 Univerzitet u Novom Sadu

 Tehnički fakultet »MihajloPupin« Zrenjanin

**SEMINARSKI RAD**

**Predmet: Razvoj softvera otvorenog koda**

Tema: ASPX web aplikacija za všenje dokumentoavanja uvoza

Profesor : Studenti:

Doc. Dr Zoltan Kazi Slobodan Nadrljanski SI 21/15   
 Maribel Karaman SI 13/15

Zrenjanin, 2019. godina

Sadržaj

[Opis poslovnog procesa 3](#_Toc17713322)

[Specifikacija funkcionalnih zahteva sa profilima korisnika 3](#_Toc17713323)

[Modeli u dizajnu softverskog rešenja 4](#_Toc17713324)

[CDM Dijagram 5](#_Toc17713325)

[Opis implementacije 6](#_Toc17713326)

[Sql skripta za kreiranje baze podataka i store procedura 7](#_Toc17713327)

[Klasa podataka sa pozivima store procedura 10](#_Toc17713328)

[Prezentaciona logika 18](#_Toc17713329)

[Korisniči interfejs 24](#_Toc17713330)

[Zakljucak 26](#_Toc17713331)

[Literatura 27](#_Toc17713332)

# 

# Opis poslovnog procesa

**Poslovni proces**

Dokumnetovanje uvoza je obavezna i zakonom definasana stvar koju mora izvršiti ovlašćeni uvoznik, za potrebu svog klijenta. Prilikom samog uvoza, uvoznik je u obavezi da ispuni nabavku odnosno da u kontaktu sa klijentom dogovori nabavku određenog artikla ili više istih.

Prilikom samog uvoza, uvoznik je u obavezi da ispuni nabavku, u kojoj će obrazložiti šta uvozi kao i svoje podatke, odnosno podatke firme čiji je predstavnik. Takođe radnik koji vrši uvoz može i ima pravo da klijentu prikaže narudžbinu kao i više istih radi utvrđivanja i verifikovanja da je to što je klijent tražio. Takve stvke su u glavnom ispisane u zasebnim tabelema kojima i klijent ima pristup, ali bez adminskog prava da neku narudžbinu prepravlja.

**Poslovno proavilo**

Nakon odrađenog broja narudžbina, odnosno nabavki preko određenog uvoznika, klijent stiče pravo da određenu novčanu uštedu, odnsno popust u određenom procentu, u zavisnosti o kakvoj se vrsti uvoza radi, odnosno samoj ceni mašine koja se uvozi.

# Specifikacija funkcionalnih zahteva sa profilima korisnika

Profil korisnika: Radnik u firmi - Uvoznik

* Unos podataka za logovanje uvoznika
* Unos podataka za naručivnjae nabavke
* Prikaz nabavki koje su naručene

