

VIŠJA STROKOVNA ŠOLA VELENJE



Program: Informatika

Predmet: Razvoj programskih aplikacij

# SEMINARSKA NALOGA

## ADMINISTRATIVNA APLIKACIJA ZA ŠTEVILO OGLED OV FILMA

Avtor: Jure Zajc

Velenje, julij 2017

## Kazalo vsebine

1.	Uvod .....	4
2.	Dodatki in reference .....	4
3.	Aplikacija .....	4
4.	Konfiguracijska datoteka .....	4
5.	Pisanje kode.....	6
5.1	Ustvarjanje tabel v bazi .....	6
5.2	Prvi pogled .....	7
5.3	Prijava .....	7
5.4	Glavno okno.....	9
6.	Navodila za uporabo.....	11
7.	Zaključek.....	15

## Kazalo slik

Slika 1 Konfiguracijska datoteka.....	5
Slika 2 Hibernate_log.txt .....	5
Slika 3 Entiteta zaposlenih.....	6
Slika 4 Entiteta filmov.....	6
Slika 5 Link label .....	7
Slika 6 Prijava log4net ter nhibernate .....	7
Slika 7 Registracija gumb.....	7
Slika 8 Session.....	8
Slika 9 LoginUser.....	8
Slika 10 Check User.....	8
Slika 11 Glavno okno .....	9
Slika 12 ToolStrip zaposleni .....	9
Slika 13 Odjava .....	9
Slika 14 Xml izvoz.....	10
Slika 15 Izbris filma .....	10
Slika 16 Xampp .....	11
Slika 17 Xampp glavni meni.....	11
Slika 18 Zagon aplikacije.....	11
Slika 19 Pozdravno okno.....	12
Slika 20 Registracija .....	12
Slika 21 Napačna registracija.....	12
Slika 22 Uspešna registracija .....	13
Slika 23 Glavno okno .....	13
Slika 24 Urejanje profila .....	14
Slika 25 Zaposleni okno .....	14
Slika 26 Odjava .....	14

## 1. Uvod

V svoji seminarski nalogi bom predstavil Windows Forms aplikacijo, katera nam omogoča dodajanje ogledov posameznega filma in spreminjanje podatkov, če bomo film ponovno dali na ogled v prihodnosti. Aplikacija je narejena v programu Visual Studio Enterprise 2017 v C#.NET okolju. Zaenkrat jo lahko uporabljamo samo na računalnikih.

## 2. Dodatki in reference

Za povezovanje z bazo sem uporabil zunanjo referenco NHibernate. Za logiranje pa sem uporabil Log4Net

## 3. Aplikacija

Aplikacija nam omogoča naslednje funkcionalnosti:

- Prijava v aplikacijo
- Povezava s podatkovno bazo
- Administriranje baze
- Vnos, spreminjanje in brisanje podatkov
- Izvoz podatkov v XML datoteko
- Logiranje
- Spreminjanje uporabniških nastavitev

## 4. Konfiguracijska datoteka

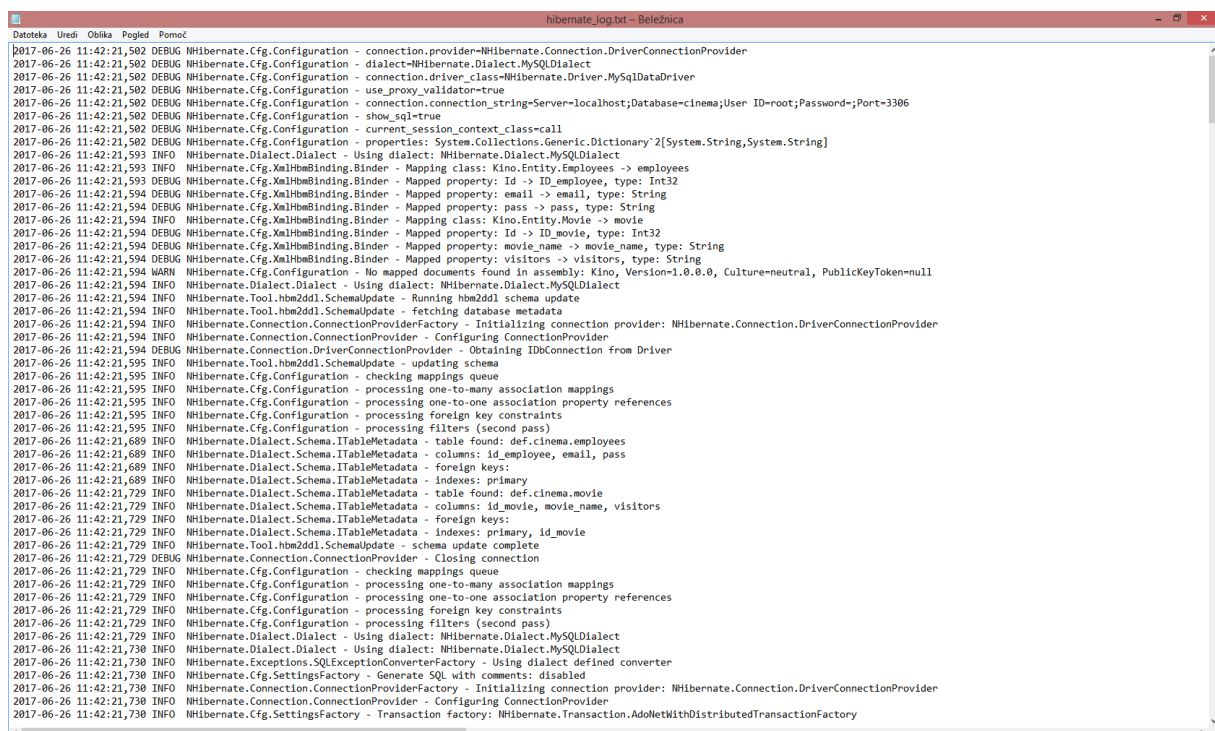
Za povezavo s bazo in logiranje sem uporabil konfigaricijsko datoteko. Datoteka je v XML formatu, katero sem spremenil da deluje na mojem primeru. Za povezovanje s bazo sem popravil connection property, kjer sem spremenil ime baze, uporabniško ime ter geslo, na koncu pa še port. Za property dialect sem nastavil MySQL, saj delam s MySQL podatkovno bazo. Nastavil sem še log4net property, kjer v tekstovno datoteko shranimo zgodimo uporabe aplikacije in baze.

