**VISOKA ŠKOLA STRUKOVNIH STUDIJA ZA INFORMACIONE TEHNOLOGIJE**

****

**Programski alati**

Objašnjenje Aplikacije

**Agencija za Nekretnine “Moj Dom”**

Predmetni nastavnik: Student:

Svetlana Anđelić Bogdan Rajković 249/15

Datum predaje 19.03.2018

**Beograd**

**Mart,2018.**

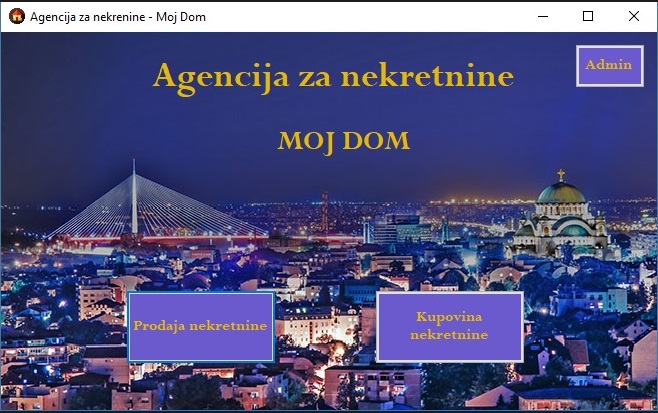
SADRŽAJ:

# VERBALNI OPIS APLIKACIJE

Aplikacija je rađena za potrebe agencije za nekretnine.Povezana je sa SQL Bayom podatake u kojoj ima 5 povezanih tabela. U aplikaciji je moguće unos podataka o prodavcu nekretnine, o samoj nekretnini, isto tako kupac moze uneti svoje podatke, dok administrator moze unositi, menjati ili brisati agente preko poseben forme. Aplikacija ima 4 forme.

**OPIS APLIKACIJE SA KORISNIČKE STRANE**

Startovanjem apliakcije otvara se forma 1. :



Forma 1:

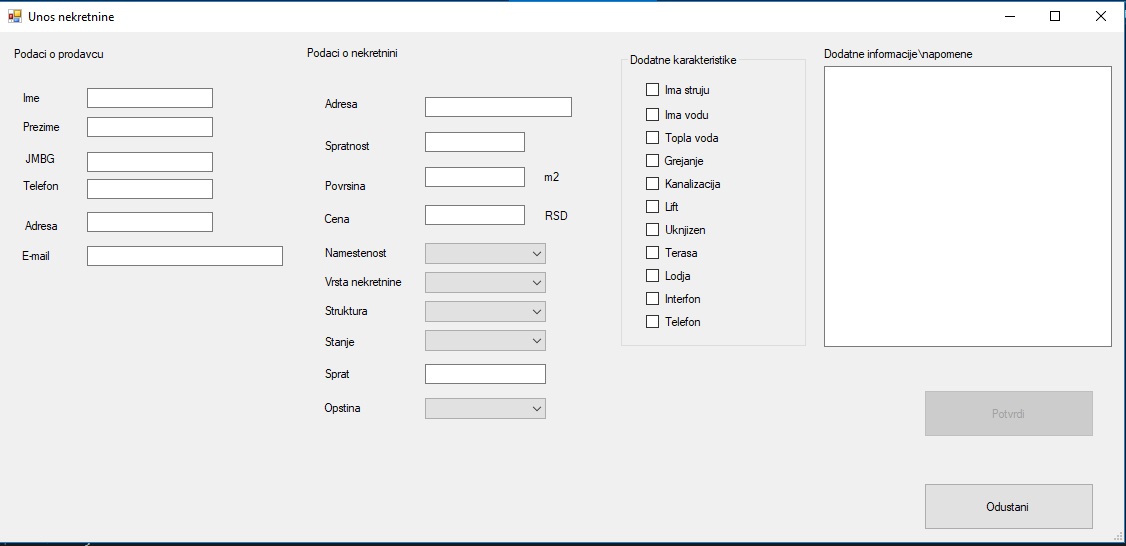
Postoje 3 dugmeta

Klikom na dugme „Prodaja nekretnine“ otvara se forma za unos podataka o pordavcu i njegovoj nekretnini.

