



**Univerzitet u Novom Sadu   
Tehnički fakultet »Mihajlo Pupin«   
Zrenjanin**

**SEMINARSKI RAD**

**iz predmeta Razvoj softvera otvorenog koda**

**Tema: E-cvecara**

**Predmetni nastavnik: Doc. dr. Zoltan Kazi Studenti:Milutinović Lara SI29/17 Okuka Sonja SI 13/17**

**Zrenjanin, 2019/20.**

# Sadrzaj

[Sadrzaj 1](#_Toc42036072)

[1. Uvod 2](#_Toc42036073)

[2. Specifikacija zahteva korisnika 3](#_Toc42036074)

[3. Faze razvoja softvera 4](#_Toc42036075)

[3.1 Tabelarni prikaz slojeva i podslojeva aplikacije 4](#_Toc42036076)

[4. Prikaz softvera 6](#_Toc42036077)

[4.1 Pokretanje softvera i ekranski prikaz 6](#_Toc42036078)

[5. Opis implementacije 13](#_Toc42036079)

[5.1 Opis primenjenih tehnologija 13](#_Toc42036080)

[5.1.1 Microsoft Visual Studio 13](#_Toc42036081)

[5.1.2 C# programski jezik 13](#_Toc42036082)

[5.1.3 WPF 13](#_Toc42036083)

[5.1.4 Microsoft SQL Server 14](#_Toc42036084)

[5.2. Git – Kratka istorija 14](#_Toc42036085)

[5.2.1 Github nalog 15](#_Toc42036086)

[5.2.2 Kreiranje repozitorijuma 16](#_Toc42036087)

[5.3 Testiranje softvera 18](#_Toc42036088)

[6. Kreiranje baze podataka 19](#_Toc42036089)

[6.1 Kreiranje konekcionog stringa 20](#_Toc42036090)

[7. Ključni delovi koda 21](#_Toc42036091)

[8. Zaključak 26](#_Toc42036092)

[9. Literatura 27](#_Toc42036093)

# Uvod

Softver za aplikaciju “E-cvecara” pruža mogućnost u online naručivanja cveća. Aplikacija je namenjena za sve ljude koji nisu u mogućnosti da na drugi način dodju do onoga što im je potrebno. Korisnicima pruža mogućnost naručivanja cveća, i da to stigne na njihovu adresu.

Aplikacija se sastoji od skupa tabela koje su međusobno povezane u jednu bazu podataka. Svaka tabela ima svoj naziv, i podatke koji su određenog tipa koji čine datu tabelu koja se nalazi u bazi podataka. Tabele koje se nalaze u bazi su: *cvece, kupac, lokacija, prodaja.*

# Specifikacija zahteva korisnika

Aplikacija obuhvata sledeće:

* Početni prozor koji ujedno predstavlja i Log in prozor
* Registraciju, ukoliko korisnik nema već svoj nalog. Postoji i mogućnost da se korisnik uloguje kao admin i na taj nacin ima pristup funkcijama koje korisnik koji nije admin nema.
* Cvećaru, gde se nalaze svi podaci u vezi prodaje
* Lokaciju, u kojem je moguće uneti podatke o gradu, broju telefona i adresu
* Dodaj Proizvod, opcija koju moze da koristi samo admin, pruža mogućnost dodavanja novog cveta ukoliko se već ne nalazi na listi dostupnih

# Faze razvoja softvera

1. Kreiranje specifikacije zahteva korisnika
2. Kreiranje repozitorijuma na gitu (git hub)
3. Kreiranje baze podataka
4. Postavka projekta

* frontend
* backend

1. Frontend Izrada Data projekta za uspostavljanje konekcije sa kreiranom bazom podataka iz SQL Servera
2. Kreiranje klasa na backend-u
3. Implementacija
4. Dizajn (frontend deo) za implementirane klase na backend-u
5. Testiranje softvera
6. Izrada dokumentacije

## 3.1 Tabelarni prikaz slojeva i podslojeva aplikacije

|  |  |  |
| --- | --- | --- |
| SLOJ | PODSLOJ | REALIZACIJA U SEMINARSKOM RADU |
| PREZENTACIONI SLOJ | Login | WPF - Login.xaml |
|  | Registracija | WPF - Registracija.xaml |
|  | MainWindow | WPF - MainWindow.xaml |
|  | KupiProizvod | WPF – KupiProizvod.xaml |
|  | DodajProizvod | WPF - DodajProizvod.xaml |
|  | Lokacija | WPF - Lokacija.xaml |
| APLIKACIONI SLOJ | Login | WPF - Login.xaml.cs |
|  | Registracija | WPF - Registracija.xaml.cs |
|  | MainWindow | WPF - MainWindow.xaml.cs |
|  | KupiProizvod | WPF – KupiProizvod.xaml.cs |
|  | DodajProizvod | WPF - DodajProizvod.xaml.cs |
|  | Lokacija | WPF - Lokacija.xaml.cs |
| SLOJ BAZE PODATAKA | Baza | SQLconnection |

*Tabela 1: Prikaz slojeva i podslojeva aplikacije „E-cvecara“.*

# Prikaz softvera

## Pokretanje softvera i ekranski prikaz

Za pokretanje ove aplikacije neophodno je pokrenuti celokupan projekat u Visual Studio‒u , React projekat u Visual Studio Code-u, kao i uspostaviti konekciju sa bazom podataka.

Nakon uspešnog pokretanja aplikacije pokreće se početna stranica koja je ujedno i *Login* stranica aplikacije. Prozor *Login*, sadrži sliku, tri labele i dva dugmeta. Labele su: *Korisničko* *ime*, *Lozinka* i *Niste* *registrovani*. Dok su dva dugmeta, *Prijavi* *se* i *Registrujte* *se*. Kada se klikne dugme Prijavi se, posle popunjavanja korisničkog imena i lozinke, otvara se MainWindow.

A screenshot of a cell phone

Description automatically generated

*Slika 1: Prozor - Login*

Na ovoj starnici se korisnik uloguje koji već ima svoj nalog, ukoliko korisnik nema nalog, on ide klikom na dugme „Registrujte se“. Prozor *Registracija* sadrži četiri labele, i dugme za registrovanje, kao i checkbox kojim se obeležava da li je korisnik, tj.kupac admin. Labele su: *Vaše korisnčko ime*, *Lozinka*, *Ponovite lozinku i MolimoVas unesite sledece podatke*.

