|  |  |  |
| --- | --- | --- |
| A video game screen with white ears and blue lines  AI-generated content may be incorrect. | Univerzitet u Novom Sadu  Tehnički fakultet „Mihajlo Pupin”  Zrenjanin | A black and white logo with a person's face  AI-generated content may be incorrect. |

Predmet: Razvoj softvera otvorenog koda

SEMINARSKI RAD

**Avio karte**

Predmetni nastavnik:

Prof. dr Zoltan Kazi

Asistent: Studenti:

MSc Velibor Premčevski Aleksandar Boškov SI 25/21

Ivan Jerkov SI 26/21

Marko Golušin SI 27/21

Dušan Milankov SI 12/19

Zrenjanin, 2025

* **Uvod**

U ovom radu biće opisan informacioni sistem za upravljanje avio kartama, koji uključuje administraciju avio karata I mogućnost korisnika da rezerviše i pregleda svoju kartu.

Projekat će biti realizovan pomoću razvojnog okruženja Visual Studio, programskog jezika C# za razvoj funkcionalnosti softverskog rešenja, takođe će biti korišteni SQL Server Management Studio i SQL u svrhu razvoja baze podataka. Pored navedenih tehnologija biće korišten i GitHub kao CMS sistem u cilju olakšanog kolaborativnog rada.

* **Specifikacija zahteva korisnika**

Svrha softverskog rešenja za avio karte jeste automatizacija i digitalizacija ključnih procesa koji su se do sada obavljali ručno ili u tabelama. To uključuje:

* Evidentiranje i upravljanje rezervacijama i prodajom karata,
* Administracija I upravljanje kartama,
* Onemogućavanje preklapanja karata i praćenje zauzetih karata.

Definisanje pojmova:

* Administrator – Osoba koja ima pristup svim funkcionalnostima sistema i upravlja čitavim sistemom: unosom, izmenom i brisanjem karata, kao i informacijama o korisnicima.
* Korisnik - Osoba koja putem sistema vrši rezervaciju karata za avion i ima uvid u informacije o letu.

Opis rada sistema – funkcionalni zahtevi:

* Korisnik može da rezerviše kartu klikom na slobodne karte I ubacivanje svojih informacija I zatim klikom na dugme Rezerviši u prozoru Rezervacija.
* Administrator može da se uloguje u sistem klikom na dugme Admin I unosom korisničkog imena I lozinke za admina.
* Rezervisane karte korisnik može da pregleda klikom na dugme Pregled karti I unošenje podataka o svojoj karti.
* Otkazivanje rezervacije od strane administratora se vrši klikom na dugme Poništi rezervaciju u prozoru Pregled karata.
* Administrator može da vrši izmenu I dodavanje novih karata u posebnim prozorima (izmena u prozoru Izmena karata i dodavanje u Unos karata).
* Tabelarni prikaz karata (za administratora).
* **Baza podataka**

Za bazu podataka korišten je SQL Server Management Studio i jezik SQL.

Potrebne tabele unutar baze su:

* Korisnik,
* Karta,
* Admini.

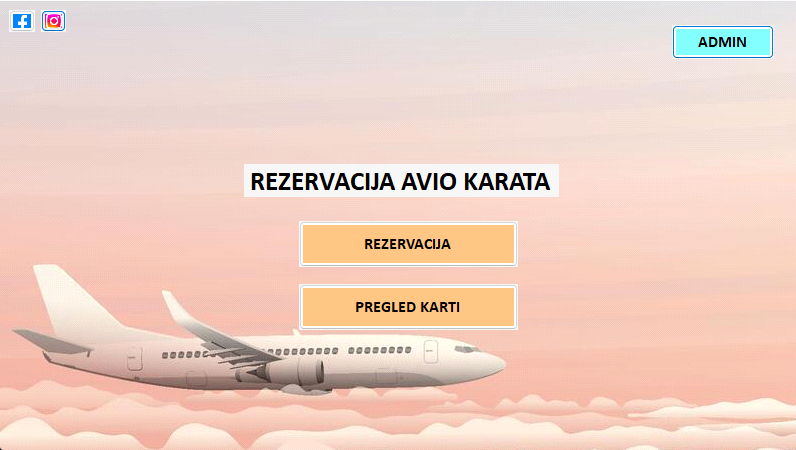
SQL skripta za kreiranje baze podataka:

|  |
| --- |
| create database AvioKarte  create table korisnik(  idKorisnika int identity(1,1) not null primary key,  imeKorisnika varchar(20) not null,  prezimeKorisnika varchar(20) not null,  emailKorisnika varchar(50) not null,  telefonKorisnika int not null,  lozinkaKorisnika varchar(20) not null  );  create table karta(  idKarte int not null primary key,  imePutnika varchar(20) not null,  prezimePutnika varchar(20) not null,  poletanje varchar(50) not null,  sletanje varchar(50) not null,  klasa varchar(20) not null,  vremePoletanja varchar(10) not null,  datumPoletanja varchar(12) not null,  sediste varchar(5) not null,  brojLeta varchar(30) not null,  kapija varchar(5) not null  );  create table admini(  idAdmina int not null primary key,  korisnickoImeAdmina varchar(20) not null,  lozinkaAdmina varchar(20) not null  );  alter table karta add foreign key (idKarte) references korisnik(idKorisnika);  insert into admini values (1,'admin','admin'); |

