilikom razgovora sa župnikom se razvila ideja za razvoj aplikacije u kojoj bi župnik mogao vidjeti sve knjige koje se nalaze u župnoj knjižnici. Aplikacija je korisna jer omogućuje pretragu i uvid svih knjiga u knjižnici na lakši i brži način od klasičnog načina traženja knjiga u knjižnici. Stoga će se u drugom poglavlju napraviti pregled područja koje pokriva ovaj diplomski rad na način da će se opisati nekoliko besplatnih aplikacija za upravljanje knjižnicama koje se trenutno mogu pronaći na tržištu. U trećem poglavlju će se pisati o korištenim tehnologijama za izradu praktičnoga rada. Korištene tehnologije za izradu diplomskog rada su Java programski jezik i SQL. U četvrtom poglavlju će se opisivati izrada praktičnog dijela diplomskog rada. U petom poglavlju će se opisati izgled i rad same aplikacije.

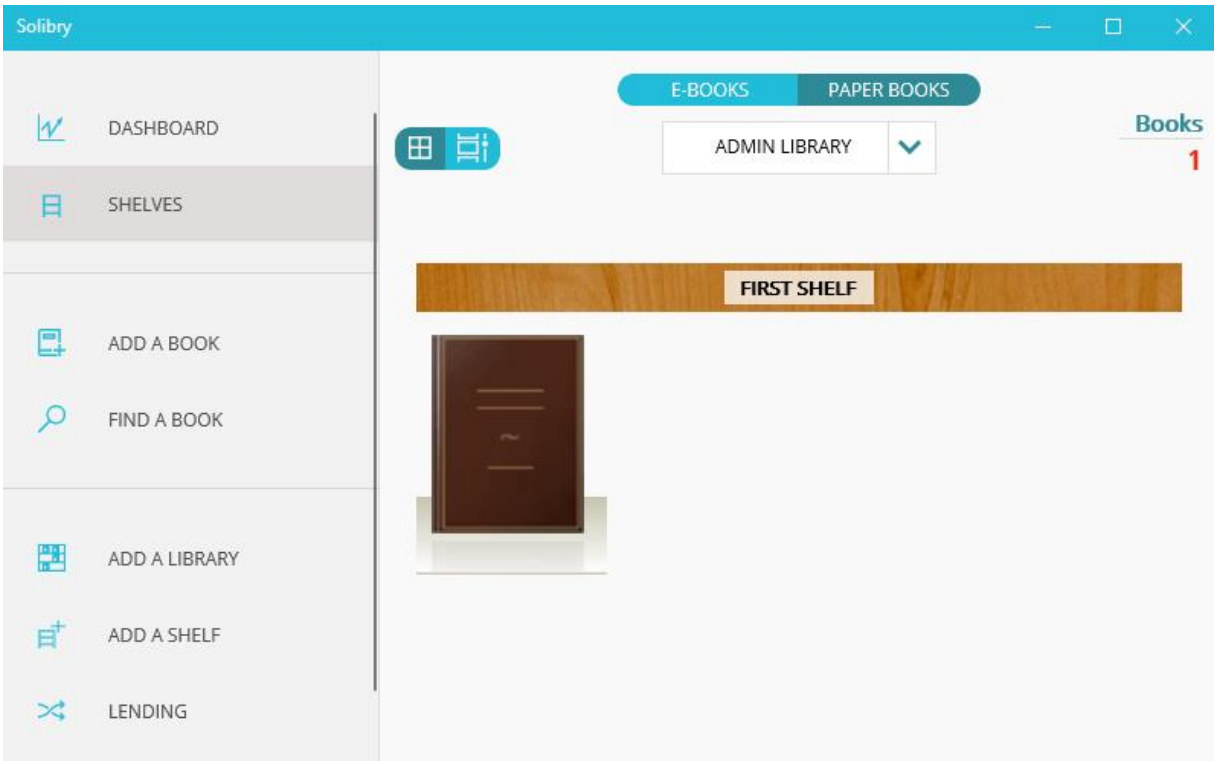
1.1. Zadatak diplomskog rada

Izraditi aplikaciju za vođenje evidencije o knjigama za potrebe župne knjižnice. Omogućiti pregled, dodavanje i brisanje knjiga. Također omogućiti evidenciju korisnika, te unos stavki posuđivanja i vraćanja knjiga. Izraditi odgovarajuće ispise posuđenih knjiga, popis svih knjiga po autorima, naslovima i vrstama.

2. Pregled sličnih aplikacija

2.1. Solibry

Solibry je besplatna aplikacija koja omogućava upravljanje knjižnicom. Može se preuzeti na *Microsoft Store* – u. Omogućava pretraživanje, dodavanje papirnatih knjiga i e-knjiga u bazu podataka. Aplikacija omogućava dodavanje i pretragu knjiga, dodavanje knjižnice posebno za knjige u papirnatom obliku i u digitalnom obliku. E-knjige se ne mogu posuđivati za razliku od knjiga u papirnatom obliku. Aplikacija omogućava i prikaz statističkih podataka vezanih za knjižnice, police i knjige, broj stranica i sekudne. Međutim, statistika nije precizna, jer se podatci ne osvježavaju u pravom vremenu, niti su ispravni.[1]

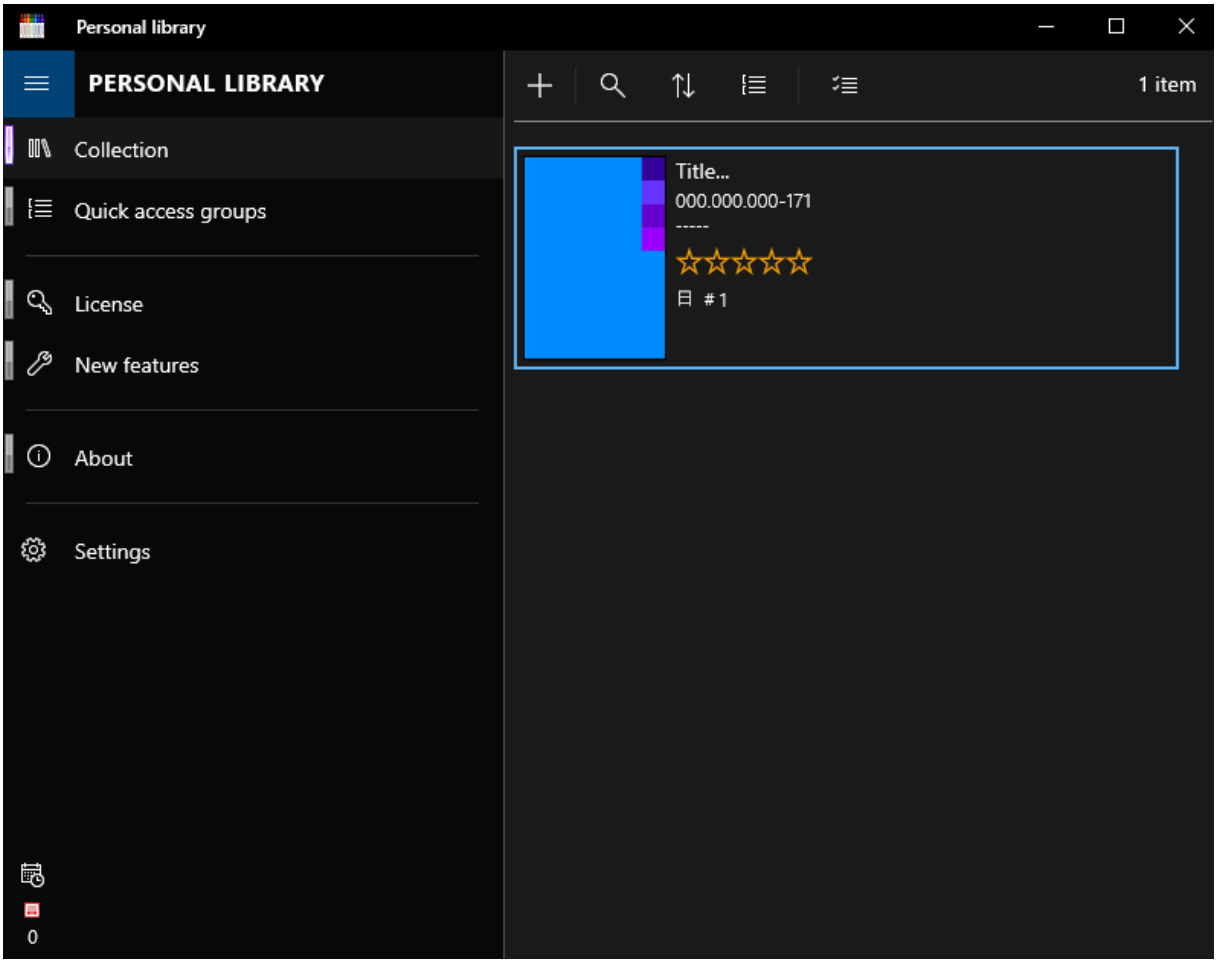


Slika 2.1. Izgled početnog zaslona Solibry aplikacije.

2.2. Personal library

Personal library je djelomično besplatna aplikacija koja omogućava upravljanje knjižnicom. Može se preuzeti na *Microsoft Store* – u. Aplikacija je predinstalirana na portugalski jezik, stoga je potrebno u postavkama promijeniti jezik na engleski. Omogućava pretraživanje, dodavanje, uređivanje i brisanje knjiga u papiranatom obliku i knjiga u digitalnom obliku. Aplikacija nema

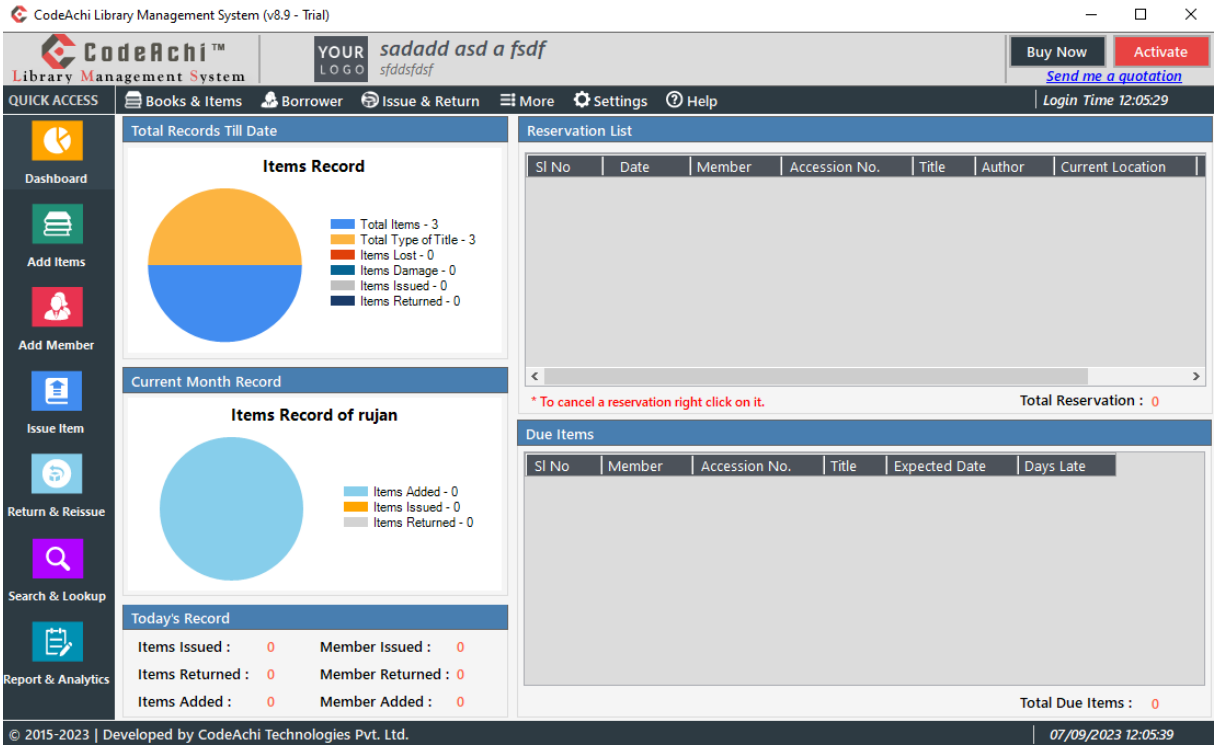
prikaz statističkih podataka vezanih za broj knjiga koje su zapisane u njenu bazu podataka, niti je aplikacija grafički za rad osim ako se ne radi o radu na cijelom zaslonu. Dodavanje knjiga je u bazu i njihovo uređivanje je vrlo jednostavno i intuitivno.[2]



Slika 2.2. Izgled početnog zaslona *Personal library* aplikacije.

