A black and white logo with a person's face

Description automatically generatedA black background with white lines and white ears

Description automatically generated with medium confidenceUniverzitet u Novom Sadu

Tehnički fakultet »Mihajlo Pupin« Zrenjanin

SEMINARSKI RAD

Predmet: Razvoj softvera otvorenog koda

**Tema: Avio Kompanija**

Predmetni nastavnik: Autori rada:

Doc. dr Zoltan Kazi Vuk Đaković SI 3/20

MSc Velibor Premčevski Stefan Milošević SI 4/20

Zrenjanin 2023.

Sadržaj

[1. Uvod 3](#_Toc144669443)

[2. Specifikacija zahteva korisnika 4](#_Toc144669444)

[3. Faze razvoja softvera 5](#_Toc144669445)

[4. Prikaz softvera 6](#_Toc144669446)

[5. Prikaz realizacije i implementacije 11](#_Toc144669447)

[6. Korišteni alati i softveri 15](#_Toc144669448)

# Uvod

Seminarski rad iz predmeta Razvoj softvera otvorenog koda pod nazivom “Avio kompanija” je aplikacija koja omogućava zaposlenim u avio kompaniji jednostavno upravljanje bazom podataka koja sadrži detalje o klijentima, mogućim destinacijama I objektima, kao I podatke o dostupnim sedištima I avionskim klasama.

Osnovna funkcionalnost je kreiranje avionskih karata na osnovu podataka unetih u navedene tabele. Svi podaci iz tabela se mogu unositi, menjati i brisati po potrebi.

# Specifikacija zahteva korisnika

* Dodavanje, izmena i brisanje Klijenata
* Dodavanje, izmena i brisanje Destinacija
* Dodavanje, izmena i brisanje Objekata
* Dodavanje, izmena i brisanje Klasa
* Dodavanje, izmena i brisanje Karata
* Prikaz svih stavki u tabelama

# Faze razvoja softvera

Razvoj ove softverske aplikacije podrazumeva niz faza, uključujući kreiranje frontend dela aplikacije u tehnologiji WPF (Windows Presentation Foundation), razvoj backend komponenti u programskom jeziku C# i dizajniranje baze podataka u SQL-u.

Faze razvoja:

* Analiza i definisanje zahteva
* Odabir tehnologija
* Kreiranje repozitorijuma na GitHub-u
* Kreiranje baze podataka
* Dizajniranje korisničkog interfejsa
* Implementacija UI logike
* Povezivanje sa backend-om
* Definisanje klasa i metoda
* Povezivanje sa bazom podataka
* Kontinuirana integracija i testiranje oba dela aplikacije
* Isporuka gotovog softvera

# Prikaz softvera

A screenshot of a computer

Description automatically generated

*Slika 1. Početni prozor*

Na početnom prozoru se nalaze navigacioni elementi za preusmeravanje korisnika ka ostalim stranicama unutar aplikacije, gde se vrši unos podataka.

A screenshot of a computer

Description automatically generated

*Slika 2. Prozor za unos klijenata*

U prozoru sa tabelom klijenata se nalaze text box-ovi u koje se unose ime, prezime i kontakt telefon klijenta.

A screenshot of a computer

Description automatically generated

*Slika 3. Prozor za unos destinacija*

*A screenshot of a computer

Description automatically generated*

*Slika 4. Prozor za unos klase i sedišta*

U ovom prozoru se nalazi list box iz kojeg se biraju postojeće klase: Ekonomska, Biznis i Prva klasa. Ovde se takođe nalazi i text box gde se unosi broj sedišta.

A screenshot of a computer

Description automatically generated

*Slika 5. Prozor za unos objekta*

U prozoru sa objektima se nalaze text box-ovi za unos vrste objekta, naziv i kontakt, kao i combo box koji je povezan sa tabelom destinacija iz kojeg se bira grad u kojem se objekat nalazi.