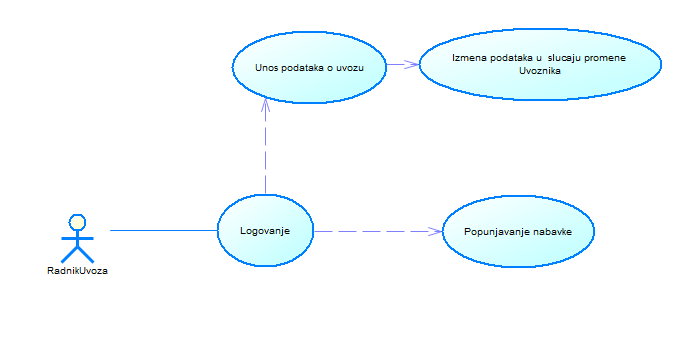
Profil: Klijent

* Tabelarni prikaz svih nabavki

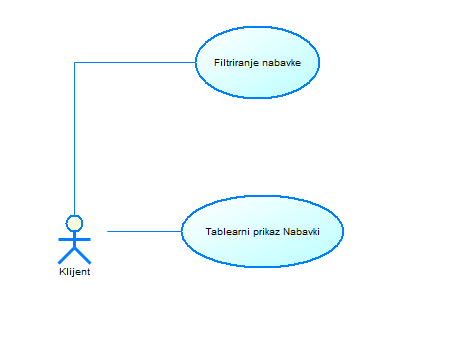
# Modeli u dizajnu softverskog rešenja

USE CASE Diagram

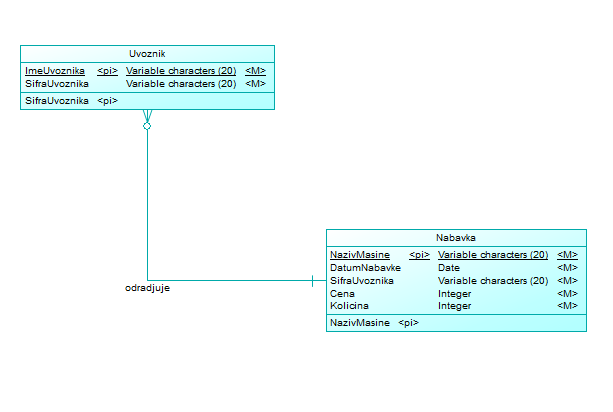
Prikaz aktivnosi radnika u uvozu

****

Prikaz aktivnosi klijenta

****

# CDM Dijagram

****

# Opis implementacije

**Kratak opis primenjenih tehnologija**

U ovom seminarskom radu korišćen je Visual Studio .NET 2010 i Microsoft SQL Server 2008 R2. U okviru realizacije programskog koda:

Korisnički interfejs predstavljaju ASPX web forme

Klase za rad sa podacima sa korisničkog interfejsa I baze podataka su iz standardnog paketa klasa za rad sa SQL serverom (SQLClient).

**Tabelarni prikaz strukture slojeva i podslojeva aplikacije**

|  |  |  |
| --- | --- | --- |
| Sloj | Podsloj | Realizacija u seminarskom radu |
| Prezentacioni sloj | Korisnicki interfejs | KI.sln (ASPX web forms) |
| Sloj za rad sa podacima | Klase podataka | KlasePodataka.dll (svaka tabela u bazi podataka ima 3 klase – pojedinac, lista i DB) |

# 

# Sql skripta za kreiranje baze podataka i store procedura

Baza podataka

|  |
| --- |
| CREATE DATABASE [UVOZ\_2019]  GO  USE [UVOZ\_2019]  GO  CREATE TABLE [dbo].[Uvoznik](  [SifraUvoznika] [nvarchar] (10) NOT NULL PRIMARY KEY,  [ImeUvoznika] [nvarchar](20) NOT NULL    )  GO  CREATE TABLE [dbo].[Nabavka](  [NazivMasine] [nvarchar] (20) NOT NULL PRIMARY KEY,  [DatumNabavke] [date] NOT NULL,  [SifraUvoznika] [nvarchar] (10) NOT NULL,  [Cena] [int] NOT NULL,  [Kolicina] [int] NOT NULL  CONSTRAINT fk\_UvoznikNabavka FOREIGN KEY(SifraUvoznika) REFERENCES Uvoznik(SifraUvoznika)  )  GO |

Store procedura Nabavke

|  |
| --- |
| CREATE PROCEDURE [DodajNovuNabavku]  (  @NazivMasine nvarchar(20),  @DatumNabavke date,  @SifraUvoznika nvarchar(10),  @Cena int,  @Kolicina int  )  AS  BEGIN  Insert into Nabavka(NazivMasine, DatumNabavke, SifraUvoznika, Cena, Kolicina) values (@NazivMasine, @DatumNabavke, @SifraUvoznika, @Cena, @Kolicina)  END  GO  CREATE PROCEDURE [DajSveNabavke]  AS  select \* from Nabavka  GO  CREATE Procedure [DajNabavkuPoNazivuMasine]  (  @NazivMasine nvarchar (20)  )  as  select \* from Nabavka where NazivMasine=@NazivMasine  Go  CREATE PROCEDURE [DajCenuNabavkePoNazivuMasine]  (  @NazivMasine nvarchar (20),  @Cena int  )  as  select \* from Nabavka where NazivMasine=@NazivMasine and Cena=@Cena  GO |

Store procedura Uvoznika

|  |
| --- |
| USE [UVOZ\_2019]  GO  CREATE PROCEDURE [DajSveUvoznike]  AS  select \* from Uvoznik  GO  CREATE PROCEDURE [DajUvoznikaPoNazivu]  (  @ImeUvoznika nvarchar(20)  )  AS  select \* from Uvoznik where ImeUvoznika=@ImeUvoznika  GO  CREATE PROCEDURE [DajUvoznikaPoSifri]  (  @SifraUvoznika nvarchar(10)  )  AS  select \* from Uvoznik where SifraUvoznika=@SifraUvoznika  GO  CREATE PROCEDURE [DodajNovgUvoznika]  (  @SifraUvoznika char(10),  @ImeUvoznika nvarchar(20)  )  AS  BEGIN  Insert into Uvoznik(SifraUvoznika, ImeUvoznika) values (@SifraUvoznika, @ImeUvoznika)  END  GO  CREATE PROCEDURE [ObrisiUvoznika]  (  @SifraUvoznika char(10)  )  AS  BEGIN  Delete from Uvoznik where SifraUvoznika=@SifraUvoznika  END  GO  CREATE PROCEDURE [IzmeniUvoznika](  @StaraSifra char(10),  @SifraUvoznika char(10),  @ImeUvoznika nvarchar(20)  )  AS  BEGIN  Update Uvoznik set SifraUvoznika=@SifraUvoznika, ImeUvoznika=@ImeUvoznika where SifraUvoznika=@StaraSifra  END  GO |

