

Desktop aplikacija za evidenciju pohađanja nastave korištenjem kamere

Gotal, Valentina

Undergraduate thesis / Završni rad

2019

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://um.nsk.hr/um:nbn:hr:200:137501>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2025-04-02**

Repository / Repozitorij:

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



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

Sveučilišni studij

**DESKTOP APLIKACIJA ZA EVIDENCIJU POHAĐANJA
NASTAVE KORIŠTENJEM KAMERE**

Završni rad

Valentina Gotal

Osijek, 2019

SADRŽAJ

1. UVOD.....	1
1.1. Zadatak završnog rada.....	1
2. KORIŠTENE TEHNOLOGIJE	2
2.1. Emgu CV	2
2.2. ZXing	3
2.3. QR kodovi.....	3
2.4. Baza podataka	6
3. RAZVOJ APLIKACIJE	9
3.1. Grafičko sučelje	9
3.2. Učitavanje slike s kamere	12
3.3. Dekodiranje QR kodova	15
3.4. Zapis podataka u bazu podataka	16
4. TESTIRANJE	20
4.1. Test uspješnosti skeniranja QR koda putem kamere	20
4.2. Realizacija komponenti aplikacije	20
4.3. Test upisivanja studenata u bazu podataka	21
4.4. Test ispis podataka iz baze podataka u Excel tablicu.....	21
5. ZAKLJUČAK	22
LITERATURA	23
SAŽETAK.....	24
ABSTRACT	25
ŽIVOTPIS.....	26
PRILOZI.....	27

1. UVOD

Uvođenjem Bolonjskog procesa u visokoobrazovni sustav povećao se postotak obavezne dolaznosti studenata na predavanja, što zahtjeva detaljnije praćenje njihove prisutnosti. Velik broj studenata i nedostatak vremena, kod ionako preopterećenog rasporeda, za prozivanja studenata od strane profesora potaknuo je ideju skeniranja studenata putem QR kodova- brzo i efikasno. QR kod (engl. *Quick Response code*) je dvodimenzionalni barkod koji je dizajniran kako bi se omogućilo njegovo brzo dekodiranje. Zamisao je da se svakom studentu dodjeli njegov jedinstven QR kod u kojem je zapisan njegov JMBAG. Desktop aplikacija radit će na principu detektiranja QR koda, njegovog dekodiranja i unosa podataka u bazu podataka. Sama procedura praćenja prisutnosti studenata započinje tako što profesor ili asistent odabire naziv kolegija, oblik nastave i broj sati koji će biti održani. Nakon toga studenti redom prilaze kameri, koja je povezana s računalom na kojem se nalazi aplikacija, pokazujući svoj QR kod. Nakon učitavanja se podaci sa slike automatski spremaju u prethodno kreiranu bazu podataka zajedno s trenutnim datumom. Također, aplikacija ima mogućnost sprječavanja upisivanja istih studenata više puta, provjeru ima li skenirani student pravo na slušanje kolegija te dodatno pruža mogućnost prebacivanja svih podataka o prisutnim studentima u Excel tablicu gdje profesori mogu bez poteškoća provjeriti zadovoljava li student pravo na polaganje određenog kolegija.

1.1.Zadatak završnog rada

Cilj ovog završnog rada je izraditi desktop aplikaciju koja će pomoću kamere moći učitati podatke s QR koda i spremati ih u bazu podataka kako bi se predavaču olakšao unos podataka o studentima koji su prisustvovali nastavi.

2. KORIŠTENE TEHNOLOGIJE

Desktop aplikacija će biti izrađena u Microsoft Visual Studiju. Microsoft Visual Studio je integrirano razvojno okruženje, koje podržava različite programske jezike kao što su C#, C++, Python, JavaScript i druge. Visual Studio je integriran s Microsoft SQL Serverom što omogućava povezivanje aplikacije s bazom podataka, koja će biti kreirana u SQL Server Management Studiju. SQL Server Management Studio je jedan od alata Microsoft SQL Servera. C# programski jezik je najpogodniji za rad s .NET Framework-om i njegovim API-jem te omogućava rad s bazom podataka. Svrha ove aplikacije je detektiranje QR koda putem kamere i spremanje dekodiranih podataka u bazu podataka. Kako bi bilo moguće dekodirati tekst na QR kodu potrebno je koristiti odgovarajuće biblioteke. U ovom završnom radu koristit će se *Emgu CV* biblioteka za računalnu obradu slike i *ZXing* biblioteka za obradu barkodova, o njima će detaljnije biti govora u nastavku.

2.1. Emgu CV

Emgu CV je biblioteka za računalnu obradu slike na različitim platformama, koja je pisana u čistom C#. Razlikuje se od ostalih sličnih biblioteka po tome što ne koristi neprovjereni kod (engl. *unsafe code*). Kako bi se razumjelo što je točno *Emgu CV*, važno se prvo upoznati s *OpenCV-om*. *OpenCV* (engl. *Open Source Computer Vision*) je biblioteka koja se koristi za računalni vid u stvarnom vremenu. Napisana je u optimiziranom C i C++ jeziku, no za razliku od *Emgu CV* nije pogodna za .NET programe. *Emgu CV* je od velikog značaja jer upravo ona omogućuje pozivanje funkcija iz *OpenCV-a* preko .NET programskih jezika. Može se reći da je *Emgu CV OpenCV* u .NET-u. Zahvaljujući toj biblioteci moguće je pozivati funkcije iz *OpenCV-a* u Visual Studio, koji je .NET okruženje i ne može izravno pozivati funkcije napisane u izvornom C/C++ jeziku. *Emgu CV* je, također, višepatformska biblioteka koja je ugrađena u Mono platformu čime je .NET-u omogućen prelazak na druge operative sustave kao što su Linux ili Mac OS. Također, može se pokrenuti i na bilo kojoj platformi koju Mono podržava, između ostalog iOS, Android, Linux, Windows Phone, Mac OS X. Uz to koristi se kod više različitih programskih jezika kao na primjer C#, C++, VB.NET, IronPython . [1]

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

Slika 2.1. Korištene klase iz *Emgu CV* biblioteke

2.2. ZXing

ZXing je biblioteka koja omogućuje obradu i skeniranje barkodova putem kamere, pa tako i QR kodova. Implementirana je u Javi, ali ima priključke prema drugim jezicima. Unutar te biblioteke nalazi se ZXing.NET priključak koji će se koristiti u ovom završnom radu, jer podržava C# i može se koristiti u Visual Studiju koji je .NET okruženje. ZXing.NET podržava i dekodiranje i generiranje barkodova, kao što su QR kod, PDF417, EAN-13 i slično unutar slike. [2]

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

Slika 2.2. Korištene klase iz ZXing biblioteke

2.3. QR kodovi

QR kod (engl. *Quick Response Code*) je zaštitni znak za svaku vrstu dvodimenzionalnog barkoda (engl. *barcode*). Barkod je strojno čitljiva optička naljepnica koja sadrži podatke o stavci na koju je pričvršćena. QR kod je prvi put dizajniran 1994. godine za automobilsku industriju u Japanu, ali su se brzo počeli koristiti izvan automobilske industrije zbog svoje brze čitljivosti i velikog kapaciteta za pohranu podataka.