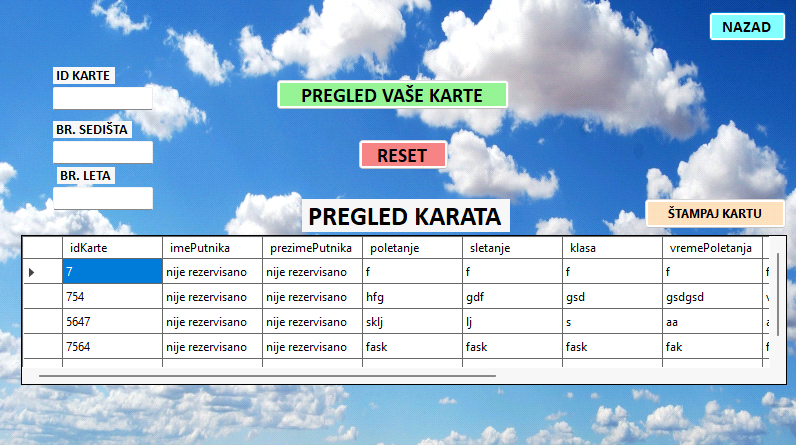
* **Korisnički interfejs**

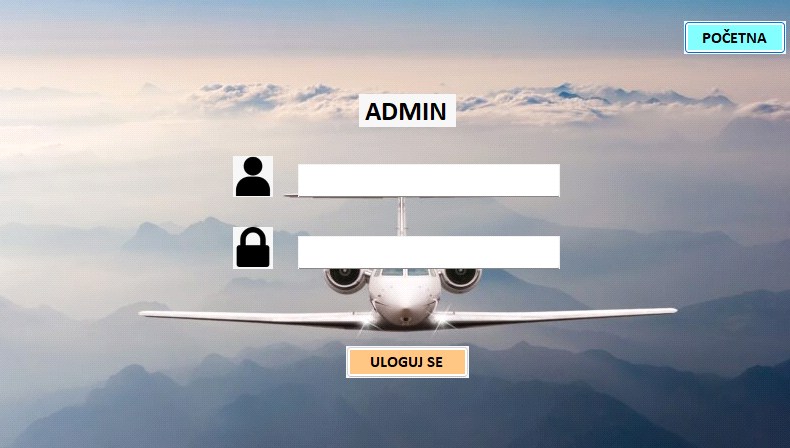
Na osnovu analize korisničkih zahteva, korisnički interfejs (za gledaoca) treba da obezbedi:

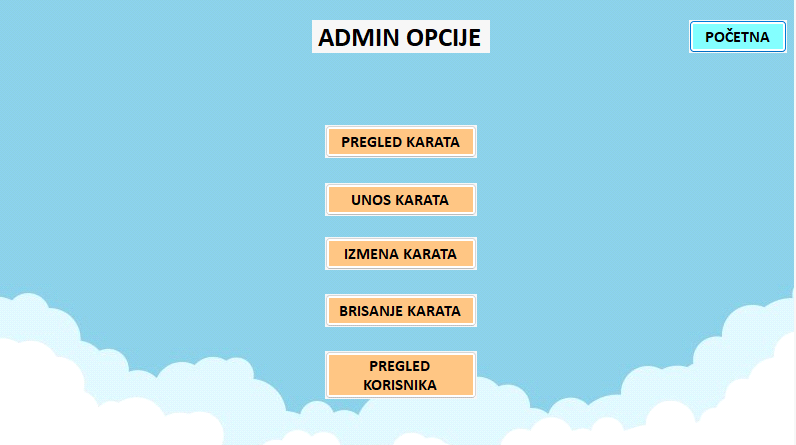
* Početnu stranicu koja poseduje navigacionu dugmad koja vodi do stranice za registraciju karte, za pregled rezervisane karte, za ulogovanje administratora i link za društvene mreže. (Slika 1).
* Stranicu za registraciju koja omogućava korisniku da izvrši registraciju i rezervaciju avionske karte iz baze dostupnih (nerezervisanih) karata, unosom ličnih podataka, kao i mogućnost ažuriranja izabrane karte sa imenom i prezimenom putnika, uz dodatne funkcionalnosti za filtriranje dostupnih karata po različitim kriterijumima (klasa, poletanje, sletanje, datum). (Slika 2).
* Stranicu koja omogućava korisnicima da, unosom identifikacionog broja karte, broja leta i sedišta, izvrše pretragu podataka o sopstvenim avionskim kartama, pregledaju rezultate u tabelarnom prikazu, i po potrebi generišu PDF dokument sa detaljnim informacijama o izabranoj karti. (Slika 3).
* Stranicu koja omogućava prijavu administratora putem unosa korisničkog imena i lozinke, validira unete podatke proverom u bazi podataka, i u slučaju uspešne autentifikacije otvara administratorski deo aplikacije. (Slika 4).
* Stranicu koja je glavni administratorski meni iz kojeg administrator može pristupiti različitim funkcionalnostima aplikacije, uključujući pregled korisnika, upravljanje letovima i kartama, statistiku i druge administrativne stranice, putem navigacije ka odgovarajućim formama. (Slika 5).
* Stranicu koja služi za pregled i upravljanje avionskim kartama u bazi podataka, omogućavajući administratoru da vidi sve karte, filtrira samo one koje su rezervisane, kao i da poništi rezervaciju odabirom karte, čime se karta ponovo stavlja na raspolaganje. (Slika 6).
* Stranicu koja omogućava unos novih avionskih karata u bazu podataka sa svim relevantnim detaljima leta i sedišta, čime se dodaju dostupne karte za rezervaciju. (Slika 7).
* Stranicu koja omogućava pregled, izbor i izmenu podataka o avionskim kartama iz baze, prikazujući informacije u tabeli i omogućavajući administratoru da ih ažurira putem unetih podataka u odgovarajuća polja. (Slika 8).
* Stranicu koja omogućava brisanje odabrane karte iz baze podataka selektovanjem i klikom na avionsku kartu u tabeli, uz osvežavanje prikaza nakon izmene. (Slika 9).
* Stranicu koja prikazuje tabelu svih korisnika sistema, učitavajući podatke iz baze, i omogućava povratak na prethodni ekran. (Slika 10).