# Klasa podataka sa pozivima store procedura

Poziv procedure nad Nabavkom

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  //  using System.Data;  using System.Data.SqlClient;  namespace KlasePodataka  {  public class clsNabavkaDB  {  private string pStringKonekcije;  public string StringKonekcije  {  get  {  return pStringKonekcije;  }  }    public clsNabavkaDB(string NoviStringKonekcije)    {  pStringKonekcije = NoviStringKonekcije;  }  public DataSet DajSveNabavke()  {    DataSet dsPodaci = new DataSet();  SqlConnection Veza = new SqlConnection(pStringKonekcije);  Veza.Open();  SqlCommand Komanda = new SqlCommand("DajSveNabavke", Veza);  Komanda.CommandType = CommandType.StoredProcedure;  SqlDataAdapter da = new SqlDataAdapter();  da.SelectCommand = Komanda;  da.Fill(dsPodaci);  Veza.Close();  Veza.Dispose();    return dsPodaci;  }  public DataSet DajNabavkuPoNazivuMasine(string NazivMasine)  {    DataSet dsPodaci = new DataSet();  SqlConnection Veza = new SqlConnection(pStringKonekcije);  Veza.Open();  SqlCommand Komanda = new SqlCommand("DajNabavkuPoNazivuMasine", Veza);  Komanda.CommandType = CommandType.StoredProcedure;  Komanda.Parameters.Add("@NazivMasine", SqlDbType.NVarChar).Value = NazivMasine;  SqlDataAdapter da = new SqlDataAdapter();  da.SelectCommand = Komanda;  da.Fill(dsPodaci);  Veza.Close();  Veza.Dispose();  return dsPodaci;  }  public int DajCenuNabavkePoNazivuMasine(string NazivMasine)  {  int Cena = 0;  DataSet dsPodaci = new DataSet();  SqlConnection Veza = new SqlConnection(pStringKonekcije);  Veza.Open();  SqlCommand Komanda = new SqlCommand("DajCenuNabavkePoNazivuMasine", Veza);  Komanda.CommandType = CommandType.StoredProcedure;  Komanda.Parameters.Add("@NazivMasine", SqlDbType.NVarChar).Value = NazivMasine;  SqlDataAdapter da = new SqlDataAdapter();  da.SelectCommand = Komanda;  da.Fill(dsPodaci);  Veza.Close();  Veza.Dispose();  Cena = int.Parse(dsPodaci.Tables[0].Rows[0].ItemArray[3].ToString());  return Cena;  }  public bool SnimiNovuNabavku(clsNabavka objNovaNabavka)  {  int brojSlogova = 0;  SqlConnection Veza = new SqlConnection(pStringKonekcije);  Veza.Open();  SqlCommand Komanda = new SqlCommand("DodajNovuNabavku", Veza);  Komanda.CommandType = CommandType.StoredProcedure;  Komanda.Parameters.Add("@NazivMasine", SqlDbType.VarChar).Value = objNovaNabavka.NazivMasine;  Komanda.Parameters.Add("@DatumNabavke", SqlDbType.Date).Value = objNovaNabavka.DatumNabavke;  Komanda.Parameters.Add("@SifraUvoznika", SqlDbType.VarChar).Value = objNovaNabavka.SifraUvoznika;  Komanda.Parameters.Add("@Cena", SqlDbType.Int).Value = objNovaNabavka.Cena;  Komanda.Parameters.Add("@Kolicina", SqlDbType.Int).Value = objNovaNabavka.Kolicina;    brojSlogova = Komanda.ExecuteNonQuery();  Veza.Close();  Veza.Dispose();  return (brojSlogova > 0);  }    }  } |

