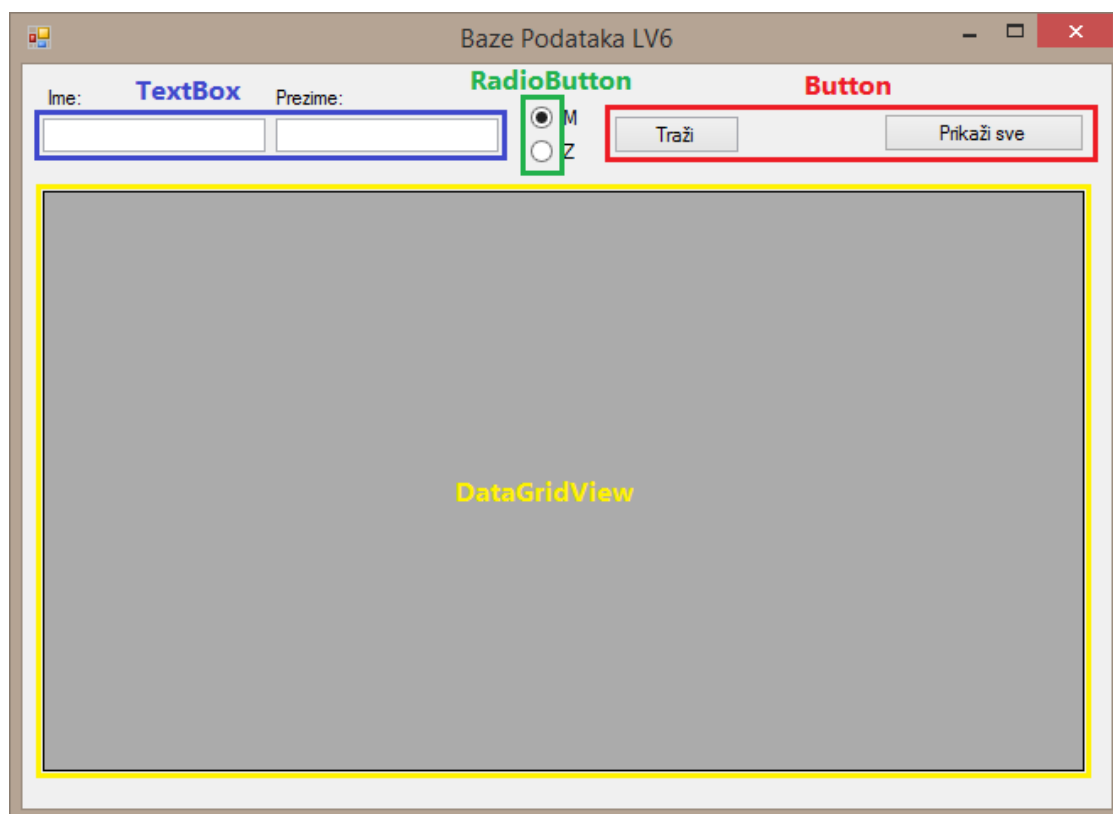


1. Povezivanje C# aplikacije s bazom podataka

Česta je potreba za povezivanjem aplikacija napisanih u različitim programskim jezicima (C++, C#, Java i dr.) s bazom podataka, kako bi iz takvih aplikacija izravno manipulirali podacima iz baze podataka (dohvaćanje, izmjena, umetanje i dr.). U ovoj vježbi će na primjeru aplikacije napisane u programskom jeziku C# biti prikazani koraci u postupku povezivanja aplikacije s bazom podataka (**stuslu**) na serveru (**161.53.201.59**), te osnovni postupci za manipulaciju podacima iz aplikacije.

Za potrebe vježbe izrađen je predložak projekta u razvojnom okruženju Microsoft Visual Studio 2012 (Programski predložak 1) koji sadrži sve preduvjete za spajanje na bazu podataka. Projekt sadržava grafičko sučelje sa svim formama i kontrolama za upravljanje aplikacijom (*Button*, *RadioButton*, *TextBox*) i prikaz podataka iz baze (*DataGridView*) kao što je vidljivo na slici:



Projekt sadrži i pozive potrebnih biblioteka od kojih je za ovu vježbu posebno važna biblioteka *System.Data.SqlClient*, koja u sebi sadrži klase potrebne za povezivanje sa SQL Serverom.