*A screenshot of a cell phone

Description automatically generatedSlika 2: Prozor „Registracija“*

Korisnik u ovom prozoru upisuje svoje informacije i pravi svoj nalog, kada upiše sve potrebne informacije ide klik na dugme „Registrujte se“. Postoji i mogućnost čekiranja „Admin“ opcije, na taj način će se korisnik ulogovati kao admin i imaće veće mogućnosti u samoj aplikaciji. Klikom na dugme „Registrujte se“., vraća se na prozor *Login*, odakle se dalje kupac prijavljuje.

Kada se korisnik uloguje otvara se prozor „Cvećara M“. Ovaj prozor sadrži, sliku, *DataGrid*, koji ispisuje podatke iz baze podatka, iz tabele *Cvece*, gde nam prikazuje *Naziv*, *Cenu* i *Vrstu* za dati cvet. Ispod *DataGrid*-a nalaze se labele i njihovi textbox-ovi. Imamo labele: *Naziv*, *Cena*, *Vrsta*. Njihovi textbox-ovi, se popunjavaju prilikom selektovanja cveta iz *DataGrid*-a. Pored toga *ovaj prozor* ima *DatePicker*, koji služi da se odredi datum. Ispod nalazi se dugme *Kupi*, klikom na dugme dobija se prozor koji sadrži sve podatke o datom proizvodu, neki vid potvrde pre kupovanja, klikom na dugme *Potvrdi*, dobijamo poruku da smo uspešno naručili proizvod. Pored slike nalaze se dva dugmeta, *Dodaj* *Proizvod* i *Lokacija*. Pristup *Dodaj* *Proizvod* prozoru, ima samo admin, dok *Lokaciji* imaju svi kupci. Klikom na jedna od ova dva dugmeta, dobija se prozor *Dodaj* *Proizvod* ili *Lokacija*. Klikom na dugme *Cencel* korisnik odustaje od kupovine.

A screenshot of a cell phone

Description automatically generated  
*Slika 3: Prozor „Cvećara M“*

Karakteristike lokacije su: *Grad*, *Telefon*, *Adresa*.Ovi podaci su tipa labele, ispod kojih se nalaze odgovarajuća textbox polja u koje se upisuju podaci. Polje *idLokacije* je tipa *TextBox,* ali njegova vidljivost je stavljena na Hidden.Preko *idLokacije*, baza podatka zna koji je grad selektovan, i biće nam prikazani podaci u TextBox-ove. U ovom prozoru se nalazi *DataGrid* kontrola, koja omogućuje prikaz podataka za sve dodate lokacije u bazu podataka, dugmad za dodavanje, promenu i brisanje podataka iz baze podataka.

A screenshot of a cell phone

Description automatically generated

*Slika 4: Prozor „Lokacija“*

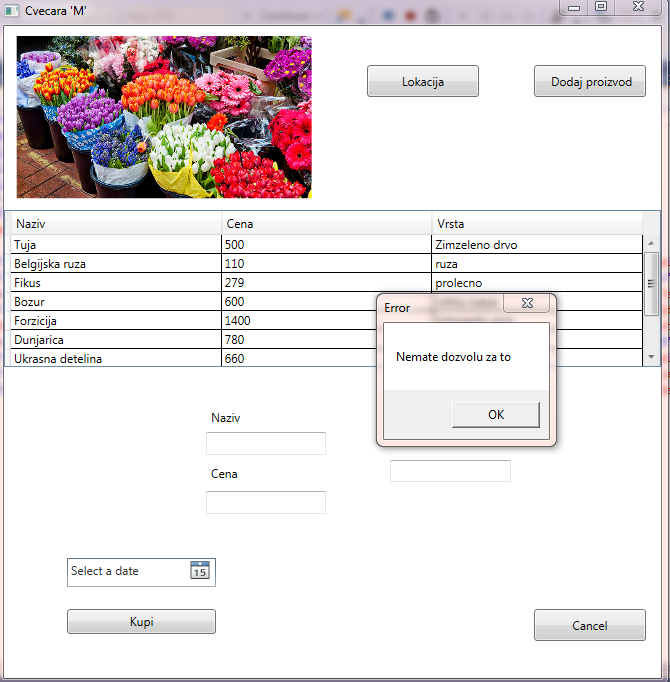
Karakteristike prozora *Dodaj proizvod* odnosno, cveća su: *Naziv*, *Cena*, *Vrsta*.Ovi podaci su tipa labele, ispod kojih se nalaze odgovarajuća textbox polja u koje se upisuju podaci. Polje *idCveca* je tipa *TextBox,* ali njegova vidljivost je stavljena na Hidden.Preko idCevca, baza podatka zna koji je cvet selektovan, i biće nam prikazani podaci u TextBox-ove. U ovom prozoru se nalazi *DataGrid* kontrola, koja omogućuje prikaz podataka za sve dodate cvetove u bazu podataka, dugmad za dodavanje, promenu i brisanje podataka iz baze podataka.

A screenshot of a cell phone

Description automatically generated

*Slika 5: Prozor „DodajProizvod“*

Ukoliko korisnik koji nije ulogovan kao admin klikne na dugme *Dodaj proizvod* na ekranu će se pojaviti sledeća poruka:



*Slika 6. Poruka (error)*

Na kraju, kada kupac želi da kupi biljku i popunio je sva tražena polja, na stranici „Cvećara M“ u donjem levom uglu ide klik na dugme „Kupi“. Kada klikne to dugme, otvara se krajnji prozor na kome se nalaze sve informacije porudžbine. U desnom donjem uglu se nalazi dugme „Potvrdi“. Kupac kada je siguran u sve što je naručio i proveri podatke, ide klik na dugme „Potvrdi“.

A screenshot of a cell phone

Description automatically generated

*Slika 7: Prozor „Kupi“*

# 5. Opis implementacije

## 5.1 Opis primenjenih tehnologija

Primenjene tehnologije predstavljaju koje su sve tehnologije upotrebljene prilikom izrade *Windows Presentation Foundation* aplikacije.

### 5.1.1 Microsoft Visual Studio

*Microsoft Visual Studio* je integrisano razvojno okruženje iz *Microsofta*. Koristi se za razvoj kompjuterskih programa za *Microsoft Windows*, kao i veb sajtove, veb aplikacije i veb servise. *Visual Studio* koristi *Microsoft* platforme za razvoj softvera kao što su *Windows API*, *Windows Forms*, *Windows Presentation Foundation*, *Windows Store* i *Microsoft Silverlight*. Može da se piše i upravlja kodom.