Poziv procedure nad Uvoznikom

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  //  using System.Data;  using System.Data.SqlClient;  namespace KlasePodataka  {  public class clsUvoznikDB  {  private string pStringKonekcije;  public string StringKonekcije  {  get  {  return pStringKonekcije;  }  set  {  if (this.pStringKonekcije != value)  this.pStringKonekcije = value;  }  }  public clsUvoznikDB(string NoviStringKonekcije)  {  pStringKonekcije = NoviStringKonekcije;  }  public DataSet DajSveUvoznike()  {  DataSet dsPodaci = new DataSet();  SqlConnection Veza = new SqlConnection(pStringKonekcije);  Veza.Open();  SqlCommand Komanda = new SqlCommand("DajSveUvoznike", Veza);  Komanda.CommandType = CommandType.StoredProcedure;  SqlDataAdapter da = new SqlDataAdapter();  da.SelectCommand = Komanda;  da.Fill(dsPodaci);  Veza.Close();  Veza.Dispose();  return dsPodaci;  }  public DataSet DajUvoznikaPoSifri(string SifraUvoznika)  {  DataSet dsPodaci = new DataSet();  SqlConnection Veza = new SqlConnection(pStringKonekcije);  Veza.Open();  SqlCommand Komanda = new SqlCommand("DajUvoznikaPoSifri", Veza);  Komanda.CommandType = CommandType.StoredProcedure;  Komanda.Parameters.Add("@SifraUvoznika", SqlDbType.NVarChar).Value = SifraUvoznika;  SqlDataAdapter da = new SqlDataAdapter();  da.SelectCommand = Komanda;  da.Fill(dsPodaci);  Veza.Close();  Veza.Dispose();  return dsPodaci.Tables[0].Rows[0].ItemArray[1].ToString()  }  public string DajNazivUvoznikaPremaSifriUvoznika(string SifraUvoznika)  {  string ImeUvoznika = "";  DataSet dsPodaci = new DataSet();  SqlConnection Veza = new SqlConnection(pStringKonekcije);  Veza.Open();  SqlCommand Komanda = new SqlCommand("DajUvoznikaPoSifri", Veza);  Komanda.CommandType = CommandType.StoredProcedure;  Komanda.Parameters.Add("@SifraUvoznika", SqlDbType.NVarChar).Value = SifraUvoznika;  SqlDataAdapter da = new SqlDataAdapter();  da.SelectCommand = Komanda;  da.Fill(dsPodaci);  Veza.Close();  Veza.Dispose();  ImeUvoznika = dsPodaci.Tables[0].Rows[0].ItemArray[1].ToString();  return ImeUvoznika;  }  public string DajSifruUvoznikaPoNazivuUvoznika(string ImeUvoznika)  {  DataSet dsPodaci = new DataSet();  SqlConnection Veza = new SqlConnection(pStringKonekcije);  Veza.Open();  SqlCommand Komanda = new SqlCommand("DajUvoznikaPoNazivu", Veza);  Komanda.CommandType = CommandType.StoredProcedure;  Komanda.Parameters.Add("@ImeUvoznika", SqlDbType.NVarChar).Value = ImeUvoznika;  SqlDataAdapter da = new SqlDataAdapter();  da.SelectCommand = Komanda;  da.Fill(dsPodaci);  Veza.Close();  Veza.Dispose();  return dsPodaci.Tables[0].Rows[0].ItemArray[0].ToString();  }  public DataSet DajUvoznikaPoNazivu(string ImeUvoznika)  {  DataSet dsPodaci = new DataSet();  SqlConnection Veza = new SqlConnection(pStringKonekcije);  Veza.Open();  SqlCommand Komanda = new SqlCommand("DajUvoznikaPoNazivu", Veza);  Komanda.CommandType = CommandType.StoredProcedure;  Komanda.Parameters.Add("@ImeUvoznika", SqlDbType.NVarChar).Value = ImeUvoznika;  SqlDataAdapter da = new SqlDataAdapter();  da.SelectCommand = Komanda;  da.Fill(dsPodaci);  Veza.Close();  Veza.Dispose();  return dsPodaci;  }  public DataSet DajUvoznikaPoNazivu(clsUvoznik objUvoznikZaFilter)  {  DataSet dsPodaci = new DataSet();  SqlConnection Veza = new SqlConnection(pStringKonekcije);  