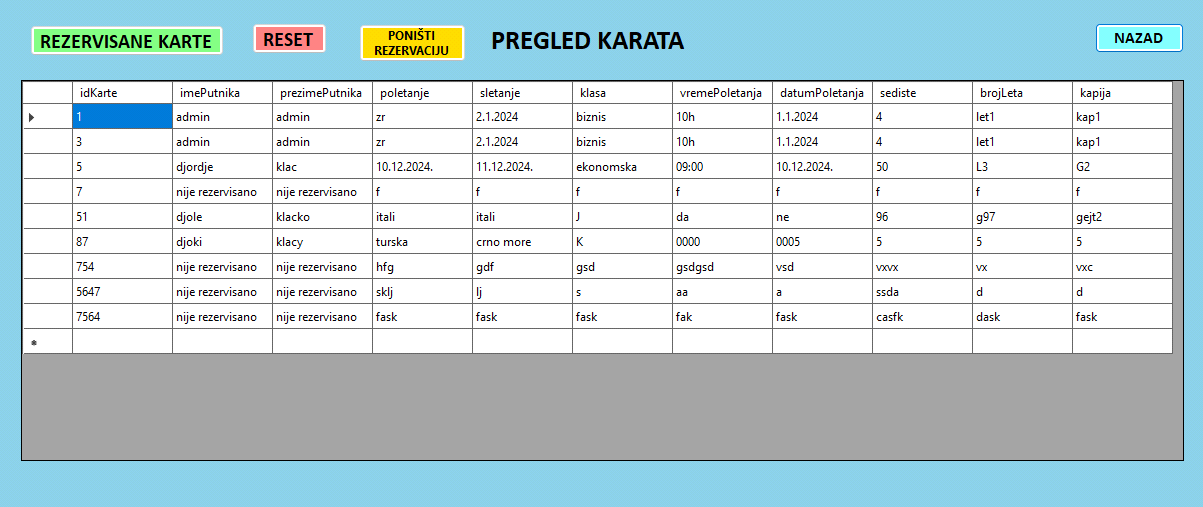
Slika 1. – Prikaz početne stranice.

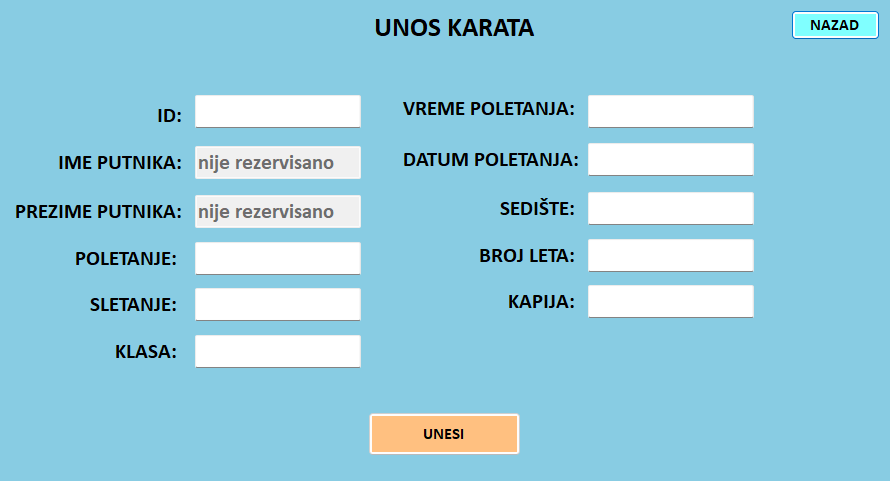
Slika 2. – Prikaz stranice za rezervisanje karata.

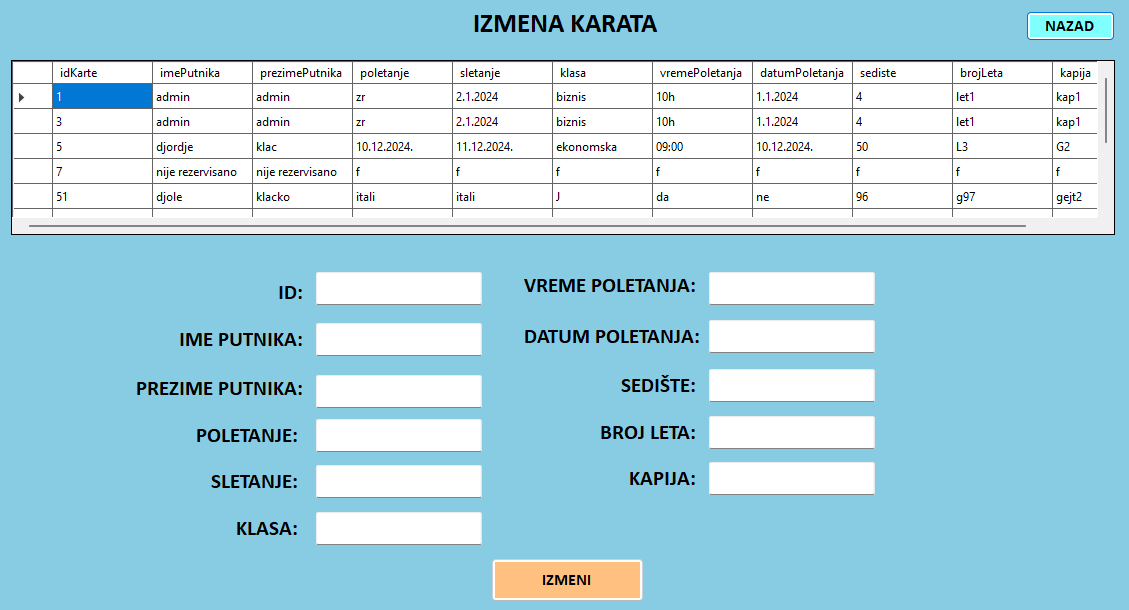
Slika 3. – Prikaz stranice za pregled karata.

