

ELEKTROTEHNIČKA I PROMETNA ŠKOLA OSIJEK

ZAVRŠNI RAD

**EVIDENCIJA KNJIGA PUTEM
QR KODA U KNJIŽNICI**

Davor Gotal

Osijek, svibanj 2020.

ELEKTROTEHNIČKA I PROMETNA ŠKOLA OSIJEK

ZAVRŠNI RAD

EVIDENCIJA KNJIGA PUTEM QR KODA U KNJIŽNICI

Učenik: Davor Gotal
Razred: 4.eT2
Obrazovni sektor: Elektrotehnika i računalstvo
Zanimanje: Tehničar za računalstvo
Mentor: Dinka Šafar Đerki

Opis zadatka:

Korištenjem Visual Studia napraviti desktop aplikaciju za evidenciju knjiga. Korisnik/Knjižničar unosi podatke o knjigama u bazu podataka putem desktop aplikacije. Nakon što unese podatke o knjizi za nju se automatski generira QR kod u kojem je zapisan ID knjige. Potrebno je povezati desktop aplikaciju s Microsoft servera kako bi mogli pristupiti bazi podataka u kojoj će biti pohranjeni podaci o knjigama i korisnicima knjižnice. Prilikom podizanja knjiga knjižničar upisuje/skenira identifikacijski broj korisnika te nakon što je skeniran kod korisnika, ako se skenira već uneseni kod od knjige, odabire se opcija vrati. Ako je novi kod onda se nudi opcija posudi. Ako korisnik nije vratio knjigu na vrijeme mora platiti zakasninu prilikom vraćanja knjige koja će biti vidljiva knjižničaru.

SADRŽAJ

1. UVOD	1
2. TEHNOLOGIJE U IZRADI APLIKACIJE	2
2.1. Visual Studio	2
2.2. Emgu CV	3
2.3. ZXing.....	4
2.4. Baza podataka.....	5
2.5. QR kodovi.....	6
3. RAZVOJ APLIKACIJE ¹	10
3.1. Grafičko sučelje.....	10
3.2. Učitavanje slike s kamere.....	18
3.3. Dekodiranje QR kodova	21
3.4. Zapis podataka u bazu podataka.	23
4. ZAKLJUČAK.....	26
LITERATURA	27
DODATAK (PRILOZI)	28

1. UVOD

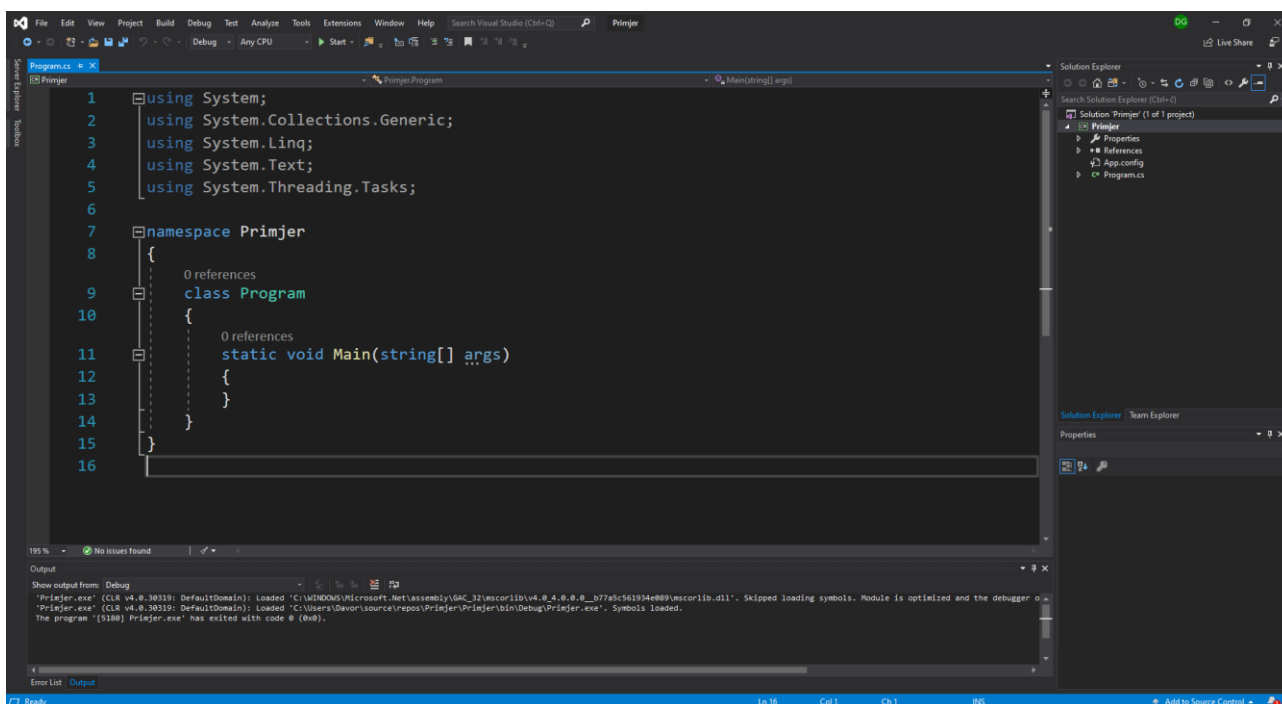
Napretkom tehnologije dolazi do povećanja digitalizacije svijeta u kojem živimo, što dovodi do izrade aplikacija koje će nam olakšati svakodnevne poslove i ubrzati pronalazak željenih informacija te upravljanje njima. Evidencija knjiga putem QR kodova u knjižnici će pružiti višestruke pogodnosti dodatnih raznovrsnih informacija koje omogućavaju popularizaciju knjiga naročito za mlađu populaciju. QR kodovi će se skenirati putem računalne kamere ukoliko želimo posuditi ili vratiti knjigu. Aplikacija nudi lako pretraživanje knjiga gdje će korisnik moći jednostavno pretražiti knjige po autoru ili nazivu, a uz njih će pisati njihov žanr.

2. TEHNOLOGIJE U IZRADI APLIKACIJE

U izradi svoje aplikacije koristio sam SSMS, Visual Studio, Emgu CV, ZXing i QR kodove jer omogućuju jednostavno pisanje koda kao i kretanje između datoteka, klasa, sučelja i implementacija. Korištene tehnologije su opisane u daljnjem tekstu.

2.1. Visual Studio

Visual Studio je integrirano razvojno okruženje koje se koristi za razvoj računalnih programa, web stranica, web usluga i mobilnih aplikacija. Podržava veliki broj različitih programskih jezika kao što su C#, Python, JavaScript, C++, NodeJS, F#, Typescript i druge. Visual Studio uključuje i program za detekciju i uklanjanje pogrešaka. Visual Studio je integriran s Microsoft SQL Serverom što omogućava povezivanje aplikacije s bazom podataka koja će biti kreirana u SQL Server Managment Studiju. Dostupan je na Windowsu i Macu. [1]



Slika 2.1: Program koji se izvršava, ali ne radi ništa

2.2. Emgu CV

Emgu CV je biblioteka za računalnu obradu slike na različitim platformama. Pisana je čistim C#. Postoje razni načini pristupa kameri vašeg sustava pomoću C#. Jedan od najboljih načina je korištenje biblioteke Emgu CV. Razlikuje se od ostalih biblioteka za računalnu obradu slika jer ne koristi neprovjereni kod (engl. unsafe code). Pomoću Emgu CV moguće je pozivati funkcije iz OpenCV-a u Visual Studio. OpenCV funkcije su napisane u izvornom C/C++ jeziku. Visual Studio, koji je .NET okruženje ne može izravno pozivati funkcije napisane u izvornom C/C++ jeziku. Emgu CV je ugrađena u Mono platformu što omogućava .NET-u prelazak na druge operativne sustave kao što su Linux i Mac OS. Također, može se pokrenuti na bilo kojoj platformi koju Mono podržava, poput iOS, Linux, Windows Phone, Android, Mac OS X. [2]

```
using Emgu.CV;  
using Emgu.CV.Structure;  
using Emgu.Util;
```

