

DataLayer

FolderModals

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace DataLayer.Modals
{
    public class Obaveza
    {
        public int Id { get; set; }
        public string Dan { get; set; }
        public string Opis_obaveze { get; set; }
    }
}
```

Repozitor

```
using DataLayer.Modals;
using System;
using System.Collections.Generic;
using System.Data.SqlClient;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace DataLayer
{
    public class ObavezaRepsitory
    {
        public string connString = "Data Source=(localdb)\\MSSQLLocalDB;Initial
Catalog=DnevneObavezeDB;Integrated Security=True";

        public List<Obaveza> GetAllObaveze()
        {
            List<Obaveza> obavezaList = new List<Obaveza>();

            using (SqlConnection sqlConnection = new SqlConnection(connString))
            {
                sqlConnection.Open();

                SqlCommand command = new SqlCommand();
                command.Connection = sqlConnection;
                command.CommandText = "SELECT * FROM Obaveza";

                SqlDataReader reader = command.ExecuteReader();

                while (reader.Read())
                {
                    Obaveza obaveza = new Obaveza();
                    obaveza.Id = reader.GetInt32(0);
                    obaveza.Dan = reader.GetString(1);
                    obaveza.Opis_obaveze = reader.GetString(2);

                    obavezaList.Add(obaveza);
                }
            }
        }
    }
}
```

```

        }
    }
    return obavezaList;
}

public int InsertObaveza(Obaveza obaveza)
{
    using (SqlConnection sqlConnection = new SqlConnection(connString))
    {
        sqlConnection.Open();

        SqlCommand command = new SqlCommand();
        command.Connection = sqlConnection;
        command.CommandText = "INSERT INTO Obaveza (Dan, Opis_obaveze)
VALUES (@Dan, @Opis_obaveze)";
        command.Parameters.AddWithValue("@Dan", obaveza.Dan);
        command.Parameters.AddWithValue("@Opis_obaveze",
obaveza.Opis_obaveze);

        return command.ExecuteNonQuery();
    }
}

public int UpdateObaveza(Obaveza obaveza)
{
    using (SqlConnection sqlConnection = new SqlConnection(connString))
    {
        sqlConnection.Open();

        SqlCommand command = new SqlCommand();
        command.Connection = sqlConnection;
        command.CommandText = "UPDATE Obaveza SET Opis_obaveze =
@Opis_obaveze WHERE Id = @Id";
        command.Parameters.AddWithValue("@Id", obaveza.Id);
        command.Parameters.AddWithValue("@Opis_obaveze",
obaveza.Opis_obaveze);

        return command.ExecuteNonQuery();
    }
}

public int DeleteObaveza(Obaveza obaveza)
{
    using (SqlConnection sqlConnection = new SqlConnection(connString))
    {
        sqlConnection.Open();

        SqlCommand command = new SqlCommand();
        command.Connection = sqlConnection;
        command.CommandText = "DELETE FROM Obaveza WHERE Id = @Id";
        command.Parameters.AddWithValue("@Id", obaveza.Id);

        return command.ExecuteNonQuery();
    }
}
}
}
}

```

BusinessLayer

```
using DataLayer;
using DataLayer.Modals;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace BusinessLayer
{
    public class BusinessObaveza
    {
        readonly ObavezaRepository obavezaRepository = new ObavezaRepository();

        public List<Obaveza> GetObaveze()
        {
            return obavezaRepository.GetAllObaveze().ToList();
        }

        public List<Obaveza> GetObavezeZaDan(string odabraniDan)
        {
            return obavezaRepository.GetAllObaveze().Where(obaveza => obaveza.Dan
== odabraniDan).ToList();
        }

        public bool InsertObaveza(Obaveza obaveza)
        {
            return obavezaRepository.InsertObaveza(obaveza) != 0;
        }

        public bool UpadteObaveza(Obaveza obaveza)
        {
            return obavezaRepository.UpdateObaveza(obaveza) != 0;
        }

        public bool DelateObaveza(Obaveza obaveza)
        {
            return obavezaRepository.DeleteObaveza(obaveza) != 0;
        }
    }
}
```

PresentationLayer

Obaveze