2.3. CodeAchi Library Management System

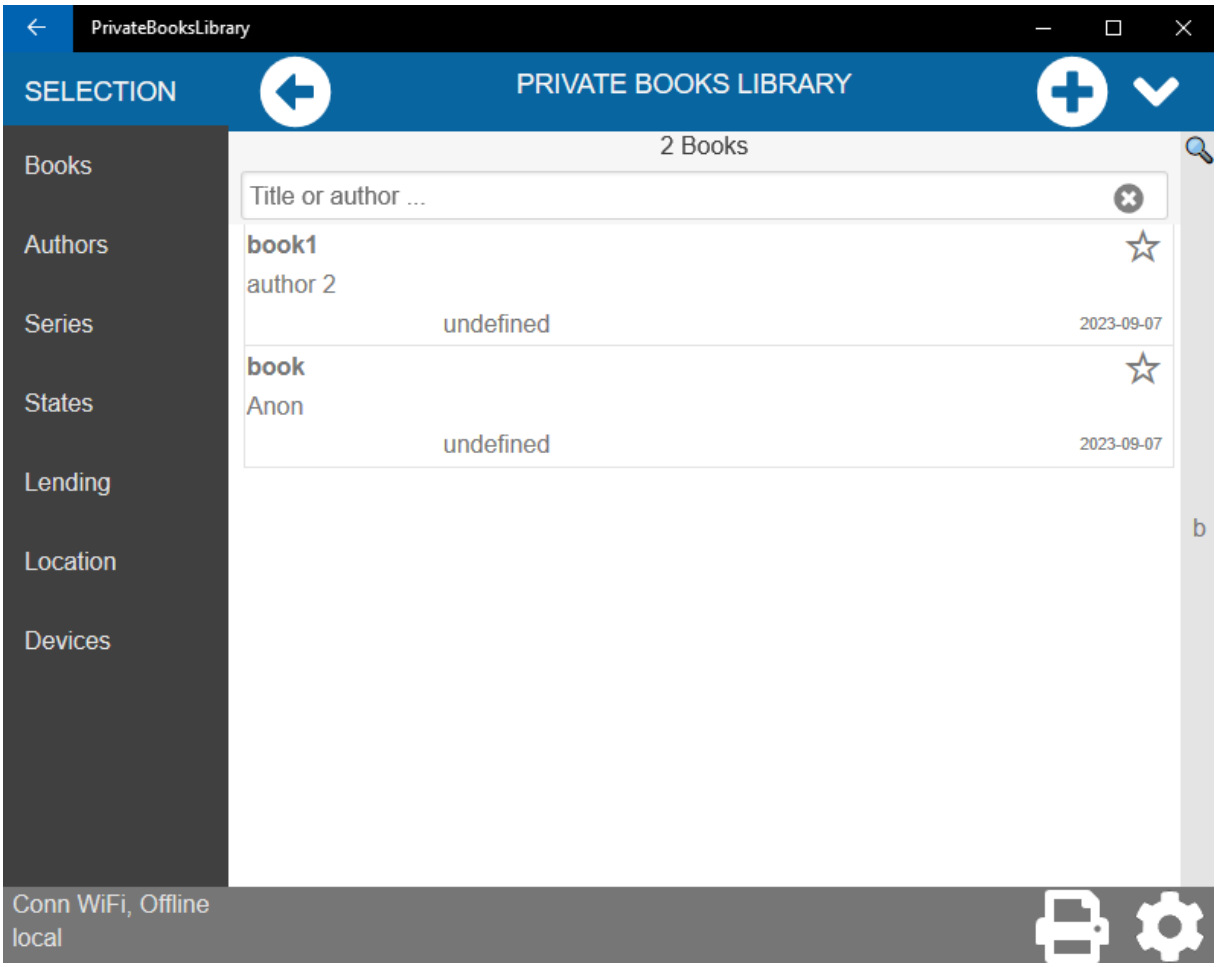
CodeAchiLibrary Management System je besplatna aplikacija za upravljanje radom knjižnice. Preuzima se besplatno sa *web* stranice proizvođača. Prilikom pokretanja aplikacije, potrebno je napraviti registraciju korisnika. Aplikacija omogućava dodavanje knjiga, CD –a i DVD –a Predviđeni korisnici ove aplikcije su učenici i nastavnici. Stoga se kod dodavanja korisnika mora odabrati funkcija korisnika. Knjige se mogu posuđivati i pretraživati. Administratoru je dostupan pregled statistike i analitike velikog broja stavki kao što su na primjer: najčešće i najrjeđe posuđene knjige, posuđene knjige, knjige kojima je istekao rok za povratak u knjižnicu i mnoge druge.[3]



Slika 2.3. Izgled početnog zaslona *CodeAchi Library Management System* aplikacije.

2.4. Private Books Library

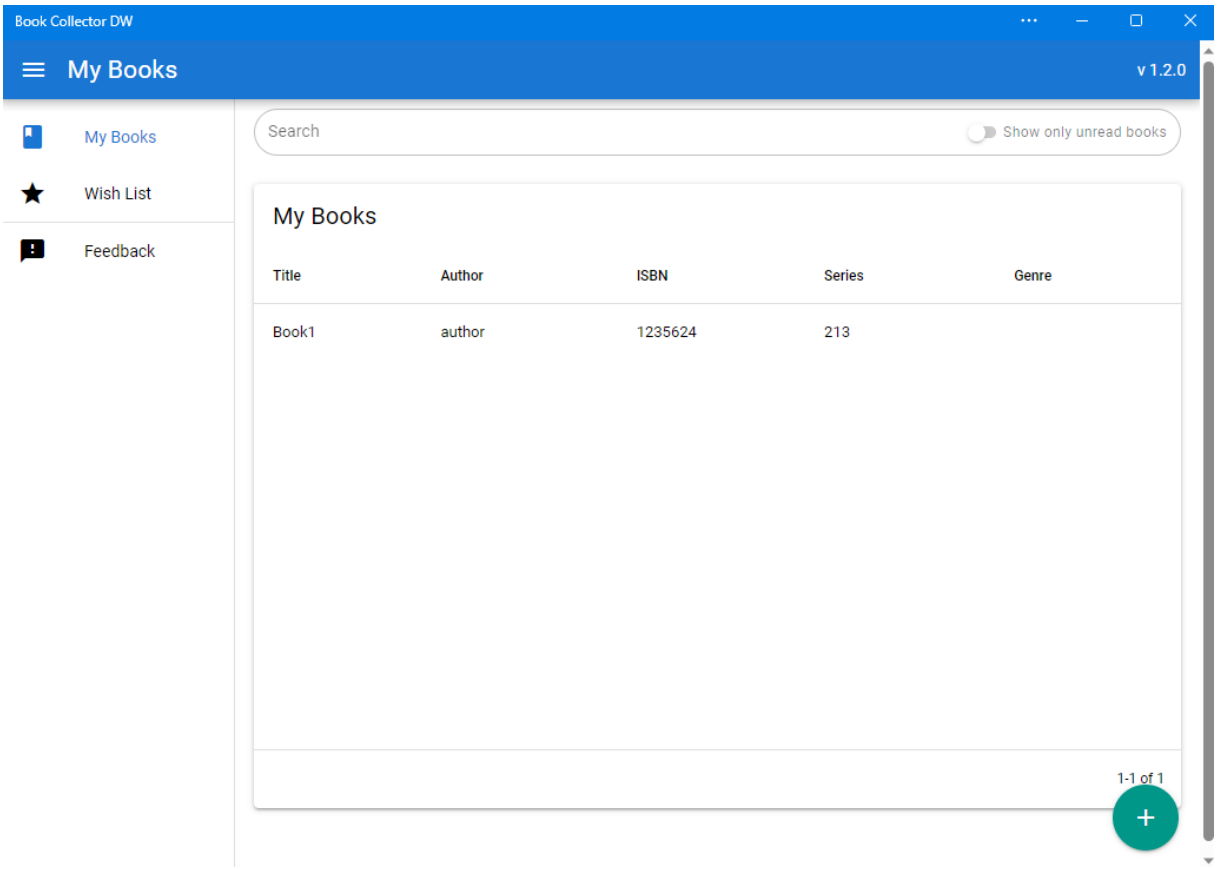
Private Books library je besplatna aplikacija za upravljanje radom knjižnice. Može se preuzeti na *Microsoft Store* – u. Omogućuje pretraživanje, dodavanje, uređivanje i brisanje knjiga, autora. Moguće je i posuđivanje knjiga. Mana aplikacije je izostanak automatskog ažuriranja, na primjer, ako korisnik doda knjigu u bazu podataka, ispisati će mu se poruka da je knjiga dodana u bazu, no ona neće biti vidljiva u bazi. Da bi se knjiga prikazala u bazi, korisnik mora osvježiti aplikaciju.[4]



Slika 2.4. Izgled početnog zaslona *Private Books Library* aplikacije.

2.5. Book Collector DW

Book Collector DW je besplatna aplikacija za upravljanje radom knjižnice. Može se preuzeti na *Microsoft Store* – u. Omogućuje pretraživanje, dodavanje, uređivanje i brisanje knjiga iz knjižnice. Vrlo je jednostavna za korištenje, U bazu se mogu dodati knjige koje korisnik posjeduje i koje ne posjeduje, ali su mu na listi želja. Aplikacija ne vodi statistiku, niti je moguće posuđivanje knjiga.[5]



Slika 2.5. Izgled početnog zaslona *Book Collectior DW* aplikacije.

3. Korištene tehnologije

3.1. Java Programski jezik

Java je programski jezik čiji je razvoj započeo 1991. godine, a njena prva verzija je objavljena 1995. godine. Od trenutka objave, popularnost Java-e je stalno rasla, zbog toga što je brza, sigurna, jednostavna za naučiti, prenosiva i objektno orjentirana. Java se danas koristi za izradu širokog spektra aplikacija poput mobilnih aplikacija, video igrica, *Desktop* aplikacija i *WEB* aplikacija. Jedna od najvećih prednosti Java-e je njena prenosivost, ona omogućuje programeru da jednom napisani programski kod može pokrenuti na svim operacijskim sustavima neovisno o kojem se operacijskom sustavu radi, to je postignuto na način da se programski kod prilikom kompiliranja prevodi u *bytecode*. Zatim se *bytecode* pokreće na JVM (*Java Virtual Machine*). JVM nije građena isto za svaki operacijski sustav, već je prilagođena specifičnostima svakog od najpopularnijih operacijskih sustava kao što su Windows, macOs i Linux.[6]