Slika 2.2: Korištene klase iz biblioteke Emgu.CV

2.3. ZXing

ZXing je biblioteka otvorenog koda koja omogućuje obradu i skeniranje QR kodova putem kamere u pejzažnom ili portretnom načinu rada. Implementirana je u Javi, ali ima priključke prema drugim jezicima. Unutar ZXing biblioteke nalazi se ZXing.NET priključak koji će se koristiti u izradi ovog završnog rada jer podržava c# kao i dekodiranje i generiranje barkodova. Kao što su: QR kod, EAN-13, PDF417 i slično unutar slike.[3]

```
using ZXing;  
using ZXing.QrCode;  
using ZXing.Common;
```

Slika 2.3: Korištene klase iz biblioteke ZXing

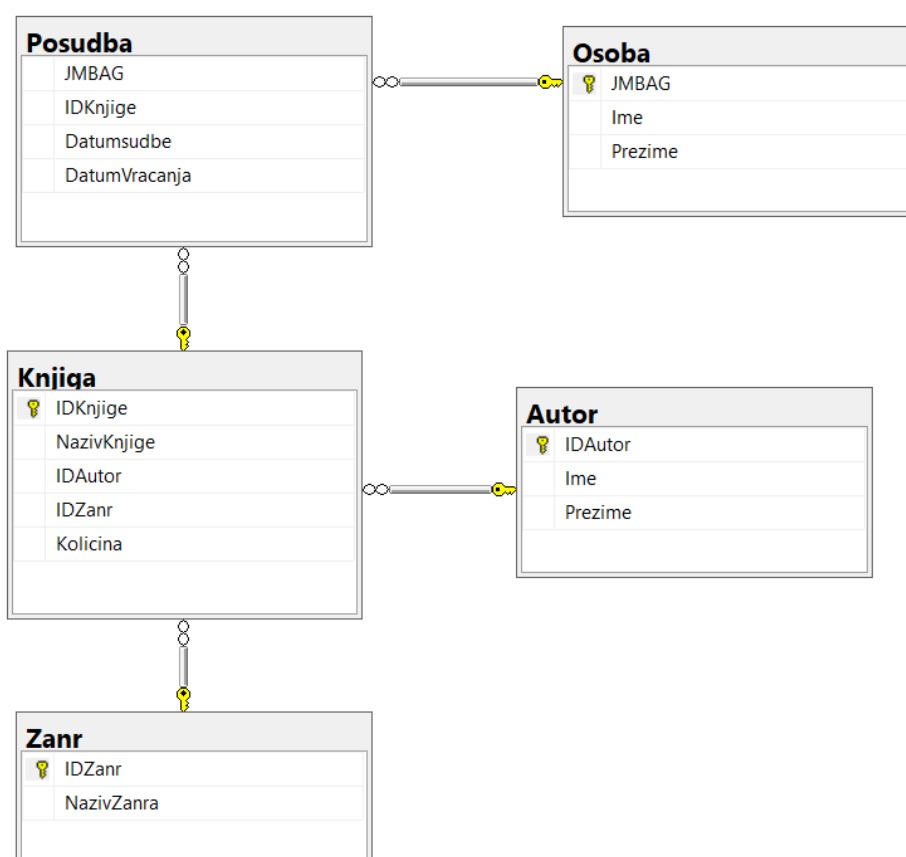
2.4. Baza podataka

Baza podataka je skup međusobno povezanih podataka koji su pohranjeni u vanjskoj memoriji računala. Sustavom za upravljanje bazom podataka možemo upisivati, mijenjati, brisati i čitati podatke.

Sustav za upravljanje bazama podataka (engl. Data Base Management System - DBMS) je poslužitelj baze podataka koji oblikuje fizički prikaz baze u skladu s željenom logičkom strukturom, obavlja sve operacije s podacima u ime klijenta, brine se za sigurnost podataka te automatizira administrativne poslove s bazom. Podržava razne baze od kojih svaka može imati svoju logičku strukturu, ali u skladu s istim modelom.

Model podataka je sustav sastavljen od skupa objekata, operacija i pravila cjelovitosti koja određuju kako sve može izgledati logička struktura baze podataka.

Sustav za upravljanje bazama podataka se po modelu podataka dijeli na: mrežne, hijerarhijske, relacijske, relacijske baze s objektno orijentiranim proširenjima i objektno orijentirane baze podataka. Najčešći korišteni model je relacijski model kojeg je uveo E.F.Codd 1970.e godine.



Slika 2.4: Dijagram kreirane baze podataka

Na slici 2.4. je prikazana baza podataka koja se sastoji od pet tablica: Autor, Posudba, Knjiga, Žanr i Osoba koje su međusobno povezane zahvaljujući jednoj od osnovnih prednosti baze podataka. Tablica Autor se sastoji od tri atributa: IDAutor koji je ujedno i

primarni ključ, Ime i Prezime. Tablica Knjiga se sastoji od 4 atributa: IDKnjige koji je ujedno i primarni ključ, NazivKnjige te dva strana ključa IDAutor i IDZanr. Tablica Autor i Knjiga povezane su preko atributa IDAutor. [4]

2.5. QR kodovi

QR kod (*engl. Quick Response Code*) je dvodimenzionalni barkod (*engl. barcode*) dizajniran od strane *Denso Wave Company* koji se sastoji od crno bijelo kvadratića, a najveća mu je primjena pretvaranje koda u informaciju o stavci na koju je pričvršćen. QR kod je prvi put dizajniran 1994. godine za automobilsku industriju u Japanu, ali su se vrlo brzo počeli koristiti izvan automobilske industrije zbog svoje čitljivosti kao i velikog kapaciteta za pohranu podataka.

QR kodovi imaju visok kapacitet pohrane podataka za razliku od konvencionalnih bar kodova koji su sposobni primiti otprilike dvadeset znamenaka, QR kod je sposoban nekoliko stotina put više informacija pohraniti. Također, QR kod mogu pohraniti različite tipove podataka, kao što su brojevi, abecedni znakovi, simboli, Kana, Hiragana, Kanji, binarni i kontrolni kodovi.

QR kodovi se mogu obnoviti ako im se nanese neka šteta poput prljavštine i manjih oštećenja zahvaljujući Reed-Solomon-ovim kodovima (*engl. Reed-Solomon code*). Reed-Solomon-ovi kodovi su skupina kodova za ispravljanje pogrešaka. Moguće je vratiti najviše 30% kodnih riječi. Kodna riječ je jedinica koja konstruira područje podatka. [5]



Slika 2.5: Razlika između QR koda i jednodimenzionalnog koda¹

¹ <http://www.nacs.org/LinkClick.aspx?fileticket=D1FpVAvvJuo%3D&tabid=1426&mid=4802>

3. RAZVOJ APLIKACIJE

U ovom poglavlju opisana je izrada desktop aplikacije, njezine funkcionalnosti i dizajn. Razvoj funkcionalnosti obuhvaća rad s Windows formama i *Microsoft SQL Serverom*. Windows forme su omogućili interakciju korisnika i programa putem grafičkog sučelja, a *Microsoft SQL Server* s osnovnom funkcijom spremanja i dohvaćanja podataka omogućio je upravljanjem relacijskim tablica.

