



**Univerzitet u Novom Sadu
Tehnički fakultet „Mihajlo Pupin“,
Zrenjanin**



SEMINARSKI RAD

**Predmet: Razvoj softvera otvorenog koda
Tema: Autobuska Stanica**

**Profesor: Prof. dr Zoltan Kazi
Asistent: MSc Velibor Premčevski**

Studenti: Tea Grubanov SI 33/19

Zrenjanin, 2024.

Sadržaj

1.	Uvod	3
2.	Specifikacija zahteva korisnika	4
3.	Faze razvoja softvera.....	8
3.1	Projektne ideje	9
4.	Prikaz softvera.....	10
5.	Prikaz realizacija i implementacije	18
5.1	Baza podataka.....	18
5.2.	Github sistem	23
5.3.	Kod	24
6.	Korišćeni alati i softveri	40
7.	Lista tabela, slika i listinga.....	41
7.1.	Lista tabela.....	41
7.2.	Lista slika.....	41
8.	Literatura	42

1. Uvod

Seminarski rad iz predmeta Razvoj softvera otvorenog koda prikazuje izradu aplikacije za rezervaciju autobuskih karata, kao i pristup aplikaciji iz ugla korisnika i admina. Izrada aplikacije je rađena u programskom jeziku C# (Windows Presentation Foundation WPF), u razvojnom okruženju Microsoft Visual Studio 2019. Svrha izrade seminarskog rada je kreiranje višeslojne aplikacije kroz softver otvorenog koda, kao i sticanje praktičnog iskustva u korišćenju raznih alata za razvoj softvera kroz timski ili individualni rad. [1]

Funkcionalnost aplikacije su registrovanje korisnika kome je omogućeno kasnije logovanje u aplikaciju, rezervisanje karata, kao i upravljanje podacima za vožnje i rezervacije.

Aplikacija ima dva pregleda, administratorski i korisnički. Prvo je potrebno da se korisnik uloguje kako bi se odredilo koji pristup se otvara. Za korisnike se omogućava jednostavna rezervacija karata za dostupne vožnje, a admini mogu da upravljaju tim vožnjama u smislu da ih mogu dodavati, brisati i menjati. Takođe, admini imaju uvid u rezervacije i mogu ih brisati.

2. Specifikacija zahteva korisnika

Spisak zahteva korisnika predstavljen je u okviru tabele broj 1. Svaki zahtev ima svoju šifru i unapred određen prioritet izražen brojevima od 1 do 4 (gde 1 predstavlja zahtev visoke važnosti, a 4 niže). Zahtevi su navedeni redosledom kojim korisnik navodi, a prioritet je određen u saglasnosti sa klijentom, a dogovorom članova tima u zavisnosti od očekivanog redosleda rešavanja aktivnosti, zauzetosti i projektnih obaveza svakog člana.

Tabela 1. Spisak zahteva korisnika

Redni broj zahteva	Zahtev	Šifra zahteva	Prioritet
1	Registracija korisnika	R1	1
2	Prijavljanje korisnika na nalog	R2	1
3	Prijavljanje admina na nalog	R3	3
4	Opis greške prilikom prijave	R4	4
5	Upravljanje podacima o vožnjama	R5	2
6	Rezervacija karata	R6	2
7	Upravljanje rezervacijama	R7	3

Specifikacija svakog od zahteva je detaljno opisana u okviru tabele broj 2 na sledećoj strani. Način na koji je sročena deskripcija je rezultat razgovora sa korisnikom, pri čemu su dati predlozi u toku pregovaranja, pa se u tabeli nalazi opis zahteva kao krajnjeg rešenja kojim su obe strane zadovoljne.

Tabela 2. Specifikacija zahteva korisnika

Šifra zahteva	Zahtev	Specifikacija zahteva
R1	Registracija korisnika	Kreiranje korisničkog naloga, nalog se sastoji od imena, prezimena, email-a, korisničkog imena i lozinke. Kada se registruje admin pored ovih podataka postoji i atribut u bazi podataka Role. Kada je vrednost tog atributa Admin

		onda će aplikacija prilikom prijave prepoznati da je u pitanju administrator i otvorice se admin pristup. Prilikom registracije se svakom korisniku dodeljuje ID.
R2	Prijavljanje korisnika na nalog	<p>Prijavljanje prethodno registrovanih korisnika unosom korisničkog imena i lozinke.</p> <p>Ukoliko su svi uneti podaci ispravni i povezani sa postojećim nalogom, klikom na dugme prelazi se na korisnički pristup.</p> <p>Ukoliko podaci nisu dobro uneti ili fali neki od podataka izlazi poruka sa informacijom.</p>
R3	Prijavljanje admina na nalog	<p>Prijavljanje prethodno registrovanih admina unosom korisničkog imena i lozinke.</p> <p>Ukoliko su svi uneti podaci ispravni i povezani sa postojećim nalogom i na njemu atribut Role ima vrednost Admin otvara se admin pristup.</p> <p>Ukoliko podaci nisu dobro uneti ili fali neki od podataka izlazi poruka sa informacijom.</p>
R4	Opis greške prilikom prijave	Kod registracije se proverava da li nalog sa istim korisničkim imenom već postoji, ukoliko postoji ispisuje se odgovarajuća poruka. Kod prijave se proverava da li korisničko ime i lozinka odgovaraju nekom nalogu, ukoliko ne ispisuje se poruka a korisničko ime i lozinka nisu ispravno uneti.
R5	Upravljanje podacima o vožnjama	U admin pristupu administrator upravlja informacijama o vožnjama: ID vožnje, naziv prevoznika, datum, vreme, polazak i dolazak. On može dodavati nove informacije, kao i brisati i menjati već evidentirane.
R6	Rezervacija karata	Korisnici mogu rezervisati voznu kartu za dostupne relacije. Korisnik se jedino mora prvo registrovati, a zatim prijaviti na svoj nalog.