U vrijeme pisanja ovog diplomskog rada prema podatcima TIOBE organizacije koja se bavi procjenjivanjem i kvalitetom softvera, Java programski jezik zauzima četvrto mjesto po popularnosti svih programskih jezika sa udjelom od 10.33%. odmah iza Python-a koji ima udjel od 13.33%, C programskog jezika koji ima udjel od 11.41% i C++ programskog jezika koji ima udjel od 10.63%. To čini Java programski jezik jednim od najboljih programskih jezika za početnike koji žele naučiti programiranje, jer imaju dostupnu veliku zajednicu koja ih može uputiti i pomoći u njihovom učenju programiranja.[7]

```
public void setUserDetailsToTable(){
    try {
        Connection connection = DBConnection.getConnection();
        Statement statement = connection.createStatement();
        ResultSet resultSet = statement.executeQuery("select * from user_details");

        while(resultSet.next()){
            String userId = resultSet.getString("user_id");
            String userName = resultSet.getString("user_name");

            Object[] object = {userId, userName};
            model =(DefaultTableModel) tbl_userDetails.getModel();
            model.addRow(object);
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

Slika 3.1. Primjer Java programskog jezika

3.2. MySQL

MySQL je sustav za upravljanje bazom podataka otvorenog koda čiji je razvoj započeo 1994. godine, a njena prva verzija je objavljena 1995. godine. To je jedna od najpopularnijih baza podataka na svijetu. Korištena je u nekim od najpoznatijih i najznačajnijih aplikacija poput Facebook-a, Netflix-a, Uber-a, Booking-a i drugih. MySQL je relacijska baza podataka što znači da se podatci mogu pohraniti u više manjih ali smislenih baza podataka umjesto da sve svi podatci pohrane u jednu veliku bazu podataka što automatski otežava njeno korištenje i smanjuje njezinu preglednost kod baza podataka sa velikom količinom podataka. Programer sam putem odnosa odlučuje kako će povezati baze podataka. Odnosi koji se mogu koristiti za stvaranje relacija su: jedan prema jedan, jedan prema više, više prema jedan, više prema više, i jedinstveno. MySQL ima vrlo jednostavnu sintaksu, što omogućuje brzo učenje i jednostavno korištenje. Prednosti MySQL-a su osim njene jednostavnosti, pouzdanost, skalabilnost performanse i sigurnost.[8]

U vrijeme pisanja ovog diplomskog rada, prema podatcima DB-Engines inicijative koja se bavi prikupljanjem i prezentiranjem podataka o sustavima za upravljanje bazama podataka, MySQL je trenutno drugi najpopularniji sustav za upravljanje bazom podataka, odmah iznad Oracle-ovog sustava, a ispred Microsoft-ovog SQL Server-a.[9]

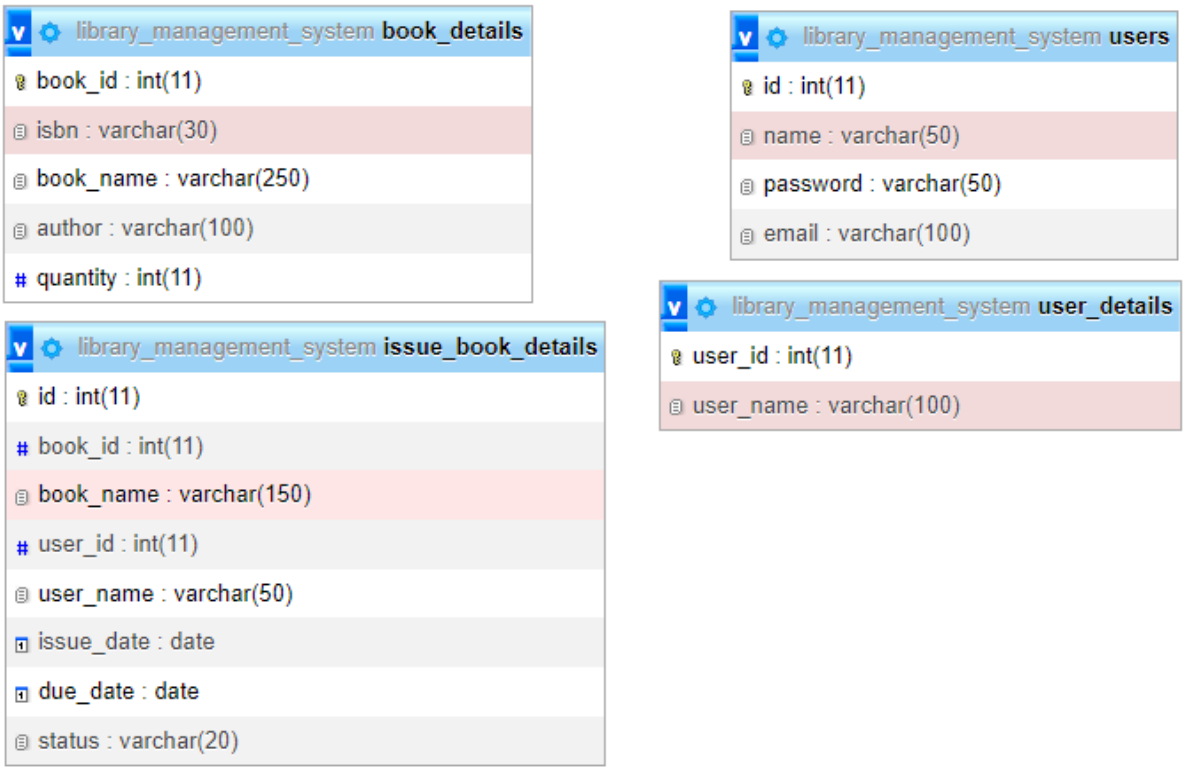
```
1 CREATE TABLE book_details(book_id INT PRIMARY KEY AUTO_INCREMENT NOT
  NULL, book_name VARCHAR(250), author VARCHAR(100), QUANTITY INT);
```

Slika 3.2. Primjer SQL programskog jezika

4. IZRADA APLIKACIJE

4.1. Izrada baze podataka

Baza podataka za aplikaciju je napravljena u phpMyAdmin-u korištenjem SQL-a uz pomoć naredbe *CREATE TABLE*. Za potrebe aplikacije napravljene su četiri tablice: *book_details, users, issue_book_details i user_details*. Tablica *book_details* sadrži stupce: id knjige, isbn, ime knjige, ime autora i količinu. Ona se koristi za upisivanje, uređivanje i brisanje knjiga u bazi podataka. Tablica *users*: sadrži stupce: id, ime, lozinku i email. Ona se koristi za upis podataka administratora aplikacije. Tablica *issue_book_details* sadrži stupce: id , id knjige, ime knjige, id korisnika, datuma posuđivanja, datuma vraćanja knjige i statusa. U ovu tablicu se zapisuju svi podataci koji su potrebni za posuđivanje knjige korisnicima župne knjižnice. U stupac status se može upisati tekst: „Posuđena“ ili „Vraćena“ u ovisnosti o tome je li knjiga posuđena korisniku i je li korisnik vratio posuđenu knjigu. Tablica *user_details* sadrži stupce id korisnika i ime korisnika, u tablicu se mogu zapisivati, uređivati i brisati podatci o korisnicima župne knjižnice.



Slika 4.1. ERD dijagram baze podataka u phpMyAdmin-u.

4.2. DBConnection.java

DBConnection je klasa u kojoj se nalazi metoda koja je odgovorna za spajanje na localhost bazu podataka. Metoda je smještena u posebnu klasu kako bi se izbjeglo ponovno pisanje istog koda koji se treba koristiti za spajanje na bazu podataka u klasama i metodama kojima je potreban pristup bazi podataka.

4.3. HomePage.java

HomePage je klasa u kojoj se nalaze metode: *setUserDetailsToTable, setBookDetailsToTable, setDataToCards*. Metoda *setUserDetailsToTable* dohvaća podatke o svim korisnicima župne knjižnice iz baze podataka i zapisuje te podatke u tablicu na početnoj stranici aplikacije. Stupci i redci iz baze podataka se dohvaćaju SQL upitom *SELECT* na tablicu *user_details.* Korištenjem while petlje se osigurava da se dohvate podatci iz svih stupaca i redaka, zatim se ti podatci spremaju u objekt koji se poziva kod stvaranja tablice. Metoda *setBookDetailsToTable* radi na identičan način kao i metoda *setUserDetailsToTable* s tim da je jedina razlika to što ona dohvaća podatke o svim knjigama zapisanim u bazu podataka župne knjižnice. Ti podatci su dohvaćeni SQL upitom *SELECT* na tablicu *book_details.* Posljednja metoda u klasi je *setDataToCards.