**Univerzitet u Novom Sadu**

**Tehnički fakultet „Mihajlo Pupin“**

**Zrenjanin**

SEMINARSKI RAD

**KUD**

Razvoj softvera otvorenog koda

Profesor: Doc. dr. Kazi Zoltan Student: Ivana Tornjanski SI 5/17

Ćorluka Milica SI 38/17

Zrenjanin, 2020. godina

**Sadržaj**:

[1. Uvod 3](#_Toc51712861)

[2. Specifikacija zahteva korisnika 4](#_Toc51712862)

[3. Faze razvoja softvera 5](#_Toc51712863)

[3.1. Tabelarni prikaz slojeva i podslojeva aplikacije 5](#_Toc51712864)

[4. Prikaz softvera 6](#_Toc51712865)

[4.1. Pokretanje softvera i ekranski prikaz 6](#_Toc51712866)

[5. Opis implementacije 12](#_Toc51712867)

[5.1. Opis primenjenih tehnologija 12](#_Toc51712868)

[5.1.1. Microsoft Visual Studio 12](#_Toc51712869)

[5.1.2 C# programski jezik 12](#_Toc51712870)

[5.1.3 WPF 13](#_Toc51712871)

[5.1.4 Microsoft SQL Server Management Studio 18 13](#_Toc51712872)

[5.2 GitHub 14](#_Toc51712873)

[5.2.1 Kreiranje naloga 15](#_Toc51712874)

[5.2.3 Kreiranje projekta 18](#_Toc51712875)

[5.3 Testiranje softvera 20](#_Toc51712876)

[6. Baza podataka 21](#_Toc51712877)

[6.1 Konekcioni string 22](#_Toc51712878)

[7. Ključni delovi koda 23](#_Toc51712879)

[8. Zaključak 25](#_Toc51712880)

[9. Literatura 26](#_Toc51712881)

# Uvod

Aplikacija Kud omogućava online unošenje kudova, kao i samih igrača koji su članovi kudova. Namenjena je za vodjenje evidencija i omogućava pregled podataka. Sastoji se od dve tabele. Jedna tabela omogućava uvid u podatke u kudovima, a druga tabela sadrži podatke o igračima.

Dve tabele koje su deo aplikacije medjusobno su povezane u jednu bazu podataka. U bazi podataka svaka tabela ima svoj naziv i podatke odredjenog tipa. Nazivi tabela u bazi su: Igrac i KUD.

# Specifikacija zahteva korisnika

Aplikacija obuhvata:

▪ početni ujedno i prvi prozor prozor sadrži dva dugmeta, korisnik ima mogućnost da kliknom na izabrano dugme pristupi podacima o kudovima/igračima (u zavisnosti za koje dugme se opredeli);

▪ ukoliko korisnik klikne na dugme kudovi, otvara se sledeći prozor (drugi po redu u tom slučaju), ovaj prozor omogućava uvid u već postojeće podatke o kudovima, takodje korisnik ima mogućnost da kreira novi kud, izmeni ili obriše već postojeći;

▪ treći prozor (prozor igrači), pruža uvid u već postojeće podatke o igračima i mogućnost unosa novih igrača, izmenu i brisanje onih koji se već nalaze u tabeli;

▪ kao jedan od podataka prilikom unosa igrača, korisnik bira kud koji je kreiran i sačuvan u prozoru kud.

# Faze razvoja softvera

-specifikacija zahteva korisnika

-dizajn baze podataka

-kreiranje baze podataka

-dizajn korisničkog interfejsa (plan/opis)

-implementacija korisničkog interfejsa

-programiranje srednjeg sloja

-spajanje svih delova u celinu

-testiranje softvera

-kreiranje dokumentacije

## 3.1. Tabelarni prikaz slojeva i podslojeva aplikacije

|  |  |  |
| --- | --- | --- |
| SLOJ | PODSLOJ | SEMINARSKI RAD |
| PREZENTACIONI SLOJ | MainWindow | WPF-MainWindow.xaml |
|  | Igrac | WPF-Igrac.xaml |
| Kud | WPF-Kud.xaml |
| Aplikacija | WPF-App.xaml |
| APLIKACIONI SLOJ | Igrac | WPF-Igrac.xaml.cs |
|  | Kud | WPF-Kud.xaml.cs |
| SQL | WPF-SQL.cs |
| SLOJ BAZE PODATAKA | Baza | SQL.Connection |

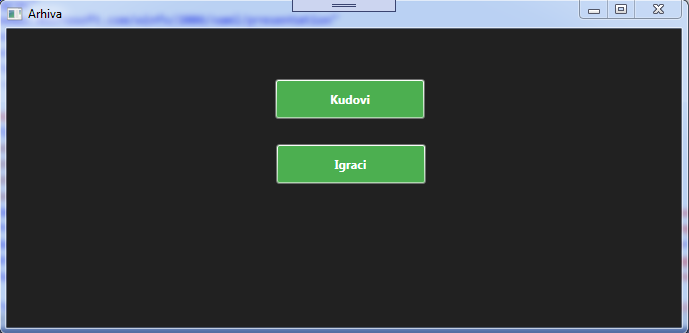
*Tabela 1- Prikaz slojeva i podslojeva aplikacije*

# Prikaz softvera

## Pokretanje softvera i ekranski prikaz

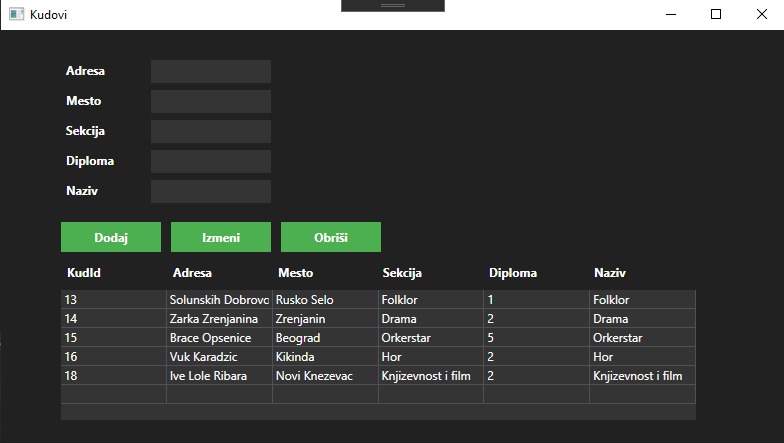
Za prikaz ove aplikacije koristi se Microsoft Visual Studio, neophodan za njeno pokretanje.

Nakon uspešnog pokretanja aplikacije dobijamo prikaz prvog prozora. Na prvom prozoru nalaze se dva dugmeta: dugme Kudovi i dugme Igraci. Prvi prozor pruža mogućnost izbora kojim podacima želimo da pristupimo.



*Slika 1- Snimak ekrana prvog prozora aplikacije Kud*

Nakon klika na jedno od dva ponudjena dugmeta, ostvara se sledeći prozor, prozor Kudovi ukoliko izaberemo to dugme.