Slijede naredbe koje je potrebno dodati na postojeći predložak projekta koje će omogućiti spajanje na server **161.53.201.59** te dohvaćanje svih podataka iz tablice **student** iz baze podataka **stuslu**. Naredbe izvršavamo na pritisak tipke *Prikaži sve* (u predlošku se naziva *btnSve*).

Prva naredba odnosi se na kreiranje konekcije (*SqlConnection*) prema serveru. Za određivanje servera na koji se želimo spojiti s određenim pristupnim podacima za autentifikaciju piše se tzv. *connection string* koji sadrži sljedeće podatke: adresa servera (*Data source*), naziv baze podataka (*Initial Catalog*), naziv korisnika (*User ID*) i lozinka korisnika (*Password*).

```
SqlConnection conn = new SqlConnection("Data Source=161.53.201.59;Initial Catalog=stuslu;User ID=stuslu;Password=stuslu");
```

Potom je potrebno otvoriti prethodno definiranu konekciju naredbom:

```
conn.Open();
```

Nakon otvorene konekcije u novu varijablu (*statement*) tipa *string* ćemo spremiti SQL upit koji želimo izvršiti na bazi podataka **stuslu**:

```
string statement = "SELECT * FROM student";
```

Potom je potrebno koristiti klasu *SqlDataAdapter* koja je predstavljena kao kombinacija SQL upita i konekcije na bazu podataka. *SqlDataAdapter* služi kao posrednik za izvršavanje upita i dohvaćanje podataka iz baze ili promjenu podataka u bazi na koju smo se spojili određenom konekcijom.

```
SqlDataAdapter dataAdapter = new SqlDataAdapter(statement, conn);
```

Ukoliko su upitom dohvaćeni neki podaci iz baze, moguće je njima popuniti instancu klase *DataTable* kako bi ih kasnije lakše prikazali u aplikaciji. Popunjavanje dohvaćenim podacima izvršava se sljedećim naredbama:

```
DataTable dt = new DataTable();  
dataAdapter.Fill(dt);
```

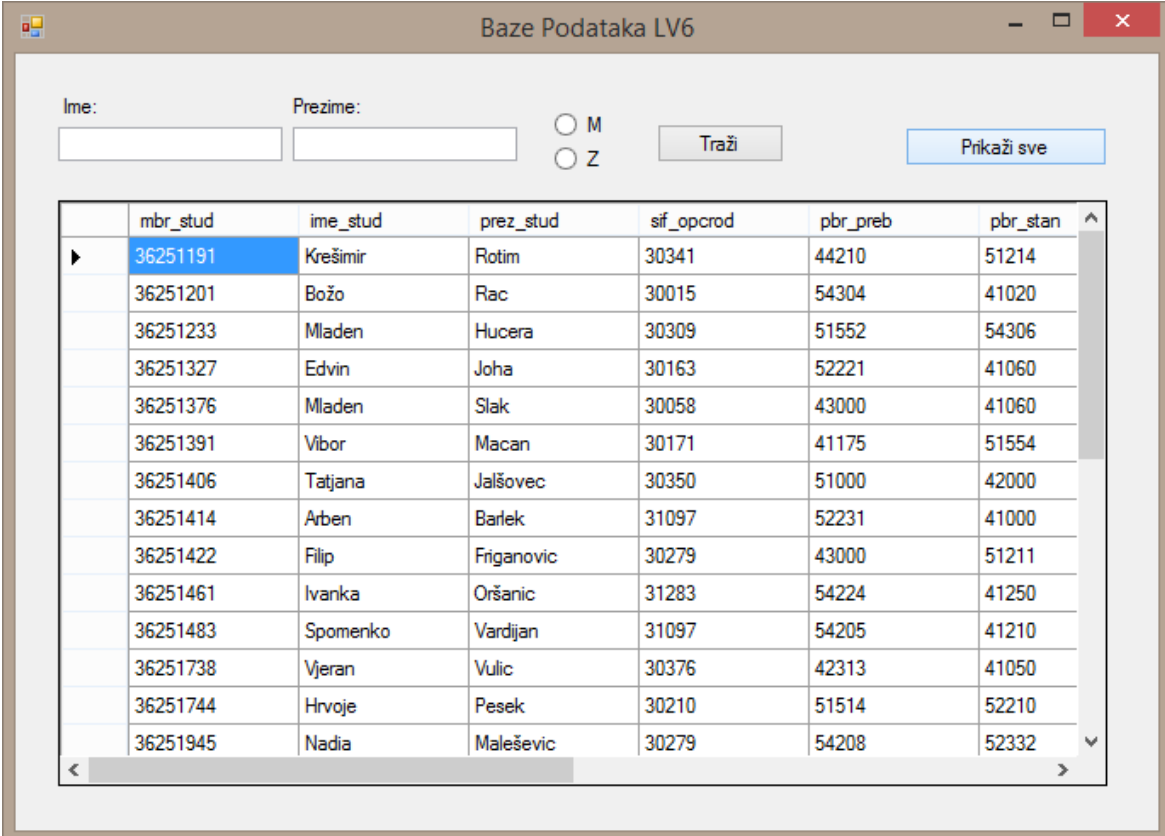
Sada je još samo potrebno prikazati dohvaćene podatke u aplikaciji pomoću ranije prikazane kontrole *DataGridView* koja se u predlošku zove *dgvPodaci*. To je moguće učiniti postavljanjem prethodno popunjene instance *DataTable* *dt* kao izvora podataka (*DataSource*) za *DataGridView* *dgvPodaci*.

```
dgvPodaci.DataSource = dt;
```