* Ta se metoda koristi za dohvaćanje broja knjiga, korisnika, posuđenih knjiga i dužnika iz baze podataka i njihovo prikazivanje u karticama na početnoj stranici aplikacije. Broj knjiga i korisnika je dohvaćen korištenjem SQL upita *SELECT* na način da je dohvaćen zadnji redak u tablicama *book_details* i *userDetails* i taj podatak je postavljen u „karticu“ na početnoj stranici. Broj posuđenih knjiga je dohvaćen na isti način kao i prethodni, jedina razlika je što je uz upit *SELECT* dodan uvjet u kojem status knjige mora biti „Posuđena“. Broj dužnika je dohvaćen na identičan način kao i broj posuđenih knjiga s razlikom što se u ovom slučaju uz status koji treba biti „Posuđena“, uspoređuje datum s kojim je korisnik trebao vratiti knjigu i današnji datum. Ukoliko je rok za vraćanje knjige bio prije današnjeg datuma, tada se taj redak u tablici smatra „dužničkim i pridodaje se ukupnom broju dužnika.

4.4. IssueBook.java

IssueBook je klasa u kojoj se nalaze metode: *setBookDetailsToTable, setUserDetailsToTable, getBookDetails, getUserDetails, issueBook, updateBookCount* i *isAlreadyIssued.* Prve dvije metode su već opisane u prethodnom potpoglavlju. Metoda *getBookDetails* uzima parametar koji korisnik unese u polje teksta i koristi ga za dohvaćanje knjige iz baze podataka. Ako tražena

knjiga postoji, u labele će se ispisati podatci o traženoj knjizi, u suprotnom će se ispisati poruka kako tražena knjiga ne postoji. Metoda *getUserDetails* radi na identičan način kao i metoda *getBookDetails*, jedina razlika je što metoda dohvaća podatke korisnika knjižnice. Metoda *issueBook* je zadužena za posuđivanje knjiga, ona dohvaća podatke koji su prethodno upisani u labele, i upisuje ih u tablicu *issue_book_details* pomoću SQL naredbe *INSERT*. Metoda *updateBookCount* uz pomoć SQL naredbe *UPDATE* smanjuje broj raspoloživih knjiga u bazi podataka za posuđivanje za 1 nakon što je knjiga posuđena korisniku. Metoda isAlreadyIssued provjerava je li određena knjiga već posuđena određenom korisniku pomoću SQL naredbe *SELECT*.

4.5. IssueBookDetails.java

Klasa *IssueBookDetails* sadrži metodu *setIssueBookDetailsToTable*. Metoda uz pomoć SQL naredbe *SELECT* dohvaća sve stupce iz tablice issue_book_details koje imaju status „Posuđena“ . Zatim se ti podatci spremaju u objekt koji se poziva kod stvaranja tablice.

4.6. ManageBooks.java

Klasa *ManageBooks* sadrži metode: *addBook, updateBook, deleteBook, clearTable* i *setBookDetailsToTable*. Potonja metoda je već opisana. Metoda *addBook* upisuje podatke u bazu podataka uz pomoć SQL naredbe *INSERT INTO*. Metoda *updateBook* se koristi kod uređivanja već upisanih knjiga u bazu podataka, nakon unosa željenih podataka u tekstualno polje, oni se u bazi ažuriraju uz pomoć SQL naredbe *UPDATE*. Metoda *deleteBook* se koristi za brisanje knjige iz baze podataka. Uz pomoć SQL naredbe *DELETE* briše knjigu čiji je ID broj korisnik upisao u tekstualno polje za ID knjige. Metoda *clearTable* se koristi za brisanje svih podataka iz tablice prije nego se podatci ažuriraju kako bi se izbjeglo višestruko upisivanje istih podataka u tablicu.

4.7. ManageUsers.java

Klasa *ManageUsers* sadrži metode *setUserDetailsToTable, addUser, clearTable, updateUser* i *deleteUser*. Navedene metode imaju iste funkcije kao i funkcije opisane u prethodnom potpoglavlju. Jedina razlika je što su prilagođene za rad s podatcima vezanim uz korisnika.

4.8. ReturnBook.java

Klasa *ReturnBook* sadrži metode: *getIssuedBookDetails, returnBook* i *updateBookCount. getIssuedBookDetails* metoda dohvaća podatke iz zapisa o posuđenim knjigama pomoću SQL metode *SELECT,* uz uvjete koje je korisnik dužan upisati u tekstualno polje, to su ID knjige i ID korisnika. Dohvaćeni podatci se zatim ispisuju u labele. Metoda *returnBook* je zadužena za vraćanje knjige u bazu podataka, vraćanje knjige u bazu podataka se odvija uz pomoć SQL naredbe *UPDATE* na način da se pronađu ID knjige i korisnika, i na njihovom mjestu se status promijeni iz „Posuđena“ u „Vraćena. Metoda *updateBookCount* je već opisana u potpoglavlju 4.4. Jedina razlika je to što se sada povećava broj određene knjige za 1.

4.9. SearchBooks.java

Klasa SearchBooks sadrži metode: setBookDetailsToTable, clearTable i search. Prve dvije metode su već opisane u prethodnim potpoglavljima. Metoda search traži i filtrira podatke iz tablice koristeći string parametar koji je korisnik unio u tekstualno polje.

4.10. ViewAllRecords.java

Klasa *ViewAllRecords* sadrži metode: *setIssueBookDetailsToTable, clearTable* i *search.* Metode *setIssueBookDetailsToTable, clearTable* su opisane u prethodnim potpoglavljima. Metoda *search* vrši pretragu svih zapisa koji su unešeni u bazu od datuma koji korisnik odredi kao početni do datuma koji korisnik odredi kao konačni, uz pomoć SQL naredbe *SELECT*. Zatim se pronađeni zapisi spremaju u objekt koji se poziva kod stvaranja tablice.

4.11. SignupPage.java

Klasa *SignupPage* sadrži metode: *insertSignupDetails, validateSignup* i *checkDuplicateUsers.* Metoda *insertSignupDetails* uz pomoć SQL naredbe *INSERT INTO* unosi podatke u bazu podataka. Metoda *validateSignup* provjerava provjerava jesu li u polja za unos imena i lozinke unesni podatci. Kod polja za unos email adrese se provjerava je li unešeni tekst ispravnog formata. Metoda *checkDuplicateUsers* uz pomoć SQL naredbe *SELECT* provjerava postoji li u bazi podataka zapisan korisnik sa istom email adresom koja je trenutno unešena u polje za unos email adrese.