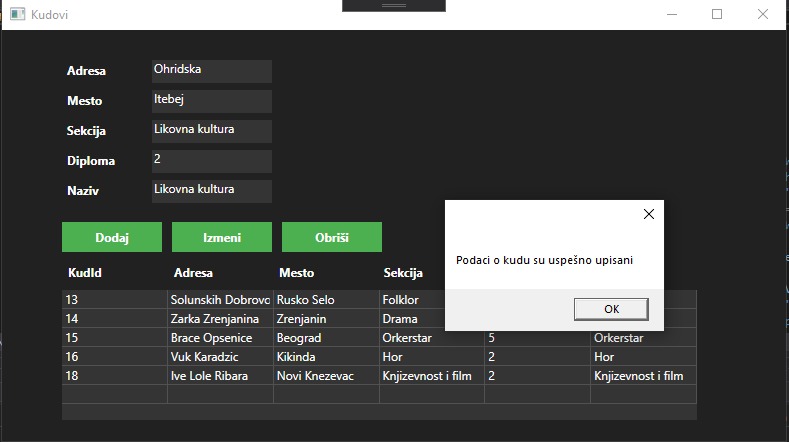


*Slika 2- Prikaz prozora Kudovi*

Ovaj prozor sadrži DataGrid u kome su prikazani svi podaci o kudovi koji su upisani u bazu podataka. Podaci koji su prikazani su KudId, Adresa, Mesto, Sekcija, Diploma, Naziv.Takodje ovaj prozor omogućava unos, izmenu i brisanje podataka. Ove akcije se obavljaju pomoćuu buttona: Dodaj, Izmeni i Obriši.

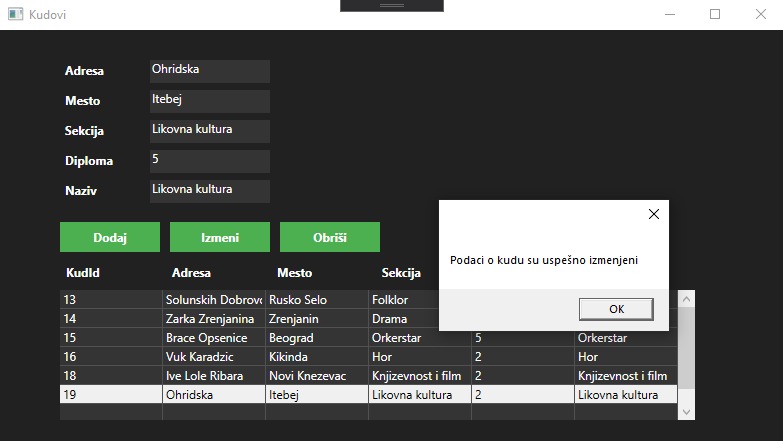
Prilikom unosa i izmene podataka popunjavaju su 5 textbox polja. Pored svakog textboxa nalazi se labela sa odredjenim nazivom i na osnovu te labele unosimo podatak u textbox.

Kada se unose podaci o novom kudu, popunjavaju se sva tražena polja i klikne se na dugme Dodaj. Nakon klika ukoliko su podaci uspešno uneti pojavljuje se poruka Podaci o kudu su uspešno upisani. Klikom na OK isti se pojavljuju u DataGrid-u.



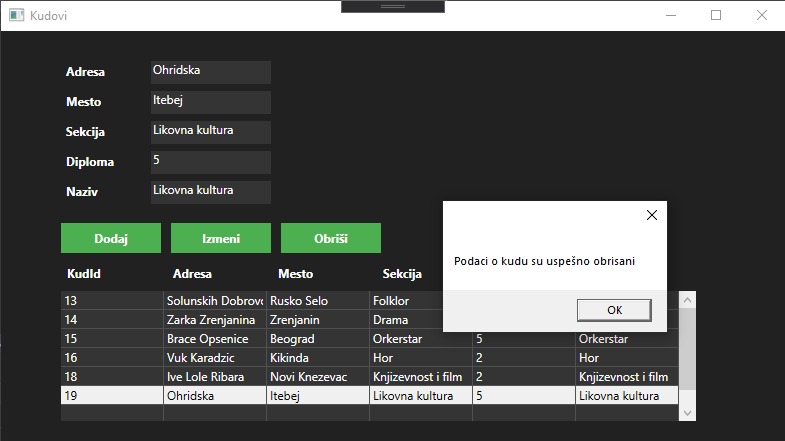
*Slika 3- Prikaz nakon dodavanja podataka*

Kada je upitanju izmena podataka, potrebno je kliknuti na red koji želimo da izmenimo i kada se postojeći podaci pojave u textbox poljima, vrši se izmena. Nakon željene izmene, potrebno je kliknuti na dugme Izmeni. Klikom na dugme pojavljuje se mali prozor sa porukom Podaci o kudu su uspešno izmenjeni. U tom prozoru potrebno je kliknuti na dugme OK, i u DataGrid-u pojaviće se izmenjeni podaci.



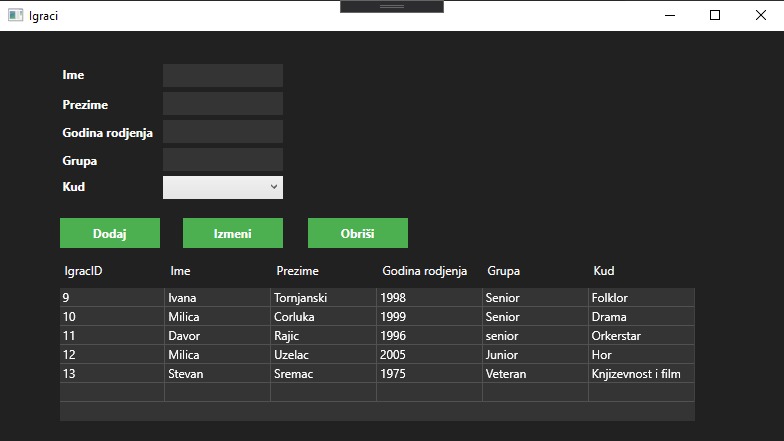
*Slika 4 – Prikaz nakon izmene podataka*

Poslednja mogućnost koju pruža je brisanja podataka iz baze. Da bi podaci bili obrisani potrebno je kliknuti na željeni red, red koji želimo da obrišemo. Potom će se podaci javiti u textbox poljima. Klikom na dugme Obriši, javlja se prozor sa porukom Podaci o kudu su uspešno obrisani.



*Slika 5 – Prikaz nakon brisanja podataka*

Treći prozor koji ova aplikacija sadrži je prozor Igraci.