3.1. Grafičko sučelje

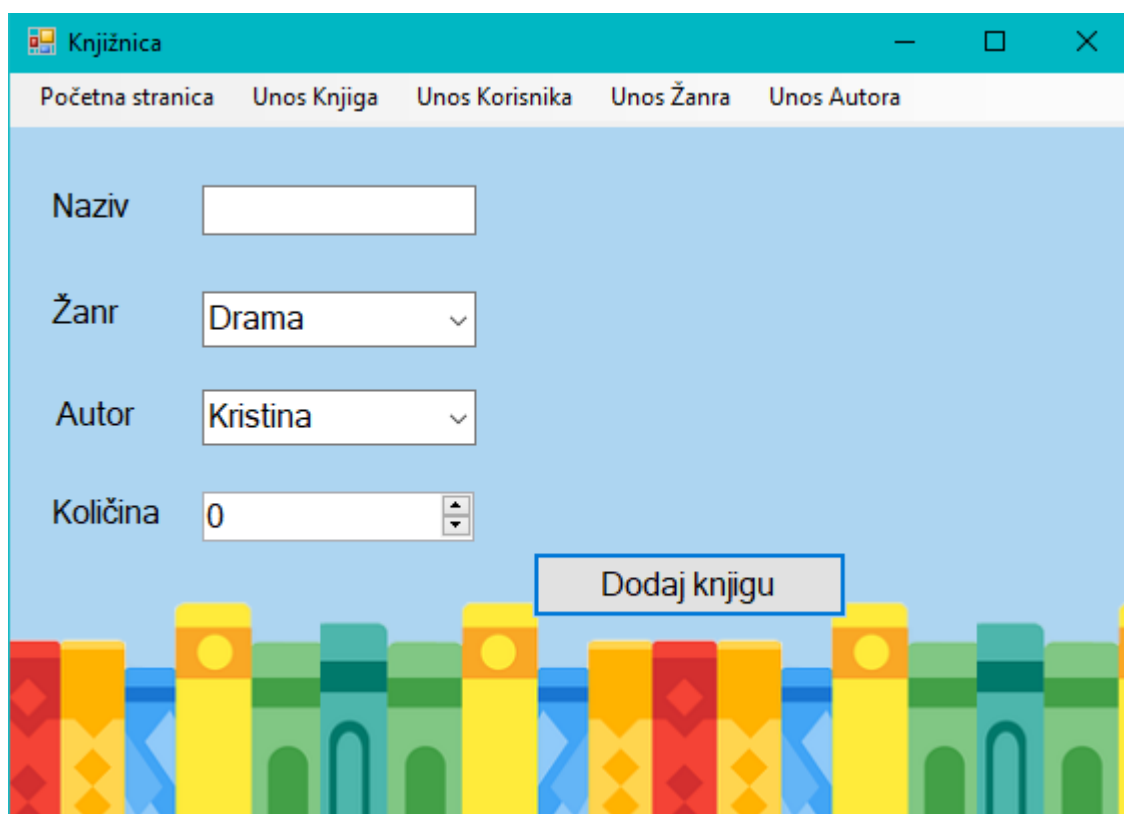
Grafičko korisničko sučelje (*engl. Graphical User Interface*, skraćeno *GUI*) omogućuje interakcije čovjeka s računalom kroz manipulaciju grafičkim elementima i dodacima uz pomoć tekstualnih poruka i obavijesti. Korisnik pritom ne mora imati prethodno znanje iz područja programiranja i baza podataka.



Slika 3.1: Izgled grafičkog sučelja

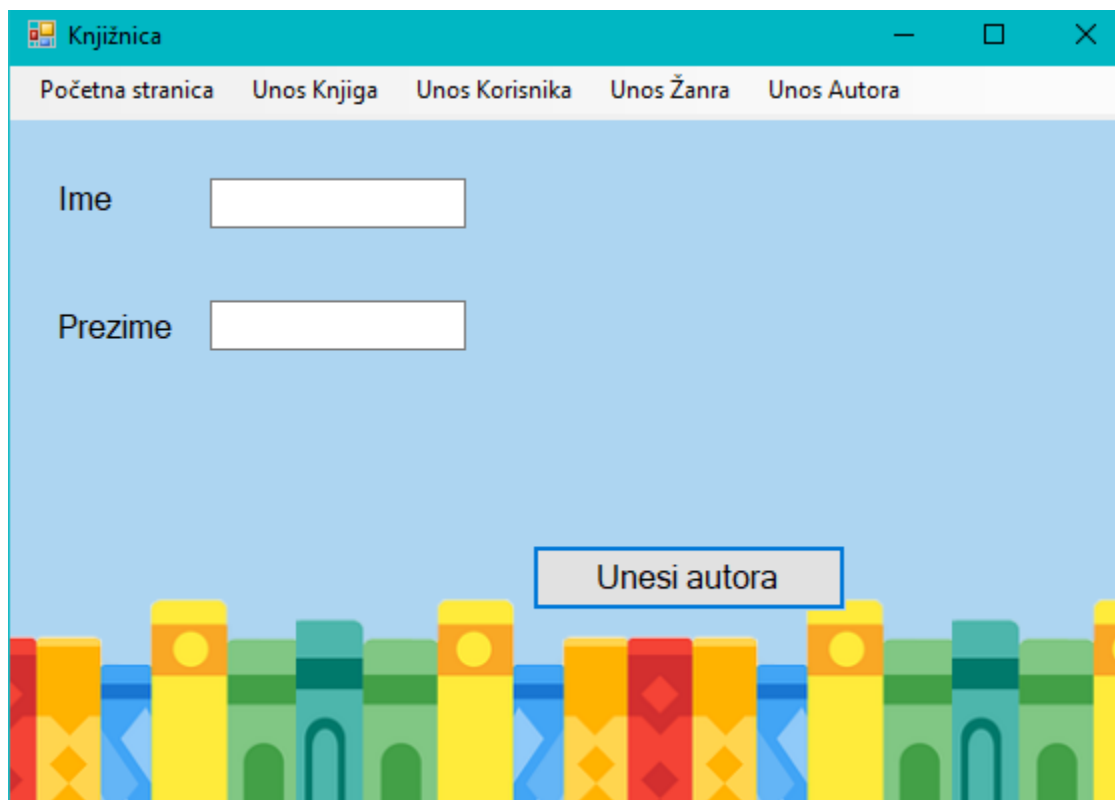
Slika 3.1 daje prikaz forme pod nazivom „Početna stranica“. Ona služi za odabir sljedećih formi pomoću dva gumba (*engl. buttons*): „posudba vraćanje“ i „unos“. Gumb „posudba vraćanje“ omogućuje vraćanje i posuđivanje knjiga kao i pretraživanje knjiga. Gumb „unos“

vodi na forme gdje se unose novi autori, žanrovi, korisnici i knjige.

The image shows a screenshot of a software application window titled "Knjižnica". The window has a teal header bar with standard window controls (minimize, maximize, close) on the right. Below the header is a menu bar with five items: "Početna stranica", "Unos Knjiga", "Unos Korisnika", "Unos Žanra", and "Unos Autora". The main area of the window is light blue and contains a form for adding a book. The form has four fields: "Naziv" (a text input field), "Žanr" (a dropdown menu with "Drama" selected), "Autor" (a dropdown menu with "Kristina" selected), and "Količina" (a numeric input field with "0" and up/down arrows). Below these fields is a button labeled "Dodaj knjigu". At the bottom of the window is a decorative border consisting of a row of colorful, stylized book spines in red, yellow, green, and blue.