*Visual Studio* uključuje editor koda koji podržava *IntelliSense*. Dolazi sa ugrađenim *C#* i *C++* , *VB NET* (*Visual Basic .NET*) i *F#* programskim jezicima. Takođe se može programirati i u drugim programskim jezicima kao što su *JavaScript*, *CSS*, *XML*, *HTML*, itd. instaliranjem jezičkih servisa za ovo razvojno okruženje. *Microsoft Visual Studio* je jedan od najpopularnijih razvojnih okruženja koji pruža velike mogućnosti u razvoju programa. Njegove mane su donekle ispravljene pomoću raznih ekstenzija razvijenih kako od „*Microsofta*“ tako i od strane drugih programera. Konkretno kod programiranja *Windows form* aplikacija, on sam generiše skelet aplikacije i sve podesi automatski kako bi korisnik mogao da se bavi problemom koji treba da reši, a ne pravljenjem prozora aplikacije. Pored desktop aplikacija takođe nudi mogućnost kreiranja i veb aplikacija. Microsoft je jako dobro razvio pronalaženje i uklanjanje grešaka pri programiranju zbog toga predstavlja jako popularnu i korisnu alatku programerima.[7]

### 5.1.2 C# programski jezik

*C#* (*C Sharp*) programski jezik je jedan od mlađih programskih jezika nastao *2002.* godine kao sastavni deo *MS .NET Framework-a 1.0*. *C#* je objektno orijentisani programski jezik namenjen razvoju aplikacija na *.NET Framework* platformi. Pošto je objekno orijentisan programski jezik, prvenstveno je sagrađen od klasa koje čine skupove objekata koji imaju zajedničku strukturu i ponašanje. Takođe kao i mnogi objektno orijentisani programski jezici, sadrži sposobnost ekapsulacije, pisanje metoda, nasleđivanje između klasa i visoke performanse tokom kompajliranja. Iz razloga enkapsulacije postoji obrazac kojim atributima različitih klasa možemo spolјa pristupiti indirektno i to metodama *get* i *set*. *C#* je *case sensitive* programski jezik, što znači da je od velike važnosti da li kucamo malim ili velikim slovima.[8]

### 5.1.3 WPF

WPF, tj. *Windows Presentation Foundation*, je najnoviji *Microsoftov* pristup *GUI framework*-u, koji se koristi sa *.NET framework*-om. *GUI framework* je skraćenica za *Graphical User Interface* (*grafički korisnički interfejs*). Windows ima *GUI* za rad sa računarom, a veb pretraživač ima *GUI* koji omogućava surfovanje internetom.

*GUI framework* omogućava da se kreira aplikacija sa širokim spektrom *GUI* elemenata, kao što su *labele*, *tekstboks-*ovi i drugi dobro poznati elementi. Bez *GUI framework*-a treba ručno da se nacrtaju ovi elementi i da obradi sve scenarije i interakcije korisnika kao što su tekst unos teksta, i pomeranje miša. Sve ovo predstavlja mnogo posla, pa zbog toga većina programera koristi *GUI framework* koji će obaviti osnovne radnje i omogućiti programerima da se fokusiraju na razvijanju aplikacija.

Postoji dosta *GUI framework*-a, ali za *.NET* programere, najzanimljivji su *Windows Forme* i *WPF*. *WPF* je noviji, ali *Microsoft* i dalje održava i podržava *Windows Forme*. Postoji nekoliko razlika između ova dva *framework*-a, ali njihova svrha je ista, da se kreira na najlakši našin sa odličnim *GUI*.[9]

### 5.1.4 Microsoft SQL Server

*Microsoft SQL Server* je relacijska baza podataka kojoj je primarni jezik za upite *Transact SQL* (*T-SQL*), što znači da osim osnovnih i klasičnih (*SELECT tipa*) *SQL* upita dozvoljava i složenije stvari poput menjanja programskog toka (*IF naredba*) i slično. *Transact SQL* nastao je kao plod saradnje između *Microsofta* i *Sybasea*. *SQL* server je baza podataka koja se smestila na prag između manjih i srednjih baza. Prvi *SQL Server* za *Windows NT* izašao je isto kada i sami *Windows*-i.

*SQL Server* je prva baza podataka na svetu koja je posedovala korisnički interfejs. Takođe, *SQL Server* je prva komercijalna baza podataka koja je podržala *Intelovu 64*-bitnu arhitekturu procesora. Od *SQL Servera 2000* do *SQL Servera 2005* napravljeni su golemi pomaci u poboljšanju same baze podataka. Najbitniji od njih su: Klijentski deo *IDE* alata, *SQL Server Integration Services*, *Reporting Server*, *OLAP* server, podrška za *data mining*, *Notification Services* itd.

*Microsoft SQL* podržava „*Open Database Connectivity*“, odnosno skraćeno *ODBC* tehnologiju. *SQL Server 2005* poseduje i podršku za *Web servise*, tj. za *Simple Object Access Protocol*, *Service Oriented Architecture Protocol* odnosno skraćeno *SOAP W3C* standard. To je izvanredna stvar zato što omogućuje ne *Windows* klijentima da komuniciraju preko te platforme sa *SQL Serverom*. U verziji 2005 takođe je izdan i sertifikovan *JDBC API* za komunikaciju sa *Java* aplikacijama. *SQL Server* također poseduje i mogućnosti poput *mirroringa*, *clusteringa* i *snapshots*-ova. *Microsoft* se stvarno potrudio, te je omogućio brojne aplikacije koje spremaju podatke na *SQL Server*. *SQL Server 2005* uključuje i *Common Language Runtime* odnosno CLR.[10]

## 5.2. **Git – Kratka istorija**

Kao i mnoge dobre stvari u životu, Git je počeo sa malo kreativnog uništenja i plahovite polemike.

Linuks jezgro je prilično širok softverski projekat otvorenog koda. Tokom većine vremena održavanja Linuksovog jezgra (1991-2002), promene u softveru slate su unaokolo kao zakrpe i arhivirani fajlovi. Godine 2002, projekat Linuks jezgra počeo je da koristi vlasnički DVCS koji se zvao *BitKeeper*.