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <configuration>
3
4
5  <configSections>
6    <section name="hibernate-configuration" type="NHibernate.Cfg.ConfigurationSectionHandler, NHibernate"/>
7    <section name="log4net" type="log4net.Config.Log4NetConfigurationSectionHandler, log4net"/>
8  </configSections>
9
10 <hibernate-configuration xmlns="urn:hibernate-configuration-2.2">
11   <session-factory>
12     <property name="connection.provider">NHibernate.Connection.DriverConnectionProvider</property>
13     <property name="dialect">NHibernate.Dialect.MySQLDialect</property>
14     <property name="connection.driver_class">NHibernate.Driver.MySqlDataDriver</property>
15     <property name="use_proxy_validator">true</property>
16     <property name="connection.connection_string">Server=localhost;Database=cinema;User ID=root;Password=;Port=3306</property>
17     <property name="show_sql">true</property>
18     <property name="current_session_context_class">call</property>
19   </session-factory>
20 </hibernate-configuration>
21
22 <log4net>
23   <appender name="DebugSQL" type="log4net.Appender.TraceAppender">
24     <layout type="log4net.Layout.PatternLayout">
25       <conversionPattern value="%date [%thread] %-5level %logger [%property{NDC}] - %message%nnewline"/>
26     </layout>
27   </appender>
28   <appender name="HibernateLogFile" type="log4net.Appender.RollingFileAppender">
29     <file value="hibernate_log.txt"/>
30     <appendToFile value="false"/>
31     <encoding value="utf-8"/>
32     <rollingStyle value="Date"/>
33     <datePattern value="yyyy/MMdd"/>
34     <layout type="log4net.Layout.PatternLayout">
35       <conversionPattern value="%date %-5level %logger - %message%nnewline"/>
36     </layout>
37   </appender>

```

Slika 1 Konfiguracijska datoteka