U praksi, QR kodovi često sadrže podatke za lokator, identifikator ili tragač koji upućuju na web stranicu ili aplikaciju. QR kod koristi četiri standardizirana načina kodiranja za spremanje podataka, a to su: numerički, alfanumerički, bajt/binarni i kanji (usvojeni logografski kineski znakovi koji se koriste u japanskom sustavu pisanja).[3]

Dvodimenzionalni kodovi mogu sadržavati veću količinu podataka na manjem prostoru u odnosu na jednodimenzionalne kodove. No, QR kodovi mogu sadržavati puno više podataka u usporedbi s ostalim dvodimenzionalnim kodovima. Zahvaljujući naprednoj metodi ispravljanja pogrešaka (engl. *error-correctionn method*) i drugim jedinstvenim karakteristikama QR kodovi su pouzdaniji pri većim brzinama u odnosu na ostale kodove. [4]

Slika 2.3. prikazuje razliku između jednodimenzionalnih i dvodimenzionalnih kodova, točnije QR kodova. QR kodovi, za razliku od jednodimenzionalnih kodova, ne sadrže samo

informacije u horizontalnoj dimenziji nego i u vertikalnoj dimenziji. Podatci su skladišteni po principu raspoređivanja tamnih i svijetlih modula. Svaki tamni i svijetli modul je predstavljen sa 0 ili 1, što ga čini razumljivim računalu.



Slika 2.3. Razlika između QR koda i jednodimenzionalnog koda¹

Moduli QR koda obavljaju nekoliko funkcija. Neki moduli sadrže podatke, dok su drugi grupirani u različite obrasce funkcija (engl. *function patterns*) koji poboljšavaju performanse i omogućuju poravnavanje simbola, ispravljanje pogrešaka i naknadna izobličenja. Vremenski obrazac omogućuje uređaju za skeniranje prepoznavanje veličine simbola. Također, potrebna je i zona bez podataka (engl. *quiet zone*). Ona se sastoji od međuspremnik (engl. *buffer area*) s četiri modula koji ne sadrže podatke, kako bi uređaj za skeniranje mogao razlikovati QR kod od ostatka teksta ili oznaka. [4] Moduli QR koda prikazani su na slici 2.4.



Slika 2.4. Moduli QR koda²

QR kodovi se mogu generirati u četrdeset različitih verzija simbola, od 21 x 21 modula do 177 x 177 modula. Na slici 2.5. prikazana je prva, druga i posljednja četrdeseta verzija simbola QR koda. Svaka iduća verzija ima 4 dodatna modula po strani, što je ukupno 16 dodatnih modula po simbolu i proporcionalno tome može skladištiti veće količine podataka. Maksimalna

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

² <http://www.nacs.org/LinkClick.aspx?fileticket=D1FpVAvvJuo%3D&tabid=1426&mid=4802>

količina podataka koju može sadržavati određeni simbol definirana je verzijom, vrstom znakova i razinom ispravljanja grešaka (engl. *error correction level*).



Slika 2.5. Usporedba verzija QR koda³

Najveći kapacitet za pohranu ima verzija četrdeset s niskom razinom za ispravljanje pogrešaka. Oznaka je 40-L. Na tablici 2.1. prikazana su sva četiri standardizirana načina kodiranja za spremanje podataka pomoću QR koda i njihovim skladištenim kapacitetima pri 40-L.

Tablica 2.1. Maksimalni kapacitet za pohranu znakova (40-L)

<i>Način unosa</i>	Maksimalan broj znakova	Bit / znakovi	Mogući znakovi / zadano kodiranje
<i>Brojevi</i>	7 089	3½	0,1,2,3,4,5,6,7,8,9
<i>Alfanumerički</i>	4 296	5½	0-9, A-Z (samo velika slova), razmak, \$, %, *, +, -, ., /, :
<i>Binarni / bajt</i>	2 953	8	ISO 8859-1
<i>Kanji / kana</i>	1 817	13	Shift JIS X 0208

QR kodovi imaju veliku sposobnost ispravljanja pogrešaka zahvaljujući dodavanju Reed-Solomon-ovih kodova (engl. *Reed-Solomon code*). Reed-Solomon-ovi kodovi su skupina kodova za ispravljanje pogrešaka, koji omogućuju dekodiranje QR kodova koji su prljavi ili oštećeni. Dostupne su četiri razine ispravljanja pogrešaka kao što je vidljivo prema tablici 2.2.

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

Tablica 2.2. Razine ispravljanja pogrešaka

<i>Razina ispravljanja grešaka</i>	<i>Približna iznos korekcije</i>
<i>L</i>	7%
<i>M</i>	15%
<i>Q</i>	25%
<i>H</i>	30%

Prilikom odabira razine ispravljanja pogrešaka, treba uzeti u obzir uvjete okoline, ali i željenu veličinu simbola QR koda. Na primjer, razina Q i H se koriste za tvornice i druge primjene kada se predviđa oštećenje ili vjerojatnost da će QR kod postati prljav. Razina L se odabire za čista okruženja i kodove koji sadrže velike količine podataka. Općenito se najviše koristi M razina.[4]

2.4. Baza podataka

Baza podataka je skup ili kolekcija povezanih podataka. Najjednostavnije je promatrati bazu podataka kao kontejner u koji se mogu spremati ogromne količine podataka. Pod podatkom se podrazumijevaju poznate činjenice koje mogu biti zabilježene i koje imaju implicitno značenje kao npr. telefonski brojevi, imena ili adrese ljudi. [5]