Godine 2005, odnos između zajednice koja je radila na Linuks jezgru i komercijalne kompanije koja je razvijala *BitKeeper* se raspao, i besplatan status alata bio je ukinut. Ovo je primoralo Linuksovu zajednicu programera (i posebno Linusa Torvaldsa, osnivača Linuksa) da osmisle svoj sopstveni alat oslanjajući se na neke lekcije koje su naučili dok su koristili *BitKeeper*. Neki od ciljeva koje je imao novi sistem bili su sledeći:

 brzina,

 jednostavan dizajn,

 snažna podrška za nelinearni razvitak (na hiljade paralelnih grana),

 potpuna distribucija,

 mogućnost da efikasno rukuje velikim projektima kao što je Linuks jezgro (brzina i veličina podataka).

Od njegovog nastanka 2005, Git je evoluirao i sazreo u alat koji je bio jednostavan za upotrebu, ali je zadržao ove kvalitete. Neverovatno je brz, veoma je efikasan sa velikim projektima, i ima odličan sistem grananja za nelinearno razviće programa. [1]

### 5.2.1 Github nalog

Kako bi kreirali github nalog, ide se na sledeću adresu: <https://github.com/> . Unosimo potrebne podatke, poput korisničkog imena, e-mail adrese i password-a. Primer je prikazan na slici. Slika broj 8.

A screenshot of a computer screen

Description automatically generated

*Slika 8: Kreiranje github naloga*

### 5.2.2 Kreiranje repozitorijuma

Za kreiranje repozitorijuma ide se na klik New repository.

A screenshot of a cell phone

Description automatically generated

*Slika 9: Kreiranje repozitorijuma*

Otvara se prozor za unos potrebnih podataka za kreiranje repozitorijuma. Navodimo željeno korisničko ime i dajemo naziv repozitorijumu. Biramo opciju da li želimo da bude javan ili ne.

A screenshot of a computer

Description automatically generated

*Slika 10: Kreiranje repozitorijuma – unos podataka*

Nakon kreiranog repozitorijuma, otvara se prozor u kom vidimo sadržaj repozitorijuma. On sadrži fajlove README, LICENSE i .gitignore. Sve je prikazano na Slika broj 11.

A screenshot of a social media post

Description automatically generated

*Slika 11: Kreiran repozitorijum*

Plan projekta – Faze razvoja: 1. Specifikacija zahteva 2. Dizajn aplikacije: Baze podataka i srednjeg aplikacionog sloja. 4. Implementacija slojeva 5. Testiranje i integracija 6. Dokumentovanje i realizacija.

Radi lakšeg praćenja završenih stavki, kao i onih koje trebaju da budu u izradi, kreiran je projekat u okviru github-a sa 3 tabele: to do(ono što treba da se uradi), in progress(ukoliko neko trenutno radi na tom zadatku) i done(završeni zadaci).

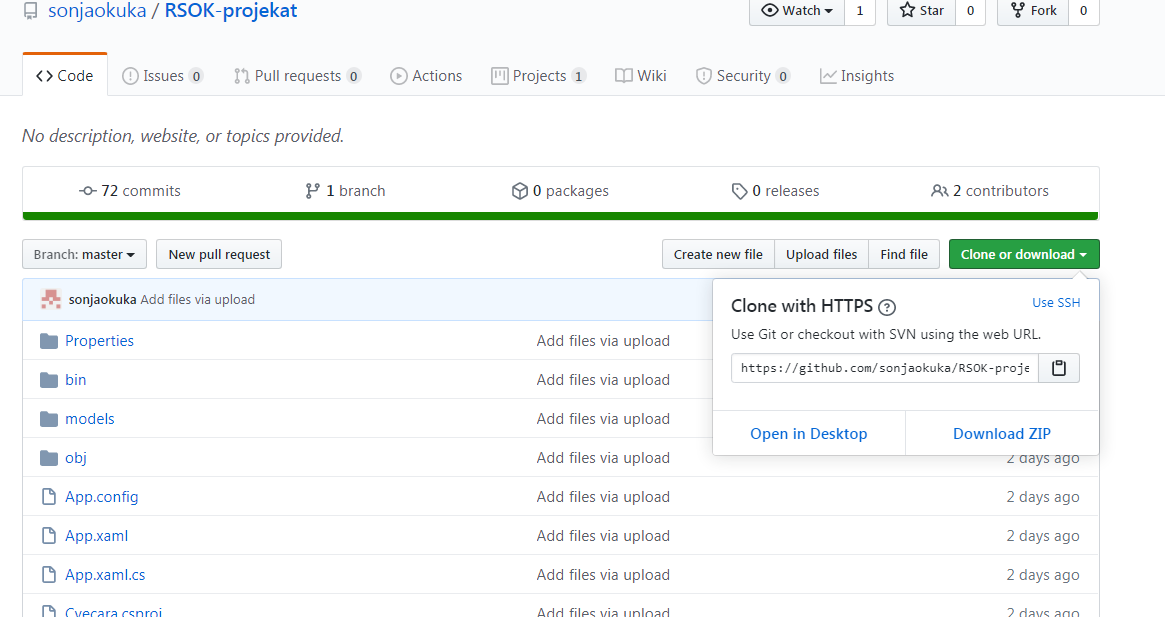
Trenutno stanje u jednom trenutku izrade aplikacije dato je na slici 12.

A screenshot of a social media post

Description automatically generated

*Slika 12: Kreiran repozitorijum i prikaz tabele sa karticama*

Preuzimanje datoteka - izvršeno opcijom “Download ZIP” čime se kompletna aplikacija preuzima kao zapakovana arhiva.



*Slika 13. Preuzimanje datoteka sa Git-huba*

Github daje mogućnost prikaza svih commit-ova, a tokom izrade ih je bilo 80.



*Slika 14. Prikaz ukupnog broja commit-a*

Github, takodje daje mogućnost prikazivanja rada svakog člana tima.

A screen shot of a computer

Description automatically generated

*Slika 15. Prikaz commit-a svakog člana tima*

## 5.3 Testiranje softvera

Testiranje softvera je izvedeno na dva načina: prilikom logovanja korisnika i prilikom upisa novog proizvoda tako što su kreirani slučajevi testiranja.