Veza.Open();  SqlCommand Komanda = new SqlCommand("DajUvoznikaPoNazivu", Veza);  Komanda.CommandType = CommandType.StoredProcedure;  Komanda.Parameters.Add("@ImeUvoznika", SqlDbType.NVarChar).Value = objUvoznikZaFilter.ImeUvoznika;  SqlDataAdapter da = new SqlDataAdapter();  da.SelectCommand = Komanda;  da.Fill(dsPodaci);  Veza.Close();  Veza.Dispose();  return dsPodaci;  }  public bool SnimiNovgUvoznika(clsUvoznik objNovUvoznik)  {  int brojSlogova = 0;  SqlConnection Veza = new SqlConnection(pStringKonekcije);  Veza.Open();  SqlCommand Komanda = new SqlCommand("DodajNovgUvoznika", Veza);  Komanda.CommandType = CommandType.StoredProcedure;  Komanda.Parameters.Add("@SifraUvoznika", SqlDbType.VarChar).Value = objNovUvoznik.SifraUvoznika;  Komanda.Parameters.Add("@ImeUvoznika", SqlDbType.NVarChar).Value = objNovUvoznik.ImeUvoznika;  brojSlogova = Komanda.ExecuteNonQuery();  Veza.Close();  Veza.Dispose();  return (brojSlogova > 0);  }  public bool ObrisiUvoznika(clsUvoznik objUvoznikZaBrisanje)  {  int brojSlogova = 0;  SqlConnection Veza = new SqlConnection(pStringKonekcije);  Veza.Open();  SqlCommand Komanda = new SqlCommand("ObrisiUvoznika", Veza);  Komanda.CommandType = CommandType.StoredProcedure;  Komanda.Parameters.Add("@SifraUvoznika", SqlDbType.VarChar).Value = objUvoznikZaBrisanje.SifraUvoznika;  brojSlogova = Komanda.ExecuteNonQuery();  Veza.Close();  Veza.Dispose();  return (brojSlogova > 0);  }  public bool ObrisiUvoznika(string SifraUvoznikaZaBrisanje)  {  int brojSlogova = 0;  SqlConnection Veza = new SqlConnection(pStringKonekcije);  Veza.Open();  SqlCommand Komanda = new SqlCommand("ObrisiUvoznika", Veza);  Komanda.CommandType = CommandType.StoredProcedure;  Komanda.Parameters.Add("@SifraUvoznika", SqlDbType.VarChar).Value = SifraUvoznikaZaBrisanje;  brojSlogova = Komanda.ExecuteNonQuery();  Veza.Close();  Veza.Dispose();  return (brojSlogova > 0);  }  public bool IzmeniUvoznika(clsUvoznik objStariUvoznik, clsUvoznik objNoviUvoznik)  {  int brojSlogova = 0;  SqlConnection Veza = new SqlConnection(pStringKonekcije);  Veza.Open();  SqlCommand Komanda = new SqlCommand("IzmeniUvoznika", Veza);  Komanda.CommandType = CommandType.StoredProcedure;  Komanda.Parameters.Add("@StaraSifra", SqlDbType.VarChar).Value = objStariUvoznik.SifraUvoznika;  Komanda.Parameters.Add("@SifraUvoznika", SqlDbType.VarChar).Value = objNoviUvoznik.SifraUvoznika;  Komanda.Parameters.Add("@ImeUvoznika", SqlDbType.NVarChar).Value = objNoviUvoznik.ImeUvoznika;  brojSlogova = Komanda.ExecuteNonQuery();  Veza.Close();  Veza.Dispose();  return (brojSlogova > 0);  }  public bool IzmeniUvoznika(string SifraStarogUvoznika, clsUvoznik objNoviUvoznik)  {  int brojSlogova = 0;  SqlConnection Veza = new SqlConnection(pStringKonekcije);  Veza.Open();  SqlCommand Komanda = new SqlCommand("IzmeniUvoznika", Veza);  Komanda.CommandType = CommandType.StoredProcedure;  Komanda.Parameters.Add("@StaraSifra", SqlDbType.VarChar).Value = SifraStarogUvoznika;  Komanda.Parameters.Add("@SifraUvoznika", SqlDbType.VarChar).Value = objNoviUvoznik.SifraUvoznika;  Komanda.Parameters.Add("@ImeUvoznika", SqlDbType.NVarChar).Value = objNoviUvoznik.ImeUvoznika;  brojSlogova = Komanda.ExecuteNonQuery();  Veza.Close();  Veza.Dispose();  return (brojSlogova > 0);  }  }  } |