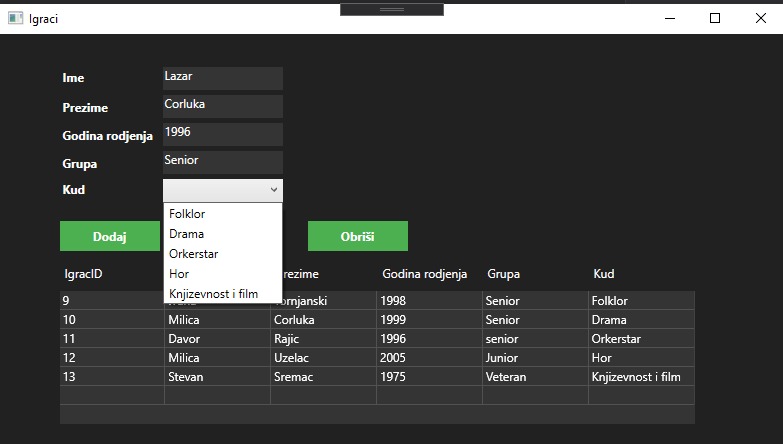
*Slika 6- Izgled prozora Igraci*

Ovaj prozor se sastoji od pet labela. Pored svake labele se nalazi textbox, osim labele Kud, pored koje se nalazi ComboBox.

Sadrži tri button-a: Dodaj, Izmeni i Obriši.

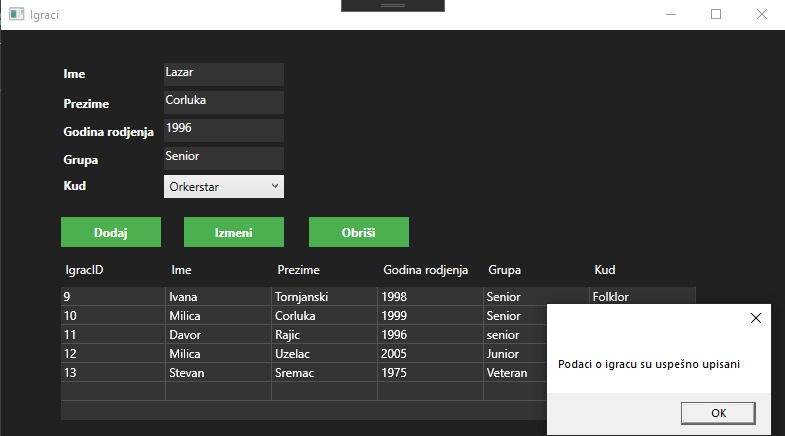
U DataGrid-u prikazani su podaci koji se već nalazi u bazi podataka.

Kada se unose novi podaci, potrebno je popuniti sva polja i u ComboBox-u izabrani postojeći Kud.



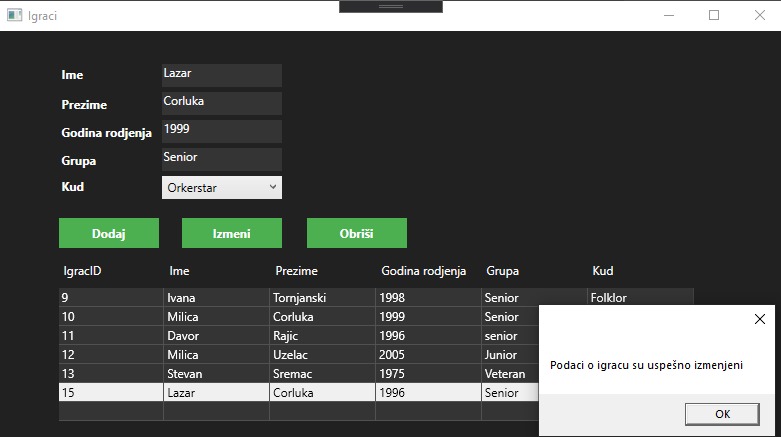
*Slika 7– Prikaz izgleda ComboBox-a*

Nakon popunjenih polja i odabira iz ComboBox-a, podaci se dodaju klikom na dugme Dodaj. Zatim se prikaže prozor sa porukom Podaci o igracu su uspešno upisani.



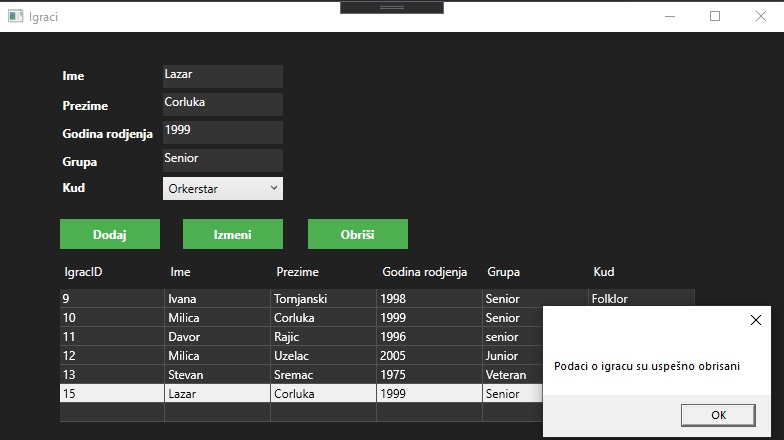
*Slika 8 - Prikaz nakon dodatih podataka*

Izmena podataka se vrši klikom na željeni red i kada se podaci pojave u textbox poljima i izvrši se izmena potrebno je kliknuti na dugme izmeni. Zatim se pojavljuje poruka Podaci o igracu su uspešno izmenjeni.



*Slika 8- Prikaz prozora nakon izmene podataka*

Podaci se briši klikom na drugme Obriši. Prvo je potrebno odabrati odredjeni red i potom klikom na dugme dobijamo poruku Podaci o igracu su uspešno obrisani.



*Slika 9 – Prikaz nakon brisanja podataka*

# Opis implementacije

## Opis primenjenih tehnologija

### Microsoft Visual Studio

Microsoft Visual Studio 2019 je Majkrosoftovo integrisano razvojno okruženje (IDE- Integrated development enviroment). Koristi se za razvoj kompjuterskih programa, sajtova, veb sajtova, veb aplikacija kao i mobilnih aplikacija. Visual Studio koristi različite platforme za kreiranje projekata, poput Windows Forms Application i Class Library. Podržava 36 raličitih programskih jezika. Visual Studio uključuje debugger koji radi i kao ispravljač grešaka na nivou izvora i kao ispravljač na nivou mašine.