4.12. LoginPage.java

Klasa *LoginPage* sadrži metode: *validateLogin* i *login*. Metoda *validateLogin* radi na isti način kao i metoda *validateSignup* opisana u prethodnom potpoglavlju. Metoda *login* provjerava odgovaraju li podatci koje je administrator upisao u polje za unos teksta, podatcima upisanim u bazu podataka uz pomoć SQL naredbe *SELECT*. Ako unešeni podatci odgovaraju podatcima iz baze podataka, prijava je uspješna.

4.13. Izrada korisničkog sučelja aplikacije

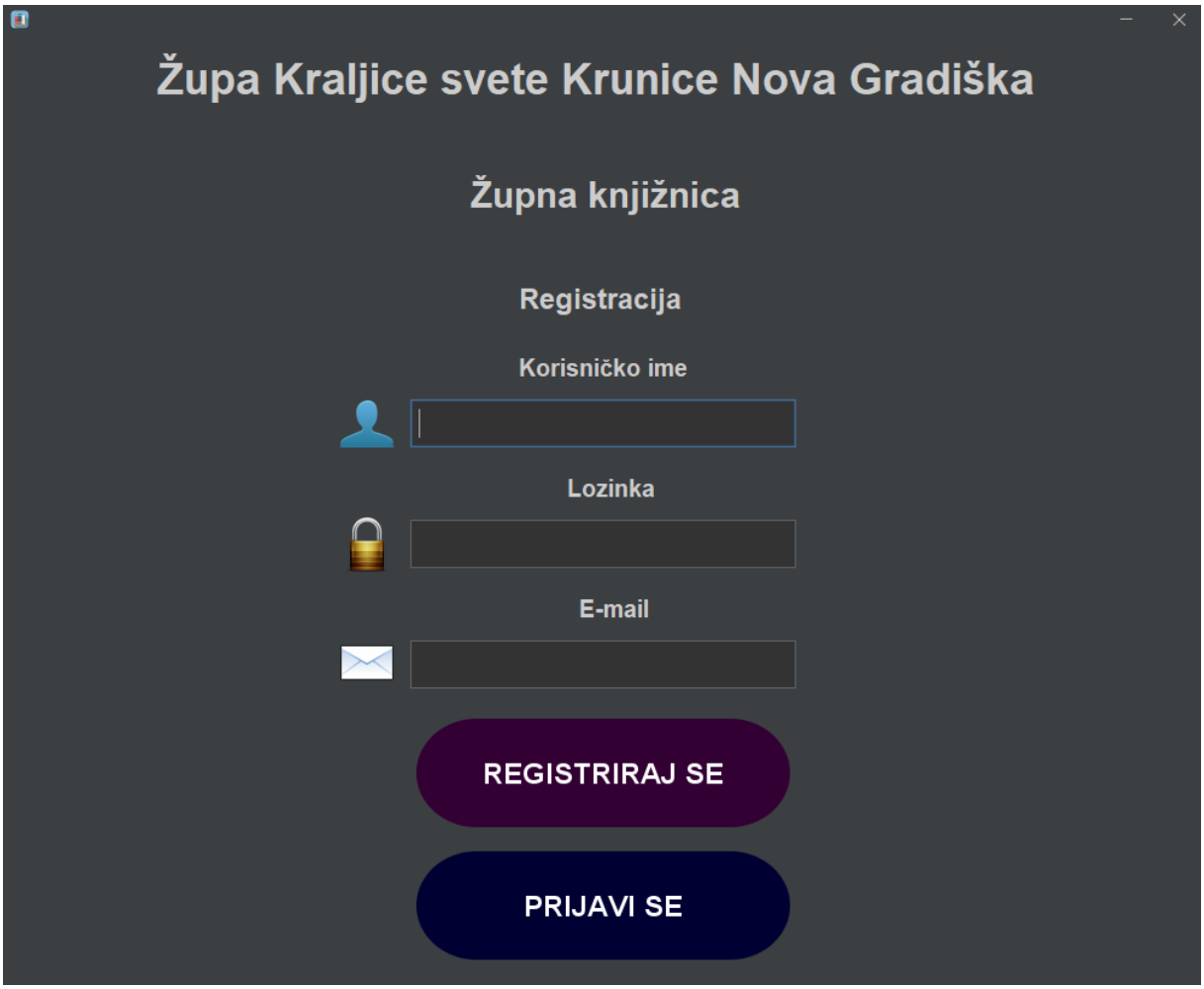
Za izradu korisničkog sučelja je korišten FlatDarkLaf. To je Java Swing *look and feel* koji omogućava moderniji izgled korisničkog sučelja. Odabran je zbog toga što pruža tamni način izgleda, koji osim što je lijep izgledom, dobar je i jer smanjuje naprezanje očiju korisnika. Za dizajn korisničkog sučelja su korišteni paneli, koji omogućavaju dobru organizaciju komponenata. Boje koje su korištene kod panela su bile sive ili tamnosive za potrebe tamnog načina rada. Za otvaranje drugih prozora u aplikaciji nisu korišteni klasični gumbi, već je omogućen klik na tekstualnu labelu koja izvodi operaciju otvaranja novog prozora. Klasični gumbi su korišteni za izvođenje određenih akcija poput prijave i registracije korisnika i slično. Boje gumba su bile tamne boje plave, ljubičaste i crvene zbog toga što bi svjetle verzije tih boja narušavale tamni način izgleda aplikacije. Za upisivanje tekstualnih podataka u aplikaciju su se koristila polja za unost teksta, dok je za odabir datuma korišten *JdateChooser widget* dostupan u NetBeans-u. Za prikaz podataka o knjigama i korisnicima su korištene tablice.

5. Prikaz rada aplikacije

U ovom poglavlju će se prikazati grafički izgled i rad aplikacije.

5.1. Registracija

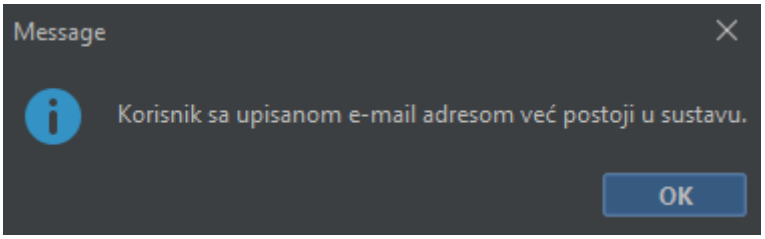
Prilikom pokretanja aplikacije, korisniku se prvo otvori prozor za registraciju.



Slika 5.1. Dizajn korisničkog sučelja za registraciju

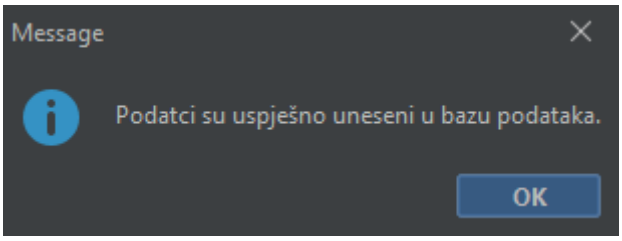
On se sastoji od dva polja za unos teksta i jednog polja za unos lozinke, gumba za registraciju i gumba za prijavu. Da bi korisnik dobio pristup knjižnici mora se prvo registrirati, to čini unosom korisničkog imena, lozinke i e-mail adrese. Ukoliko korisnik pokuša kliknuti na gumb za registraciju prije nego li je unio podatke ispravnog formata u sva tri polja za unos teksta, neće se moći uspješno registrirati, te će mu program ispisati poruku kako nisu unešeni ispravni podatci. Nakon što korisnik unese ispravne podatke i klikne na gumb za registraciju, prvo se izvršava spajanje na bazu podataka, nakon toga se izvršava provjera postoji li u bazi podataka

registrirani korisnik sa istim podatcima kao i podatci koji su upravo unešeni, ukoliko postoji, korisnik se ne može registrirati, već mora unijeti druge podatke.



Slika 5.2. Ispis poruke kod pokušaja registracije sa već registriranom e-mail adresom

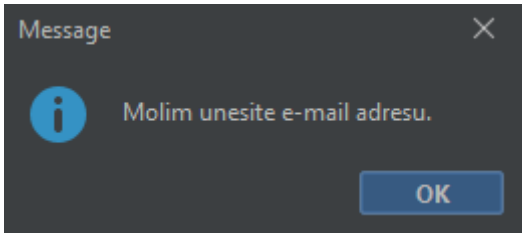
Ukoliko u bazi podataka nema registriranog korisnika, u bazu se unose unešeni podatci, te se registracija uspješno izvršava, korisniku se ispisuje poruka kako je unos podataka u bazu uspješno izvršen, i zatim se korisniku otvara prozor za prijavu u aplikaciju župne knjižnice. U slučaju da je korisnik već napravio registraciju, može odmah otvoriti prozor za prijavu u knjižnicu klikom na gumb za prijavu.



Slika 5.3. Ispis poruke kod uspješne registracije administratora

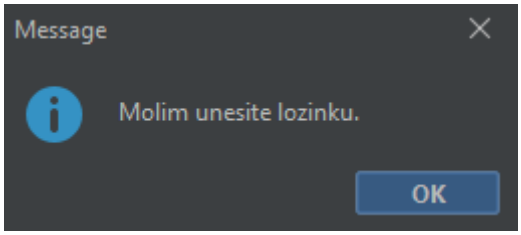
5.2. Prijava

Da bi korisnik kao budući administrator mogao koristiti aplikaciju župne knjižnice, nakon registracije, mora se i prijaviti. Prijava se izvršava na način da korisnik unosi e-mail u polje za unos teksta i lozinku u polje za unos teksta u formatu lozinke. Ako korisnik pokuša kliknuti na prijavu, bez da je unio e-mail adresu ispisati će mu se poruka da mora unijeti e-mail adresu.



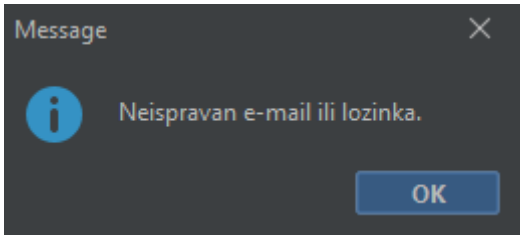
Slika 5.4. Ispis poruke kod pokušaja prijave bez unešene e-mail adrese

Ako nakon unešene e-mail adrese korisnik pokuša kliknuti na gumb za prijavu bez da je upisao lozinku, ispisati će mu se poruka kako mora unijeti lozinku.



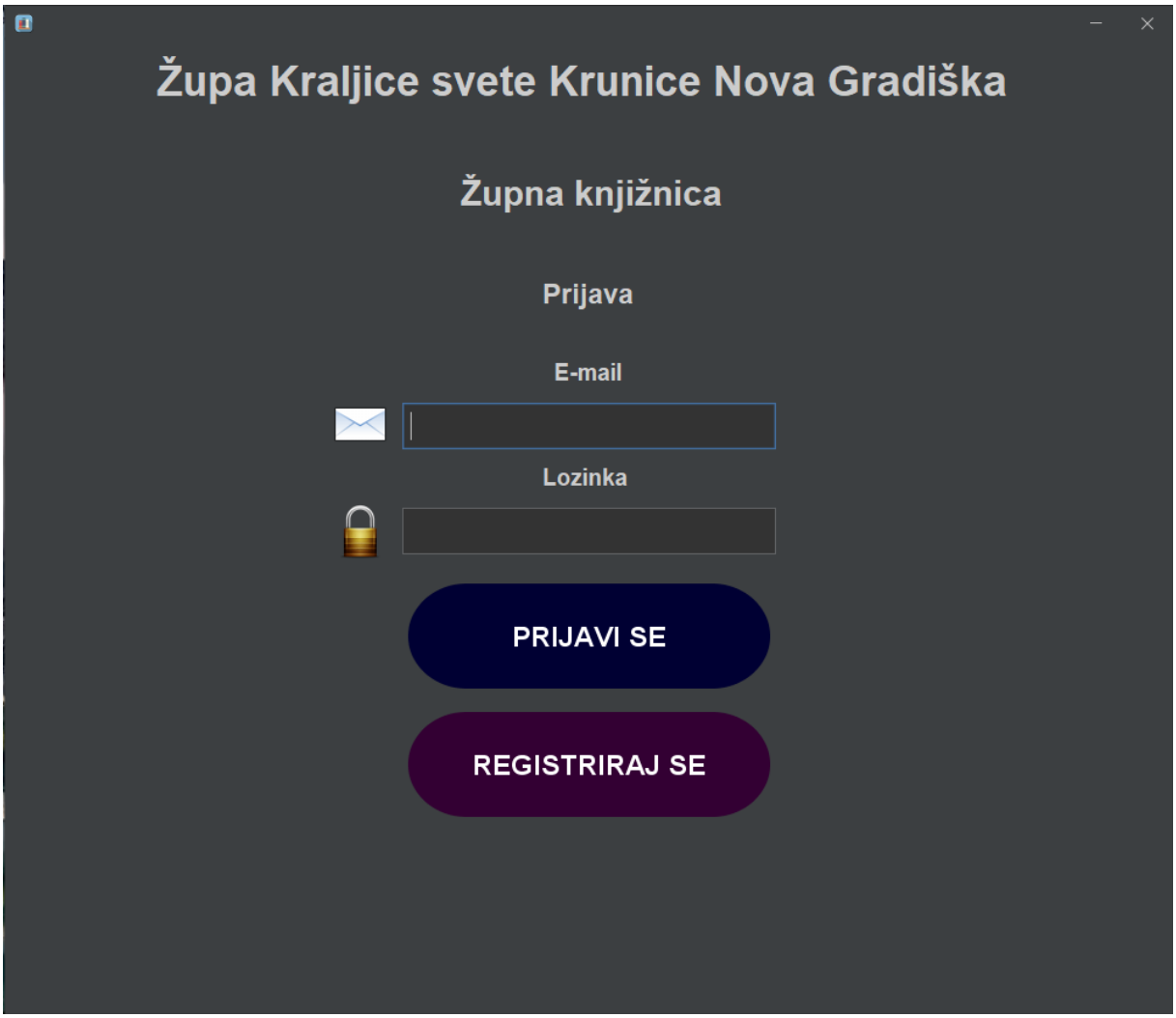
Slika 5.5. Ispis poruke kod pokušaja prijave bez unešene lozinke

Nakon unosa podataka i klika na gumb za prijavu, izvršava se provjera postoji li u bazi podataka korisnik sa unešenim podatcima. Ukoliko korisnik unese neispravan e-mail ili lozinku, neće se moći prijaviti, već će dobiti poruku da su unešeni podatci neispravni.



Slika 5.6. Ispis poruke kod pokušaja prijave sa neispravnim podatcima

Nakon unosa ispravnog formata podataka i klika na gumb za prijavu, izvršava se provjera postoji li u bazi podataka registriran korisnik sa unesenim podatcima, ukoliko postoji, korisniku se otvara početni program aplikacije, i on nastavlja koristiti aplikaciju kao administrator. Ukoliko je korisnik greškom kod registracije kliknuo na gumb za prijavu, i otvorio mu se prozor za prijavu, može se vratiti natrag na prozor za registriranje klikom na gumb za registraciju.

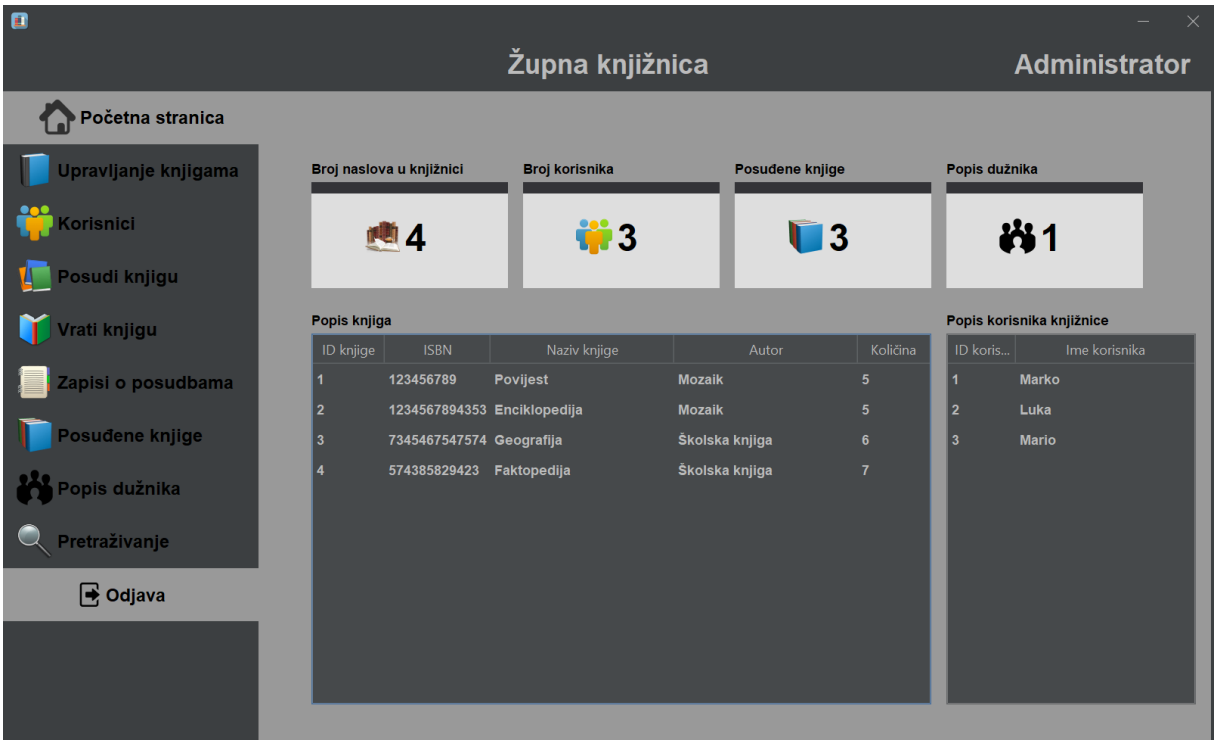