# 

# Prezentaciona logika

Kod tabelarnog prikaza

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  //  using System.Data;  using KlasePodataka;  namespace PrezentacionaLogika  {  public class clsNabavkaTabelarnaForma  {  public string pStringKonekcije;  public clsNabavkaTabelarnaForma(string NoviStringKonekcije)  {    pStringKonekcije = NoviStringKonekcije;  }  public DataSet DajPodatkeZaGrid(string Filter)  {  DataSet dsPodaci = new DataSet ();  clsNabavkaDB objRacunDB = new clsNabavkaDB(pStringKonekcije);  if (Filter.Equals (""))  {  dsPodaci = objRacunDB.DajSveNabavke();  }  else  {  dsPodaci = objRacunDB.DajNabavkuPoNazivuUvoznika(Filter);  }  return dsPodaci;  }  }  } |

Kod tabelarnog unosa

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  //  using System.Data;  using KlasePodataka;  using Polsovnalogika;  namespace PrezentacionaLogika  {  public class clsNabavkaUnosForma  {  private string pNazivMasine;  private DateTime pDatumNabavke;  private string pSifraUvoznika;  private string pCena;  private string pKolicina;  private string pStringKonekcije;    public clsNabavkaUnosForma(string NoviStringKonekcije)  {  pStringKonekcije = NoviStringKonekcije;  }  public string NazivMasine  {  get { return pNazivMasine; }  set { pNazivMasine = value; }  }  public DateTime DatumNabavke  {  get { return pDatumNabavke; }  set { pDatumNabavke = value; }  }  public string SifraUvoznika  {  get { return pSifraUvoznika; }  set { pSifraUvoznika = value; }  }  public string Cena  {  get { return pCena; }  set { pCena = value; }  }  public string Kolicina  {  get { return pKolicina; }  set { pKolicina = value; }  }  public DataSet DajPodatkeZaCombo()  {  WSMasine1.Service1 objServisMasine = new WSMasine1.Service1();  DataSet dsPodaci = new DataSet();  dsPodaci = objServisMasine.DajSveMasine();  return dsPodaci;  }    public bool DaLiSuPodaciIspravni()  {  bool IspravniPodaci = false;  if ((this.pNazivMasine.Equals("")) || (this.pDatumNabavke.Equals("")) || (this.pSifraUvoznika.Equals("")) || (this.pCena.Equals("")) || (this.pKolicina == null))  {  IspravniPodaci = false;  }  else  {  int KolicinaInteger = 0;  bool uspehKonvezijeUInt = int.TryParse(this.pKolicina, out KolicinaInteger);  if (!uspehKonvezijeUInt)  {  IspravniPodaci = false;  }  else  {  if (this.pNazivMasine.Length > 3)  {  IspravniPodaci = false;  }  else  {  clsNabavkaDB objNabavkaDB = new clsNabavkaDB(pStringKonekcije);  DataSet dsPodaciNabavke = objNabavkaDB.DajNabavkuPoNazivuMasine(this.pNazivMasine);  bool NePostojiSifraUBazi = (dsPodaciNabavke.Tables[0].Rows.Count == 0);  IspravniPodaci = NePostojiSifraUBazi;  }  }  }  return IspravniPodaci;  }  public bool SnimiPodatke()  {  bool uspeh = false;  clsNabavkaDB objNabavkaDB = new clsNabavkaDB(pStringKonekcije);  clsNabavka objNabavka = new clsNabavka();    objNabavka.NazivMasine = this.pNazivMasine;  objNabavka.DatumNabavke = this.pDatumNabavke;  objNabavka.SifraUvoznika = this.pSifraUvoznika;  objNabavka.Cena = int.Parse(this.Cena);  objNabavka.Kolicina = int.Parse(this.Kolicina);    uspeh = objNabavkaDB.SnimiNovuNabavku(objNabavka);  return uspeh;  }  }  } |