R7	Upravljanje rezervacijama	Admini imaju pristup rezervacijama karata, pri čemu se prikazuju sledeći podaci: ID vožnje, naziv prevoznika, datum, vreme, polazak, dolazak, ID korisnika i ime i prezime korisnika. Admin može brisati ove rezervacije, a prilikom izmene vožnje za koju postoji rezervacija podaci se automatski menjaju i u tabeli rezervacije.
----	---------------------------	---

3. Faze razvoja softvera

Životni ciklus razvoja softvera je proces koji se koristi u softverskoj industriji za dizajn, razvoj i testiranje visoko kvalitetnih softverskih rešenja. Ovakav pregled razvoja softvera po fazama ima za cilj proizvodnju kvalitetnih softvera koji ispunjavaju ili prevazilaze očekivanja kupaca, dostižu završetak u vremenu i troškovima koji su predviđeni. [2]

Originalno, postojalo je pet faza u razvoju softvera, sa tim da nisu uključivale analizu i održavanje. Upotpunjena kategorizacija faza razvoja softvera danas se sastoji od sedam faza. [3]

U toku izrade ovog semianrskog rada, uviđa se tih 7 faza, sa tim da je uvedena i osma (ispod navedena pod brojem 3) koja podrazumeva korišćenje sistema za kontrolu verzija. U izradi rada učestvovala su 4 člana, pri čemu nije svaki član imao zaduženja u svakoj od faza. Sedam faza je navedeno i opisano u daljem tekstu.

Faze razvoja:

1. Analiza i navođenje specifikacija zahteva klijenta ili korisnika
2. Razmatranje i odabir tehnologija, definisanje obaveza svakog korisnika (uzimajući u obzir stručnost i stepen poznavanja svake od faza)
3. Kreiranje repozitorijuma na sistemu za kontrolu verzijama - na Githubu (engl. "GitHub")
4. Kreiranje baze podataka – koristeći program Microsoft SQL Server Management Studio
5. Kreiranje bekenda (engl. backend)
 - Kreiranje servera
 - Povezivanje sa bazom podataka
6. Kreiranje frontenda
 - Izgled prozora za registrovanje, prijavljivanje i rezervisanje
 - Dizajn svakog od prozora (functionalnosti prozora na svakom od njih, dodavanje dugmadi za potvrdu ili kretanje po prozorima, dodavanje slika, tekstova...)
 - Dizajn slika korišćenih za pozadinu, uređivanje fotografija
7. Kontinuirana integracija i testiranje svih slojeva aplikacije
8. Isporuka gotovog softvera [3]

3.1 Projektne ideje

U toku faze odabira tehnologija, analizirane su i ideje koje se tiču planiranja tipa softvera, izbora baze podataka i ostalih alata za razvoj, testiranje i izvršavanje softvera. U tabeli broj 3 na sledećoj strani navedene su projektne ideje po redu nastalom u tomu timskog brejnstorminga (engl. “brainstorming”), sa ciljem da ideje prate hronološki sled događaja.

Tabela 3. Projektne ideje i njihovi opisi

Projektna ideja	Opis projektnih ideja
Vrsta (tip) softvera	Računarska aplikacija
Funkcije softvera	Registracija i prijava korisnika i admina, rezervacija. Mogućnosti dodavanja, izmene i brisanja podataka o vožnjama za koje se kreiraju rezervacije voznih karata.
Korisnici softvera	1) Admin: dodavanje, izmena i brisanje sadržaja. 2) Registrovan, ulogovan korisnik: pregled dostupnih karata za određene relacije i njihova rezervacija
OS	“MS Windows“
Arhitektura softvera	Trojslojna. Slojevi: 1) Sloj podataka: “MSSMS“, relaciona baza podataka 2) Aplikacioni sloj: programski jezik C#, razvojno okruženje: “Visual Studio“ 3) Prezentacioni sloj: XAML
Server	SQL Server Management Studio
Baza podataka	“MSSMS“
Testiranje	Metoda crne kutije
Dokumentovanje	„MS Word“
CVS sistem	“GitHub“ sistem za kontrolu verzija

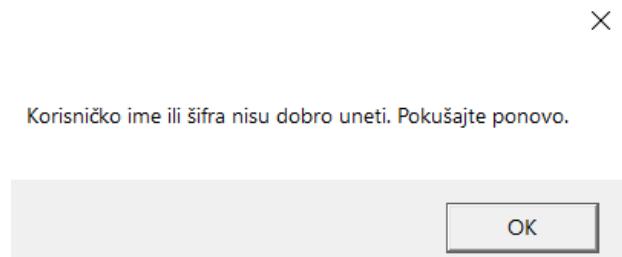
4. Prikaz softvera

Na slici 1. prikazan je izgled pokrenute aplikacije tj. prozor za prijavu.