Na kraju je još moguće zatvoriti konekciju sljedećom naredbom:

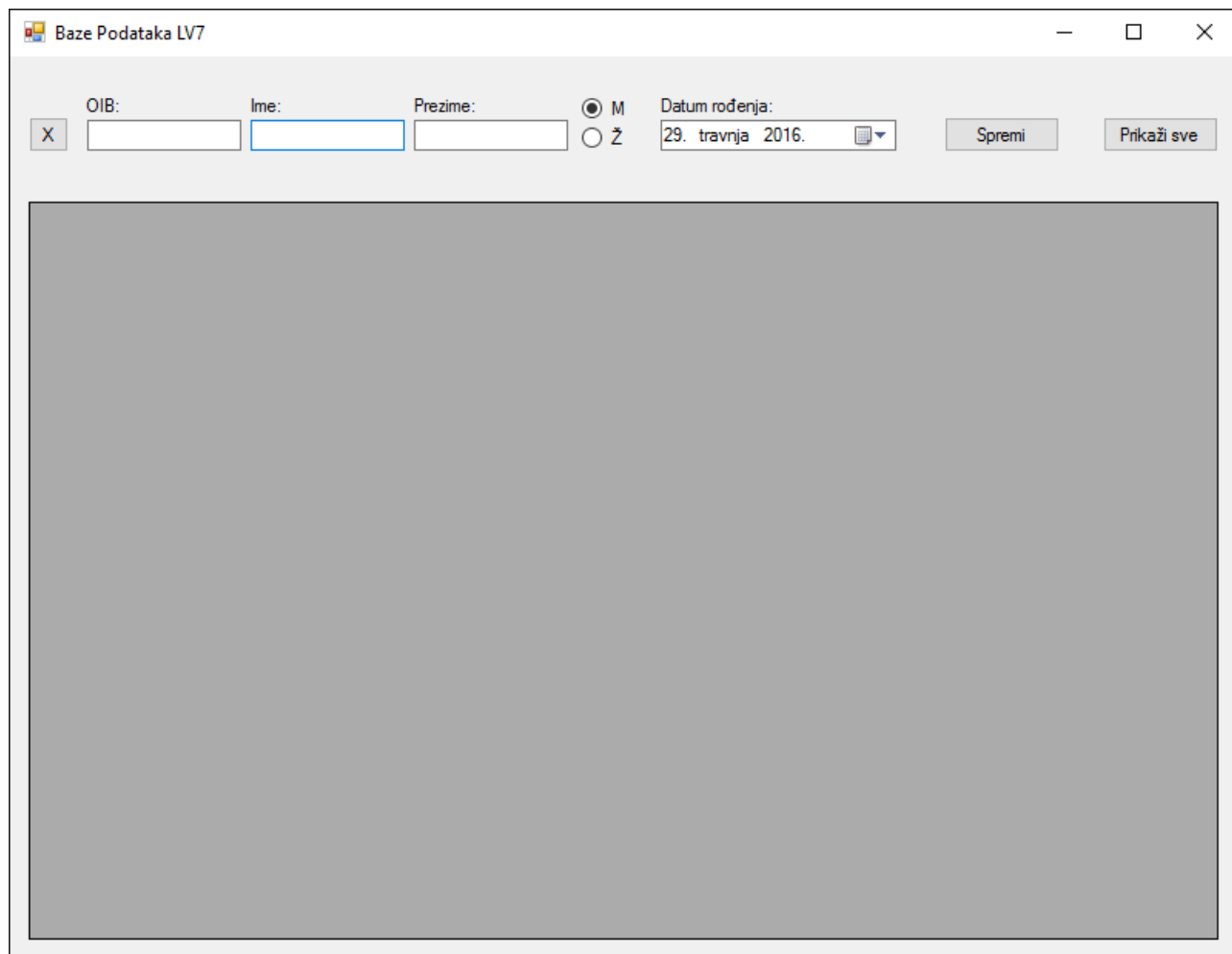
```
conn.Close();
```

Nakon pokretanja programa i pritiska na tipku *Prikaži sve* izvršava se dio koda koji je prethodno opisan te se u *DataGridView dgvPodaci* prikazuje rezultat upita kao što je prikazano na slici:



	mbr_stud	ime_stud	prez_stud	sif_opcrood	pbr_preb	pbr_stan
▶	36251191	Krešimir	Rotim	30341	44210	51214
	36251201	Božo	Rac	30015	54304	41020
	36251233	Mladen	Hucera	30309	51552	54306
	36251327	Edvin	Joha	30163	52221	41060
	36251376	Mladen	Slak	30058	43000	41060
	36251391	Vibor	Macan	30171	41175	51554
	36251406	Tatjana	Jalšovec	30350	51000	42000
	36251414	Arben	Barlek	31097	52231	41000
	36251422	Filip	Frganovic	30279	43000	51211
	36251461	Ivanka	Oršanic	31283	54224	41250
	36251483	Spomenko	Vardijan	31097	54205	41210
	36251738	Vjeran	Vulic	30376	42313	41050
	36251744	Hrvoje	Pesek	30210	51514	52210
	36251945	Nadia	Maleševic	30279	54208	52332

U programskom predlošku 2 nalazi se aplikacija s korisničkim sučeljem prikazanim na sljedećoj slici:



Doradom programskog predloška bit će potrebno povezati aplikaciju s bazom podataka **student** na serveru **161.53.201.59** te osim naredbe SELECT izvršiti i naredbe za manipulaciju podacima INSERT i UPDATE. Projekt u programskom predlošku sadrži sve potrebne forme i kontrole za upravljanje aplikacijom (*Button*, *RadioButton*, *TextBox*, *DateTimePicker*). Kao i u prethodnoj vježbi, projekt sadrži pozive potrebnih biblioteka uključujući *System.Data.SqlClient* koja u sebi sadrži klase potrebne za povezivanje sa SQL Serverom.

Za razliku od prethodne vježbe gdje smo na slanje upita (SELECT naredbe) dobivali tablicu kao rezultat, u slučaju INSERT ili UPDATE naredbe nema tablice kao rezultata upita pa je i postupak slanja naredbi na izvršavanje nešto drugačiji što će biti pokazano u nastavku.