Slika 3.2: Prikaz grafičkog sučelja „Knjižnica“

Na slici 3.2 prikazana je forma pod nazivom „Knjižnica“ koja otvara formu „Unos knjiga“ u objektu panel kao zadanu formu. Iznad objekta panel nalazi se MenuStrip s nazivima formi za unos: „Unos Knjiga“, „Unos Korisnika“, „Unos Žanra“ i „Unos Autora“ i nazivom „Početna Stranica“. Prilikom klika na naziv forme, otvara nam se željena forma.

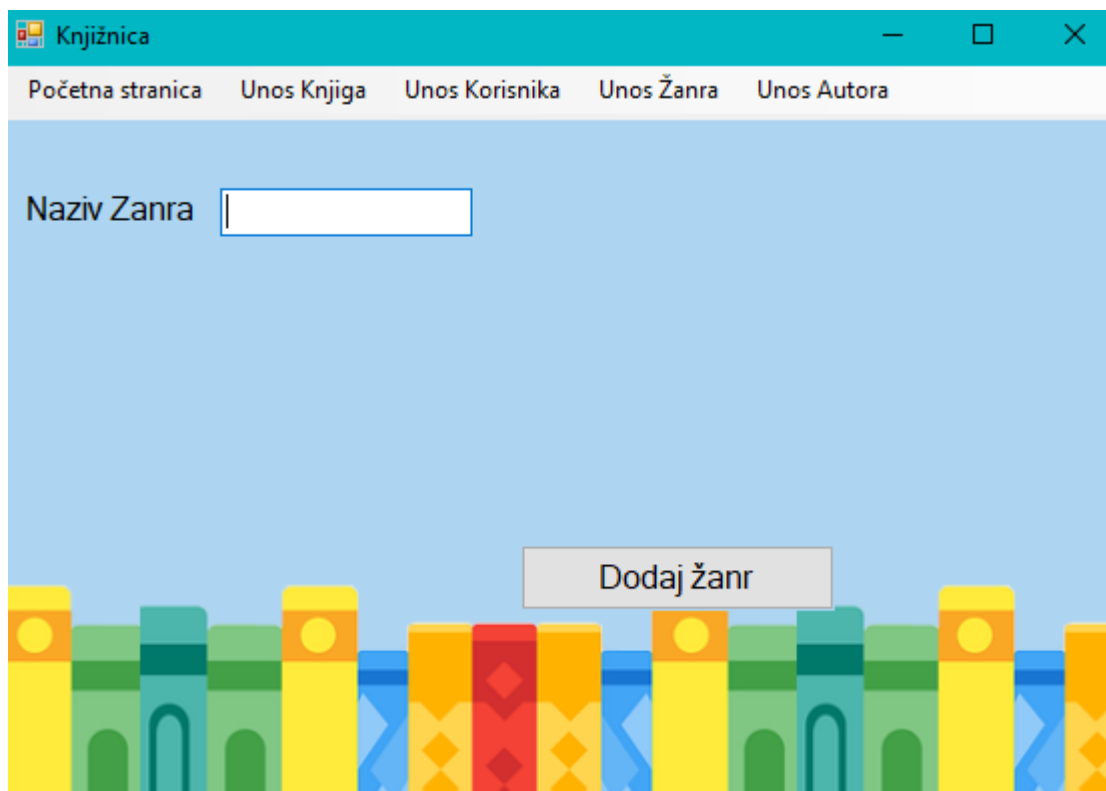


Slika 3.3: Prikaz grafičkog sučelja „Unos Autora“

Na slici 3.3 prikazana je forma „Unos Autora“. Forma „Unos Autora“ sastoji se od dvije labela: „Ime“ i „Prezime“, dva textboxa te gumba (*engl. button*) „Unesi autora“. Korisnik mora popuniti Ime i Prezime u textbox te pritisnuti gumb „Unesi autora“ kako bi spremio novog autora u bazu podataka. Prilikom unošenja nepotpunih podataka javlja se greška u obliku label objekta.

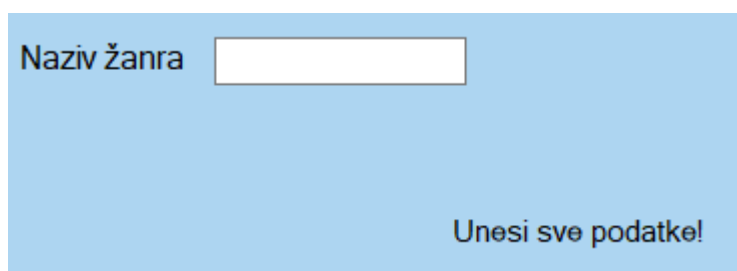


Slika 3.4: Prikaz label upozorenja



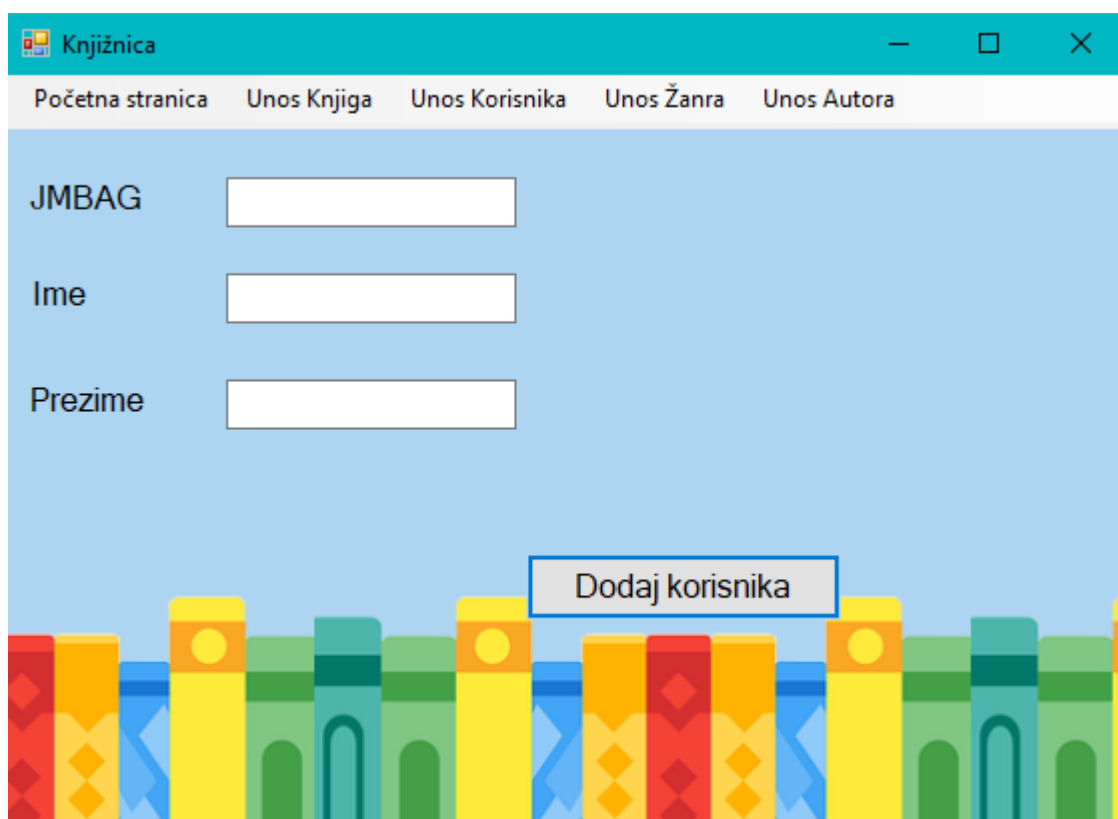
Slika 3.5: Prikaz grafičkog sučelja „Unos Žanra“

Na slici 3.4 prikazana je forma „Unos Žanra“ koja se sastoji od label „Naziv Žanra“, textboxa i gumba „Dodaj žanr“. Korisnik upisuje novi žanr u textbox te prilikom pritiska na gumb „Dodaj žanr“ dodaje novi žanr u bazu podataka. Ukoliko podatak nije unesen u textbox javlja se greška u obliku lablea koja govori korisniku da unese sve podatke.