Slika 1. Prijava

Iskačući prozor koji upućuje na postojanje greške u unosu podataka prikazan je na slici 4.



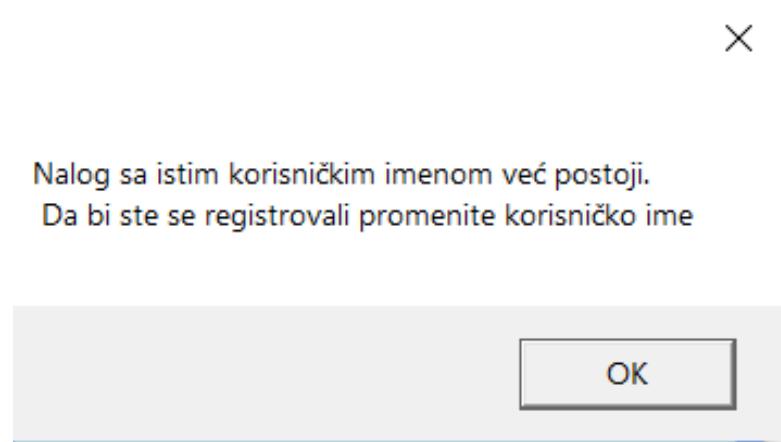
Slika 2. Greška Prijava

Na slici 3. prikazan je izgled prozora za registraciju.



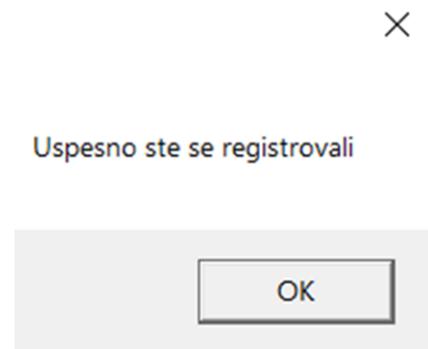
Slika 3. Registracija

Iskačući prozor koji nas informiše da nalog sa unetim podacima već postoji, prikazan je na slici 6.



Slika 4. Greška Registracija

Iskačući prozor koji nas obaveštava o uspešnoj registraciji prikazan je na slici 5.



Slika 5. Uspešna registracija

Na slici 6. nalazi se izgled početne strane admin pristupa.



Slika 6. Početna strana

Na slici 7. prikazana je lista vožnji kojima admin može da upravlja, odnosno može da doda, izmeni ili obriše neku od stavki.

ID Vožnje	Naziv prevo:	Datum polas	Vreme polas	Mesto polas	Mesto dolas
2	Kikinda	8/21/2024 12:00:00	12:00:00	Kikinda	Zrenjanin
9	Banat Trans	7/25/2024 12:00:00	09:00:00	Beograd	Zrenjanin
10	Lasta	9/13/2024 12:16:35:00	16:35:00	Leskovac	Zlatibor

Prevoznik

Datum polaska Select a date 15

Vreme polaska

Mesto polaska

Mesto dolaska

Dodaj **Izmeni** **Obrisi**

Nazad **Obriši selekciju**

Slika 7. Upravljanje Vožnjama

Na slici 8. nalazi se izgled prikaza podataka prilikom odabira željene vožnje kako bi se olakšala izmena i brisanje podataka.



Slika 8. Prikaz podataka

Na slici 9. prikazana je lista vožnji sa svim potrebnim informacijama.

ID Vožnje	Naziv prevoznika	Datum polaska	Vreme polaska	Mesto polaska	Mesto dolaska
2	Kikinda	8/21/2024 12:00:00 AM	12:00:00	Kikinda	Zrenjanin
9	Banat Trans	7/25/2024 12:00:00 AM	09:00:00	Beograd	Zrenjanin
10	Lasta	9/13/2024 12:00:00 AM	16:35:00	Leskovac	Zlatibor
11	Banat Trans	10/11/2024 12:00:00 AM	07:00:00	Beograd	Zrenjanin

[Nazad](#) [Odjavi se](#)

Slika 9. Pregled vožnji

Na slici 10. prikazana je lista rezervacija i njiove informacije sa mogućnošću brisanja rezervacije od strane admina