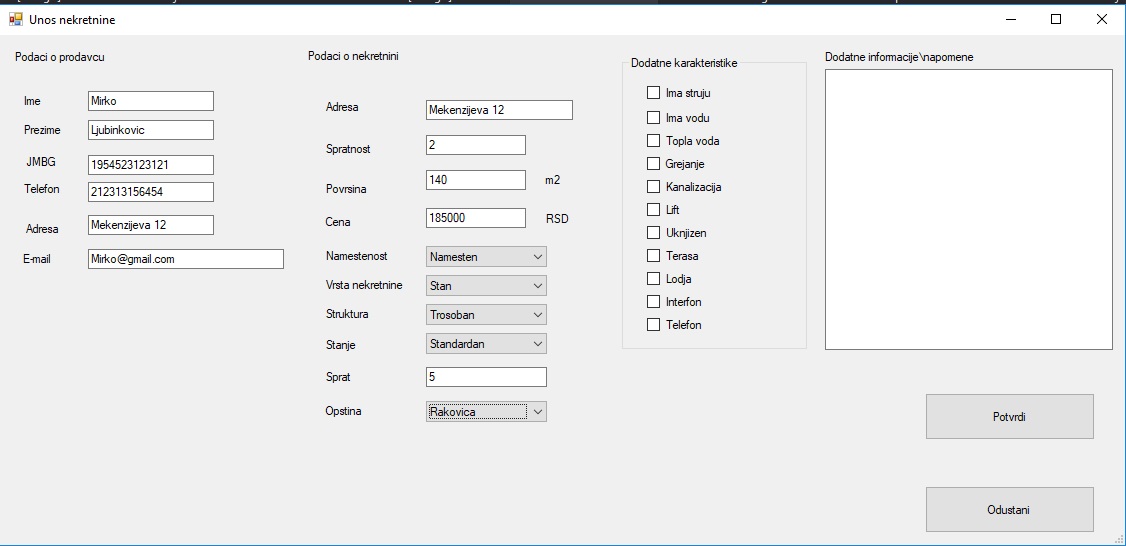
Klikom na dugme „Kupovina nekretnine“ otvara se forma za unos podataka o kupcu njegovom zahtevu i mogućnost izlistavanja svih trenutnih nekretnina.

Klikom na dugme „Admin“ otvara se forma za unos podataka o agentima, i mogućnost brisanja, ažuriranja i sam pregled agenata.

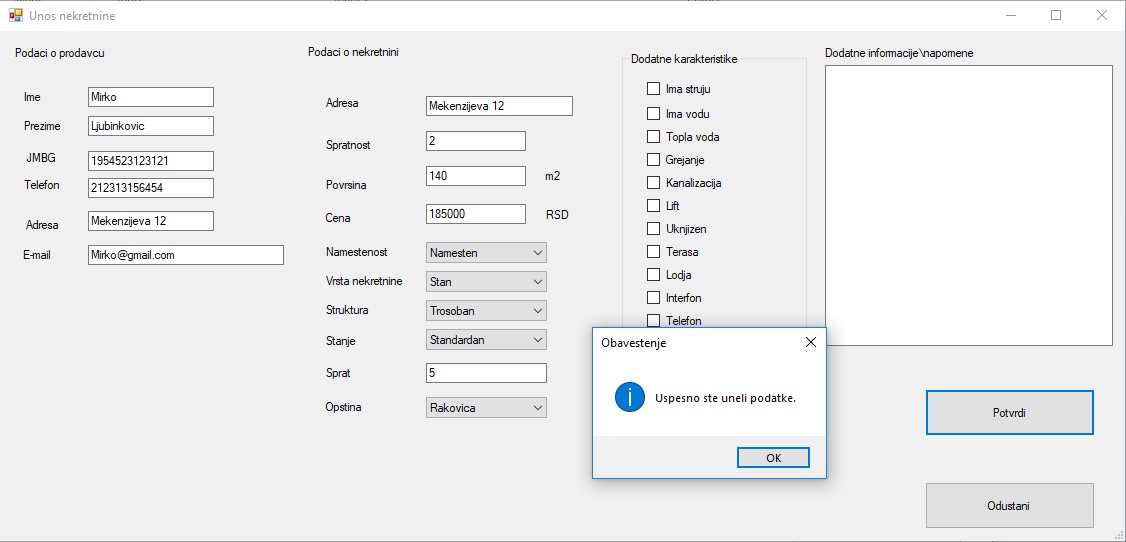
Pritiskom na dugme “Prodaja nekretnine” ovara se forma 2:



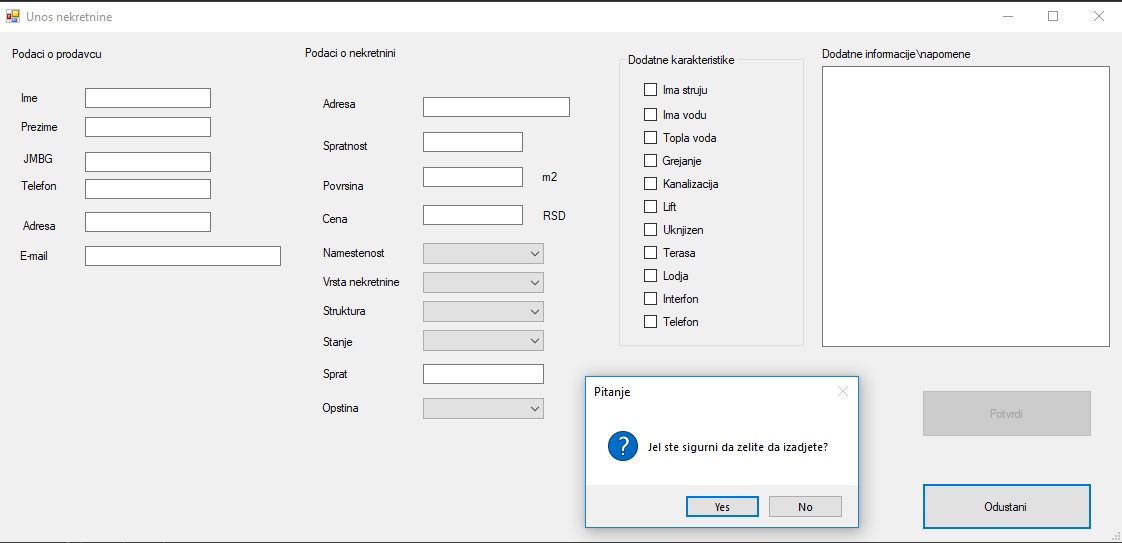
Potrebno je uneti sve podatke o kupcu kao I sve podatke o nekretnini da bi dugme “Potvrdi” postalo omogućeno, dodaten karakteristike i napomena su opcioni za unos.



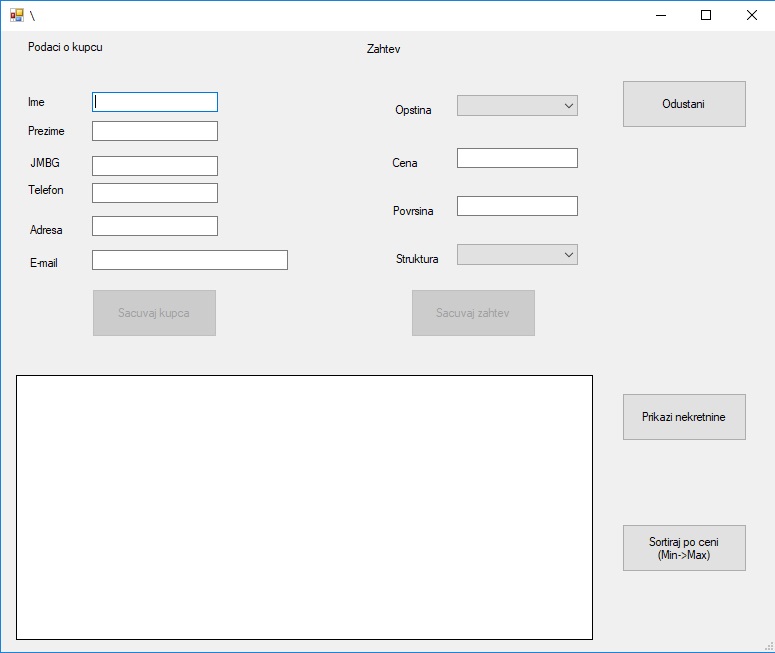
Klikom na dugeme potvrdi Podaci se unose u bazu, a korisnik dobija MessageBox o uspesnom unosu.