```

2017-06-26 11:42:21,502 DEBUG NHibernate.Cfg.Configuration - connection.provider=NHibernate.Connection.DriverConnectionProvider
2017-06-26 11:42:21,502 DEBUG NHibernate.Cfg.Configuration - dialect=NHibernate.Dialect.MySQLDialect
2017-06-26 11:42:21,502 DEBUG NHibernate.Cfg.Configuration - connection.driver_class=NHibernate.Driver.MySqlDataDriver
2017-06-26 11:42:21,502 DEBUG NHibernate.Cfg.Configuration - use_proxy_validator=true
2017-06-26 11:42:21,502 DEBUG NHibernate.Cfg.Configuration - connection.connection_string=Server=localhost;Database=cinema;User ID=root;Password=;Port=3306
2017-06-26 11:42:21,502 DEBUG NHibernate.Cfg.Configuration - show_sql=true
2017-06-26 11:42:21,502 DEBUG NHibernate.Cfg.Configuration - current_session_context_class=call
2017-06-26 11:42:21,502 DEBUG NHibernate.Cfg.Configuration - properties: System.Collections.Generic.Dictionary`2[System.String,System.String]
2017-06-26 11:42:21,593 INFO NHibernate.Dialect.Dialect - Using dialect: NHibernate.Dialect.MySQLDialect
2017-06-26 11:42:21,593 INFO NHibernate.Cfg.XmlHbmBinding.Binder - Mapping class: Kino.Entity.Employees -> employees
2017-06-26 11:42:21,593 DEBUG NHibernate.Cfg.XmlHbmBinding.Binder - Mapped property: Id -> ID_employee, type: Int32
2017-06-26 11:42:21,594 DEBUG NHibernate.Cfg.XmlHbmBinding.Binder - Mapped property: email -> email, type: String
2017-06-26 11:42:21,594 DEBUG NHibernate.Cfg.XmlHbmBinding.Binder - Mapped property: pass -> pass, type: String
2017-06-26 11:42:21,594 INFO NHibernate.Cfg.XmlHbmBinding.Binder - Mapping class: Kino.Entity.Movie -> movie
2017-06-26 11:42:21,594 DEBUG NHibernate.Cfg.XmlHbmBinding.Binder - Mapped property: Id -> ID_movie, type: Int32
2017-06-26 11:42:21,594 DEBUG NHibernate.Cfg.XmlHbmBinding.Binder - Mapped property: movie_name -> movie_name, type: String
2017-06-26 11:42:21,594 DEBUG NHibernate.Cfg.XmlHbmBinding.Binder - Mapped property: visitors -> visitors, type: String
2017-06-26 11:42:21,594 WARN NHibernate.Cfg.Configuration - No mapped documents found in assembly: Kino, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null
2017-06-26 11:42:21,594 INFO NHibernate.Dialect.Dialect - Using dialect: NHibernate.Dialect.MySQLDialect
2017-06-26 11:42:21,594 INFO NHibernate.Tool.hbm2ddl.SchemaUpdate - Running hbm2ddl schema update
2017-06-26 11:42:21,594 INFO NHibernate.Tool.hbm2ddl.SchemaUpdate - fetching database metadata
2017-06-26 11:42:21,594 INFO NHibernate.Connection.ConnectionProviderFactory - Initializing connection provider: NHibernate.Connection.DriverConnectionProvider
2017-06-26 11:42:21,594 INFO NHibernate.Connection.ConnectionProvider - Configuring ConnectionProvider
2017-06-26 11:42:21,594 DEBUG NHibernate.Connection.DriverConnectionProvider - Obtaining IDbConnection from Driver
2017-06-26 11:42:21,595 INFO NHibernate.Tool.hbm2ddl.SchemaUpdate - updating schema
2017-06-26 11:42:21,595 INFO NHibernate.Cfg.Configuration - checking mappings queue
2017-06-26 11:42:21,595 INFO NHibernate.Cfg.Configuration - processing one-to-many association mappings
2017-06-26 11:42:21,595 INFO NHibernate.Cfg.Configuration - processing one-to-one association property references
2017-06-26 11:42:21,595 INFO NHibernate.Cfg.Configuration - processing foreign key constraints
2017-06-26 11:42:21,595 INFO NHibernate.Cfg.Configuration - processing filters (second pass)
2017-06-26 11:42:21,689 INFO NHibernate.Dialect.Schema.ITableMetadata - table found: def.cinema.employees
2017-06-26 11:42:21,689 INFO NHibernate.Dialect.Schema.ITableMetadata - columns: id_employee, email, pass
2017-06-26 11:42:21,689 INFO NHibernate.Dialect.Schema.ITableMetadata - foreign keys:
2017-06-26 11:42:21,689 INFO NHibernate.Dialect.Schema.ITableMetadata - indexes: primary
2017-06-26 11:42:21,729 INFO NHibernate.Dialect.Schema.ITableMetadata - table found: def.cinema.movie
2017-06-26 11:42:21,729 INFO NHibernate.Dialect.Schema.ITableMetadata - columns: id_movie, movie_name, visitors
2017-06-26 11:42:21,729 INFO NHibernate.Dialect.Schema.ITableMetadata - foreign keys:
2017-06-26 11:42:21,729 INFO NHibernate.Dialect.Schema.ITableMetadata - indexes: primary, id_movie
2017-06-26 11:42:21,729 INFO NHibernate.Tool.hbm2ddl.SchemaUpdate - schema update complete
2017-06-26 11:42:21,729 DEBUG NHibernate.Connection.ConnectionProvider - Closing connection
2017-06-26 11:42:21,729 INFO NHibernate.Cfg.Configuration - checking mappings queue
2017-06-26 11:42:21,729 INFO NHibernate.Cfg.Configuration - processing one-to-many association mappings
2017-06-26 11:42:21,729 INFO NHibernate.Cfg.Configuration - processing one-to-one association property references
2017-06-26 11:42:21,729 INFO NHibernate.Cfg.Configuration - processing foreign key constraints
2017-06-26 11:42:21,729 INFO NHibernate.Cfg.Configuration - processing filters (second pass)
2017-06-26 11:42:21,729 INFO NHibernate.Dialect.Dialect - Using dialect: NHibernate.Dialect.MySQLDialect
2017-06-26 11:42:21,730 INFO NHibernate.Dialect.Dialect - Using dialect: NHibernate.Dialect.MySQLDialect
2017-06-26 11:42:21,730 INFO NHibernate.Exceptions.SQLExceptionConverterFactory - Using dialect defined converter
2017-06-26 11:42:21,730 INFO NHibernate.Cfg.SettingsFactory - Generate SQL with comments: disabled
2017-06-26 11:42:21,730 INFO NHibernate.Connection.ConnectionProviderFactory - Initializing connection provider: NHibernate.Connection.DriverConnectionProvider
2017-06-26 11:42:21,730 INFO NHibernate.Connection.ConnectionProvider - Configuring ConnectionProvider
2017-06-26 11:42:21,730 INFO NHibernate.Connection.ConnectionProvider - Configuring ConnectionProvider
2017-06-26 11:42:21,730 INFO NHibernate.Cfg.SettingsFactory - Transaction factory: NHibernate.Transaction.AdoNetWithDistributedTransactionFactory

```

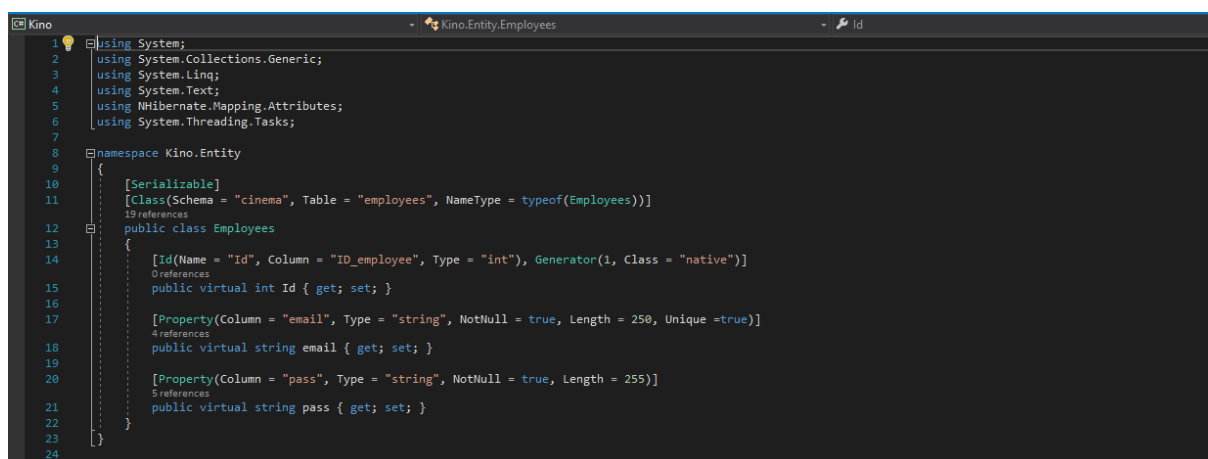
Slika 2 Hibernate\_log.txt

## 5. Pisanje kode

V današnjih časih je pomembno da je koda berljiva in da ima komentarje. Za programiranje samo (imena spremenljivk, razredov...) sem uporabil angleški jezik. Za komentiranje pa sem se odločil da bo v slovenskem jeziku.

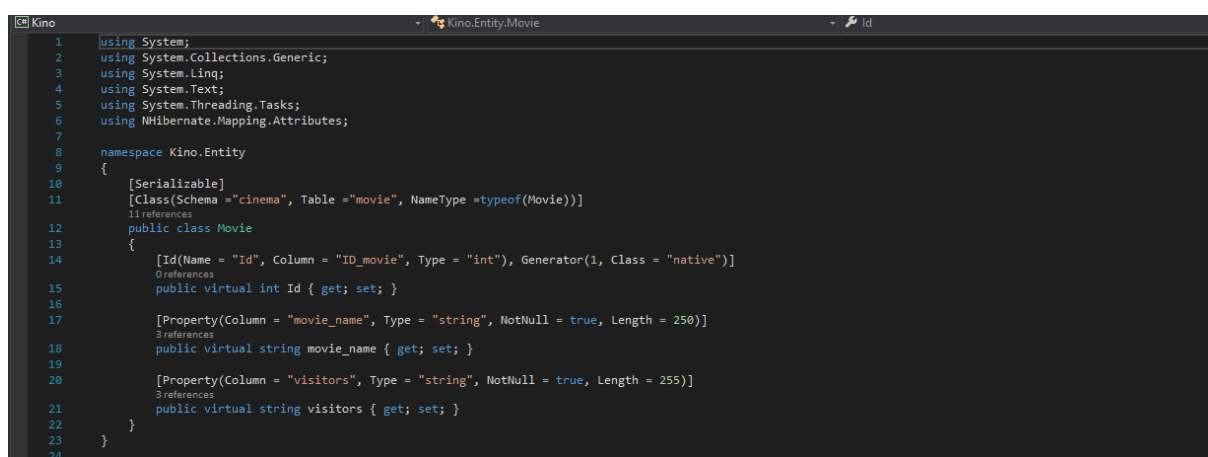
### 5.1 Ustvarjanje tabel v bazi

Za ustvarjanje tabel v bazi sem uporabil razrede s funkcijo `Serializable`, ta funkcija nam omogoča da ustvarimo tabele v bazi. Za vsako tabelo sem uporabil v svoj razred. Preden sem začel delati razrede entitet, sem ustvaril novo mapo v programskem okolju visual studio, zato da sem imel bolj pregledno okolje. Ko sem potreboval te razrede v drugih namespaceih, sem uporabil `using.Kino.Entity;`