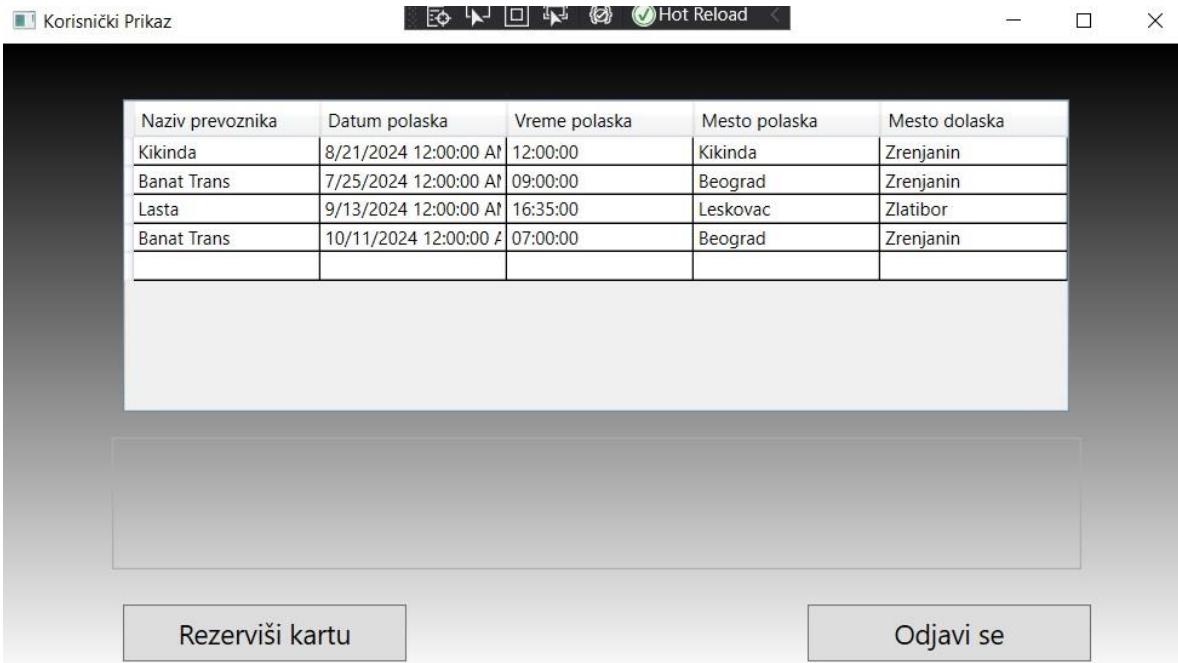
ID Vožnje	Naziv prevoznika	Datum polaska	Vreme polaska	Mesto polaska	Mesto dolaska	ID Korisnika	Putnik
9	Banat Trans	7/25/2024 12:00:00	09:00:00	Beograd	Zrenjanin	4	Nikola Nikolic
11	Banat Trans	10/11/2024 12:00:00	07:00:00	Beograd	Zrenjanin	4	Nikola Nikolic
2	Kikinda	8/21/2024 12:00:00	12:00:00	Kikinda	Zrenjanin	3	2 2
9	Banat Trans	7/25/2024 12:00:00	09:00:00	Beograd	Zrenjanin	3	2 2
11	Banat Trans	10/11/2024 12:00:00	07:00:00	Beograd	Zrenjanin	5	Jovana Mihajlovic

Istekle rezervacije Otkazane rezervacije Otkaži rezervaciju

Nazad

Slika 10. Rezervacije

Na slici 11. se nalazi korisnički prikaz prozora za rezervaciju karata gde korisnik može da odabere željenu vožnju i rezerviše kartu za istu.



Slika 11. Rezervacija karata

Izgled rezervisane karte korisnika se nalazi na slici 12.



Slika 12. Karta

5. Prikaz realizacija i implementacije

5.1 Baza podataka

Baza podataka Zeleznicka__Stanica se sastoji od četiri tabele - korisnik, vožnja, rezervacija, prevoznik.

U listingu 1. prikazana je tabela korisnik

Listing 1. Tabela korisnik

```
USE [AutobuskaStanica]
GO

/***** Object: Table [dbo].[Korisnik]      Script Date:
05/16/2024 22:02:35 *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[Korisnik](
[idKorisnika] [int] IDENTITY(1,1) NOT NULL,
[Ime] [nvarchar](30) NOT NULL,
[Prezime] [nvarchar](30) NOT NULL,
[Korisnickoime] [nvarchar](255) NOT NULL,
[Lozinka] [nvarchar](255) NOT NULL,
[Status] [nvarchar](50) NULL,
PRIMARY KEY CLUSTERED
(
    [idKorisnika] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
```

U listingu 2. prikazana je tabela rezervacija

Listing 2. Tabela rezervacija

```
USE [AutobuskaStanica]
GO
```

```

/***** Object: Table [dbo].[Rezervacija]      Script Date:
05/16/2024 22:03:30 *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[Rezervacija](
    [idKorisnika] [int] NOT NULL,
    [idVoznje] [int] NOT NULL,
    [Status] [nvarchar] (50) NULL
) ON [PRIMARY]

GO

ALTER TABLE [dbo].[Rezervacija] WITH CHECK ADD CONSTRAINT
[FK_rezervacija_korisnik] FOREIGN KEY([idKorisnika])
REFERENCES [dbo].[Korisnik] ([idKorisnika])
GO

ALTER TABLE [dbo].[Rezervacija] CHECK CONSTRAINT
[FK_rezervacija_korisnik]
GO

ALTER TABLE [dbo].[Rezervacija] WITH CHECK ADD CONSTRAINT
[FK_rezervacija_statusrezervacija] FOREIGN KEY([Status])
REFERENCES [dbo].[StatusRezervacije] ([Status])
GO

ALTER TABLE [dbo].[Rezervacija] CHECK CONSTRAINT
[FK_rezervacija_statusrezervacija]
GO

ALTER TABLE [dbo].[Rezervacija] WITH CHECK ADD CONSTRAINT
[FK_rezervacija_voznja] FOREIGN KEY([idVoznje])
REFERENCES [dbo].[Voznja] ([idVoznje])
GO

ALTER TABLE [dbo].[Rezervacija] CHECK CONSTRAINT
[FK_rezervacija_voznja]
GO