Kod izmene i brisanja uvoznika u bazi

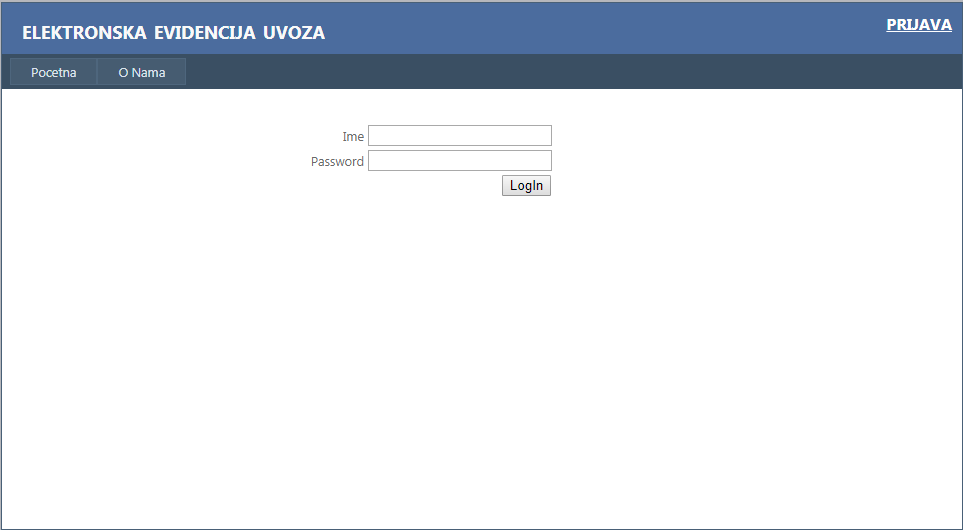
|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  //  using System.Data;  using KlasePodataka;  namespace PrezentacionaLogika  {  public class clsFormaUvoznikDetaljiEdit  {    private string pStringKonekcije;  private clsUvoznikDB objUvoznikDB;  private clsUvoznik objPreuzetiUvoznik;  private clsUvoznik objIzmenjeniUvoznik;  private string pSifraPreuzetogUvoznika;  private string pNazivPreuzetogUvoznika;  private string pSifraIzmenjenogUvoznika;  private string pNazivIzmenjenogUvoznika;    public string SifraPreuzetogUvoznika  {  get { return pSifraPreuzetogUvoznika; }  set { pSifraPreuzetogUvoznika = value; pNazivPreuzetogUvoznika = DajNaziv(pSifraPreuzetogUvoznika); }  }  public string NazivPreuzetogUvoznika  {  get { return pNazivPreuzetogUvoznika; }    }  public string SifraIzmenjenogUvoznika  {  get { return pSifraIzmenjenogUvoznika; }  set { pSifraIzmenjenogUvoznika = value; }  }  public string NazivIzmenjenogUvoznika  {  get { return pNazivIzmenjenogUvoznika; }  set { pNazivIzmenjenogUvoznika = value; }  }  public clsFormaUvoznikDetaljiEdit(string NoviStringKonekcije)  {  pStringKonekcije = NoviStringKonekcije;  objUvoznikDB = new clsUvoznikDB(pStringKonekcije);  }  private string DajNaziv(string SifraUvoznika)  {  string pomNaziv ="";  DataSet dsPodaci = new DataSet();  pomNaziv = objUvoznikDB.DajNazivUvoznikaPremaSifriUvoznika(SifraUvoznika);  return pomNaziv;  }  public bool ObrisiUvoznika()  {  bool uspehBrisanja = false;  uspehBrisanja = objUvoznikDB.ObrisiUvoznika(pSifraPreuzetogUvoznika);  return uspehBrisanja;  }  public bool IzmeniUvoznika()  {  bool uspehIzmene = false;  objPreuzetiUvoznik = new clsUvoznik();  objIzmenjeniUvoznik = new clsUvoznik();  objPreuzetiUvoznik.SifraUvoznika = pSifraPreuzetogUvoznika;  objPreuzetiUvoznik.ImeUvoznika = pNazivPreuzetogUvoznika;  objIzmenjeniUvoznik.SifraUvoznika = pSifraIzmenjenogUvoznika;  objIzmenjeniUvoznik.ImeUvoznika = pNazivIzmenjenogUvoznika;  uspehIzmene = objUvoznikDB.IzmeniUvoznika(objPreuzetiUvoznik, objIzmenjeniUvoznik);  return uspehIzmene;  }  }  } |

Kod stampanja spiska nabavki uz pomoć filtera

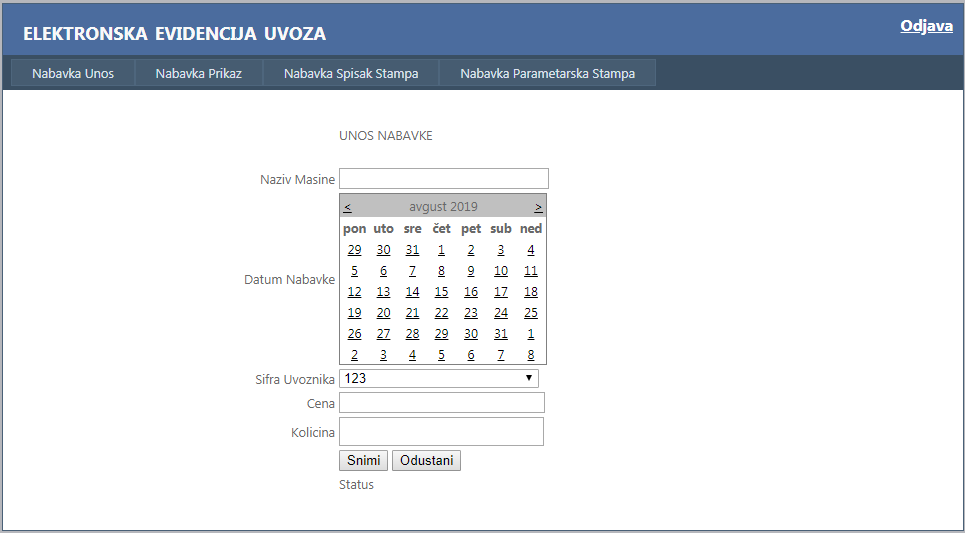
|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  //  using System.Data;  using KlasePodataka;  namespace PrezentacionaLogika  {  public class clsNabavkaStampaSpisak  {  #region  private string pStringKonekcije;  #endregion;  public clsNabavkaStampaSpisak(string NoviStringKonekcije)  {  pStringKonekcije = NoviStringKonekcije;  }  public DataSet DajPodatkeZaGrid(string Filter)  {  DataSet dsPodaci = new DataSet();  clsNabavkaDB objNabavkaDB = new clsNabavkaDB(pStringKonekcije);  if (Filter.Equals(""))  {  dsPodaci = objNabavkaDB.DajSveNabavke();  }  else  {  dsPodaci = objNabavkaDB.DajNabavkuPoNazivuMasine(Filter);  }  return dsPodaci;  }  }  } |