Slika 5.7. Dizajn korisničkog sučelja za prijavu

5.3. Početna stranica

Na početnoj stranici administrator ima vidljive sve bitne informacije izvučene iz baze podataka, kao što je broj knjiga u knjižnici, broj korisnika, broj posuđenih knjiga, broj dužnika, popis knjiga i popis korisnika knjižnice. Ti podatci su isključivo informativnog tipa i oni se ne mogu uređivati, prikazani su svaki zasebno u obliku pločice sa brojem. Broj naslova knjiga u knjižnici je dohvaćen na način da je iz baze podataka dohvaćen broj redova, jer su svi podatci za pojedinu knjigu zapisani u jednom redu. Broj korisnika knjižnice je dohvaćen na isti način kao i broj naslova knjiga u knjižnici. Broj posuđenih knjiga u knjižnici je dohvaćen na način da je dohvaćen broj redova u bazi podataka u kojima je status knjige označen kao ‚posuđen“. Broj dužnika je dohvaćen na način da je iz baze podataka dohvaćen broj knjiga kojima je datum za rok vraćanja posuđene knjige manji od današnjeg datuma, i gdje je status knjige označen kao „posuđen“. S lijeve strane administrator ima izbornik sa gumbima, klikom na neki od

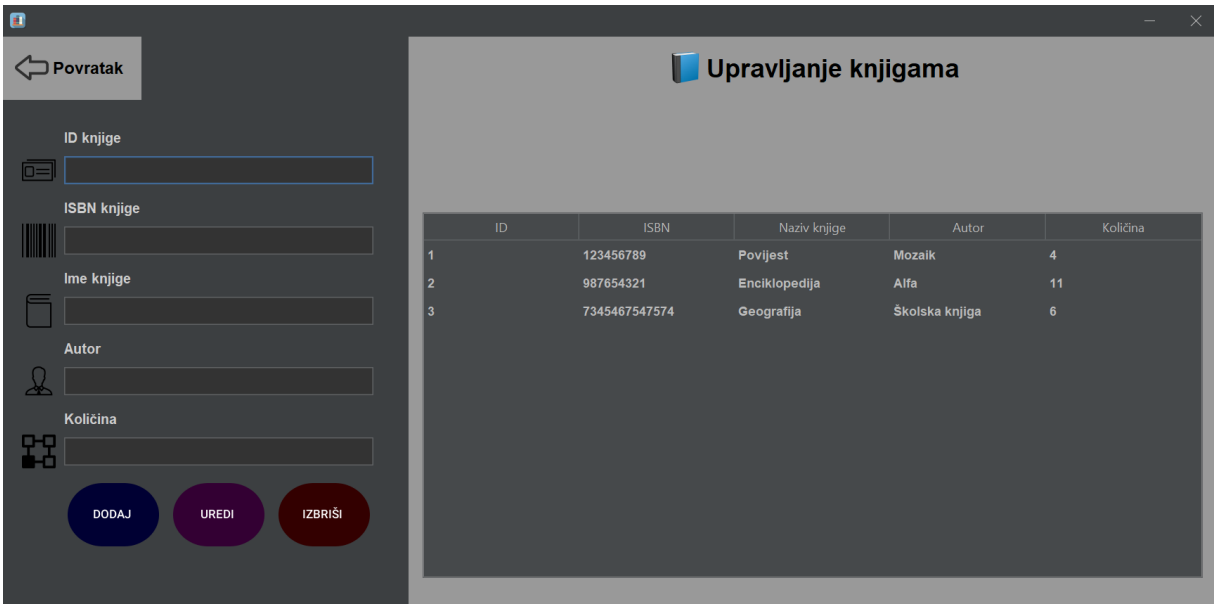
ponuđenih gumba se otvara novi prozor u kojem administrator može uređivati podatke. Prelaskom miša preko izbornika, svaki gumb na kojega miš pokaže promijeni pozadinu u bijelu boju, kako bi administratoru dao do znanja da se na taj gumb može kliknuti, nakon što se miš pomakne izvan granica gumba, pozadina gumba se vrati na početnu tamnosivu boju. Iznimka je gumb za odjavu, njegova pozadina postane crvena kao upozorenje da klikom na nju administrator više neće moći uređivati podatke do ponovne prijave.



Slika 5.8. Dizajn korisničkog sučelja početne stranice

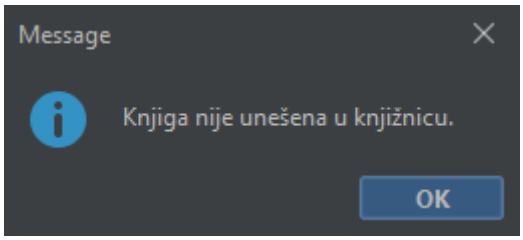
5.4. Upravljanje knjigama

Klikom na gumb za upravljanje knjigama, administratoru se otvara novi prozor u kojem administrator može dodati nove knjige, uređivati postojeće ili brisati knjige iz baze podataka.



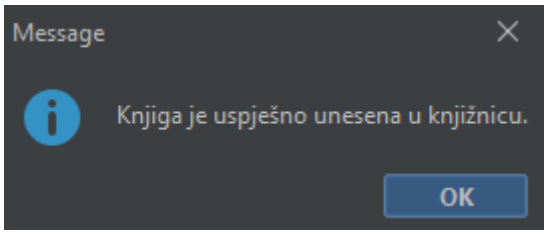
Slika 5.9. Dizajn korisničkog sučelja za upravljanje knjigama

Da bi administrator dodao knjigu u bazu podataka, u polja za unos teksta mora unijeti identifikacijski broj knjige, koji u ovom slučaju označava redni broj knjige, ISBN knjige, ime knjige, autora i količinu. Iako je ISBN isključivo brojčanog formata, za potrebe župne knjižnice je ovdje iznimno dopušten unos i slovčanih vrijednosti, zbog toga što župna knjižnica posjeduje veliki broj starih knjiga koje nemaju ISBN broj, pa administrator može umjesto ISBN broja napisati da knjiga nema ISBN broj i na taj način može uspješno unijeti knjigu u bazu podataka. Ako administrator unese slovčane vrijednosti u polje za ID ili količinu, klikom na gumb za dodavanje ili uređivanje, podatci neće biti niti uređeni u bazi u slučaju da je knjiga već u bazi i da joj se podatci trebaju samo urediti, niti će se zapisati u bazu, u slučaju da knjiga nije unesena u bazi. S desne strane administrator ima vidljivu tablicu s popisom knjiga u bazi podataka. Tablica je povezana sa poljima za unos teksta na način da administrator klikom na neku knjigu u tablici automatski ispuni sva polja za unos teksta sa podatcima iz tablice. Na taj način je administratoru uvelike olakšano uređivanje knjiga iz baze podataka, jer u slučaju da želi obrisati određenu knjigu iz baze podataka, sada samo treba kliknuti na gumb za brisanje knjiga, bez unošenja podataka. Ako administrator unese podatke za knjigu sa već postojećim identifikacijskim brojem, klikom na gumb za dodavanje će mu se ispisati poruka kako unos podataka u knjižnicu nije izvršen.



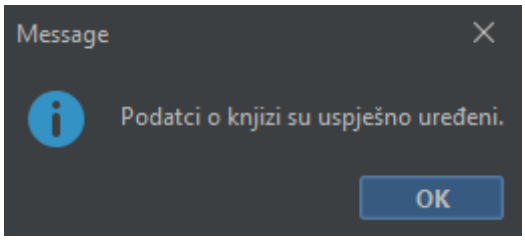
Slika 5.10. Ispis poruke kod upisivanja knjige sa već korištenim identifikacijskim brojem.

Nakon unosa ispravnih podataka i klika na gumb za dodavanje knjige u bazu podataka knjiga će biti upisana u bazu podataka, i administratoru će biti ispisana poruka da je knjiga uspješno upisana u bazu podataka, upis knjige će administratoru biti odmah vidljiv, jer će mu se u tablici sa desne strane prikazati upisana knjiga.



Slika 5.11. Ispis poruke kod uspješnog upisivanja knjige u bazu podataka.

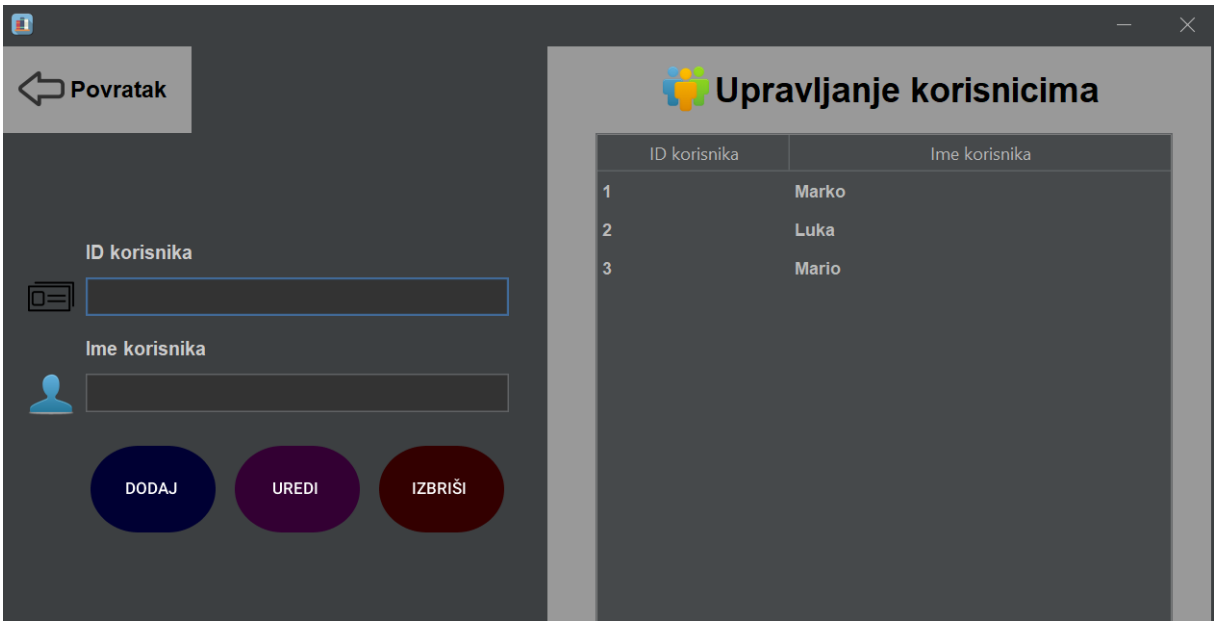
Nakon unosa ispravnih podataka i klika na gumb za uređivanje knjige u bazi podataka, administratoru će se ispisati poruka da su podatci uspješno uređeni, i podatci kod knjige koju je administrator uredio će se odmah promijeniti i u bazi podataka, i u talbici sa desne strane.



Slika 5.12. Ispis poruke kod uspješnog uređivanja knjige u bazi podataka.

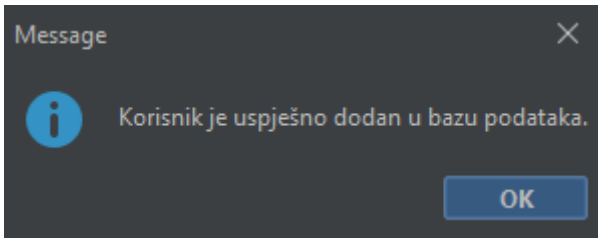
5.5. Upravljanje korisnicima

Klikom na gumb za upravljanje korisnicima administratoru se otvara novi prozor u kojem administrator može dodati korisnike knjižnice.



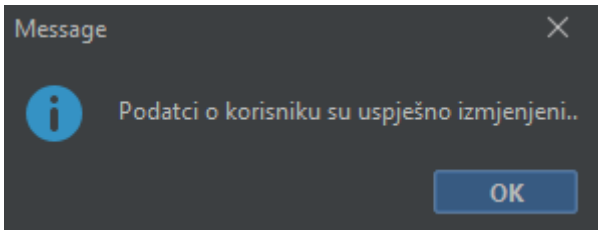
Slika 5.13. Dizajn korisničkog sučelja za upravljanje korisnicima

Taj korak je ključan, jer će omogućiti administratoru da vodi evidenciju kojem korisniku je posudio koju knjigu i kada. Administratoru je osim dodavanja omogućeno uređivanje i brisanje korisnika. Te dvije radnje može odraditi na način da klikne na željenog korisnika iz tablice, zatim može izmjeniti ime i identifikacijski broj korisnika, tu radnju može potvrditi klikom na gumb uredi. Ako administrator u polje za unos identifikacijskog broja unese slovčane vrijednosti, podatci se neće niti zapisati u bazu podataka, niti će se podatci moći urediti. Nakon unosa podataka ispravnog formata, i klika na gumb za dodavanje korisnika, korisnik će se dodati u bazu podataka, podatci će biti odmah vidljivi tablici s desne strane, i administratoru će se ispisati poruka da je korisnik uspješno upisan u bazu podataka.