```

```

ALTER TABLE Rezervacija
ADD CONSTRAINT FK_Rezervacija_Voznja FOREIGN KEY (IDVoznje)
REFERENCES Voznja (IDVoznje)

INSERT      [dbo].[Rezervacija] ([IDKorisnika], [IDVoznje])
VALUES (7, 1)

```

U listingu 3. prikazana je tabela vožnja

Listing 3. Tabela vožnja

```

USE [AutobuskaStanica]
GO

/***** Object: Table [dbo].[Voznja]      Script Date:
05/16/2024 22:04:49 *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[Voznja] (
    [idVoznje] [int] IDENTITY(1,1) NOT NULL,
    [NazivPrevoznika] [nvarchar](255) NOT NULL,
    [Datum] [date] NOT NULL,
    [Vreme] [time](7) NOT NULL,
    [MestoPolaska] [nvarchar](50) NOT NULL,
    [MestoDolaska] [nvarchar](50) NOT NULL,
PRIMARY KEY CLUSTERED
(
    [idVoznje] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO

ALTER TABLE [dbo].[Voznja] WITH CHECK ADD CONSTRAINT
[FK_voznja_prevoznik] FOREIGN KEY([NazivPrevoznika])
REFERENCES [dbo].[Prevoznik] ([NazivPrevoznika])
GO

ALTER TABLE [dbo].[Voznja] CHECK CONSTRAINT
[FK_voznja_prevoznik]

```

```
GO
```

U listingu 4. prikazana je tabela prevoznik

Listing 4. Tabela prevoznik

```
USE [AutobuskaStanica]
GO

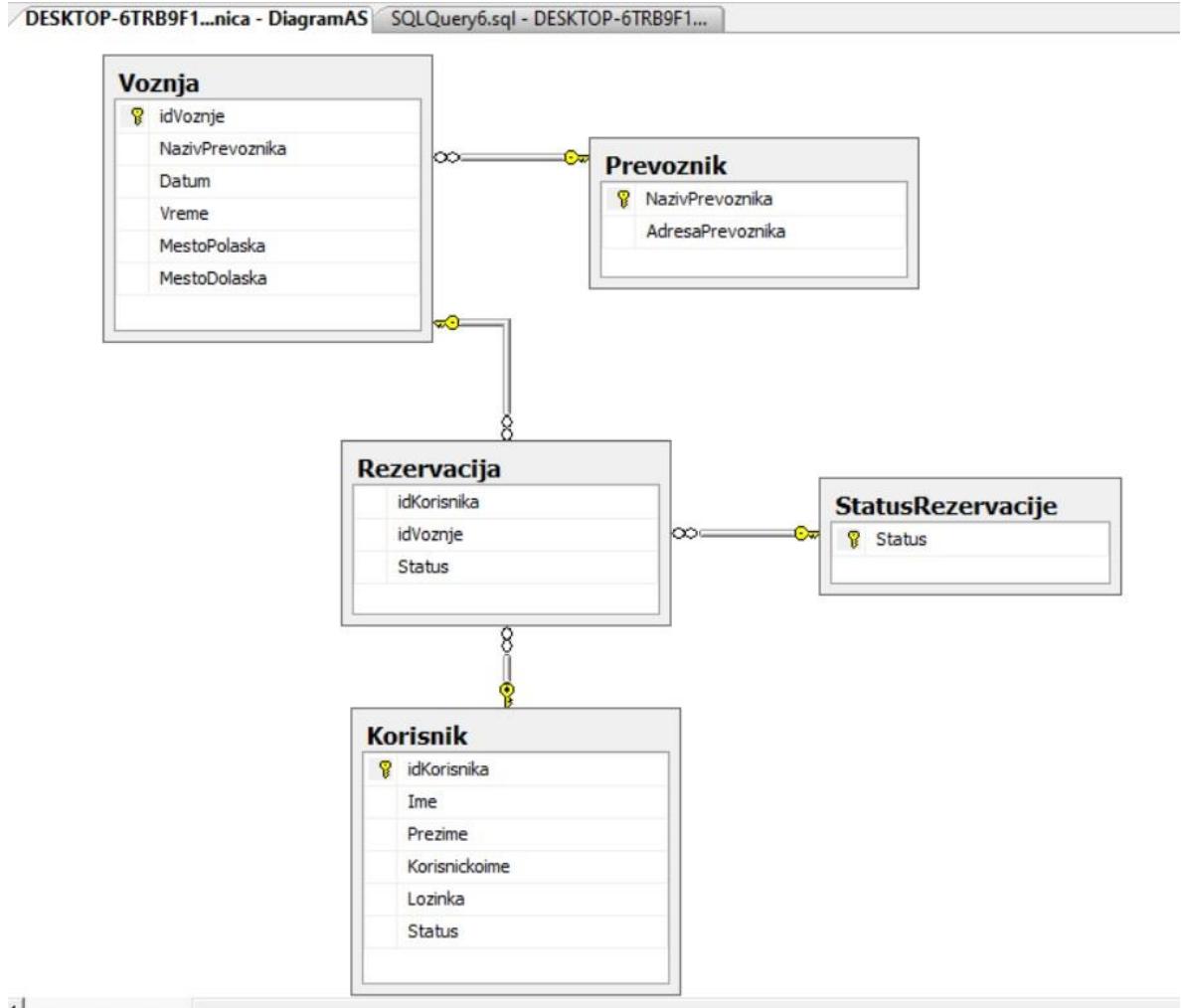
/***** Object: Table [dbo].[Prevoznik]      Script Date:
05/16/2024 22:05:23 *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[Prevoznik] (
    [NazivPrevoznika] [nvarchar](255) NOT NULL,
    [AdresaPrevoznika] [nvarchar](255) NOT NULL,
PRIMARY KEY CLUSTERED
(
    [NazivPrevoznika] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
```