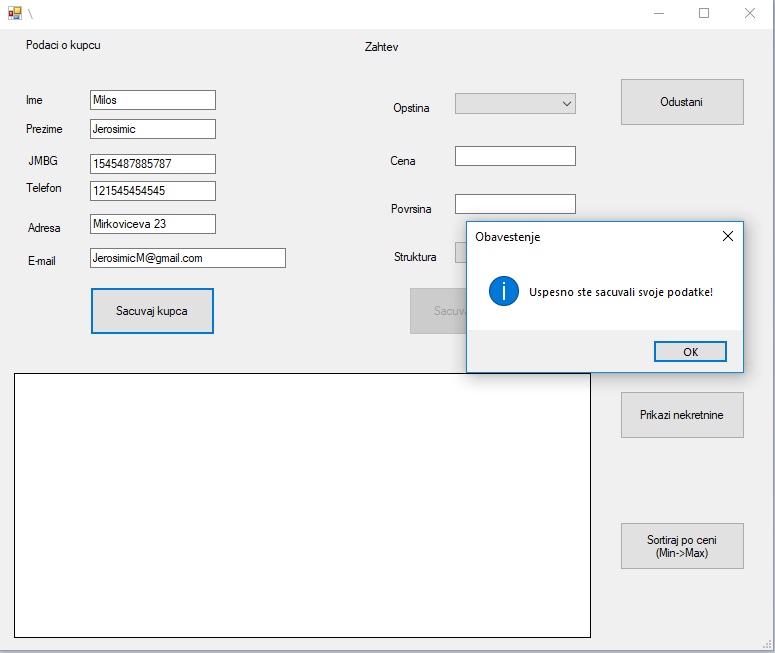
**Slika 4. Forma 2.2**

Klikom na duge odustani “Odustani” pojavljuje se MessageBox sa pitanjem o zatvaranju forme, ukoliko se pretisne Yes dugme forma ce se zatvoriti, ukoliko se pritisne dugme No forma ce ostati aktivna.

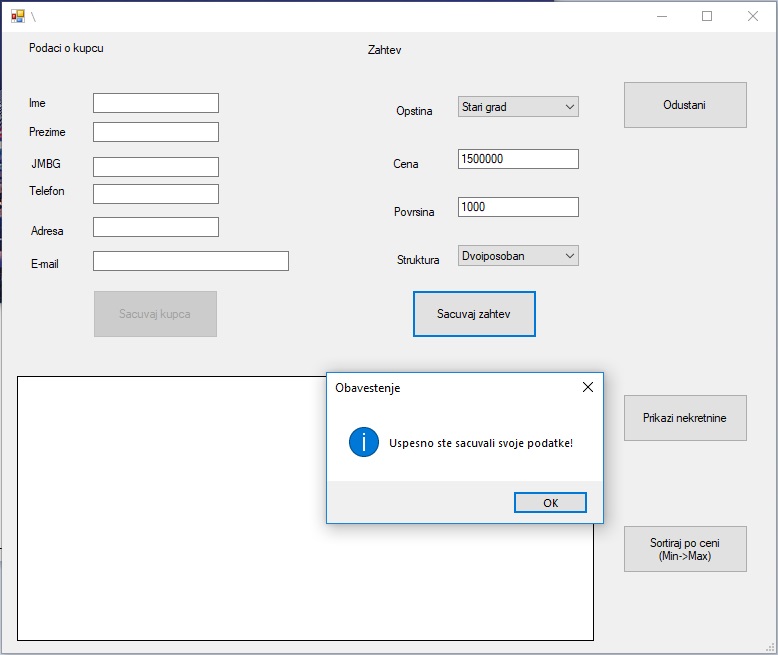
Pritiskom na dugme “Kupovina nekretnine” ovara se forma 3:



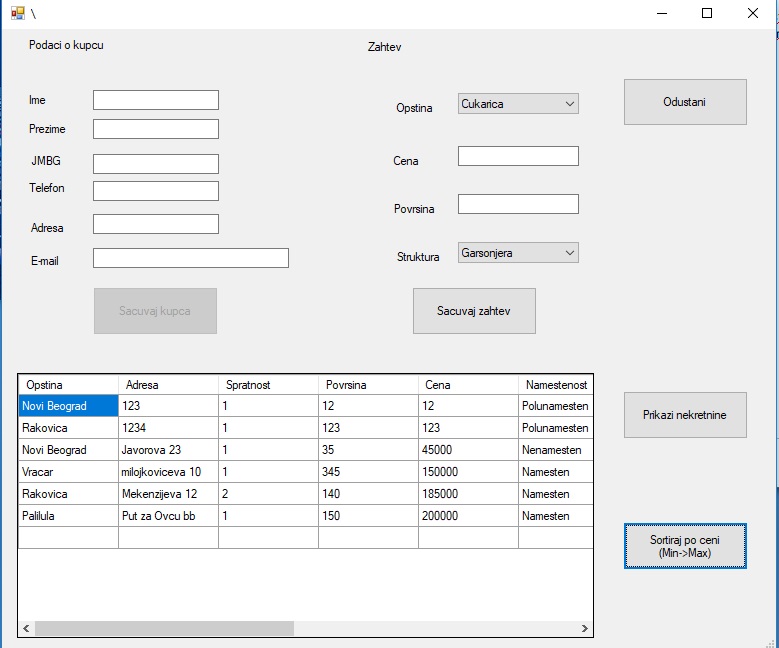
Unosom svih podataka za kupca Dugme “Sacuvaj kupca” se otkljucava, pritikom na njega unose se informacija za kupca I korinsik dobija MessageBox sa informacijom o uspesnom unosu.