1. Testiranje izvršavanja uslova klase Registacija.xaml.cs:

if (txtLozinka.Password.Equals(txtPonovi.Password))

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Broj slučaja | Opis slučaja | Test podaci | Očekivani rezultat | Trenutni rezultat | Status  (uspeh/neuspeh) |
| 1. | Proverava da li je lozinka ispravno ponovljena?  Potrebno je zadovoljiti uslov da ponovljena lozinka bude potpuno ista kao prethodno uneta lozinka tokom registracije  **Prva uneta lozinka se ne poklapa sa ponovljenom lozinkom.** | Prva uneta lozinka: lara123  Ponovljena lozinka:  lara122 | Lozinke se ne poklapaju. | Lozinke se ne poklapaju. | uspeh |

*Tabela 2. Prvo testiranje*

2. Testiranje izvršavanja metode klase DodajProizvod.xaml.cs:

command.Parameters.AddWithValue("@Cena", txtCena.Text);

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Broj slučaja | Opis slučaja | Test podaci | Očekivani rezultat | Trenutni rezultat | Status  (uspeh/neuspeh) |
| 1. | Proverava da li je cena ispravno uneta?  Potrebno je zadovoljiti uslov da podatak cena bude uneta u ciframa a ne napisana slovima.  **Cena je pogrešno uneta(mora biti int, a ne string).** | Uneta cena: trista dvadeset | Nije moguće uneti cenu slovima. | Cenu je nemoguće uneti slovima. | uspeh |

*Tabela 3. Drugo testiranje*

Kod drugog testiranja pronađena je greška u bazi podataka jer je Cena uneta kao Int i ne može se konvertovati u strung.

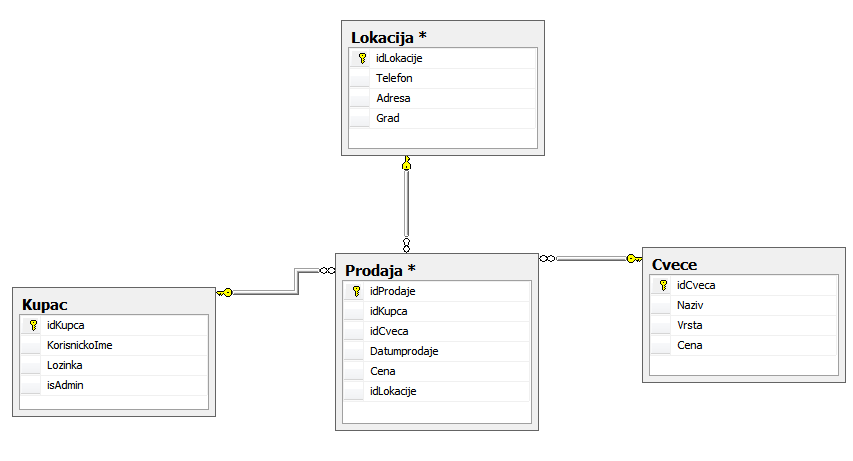
# 6. Kreiranje baze podataka

Potrebno je da se kreira baza podataka sa nazivom *Cvecara*. Nakon toga kreiraju se tabele sa njihovim atributima. U svakoj tabeli treba da se nalazi jedan od atributa koji će biti primarni ključ te tabele.

Podaci tabele *cvece* su: *idCveca, Naziv, Cena, Vrsta.*Podatak *idCveca* je primarni ključ tabele *Cvece* i ima vrednost *IDENTITY*, što znači da se vrednost *idCveca* povećava za jedan za svako novo uneto cveće. Tabela *Cvece* je povezana sa tabelom *Kupac,* vezom više prema više. Što dovodi do toga da se svara nova tabela, *Prodaja*, koja pored stranih ključa *idCveca* i *idKupca*, ima primaran ključ *idProdaje*, a Datumprodaje se bira iz aplikacije, dok se Cena unosi direktno u bazu podataka.

Podaci tabele *Kupac* su: *idKupca, KorisnickoIme, Lozinka, isAdmin.* Podatak *idKupca* je primarni ključ tabele *Kupac* i ima vrednost *IDENTITY*, što znači da se vrednost *idKupca* povećava za jedan za svakog novog uneto kupca. Tabela *Kupac* je povezana sa tabelom *Cvece*. *KorisnickoIme* predstavlja ime kojim se korisnik povezuje sa aplikacijom dok *Lozinka* predstavlja šifru koja je takođe potrebna prilikom povezivanja. A *isAdmin* proverava da li je korsnik admin ili ne.

Podaci tabele *Lokacija* su: *idLokacije, Telefon, Adresa, Grad.* Podatak *idLokacije* je primarni ključ tabele *Lokacija* i ima vrednost IDENTITY, što znači da se vrednost *idLokacije* povećava za jedan za svaku novu unetu lokaciju*.* Podaci se unose iz aplikacije.