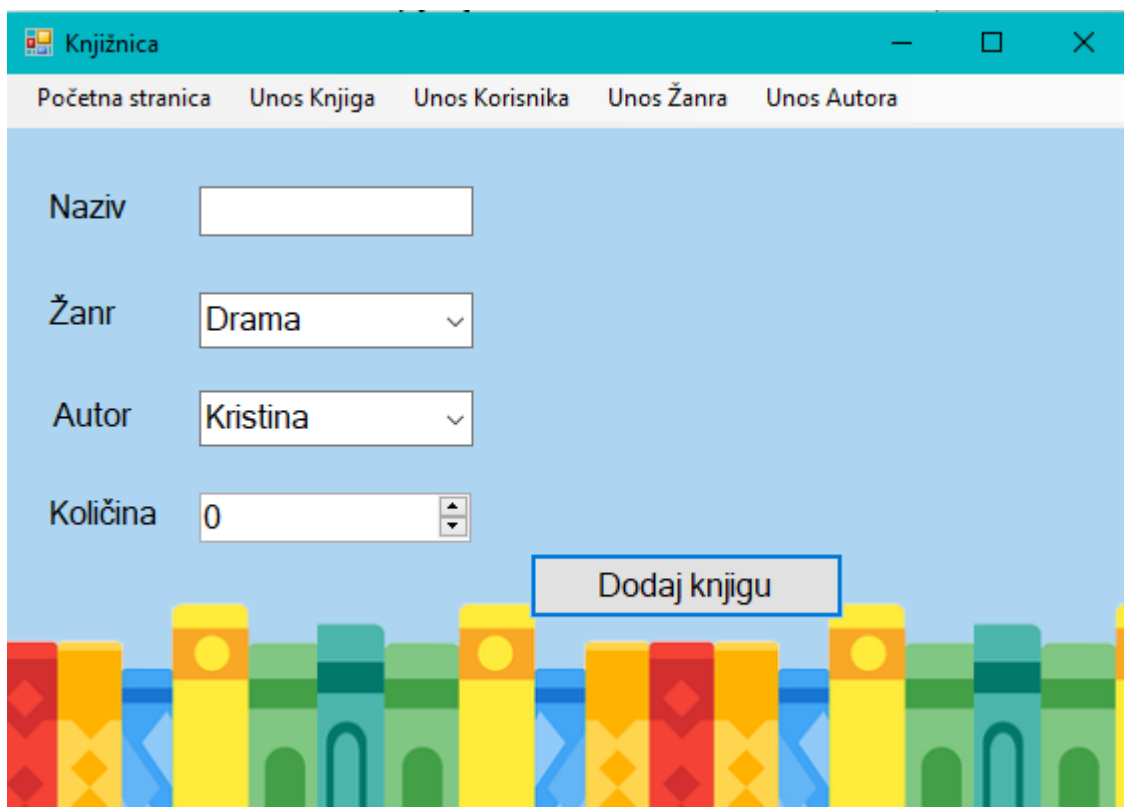
Slika 3.6: Prikaz label upozorenje prilikom unosa žanra

Na slici 3.6 vidljiva je poruka upozorenja: „Unesi sve podatke!“.



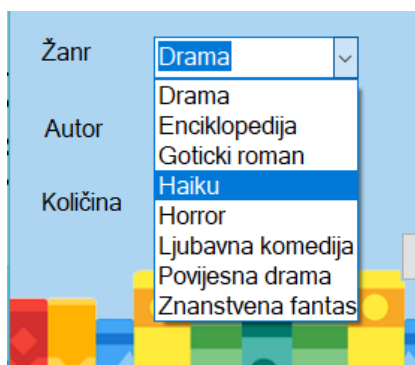
Slika 3.6: Prikaz grafičkog sučelja „Unos Korisnika“

Na slici 3.5 prikazana je forma „Unos Korisnika“ koja omogućuje pohranu korisnika u bazu podataka. Na formi se nalaze tri labela: „JMBAG“, „Ime“ i „Prezime“, tri textboxa i jednog gumba (*engl. button*): „Dodaj korisnika“. Korisnik unosi podatke u textboxove te ih pohranjuje u bazu prilikom pritiska na gumb „Dodaj korisnika“. Ukoliko nisu uneseni svi podaci javlja se greška u obliku labela.



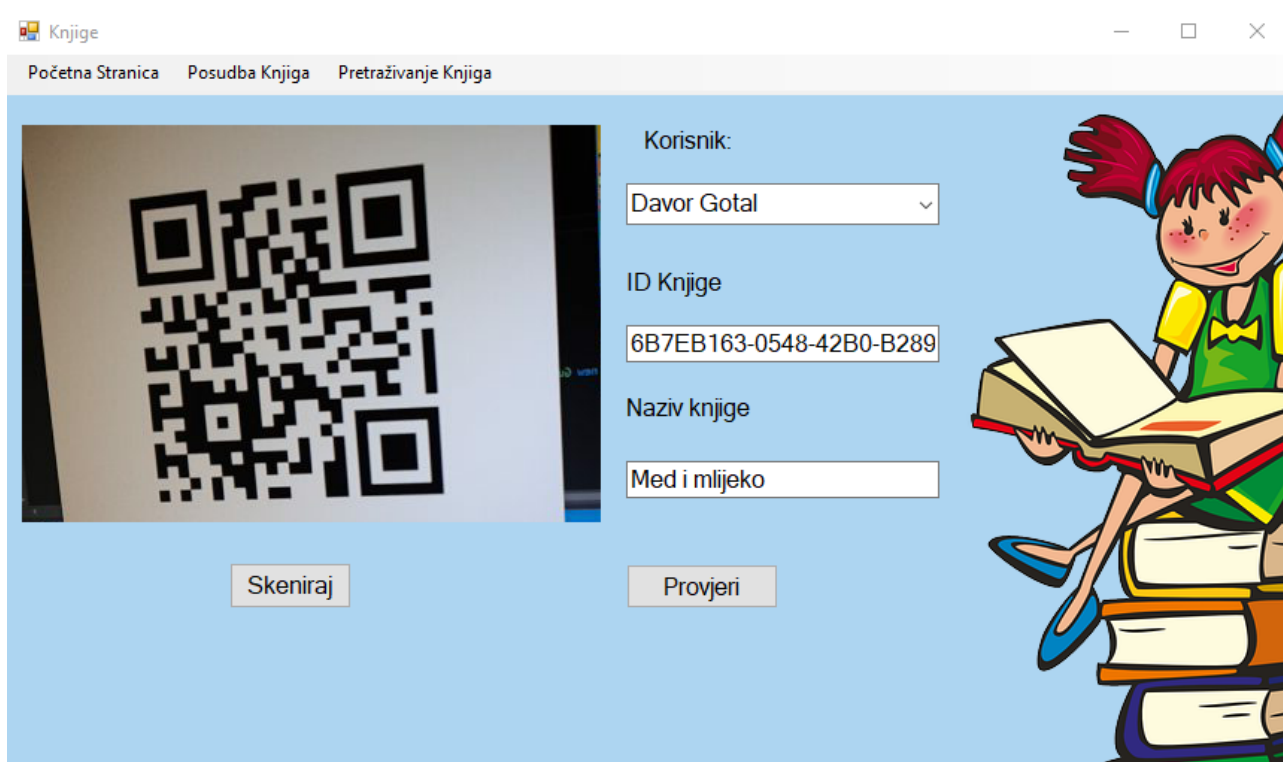
Slika 3.7: Prikaz grafičkog sučelja „Unos Knjiga“

Na slici 3.6 prikazana je forma „Unos Knjiga“ koja se sastoji od četiri lablea: „Naziv“, „Žanr“, „Autor“ i „Količina“, jedan textboxa za unos naziva knjige, dva ComboBoxa koji prikazuju žanrove i autore iz baze podataka te od NumericUpDown koji služi za unos broja knjiga.