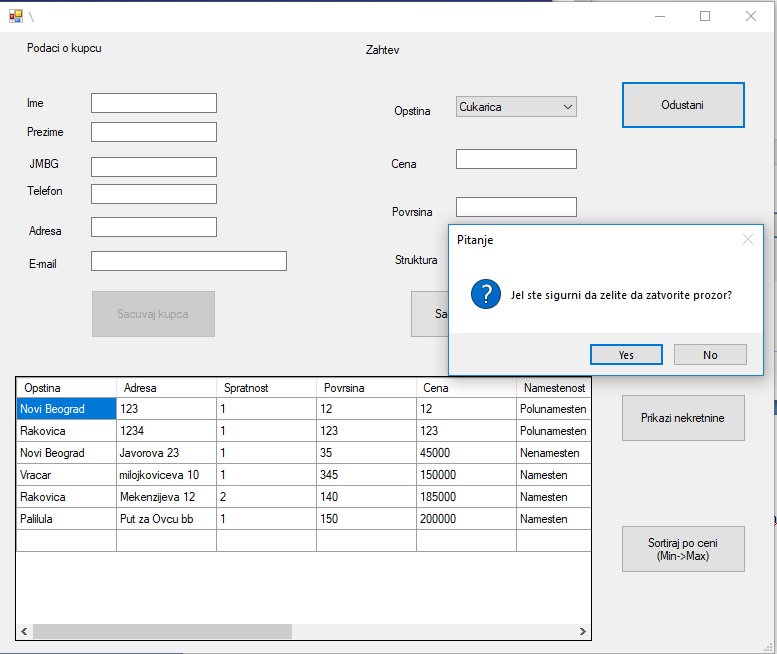
Unosenjem opstine kao jedinog obaveznog polja otkljucava se dugme za “Sacuvaj zahtev”, nakon cega izlazi poruka o uspešnoj akciji.



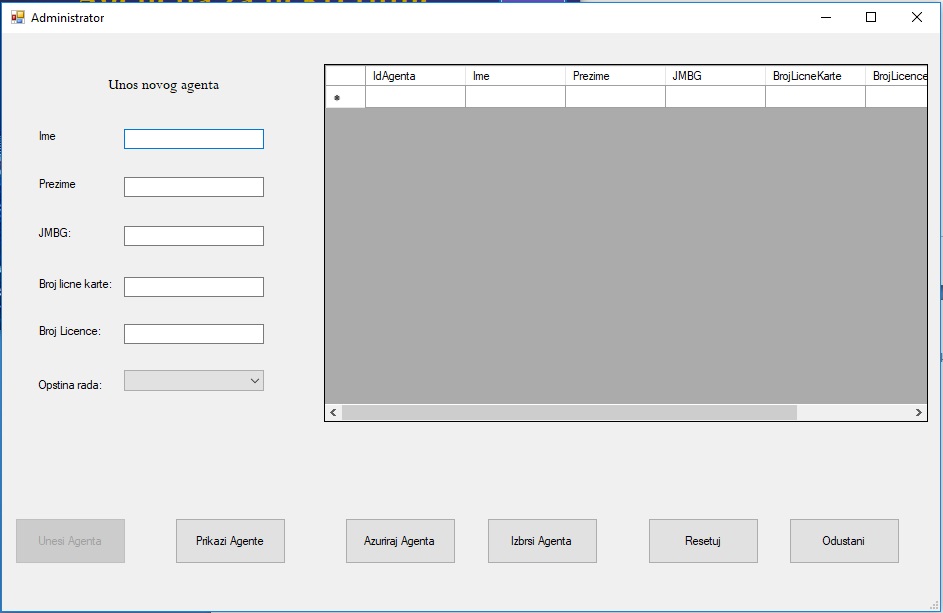
Pritiskom na dugme “Prikazi nekretnine” u DataGridView-u se iylistavaju sve trenutne nekretnine, a klikom na dugme “Sortiraj po ceni” nekretnine se ređaju po ceni od najmanje do najveće.



Pritiskom na duge “Odustani” korisnik dobija poruku , ukoliko klikne dugme Yes forma se zatvara, ukoliko No forma ostaje otvorena.