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. [1]

### 5.1.2 C# programski jezik

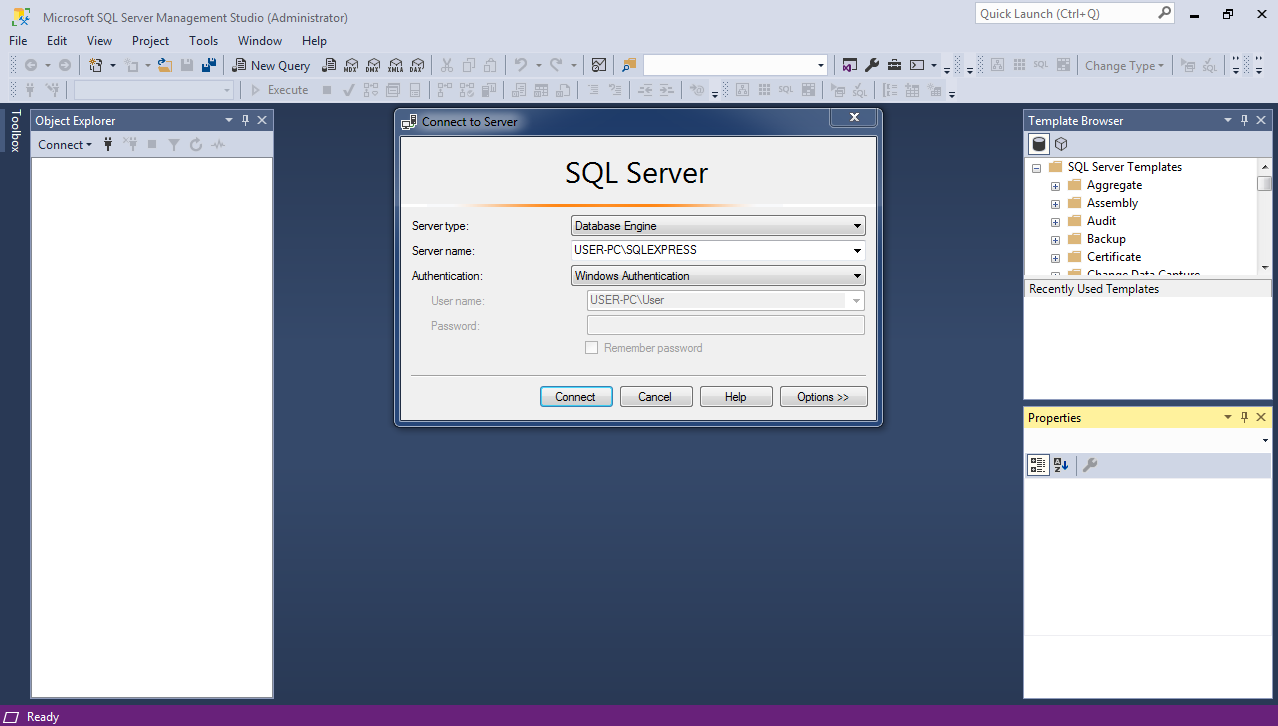
C# (C Sharp) programski jezik je jedan od mlađih programskih jezika nastao 2002. godine kao sastavni deo MS .NET Framework-a 1.0. C# je objektno orijentisani programski jezik namenjen razvoju aplikacija na .NET Framework platformi. Pošto je objektno orijentisan programski jezik, prvenstveno je sagrađen 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.[2]

### 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 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.[3]

### Microsoft SQL Server Management Studio 18

SQL Server Management Studio (SSMS) je softverska aplikacija koja se koristi za konfigurisanje, upravljanje i administraciju svih komponenti u Microsoft SQL Serveru. Alat uključuje uređivače skripti i grafičke alate koji rade sa objektima i komponentama servera.[4]



*Slika 10- Početni ekran pokretanje Microsoft Sql Server Management Studia 18*

Kada se pokrene alat, potrebno je najpre konektovati se na server. Ukoliko je naziv servera dobar, moguće je izvršiti konekciju, klikom na dugme Connect. Baza se kreira desnim klikom na Databases→New Database. Polje Databases nalazi sa u prozoru Object Explorer (Slika 4.). Tabele unutar baze kreiraju se klikom na Tables→New→Table.

## 5.2 GitHub

„GitHub je veb-baziran hosting servis za kontrolu verzije, Git. Pruža Git funkcionalnosti: distribuiranu kontrolu revizija i menadžment izvornog koda (Source Control Management - SCM), dodajući dodatne funkcije.

Za razliku od Git-a, koji je striktno alat koji se koristi iz komandne linije, GitHub pruža veb grafički interfejs, radnu površinu i mobilnu integraciju. Takođe pruža kontrolu pristupa i nekoliko funkcija za saradnju, kao što su praćenje grešaka, zahteve za dodavanje novih karakteristika, upravljanje zadacima i mogućnost pravljenja viki dokumentacije za svaki projekat.GitHub pruža planove za privatna skladišta kao i besplatne naloge, koji se obično koriste kao hostovi za sofverske projekte otvorenog koda.

Projektima na GitHub-u se pristupa i manipuliše koristeći standardni Git interfejs iz komandne linije preko kojeg su dostupne i sve standardne Git komande. GitHub dopušta registrovanim i neregistrovanim korisnicima da pregledaju javna skladišta na sajtu.

GitHub sajt pruža funkcije slične društvenim mrežama kao što su: dovodi (engl. feeds), pratioci (engl. followers), viki (koristeći softver Gollum) i grafikone saradnje koji ukazuju kako programeri rade na svojim verzijama (&quot;račvama&quot;) skladišta i koja račva je najnovija.

Korisnik mora da napravi nalog kako bi doprineo stranici, ali javna skladišta mogu biti pregledana i preuzeta od strane bilo koga. Sa registrovanim korisničkim nalogom, korisnici mogu da diskutuju, upravljaju skladištima, prave nova skladišta, postavljaju doprinose drugim skladištima i pregledaju izmene u kodu.“

„GitHub se najviše koristi za izvorne kodove. Osim izvornog koda, GitHub podržava sledeće formate i funkcije:

- Dokumentaciju, uključujući automatski generisane README fajlove u raznim Markdown fajl formatima (pogledati README fajlovi na GitHub-u),

- Praćenje problema (uključujući zahteve za novim karakteristikama) sa labelama, prekretnicama, zastupnicima i pretraživačem,

- Viki podrška,

- Zahteve za pregled izmena sa recenzijama koda i komentarima,

- Istorija izvršavanja,

- Grafikoni (puls, saradnici, frekvencije koda, mreže, članovi),

- Adresar integracija,

- E-mail notifikacije,

- Opcija prijave za notifikacije o nekome, koristeći znak @,

- Emoji podrška,

- Ugnježdene liste zadataka unutar fajlova,

-Vizuelizacija geoprostornih 3D podataka koji se mogu pregledati koristeći novointegrisani STL fajl pregledač koji otvara fajlove na 3D platnu,

-Photoshop-ov nativni PSD format se može pregledati i porediti sa prethodnim verzijama istog fajla.“