```
using BusinessLayer;
using DataLayer.Modals;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace DnevneObaveze
{
    public partial class Obaveze : Form
    {
        readonly BusinessObaveza bo = new BusinessObaveza();
        public Obaveze()
        {
            InitializeComponent();

            private void Obaveze_Load(object sender, EventArgs e)
            {
                RefreshList();
            }

            public void RefreshList()
            {
                listBoxObaveze.Items.Clear();
                foreach (Obaveza o in bo.GetObaveze())
                {
                    listBoxObaveze.Items.Add($"{o.Opis_obaveze}");
                }
            }

            private void btnUnesi_Click(object sender, EventArgs e)
            {
                DodajObavezu dodaj = new DodajObavezu();

                dodaj.Show();

                this.Hide();
            }

            private void btnIzlistaj_Click(object sender, EventArgs e)
            {
                if (string.IsNullOrEmpty(comboBoxDan.Text))
                {
                    MessageBox.Show("Molimo vas da izaberete dan.", "Niste izabrali dan", MessageBoxButtons.OK, MessageBoxIcon.Warning);
                    comboBoxDan.Focus();
                }
                else
                {
                    listBoxObaveze.Items.Clear();
                    foreach (Obaveza o in bo.GetObavezeZaDan(comboBoxDan.Text))
                    {
                        listBoxObaveze.Items.Add($"{o.Opis_obaveze}");
                    }
                }
            }
        }
    }
}
```

```

    }
}
}
}
}

```

DodajObavezu

```

using BusinessLayer;
using DataLayer.Modals;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace DnevneObaveze
{
    public partial class DodajObavezu : Form
    {
        readonly BusinessObaveza bo = new BusinessObaveza();
        public DodajObavezu()
        {
            InitializeComponent();

            private void btnVratise_Click(object sender, EventArgs e)
            {
                Obaveze o = new Obaveze();

                o.Show();

                this.Hide();
            }

            private void btnUnesi_Click(object sender, EventArgs e)
            {
                if (string.IsNullOrEmpty(richTextBoxNovaObaveza.Text))
                {
                    MessageBox.Show("Molimo vas da unesete novu obavezu.", "Niste uneli obavezu", MessageBoxButtons.OK, MessageBoxIcon.Warning);
                    richTextBoxNovaObaveza.Focus();
                }
                else if (string.IsNullOrEmpty(comboBoxOdaberiDan.Text))
                {
                    MessageBox.Show("Molimo vas da izaberete dan.", "Niste izabrali dan", MessageBoxButtons.OK, MessageBoxIcon.Warning);
                    comboBoxOdaberiDan.Focus();
                }
                else
                {
                    Obaveza o = new Obaveza()
                    {
                        Opis_obaveze = richTextBoxNovaObaveza.Text,
                        Dan = comboBoxOdaberiDan.Text
                    };
                    bo.InsertObaveza(o);
                }
            }
        }
    }
}

```

```

        comboBoxOdaberiDan.Text = string.Empty;
        richTextBoxNovaObaveza.Text = string.Empty;
    }
}
}

```

Login

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace DnevneObaveze
{
    public partial class Login : Form
    {
        public Login()
        {
            InitializeComponent();

            private void Login_Load(object sender, EventArgs e)
            {
                txtIme.Focus();

            private void btnLogin_Click(object sender, EventArgs e)
            {
                if (string.IsNullOrEmpty(txtIme.Text))
                {
                    MessageBox.Show("Molimo vas da popunite polje ime pre nego sto nastavite.", "Nedostaju podaci", MessageBoxButtons.OK, MessageBoxIcon.Warning);
                    txtIme.Focus();
                }
                else if (txtIme.Text != "Marko") {
                    MessageBox.Show("Molim vas unesite pravlnu korisnicko ime", "Pogresno korisicko ime", MessageBoxButtons.OK, MessageBoxIcon.Warning);
                    txtIme.Text = " ";
                    txtIme.Focus();
                }
                else if (string.IsNullOrEmpty(txtPrezime.Text))
                {
                    MessageBox.Show("Molimo vas da popunite polje prezime pre nego sto nastavite.", "Nedostaju podaci", MessageBoxButtons.OK, MessageBoxIcon.Warning);
                    txtPrezime.Focus();
                }
                else if (txtPrezime.Text != "Mikic")
                {
                    MessageBox.Show("Molim vas unesite pravlnu korisnicko prezime", "Pogresno korisicko prezime", MessageBoxButtons.OK, MessageBoxIcon.Warning);
                    txtPrezime.Text = " ";
                    txtPrezime.Focus();
                }
            }
        }
    }
}