*Slika 16.* *Šema baze podataka- Cvećara*

*USE [Cvecara]*

*GO*

*ALTER DATABASE [Cvecara] SET COMPATIBILITY\_LEVEL = 140*

*GO*

*IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))*

*begin*

*EXEC [Cvecara].[dbo].[sp\_fulltext\_database] @action = 'enable'*

*end*

*GO*

*ALTER DATABASE [Cvecara] SET ANSI\_NULL\_DEFAULT OFF*

*GO*

*ALTER DATABASE [Cvecara] SET ANSI\_NULLS OFF*

*GO*

*ALTER DATABASE [Cvecara] SET ANSI\_PADDING OFF*

*GO*

*ALTER DATABASE [Cvecara] SET ANSI\_WARNINGS OFF*

*GO*

*ALTER DATABASE [Cvecara] SET ARITHABORT OFF*

*GO*

*ALTER DATABASE [Cvecara] SET AUTO\_CLOSE OFF*

*GO*

*ALTER DATABASE [Cvecara] SET AUTO\_SHRINK OFF*

*GO*

*ALTER DATABASE [Cvecara] SET AUTO\_UPDATE\_STATISTICS ON*

*GO*

*ALTER DATABASE [Cvecara] SET CURSOR\_CLOSE\_ON\_COMMIT OFF*

*GO*

*ALTER DATABASE [Cvecara] SET CURSOR\_DEFAULT GLOBAL*

*GO*

*ALTER DATABASE [Cvecara] SET CONCAT\_NULL\_YIELDS\_NULL OFF*

*GO*

*ALTER DATABASE [Cvecara] SET NUMERIC\_ROUNDABORT OFF*

*GO*

*ALTER DATABASE [Cvecara] SET QUOTED\_IDENTIFIER OFF*

*GO*

*ALTER DATABASE [Cvecara] SET RECURSIVE\_TRIGGERS OFF*

*GO*

*ALTER DATABASE [Cvecara] SET DISABLE\_BROKER*

*GO*

*ALTER DATABASE [Cvecara] SET AUTO\_UPDATE\_STATISTICS\_ASYNC OFF*

*GO*

*ALTER DATABASE [Cvecara] SET DATE\_CORRELATION\_OPTIMIZATION OFF*

*GO*

*ALTER DATABASE [Cvecara] SET TRUSTWORTHY OFF*

*GO*

*ALTER DATABASE [Cvecara] SET ALLOW\_SNAPSHOT\_ISOLATION OFF*

*GO*

*ALTER DATABASE [Cvecara] SET PARAMETERIZATION SIMPLE*

*GO*

*ALTER DATABASE [Cvecara] SET READ\_COMMITTED\_SNAPSHOT OFF*

*GO*

*ALTER DATABASE [Cvecara] SET HONOR\_BROKER\_PRIORITY OFF*

*GO*

*ALTER DATABASE [Cvecara] SET RECOVERY SIMPLE*

*GO*

*ALTER DATABASE [Cvecara] SET MULTI\_USER*

*GO*

*ALTER DATABASE [Cvecara] SET PAGE\_VERIFY CHECKSUM*

*GO*

*ALTER DATABASE [Cvecara] SET DB\_CHAINING OFF*

*GO*

*ALTER DATABASE [Cvecara] SET FILESTREAM( NON\_TRANSACTED\_ACCESS = OFF )*

*GO*

*ALTER DATABASE [Cvecara] SET TARGET\_RECOVERY\_TIME = 60 SECONDS*

*GO*

*ALTER DATABASE [Cvecara] SET DELAYED\_DURABILITY = DISABLED*

*GO*

*ALTER DATABASE [Cvecara] SET QUERY\_STORE = OFF*

*GO*

*USE [Cvecara]*

*GO*

*/\*\*\*\*\*\* Object: Table [dbo].[Cvece] Script Date: 26.5.2020. 11:47:39 \*\*\*\*\*\*/*

*SET ANSI\_NULLS ON*

*GO*

*SET QUOTED\_IDENTIFIER ON*

*GO*

*CREATE TABLE [dbo].[Cvece](*

*[idCveca] [int] IDENTITY(1,1) NOT NULL,*

*[Naziv] [varchar](50) NOT NULL,*

*[Vrsta] [varchar](50) NOT NULL,*

*[Cena] [int] NULL,*

*CONSTRAINT [PK\_\_Cvece\_\_B0D7F48097949D8D] PRIMARY KEY CLUSTERED*

*(*

*[idCveca] ASC*

*)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]*

*) ON [PRIMARY]*

*GO*

*/\*\*\*\*\*\* Object: Table [dbo].[Kupac] Script Date: 26.5.2020. 11:47:39 \*\*\*\*\*\*/*

*SET ANSI\_NULLS ON*

*GO*

*SET QUOTED\_IDENTIFIER ON*

*GO*

*CREATE TABLE [dbo].[Kupac](*

*[idKupca] [int] IDENTITY(1,1) NOT NULL,*

*[KorisnickoIme] [varchar](100) NOT NULL,*

*[Lozinka] [varchar](100) NOT NULL,*

*[isAdmin] [nvarchar](5) NULL,*

*CONSTRAINT [PK\_\_Kupac\_\_5A2BCF9D1FAAC9FF] PRIMARY KEY CLUSTERED*

*(*

*[idKupca] ASC*

*)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]*

*) ON [PRIMARY]*

*GO*

*/\*\*\*\*\*\* Object: Table [dbo].[Lokacija] Script Date: 26.5.2020. 11:47:39 \*\*\*\*\*\*/*

*SET ANSI\_NULLS ON*

*GO*

*SET QUOTED\_IDENTIFIER ON*

*GO*

*CREATE TABLE [dbo].[Lokacija](*

*[idLokacije] [int] IDENTITY(1,1) NOT NULL,*

*[Telefon] [nvarchar](100) NOT NULL,*

*[Adresa] [nvarchar](200) NOT NULL,*

*[Grad] [nchar](200) NOT NULL,*

*CONSTRAINT [PK\_Lokacija] PRIMARY KEY CLUSTERED*

*(*

*[idLokacije] ASC*

*)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]*

*) ON [PRIMARY]*

*GO*

*/\*\*\*\*\*\* Object: Table [dbo].[Prodaja] Script Date: 26.5.2020. 11:47:39 \*\*\*\*\*\*/*

*SET ANSI\_NULLS ON*

*GO*

*SET QUOTED\_IDENTIFIER ON*

*GO*

*CREATE TABLE [dbo].[Prodaja](*

*[idProdaje] [int] IDENTITY(1,1) NOT NULL,*

*[idKupca] [int] NOT NULL,*

*[idCveca] [int] NOT NULL,*

*[Datumprodaje] [varchar](50) NOT NULL,*

*[Cena] [bigint] NOT NULL,*

*CONSTRAINT [PK\_\_Prodaja\_\_36FE0BC499FA4018] PRIMARY KEY CLUSTERED*

*(*

*[idProdaje] ASC*

*)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]*

*) ON [PRIMARY]*

*GO*

*ALTER TABLE [dbo].[Prodaja] WITH CHECK ADD CONSTRAINT [fk\_prodajacveca] FOREIGN KEY([idCveca])*

*REFERENCES [dbo].[Cvece] ([idCveca])*

*GO*

*ALTER TABLE [dbo].[Prodaja] CHECK CONSTRAINT [fk\_prodajacveca]*

*GO*

*ALTER TABLE [dbo].[Prodaja] WITH CHECK ADD CONSTRAINT [fk\_prodaje] FOREIGN KEY([idKupca])*

*REFERENCES [dbo].[Kupac] ([idKupca])*

*GO*

*ALTER TABLE [dbo].[Prodaja] CHECK CONSTRAINT [fk\_prodaje]*

*GO*

*USE [master]*

*GO*

*ALTER DATABASE [Cvecara] SET READ\_WRITE*

*GO*

*Listting 1. Prikaz baze podataka*

## 6.1Kreiranje konekcionog stringa

Nakon kreiranja baze podataka, potrebno je da se kreira konekcioni String, koji omogućava povezivanje na bazu podataka. Potrebno je da se u *App.config* doda konekcioni String, a u njemu *Data Source* (*Listting 2.*)*.*