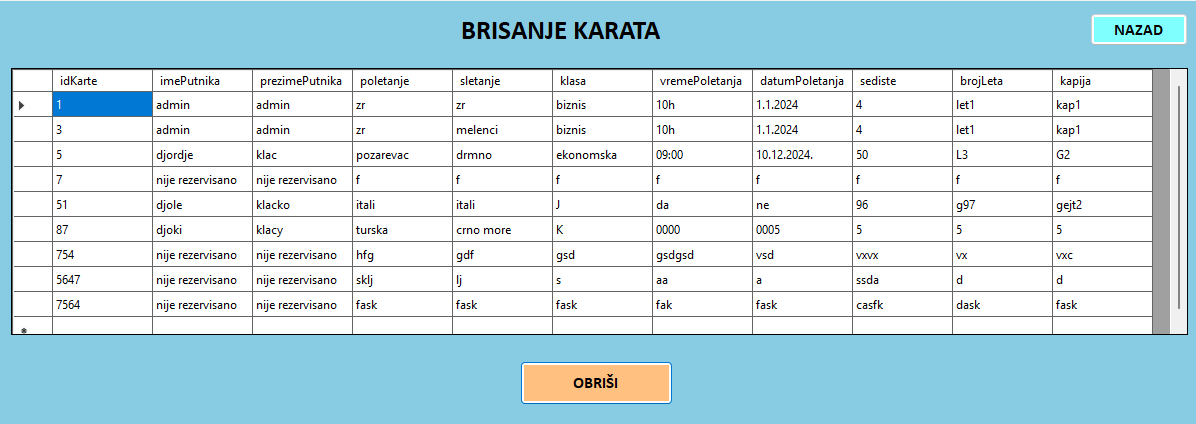
Slika 4. – Prikaz stranice za logovanje administratora.

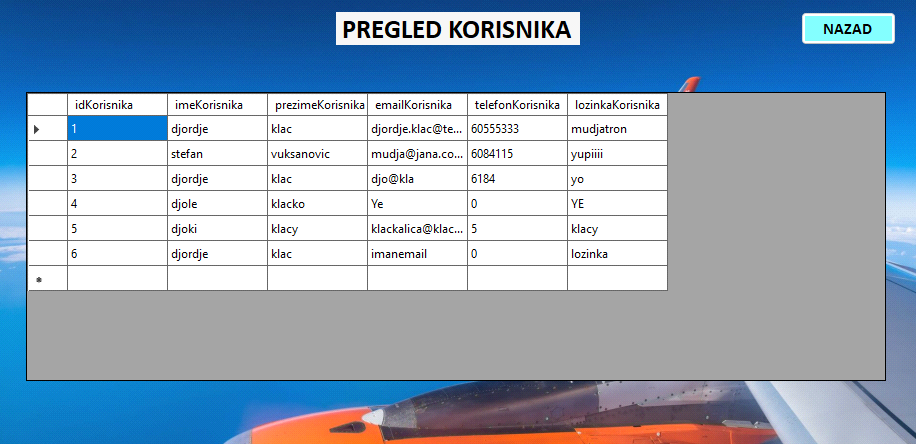
Slika 5. – Prikaz stranice za admin opcije.

Slika 6. – Prikaz stranice za pregled karata (za admina).

Slika 7. – Prikaz stranice za unos karata.

Slika 8. – Prikaz stranice za izmenu karata.

Slika 9. – Prikaz stranice za brisanje karata.

Slika 10. – Prikaz stranice za pregled korisnika.

* **Ključni delovi koda**

U cilju razvoja funkcionalnosti ovog softverskog rešenja korišten je programski jezik C#.

U daljem tekstu će biti navedeni listinzi nekih ključnih delova koda sa kratkim opisima.

Jedan od ključnih delova koda jeste unos karata (Listing 1). U ovom delu se nalazi logika koja unosi kartu u bazu.

|  |
| --- |
| 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;  using System.Data.SqlClient;  using static System.Runtime.InteropServices.JavaScript.JSType;  namespace SeminarskiRSOK  {  public partial class Form8 : Form  {  public Form8()  {  InitializeComponent();  }  private void label1\_Click(object sender, EventArgs e)  {  }  private void label2\_Click(object sender, EventArgs e)  {  }  private void button1\_Click(object sender, EventArgs e)  {  SqlConnection con = new SqlConnection("Data Source = DESKTOP-1VNGR39; Initial Catalog = AvioKarte; Integrated Security = True;");  con.Open();  SqlCommand cmd = new SqlCommand("insert into karta values (@idKarte, @imePutnika, @prezimePutnika, @poletanje, @sletanje, @klasa, @vremePoletanja, @datumPoletanja, @sediste, @brojLeta, @kapija)", con);  cmd.Parameters.AddWithValue("@idKarte", int.Parse(textBox1.Text));  cmd.Parameters.AddWithValue("@imePutnika", textBox2.Text);  cmd.Parameters.AddWithValue("@prezimePutnika", textBox3.Text);  cmd.Parameters.AddWithValue("@poletanje", textBox4.Text);  cmd.Parameters.AddWithValue("@sletanje", textBox5.Text);  cmd.Parameters.AddWithValue("@klasa", textBox6.Text);  cmd.Parameters.AddWithValue("@vremePoletanja", textBox7.Text);  cmd.Parameters.AddWithValue("@datumPoletanja", textBox8.Text);  cmd.Parameters.AddWithValue("@sediste", textBox9.Text);  cmd.Parameters.AddWithValue("@brojLeta", textBox10.Text);  cmd.Parameters.AddWithValue("@kapija", textBox11.Text);  cmd.ExecuteNonQuery();  con.Close();  MessageBox.Show("Podaci su uspešno uneti!");  }  private void button2\_Click(object sender, EventArgs e)  {  Form3 f3 = new Form3();  f3.Show();  this.Hide();  }  private void Form8\_Load(object sender, EventArgs e)  {  }  private void label7\_Click(object sender, EventArgs e)  {  }  }  } |