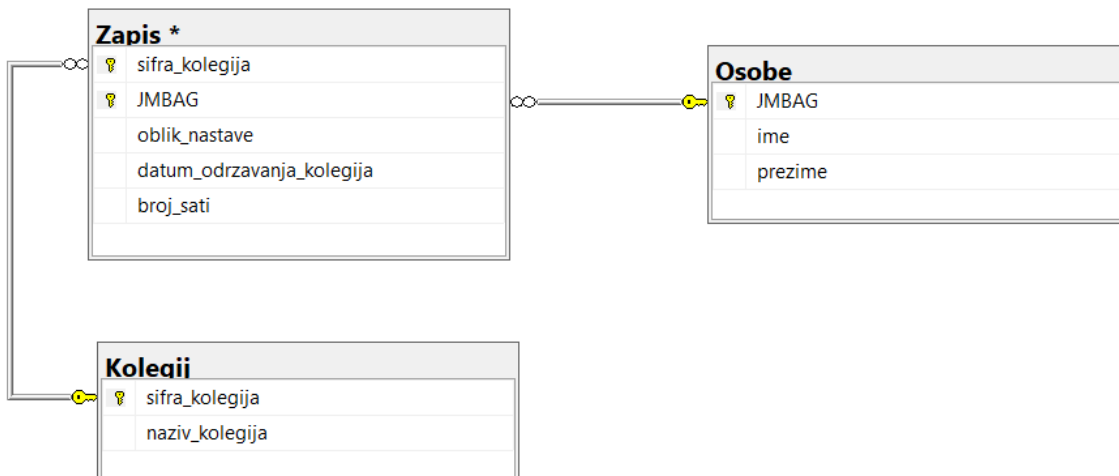
Sustav za upravljanje bazama podataka, SUBP (engl. *Data Base Management System, DBMS*) je zbirka programa koja omogućava korisnicima stvaranje i održavanje baze podataka. To je softverski sustav opće namjene koji olakšava procese definiranja, konstruiranja, manipulacije i dijeljenja baze podataka među različitim korisnicima i aplikacijama. Primarni cilj SUBP-a je pružiti korisnicima okruženja koje je prikladno i učinkovito za spremanje i dohvaćanje podataka. SUBP se može kvalificirati prema različitim kriterijima kao što su: model podataka, broj korisnika, broj web stranica, vrste pristupa i trošak. [5] Nadalje će se govoriti o podjeli SUBP-a po modelu podataka.

SUBP se po modelu podataka dijeli na: hijerarhijske, mrežne, relacijske, relacijske baze s objektno orijentiranim proširenjima i objektno orijentirane baze podataka. Relacijski model SUBP-a je uveo E. F. Codd 1970-e godine i trenutno je najčešće korišten model. Upravo će se zbog svoje raširenosti relacijski model koristiti u ovom radu pri kreiranju baze podataka. Tablice su osnovna struktura podatka u relacijskom modelu. Podatci se pohranjuju u redove tih tablica. [6]

Prije rada s bazom podataka važno je objasniti ključne pojmove. Baza podataka se sastoji od tablica, koje su međusobno povezane vezama. Veze opisuju odnos među tablicama. Svaka tablica se sastoji od dva dijela: redaka (slogova) i stupaca (polja). U stupcima se nalaze entiteti (objekt, pojava, proces) o kojima se čuva informacija u bazi podataka, a u redovima atributi koji pobliže opisuju entitete. Svi atributi mogu biti kandidati za ključ. Važno je odrediti koji atribut ili koji atributi jednoznačno određuju svaki redak u tablici. Takav se atribut, odnosno atributi tada nazivaju primarnim ključ (engl. *primary key*). Primarni ključ se koristi za uspostavljanje relacija između tablica i to tako da dva ili više primarna ključa različitih tablica, koje želimo povezati zajedno, čine strani ključ u tablici koja ih povezuje. Dakle, strani ključ je redak u zajedničkoj tablici s kojim se izvodi povezivanje među različitim tablicama i ne mora biti jedinstven.

Strukturni jezik za pretraživanje ili skraćeno SQL (engl. *Structured Query Language*) koristi se za upravljanje relacijskim bazama podataka i obavljanje različitih operacija nad podacima. Relacijski sustavi, poznati kao SQL baze podataka, sadrže skup tablica koje sadrže podatke u redovima i stupcima. Svaki stupac u tablici odgovara kategoriji podataka, dok svaki red sadrži vrijednost podataka za svaki stupac u kojem se nalazi.

Postoji više vrsta sustava za upravljanje bazama podataka s primarnim relacijskim modelom i koji koriste SQL programski jezik kao što su MySQL, Oracle, Microsoft SQL Server, SQL Lite, Sybase i drugi. Pri izradi ovog rada korišten je Microsoft SQL Server koji sadržava alat (engl. *tool*) Microsoft SQL Server Management Studio. Microsoft SQL Server Management Studio je grafičko korisničko sučelje (engl. *Graphical user interface, GUI*) koje nudi razne mogućnosti kao što su stvaranja nove baze podataka, mijenjanje postojećih shema baze podataka dodavanjem izmjena tablica i indeksa ili analiziranje izvedbe. Microsoft SQL Server Management Studio omogućuje povezivanje i upravljanje SQL serverom iz grafičkog sučelja umjesto korištenja naredbenih linija (engl. *the command line*), čime je olakšano kreiranje baze podataka.