```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using NHibernate.Mapping.Attributes;
6 using System.Threading.Tasks;
7
8 namespace Kino.Entity
9 {
10     [Serializable]
11     [Class(Schema = "cinema", Table = "employees", NameType = typeof(Employees))]
12     public class Employees
13     {
14         [Id(Name = "Id", Column = "ID_employee", Type = "int"), Generator(1, Class = "native")]
15         public virtual int Id { get; set; }
16
17         [Property(Column = "email", Type = "string", NotNull = true, Length = 250, Unique = true)]
18         public virtual string email { get; set; }
19
20         [Property(Column = "pass", Type = "string", NotNull = true, Length = 255)]
21         public virtual string pass { get; set; }
22     }
23 }
```

Slika 3 Entiteta zaposlenih



```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6 using NHibernate.Mapping.Attributes;
7
8 namespace Kino.Entity
9 {
10     [Serializable]
11     [Class(Schema = "cinema", Table = "movie", NameType = typeof(Movie))]
12     public class Movie
13     {
14         [Id(Name = "Id", Column = "ID_movie", Type = "int"), Generator(1, Class = "native")]
15         public virtual int Id { get; set; }
16
17         [Property(Column = "movie_name", Type = "string", NotNull = true, Length = 250)]
18         public virtual string movie_name { get; set; }
19
20         [Property(Column = "visitors", Type = "string", NotNull = true, Length = 255)]
21         public virtual string visitors { get; set; }
22     }
23 }
```

Slika 4 Entiteta filmov

## 5.2 Prvi pogled

Ko zaženemo aplikacijo se prikaže okno, kjer kliknemo na povezavo login da se prijavimo. Uporabil sem funkcijo linkLabel, katera me je preusmerila na login in kasneje na movies.

```
// gre na movies.cs
1 reference
private void linkLabel3_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
{
    //skrijemo okno
    this.Hide();
    var movies = new Movies();
    movies.ShowDialog();
    //zapremo okno
    this.Close();
}
```

Slika 5 Link label

## 5.3 Prijava

Da lahko uporabljamo aplikacijo se moramo prijaviti. Ko se nam odpre okno za prijavo nam aplikacija ponuja možnost tudi za registracijo. Potrebno je skonfigurirati tudi Log4Net in Nhibernate za logiranje in povezavo s bazo.

```
}
//nhibernate konfiguracija
1 reference
private void ConfigureNHibernate()
{
    Configuration config = new Configuration();
    config.Configure();
    HbmSerializer.Default.Validate = true;
    config.AddInputStream(HbmSerializer.Default.Serialize(System.Reflection.Assembly.GetExecutingAssembly()));
    config.AddAssembly(typeof(Employees).Assembly);
    //schema update namesto export in bool za brisanje baze - ni potrebno zakomentirati ob naslednji uporabi
    new SchemaUpdate(config).Execute(true, true);
    mSessionFactory = config.BuildSessionFactory();
    mSession = mSessionFactory.OpenSession();
}

//log4net konfiguracija
1 reference
private void ConfigureLog4Net()
{
    log4net.Config.XmlConfigurator.Configure();
}
}
```

Slika 6 Prijava log4net ter nhibernate

Da pridemo na registracijo sem uporabil button\_click .