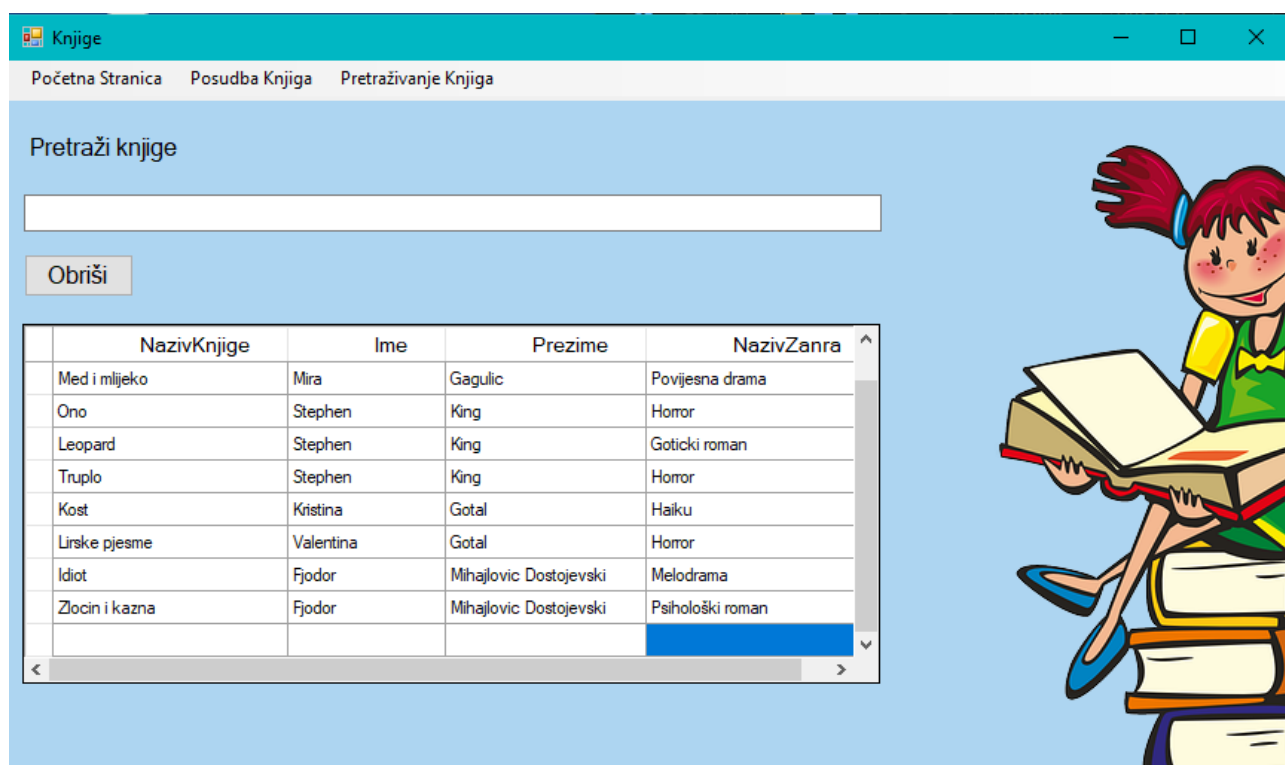
Slika 3.8: Prikaz ComboBoxa „Žanr“

Na slici 3.7 prikazan je ComboBox „Žanr“ s ponuđenim žanrovima iz baze podataka.



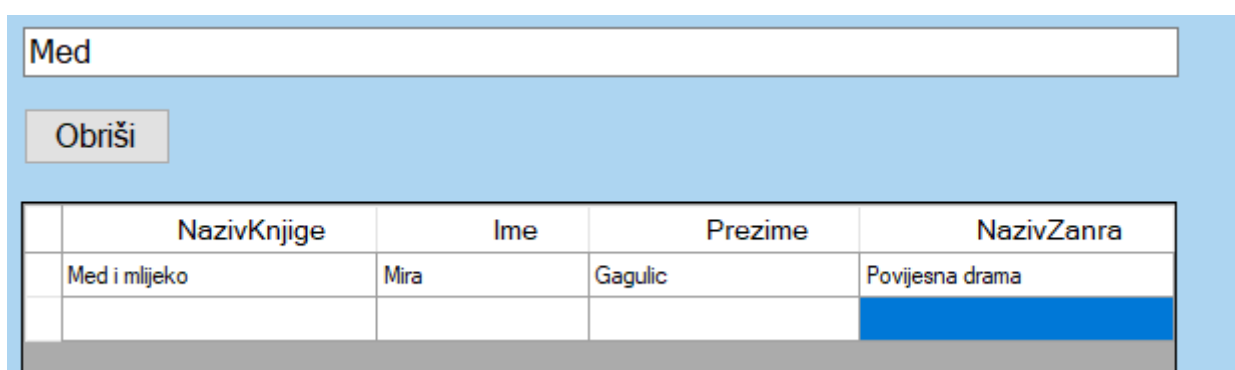
3.9: Prikaz grafičkog sučelja „Posudba Knjiga“

Na slici 3.8 prikazana je forma „Posudba Knjiga“ koja se sastoji od ImageBoxa koji služi za skeniranje QR koda, pet dugmadi (*engl. buttons*) dva su odmah vidljiva: „Skeniraj“ koje pokreće kameru i gumb „Provjeri“, a „Plati“, „Vrati“ i „Posudi“ postaju vidljivi nakon što pritisnemo „Provjeri“. Gumb „Vrati“ će se prikazati ukoliko je knjiga već posuđena, a gumb „Posudi“ će se prikazati, ako se knjiga ne nalazi u tablici „Posudba“ tj. nije posuđena. Gumb „Plati“ se prikazuje zajedno s labelom „Zakasnina: “ i textboxom u kojem se nalazi zakasnina koju korisnik mora platiti. Nakon što korisnik plati zakasninu knjižničar prilikom pritiska na gumb „Plati“ briše posudbu korisnika.



Slika 3.10: Prikaz grafičkog sučelja „Pretraživanje Knjiga“

Na slici 3.10 prikazana je forma „Pretraživanje knjiga“. Ona se sastoji od textboxa pomoću kojega se pretražuju knjige u knjižnici, buttona „Obriši“ koji briše unos u textbox i GridData objekta u kojem se prikazuju knjige knjižnice ELPROS. Unosom teksta u textbox DataGrid će prikazati knjige slične nazivom knjige i autora.



3.11: Prikaz pretraživanja knjiga

Na slici 3.11 vidi se princip rada textbox elementa.