Slika 2.6. Dijagram kreirane baze podataka

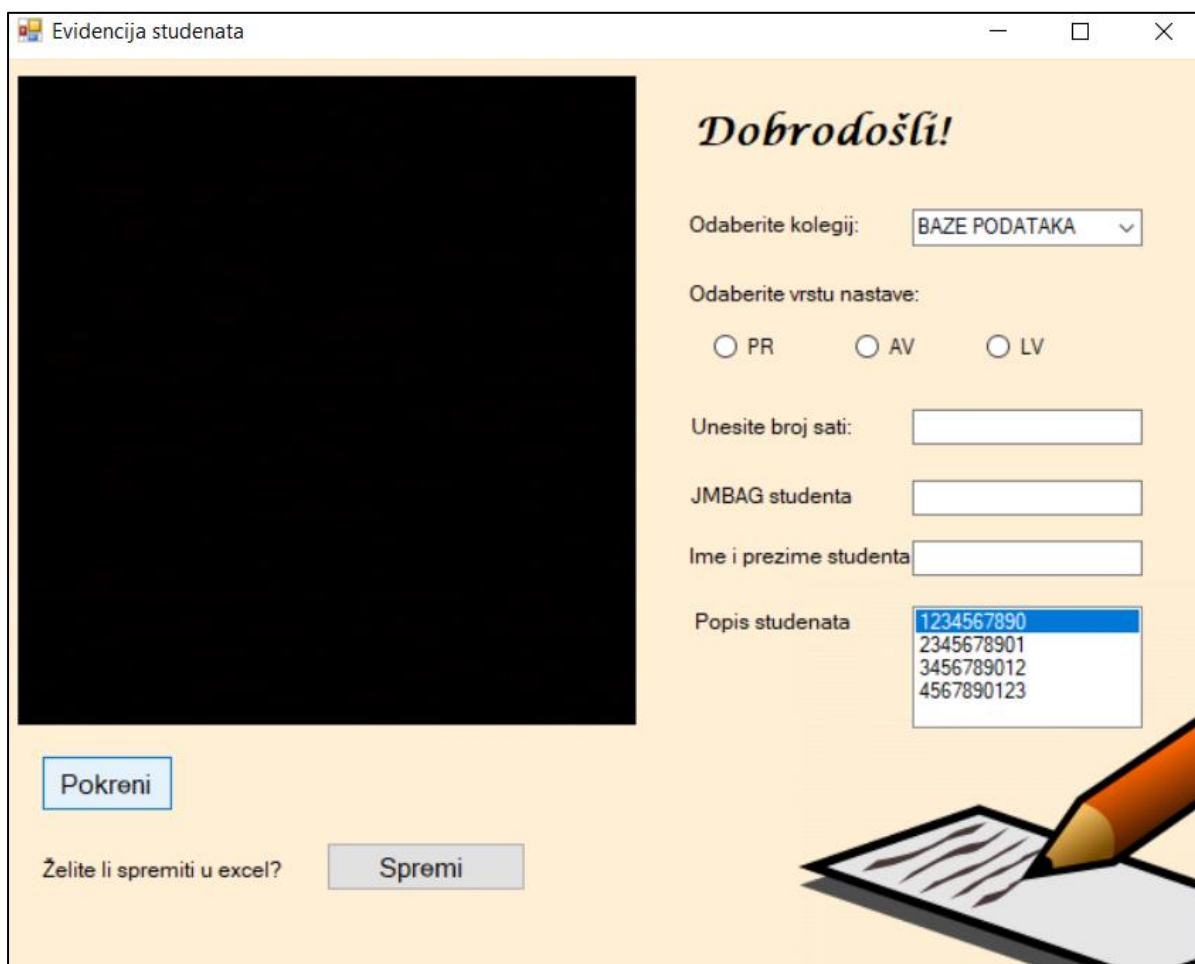
Na slici 2.6. prikazana je kreirana baza podataka. Ona se sastoji od tri tablice: Zapis, Kolegij i Osobe. Tablica Osobe se sastoji od tri atributa: JMBAG koji je ujedno i primarni ključ, ime i prezime. Tablica Kolegij se sastoji od dva atributa sifra_kolegija koji je primarni ključ i naziv_kolegija. Tablica Zapis ima primarni ključ koji se sastoji od dva strana ključa iz tablica Kolegij i Osobe. Uz strane ključeve ima još tri atributa: oblik_nastave, datum_održavanja_kolegija i broj sati.

3. RAZVOJ APLIKACIJE

U ovom poglavlju opisan je način na koji je desktop aplikacija kreirana te njezine funkcionalnosti i dizajn. Naglasak je na ključnim dijelovima koda koji izvršavaju samu svrhu aplikacije.

3.1. Grafičko sučelje

Grafičko sučelje omogućuje interakciju čovjeka s računalom pomoću tipkovnice i miša. Korisnik pritom nije opterećen s programskim dijelom koji stoji iza aplikacije i na taj se način aplikacija može koristiti svaki korisnika bez obzira na njegovo prethodno znanje iz područja programiranja i baza podataka.

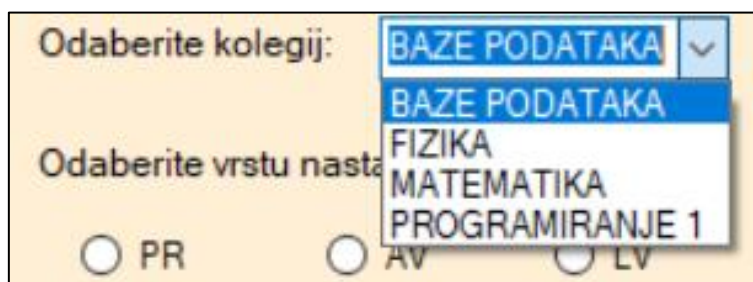


Slika 3.1. Izgled grafičkog sučelja

Slika 3.1. daje prikaz forme pod nazivom „Evidencija studenata“. Ona služi za unos prisutnih studenata u bazu podataka putem QR koda. Klikom na gumb *Pokreni* pokreće se kamera

odnosno metoda `btnStart_Click()`, koja će biti detaljno objašnjena u idućem potpoglavlju. Nakon toga aplikacija je spremna za unos podataka.

Za odabir kolegija korišten je padajući izbornik (engl. *ComboBox*) koji nudi popis kolegija. Popis kolegija se povlači iz baze podataka, točnije tablice Kolegij. U tablici Kolegij ručno su uneseni kolegiji i ti podaci se ispisuju u padajući izbornik.



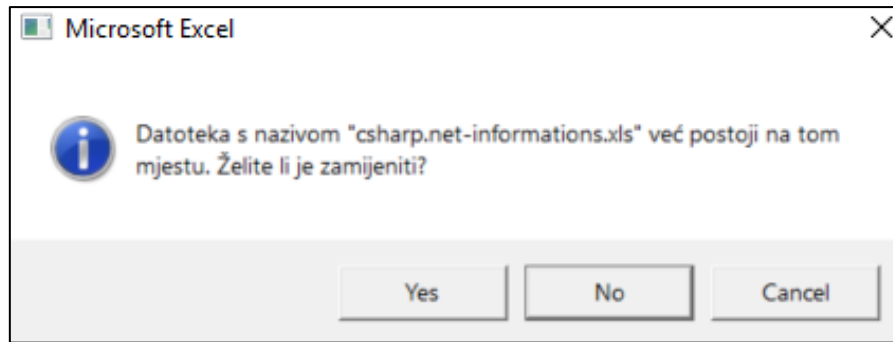
Slika 3.2. Popis kolegija u padajućem izborniku

Nakon odabira kolegija, odabire se vrsta nastave koja se izvodi. Može se odabrati samo jedan oblik nastave, time je onemogućeno krivo unošenje. Svaki odabir je predstavljen jednim radiogumbom (engl. *RadioButton*). Broj sati se unosi ručno, i to radi pojednostavljene i lake mogućnosti promjene ako se planira održati dio predviđenih sati ili neka nadoknada. Nakon što se odabere kolegij, oblik nastave i broj sati slijedi upisivanje studenata. Studenti se upisuju tako što kameri približe svoj jedinstven QR kod, koji može biti isprintan na papiru ili prikazan na zaslonu mobilnog uređaja. Nakon što program učita QR kod, koji se u realnom vremenu prikazuje u aplikaciji putem kamere, JMBAG studenta je vidljiv u tekstualnom polju (engl. *TextBox*) i ime i prezime studenta u tekstualnom polju ispod, što je vidljivo na slici 3.3.. Za povlačenje imena i prezimena studenta potrebno je prvo naredbom *SELECT* pretražiti tablicu Osobe, i to uz pomoć primarnog ključa navedene tablice. Pronalaskom željenog studenta, povlače se njegovo ili njeno ime i prezime. Time se studentu daje sigurnost da je baš on upisan na kolegij, odnosno da nije došlo do greške pri upisivanju. Također, onemogućeno je upisivanje istog studenta više puta zaredom.