Listing 1. – listing koda za unos karata u bazu.

Listing 2 prikazuje deo koda čija je svrha brisanje karata .

|  |
| --- |
| 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;  using System.Data.SqlClient;  namespace SeminarskiRSOK  {  public partial class Form5 : Form  {  public Form5()  {  InitializeComponent();  BinGrid();  }  private void binDataGrid()  {  string constring = "Data Source=DESKTOP-1VNGR39;Initial Catalog=AvioKarte;Integrated Security=True;Trust Server Certificate=True";  using (SqlConnection con = new SqlConnection(constring))  {  }  }  private void Form1\_Load(object sender, EventArgs e)  {  this.BinGrid();  }  private void BinGrid()  {  string constring = @"Data Source=DESKTOP-1VNGR39;Initial Catalog=AvioKarte;Integrated Security=True;";  using (SqlConnection con = new SqlConnection(constring))  {  using (SqlCommand cmd = new SqlCommand("Select \* FROM karta", con))  {  using (SqlDataAdapter sda = new SqlDataAdapter(cmd))  {  using (DataTable dt = new DataTable())  {  sda.Fill(dt);  dataGridView1.DataSource = dt;  dataGridView1.DefaultCellStyle.ForeColor = Color.Black;  dataGridView1.DefaultCellStyle.BackColor = Color.White;  }  }  }  }  }  private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)  {  }  private void button2\_Click(object sender, EventArgs e)  {  Form3 form3 = new Form3();  form3.Show();  this.Hide();  }  private void button1\_Click(object sender, EventArgs e)  {  try  {  if (dataGridView1.CurrentRow != null)  {  int idKarte = (int)dataGridView1.CurrentRow.Cells[0].Value;  string constring = @"Data Source=DESKTOP-1VNGR39;Initial Catalog=AvioKarte;Integrated Security=True;";  using (SqlConnection con = new SqlConnection(constring))  {  con.Open();  using (SqlCommand cmd = new SqlCommand("DELETE FROM karta WHERE idKarte=@idKarte", con))  {  cmd.Parameters.AddWithValue("@idKarte", idKarte);  int rowsAffected = cmd.ExecuteNonQuery();  if (rowsAffected > 0)  {  MessageBox.Show("Obrisano");  }  else  {  MessageBox.Show("Nije pronadjen id za brisanje");  }  }  }  ResetTabele();  }  else  {  MessageBox.Show("Izaberite red za brisanje");  }  }  catch (Exception ex)  {  MessageBox.Show("Greska: " + ex.Message);  }  }  private void ResetTabele()  {  try  {  string constring = @"Data Source=DESKTOP-1VNGR39;Initial Catalog=AvioKarte;Integrated Security=True;";  using (SqlConnection con = new SqlConnection(constring))  {  con.Open();  string selectQuery = "SELECT \* FROM karta";  using (SqlCommand cmd = new SqlCommand(selectQuery, con))  {  DataTable dt = new DataTable();  using (SqlDataAdapter da = new SqlDataAdapter(cmd))  {  da.Fill(dt);  }  dataGridView1.DataSource = dt;  }  }  }  catch (Exception ex)  {  MessageBox.Show("Greska pri prikazivanju nove tabele " + ex.Message);  }  }  private void button2\_Click\_1(object sender, EventArgs e)  {  Form3 f3 = new Form3();  f3.Show();  this.Hide();  }  private void Form5\_Load(object sender, EventArgs e)  {  }  private void label6\_Click(object sender, EventArgs e)  {  }  }  } |

Listing 2. – listing koda za brisanje karata.

Listing 3 nam omogućuje da rezervišemo karte (za korisnika) I takođe da pretražimo karte pomoću 4 različita filtera. Unošenjem podataka u textbox-ove za filter nam ispisuje podatke ukoliko su ti podaci uneti u bazu podataka, ukoliko nisu, tabela će biti prazna.