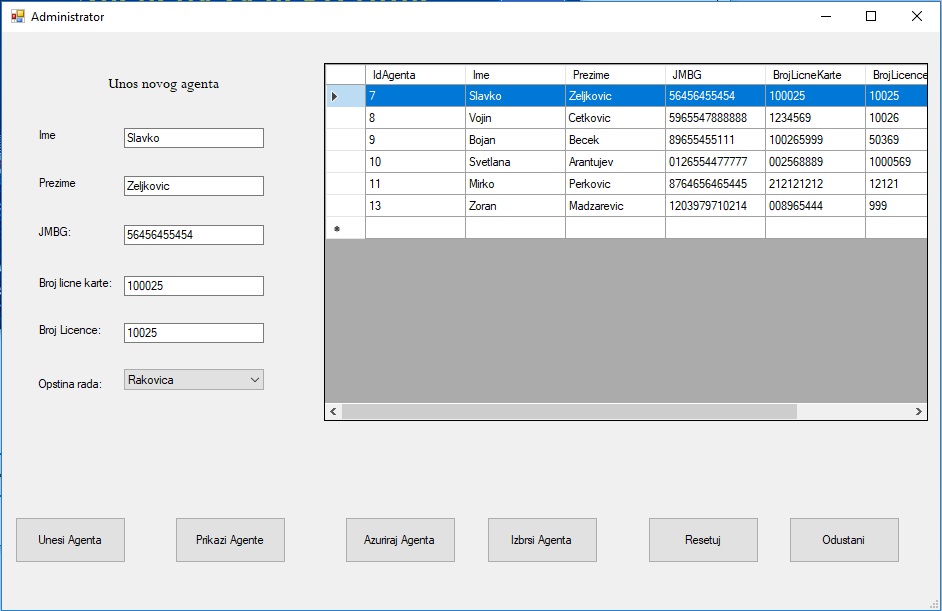


Pritiskom na dugme “Admin” ovara se forma 3:



Unosenjem svih polja ya unos novog agenta, otkljucava se polje “Unos Agenta”, nakon cega se klikom na dugem agent unosi u bazu.

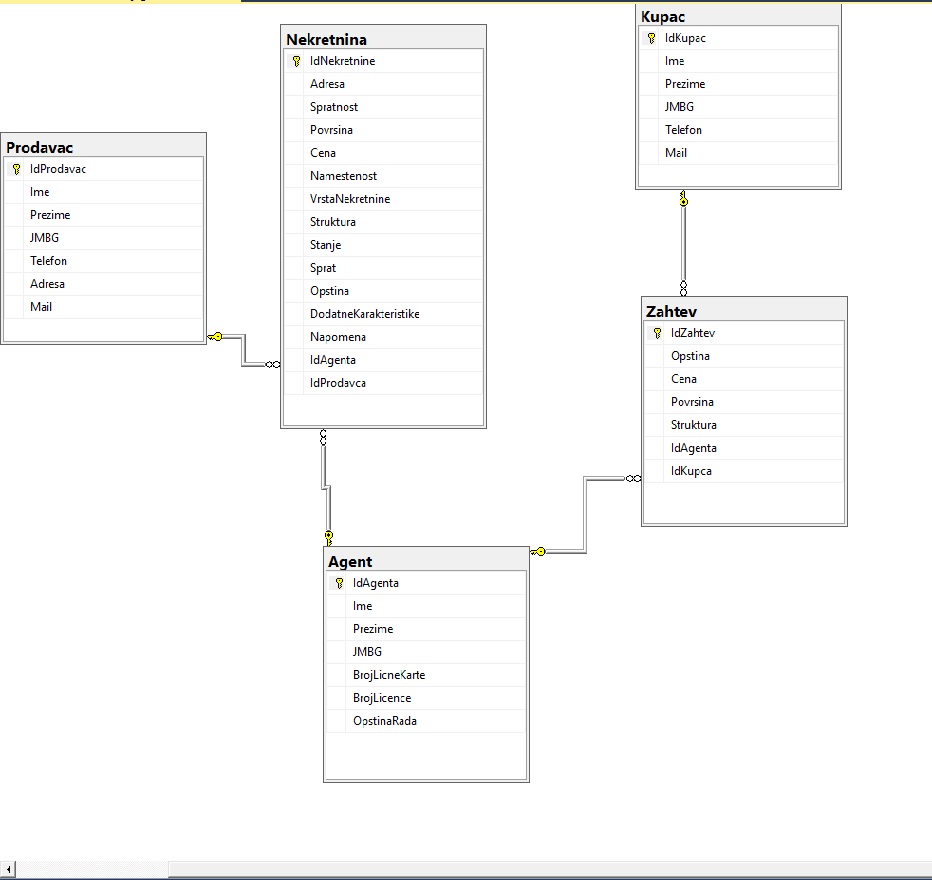
Pritiskom na dugme “Prikayi Agente” prikayuju se svi agenti u DataGridView-u.