Slika 3.3. Prikaz forme nakon što je student upisan

Na slici 3.3. prikazana je lista (engl. *listBox*) koja prikazuje popis studenata, tj. popis njihovih JMBAG-ova koji se povlače iz baze podataka. Svaki student je ručno unesen u bazu podataka, točnije u tablicu Osobe, kako bi se onemogućilo upisivanje studenata koji nisu prijavljeni kao slušači na kolegiju. Način na koji se vrši sama provjera i upis studenta i ostalih podataka u bazu podataka bit će detaljno objašnjen u potpoglavlju 3.4.

Nakon što su studenti uneseni u bazu, profesori imaju mogućnost prijenosa podataka iz baze podataka u Excel tablicu radi lakšeg obrađivanja podataka. Nakon klika na gumb *Spremi*, pojavit će se poruka prikazana na slici 3.4. U Excel tablici ostaju spremljeni podatci od prije i svaki put će aplikacija nuditi mogućnost unošenja novih podataka na već postojeće podatke.



Slika 3.4. Poruka nakon što kliknemo gumb Spremi

Excel tablica je poznata po tome što omogućava laku obradu podataka i izvođenje raznih formula kao što su zbrajanje, sortiranje i slično, što je profesorima i asistentima jako korisno pri obradi podataka i provjeri zadovoljavanja minimalnih uvjeta dolaznosti koji su potrebni za dobivanje prava na polaganje kolegija. Na slici 3.5. vidljivo je kako izgledaju podatci iz baze podataka nakon što su prebačeni u Excel tablicu.

	A	B	C	D	E
1	BAZE PODATAKA	2345678901	LV	19. 8. 2019.	1
2	BAZE PODATAKA	3456789012	PR	19. 8. 2019.	3
3	FIZIKA	3456789012	AV	19. 8. 2019.	3
4	MATEMATIKA	3456789012	AV	18. 8. 2019.	2
5	PROGRAMIRANJE 1	2345678901	LV	19. 8. 2019.	3
6	PROGRAMIRANJE 1	2345678901	LV	19. 8. 2019.	4

Slika 3.5. Ispis podataka u Excel tablicu

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 laptopa ili računala.

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

Slika 3.6. Varijable klase QRCheckIn

Na slici 3.6. prikazane su globalne varijable korištene za potrebe učitavanja slike s kamere. Stoga je prije nastavka potrebno objasniti pojmove varijabla, objekt, atribut i metoda. Varijabla je ime koje služi za pozivanje podataka spremljenih u memoriju. Ona može biti privatna (engl.

private), javna (engl. *public*) ili zaštićena (engl. *protected*). Privatne varijable su one koje 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 gdje su deklarirane i podklasama te klase. Klasa specificira određenu vrstu predmeta, odnosno objekata iz stvarnog svijeta. Predstavlja obrazac koji opisuje različita stanja i ponašanja određenih objekata (kopija), koji su stvoreni iz klase (uzorak). [7] Prema tome, objekt je instanca klase. Klasu je nacrt koji definira metode i atribute. Metoda opisuje kako se objekt ponaša, dok atribut njegov izgled.

```
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 3.7. Metoda *btnStart_Click()*

Na slici 3.7. vidi se metoda *btnStart_Click()* koja se poziva na klik na gumb (engl. *button*) *Start* u grafičkom sučelju. U nastavku se detaljno prolazi kroz svaku liniju koda, kako bi se objasnio prikazani kod.

Pri kliku na gumb *Start*, prvo se provjerava je li kamera uključena. Ako nije, uključuje se. Drugim riječima *capture==null* provjerava je li varijabli pridružen objekt, ako nije, stvara se novi objekt klase *VideoCapture*, što je vidljivo na slici 3.8. Parametar (broj) u zagradi označava broj kamere. U *VideoCapture(0)* 0 predstavlja ugrađenu kameru s laptopa, a ako bi bila spojena

dotatna kamera na USB, ona bi dobila broj 1 i tako dalje. To omogućuje spajanje više kamera na isto računalo i rad s njima.

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

Slika 3.8. Provjera varijable *capture*

Nakon uključivanja kamere, provjerava se snima li kamera. Atribut *captureInProgress* omogućava laku provjeru uvjeta, jer je on logičkog tipa (engl. *boolean*) što je vidljivo na slici 3.8. Logički tip ima dva stanja: 1 i 0. Stanje 1 označava istinu (engl. *true*), a stanje 0 laž (engl. *false*). U ovom programu stanje 0 označava da je kamera ugašena, dok stanje 1 označava da kamera snima. Pomoću funkcije *ProcessFrame()*, koja će detaljno biti objašnjena u idućem potpoglavlju, dekodira se QR kod. U ovom trenutku je važno znati da ona prenosi ono što će se dalje događati sa snimljenom slikom. *Application.Idle* je događaj (engl. *event*) koji obradi okvir (engl. *frame*) i zaustavi kameru, nakon čega čeka da mu funkcija *ProcessFrame()* javi 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. Također, ovim postupkom mogu se neprestano dohvaćati novi okviri s kamere i pritom se svaki put izvršava funkcija *ProcessFrame()*.

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

Slika 3.9. Provjera uvjeta varijable *captureInProgress*

3.3. Dekodiranje QR kodova

U prošlom potpoglavlju spomenuta je funkcija `ProcessFrame()`, koja će ovdje biti detaljno objašnjena pri analizi dekodiranja QR kodova pomoću `ZXing` biblioteke.