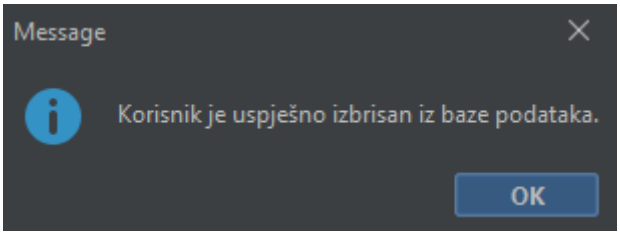
Slika 5.14. Ispis poruke kod uspješnog upisivanja korisnika u bazu podataka.

Nakon unosa podataka ispravnog formata i klika na gumb za uređivanje, podatci će biti uređeni u bazi podataka, biti će odmah vidljivi u tablici s desne strane i administratoru će se ispisati poruka da su podatci o korisniku uspješno uređeni.



Slika 5.15. Ispis poruke kod uspješnog uređivanja podataka o korisniku u bazi podataka.

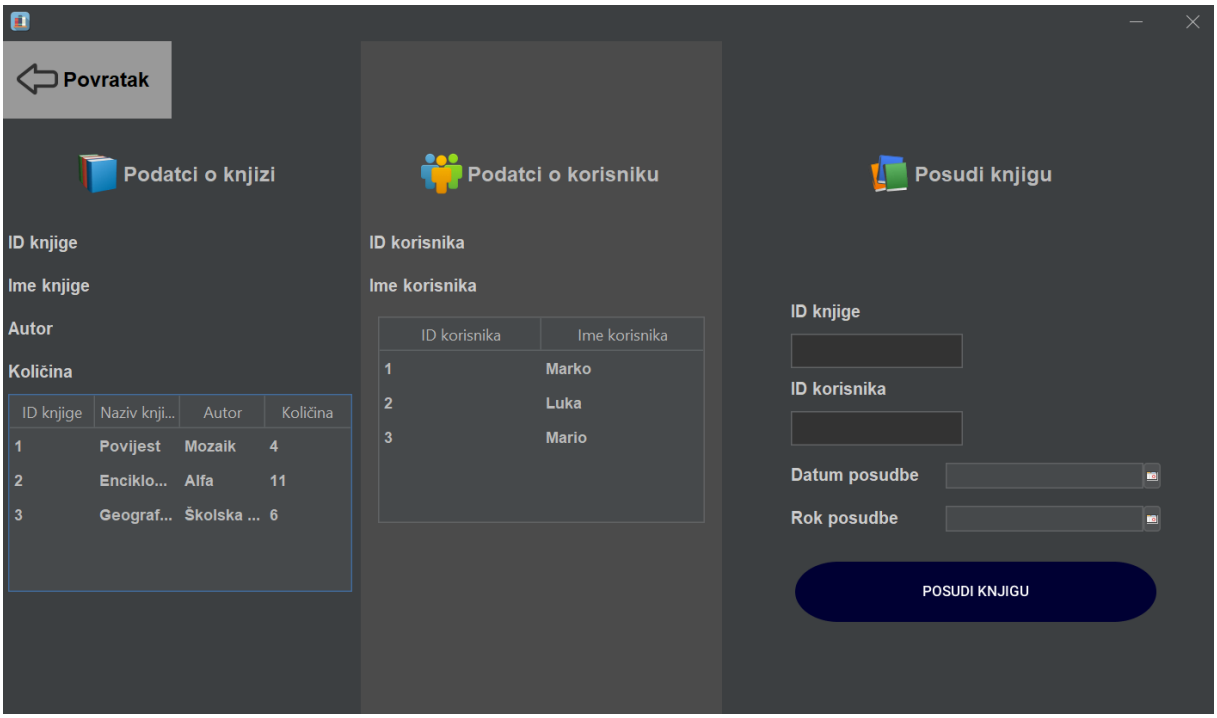
Za brisanje korisnika administrator može unijeti identifikacijski broj i ime korisnika u polja za unos teksta, ili kliknuti na nekog korisnika iz tablice sa desne strane, i zatim kliknuti na gumb za brisanje korisnika. Nakon klika na gumb za brisanje korisnika korisnik se briše iz baze podataka, korisnik će biti odmah izbrisan i iz tablice s desne strane, i administratoru će se ispisati poruka kako je korisnik uspješno izbrisan iz baze podataka.



Slika 5.16. Ispis poruke kod uspješnog brisanja korisnika iz baze podataka.

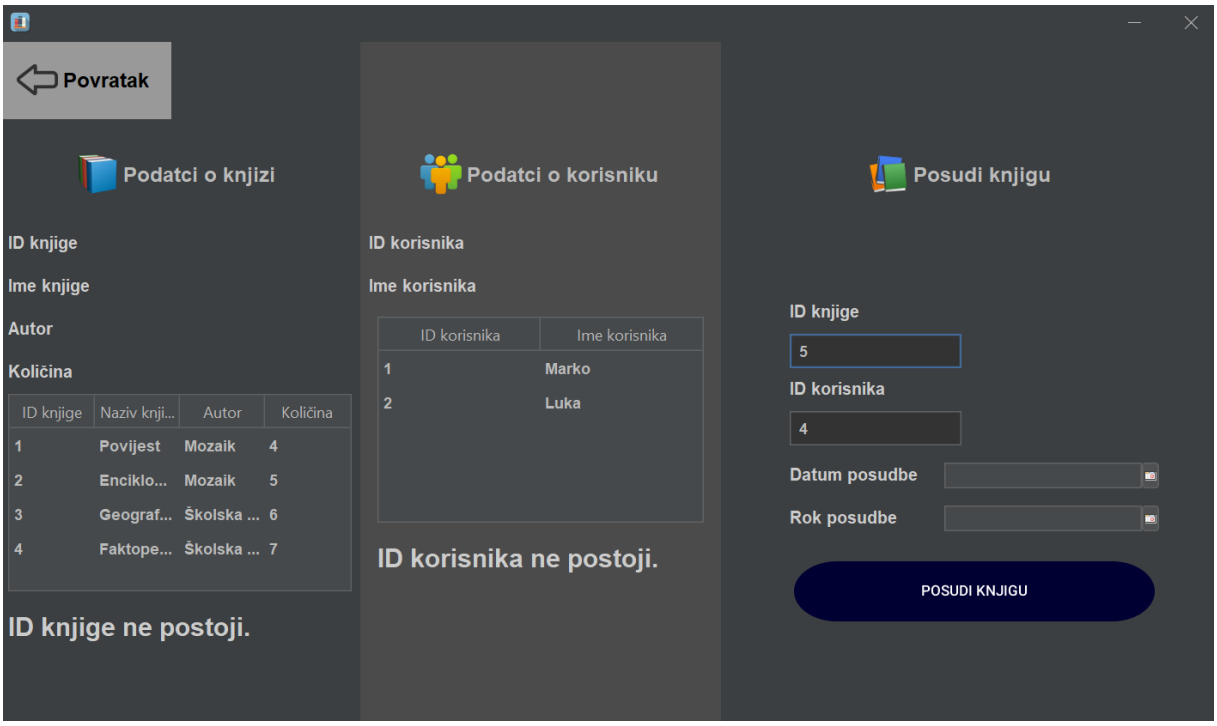
5.6. Posuđivanje knjiga korisnicima

Klikom na gumb za posuđivanje knjiga, administratoru se otvara novi prozor i kojem administrator može određenom korisniku posuditi određenu željenu knjigu.



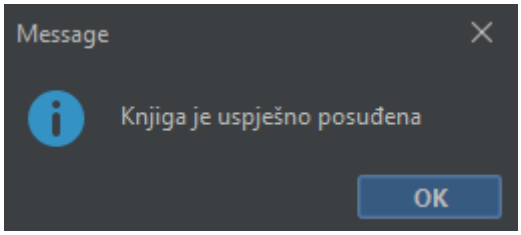
Slika 5.17. Dizajn korisničkog sučelja za posuđivanje knjiga

Posuđivanje knjiga se može napraviti na dva načina, administrator može kliknuti na knjigu iz tablice, i korisnika iz tablice, ta akcija će rezultirati ispisom podataka o knjizi i korisniku, a može to napraviti i na način da u polje za unos teksta unese identifikacijski broj knjige i korisnika, što će također rezultirati ispisom podataka o knjizi i korisniku. Ukoliko administrator unese identifikacijski broj knjige ili korisnika koji ne postoji u bazi, ispisati će mu se poruka kako knjiga i/ili korisnik nisu zapisani u bazi podataka.

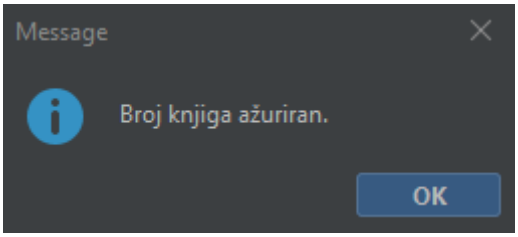


Slika 5.18. Ispis poruke o nepostojanju traženih knjiga i korisnika.

Nakon što je administrator odabrao knjigu i korisnika, u kalendar mora unijeti datum posudbe i rok za vraćanje knjige, nakon toga, potrebno je još samo kliknuti na gumb za posuđivanje knjiga, koji u bazu sprema podatke o posuđenoj knjizi, i u bazi smanjuje brojno stanje određeno knjige za 1 i ispisuje dvije poruke, prvu kojom obavještava administratora da je knjiga uspješno posuđena, i drugu kojom obavještava korisnika kako je broj knjiga ažuriran, odnosno, smanjen za jedan.



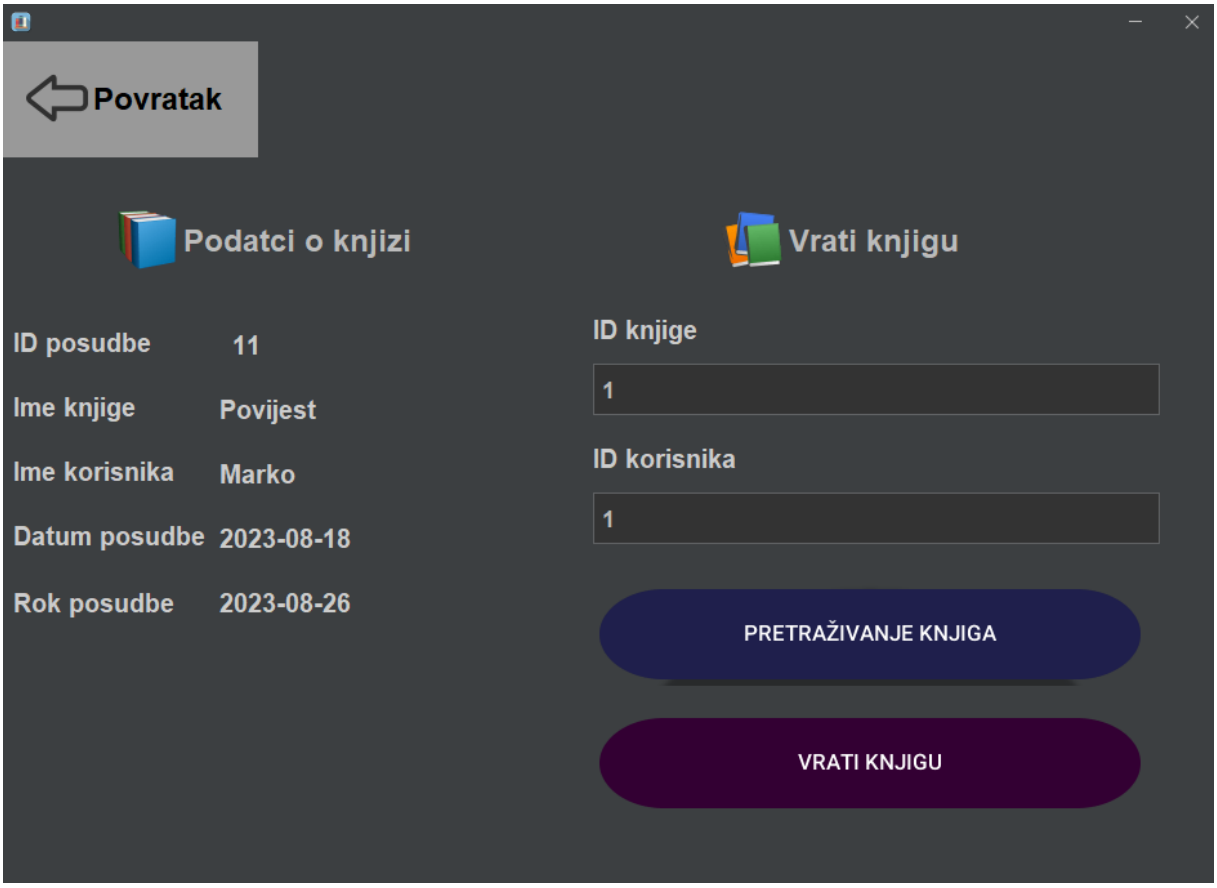
Slika 5.19. Ispis poruke kod uspješnog posuđivanja knjige korisniku.



Slika 5.20. Ispis poruke o uspješnom ažuriranju broja knjiga u bazi podataka.

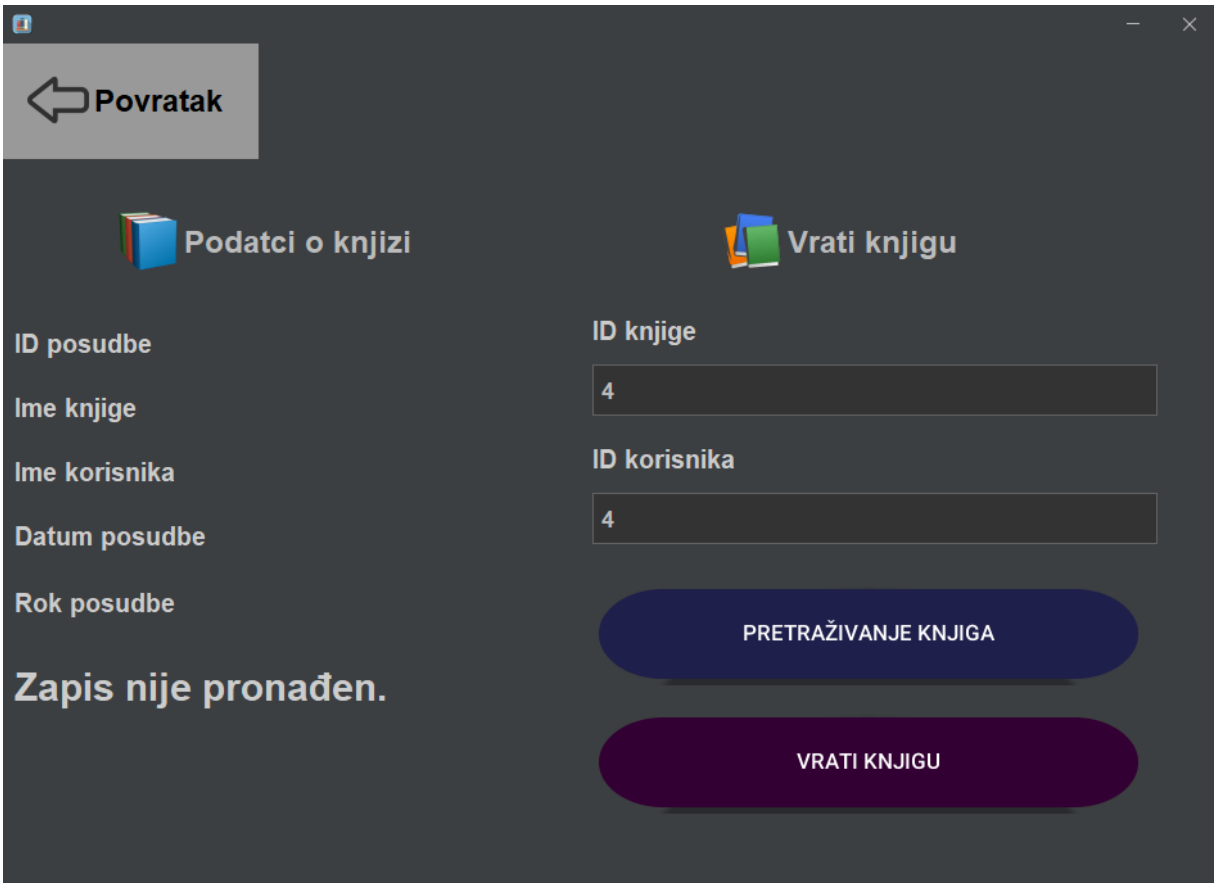
5.7. Vraćanje knjige

Klikom na gumb za vraćanje knjiga, administratoru se otvara novi prozor i kojem administrator može zabilježiti povratak posuđene knjige u župnu knjižnicu.



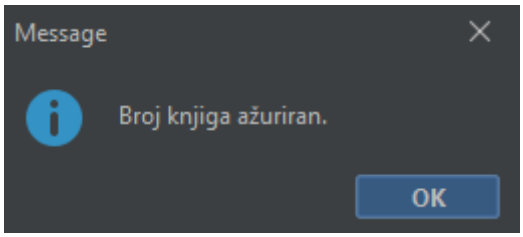
Slika 5.21. Dizajn korisničkog sučelja za vraćanje knjige u knjižnicu

Administrator mora u polja za unos teksta unijeti identifikacijski broj knjige i korisnika, i zatim kliknuti na pretraživanje ukoliko traženi zapis ne postoji, administratoru će se ispisati poruka kako zapis nije pronađen.



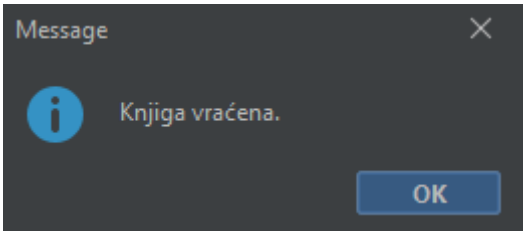
Slika 5.22. Prikaz obavijesti kako ne postoji posuđena knjiga za upisane identifikacijske brojeve knjige i korisnika.

Uoliko su podatci ispravni, ondnosno ako za upisane identifikacijske brojeve knjige i korisnika postoji zapis o posuđivanju, administratoru će se ispisati podatci i korisniku i knjizi koja je posuđena, nakon toga, administrator samo treba kliknuti na gumb za vraćanje knjige, ta akcija će u bazu podataka upisati da je posuđena knjiga vraćena u knjižnicu, i povećat će brojno stanje određene knjige za 1 što će biti vidljivo ispisom poruke kako je broj knjiga ažuriran u bazi podataka.



Slika 5.23. Prikaz obavijesti kako je broj knjiga u bazi ažuriran.

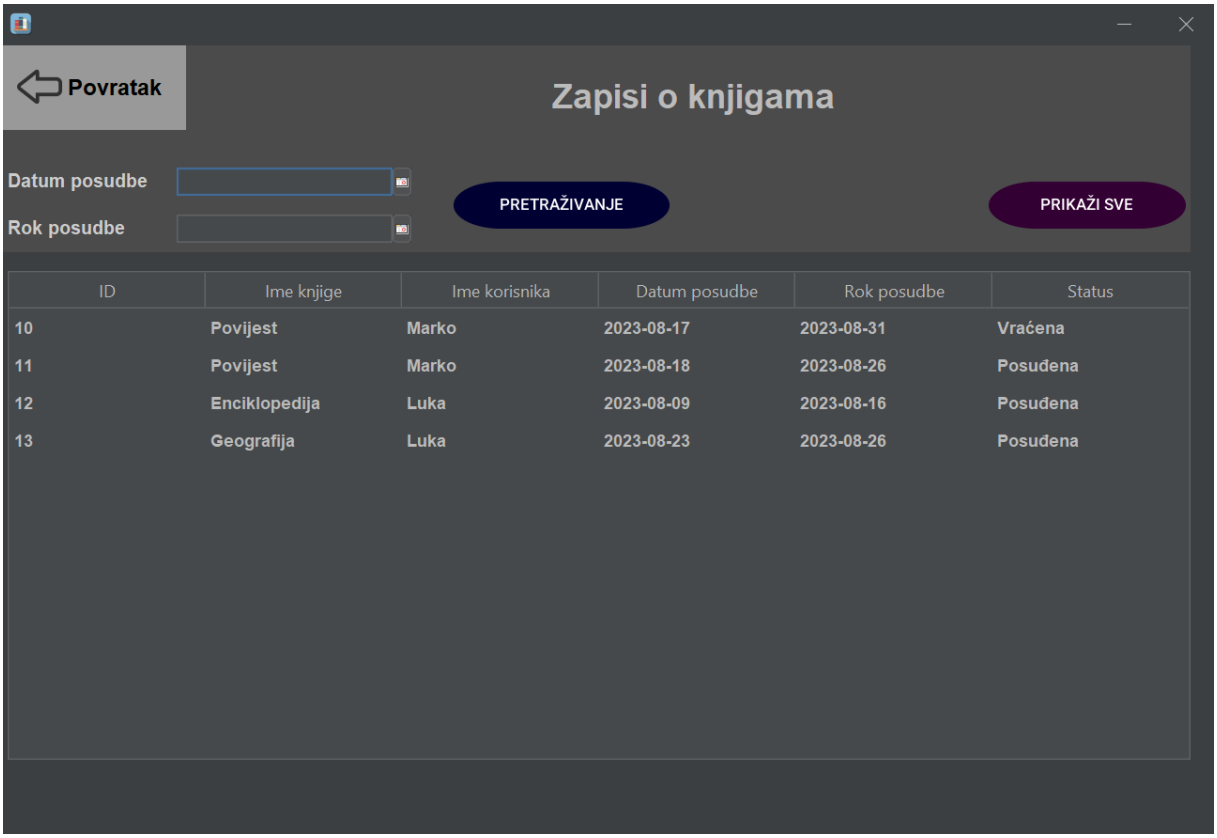
Administratoru će se ispisati poruka s obavijesti kako je posuđena knjiga vraćena u bazu podataka.



Slika 5.24. Prikaz obavijesti kako je posuđena knjiga vraćena u bazu podataka.

5.8. Zapisi o posudbama knjiga

Klikom na gumb za zapise o posudbama, administratoru se otvara novi prozor u kojem može vidjeti sve zapise o posudbama koji su se dogodili.



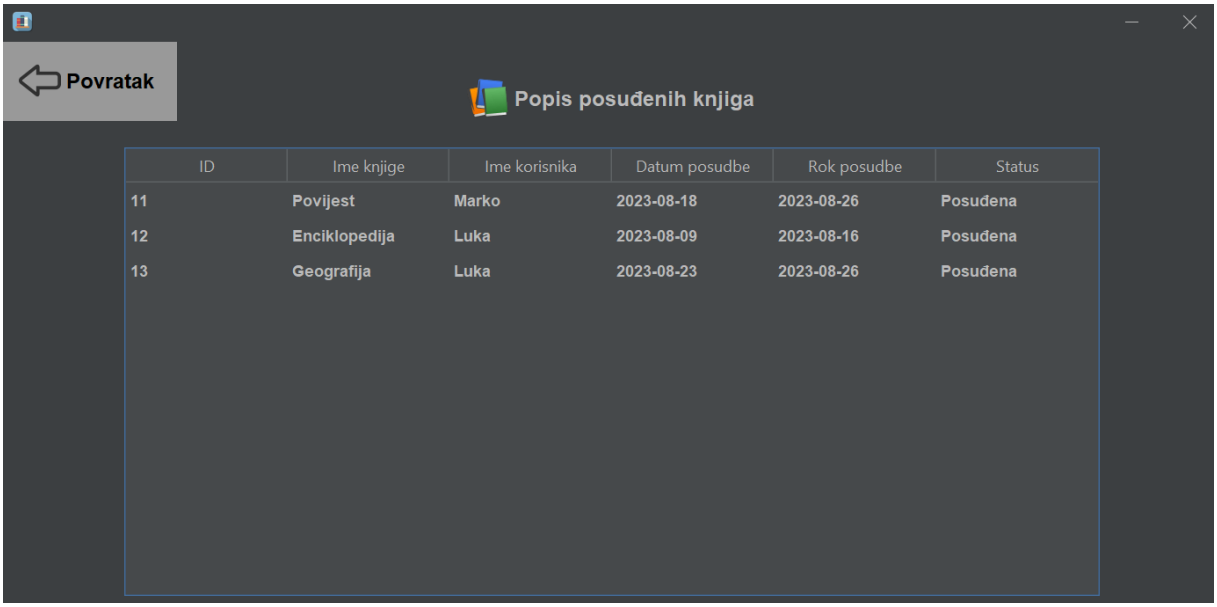
Slika 5.25. Dizajn korisničkog sučelja zapisa o posuđenim knjigama

Ispisani su podatci o knjigama koje su posuđene i vraćene. Ako administrator želi vidjeti zapise iz specifičnog vremenskog perioda, može u kalendaru odabrati datum posudbe i datum roka za povratak knjige, i bit će mu ispisani samo podatci iz tog perioda, no ako pak želi vidjeti sve

podatke, samo je potrebno kliknuti na gumb koji prikazuje sve, i svi podatci će biti ispisani u tablici.