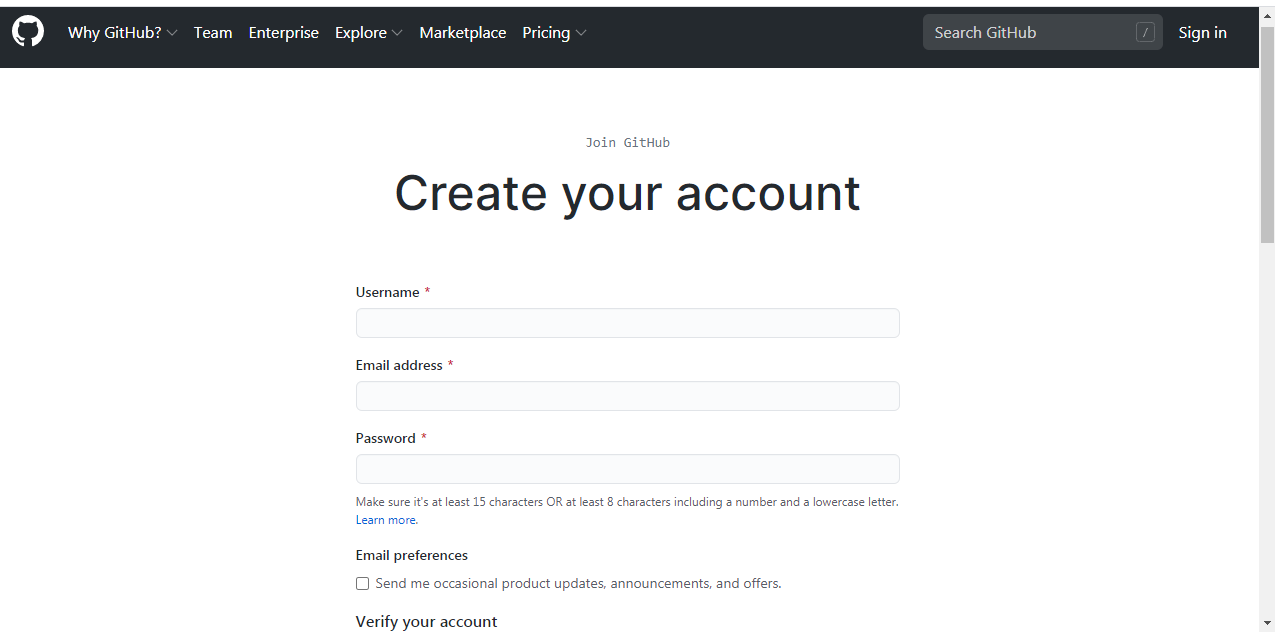
### 5.2.1 Kreiranje naloga

Nalog na GitHubu može se kreirati koristeći jedan od dva sledeća linka:  
▪<https://github.com/>

▪<https://github.com/join?ref_cta=Sign+up&ref_loc=header+logged+out&ref_page=%2F&source=header-home>

Prilikom kreiranja potrebno je uneti korisničko ime (eng. username), emajl adresu i lozinku (eng. password)

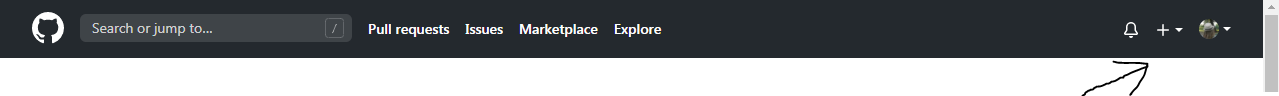
Na slici ispod je prikazana stranica na kojoj kreiramo nalog.



*Slika 11 – Kreiranje GitHub naloga*

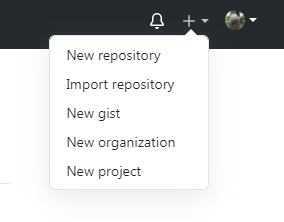
**5.2.2 Kreiranje repozitorijuma**

Svaki korisnik ima mogućnost kreiranja repozitorijuma. Novi repozitorijum se kreira klikom na znak plus koji se nalazi u gornjem desnom uglu.



*Slika 12-Prikaz lokacije znaka plus za kreiranje repozitorijuma*

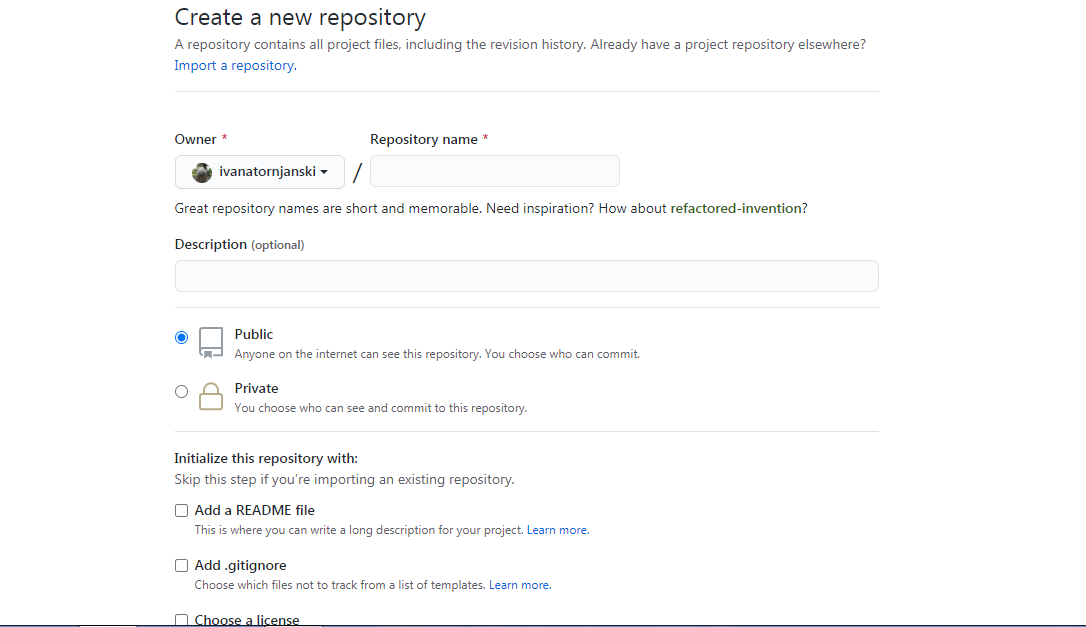
Zatim se u padajućem meniju izabere New repository.



*Slika 13– Izgled padajućeg menija*

Stranica za kreiranje repozitoriju i podaci koje je potrebno uneti prilikom kreiranja istog prkazani su na sledećoj slici.

Konkretno u ovom seminarskog radu repozitorijum je nazvan PUPIN-KUD-2020.