```
private void ProcessFrame(object sender, EventArgs arg)
{
    Image<Bgr, byte> colImage = capture.QueryFrame().ToImage<Bgr, byte>();
    ibCamera.Image = colImage;
    Bitmap BmpInput = colImage.ToBitmap();
    IBarcodeReader reader = new BarcodeReader();
    var barcodeBitmap = BmpInput;
    var result = reader.Decode(barcodeBitmap);
}
```

Slika 3.10. Funkcija `ProcessFrame()`

Na slici 3.10. prikazan je kod za dekodiranje slike QR koda. Nakon što se slika učita 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. Prije svega klasi `Image`, koja je pozvana iz `ZXing` biblioteke, se dodjeljuju dva parametra: boja i dubina. Time se kreira objekt te klase. Za boju je izabran `BGR` (engl. *Blue, Green, Red*) što označava sliku u boji, dok je za dubinu odabran bajt (engl. *byte*) jer je svaki piksel na slici predstavljen s jednim bajtom za svaku boju. Drugim riječima, `Image<Bgr,Byte>` omogućava stvaranje prazne slike u koju se sprema uhvaćena slika s kamere. Ako je potrebno spremiti crno-bijelu sliku, koristi se `Image<Gray, Byte>`.

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

Slika 3.11. Kod koji omogućuje prikaz slike

U prošlom potpoglavlju spomenut je objekt klase `VideoCapture`, u kojem se sprema slika s kamere. Pomoću metode te klase, točnije metode `QueryFrame()` dohvaća se slika. Metodom `ToImage<Bgr, byte>()` prebacuje se slika u tip podatka koji odgovara klasi `Image`. `ibCamera, Image` omogućuje prikazivanje slike u prozoru za slike (engl. `ImageBox`), gdje je prozor za slike nazvan `ibCamera`.

Kako bi se QR kod mogao raspoznati i snimiti za računalo prihvatljiv oblik, koristi se klasa *Bitmap*. Bitna mapa (engl. *Bitmap*) je format datoteke slike koji se može koristiti za stvaranje i pohranu računalne grafike. Najjednostavnije rečeno bitna mapa opisuje vrstu slike s kojom se korisnici web stranica konstantno susreću, iako toga nisu svjesni. Bitna mapa se u osnovi sastoji od malih točkica u obrascu koji stvaraju cijelu sliku kada su gledani iz daleka. Takav format slike predstavlja sliku kao mrežu sastavljenu od redaka i stupaca gdje određena ćelija predstavlja vrijednost koja ju ispunjava ili ostavlja praznom, stvarajući tako sliku iz podataka. Kako bi uopće bilo moguće stvoriti bitnu sliku, slika s kamere se razbija na najmanju moguću jedinicu, piksel. Nakon toga se informacija o dubini boje svakog piksela sprema u bitove koji su preslikani u stupce i redove. To može biti vidljivo kada korisnik pokuša postepeno povećati bitnu sliku, jer do izražaja počnu dolaziti pikseli. Ključne karakteristike bitne mape su broj piksela (ili kvadrata u mreži) i količina informacija sadržana u svakom kvadratnom okviru mreže (piksela).[8]

Na slici 3.12. prikazan je kod za dekodiranje QR koda. Kamerom snimljena slika *colImage* se pomoću metode *ToBitmap()* pretvara u bitnu sliku. Novostvorena bitna slika sprema se u objekt klase *Bitmap*. *IBarcodeReader* je sučelje koje omogućava dekodiranje bitnih slika. Bitna slika se sprema u novu varijablu tipa *var*. *Var* je implicitnog tipa što znači da joj kompajler 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.

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

Slika 3.12. Kod za dekodiranje QR koda

3.4. Zapis podataka u bazu podataka

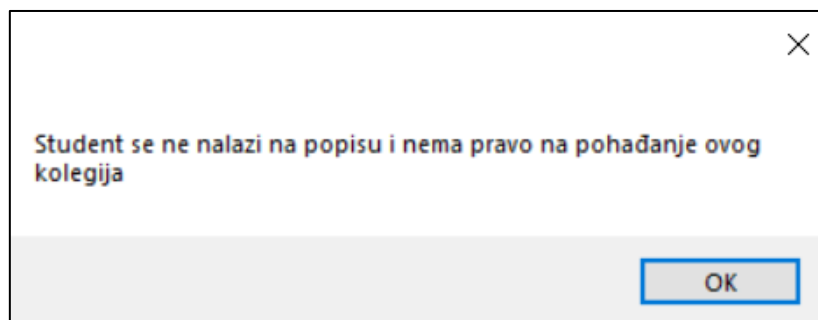
Prije nego što se podatci mogu spremiti u prethodno kreiranu bazu podataka, treba se uspostaviti veza sa poslužiteljem. Nakon što je poslužitelj uključen i spojen u Microsoft SQL Server Management Studiju na kreiranu bazu podataka, uključuje se biblioteka *System.Data.SqlClient* čija metoda *Open()* omogućuje stvaranje veze između Microsoft SQL Server Management

Studia i Visual Studia. Nakon što je QR kod dekodiran i varijabla *result* poprimi vrijednost, vrši se provjera nalazi li se taj student u bazi podataka.

```
public bool provjera(string textBox2)
{
    foreach (string stari_student in student)
    {
        if (String.Equals(textBox2, stari_student))
        {
            return true;
        }
    }
    return false;
}
```

Slika 3.13. Kod funkcije provjera()

U foreach petlji provjerava se nalazi li se skenirani student na popisu studenta koji imaju pravo na pohađanje tog kolegija, što je vidljivo na slici 3.13. Ako se nalazi, vraća se *bool* vrijednost *true* te se izvršava metoda *Unesi()*. Ukoliko JMBAG studenta nije na popisu, pojavljuje se poruka prikazana na slici 3.14.



Slika 3.14. Poruka koja se javlja prilikom upisa studenta koji nije na popisu

Na slici 3.15. je prikazana metoda *Unesi()* kojom se vrši unos podatka u tablicu Zapis. Metoda kao parametar prima vrijednost koja se nalazi u *textBox2*. Pomoću *IspisKolegija.SelectedValue* dohvaća se kolegij koji je odabran, a pomoću metode *dohvati_vrstu_nastave()* koji oblik nastave je označen, dok se u *textBox2* sprema učitani JMBAG. U *textBox1* profesor ili asistent unose broj sati koji će biti održani. Zbog jednostavnosti u aplikaciji se kao datum uzima

trenutni datum, ali ako je potrebno vrlo lako se može proširiti i uvesti ručni unos datuma. Metoda *Unesi()* kao povratnu vrijednost vraća upit u koji je spremljena naredba za unos podataka.

```
public string Unesi(string textBox2)
{
    DateTime dateAndTime = DateTime.Now;
    string datum=dateAndTime.ToString("yyyy/MM/dd");
    string statement = "INSERT INTO Zapis VALUES('"+IspisKolegija.Selected.Value + "', " +
        "'" + textBox2 + "', '" + dohvati_vrstu_nastave() + "', '" + datum + "', " +
        "'"+textBox1.Text+"') ";
    return statement;
}
```

Slika 3.15. Metoda *Unesi()*

Sam unos podataka u bazu podataka će se dogoditi nakon što se provjeri nalazi se skenirani student na popisu studenata koji slušaju kolegij, odnosno kada metoda *provjera()* vrati *true*.