Dijagram koji je kreiran u SQL Server Management Studio-u prikazan je na slici 13.



Slika 13. Dijagram baze podataka

5.2. Github sistem

Kreirani repozitorijum: <https://github.com/teagrubanov/AutobuskaStanica---Razvoj-softvera-otvorenog-koda/tree/master>

Kreirane grane: 1. Backend

2. Frontend

3. Baza-podataka

Ove grane su prikazane na slici 14.

Name	Last commit message	Last commit date
AS	Add project files.	last week
.gitattributes	Add .gitignore and .gitattributes.	last week
.gitignore	Add .gitignore and .gitattributes.	last week
AS.sln	Add project files.	last week
Full Database query.sql	Add files via upload	last week
README.md	Create README.md	last week

Slika 14. Lista kreiranih grana na Githubu

Postavljanje datoteka: Sve datoteke su, redom kako su kompletirane, postavljanje putem “Upload files” mogućnosti. Ovo se odnosi na postavljanje na bilo koju od tri grane, u zavisnosti od korisnika i fazi kojoj pripada datoteka koja se postavlja.

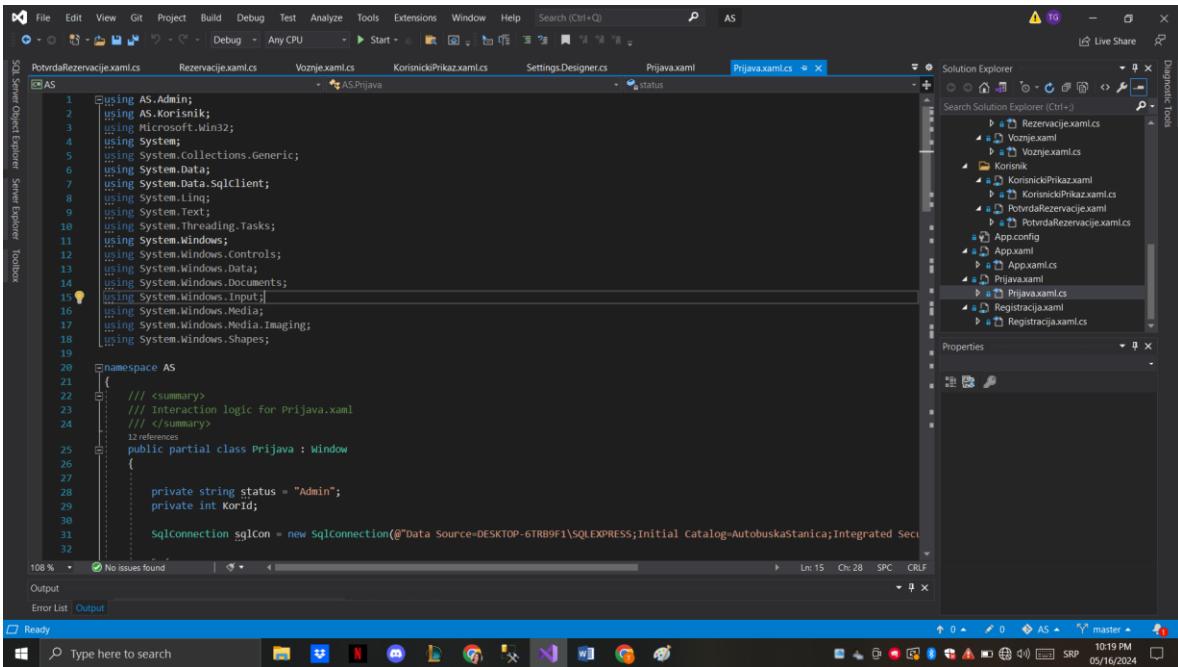
Preuzimanje datoteka: Preuzimanje čitave aplikacije kao zapakovane arhive se izvršava opcijom “Download ZIP“.