```
}
//preusmeritev na registracijo
1 reference
private void button3_Click(object sender, EventArgs e)
{
    fRegister = new Register();
    fRegister.SetNhib(mSessionFactory, mSession);
    fRegister.ShowDialog();
}
}
```

Slika 7 Registracija gumb

Za session sem uporabil button\_Click. Shranim si e-mail od prijavljenega uporabnika v session.txt.

```
//shranimo uporabnika v session.txt
1 reference
private void button1_Click(object sender, EventArgs e)
{
    string path = AppDomain.CurrentDomain.BaseDirectory + "session.txt";
    File.WriteAllText(path, textBox1.Text);
}
```

Slika 8 Session

Za preverjanje prijave sem v kodi uporabil dve funkciji: loginUser ter CheckUser.

```
//prijava
1 reference
private void LoginUser()
{
    //dovolimo 2 napačni prijavi textBox1 = username, textBox2 = password
    int count = 2;
    var fLoginCount = new Login();
    if(fLoginCount.IsDisposed)
    {
        fLoginCount = new Login();
    }
    while (count > 0)
    {
        if (fLoginCount.ShowDialog() == DialogResult.OK)
        {
            if (CheckUser (fLoginCount.Controls["textBox1"].Text, (fLoginCount.Controls["textBox2"].Text)))
            {
                count = -1;
                this.Text = "user:" + fLoginCount.Controls["textBox1"].Text;
            }
            else
            {
                MessageBox.Show("Wrong entry");
                count--;
            }
        }
    }
    fLoginCount.Dispose();
    if (count == 0)
    {
        System.Environment.Exit(0);
    }
}
```

Slika 9 LoginUser

```
//preverimo če so podatki pravilni
1 reference
private bool CheckUser(string email, string pass)
{
    bool result = false;
    try
    {
        ITransaction tx = mSession.BeginTransaction();
        IQuery query = mSession.CreateQuery("from Employees d where d.email = ?");
        Employees res = query.SetString(0, email).UniqueResult<Employees>();
        tx.Commit();
        if (res != null && res.pass.Equals(pass)){
            result = true;
        }
    }
    catch (Exception e1)
    {
        ResetSession();
    }
    return result;
}
```

Slika 10 Check User



## 5.4 Glavno okno

V glavnem oknu se prikažejo vsi filmi in število ogledov le njih.

```
//seznam filmov
1reference
private void Movies_Load(object sender, EventArgs e)
{
    movieList = mSession.CreateCriteria(typeof(Movie)).List<Movie>();
    dataMovies.ReadOnly = false;
    bs = new BindingSource();
    bs.DataSource = movieList;
    bs.AllowNew = true;
    // bs.AllowRemove = true;
    dataMovies.DataSource = bs;
    bs.ListChanged += new System.ComponentModel.ListChangedEventHandler(bindingSource1_ListChanged);
}

1reference
private void bindingSource1_ListChanged(object sender, ListChangedEventArgs e)
{
    switch (e.ListChangedType)
    {
        case ListChangedType.ItemChanged:
        {
            using (ITransaction tx = mSession.BeginTransaction())
            {
                Movie newMovie = (Movie)(bs.List[e.NewIndex]);
                if (newMovie.movie_name == null)
                    newMovie.movie_name = "";
                if (newMovie.visitors == null)
                    newMovie.visitors = "";
                mSession.Save(newMovie);
                tx.Commit();
            }
            break;
        }
    }
}
//switch
}
```

Slika 11 Glavno okno

Glavno okno ima na vrhu toolStripMenu.

```
// prikaz okna s seznamom zaposlenih
1reference
private void employeesToolStripMenuItem_Click(object sender, EventArgs e)
{
    var fEmployeeList = new Employee();
    fEmployeeList.SetNhib(mSessionFactory, mSession); // posreduje povezavo z bazo
    fEmployeeList.ShowDialog();
}
```

Slika 12 ToolStrip zaposleni

```
//odjava
1reference
private void logoutToolStripMenuItem_Click(object sender, EventArgs e)
{
    Application.Exit();
}

//spremeni podatke
1reference
private void accountChangeToolStripMenuItem_Click(object sender, EventArgs e)
{
    fAcInfo = new AccountInfo();
    fAcInfo.SetNhib(mSessionFactory, mSession);
    fAcInfo.ShowDialog();
}
```

Slika 13 Odjava

Dodal sem tudi možnost izvoza v XML datoteko.