```
string statement = Unesi(textBox2.Text);
SqlCommand cmd = new SqlCommand(statement, conn);
cmd.ExecuteNonQuery();
```

Slika 3.16. Dio koda gdje se izvršava unos podataka u bazu podataka

Na slici 3.16. vidljiv je dio koda koji vrši samu operaciju nad bazom podataka i bez kojeg ne bi bilo moguće izvršiti niti jednu naredbu nad bazom. Objekt klase *SqlCommand* omogućuje izbor interakcije koju želimo izvršiti nad bazom. Na primjer, između ostalog moguće je dodavati, brisati, mijenjati redove u bazi podataka. U kodu na slici 3.16. kreiran je objekt klase *SqlCommand*. Kao parametri pri njegovom kreiranju navedeni su *statement*, koji u sebi sadrži naredbu INSERT, i referenca na objekt klase *SqlConnection*, koji služi za uspostavljanje veze s bazom podataka. Za samo umetanje podataka u bazu podataka koristi se metoda *ExecuteNonQuery()* objekta *cmd*. Nakon izvršavanja metode *ExecuteNonQuery()* podatci se upisuju u bazu podatak, što je vidljivo na slici 3.17.

Results		Messages			
	sifra_kolegija	JMBAG	oblik_nastave	datum_odrza...	broj_sati
1	BAZE PODATAKA	2345678901	LV	2019-08-19 ...	1
2	BAZE PODATAKA	3456789012	PR	2019-08-19 ...	3
3	FIZIKA	3456789012	AV	2019-08-19 ...	3
4	MATEMATIKA	3456789012	AV	2019-08-18 ...	2
5	PROGRAMIRANJE 1	2345678901	LV	2019-08-19 ...	3

Slika 3.17. Upisani podatci u bazu podataka

Zbog strukture baze podataka prikazane ranije na slici 2.6. sada je jednostavno dobiti i detaljnije podatke o svakom zapisu, kao što su ime i prezime studenata te puni naziv kolegija kao što je prikazano na slici 3.18.

Results		Messages					
	ime	prezime	JMBAG	oblik_nastave	naziv_kolegija	datum_odrza...	broj_sati
1	Pero	Peric	1234567890	PR	BAZE PODATAKA	2019-09-03 ...	3
2	Franjo	Franjic	4567890123	AV	FIZIKA	2019-09-03 ...	2
3	Klaudija	Lovric	3456789012	LV	PROGRAMIRANJE	2019-09-03 ...	2
4	Lucija	Stipic	2345678901	AV	PROGRAMIRANJE	2019-09-03 ...	1
5	Klaudija	Lovric	3456789012	LV	BAZE PODATAKA	2019-09-03 ...	2

Slika 3.18. Pogled na bazu podataka

4. TESTIRANJE

Kako bi se izvršila provjera uspješnosti same aplikacije, napravljeno je testiranje. Testiranje je proces pokazivanja odsustva pogrešaka i ono se sastoji od provjere više stavaka: uspješnosti skeniranja QR koda putem kamere, realizacije samih komponenti i njihovih funkcionalnosti, upisa studenata u bazu podataka i ispisa studenata u Excel tablicu. Svaka od tih stavaka će biti zasebno opisana u nastavku.

4.1. Test uspješnosti skeniranja QR koda putem kamere

Prva stavka koja se ispitivala je vremenska razlika između skeniranja QR koda koji je isprintan na papiru i onog koji se prikazuje na zaslonu mobilnog uređaja. Nakon provedenog testiranja, dokazano je da kamera lakše detektira QR kod na papiru, gotovo odmah čim se približi kameri, dok je s mobilnim uređajem teže detektira i ovisi o više parametara, kao što su svjetlina na mobitelu, refleksija svjetlosti na zaslonu mobilnog uređaja, kut držanja mobitela i slično. U idealnim slučajevima kada je svjetlina mobitela pojačana na najjače i svjetlost u prostoriji ne udara direktno na zaslon mobilnog uređaja, razlika u brzini skeniranja između QR koda putem zaslona i onog na papiru je neprimjetna. Tome u korist ide i činjenica da je QR kod čitljiv iz 360 ° zahvaljujući obrascu za otkrivanje položaja koji se nalazi u tri ugla simbola. Obrasci za otkrivanje položaja osiguravaju stabilno očitavanje velike brzine bez obzira na pozadinske smetnje.[9]

4.2. Realizacija komponenti aplikacije

Druga stavka koja je testirana je uspješnost svake komponente aplikacije. Prilikom klika na gumb *Pokreni*, pokreće se aplikacija, bez ikakvih poteškoća, pri tome automatski povlačeći sve podatke iz baze podataka u padajući izbornik i listu studenata. Uspješno je realizirana i stavka odabira vrste nastave gdje se može odabrati samo jedan oblik nastave. Tekstualno polje u koje se unosi broj sati uzima upisan podatak i prepisuje ga u bazu podataka. Unos krivog znaka ili vrijednost ne predstavlja problem, jer se svaki podatak može promijeniti u Excel tablici. Skenirani QR kod se prikazuje u tekstualnom polju kraj natpisa JMBAG studenta gdje je vidljivo kada je QR kod skeniran. Kada se sve sumira, svaka komponenta grafičkog sučelja je zadovoljila test.

4.3. Test upisivanja studenata u bazu podataka

Treća stavka je provjera uspješnog upisivanja studenata u bazu podataka. Ako se student nalazi na popisu studenata u bazi podataka, nakon što se njegov QR skenira, pojavit će se njegovo ime i prezime u tekstualnom polju kraj natpisa *Ime i prezime student*. Ime i prezime studenta povlače se iz baze podataka radi olakšavanja same provjere studenta da je baš on upisan, zbog činjenice da mali postotak studenata zna svoj JMBAG napamet. Ukoliko se student ne nalazi na popisu studenata u bazi podataka, pojavit će se poruka da student nema pravo na slušanje kolegija te je time eliminirana mogućnost upisivanja studenata bez prava na slušanje kolegija. Onemogućeno je i upisivanje istog studenta više puta zaredom. Pozivom naredbe *SELECT * FROM Zapis*; u Microsoft SQL Server Managment Studiju dobije se popis svih unesenih podataka u tablicu Zapis i prema tome je vidljiva ispravnost metode za upisivanje studenata.

4.4. Test ispis podataka iz baze podataka u Excel tablicu

Zadnji korak u samom testiranju, ali i radu aplikacije je provjera uspješnog prepisivanja podataka iz baze podataka u Excel tablicu. Prilikom klika na gumb *Spremi*, pojavljuje se poruka s upitom želimo li izmijeniti upisane podatke u tablici i nakon potvrdnog odabira u mapi se pojavljuje Excel tablica gdje su vidljivi svi podatci koji se nalaze i u bazi podataka. Usporedbom podataka u bazi podataka i u Excel tablici, dolazi se do zaključka da je test uspješan.

5. ZAKLJUČAK

U današnje vrijeme tehnologija izrazito brzo napreduje te postoje razne aplikacije koje nam uvelike olakšavaju svakodnevne zadatke kako u privatnom tako i u poslovnom životu. Tim promjenama nije ostao imun ni obrazovni sustav koji prati korak s vremenom kako bi učenicima i studentima omogućio što bolju izobrazbu i znanja koja će im pomoći da se prilagode suživotu s tehnologijom.

