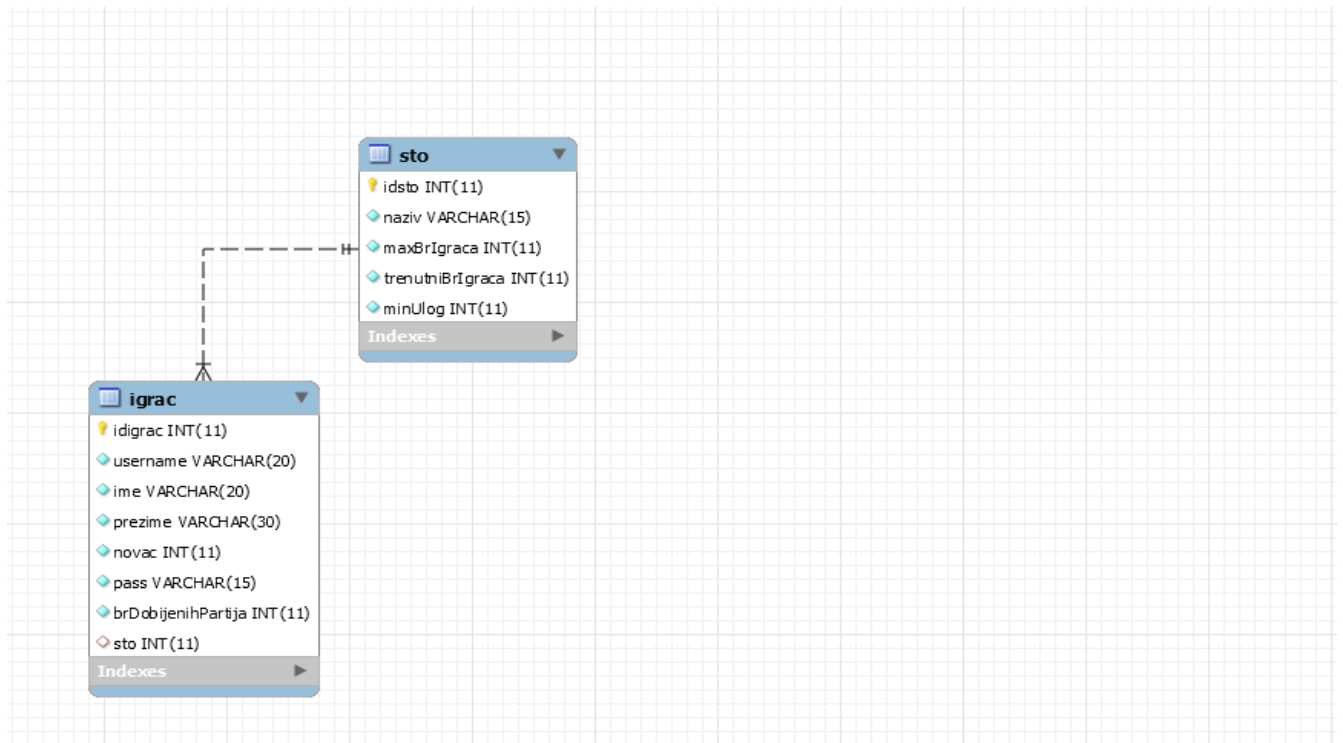


Izgled baze



Klasa Igrac

```
public class Igrac
{
    1 reference
    public virtual int id { get; set; }
    1 reference
    public virtual string username { get; set; }
    1 reference
    public virtual string Ime { get; set; }
    1 reference
    public virtual string Prezime { get; set; }
    2 references
    public virtual string password { get; set; }
    3 references
    public virtual int novac { get; set; }
    1 reference
    public virtual int brDobijenihPartija { get; set; }
    1 reference
    public virtual Sto sto { get; set; }
}
```