```
//xml
1 reference
private void button2_Click(object sender, EventArgs e)
{
    List<string> movies = new List<string>();
    //polnjenje seznama
    foreach( DataGridViewRow row in dataMovies.Rows)
    {
        currMovie += row.Cells["movie_name"].Value;
        currMovie2 += row.Cells["visitors"].Value;
        movies.Add(currMovie);
        movies.Add(currMovie2);
    } //foreach
    //export v pdf
    var ser = new XmlSerializer(typeof(List<string>));
    TextWriter writer = new StreamWriter(@"E:\Movie.xml");
    // o is List<place> here
    ser.Serialize(writer, movies);
}
```

Slika 14 Xml izvoz

Dodal pa sem tudi možnost izbrisa posameznega filma.

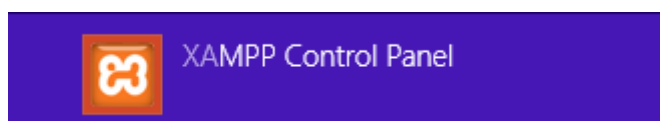
```
//delete
1 reference
private void button1_Click(object sender, EventArgs e)
{
    if (dataMovies.SelectedRows.Count > 0)
    {
        int selectedIndex = dataMovies.SelectedRows[0].Index;
        // dobimo rowID iz prve vrstice označene vrstice
        int rowID = int.Parse(dataMovies[0, selectedIndex].Value.ToString());
        dataMovies.AllowUserToDeleteRows = true;
        string query = "DELETE FROM movie WHERE ID_movie =" + rowID;