```

```
else
{
    Obaveze o = new Obaveze();
    o.Show();
    this.Hide();
}
}
}
```

Baza

-- Kreiranje tabele Dan

CREATE TABLE Dan

```
(
    Dan NVARCHAR(50) PRIMARY KEY
);
```

-- Kreiranje tabele Obaveza

CREATE TABLE Obaveza

```
(
    Id INT PRIMARY KEY IDENTITY,
    Dan NVARCHAR(50) REFERENCES dan(dan),
    Opis_obaveze NVARCHAR(255)
);
```

-- Unos podataka za tabelu Dan

```
INSERT INTO Dan (Dan) VALUES ('Ponedjeljak');
INSERT INTO Dan (Dan) VALUES ('Utorak');
INSERT INTO Dan (Dan) VALUES ('Sreda');
INSERT INTO Dan (Dan) VALUES ('Cetvrtak');
INSERT INTO Dan (Dan) VALUES ('Petak');
INSERT INTO Dan (Dan) VALUES ('Subota');
INSERT INTO Dan (Dan) VALUES ('Nedjelja');
```

-- Unos podataka za Ponedjeljak

```
INSERT INTO Obaveza (Dan, Opis_obaveze) VALUES ('Ponedjeljak', 'Skuvaj Kafu');
INSERT INTO Obaveza (Dan, Opis_obaveze) VALUES ('Ponedjeljak', 'Doruckuj');
INSERT INTO Obaveza (Dan, Opis_obaveze) VALUES ('Ponedjeljak', 'Iznesi smece');
```

-- Unos podataka za Utorak

```
INSERT INTO Obaveza (Dan, Opis_obaveze) VALUES ('Utorak', 'Operi auto');
```



```
INSERT INTO Obaveza (Dan, Opis_obaveze) VALUES ('Utorak', 'Prosetaj psa');
INSERT INTO Obaveza (Dan, Opis_obaveze) VALUES ('Utorak', 'Idi u teretanu');
```

-- Unos podataka za Sredu

```
INSERT INTO Obaveza (Dan, Opis_obaveze) VALUES ('Sreda', 'Uradi projekat');
INSERT INTO Obaveza (Dan, Opis_obaveze) VALUES ('Sreda', 'Procitaj knjigu');
INSERT INTO Obaveza (Dan, Opis_obaveze) VALUES ('Sreda', 'Kupi karte za bioskop');
```

-- Unos podataka za Cetvrtak

```
INSERT INTO Obaveza (Dan, Opis_obaveze) VALUES ('Cetvrtak', 'Ispeglaj ves');
INSERT INTO Obaveza (Dan, Opis_obaveze) VALUES ('Cetvrtak', 'Spakuj ves');
INSERT INTO Obaveza (Dan, Opis_obaveze) VALUES ('Cetvrtak', 'Usisaj kucu');
```

-- Unos podataka za Petak

```
INSERT INTO Obaveza (Dan, Opis_obaveze) VALUES ('Petak', 'Idi do pijace');
INSERT INTO Obaveza (Dan, Opis_obaveze) VALUES ('Petak', 'Kupi nove knjige');
INSERT INTO Obaveza (Dan, Opis_obaveze) VALUES ('Petak', 'Plati racune');
```

-- Unos podataka za Subotu

```
INSERT INTO Obaveza (Dan, Opis_obaveze) VALUES ('Subota', 'Sredi sobu');
INSERT INTO Obaveza (Dan, Opis_obaveze) VALUES ('Subota', 'Napisi seminarski');
INSERT INTO Obaveza (Dan, Opis_obaveze) VALUES ('Subota', 'Idi u bioskop');
```

-- Unos podataka za Nedelju

```
INSERT INTO Obaveza (Dan, Opis_obaveze) VALUES ('Nedelja', 'Idi u prodavnicu');
INSERT INTO Obaveza (Dan, Opis_obaveze) VALUES ('Nedelja', 'Idi u teretanu');
INSERT INTO Obaveza (Dan, Opis_obaveze) VALUES ('Nedelja', 'Spremi veceru');
```