Klasa Sto

```
20 references
public class Sto
{
    2 references
    public virtual int id {get; set;}
    2 references
    public virtual String Naziv {get; set; }
    2 references
    public virtual int maxBrIgraca { get; set; }
    1 reference
    public virtual int trenutniBrIgraca { get; set; }
    5 references
    public virtual int minUlog { get; set; }
    2 references
    public virtual IList<Igrac> igraci { get; set; }
    0 references
    public Sto()
    {
        igraci = new List<Igrac>();
    }
}
```

Mapiranje klase Igrac

```
0 references
public IgracMapiranje()
{
    Table("IGRAC");
    Id(x => x.id, "IDIGRAC").GeneratedBy.TriggerIdentity();
    Map(x => x.Ime, "IME");
    Map(x => x.Prezime, "PREZIME");
    Map(x => x.username, "USERNAME");
    Map(x => x.password, "PASS");
    Map(x => x.novac, "NOVAC");
    Map(x => x.brDobijenihPartija, "BRDOBIJENIHPARTIJA");
    References(x => x.sto).Column("STO").LazyLoad();
}
}
```

Mapiranje klase Sto

```
0 references
public StoMapiranje()
{
    Table("STO");
    Id(x => x.id, "IDSTO").GeneratedBy.TriggerIdentity();
    Map(x => x.Naziv, "NAZIV");
    Map(x => x.minUlog, "MINULOG");
    Map(x => x.maxBrIgraca, "MAXBRIGRACA");
    Map(x => x.trenutniBrIgraca, "TRENUTNIBRIGRACA");
    HasMany(x => x.igraci).KeyColumn("STO").Inverse().Cascade.All();
}
```

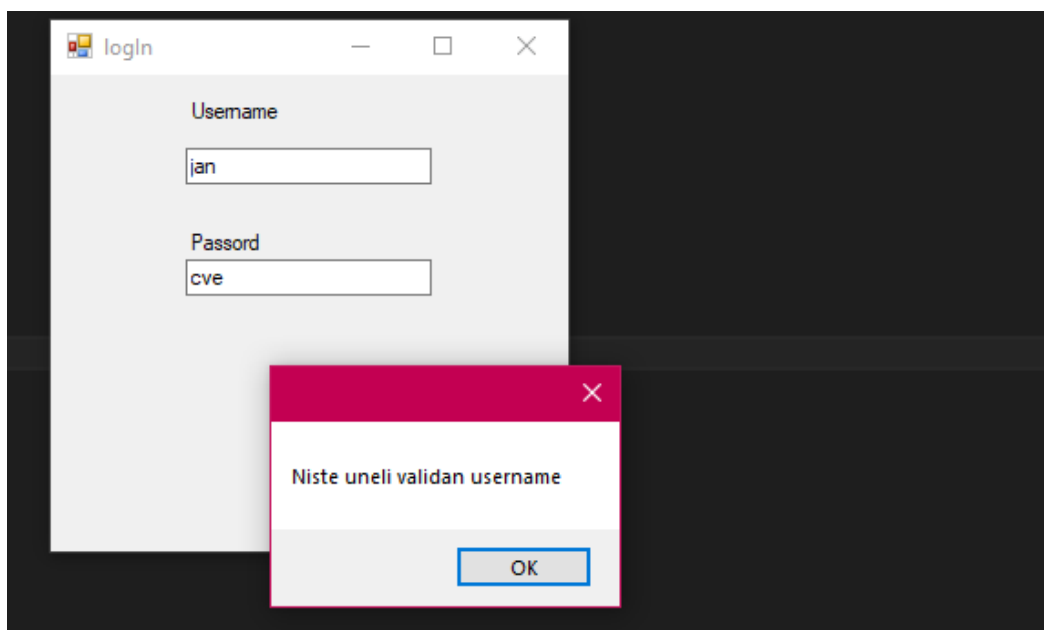
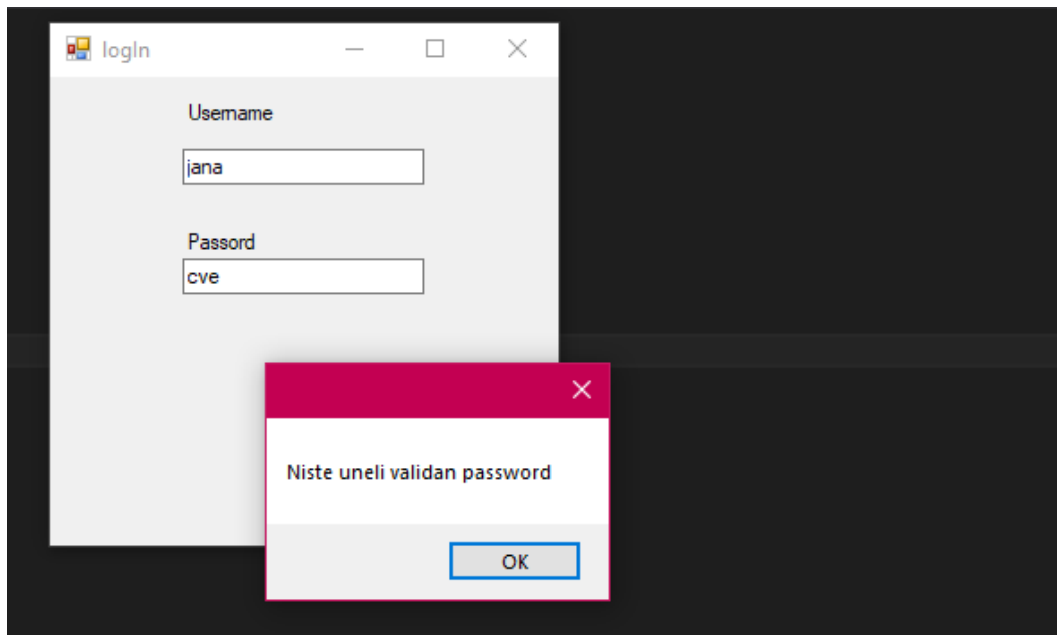
Funkcije za rad sa bazom podataka