        mSession.CreateSQLQuery(query).ExecuteUpdate();
        MessageBox.Show("Movie deleted");
    } //if
}
}
```

Slika 15 Izbris filma

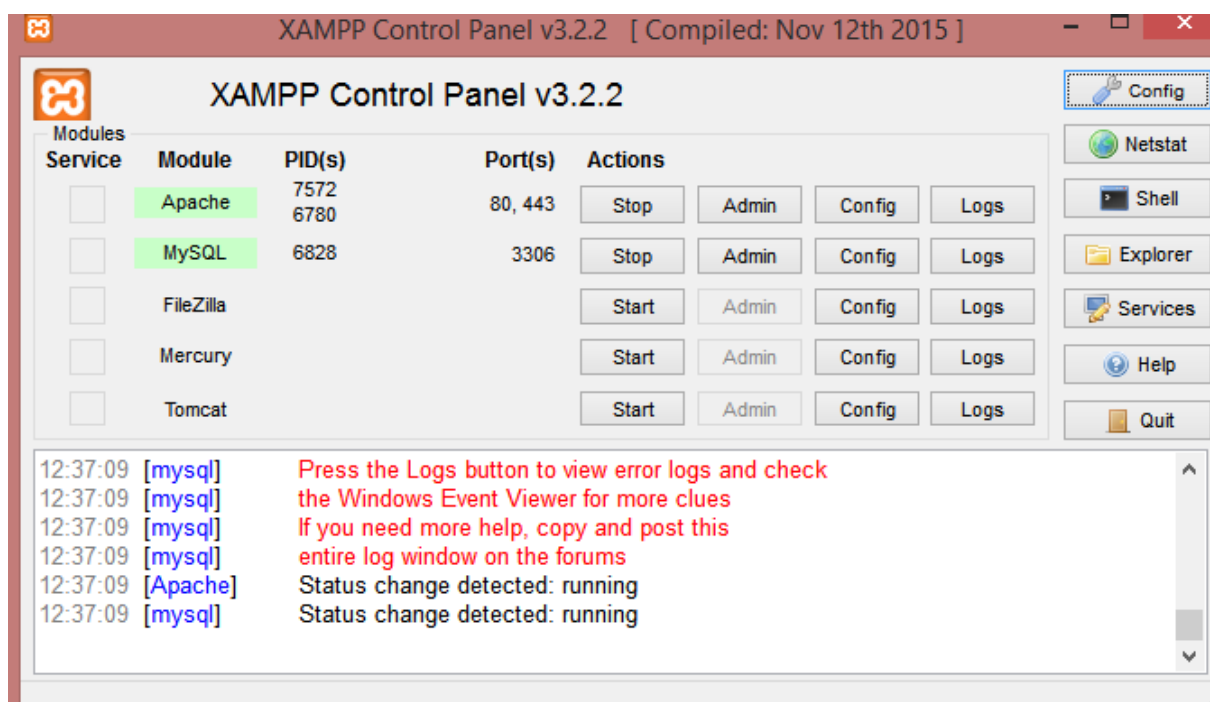
Ker je koda pri zaposlenih enaka, sem se odločil da ne bom dodajal vseh slik še pri njih. Celotna koda je na voljo v izvorni kodi.

## 6. Navodila za uporabo

Pred uporabo moramo zagnati bazo preko XAMPP.

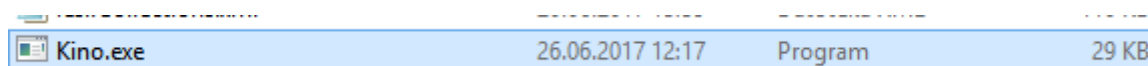


Slika 16 Xampp



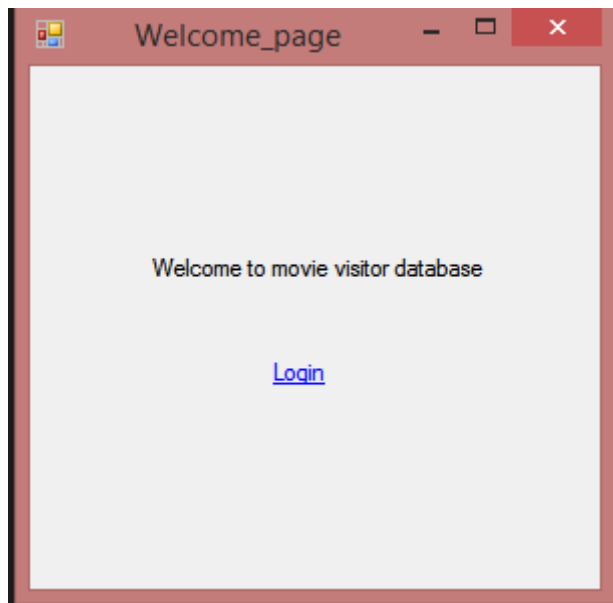
Slika 17 Xampp glavni meni

Aplikacijo zaženemo z dvoklikom na Kino.exe



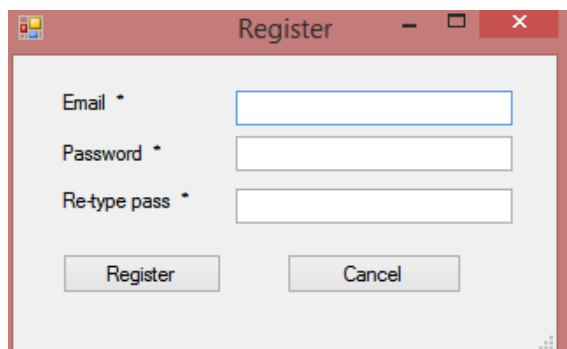
Slika 18 Zagon aplikacije

Odpre se nam pozdravno okno, kjer lahko kliknemo na prijavo

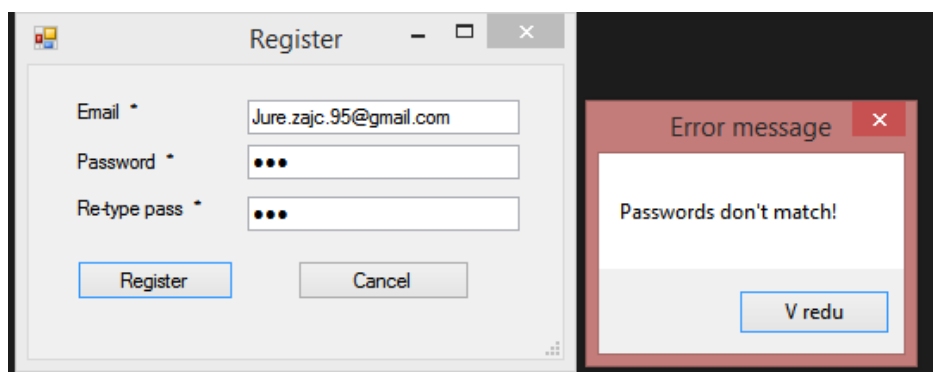


Slika 19 Pozdravno okno

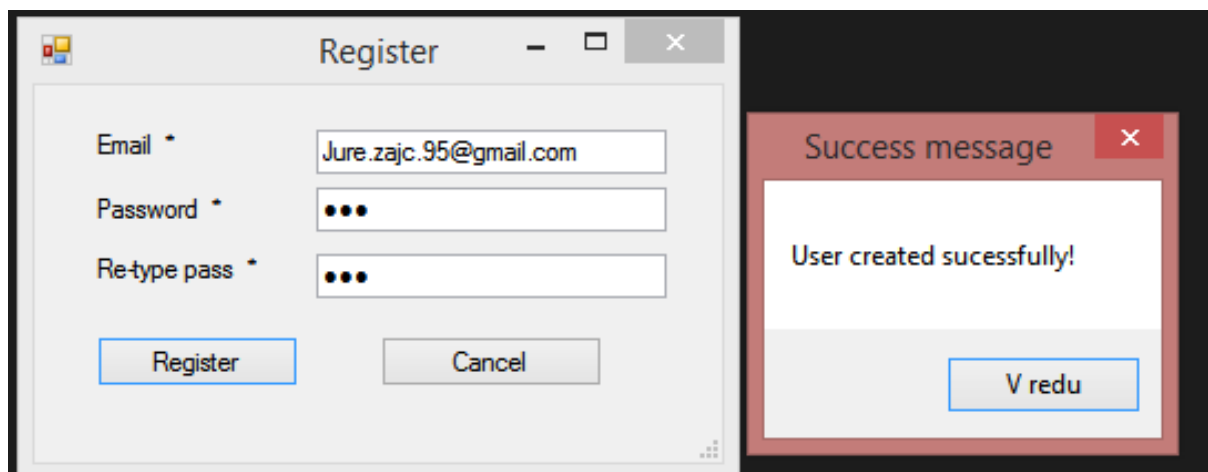
S klikom na Register se nam odpre novo okno kjer moramo izpolniti podatke za vpis. Geslo ponovimo dvakrat, če se gesli ne ujemata pozovemo uporabnika k ponovnemu vpisu gesla



Slika 20 Registracija

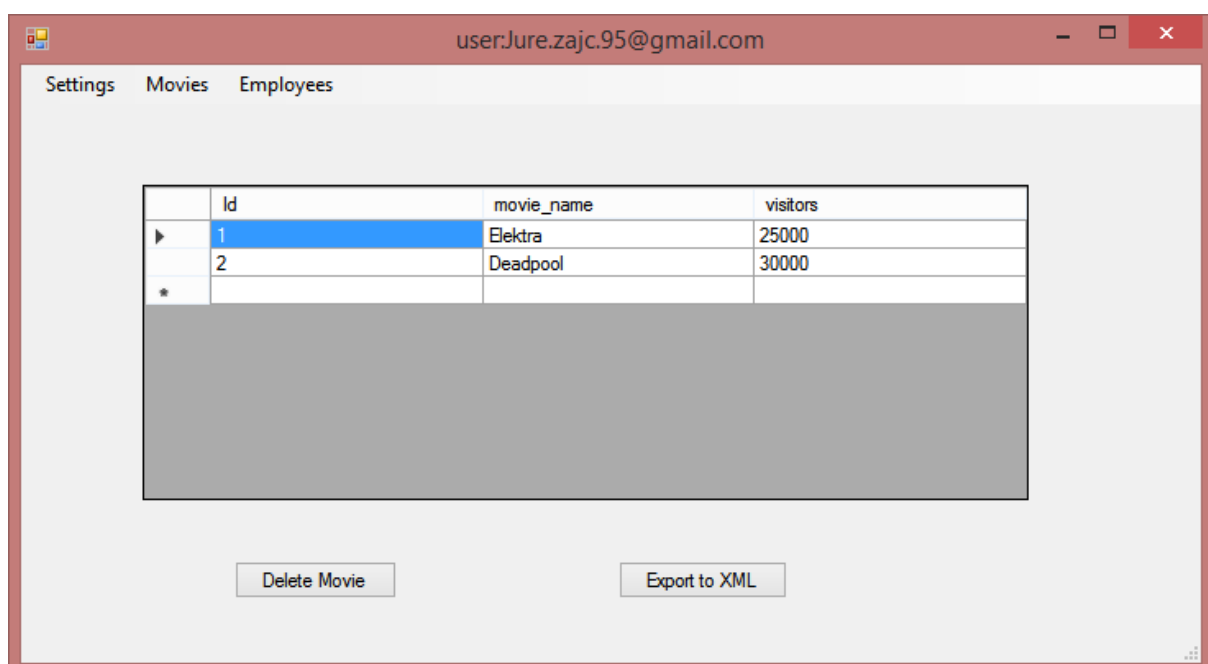


Slika 21 Napačna registracija



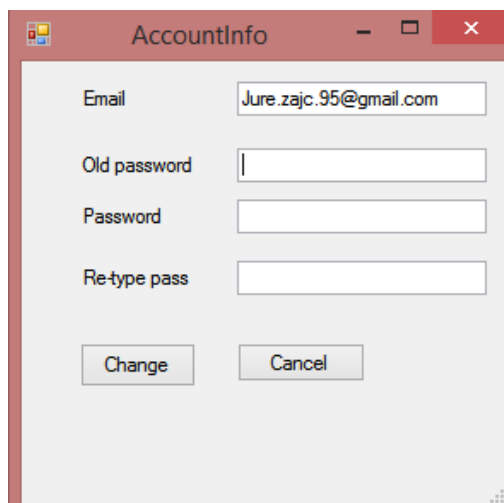
Slika 22 Uspešna registracija