5.9. Posuđene knjige

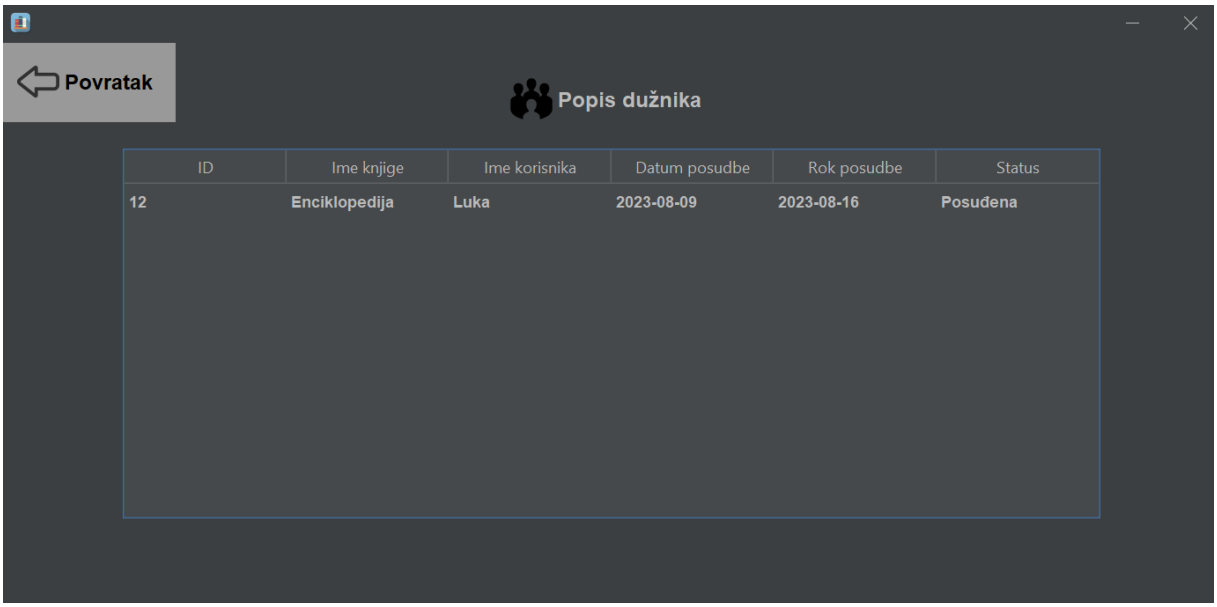
Klikom na gumb za posuđene knjige, administratoru se otvara novi prozor u kojem može vidjeti popis svih trenutno posuđenih knjiga. Ovaj prozor je namjenjen isključivo uvidu u zapise koja knjiga je posuđena kojem korisniku, kako bi administrator znao koje su knjige trenutno posuđene.



Slika 5.26. Dizajn korisničkog sučelja za posuđene knjige

5.10. Popis dužnika

Klikom na gumb za popis dužnika, administratoru se otvara novi prozor u kojem može vidjeti popis svih korisnika kojima je istekao rok za vraćanje posuđene knjige. Ovaj prozor je namjenjen isključivo uvidu u zapise koja knjiga je posuđena kojem korisniku, kako bi administrator znao koji korisnik mu nije vratio koju knjigu.



Slika 5.27. Dizajn korisničkog sučelja popisa dužnika

5.11. Pretraživanje

Klikom na gumb za pretraživanje, administratoru se otvara novi prozor u kojem administrator može pretraživati knjige i autore. Klikom na polje za unos teksta, administrator unošenjem naziva knjige ili autora pokreće filter koji će mu ispisati željeni rezultat ukoliko se knjiga ili autor nalazi u bazi podataka.

Slika 5.28. Dizajn korisničkog sučelja za pretraživanje

5.12. Odjava

Klikom na gumb za odjavu na početnoj stranici, administrator se odjavljuje iz aplikacije, i otvara mu se prozor za ponovnu prijavu.

6. Zaključak

Za izradu ovog diplomskog rada korištenja su znanja stečena tijekom fakultetskog obrazovanja, ponajviše iz predmeta poput programiranja u Java- i, i baza podataka. Aplikacija je korisna zbog toga što će biti korištena kod upravljanja župnom knjižnicom, jer će župniku omogućiti uvid u sve knjige koje župa posjeduje, dodavanje knjiga, njihovo uređivanje i brisanje. Ako župnik posudi nekome knjigu, moći će zapisati kome je posudio koju knjigu, Ako posuđena knjiga ne bude vraćena na vrijeme, osoba kojoj je knjiga posuđena će biti zapisana na listi dužnika, tako da će župnik imati uvid kod koga je knjiga koja nije vraćena na vrijeme. Tako ova aplikacija ima sve bitne funkcionalnosti koje jedna knjižnica treba imati. Trenutna aplikacija je gotova i spremna za korištenje, no, u budućnosti se može nadograditi i poboljšati. Neka od mogućih nadogradnji bi bile spajanje na lokalnu bazu podataka kako bi se izbjegla potreba za lokalnim serverom. Može se izbaciti dio u kojem korisnik sam unosi ID i napraviti se da se ID ispunjava automatski. Moglo bi se implementirati mogućnost da administrator posuđuje knjige uz pomoć tražilice, umjesto upisivanja ID – a knjige.

Literatura

[1] Skupina autora, [online], „Solibry“.
https://apps.microsoft.com/store/detail/solibry/9PJHHFC6D2W5, 7. rujan 2023.

[2] S. H. O. Santos, [online] „Personal Library“.
https://apps.microsoft.com/store/detail/personal-library/9NBLGGH360LG, 7. rujan 2023.

[3] Skupina autora, [online] „CodeAchi Library Management System“.
https://codeachi.com/product/library-management-system/, 7. rujan 2023.

[4] M. Bielemeier, [online] „Private Books Library“.
https://apps.microsoft.com/store/detail/private-books-library/9PJ0393NR5XX, 7. rujan 2023.

[5] Skupina autora, [online] „Book Collector DW“.

https://apps.microsoft.com/store/detail/book-collector-dw/9MWWF80F34PH, 7. rujan 2023.

[6] M. Cindrić, [online], „Izrada mobilne aplikacije za pretvorbu mjernih jedinica“.
https://zir.nsk.hr/islandora/object/etfos%3A2553/datastream/PDF/view, 21. kolovoz 2023.

[7] Skupina autora, [online], „TIOBE Index for August 2023“.

https://www.tiobe.com/tiobe-index/, 21. kolovoz 2023.

[8] Skupina autora, [online], „What is MySQL?“.

https://www.oracle.com/mysql/what-is-mysql/, 22. kolovoz 2024.

[9] Skupina autora, [online], „DB-Engines Ranking“,

https://db-engines.com/en/ranking, 22. kolovoz 2024.

Sažetak

Ovom diplomskom radu je zadatak bio izraditi aplikaciju knjižnice koja će se moći koristiti za potrebe župnika. Razvijena je za Windows operacijski sustav s obzirom da je to operacijski sustav koji župa koristi. U prvom dijelu rada su opisane korištene tehnologije, a u drugom dijelu je opisivana izrada praktičnog dijela diplomskog rada.

Ključne riječi: Java, SQL, knjižnica

Abstract

Creation of the parish library application

The task of this master's thesis was to develop a library application that would be usable for the needs of priest. It was developed for the Windows operating system, given that it is the operating system used by the parish. The first part of the thesis describes the used technologies, while the second part elaborates on the creation of the practical component of the master's thesis.

Keywords: Java, SQL, library

Životopis

Marko Cindrić rođen je 11.06.1997. u Novoj Gradiški. Osnovnu školu završio je u rodnom gradu. Nakon završetka osnovne škole upisuje Elektrotehničku i ekonomsku školu Nova Gradiška gdje stječe strukovnu kvalifikaciju tehničara za računalstvo. Po završetku srednje škole svoje obrazovanje nastavlja upisivanjem preddiplomskog stručnog studija elektrotehnike, smjer informatika na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija Osijek, gdje stječe titulu stručnog prvostupnika inženjera elektrotehnike. Na istom fakultetu upisuje sveučilišni diplomski studij računarstva, smjer informacijske i podatkovne znanosti.

Prilozi

Prilozi su priloženi na DVD-u.