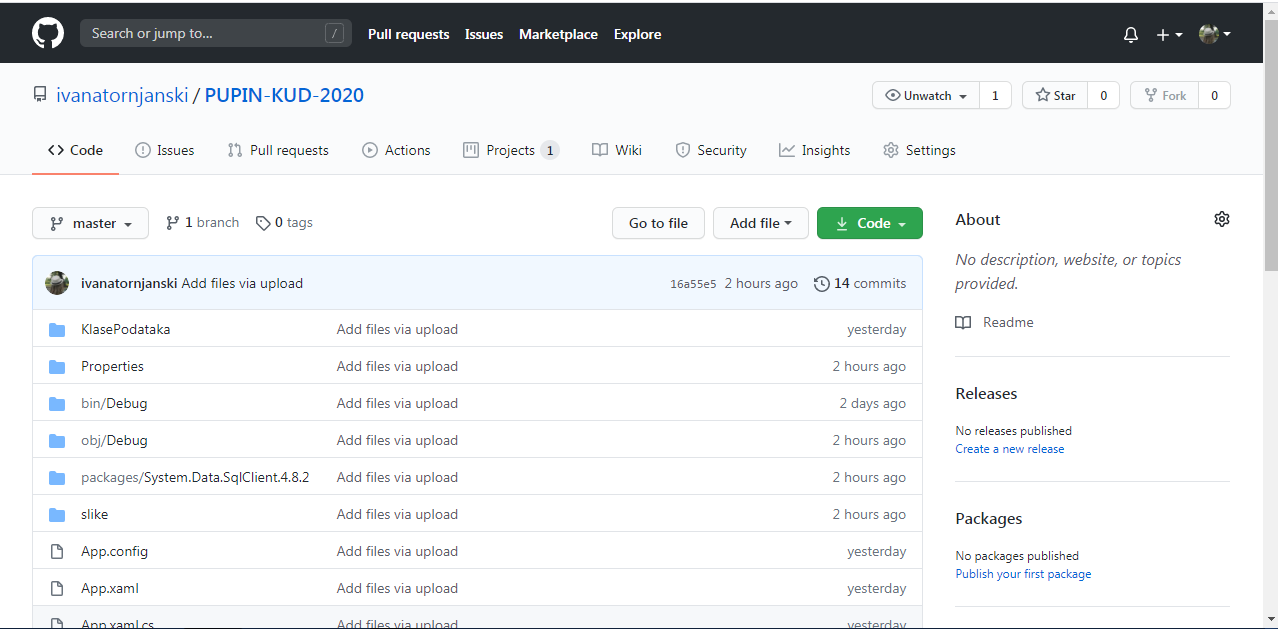


*Slika 14- Stranica na kojoj se kreira repozitorijum*

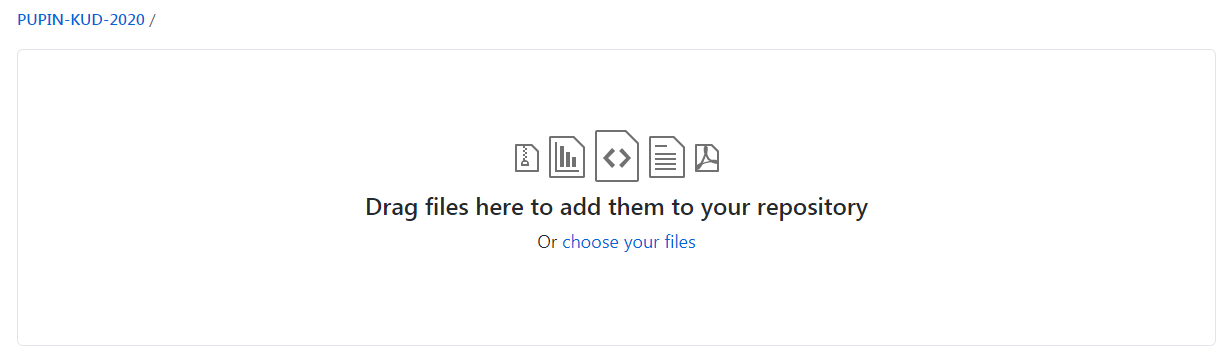
Repozitorijum može da sadrži sledeće fajlove:

* README,
* LICENSE
* .gitignore.

Nakon kreiranja dobija se sledeći prikaz:

*Slika 15- Izgleda repozitorijuma PUPIN-KUD-2020*

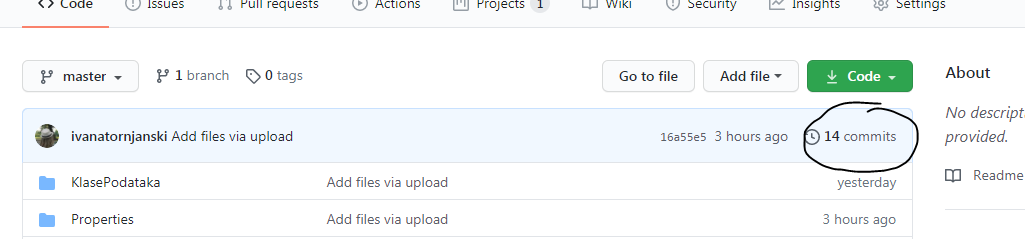
Na slici iznad takodje se mogu videti fajlovi koji su ubačeni u repozitorijum. Fajlovi se mogu dodati klikom na Add file. Add file dugme pruža sve opcija: Create new file i Upload files. Fajlove je moguće dodati biranjem sopstvenih iz foldera ili prevlačenjem.



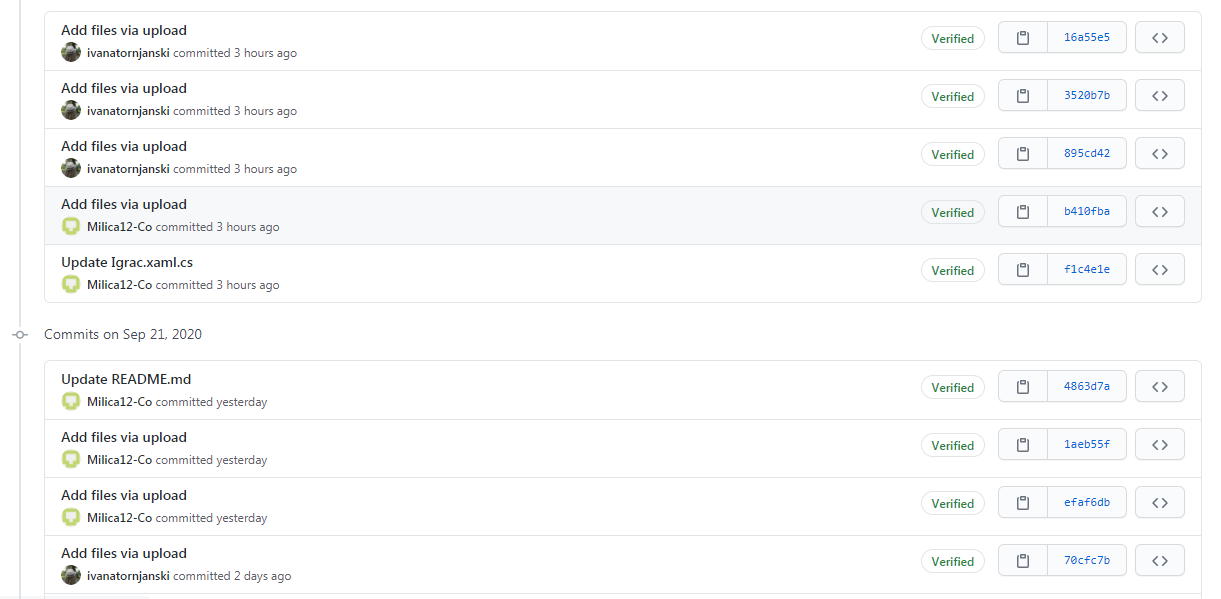
*Slika 16– Dodavanje fajlova*

Repozitoriju pruža uvid u akcije koje su preduzete od strane članova tima.

Potrebno je kliknuti na dugme koje je obeleženo na sledećoj slici.