Klikom na neki do redova svi podaci iz reda se unose u polja za unos agenta, promenom tih podataka I klikom na dugem “Azuriraj agenta” promenjeni podaci ce se uneti, klikom na dugme “Izbrisi Agenta” agnet ce se izbirati iz baz, Klikom na dugme “Resetuj” svi podaci iz kontrola ce se resetovati, klikom na dugme.

**IZGLED BAZE PODATAKA**

Baza podataka se sastoji od 5 povezanih tabela



Tabele su : Poravac, Nekretnina,Agent, Zahtev, Kupac

Tabela Pordavac(PK IdProdavac) i Nekretnina (FK IdProdavac) su povezani preko IdProdavac

Tabela Agent(PK IdAgenta) i Nekretnina (FK IdAgenta) su povezani preko IdAgenta

Tabela Agent(PK IdAgenta) i Zahtev(FK IdAgenta) su povezani preko IdAgenta

Tabela Kupac(PK IdKupca) i Nekretnina(FK IdKupca) su povezani preko IdKupca

**APLIKACIJA SA STANOVIŠTA PROGRAMERA**

Forma frmProdajNekretnine.cs :

private void resetuj()

{

tbIme.Clear();

tbPrezime.Clear();

tbJMBG.Clear();

tbMail.Clear();

tbAdresaProdavca.Clear();

tbAdresa.Clear();

tbSprat.Clear();

tbPovrsina.Clear();

tbCena.Clear();

tbSpratnost.Clear();

tbSprat.Clear();

tbNapomena.Clear();

cbNamestenost.SelectedIndex = 0;

cbVrsta.SelectedIndex = 0;

cbStruktura.SelectedIndex = 0;

cbStanje.SelectedIndex = 0;

cbOpstina.SelectedIndex = 0;

foreach(CheckBox cb in groupBox1.Controls)

{

cb.Checked = false;

}

}

**Resetuje sve kontrole na formi**

void setButton()

{

btnPotvrdi.Enabled = (tbIme.Text != "" && tbPrezime.Text != "" && tbJMBG.Text !="" && tbTelefon.Text != "" && tbAdresa.Text !="" && tbMail.Text != "" && tbAdresa.Text != "" && tbSpratnost.Text != "" && tbPovrsina.Text != "" && tbCena.Text != "" && tbSprat.Text != "" && cbNamestenost.Text != "" && cbVrsta.Text != "" && cbStruktura.Text != "" && cbStanje.Text != "" && cbOpstina.Text != "");

}

**Dugme ce se otkljucati ukolio su sve kontrole različite od praznog stringa, ova metoda je implementirana za sve kontrole na dogadjaju TextChanged za textbox I SelectedIndexChanged za comboBox.**

public void samoSlova(KeyPressEventArgs e)

{

e.Handled = !(char.IsLetter(e.KeyChar) || e.KeyChar == (char)Keys.Back);

}

**Onemogućava korinsiku unos brojeva u kontrolu**

public void samoBrojevi(KeyPressEventArgs e)

{

e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar);

}

**Onemogućava korinsiku unos slova u kontrolu**

private void btnPotvrdi\_Click(object sender, EventArgs e)

{

SqlConnection konekcija1 = new SqlConnection(Konekcija.konekcija);

try

{

SqlCommand komandaProdavac = new SqlCommand("INSERT INTO Prodavac(Ime, Prezime, JMBG, Telefon, Adresa, Mail) VALUES (@Ime, @Prezime, @JMBG, @Telefon, @Adresa, @Mail)", konekcija1);

komandaProdavac.Parameters.AddWithValue("@Ime",tbIme.Text);

komandaProdavac.Parameters.AddWithValue("@Prezime",tbPrezime.Text);

komandaProdavac.Parameters.AddWithValue("@JMBG",tbJMBG.Text);

komandaProdavac.Parameters.AddWithValue("@Telefon",tbTelefon.Text);

komandaProdavac.Parameters.AddWithValue("@Adresa",tbAdresa.Text);

komandaProdavac.Parameters.AddWithValue("@Mail",tbMail.Text);

konekcija1.Open();

komandaProdavac.ExecuteNonQuery();

}

catch(Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

konekcija1.Close();

}

foreach (CheckBox cb in groupBox1.Controls)

if (cb.Checked)

{

pom += cb.Text + "; ";

}

SqlConnection konekcija = new SqlConnection(Konekcija.konekcija);

try {

SqlCommand komand1 = new SqlCommand("SELECT IdAgenta FROM Agent WHERE OpstinaRada='"+cbOpstina.Text+ "'", konekcija);