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Data.SqlClient;  using System.Drawing;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Windows.Forms;  namespace SeminarskiRSOK  {  public partial class Form6 : Form  {  public Form6()  {  InitializeComponent();  BinGrid();  }  private void binDataGrid()  {  string constring = "Data Source=DESKTOP-1VNGR39;Initial Catalog=AvioKarte;Integrated Security=True;";  using (SqlConnection con = new SqlConnection(constring))  {  }  }  private void Form1\_Load(object sender, EventArgs e)  {  this.BinGrid();  }  private void BinGrid()  {  string constring = @"Data Source=DESKTOP-1VNGR39;Initial Catalog=AvioKarte;Integrated Security=True;";  using (SqlConnection con = new SqlConnection(constring))  {  using (SqlCommand cmd = new SqlCommand("SELECT \* FROM karta WHERE imePutnika = 'nije rezervisano' AND prezimePutnika = 'nije rezervisano'", con))  {  using (SqlDataAdapter sda = new SqlDataAdapter(cmd))  {  using (DataTable dt = new DataTable())  {  sda.Fill(dt);  dataGridView1.DataSource = dt;  }  }  }  }  }  private void textBox1\_TextChanged(object sender, EventArgs e)  {  }  private void button2\_Click(object sender, EventArgs e)  {  Form1 form1 = new Form1();  form1.Show();  Visible = false;  }  private void Form6\_Load(object sender, EventArgs e)  {  }  private void button1\_Click(object sender, EventArgs e)  {  try  {  int idKarte = (int)dataGridView1.CurrentRow.Cells[0].Value;  SqlConnection con = new SqlConnection("Data Source = DESKTOP-1VNGR39; Initial Catalog = AvioKarte; Integrated Security = True;");  con.Open();  SqlCommand cmd = new SqlCommand("insert into korisnik values (@imeKorisnika, @prezimeKorisnika, @emailKorisnika, @telefonKorisnika, @lozinkaKorisnika) UPDATE karta SET imePutnika=@imeKorisnika, prezimePutnika=@prezimeKorisnika WHERE idKarte=@idKarte", con);  cmd.Parameters.AddWithValue("@imeKorisnika", textBox1.Text);  cmd.Parameters.AddWithValue("@prezimeKorisnika", textBox2.Text);  cmd.Parameters.AddWithValue("@telefonKorisnika", int.Parse(textBox3.Text));  cmd.Parameters.AddWithValue("@emailKorisnika", textBox4.Text);  cmd.Parameters.AddWithValue("@lozinkaKorisnika", textBox5.Text);  cmd.Parameters.AddWithValue("@idKarte", idKarte);  cmd.ExecuteNonQuery();  con.Close();  MessageBox.Show("Podaci su uspešno uneti!");  ResetTabele();  }  catch (Exception ex)  {  MessageBox.Show("Niste popunili sva polja!");  }  }  private void ResetTabele()  {  try  {  string constring = @"Data Source=DESKTOP-1VNGR39;Initial Catalog=AvioKarte;Integrated Security=True;";  using (SqlConnection con = new SqlConnection(constring))  {  con.Open();  string selectQuery = "SELECT \* FROM karta WHERE imePutnika = 'nije rezervisano' AND prezimePutnika = 'nije rezervisano'";  using (SqlCommand cmd = new SqlCommand(selectQuery, con))  {  DataTable dt = new DataTable();  using (SqlDataAdapter da = new SqlDataAdapter(cmd))  {  da.Fill(dt);  }  dataGridView1.DataSource = dt;  }  }  }  catch (Exception ex)  {  MessageBox.Show("Greska pri prikazivanju nove tabele " + ex.Message);  }  }  private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)  {  }  private void label7\_Click(object sender, EventArgs e)  {  }  private void label7\_Click\_1(object sender, EventArgs e)  {  }  private void button3\_Click\_1(object sender, EventArgs e)  {  Form11 form11 = new Form11();  form11.Show();  Visible = false;  }  private void ApplyFilter()  {  string constring = @"Data Source=DESKTOP-1VNGR39;Initial Catalog=AvioKarte;Integrated Security=True;";  using (SqlConnection con = new SqlConnection(constring))  {  StringBuilder query = new StringBuilder("SELECT \* FROM karta WHERE imePutnika = 'nije rezervisano' AND prezimePutnika = 'nije rezervisano'");  if (!string.IsNullOrWhiteSpace(txtKlasa.Text))  query.Append(" AND klasa LIKE @klasa");  if (!string.IsNullOrWhiteSpace(txtPoletanje.Text))  query.Append(" AND poletanje LIKE @poletanje");  if (!string.IsNullOrWhiteSpace(txtSletanje.Text))  query.Append(" AND sletanje LIKE @sletanje");  if (!string.IsNullOrWhiteSpace(txtdatumP.Text))  query.Append(" AND datumPoletanja LIKE @datumPoletanja");  using (SqlCommand cmd = new SqlCommand(query.ToString(), con))  {  if (!string.IsNullOrWhiteSpace(txtKlasa.Text))  cmd.Parameters.AddWithValue("@klasa", "%" + txtKlasa.Text + "%");  if (!string.IsNullOrWhiteSpace(txtPoletanje.Text))  cmd.Parameters.AddWithValue("@poletanje", "%" + txtPoletanje.Text + "%");  if (!string.IsNullOrWhiteSpace(txtSletanje.Text))  cmd.Parameters.AddWithValue("@sletanje", "%" + txtSletanje.Text + "%");  if (!string.IsNullOrWhiteSpace(txtdatumP.Text))  cmd.Parameters.AddWithValue("@datumPoletanja", "%" + txtdatumP.Text + "%");  using (SqlDataAdapter sda = new SqlDataAdapter(cmd))  {  DataTable dt = new DataTable();  sda.Fill(dt);  dataGridView1.DataSource = dt;  }  }  }  }  private void btnFilter\_Click(object sender, EventArgs e)  {  ApplyFilter();  }  private void textBox8\_TextChanged(object sender, EventArgs e)  {  }  private void textBox7\_TextChanged(object sender, EventArgs e)  {  }  private void label10\_Click(object sender, EventArgs e)  {  }  private void label9\_Click(object sender, EventArgs e)  {  }  private void textBox6\_TextChanged(object sender, EventArgs e)  {  }  }  } |

Listing 3. – Prikaz koda za rezervaciju karata i odabir karata pomoću filtera.