*Slika – Lokacija Commits-a*

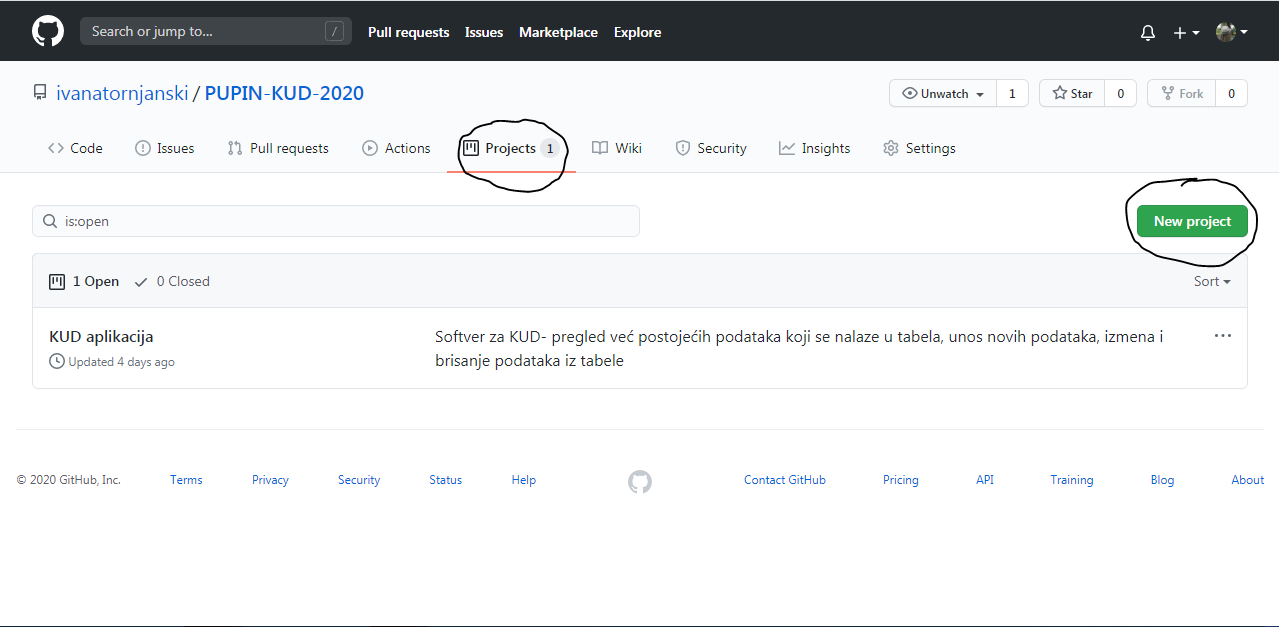


*Slika – Prikaz rada korisnika*

### 5.2.3 Kreiranje projekta

U repozitorijumu postoji mogućnost kreiranja projekta, klikom na Projects, a zatim kada se prikaže sledeća stranica potrebno je kliknuti New Project.

U izradi ovog seminarskog rada kao naziv projekta postavljano je KUD aplikacija.



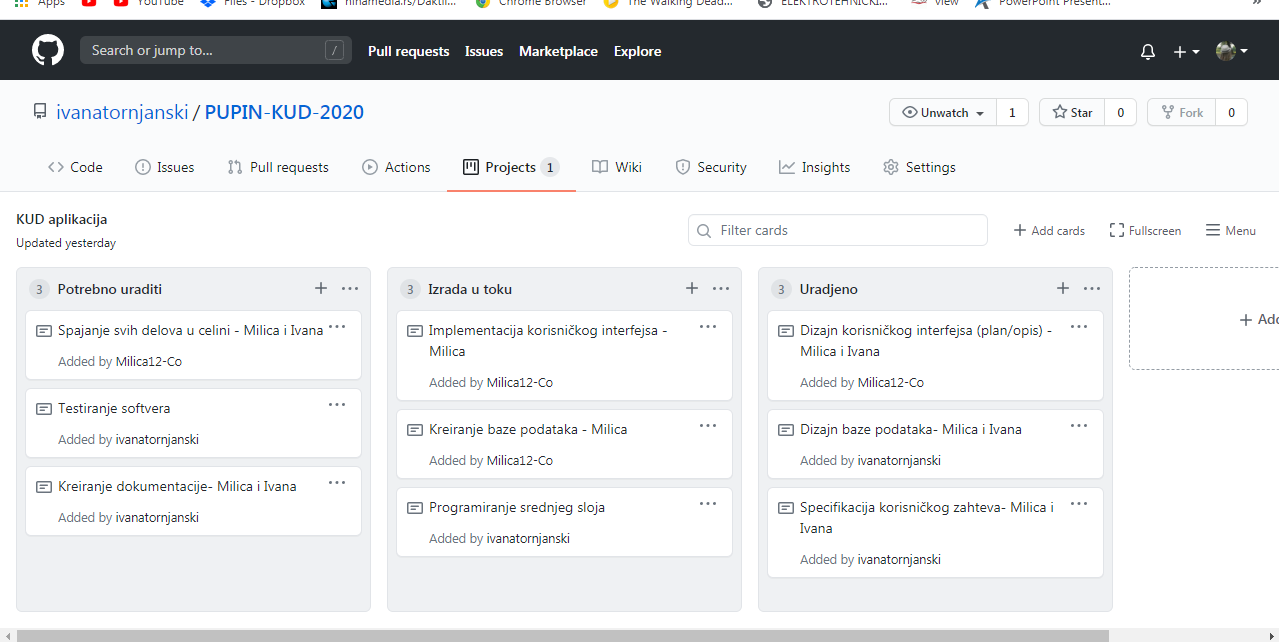
*Slika 17– Prikaz opcija navedenih u tekstu iznad slike*

GitHub pruža mogućnost planiranja izrade projekta, odnosno kreiranje kartica radi lakše organizacije i podele poslova.

Postoje tri tipa kartica:

* To Do (ono što je potrebno uraditi),
* In Progress (ono na čemu se trenutno radi),
* Done (ono što je uradjeno).

U toku izrade ovog seminarskog rada zabeležen je sledeći snimak ekrana na kome je prikazano tadašnje stanje podataka u karticama.



*Slika 18- Prikaz kartica*

Nakon svih ispunjenih zadataka i završenog seminrskog rada, sve beleške prebačene su u karticu pod nazivom Uradjeno.

Kada se govori konkretno o ovom radu, kartice su nazvana na sledeći način:

* kartica To Do nazvana je Potrebno uraditi,
* kartica In Progress nazvana je Izrada u toku,
* kartica Done nazvana je Uradjeno.

## 5.3 Testiranje softvera

Testiranje izvršavanja metode klase Kud.xaml.cs

komanda.Parameters.AddWithValue("@Diploma", txtDiploma.Text);

Provera da li je podatak diploma ispravno unet. Ovaj podatak je potrebno uneti u obliku cifara, a ne slovima. U bazi podataka za ovaj podatak postavljen je tip INT i zbog toga ako se ovaj podatak popuni tipom string, podatak će biti pogrešno unesen.