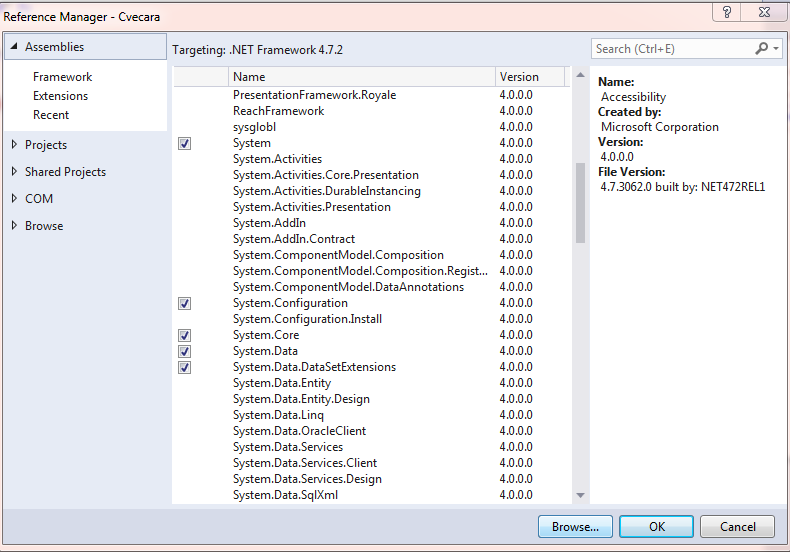
<connectionStrings>

<add name="connCvecara" connectionString="Data Source=WIN7-PC\SQLEXPRESS;Initial Catalog=Cvecara;Integrated Security=True"/>

</connectionStrings>

*Listting 2. Konekcioni String za povezivanje na bazu podataka*

Kada se kreira konekcioni String potrebno je da se doda referenca da prozori mogu da rade sa konfiguracionom bazom u kojem je putanja do baze podataka. Referenca se dodaje klikom na desni klik na *References* – *Add Reference* i otvara se *Reference Manager* u kojem je potrebno da se čekira *System Configuration.*



*Slika 17.* *Dodavanje reference System Configuration*

# 7. Ključni delovi koda

U prozoru *Login* nalaze se sql komande koje čitaju upisani useranme i password korisnika prilikom prijavljivanja. Ako se upisani podaci ne poklapaju sa podacima iz baze podataka odnosno ako je neki od ovih podataka netačan onda se pojavljuje poruka *„Username or password is inccorect!“*

private void Prijavi\_Click(object sender, RoutedEventArgs e)

{

using (SqlConnection sqlConnection = new SqlConnection("Data Source=LARA\\SQLEXPRESS;Initial Catalog=Cvecara;Integrated Security=True"))

{

sqlConnection.Open();

using (SqlCommand command = sqlConnection.CreateCommand())

{

command.CommandText = "Select count(1) from Kupac where KorisnickoIme=@Username and Lozinka=@Password";

command.Parameters.AddWithValue("@Username", txtUsername.Text);

command.Parameters.AddWithValue("@Password", txtPassword.Password);

if (Convert.ToInt32(command.ExecuteScalar()) == 1)

{

MainWindow main = new MainWindow(GetUserContents());

main.Show();

Close();

}

else

{

MessageBox.Show("Username or password is incorrect!");

}

}

}}

*Listting 3. Prijavljivanje u aplikaciju, provera Korisničkog imena i lozinke*

U prozoru *Login* putem sql komande se proverava da li je korisnik koji se prijavljuje *Admin* ili nije.

private Korisnik GetUserContents()

{

Korisnik user = new Korisnik();

using (SqlConnection sqlConnection = new SqlConnection("Data Source=LARA\\SQLEXPRESS;Initial Catalog=Cvecara;Integrated Security=True"))

{

sqlConnection.Open();

using (SqlCommand command = sqlConnection.CreateCommand())

{

command.CommandText = "Select \* from Kupac where KorisnickoIme=@Username and Lozinka=@Password";

command.Parameters.AddWithValue("@Username", txtUsername.Text);

command.Parameters.AddWithValue("@Password", txtPassword.Password);

using (SqlDataReader reader = command.ExecuteReader())

{

while (reader.Read())

{

user.Id = (int)reader["idKupca"];

user.Username = reader["KorisnickoIme"] as string;

user.Password = reader["Lozinka"] as string;

user.isAdmin = reader["isAdmin"] as string;

}

}

}

}

return user;

}

*Listting 4. Provera da li je korisnik admin pri prijavljivanju*

Registracija novog korisnika korišćenjem sql komandi za upis i proveru podataka kupca.

private void RegisterButton\_Click(object sender, RoutedEventArgs e)

{

if (Admin.IsChecked == true)

{

isAdmin = "Yes";

}

else

{

isAdmin = "No";

}

using (SqlConnection sqlConnection = new SqlConnection("Data Source=LARA\\SQLEXPRESS;Initial Catalog=Cvecara;Integrated Security=True"))

{

sqlConnection.Open();

using (SqlCommand command = sqlConnection.CreateCommand())

{

if (txtLozinka.Password.Equals(txtPonovi.Password))

{

command.CommandText = "Insert into Kupac([KorisnickoIme], [Lozinka], [isAdmin]) values (@Username,@Password,@isAdmin)";

command.Parameters.AddWithValue("@Username", txtKorisnickoime.Text);

command.Parameters.AddWithValue("@Password", txtLozinka.Password);

command.Parameters.AddWithValue("@isAdmin", isAdmin);

command.ExecuteNonQuery();

}

else

{

MessageBox.Show("Lozinke se ne poklapaju!");

}

}

}

Close();

}

*Lissting 5. Registracija novog korisnika, odnosno kupca*

Korišćenjem sql komandi u prozoru *MainWindow* prikazuju se uneti podaci(*idCveca, naziv, cena, vrsta*) cveća iz baze podataka.

private void prikazCveca()

{

using (SqlConnection connection = new SqlConnection("Data Source=LARA\\SQLEXPRESS;Initial Catalog=Cvecara;Integrated Security=True"))

{

connection.Open();

using (SqlCommand command = connection.CreateCommand())

{

command.CommandText = "SELECT idCveca, naziv, cena, vrsta FROM Cvece";

using (SqlDataAdapter dataAdapter = new SqlDataAdapter(command))