Listing 4 nam omogućuje izmenu karata tako što u tabeli gde su nam ispisane karte kliknemo na jednu I popunjavaju se sva textbox polja, nakon čega možemo da biramo šta želimo da izmenimo za tu određenu kartu. Nakon ismene klikom na dugme određeno za izmenu čuvamo izmenjene podatke I baza podataka se ažurira.

|  |
| --- |
| 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;  using System.Data.SqlClient;  using static System.Runtime.InteropServices.JavaScript.JSType;  namespace SeminarskiRSOK  {  public partial class Form9 : Form  {  public Form9()  {  InitializeComponent();  BinGrid();  }  private void binDataGrid()  {  string constring = "Data Source=DESKTOP-1VNGR39;Initial Catalog=AvioKarte;Integrated Security=True;";  using (SqlConnection con = new SqlConnection(constring))  {  }  }  private void Form1\_Load(object sender, EventArgs e)  {  this.BinGrid();  }  private void BinGrid()  {  string constring = @"Data Source=DESKTOP-1VNGR39;Initial Catalog=AvioKarte;Integrated Security=True;";  using (SqlConnection con = new SqlConnection(constring))  {  using (SqlCommand cmd = new SqlCommand("Select \* FROM karta", con))  {  using (SqlDataAdapter sda = new SqlDataAdapter(cmd))  {  using (DataTable dt = new DataTable())  {  sda.Fill(dt);  dataGridView1.DataSource = dt;  }  }  }  }  }  private void dataGridView1\_CellClick(object sender, DataGridViewCellEventArgs e)  {  if (e.RowIndex >= 0)  {  DataGridViewRow row = dataGridView1.Rows[e.RowIndex];  textBox1.Text = row.Cells["idKarte"].Value.ToString();  textBox2.Text = row.Cells["imePutnika"].Value.ToString();  textBox3.Text = row.Cells["prezimePutnika"].Value.ToString();  textBox4.Text = row.Cells["poletanje"].Value.ToString();  textBox5.Text = row.Cells["sletanje"].Value.ToString();  textBox6.Text = row.Cells["klasa"].Value.ToString();  textBox7.Text = row.Cells["vremePoletanja"].Value.ToString();  textBox8.Text = row.Cells["datumPoletanja"].Value.ToString();  textBox9.Text = row.Cells["sediste"].Value.ToString();  textBox10.Text = row.Cells["brojLeta"].Value.ToString();  textBox11.Text = row.Cells["kapija"].Value.ToString();  }  }  private void button2\_Click(object sender, EventArgs e)  {  Form3 f3 = new Form3();  f3.Show();  this.Hide();  }  private void Form9\_Load(object sender, EventArgs e)  {  dataGridView1.CellClick += dataGridView1\_CellClick;  }  private void button1\_Click(object sender, EventArgs e)  {  SqlConnection con = new SqlConnection("Data Source = DESKTOP-1VNGR39; Initial Catalog = AvioKarte; Integrated Security = True;");  con.Open();  SqlCommand cmd = new SqlCommand("update karta set imePutnika=@imePutnika, prezimePutnika=@prezimePutnika, poletanje=@poletanje, sletanje=@sletanje, klasa=@klasa, vremePoletanja=@vremePoletanja, datumPoletanja=@datumPoletanja, sediste=@sediste, brojLeta=@brojLeta, kapija=@kapija where idKarte=@idKarte", con);  cmd.Parameters.AddWithValue("@idKarte", int.Parse(textBox1.Text));  cmd.Parameters.AddWithValue("@imePutnika", textBox2.Text);  cmd.Parameters.AddWithValue("@prezimePutnika", textBox3.Text);  cmd.Parameters.AddWithValue("@poletanje", textBox4.Text);  cmd.Parameters.AddWithValue("@sletanje", textBox5.Text);  cmd.Parameters.AddWithValue("@klasa", textBox6.Text);  cmd.Parameters.AddWithValue("@vremePoletanja", textBox7.Text);  cmd.Parameters.AddWithValue("@datumPoletanja", textBox8.Text);  cmd.Parameters.AddWithValue("@sediste", textBox9.Text);  cmd.Parameters.AddWithValue("@brojLeta", textBox10.Text);  cmd.Parameters.AddWithValue("@kapija", textBox11.Text);  cmd.ExecuteNonQuery();  con.Close();  MessageBox.Show("Podaci su uspešno izmenjeni");  BinGrid();  }  }  } |

Listing 4. – Prikaz koda za izmenu karata

Listing 5 nam omogućuje pregled korisničke karte, gde mora da unese specifičan ID karte, broj sedišta I broj leta kako bi se ta karta ispisala u tabeli. Takođe korisnik ima mogućnost štampanja svoje karte u vidu PDF formata tako što klikne na prazno polje u tabeli pored ID-a karte I klikne na dugme za štampanje karte nakon čega mu se PDF fajl čuva na računar.