A screenshot of a computer

Description automatically generated

*Slika 6. Prozor za kreiranje karte*

Karta se kreira pomoću combo i list-boxova iz kojih se biraju unete stavke iz prethodnih tabela i date picker-a i text box-a za cenu.

# Prikaz realizacije i implementacije

|  |
| --- |
| <connectionStrings>  <add name="BazaAvioKompanija" connectionString="Data Source=NOTEBOOK\SQLEXPRESS;Initial Catalog=BazaAvioKompanija;Integrated Security=True"/>  </connectionStrings> |

*Listing 1. Konekcioni string*

Baza podataka je kreirana lokalno u SQL Server Management Studio okruženju i sa aplikacijom je povezana uz pomoć konekcionog stringa koji je unet u app.config.

|  |
| --- |
| private void binDataGrid()  {  SqlConnection connection = new SqlConnection();  connection.ConnectionString = ConfigurationManager.ConnectionStrings["BazaAvioKompanija"].ConnectionString;  connection.Open();  SqlCommand command = new SqlCommand();  command.CommandText = "SELECT \* FROM [Karta] INNER JOIN [Objekat] ON [Karta].ID\_objekta = [Objekat].ID\_objekta INNER JOIN [Klijent] ON [Karta].ID\_klijenta = [Klijent].ID\_klijenta INNER JOIN [Klasa] ON [Karta].ID\_klase = [Klasa].ID\_klase INNER JOIN [Destinacija] ON [Objekat].ID\_destinacije = [Destinacija].ID\_destinacije";  command.Connection = connection;  SqlDataAdapter dataAdapter = new SqlDataAdapter(command);  DataTable dataTable = new DataTable("Karta");  dataAdapter.Fill(dataTable);  DataGrid.ItemsSource = dataTable.DefaultView;  } |

*Listing 2. Prikaz podataka iz baze u tabelu Karta*

|  |
| --- |
| private void Dodaj\_Click(object sender, RoutedEventArgs e)  {  SqlConnection connection = new SqlConnection();  connection.ConnectionString = ConfigurationManager.ConnectionStrings["BazaAvioKompanija"].ConnectionString;  connection.Open();  DateTime Datum = Convert.ToDateTime(dtDatum.Text);  string myDate = "30-12-1899 07:50:00:AM";  DateTime dt1 = DateTime.ParseExact(myDate, "dd-MM-yyyy hh:mm:ss:tt",  CultureInfo.InvariantCulture);  SqlCommand command = new SqlCommand();  command.CommandText = "INSERT INTO [Karta](Cena, Datum, ID\_objekta, ID\_klijenta, ID\_klase) VALUES(@Cena, @Datum, @ID\_objekta, @ID\_klijenta, @ID\_klase)";  command.Parameters.AddWithValue("@Cena", txtCena.Text);  command.Parameters.AddWithValue("@Datum", dtDatum.SelectedDate);  command.Parameters.AddWithValue("@ID\_objekta", lbxID\_objekta.SelectedValue);  command.Parameters.AddWithValue("@ID\_klijenta", cbxID\_klijenta.SelectedValue);  command.Parameters.AddWithValue("@ID\_klase", cbxID\_klase.SelectedValue);  command.Connection = connection;  int provera = command.ExecuteNonQuery();  if (provera == 1)  {  MessageBox.Show("Podaci su uspešno upisani");  binDataGrid();  }  ponistiUnosTxt();  } |

*Listing 3. Dugme za unos podataka u bazu*

Nakon što se text, combo i list box-ovi popune i kada se izabere datum, pritiskom na dugme „Dodaj” se podaci unose u bazu.