SqlCommand komand2 = new SqlCommand("SELECT IdProdavac from Prodavac WHERE Adresa = '"+tbAdresa.Text+"'", konekcija);

konekcija.Open();

int idAgenta = Convert.ToInt32(komand1.ExecuteScalar());

int idProdavac = Convert.ToInt32(komand2.ExecuteScalar());

SqlCommand komand = new SqlCommand("INSERT INTO Nekretnina (Adresa, Spratnost, Povrsina, Cena, Namestenost, VrstaNekretnine, Struktura, Stanje, Sprat, Opstina, DodatneKarakteristike, Napomena, IdAgenta,idProdavca) VALUES (@Adresa, @Spratnost, @Povrsina, @Cena, @Namestenost, @VrstaNekretnine, @Struktura, @Stanje, @Sprat, @Opstina,@DodatneKarakteristike,@Napomena, @IdAgenta,@idProdavac);", konekcija);

komand.Parameters.AddWithValue("@Adresa", tbAdresa.Text);

komand.Parameters.AddWithValue("@Spratnost", tbSpratnost.Text);

komand.Parameters.AddWithValue("@Povrsina", tbPovrsina.Text);

komand.Parameters.AddWithValue("@Cena", tbCena.Text);

komand.Parameters.AddWithValue("@Namestenost", cbNamestenost.Text);

komand.Parameters.AddWithValue("@VrstaNekretnine", cbVrsta.Text);

komand.Parameters.AddWithValue("@Struktura", cbStruktura.Text);

komand.Parameters.AddWithValue("@Stanje", cbStanje.Text);

komand.Parameters.AddWithValue("@Sprat", tbSprat.Text);

komand.Parameters.AddWithValue("@Opstina", cbOpstina.Text);

komand.Parameters.AddWithValue("@DodatneKarakteristike", pom);

komand.Parameters.AddWithValue("@Napomena", tbNapomena.Text);

komand.Parameters.AddWithValue("@IdAgenta", idAgenta);

komand.Parameters.AddWithValue("@IdProdavac", idProdavac);

komand.ExecuteNonQuery();

}

catch(Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

konekcija.Close();

}

pom = "";

MessageBox.Show("Uspesno ste uneli podatke.", "Obavestenje", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

**Na dugme Potvdi pravi se objekat kalse SqlConnection, kao parametar konstruktora se upisuje odredjeni SQL upit preko SqlCommand klase, nakon toga se try bolcku otvara konekcija sa bazom i izvršava SQL upit za upis podataka iz kontrola u tabelu Prodavac i Nekretnina, u catch bolcku se hvataju svi exeptioni i u finally bolku se zatvara konekcija iz razloga sto ce se finally uvek izvršiti, nakon toga korisnik dobija message da je usepsno izvršen unos.**

Administrator.cs

private void dataGridView1\_RowHeaderMouseClick(object sender, DataGridViewCellMouseEventArgs e)

{

ID = Convert.ToInt32(dataGridView1.Rows[e.RowIndex].Cells[0].Value.ToString());

tbIme.Text = dataGridView1.Rows[e.RowIndex].Cells[1].Value.ToString();

tbPrezime.Text = dataGridView1.Rows[e.RowIndex].Cells[2].Value.ToString();

tbJMBG.Text = dataGridView1.Rows[e.RowIndex].Cells[3].Value.ToString();

tbBrojLicne.Text = dataGridView1.Rows[e.RowIndex].Cells[4].Value.ToString();

tbBrojLicence.Text = dataGridView1.Rows[e.RowIndex].Cells[5].Value.ToString();

cbOpstina.Text = dataGridView1.Rows[e.RowIndex].Cells[6].Value.ToString();

}

**Na Select citave kolone u DataGridView Popunjava sve kontorole sa podacima iz baze radi lakšek ayuriranja podataka**

int ID = 0;

private void btnIzbrsi\_Click(object sender, EventArgs e)

{

if (ID != 0)

{

SqlConnection konekcija = new SqlConnection(Konekcija.konekcija);

SqlCommand komand = new SqlCommand("DELETE Agent WHERE IdAgenta=@IdAgenta", konekcija);

try

{

konekcija.Open();

komand.Parameters.AddWithValue("@IdAgenta", ID);

komand.ExecuteNonQuery();

MessageBox.Show("Uspesno Izbrisan Agent!");

}

catch(Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

konekcija.Close();

prikaziAgente();

}

}

else

{

MessageBox.Show("Izaberite agenta kojeg zelite da izbrisete");

}

}

}

**Dugme Izbrisi brise red iz baze agent ukoliko je ID rayličit od 0, u trz blocku se otvara konekcija izvršava se SQL komanda, catch hvata sve exeptione, I finallz zatvara konekciju, prikazuje sve iz tabele Agnet u DataGridView**