Ako u textbox koji se nalazi pored labele diploma upišemo slovima na primer pet, kod ovog testiranja javiće se greška u bazi podataka.

# Baza podataka

Za potrebe ovog seminarskog rada kreirana je baza podataka pod nazivom KUDBaza. Ova baza podataka kreirana je u Microsoft SQL Server Management Studiu.

Sastoji se od dve tabele:

* Igrac,
* KUD.

Nakon kreiranja tabela, u svaku tabelu dodati su atributi.

Tabela Igrac ima sledeće atribute:  
ID, Ime, Prezime, GodRodj, Grupa, Kud.

Tabela KUD ima sledeće atribute:  
ID, Adresa, Mesto, Sekcija, Diploma, Naziv.

Svaka tabela mora da ima primarni ključ. Kao primarni ključ potrebno je odabrati jedan atributi. U ove dve tabela, primarni ključ je ID (u oba slučaja).

Tabela KUD:

USE [master]

GO

CREATE DATABASE [KUDBaza]

GO

USE [KUD]

GO

CREATE TABLE [dbo].[KUD](

[ID] [int] NOT NULL identity(1,1) PRIMARY KEY,

[Adresa] [nvarchar](60) NOT NULL,

[Mesto] [nvarchar](60) NOT NULL,

[Sekcija] [nvarchar](100) NOT NULL,

[Diploma] [int] NOT NULL,

[Naziv] [nvrachar](60) NOT NULL,

GO

Tabela Igrac:

USE [Igrac]

GO

CREATE TABLE [dbo].[KUD](

[ID] [int] NOT NULL identity(1,1) PRIMARY KEY,

[Ime] [nvarchar](60) NOT NULL,

[Prezime] [nvarchar](60) NOT NULL,

[GodRodj] [nvarchar](100) NOT NULL,

[Grupa] [nvarchar](30) NOT NULL,

[Kud] [nvrachar](30) NOT NULL,

GO

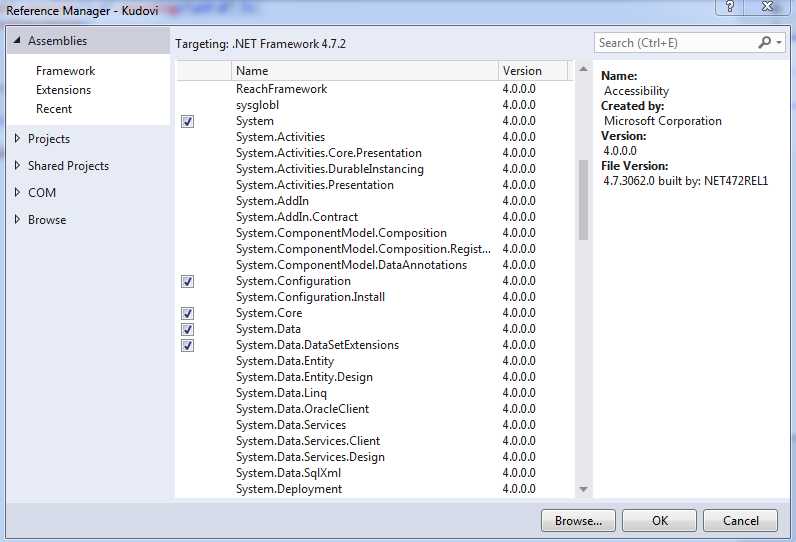
## 6.1 Konekcioni string

Konekcioni string se kreira nakon baze podataka.

|  |
| --- |
| <?xml version="1.0" encoding="utf-8" ?>  <configuration>  <startup>  <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.7.2" />  </startup>  <connectionStrings>  <add name="connKudovi" connectionString="Data Source=DESKTOP-SRSSF3P\SQLEXPRESS;Initial Catalog=KUDBaza;Integrated Security=True"/>  </connectionStrings>  </configuration> |

*Listing 1- Konekcioni string, App.config*

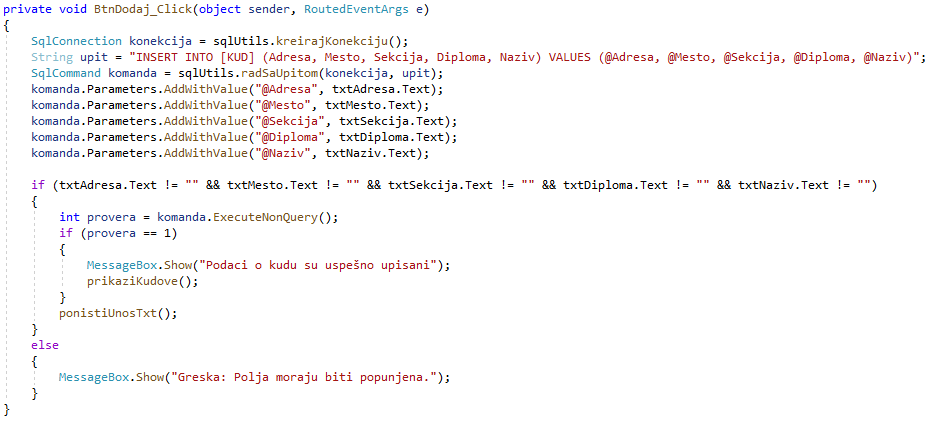
Potrebno je da se dodati referenca da prozori mogu da radesa konfiguracionom bazom. Referenca se dodaje klikom na References – Add Reference i otvara se Reference Manager u kojemje potrebno da se čekira System Configuration.

**

*Slika 19- Dodavanje reference*

# Ključni delovi koda

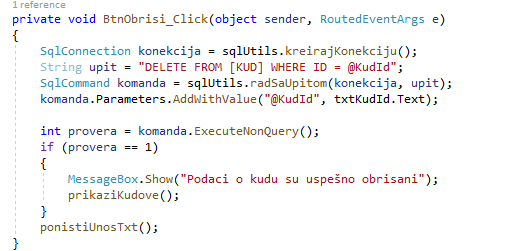
U prozoru Kud moguće je uneti podatke o novom kudu korišćenjem datih sql komandi čime se podaci direktno upisuju u bazu.



*Listing 2- Dodavanje novog kuda u bazu podataka*



*Listing 3- Izmena postojećeg kuda u bazu podataka*



*Listing 3- Brisanje postojećeg kuda iz baze podataka*

# Zaključak

Ovaj seminarski rad upoznaje nas sa GitHub-om koji ljudima omogućava da uče, dele svoje znanje i saradjuju kako bi izgradili novi softver. Pomoću aplikacije KUD savladano je korišćenje i snalaženje na GitHubu.

# Literatura

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

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

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

[4]: <https://en.wikipedia.org/wiki/SQL_Server_Management_Studio>

[5]: <http://www.tfzr.uns.ac.rs/Content/files/0/GITHUB%20uputstvo%20FINALNO.pdf>

[6]: <http://www.tfzr.uns.ac.rs/Content/files/0/RSOKVezbe%2020192020%20-%20dokument%20FINALNO.pdf>