Spajanje grana: Sve grane se spajaju nakon što je projekat realizovan i sve datoteke postavljene na neku od grana. Postiže se opcijom “Pull request“.

5.3. Kod

Prvo je potrebno povezati projekat sa bazom podataka. Konekcioni string se može dodati na dva načina. Prvi način je dodavanje u App.config-u. U Data Source-u je naziv računara koji se koristi i instanca servera, a u Initial Catalog-u je naziv baze podataka. Kada se kreira konekcioni string potrebno je da se doda referenca da prozori mogu da rade sa konfiguracionom bazom u kojem je putanja do baze podataka. Referenca se dodaje klikom na desni klik na *Add - Reference* i otvara se *Reference Manager* u kojem je potrebno da se čekira *System Configuration*. Drugi način povezivanja sa bazom podataka se vrši prilikom kreiranja svakog prozora u xaml.cs fajlu

Na sledećem listingu prikazan je kod za prozor Prijava. To je prvi prozor koji se otvara prilikom pokretanja aplikacije. Nakon ostvarene konekcije sa bazom podataka, proverava se unos korisničkog imena i lozinke klikom na dugme za prijavu. Prvo se proverava da li korisnik ima Role atribut sa vrednosti Admin. Ako ima znači da se prijavljuje admin i da mu se otvara admin pristup. Ako nema onda se otvara korisnik pristup. Pored ovog atributa proverava se da li uneto korisničko ime i lozinka odgovaraju nekom nalogu, ako ne ispisuje se poruka da korisničko ime ili lozinka nisu dobro uneti. Ako korisnik nema nalog klikom na dugme za registraciju mu se otvara prozor u kom može da se registruje.



The screenshot shows the Microsoft Visual Studio IDE interface. The code editor displays the `Prijava.xaml.cs` file, which contains C# code for a login window. The Solution Explorer on the right shows the project structure with files like `PotvrdaRezervacije.xaml.cs`, `Rezervacije.xaml.cs`, `Voznje.xaml.cs`, `KorisnickiPrikaz.xaml.cs`, `Settings.Designer.cs`, `Prijava.xaml`, and `Prijava.xaml.cs`. The Properties window is also visible. The taskbar at the bottom shows various pinned icons, and the system tray indicates the date and time as 05/16/2024 at 10:19 PM.

```
1  using AS.Admin;
2  using AS.Korisnik;
3  using Microsoft.Win32;
4  using System;
5  using System.Collections.Generic;
6  using System.Data;
7  using System.Data.SqlClient;
8  using System.Linq;
9  using System.Text;
10 using System.Threading.Tasks;
11 using System.Windows;
12 using System.Windows.Controls;
13 using System.Windows.Data;
14 using System.Windows.Documents;
15 using System.Windows.Input;
16 using System.Windows.Media;
17 using System.Windows.Media.Imaging;
18 using System.Windows.Shapes;
19
20 namespace AS
21 {
22     /// <summary>
23     /// Interaction logic for Prijava.xaml
24     /// </summary>
25     public partial class Prijava : Window
26     {
27         private string status = "Admin";
28         private int korid;
29
30         SqlConnection sqlCon = new SqlConnection(@"Data Source=DESKTOP-6TRB9F1\SQLEXPRESS;Initial Catalog=AutobuskaStanica;Integrated Security=True");
31
32     }
33 }
```

Two screenshots of the Microsoft Visual Studio IDE interface showing the code editor and Solution Explorer.

Screenshot 1 (Top):

- Code Editor:** Displays the file `Potvrdarezervacije.xaml.cs`. The code implements a `Prijava()` constructor and a `btnPrijava_Click(object sender, RoutedEventArgs e)` event handler. The event handler checks if the connection state is closed, opens it, and then executes two SQL commands: one to count users by name and another to get the user ID by name and password.
- Solution Explorer:** Shows the project structure with files like `Rezervacije.xaml.cs`, `Voznje.xaml.cs`, `KorisnickiPrikaz.xaml.cs`, `Settings.Designer.cs`, `Prijavaxaml.cs`, and `Potvrdarezervacije.xaml.cs`.
- Properties:** Shows the properties for the selected file.

Screenshot 2 (Bottom):

- Code Editor:** Displays the file `Potvrdarezervacije.xaml.cs`. This version of the code includes a try-catch block around the command execution. It handles exceptions by displaying a message box and closing the connection.
- Solution Explorer:** Shows the same project structure as the first screenshot.
- Properties:** Shows the properties for the selected file.