Ko kliknemo na prijavo se nam odpre seznam filmov. Kot vidimo v zgornji vrstici si zapolni e-mail uporabnika. S klikom na export to XML se tabela filmov prenese v XML datoteko. Lahko pa tudi izbrišemo posamezen film iz baze to naredimo tako da označimo vrstico katero želimo izbrisati in kliknemo na gumb Delete Movie.



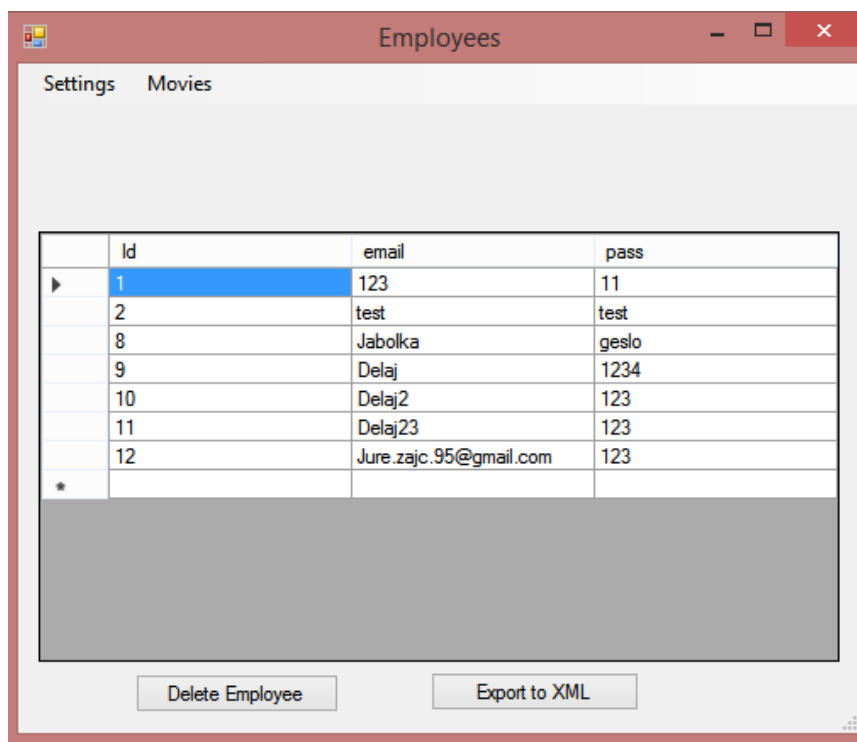
Slika 23 Glavno okno

Ko kliknemo na settings-account info se nam odpre pogovorno okno kjer lahko spremenimo geslo ter e-mail. Pri prehodu na to okno si program zapolni email uporabnika



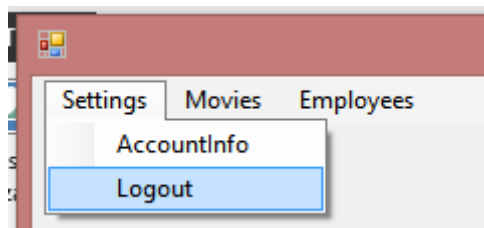
Slika 24 Urejanje profila

Ko kliknemo na gumb Employees se nam odpre seznam zaposlenih. Tudi ta seznam lahko izvozimo v XML dokument lahko tudi izbrisemo posameznega uporabnika



Slika 25 Zaposleni okno

S klikom na logout se uporabnik odjavi in aplikacija se zapre



Slika 26 Odjava

## 7. Zaključek

Za zaključek bi rad povedal, da je bilo ustvarjanje takšne aplikacija nekakšna nova zanimiva izkušnja zame. Aplikacija je praktične narave in primerna za ustvarjanje statistike.

Naučil sem se veliko novega in uporabnega znanja, verjamem da mi bo takšno znanje prišlo prav pri nadaljevanju študija ter v moji bodoči poklicni poti.