Kao i u prethodnoj vježbi potrebno je prvo definirati podatke za spajanje na bazu podataka i otvoriti konekciju:

```
SqlConnection conn = new SqlConnection("Data Source=161.53.201.59;Initial  
Catalog=student;User ID=student;Password=student");
```

```
conn.Open();
```

Potom je potrebno zapisati naredbu u obliku stringa. U ovom primjeru zapisat ćemo INSERT naredbu koja će dodati osobu u tablicu *osobe* na bazi podataka *student*:

```
String statement = "INSERT INTO osobe  
VALUES('98765432109','Pero','Perić','M','1/5/2000')";
```

S obzirom da INSERT naredba neće vratiti tablicu kao rezultat neće biti potrebno koristiti *DataAdapter* kao u prethodnoj vježbi nego će se za potrebu slanja naredbe za izvršavanje na bazi podataka koristiti *SqlCommand* kojeg definiramo i šaljemo na sljedeći način:

```
SqlCommand cmd = new SqlCommand(statement, conn);  
  
cmd.ExecuteNonQuery();
```

Na ovaj način definiramo da će *cmd* biti SQL naredba koju smo zapisali u string *statement*, a šalje se preko konekcije *conn*. Slanje izvršavamo pozivom funkcije *ExecuteNonQuery()* za definiranu naredbu *cmd*. Na kraju je potrebno samo zatvoriti konekciju *conn*:

```
conn.Close();
```

Istim postupkom moguće je izvršiti i UPDATE naredbu na bazi podataka tako da *String statement* zamijenimo s naredbom oblika:

```
String statement = "UPDATE osobe SET ime='Perica' WHERE OIB='98765432109'";
```

2. Zadaci

1. Potrebno je proširiti programski predložak uz laboratorijsku vježbu (*Baze_LV5_predlozak1.zip*), na način da se u aplikaciji pritiskom na tipku *Traži* (u predlošku se naziva *btnTraži*), izvrši pretraga studenata u tablici **student** (server: **161.53.201.59**; baza: **stuslu**) po upisanom imenu i/ili prezimenu, te odabranom spolu. U upitu na bazu podataka potrebno je koristiti podatke koje korisnik u aplikaciji unese u *TextBox* za ime i prezime (u predlošku se zovu *txtIme* i *txtPrezime*), te odabir spola ovisno o označenom *RadioButtonu* (u predlošku su to *rbM* i *rbZ*). Rezultat pretrage potrebno je prikazati kao tablicu u *DataGridView dgvPodaci*.
2. Potrebno je proširiti programski predložak *Baze_LV5_predlozak2.zip* s dvije funkcionalnosti:
 - a. Pritiskom na tipku *Prikaži sve* potrebno je spojiti se na bazu podataka **student** (server: **161.53.201.59**; username: **student**; password: **student**) i u *DataGridView dgvPodaci* ispisati sve osobe iz tablice **osobe** poredane abecedno po prezimenu.

- b. Pritiskom na tipku *Spremi* potrebno je spremiti novu osobu s podacima (OIB, ime, prezime, spol, datum rođenja) koje je korisnik upisao u sučelju aplikacije (*txtOIB*, *txtIme*, *txtPrezime*, *rbM*, *rbZ*, *txtDatum*).

3. Assignments

1. Expand the program from the file *Baze_LV5_predlozak1.zip* so that by clicking the button „Trazi” (named *btnTrazi*) a database search is conducted by the first and/or last name and gender of the student (table *student*, server: 161.53.201.59, database *stuslu*). In the SQL query the data entered in the textboxes for the first and last name (named *txtIme* and *txtPrezime*) as well as the chosen gender (from the radio buttons *rbM* and *rbZ*) have to be used. The result of the query must be displayed in a table form within the *DataGridView* object named *dgvPodaci*.
2. Expand the program from the file *Baze_LV5_predlozak2.zip* with the two additional functionalities:
 - a. By pressing the button „Prikaži sve”, a connection to the database *student* has to be established (server: 161.53.201.59, username: *student*, password: *student*) and then all the people from the table „*osobe*” have to be shown in the object of the class *DataGridView* named *dgvPodaci*. The people should be ordered by the ascending order of their last name.
 - b. By pressing the button „Spremi”, a new person has to be stored in the database. The data used (personal number, first name, last name, gender, date of birth) must be the one the user entered in the application interface (*txtOIB*, *txtIme*, *txtPrezime*, *rbM*, *rbZ*, *txtDatum*).