# 

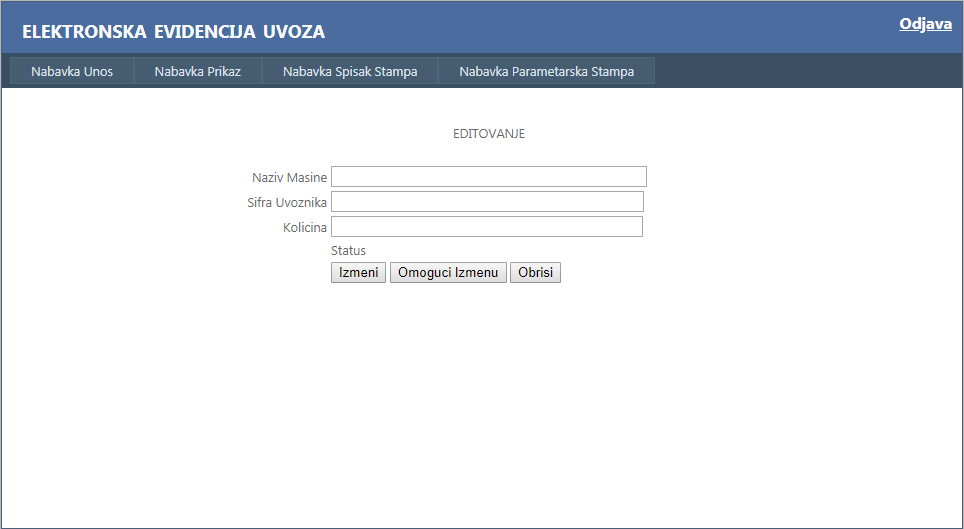
# Korisniči interfejs

  
Slika 1

Na Slici 1 prikazana LogIn forma koju je neohodno popuniti da bi se dobio pristup unosu podataka.

  
Slika 2

Na Slici 2 prikazana je stranica za unos podataka nove nabavke. Pored stranice za unos nabavke vidimo i još par stranica koje služe za prikazivanje dosad unetih narudžbina.

  
Slika 3

Na datoj Slici 3 prikazana je mogućnost izamene nabavke ukoliko je došlo do greške prilikom samog unosa. Takođe može se vršiti brisanje određene nardžbine ako se uaže potreba za tim. Da bi se omogućila izmena, prvo je potrebno omogućiti izmenu klikom na „Omogući Izmenu“ polje.

# Zakljucak

U ovom seminarskom radu prikazana je realizovana prototipska aplikacija za podršku radu evidencije uvoza

U delu specifikacije I dizajna obuhvaćene su sve najvažnije funkcije, a u delu implementacije su realizovane samo osnovne funkcije unosa, tabelarnog prikaza čime se ilustruje poznavanje osnovnih tehnika programiranja.

Dalji razvoj aplikacije odnosio bi se na realizaciju preostalih softverskih funkcija potrebnih za pokrivanje svih očekivanih funkcionalnih karakteristika softvera, kao I dodavanje automatizama, kontrole grešaka korisnika I slično.

# Literatura

Korišćena literature za izradu ovog seminarskog rada:

[1]Jesse Liberty: “Programiranje na jeziku C#”, Mikro knjiga, 2017.

[2]Standardni web vido sajt https://www.youtube.com/watch?v=GhQdlIFylQ8