Upravo je to razlog izrade ove aplikacije, koja će zauvijek u prošlost baciti stare načine popisivanja prisutnih studenata na predavanjima. Ogromna količina vremena se troši na prozivanje i popisivanje studenata, uz i ovako opširno građivo i premalu satnicu za pojedine kolegije. Aplikacija je jednostavnog i pristupačnog dizajna, što ju čini jako jednostavnom za korištenje.

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 biblioteke koje su bile neophodne za rad s QR kodovima su *Zxing* i *Emgu CV*. Problem velike papirologije riješen je tako da se svakom studentu dodjeli njegov jedinstven QR kod. QR kod se skenira putem kamere i dekodira programom, koji na kraju vrši upis studenata u bazu podataka zajedno s nazivom kolegija, vrste održane nastave, broja sati i trenutnog datuma. Aplikacija nudi i mogućnost prebacivanja svih podataka iz baze podataka u Excel tablicu gdje se mogu dalje obrađivati. Tako je uveliko olakšano praćenje prisutnih studenata i daljnja obrada podataka u vidu zadovoljava li student pravo na polaganje kolegija, što se lako može odrediti pomoću zbroja sati u Excel tablici. Nema više potrebe ni za ručnim unošenjem studenata u Excel tablicu, što je velika prednost uz pravedno vrednovanje tko je pohađao predavanja.

Iako je aplikacija zadovoljila postavljene ciljeve, ima mjesta za daljnji razvoj. Moglo bi se, između ostalog, doraditi ručno unošenje studenata od strane profesora u slučaju kašnjenja, napraviti izbor Excel tablica za svaki kolegij zasebno ili automatski ispis studenata koji zadovoljavaju pravo na polaganje kolegija jer su zadovoljili uvjet od 70% dolaznosti.

LITERATURA

- [1] „Documentation - Emgu CV: OpenCV in .NET (C#, VB, C++ and more)“. [Na internetu]. Dostupno na: <http://www.emgu.com/wiki/index.php/Documentation>. [Pristupljeno: 27-lip-2019].
- [2] *ZXing („Zebra Crossing“) barcode scanning library for Java, Android: zxing/zxing*. ZXing Project, 2019.
- [3] „QR Code features | QR Code.com“, 29-sij-2013. [Na internetu]. Dostupno na: <https://web.archive.org/web/20130129064920/http://www.qrcode.com/en/qrcodefeature.html>. [Pristupljeno: 03-ruj-2019].
- [4] „QR Code® Essentials“. DENSO Wave Incorporated.
- [5] R. Elmasri i S. Navathe, *Fundamentals of database systems*, 6th ed. Boston: Addison-Wesley, 2011.
- [6] „Database Design – 2nd Edition – The BC Open Textbook Project“. [Na internetu]. Dostupno na: <https://opentextbc.ca/dbdesign01/>. [Pristupljeno: 18-kol-2019].
- [7] S. Nakov, *Fundamentals of computer programming with C#: The Bulgarian C# programming book*. Sofia, Bulgari: Svetlin Nakov & co., 2013.
- [8] J. Miano, *Compressed image file formats : JPEG, PNG, GIF, XBM, BMP*. Reading, Mass. ; Harlow : Addison Wesley, 1999.
- [9] „What is a QR Code? | QRcode.com | DENSO WAVE“. [Na internetu]. Dostupno na: <https://www.qrcode.com/en/about/>. [Pristupljeno: 03-ruj-2019].

SAŽETAK

Naslov: Desktop aplikacija za evidenciju pohađanja nastave korištenjem kamere

Napretkom tehnologije dolazi do povećanja digitalizacije svijeta u kojem živimo, što dovodi do izrade raznih aplikacija koje nam uvelike olakšavaju svakodnevne poslove i zadatke. Izradom ove aplikacije riješen je problem kontroliranja dolaznosti studenata. Po pravilniku bi studenti trebali prisustvovati 70% predavanja i vježbi, no sustav potpisivanja omogućuje studentima međusobno potpisivanje. Osim toga, olakšana je daljnja obrada podataka u Excel tablici. Samo grafičko sučelje aplikacije je jednostavno kako izgledom tako i samim unosom podataka. Nakon što se aplikacija pokrene, pritiskom na gumb *Pokreni*, uključuje se kamera i aplikacija je spremna za korištenje. Prije nego što krenemo sa skeniranjem QR kodova studenata, profesor ili asistent odabire kolegij, vrstu nastave i unosi broj sati. Nakon što je QR kod skeniran vidljiv je na mjestu JMBAG studenta. Sama aplikacija radi provjeru ima li skenirani student pravo na slušanje kolegija i ovisno o tome javlja povratnu informaciju u obliku poruke. Nakon što su svi studenti upisani u bazu podataka zajedno sa nazivom kolegija, oblika nastave, broja sati i trenutnim datumom postoji mogućnost prebacivanja svih tih podataka u Excel tablicu radi lakše daljnje obrade.

Ključne riječi: C#, QR kod, baza podataka, MS SQL

ABSTRACT

Title: A Desktop Application for Recording Class Attendance Using Digital Camera

With the advancement of technology, the digitalization of the world we live in is increasing and that leads to the creation of various applications that make our daily jobs and tasks easier. The application made for this thesis solves the problem of attendance controlling. According to the rulebook, each student is obligated to attend 70% of lectures and exercises, but the current signing system allows them to sign each other. Not only that, but processing of the gathered data in the Excel sheet is facilitated. The graphical interface is simple in appearance, and entering data is made easier. After we press the Start button, the application is initiated, and the camera is turned on. With that, the application is ready to use. Firstly, the professor or assistant need to select the course, course form and number of hours they are going to hold. Scanning of the students QR codes can now begin. After the QR code is scanned, it is visible on the textbox with the label JMBAG of student. The application checks if the scanned student has the right to attend the course and based on that it shows a message. After all the students have been entered in the database (along with the course title, course form, number of hours and current date) it is possible to transfer all this data to an Excel table for easier further processing.

Keywords: C#, QR code, database, MS SQL

ŽIVOTPIS

Valentina Gotal rođena je 26. travnja 1997. godine u Osijeku, Hrvatska. Prvih šest razreda osnovne škole pohađa u osnovnoj školi Mladost u Osijeku, nakon čega se seli u Bilje i tamo završava svoje osnovnoškolsko obrazovanje u OS Bilje. Nakon osnovne škole upisuje III. Gimnaziju, Osijek, odnosno prirodoslovno – matematičku gimnaziju. 2016. godine dobiva izravan upis na preddiplomski sveučilišni studij elektrotehnike na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija te ga upisuje iste godine. Na 2. godini preddiplomskog sveučilišnog studija opredjeljuje se za izborni blok komunikacije i informatika te iste godine dobiva Priznanje za uspješnost tokom studiranja.

PRILOZI

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