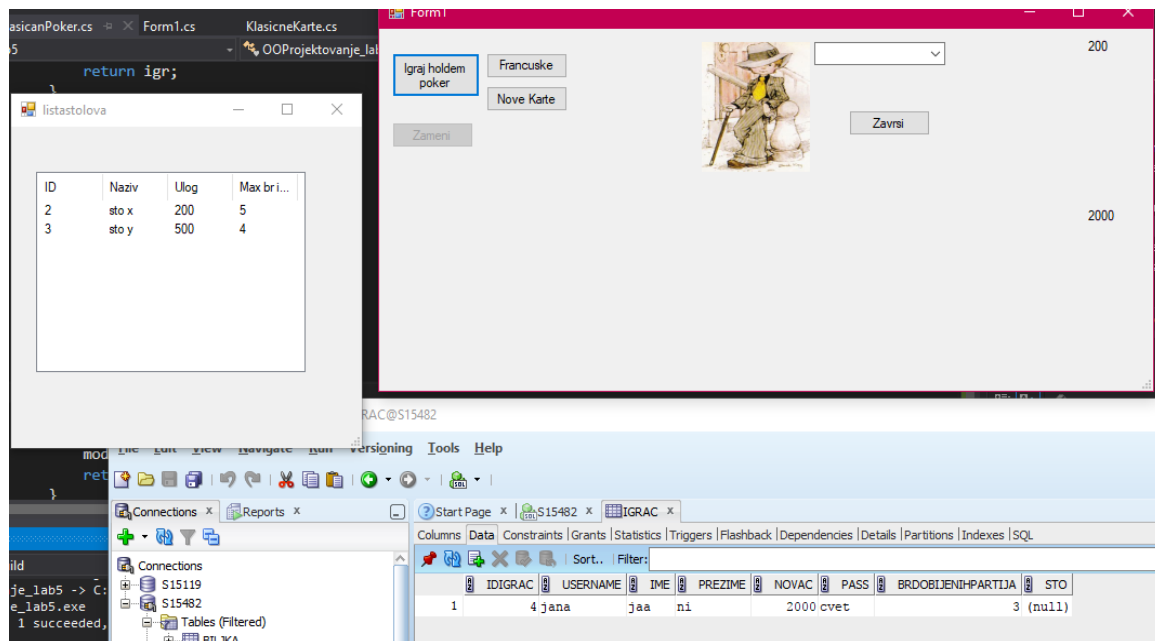
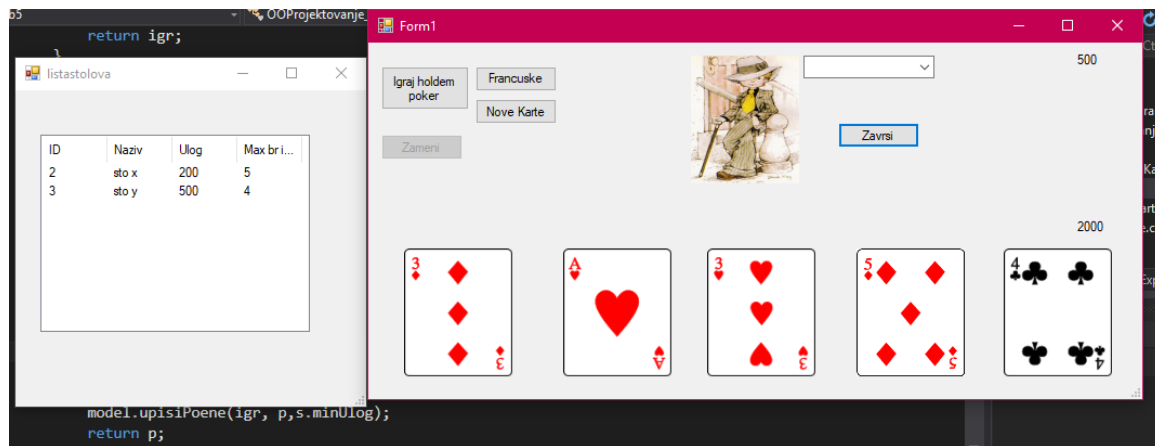
```
public void upisiPoene(Igrac id, int iznos, int ulog)
{
    ISession s = DataLayer.GetSession();
    id.novac += iznos - ulog;
    s.Update(id);
    s.Flush();
    s.Close();
}

4 references
public Igrac vratiIgraca(string username)
{
    ISession s = DataLayer.GetSession();
    IQuery q = s.CreateQuery("from Igrac ig where ig.username=:username");
    q.SetString("username", username);
    Igrac i = q.UniqueResult<Igrac>();
    s.Close();
    return i;
}
```

```
4 references
public IList<Sto> vratiStoLove()
{
    ISession s = DataLayer.GetSession();
    IQuery q = s.CreateQuery("from Sto");
    IList<Sto> rez = q.List<Sto>();
    s.Close();
    return rez;
}
```

Primeri rada aplikacije i baze





klasicanPoker.cs Form1.cs KlasicneKarte.cs

return igr;

listastolova

ID	Naziv	Ulog	Max bri...
2	sto x	200	5
3	sto y	500	4


Form1

Igraj holdem poker

Francuske

Nove Karte


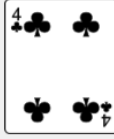
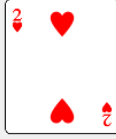
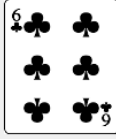

Zameni



200

Završi

1800



SQL Server Enterprise Edition

Start Page S15482 TGRAC

Columns Data Constraints Grants Statistics Triggers Flashback Dependencies Details Partitions Indexes SQL

IDIGRAC

USERNAME

IME

PREZIME

NOVAC

PASS

BRDOBIJENIHPARTIJA

STO

1	4 jana	jaa	ni	1800 cvet		3 (null)
---	--------	-----	----	-----------	--	----------