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Data.SqlClient;  using System.Drawing;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Windows.Forms;  using iTextSharp.text;  using iTextSharp.text.pdf;  using System.IO;  namespace SeminarskiRSOK  {  public partial class Form11 : Form  {  public Form11()  {  InitializeComponent();  BinGrid();  }  private void BinGrid()  {  string constring = @"Data Source=DESKTOP-1VNGR39;Initial Catalog=AvioKarte;Integrated Security=True;";  using (SqlConnection con = new SqlConnection(constring))  {  using (SqlCommand cmd = new SqlCommand("SELECT \* FROM karta WHERE imePutnika = 'nije rezervisano' AND prezimePutnika = 'nije rezervisano'", con))  {  using (SqlDataAdapter sda = new SqlDataAdapter(cmd))  {  using (DataTable dt = new DataTable())  {  sda.Fill(dt);  dataGridView1.DataSource = dt;  }  }  }  }  }  private void ResetTabele()  {  try  {  string constring = @"Data Source=DESKTOP-1VNGR39;Initial Catalog=AvioKarte;Integrated Security=True;";  using (SqlConnection con = new SqlConnection(constring))  {  con.Open();  string selectQuery = "SELECT \* FROM karta WHERE imePutnika = 'nije rezervisano' AND prezimePutnika = 'nije rezervisano'";  using (SqlCommand cmd = new SqlCommand(selectQuery, con))  {  DataTable dt = new DataTable();  using (SqlDataAdapter da = new SqlDataAdapter(cmd))  {  da.Fill(dt);  }  dataGridView1.DataSource = dt;  }  }  }  catch (Exception ex)  {  MessageBox.Show("Greska pri prikazivanju nove tabele " + ex.Message);  }  }  private void button3\_Click\_1(object sender, EventArgs e)  {  SqlConnection con = new SqlConnection("Data Source = DESKTOP-1VNGR39; Initial Catalog = AvioKarte; Integrated Security = True;");  con.Open();  SqlCommand cmd = new SqlCommand("select \* from karta where sediste=@sediste AND brojLeta=@brojLeta AND idKarte=@idKarte", con);  cmd.Parameters.AddWithValue("@idKarte", textBox8.Text);  cmd.Parameters.AddWithValue("@sediste", textBox7.Text);  cmd.Parameters.AddWithValue("@brojLeta", textBox6.Text);  using (SqlDataAdapter sda = new SqlDataAdapter(cmd))  {  using (DataTable dt = new DataTable())  {  sda.Fill(dt);  dataGridView1.DataSource = dt;  }  }  }  private void button1\_Click(object sender, EventArgs e)  {  Form1 form1 = new Form1();  form1.Show();  this.Hide();  }  private void button4\_Click\_1(object sender, EventArgs e)  {  ResetTabele();  }  private void buttonPrintPdf\_Click(object sender, EventArgs e)  {  if (dataGridView1.SelectedRows.Count == 0)  {  MessageBox.Show("Molimo izaberite kartu iz tabele!");  return;  }  DataGridViewRow row = dataGridView1.SelectedRows[0];  string idKarte = row.Cells["idKarte"].Value.ToString();  string ime = row.Cells["imePutnika"].Value.ToString();  string prezime = row.Cells["prezimePutnika"].Value.ToString();  string poletanje = row.Cells["poletanje"].Value.ToString();  string sletanje = row.Cells["sletanje"].Value.ToString();  string klasa = row.Cells["klasa"].Value.ToString();  string vreme = row.Cells["vremePoletanja"].Value.ToString();  string datum = row.Cells["datumPoletanja"].Value.ToString();  string sediste = row.Cells["sediste"].Value.ToString();  string brojLeta = row.Cells["brojLeta"].Value.ToString();  string kapija = row.Cells["kapija"].Value.ToString();  string folderPath = Environment.GetFolderPath(Environment.SpecialFolder.Desktop);  string filePath = Path.Combine(folderPath, $"Karta\_{idKarte}.pdf");  try  {  Document doc = new Document(PageSize.A4);  PdfWriter.GetInstance(doc, new FileStream(filePath, FileMode.Create));  doc.Open();  Paragraph title = new Paragraph("Avio Karta", FontFactory.GetFont(FontFactory.HELVETICA\_BOLD, 20));  title.Alignment = Element.ALIGN\_CENTER;  doc.Add(title);  doc.Add(new Paragraph("\n"));  doc.Add(new Paragraph($"ID karte: {idKarte}"));  doc.Add(new Paragraph($"Putnik: {ime} {prezime}"));  doc.Add(new Paragraph($"Polazak: {poletanje}"));  doc.Add(new Paragraph($"Sletanje: {sletanje}"));  doc.Add(new Paragraph($"Klasa: {klasa}"));  doc.Add(new Paragraph($"Datum: {datum}"));  doc.Add(new Paragraph($"Vreme: {vreme}"));  doc.Add(new Paragraph($"Sedište: {sediste}"));  doc.Add(new Paragraph($"Broj leta: {brojLeta}"));  doc.Add(new Paragraph($"Kapija: {kapija}"));  doc.Close();  MessageBox.Show($"PDF karta je uspešno sačuvana na: {filePath}");  System.Diagnostics.Process.Start(filePath);  }  catch (Exception ex)  {    }  }  }  } |

Listing 5. – Prikaz koda za pregled karte

* **Zaključak**

Digitalizacijom procesa upravljanja avionskim kartama znatno je pojednostavljena i ubrzana administracija kompletnog sistema, čime su smanjene greške koje nastaju usled ručne obrade podataka, kao što su duple rezervacije ili neažurirane informacije o kartama.

Zahvaljujući mogućnosti brzog pristupa i ažuriranja podataka o kartama i korisnicima, administratori sada mogu daleko efikasnije da upravljaju rezervacijama, što značajno štedi vreme, smanjuje potrebu za papirologijom i omogućava tačniju evidenciju u svakom trenutku.

Korisnički interfejs je jednostavan, pregledan i intuitivan, kako za krajnje korisnike koji pretražuju karte, tako i za administratore koji upravljaju celokupnom bazom.

Mogućnosti za unapređenje sistema uključuju uvođenje korisničkih naloga sa različitim nivoima pristupa i zaštitu podataka enkripcijom.