|  |
| --- |
| private void Izmeni\_Click(object sender, RoutedEventArgs e)  {  SqlConnection connection = new SqlConnection();  connection.ConnectionString = ConfigurationManager.ConnectionStrings["BazaAvioKompanija"].ConnectionString;  connection.Open();  SqlCommand command = new SqlCommand();  command.CommandText = "UPDATE [Karta] SET Cena = @Cena, Datum = @Datum, ID\_objekta = @ID\_objekta, ID\_klijenta = @ID\_klijenta, ID\_klase = @ID\_klase WHERE ID\_karte = @ID\_karte";  command.Parameters.AddWithValue("@ID\_karte", txtID\_karte.Text);  command.Parameters.AddWithValue("@Cena", txtCena.Text);  command.Parameters.AddWithValue("@Datum", dtDatum.SelectedDate);  command.Parameters.AddWithValue("@ID\_objekta", lbxID\_objekta.SelectedItem);  command.Parameters.AddWithValue("@ID\_klijenta", cbxID\_klijenta.SelectedValue);  command.Parameters.AddWithValue("@ID\_klase", cbxID\_klase.SelectedValue);  command.Connection = connection;  int provera = command.ExecuteNonQuery();  if (provera == 1)  {  MessageBox.Show("Podaci su uspešno promenjeni");  binDataGrid();  }  ponistiUnosTxt(); |

*Listing 4. Dugme za izmenu izabranih podataka*

|  |
| --- |
| private void Obrisi\_Click(object sender, RoutedEventArgs e)  {  {  SqlConnection connection = new SqlConnection();  connection.ConnectionString = ConfigurationManager.ConnectionStrings["BazaAvioKompanija"].ConnectionString;  connection.Open();  SqlCommand command = new SqlCommand();  command.CommandText = "DELETE FROM [Karta] WHERE ID\_karte = @ID\_karte";  command.Parameters.AddWithValue("@ID\_karte", txtID\_karte.Text);  command.Connection = connection;  int provera = command.ExecuteNonQuery();  if (provera == 1)  {  MessageBox.Show("Podaci su uspešno obrisani");  binDataGrid();  }  ponistiUnosTxt();  }  } |

*Listing 5. Dugme za brisanje izabranih podataka*

|  |
| --- |
| private void cbxID\_klijenta\_Loaded(object sender, RoutedEventArgs e)  {  SqlConnection connection = new SqlConnection();  connection.ConnectionString = ConfigurationManager.ConnectionStrings["BazaAvioKompanija"].ConnectionString;  connection.Open();  SqlCommand commandCbx = new SqlCommand();  commandCbx.CommandText = "SELECT \* FROM [Klijent] ORDER BY ID\_klijenta";  commandCbx.Connection = connection;  SqlDataAdapter dataAdapterCbx = new SqlDataAdapter(commandCbx);  DataTable dataTableCbx = new DataTable("Karta");  dataAdapterCbx.Fill(dataTableCbx);  for (int i = 0; i < dataTableCbx.Rows.Count; i++)  {  cbxID\_klijenta.Items.Add(dataTableCbx.Rows[i]["ID\_klijenta"]);  }  } |

*Listing 6. Kod za combo box*

Ovim combo box-om se omogućava odabir klijenata iż baze za unos u tabelu Karta.

|  |
| --- |
| private void lbxID\_objekta\_Loaded(object sender, RoutedEventArgs e)  {  SqlConnection connection = new SqlConnection();  connection.ConnectionString = ConfigurationManager.ConnectionStrings["BazaAvioKompanija"].ConnectionString;  connection.Open();  SqlCommand commandLbx = new SqlCommand();  commandLbx.CommandText = "SELECT \* FROM [Objekat] ORDER BY ID\_objekta";  commandLbx.Connection = connection;  SqlDataAdapter dataAdapterLbx = new SqlDataAdapter(commandLbx);  DataTable dataTableLbx = new DataTable("Karta");  dataAdapterLbx.Fill(dataTableLbx);  for (int i = 0; i < dataTableLbx.Rows.Count; i++)  {  lbxID\_objekta.Items.Add(dataTableLbx.Rows[i]["ID\_objekta"]);  }  } |

*Listing 7. Kod za list box*

# Korišteni alati i softveri

* Microsoft Visual Studio 2022
  + WPF (Windows Presentation Foundation)
  + C#
* SQL Server Management Studio 19