3.2. Učitavanje slike s kamere

Emgu CV je biblioteka koja nam omogućuje dohvaćanje slike s kamere pomoću objekta klase *VideoCapture* koja definira izvor slike. U ovom slučaju riječ je o kameri računala ili laptopa.

```
private VideoCapture capture;  
private bool captureInProgress = false;
```

Slika 3.12: Globalne varijable kamere

Na slici 3.10. prikazane su globalne varijable koje su korištene za učitavanje slike s kamere. Prije nastavka potrebno je objasniti pojmove: varijabla, atribut, metoda i objekt. Varijabla je prostor za skladištenje vrijednosti pomoću koje pozivamo podatke. Ona može biti privatna (*engl. private*), javna (*engl. public*) ili zaštićena (*engl. protected*). Privatne varijable se mogu dohvatiti samo unutar klase u kojoj su deklarirane i mogu se koristiti samo u njenim metodama dok javna varijabla označava varijablu koju mogu pozivati metode svih klasa. Zaštićene varijable su dostupne samo klasi u kojoj su deklarirane i podklasama te klase. Klasa predstavlja tip podatka, odnosno objekta iz stvarnog svijeta. Klase sadrži različita stanja i ponašanja određenih objekata (kopija), koji su stvoreni iz klase (razreda). Objekt je instanca klase. Klasa je nacrt koji definira metode i attribute. Metoda opisuje kako se objekt ponaša, dok atribut njegov izgled.

```

1 reference
private void btnStart_Click(object sender, EventArgs e)
{
    if (capture == null)
    {
        capture = new VideoCapture(0);
    }
    if (capture != null)
    {
        if (captureInProgress)
        {
            Application.Idle -= ProcessFrame;
        }
        else
        {
            Application.Idle += ProcessFrame;
        }
    }
}

```

Slika 3.13: Metoda koja pokreće kameru

Na slici 3.13. prikazana je metoda `btnStart_Click()` koja se poziva klikom na gumb (*engl. button*) Skeniraj u grafičkom sučelju. Kako bi razumjeli metodu `btnStar_Click()` u nastavku slijedi objašnjenje svake linije koda.

Pri kliku na gumb Skeniraj, prvo se provjerava je li kamera uključena. Ako nije, uključuje se. `Capture==null` provjerava je li varijabli pridružen objekt, ako nije stvara se novi objekt klase `VideoCapture`, što je vidljivo na slici 3.13. Parametar (broj) u zagradi označava broj kamere. U `VideoCapture(0)` 0 predstavlja ugrađenu kameru s računala, a ako bi spojili dodatnu kameru, ona bi dobila broj 1 i tako dalje. Na taj način omogućeno spajanje više kamera na isto računala kao i rad s njima.

```

if (capture == null)
{
    capture = new VideoCapture(0);
}

```

Slika 3.14: Parametar (broj) u zagradi

Nakon provjere i uključivanja kamere, provjerava se snima li kamera. Atribut *captureInProgress* omogućava provjeru uvjeta jer je on logičkog tipa (engl. boolean). Logički tip ima dva stanja: 0 i 1. Stanje 0 označava laž (engl. false), a stanje 1 označava istinu (engl. true). U ovom programu 0 označava ugašenu kameru, a 1 označava kameru koja je uključena tj. snima. Funkcija *ProcessFrame()*, koja će biti detaljnije objašnjena u idućem poglavlju, dekodira QR kod i prenosi ono što će se dalje događati sa snimljenom slikom. *Application.Idle* je događaj (engl. event) koji obradi okvir (engl. frame) i zaustavi rad kamere. *ProcessFrame* mu javlja treba li zaustaviti *Application.Idle -= ProcessFrame* ili nastaviti *Application.Idle += ProcessFrame* snimati. Ukoliko je odgovarajući okvir uhvaćen putem kamere, funkcija se izvršila uspješno i učitana je slika s kamere. Na ovaj način, mogu se neprestano dohvaćati novi okviri s kamere i pritom će se svaki put izvršiti funkcija *ProcessFrame()*.

```
if (captureInProgress)
{
    Application.Idle -= ProcessFrame;
}
else
{
    Application.Idle += ProcessFrame;
}
```

Slika 3.15: Prikaz provjere ugašene ili upaljene kamere

3.3. Dekodiranje QR kodova

Spomenuta funkcija *ProcessFrame()*, koja će ovdje biti detaljno objašnjena pri anali dekodiranja QR kodova pomoću ZXing biblioteke.

```
2 references
private void ProcessFrame(object sender, EventArgs arg)
{
    Image<Bgr, byte> colImage = capture.QueryFrame().ToImage<Bgr, byte>();
    ibCamera.Image = colImage;
    //Kod za čitanje QR kdoa
    Bitmap BmpInput = colImage.ToBitmap();
    IBarcodeReader reader = new BarcodeReader(); // create a barcode reader instance
    var barcodeBitmap = BmpInput; // load a bitmap
    var result = reader.Decode(barcodeBitmap); // detect and decode the barcode inside the bitmap
}
```

Slika 3.16: Funkcija ProcessFrame()

Na slici 3.16 prikazan je kod za dekodiranje slike QR koda. Nakon što se slika učitava s kamere, dekodira se kako bi se znalo koji podatak će biti zapisan u bazu podataka. ZXing biblioteka pruža mogućnost dekodiranja slike u boji i crno bijelo. Klasi Image koja je pozvana iz ZXing biblioteke, dodjeljuju se dva parametra: dubina i boja. Za boju je izabran BGR (engl. *Blue, Green, Red*) što označava sliku u boji, a za dubinu odabran je bajt (engl. *byte*) jer je svaki piksel na slici predstavljen s jednim bajtom za svaku boju. Image<Bgr,Byte> stvara praznu sliku u koju se sprema uhvaćena slika s kamere.

```
Image<Bgr, byte> colImage = capture.QueryFrame().ToImage<Bgr, byte>();
ibCamera.Image = colImage;
```

Slika 3.17: Kod koji omogućuje prikaz slike

Pomoću metode klase *VideoCapture* točnije metode *QueryFrame()* dohvaća se slika. Metoda *ToImage<Bgr, byte>* prebacuje sliku u tip podatka koji odgovara klasi Image. *ibCamera.Image* omogućuje prikaz slike u prozoru za slike (engl. *ImageBox*), gdje je prozor za slike nazvan *ibCamera*.

Klasa `Bitmap` omogućuje raspoznavanje i snimanje QR koda u prihvatljivom obliku računalu. Bitna mapa (*engl. Bitmap*) je format datoteke slike koji se može koristiti za stvaranje i pohranu računalne grafike. Ona se u osnovi sastoji od malih točkica u obrascu koji stvaraju cijelu sliku kada su gledali iz daleka. Kako bi uopće bilo moguće stvoriti bitnu sliku, slika s kamere se razbija na najmanju moguću jedinicu, piksel. Ključne karakteristike bitne mape su broj piksela i količina informacija sadržana u svakom kvadratnom okviru mreže (piksela). [6]

```
Bitmap BmpInput = collImage.ToBitmap();
IBarcodeReader reader = new BarcodeReader();
var barcodeBitmap = BmpInput;
var result = reader.Decode(barcodeBitmap);
```

Slika 3.18: Kod za dekodiranje QR koda

Na slici 3.18. prikazan je kod za dekodiranje QR koda. Snimljena slika `collImage` kamerom se pretvara u bitnu sliku pomoću metode `ToBitmap()` te se sprema u objekt klase `Bitmap`. `IBarcodeReader` je sučelje koje omogućava dekodiranje bitnih slika. Bitna slika se sprema u novu varijablu tipa `var`. `Var` je varijabla kojoj kopmajler pridružuje tip podataka. U varijablu `result` sprema se rezultat dekodiranja slike, tj. QR koda. Rezultat je dobiven tako što se na varijablu `reader` poziva metoda `Decode()` s parametrom `barcodeBitmap` koji predstavlja spremljenu bitnu sliku.