{

DataTable dataTable = new DataTable("Cvece");

dataAdapter.Fill(dataTable);

CveceDataGrid.ItemsSource = dataTable.DefaultView;

}

}

}

}

*Listting 6. Prikaz Cveca iz baze podataka u DataGrid-u u MainWindow-u*

U prozoru *DodajProizvod* moguće je uneti podatke o novom proizvodu korišćenjem datih sql komandi čime se podaci direktno upisuju u bazu.

private void Dodaj\_Click(object sender, RoutedEventArgs e)

{

using (SqlConnection connection = new SqlConnection("Data Source=DESKTOP-Q7JODCU\\SQLEXPRESS;Initial Catalog=Cvecaraa;Integrated Security=True"))

{

connection.Open();

using(SqlCommand command = connection.CreateCommand())

{

command.CommandText = "INSERT INTO Cvece (Naziv, Vrsta, Cena) VALUES(@Naziv, @Vrsta, @Cena)";

command.Parameters.AddWithValue("@Naziv", txtNaziv.Text);

command.Parameters.AddWithValue("@Cena", txtCena.Text);

command.Parameters.AddWithValue("@Vrsta", txtVrsta.Text);

command.Connection = connection;

int provera = command.ExecuteNonQuery();

if (provera == 1)

{

MessageBox.Show("Podaci su uspešno upisani");

prikazCveca();

}

ponistiUnosTxt();

}

}

}

*Listting 7. Dodavanje novog proizvoda u bazu podataka, preko aplikacije*

U prozoru *DodajProizvod* takodje je moguće obrisati prethodno upisane podatke kako iz aplikacije tako automatski i iz baze podataka korišćenjem sql komandi za brisanje primenjenih na dumetu *Obriši*.

private void Obrisi\_Click(object sender, RoutedEventArgs e)

{

using (SqlConnection connection = new SqlConnection("Data Source=DESKTOP-Q7JODCU\\SQLEXPRESS;Initial Catalog=Cvecaraa;Integrated Security=True"))

{

connection.Open();

using (SqlCommand command = connection.CreateCommand())

{

command.CommandText = "DELETE FROM Cvece WHERE idCveca = @idCveca";

command.Parameters.AddWithValue("@idCveca", txtidCveca.Text);

command.Connection = connection;

int provera = command.ExecuteNonQuery();

if (provera == 1)

{

MessageBox.Show("Podaci su uspesno obrisani");

prikazCveca();

}

ponistiUnosTxt();

}

}

}

*Listting 8. Brisanje proizvoda iz baze podataka, preko aplikacije*

Pored ovoga moguće je i menjati prethodno unete podatke iz aplikacije u bazi podataka. To nam omogućava sql upit *UPDATE* posle koje nakon provere dobijamo poruku o uspešnosti izmene.

private void Izmeni\_Click(object sender, RoutedEventArgs e)

{

using (SqlConnection connection = new SqlConnection("Data Source=DESKTOP-Q7JODCU\\SQLEXPRESS;Initial Catalog=Cvecaraa;Integrated Security=True"))

{

connection.Open();

using (SqlCommand command = connection.CreateCommand())

{

command.CommandText = "UPDATE Cvece SET Naziv = @naziv, Cena = @cena, Vrsta = @vrsta WHERE idCveca = @idCveca";

command.Parameters.AddWithValue("@idCveca", txtidCveca.Text);

command.Parameters.AddWithValue("@naziv", txtNaziv.Text);

command.Parameters.AddWithValue("@cena", txtCena.Text);

command.Parameters.AddWithValue("@vrsta", txtVrsta.Text);

command.Connection = connection;

int provera = command.ExecuteNonQuery();

if (provera == 1)

{

MessageBox.Show("Podaci su uspesno promenjeni");

prikazCveca();

}

ponistiUnosTxt();

}

}}

*Listting 9. Izmena proizoda u bazi podataka iz aplikacije*

DataGrid omogućava prikaz prethodno unetih podataka o cveću iz prozora *DodajProizvod* u prozoru *MainWindow*.

private void CveceDataGrid\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

DataGrid dg = sender as DataGrid;

DataRowView dr = dg.SelectedItem as DataRowView;

if (dr != null)

{

txtNaziv.Text = dr["Naziv"].ToString();

txtCena.Text = dr["Cena"].ToString();

txtVrsta.Text = dr["Vrsta"].ToString();

txtidCveca.Text = dr["idCveca"].ToString();

}

}

*Listting 10. Prikaz podataka o proizvodu u TextBox-ove van DataGrid-a*

Kodovi za citanje iz baze, dodavanje, brisanje i izmenu u bazi kao i prikaz podataka o proizvodu u TexBox-ovima van DataGrid-a, su slični kao kodovi za čitanje, dodavanje, brisanje i izmenu iz baze podataka za Lokacije, kao i prikaz podataka o lokaciji u TextBox-ovima van DataGrid-a.

# 

# 8. Zaključak

Git čuva i misli na podatke na mnogo drugačiji način od ostalih sistema, iako je korisnički interfejs veoma sličan, i razumevanje tih razlika će pomoći da ne dođe do zabune tokom njegovog korišćenja. [3]

Mogućnosti za dalji razvoj aplikacije jeste da se dozvoli prikaz lokacija cvećara, ali da se onemogući izmena, dodavanje i brisanje lokacija ukoliko niste adim. Pored toga moglo bi se dodati i stanje cveća, odnosno koliko trenutno cvećara raspolaže sa datim cvećem, kao i da pri naručivanju određenog cveta, njegov broj na stanju da se smanji.

# 9. Literatura

[1] https://git-scm.com/book/sr/v2/Po%C4%8Detak-Kratka-istorija-Gita

[3] <https://git-scm.com/book/sr/v2/Po%C4%8Detak-Osnove-Gita>

[4] <https://www.microsoft.com/en-us/sql-server/sql-server-downloads>

[5] <https://guides.github.com/introduction/git-handbook/>

[6] <https://reactjs.org/>

[7] <http://en.wikipedia.org/wiki/Microsoft_Visual_Studio>

[8] <https://en.wikipedia.org/wiki/C_Sharp_%28programming_language%29>

[9] <https://wpf-tutorial.com/about-wpf/what-is-wpf/>

[10] <https://en.wikipedia.org/wiki/Microsoft_SQL_Server>