3.4. Zapis podataka u bazu podataka

Kako bi se omogućilo spremanje podataka u prethodno kreiranu bazu podataka potrebno je uspostaviti vezu sa poslužiteljem. Nakon što je poslužitelj uključen i spojen u *Microsoft SQL Server Managment Studiju* na kreiranu bazu podataka, uključuje se biblioteka *System.Data.SqlClient* u kojoj se nalazi metoda *Open()* koja omogućuje stvaranje veze između *Microsoft SQL Server Managment Studia* i *Visual Studia*.

```
private void BtnAddAutor_Click(object sender, EventArgs e)
{
    SqlConnection connection = new SqlConnection(connectionString);
    connection.Open();
    if (connection.State == System.Data.ConnectionState.Open)
    {
        if (!String.IsNullOrEmpty(txtUnosImenaAutora.Text) && !String.IsNullOrEmpty(txtUnosPrezimeAutora.Text))
        {
            string query = "INSERT INTO Autor(IDAutor, Ime, Prezime) VALUES ('" + Guid.NewGuid() + "', '" + txtUnosImenaAutora.Text.ToString() + "', '" + txtUnosPrezimeAutora.Text.ToString() + "')";
            SqlCommand cmd = new SqlCommand(query, connection);
            cmd.ExecuteNonQuery();
            MessageBox.Show("New row successfully added to Autor table.");
        }
        else
        {
            lblUpozorenjeAutor.Visible = true;
        }
    }
}
```

Slika 3.19: Prikaz koda za unos novih autora

Na slici 3.19 prikazan je kod za unos novih autora. Prije nego što se podatak unese u bazu podataka mora zadovoljiti dva uvjeta, a to su da je uspostavljena veza s bazom podataka i da je korisnik unio podatke koji će biti uneseni u bazu podataka.

Na isti način se unose podaci za nove žanrove, korisnike i knjige.

```
2 references
private TimeSpan RacunanjeZakasnine()
{
    using (SqlConnection cnn = Helper.ConnectToDB())
    {
        cnn.Open();
        using (var datumV = new SqlCommand("SELECT DatumVracanja FROM Posudba where JMBAG='" + cmbKorisnik.SelectedValue.ToString() + "' ", cnn))
        {
            string iDate = datumV.ExecuteScalar().ToString();
            Console.WriteLine(iDate);
            DateTime DatumVracanja = Convert.ToDateTime(iDate);
            return DateTime.Now.Subtract(DatumVracanja);
        }
    }
}
```

Slika 3.20: Računanje zakasnine

Na slici 3.20. prikazan je kod za računanje zakasnine. Kreirana je privatna metoda tipa

TimeSpan naziva *RacunanjeZakasnine*. Za računanje zakasnine smo dohvatili podataka datum vraćanja iz baze podataka koji je za četrnaest dana veći nego *DatumPosudbe*.

```
DateTime DatumVracanja = DateTime.Today.AddDays(+14);
```

Slika 3.21: Prikaz dobivanja vrijednosti „DatumVracanja“

Metoda *RacunanjeZakasnine()* vraća vrijednost *DateTime.Now.Subtract(DatumVracanja)*. Drugim riječima, oduzima današnji datum od datuma vraćanja. Ukoliko je broj pozitivan (veći od nula). Korisnik će dobiti *MessageBox* da mora platiti zakasninu i pojaviti će se u textboxu cijena zakanine te ako korisnik plati zakasninu na gumb „Plati“ će se posudba obrisati iz baze podataka. Ako korisnik vrati knjigu u roku od dva tjedna prilikom vraćanja knjige pojaviti će se samo *MessageBox* sa porukom: „Podaci obrisani. Nema zakasnine.“

```
private void btnVrati_Click(object sender, EventArgs e)
{
    using (SqlConnection cnn = Helper.ConnectToDB())
    {
        cnn.Open();
        using (SqlCommand Brisanje = new SqlCommand("DELETE Posudba Where JMBAG = " + cmbKorisnik.SelectedValue.ToString() + " AND IDKnjige = " + textBox2.Text + " ", cnn))
        {
            txtZakasnina.Text = Convert.ToInt32(RacunanjeZakasnine().TotalDays).ToString();
            if (Int32.Parse(txtZakasnina.Text) <= 0)
            {
                MessageBox.Show("Podaci obrisani. Nema zakasnine");
                Brisanje.ExecuteNonQuery();
            }
            else
            {
                string x = RacunanjeZakasnine().Minutes.ToString();
                MessageBox.Show($"Korisnik mora platiti zakasninu.");
                btnPlati.Visible = true;
                label3.Visible = true;
                txtZakasnina.Visible = true;
            }
        }
    }
}
```

Slika 3.22: Kod za vraćanje knjige

```
using (SqlConnection cnn = Helper.ConnectToDB())
{
    cnn.Open();
    using (SqlCommand Brisanje = new SqlCommand("DELETE Posudba Where JMBAG = " + cmbKorisnik.SelectedValue.ToString() + " AND IDKnjige = " + textBox2.Text + " ", cnn))
    {
        Brisanje.ExecuteNonQuery();
    }
}
```

3.23: Dio koda gdje se izvršava brisanje podataka iz baze podataka

Na slici 3.23. prikazan je kod koji vrši operaciju nad bazom podataka. Objekt klase *SqlCommand* omogućuje izbor interakcije koju želimo izvršiti nad bazom. U ovom slučaju će podaci biti obrisani, a moguće je dodavati, brisati te mijenjati redove u bazi podataka. *SqlCommand* u sebi sadrži naredbu DELETE koja briše podatke iz baze podataka. Objekt klase *SqlConnection* služi za uspostavljanje veze s bazom podataka. *ExecuteNonQuery* objekta *Brisanje* služi za brisanje podataka iz baze podataka. Nakon izvršavanja metode *ExecuteNonQuery()* podaci se brišu iz baze podataka.

4. ZAKLJUČAK

U današnje vrijeme tehnologija izrazito brzo napreduje kao i količina podataka postaje sve veća te otežava odabir željenih podataka.

Razlog izrade ove aplikacije je mogućnost lakšeg pronalaska željenih knjiga prilikom dolaska u knjižnicu gdje će korisnik putem svog omiljenog autora i naziva knjige pronaći željenu knjigu uz koju će pisati i žanr.

Aplikacija je izrađena u C# programskom jeziku radi njegovog lakog povezivanja s bazom podataka i izrade forme. Baza podataka je izrađena pomoću Microsoft SQL Servera, a neophodne biblioteke za rad s QR kodovima su Zxing i Emgu CV.

Iako je aplikacija zadovoljila postavljene ciljeve, ima mjesta za daljnji razvoj. Moglo bi se, između ostaloga dodati mogućnost pohrane komentara i ocjena putem mobilne aplikacije gdje će korisnici napisati kritički stav prema knjizi i dati ocjenu. Također, mogućnost predlaganja knjige koje bi se mogle nabaviti u knjižnici.

LITERATURA

- [1.]“Overview of Visual Studio | Microsoft Docs.” Accessed May 18, 2020. <https://docs.microsoft.com/en-us/visualstudio/get-started/visual-studio-ide?view=vs-2019>.
- [2.]“Imenskom Prostoru.” Accessed May 18, 2020. <http://www.emgu.com/wiki/files/4.2.0/document/html/8dee1f02-8c8a-4e37-87f4-05e10c39f27d.htm>.
- [3.]“NuGet Galerija | ZXing.Net 0,16.5.” Accessed May 18, 2020. <https://www.nuget.org/packages/ZXing.Net>.
- [4.]“13576.Pdf.” Accessed May 18, 2020. <https://element.hr/artikli/file/1710/baze-podataka/13576>.
- [5.]“What Is a QR Code? | QRcode.Com | DENSO WAVE.” Accessed May 18, 2020. <https://www.qrcode.com/en/about/>.
- [6.]“Final-Report4.Pdf.” Accessed May 18, 2020. <http://spact-center.org/wp-content/uploads/2014/07/Final-Report4.pdf>.

DODATAK – PRILOG

Programski kod

Programski kod je priložen na CD-u uz ovaj rad.