The screenshot shows the Microsoft Visual Studio IDE interface. The main window displays the code for 'Prijava.xaml.cs'. The code handles user input and database operations, specifically checking if a user exists and displaying error messages if necessary. It also includes logic for opening a registration window ('Registracija') if the user does not exist. The Solution Explorer on the right shows the project structure, including files like 'Rezervacije.xaml.cs', 'Voznje.xaml.cs', 'KorisnickiPrikaz.xaml.cs', 'Settings.Designer.cs', 'Prijava.xaml', and 'Prijava.xaml.cs'. The Properties window is also visible.

```
75     dashboard.Show();
76     this.Close();
77   }
78   else
79   {
80     MessageBox.Show("Korisničko ime ili šifra nisu dobro uneti. Pokušajte ponovo.");
81   }
82 }
83 catch (Exception ex)
84 {
85   MessageBox.Show(ex.Message);
86 }
87 finally
88 {
89   sqlCon.Close();
90 }
91 }
92 }
93
94 reference
95 private void btnRegistracija_Click(object sender, RoutedEventArgs e)
96 {
97   Registracija dashboard = new Registracija();
98   dashboard.Show();
99   this.Close();
100 }
101 }
102 }
```

Slike 15, 16, 17 i 18. – Prijava prozor

U prozoru Registracija korisnik unosi ime, prezime, email, korisničko ime i lozinku. Klikom na dugme registruj se se proverava da li korisničko ime odgovara nekom korisničkom imenu koje se već nalazi u bazi podataka, ako odgovara onda se ispisuje poruka da korisničko ime već postoji i da korisnik izabere novo korisničko ime. Ako se utvrdi da korisničko ime nije već zauzeto, onda se ovi uneti podaci upisuju u tabelu Korisnik i zatim se korisniku ispisuje poruka da se uspešno registrovao i otvara mu se Prijava prozor gde tada može da se prijavi korišćenjem prethodno kreiranog korisničkog imena i lozinke. Dugme Nazad vraća korisnika na prozor za prijavljivanje.

```

7  using System.Threading.Tasks;
8  using System.Windows;
9  using System.Windows.Controls;
10 using System.Windows.Data;
11 using System.Windows.Documents;
12 using System.Windows.Input;
13 using System.Windows.Media;
14 using System.Windows.Media.Imaging;
15 using System.Windows.Shapes;
16
17 namespace AS
18 {
19     /// <summary>
20     /// Interaction logic for Registracija.xaml
21     /// </summary>
22     public partial class Registracija : Window
23     {
24         SqlConnection sqlCon = new SqlConnection(@"Data Source=DESKTOP-6TRB9F1\SQLEXPRESS;Initial Catalog=AutobuskaStanica;Integrated Security=True");
25
26         public void OcistiPodatke()
27         {
28             ImeTextBox.Text = "";
29             PrezimeTextBox.Text = "";
30             KorisnickoImeTextBox.Text = "";
31             SifraPasswordBox.Password = "";
32         }
33
34         public Registracija()
35         {
36             InitializeComponent();
37         }
38     }
39 
```

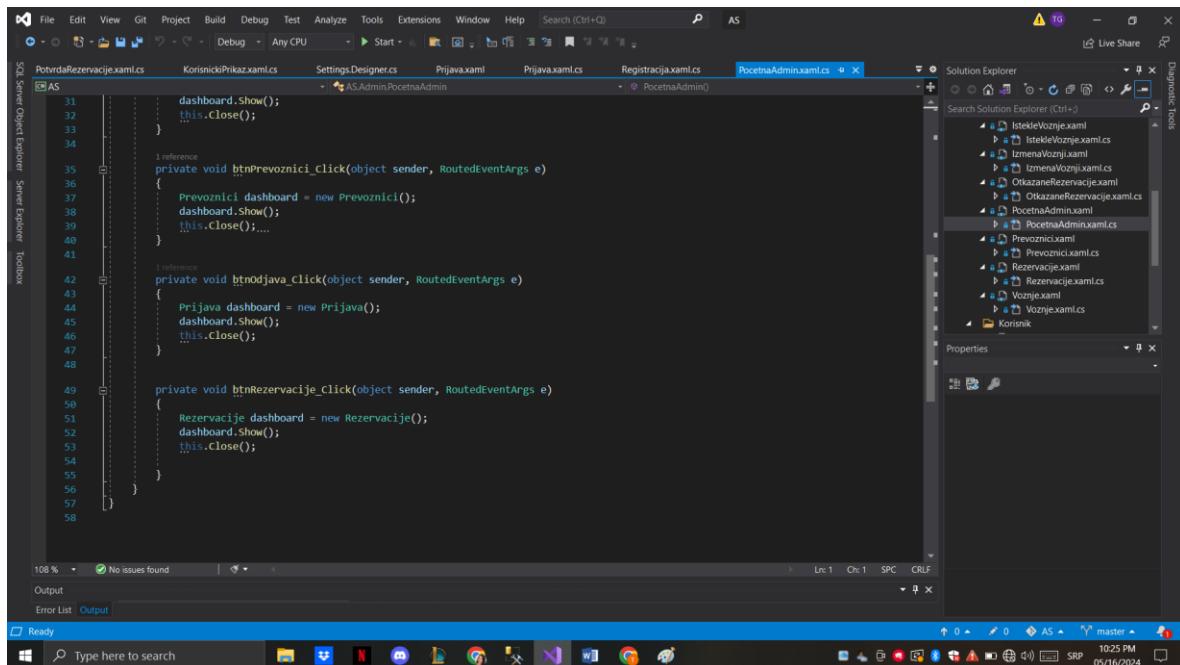
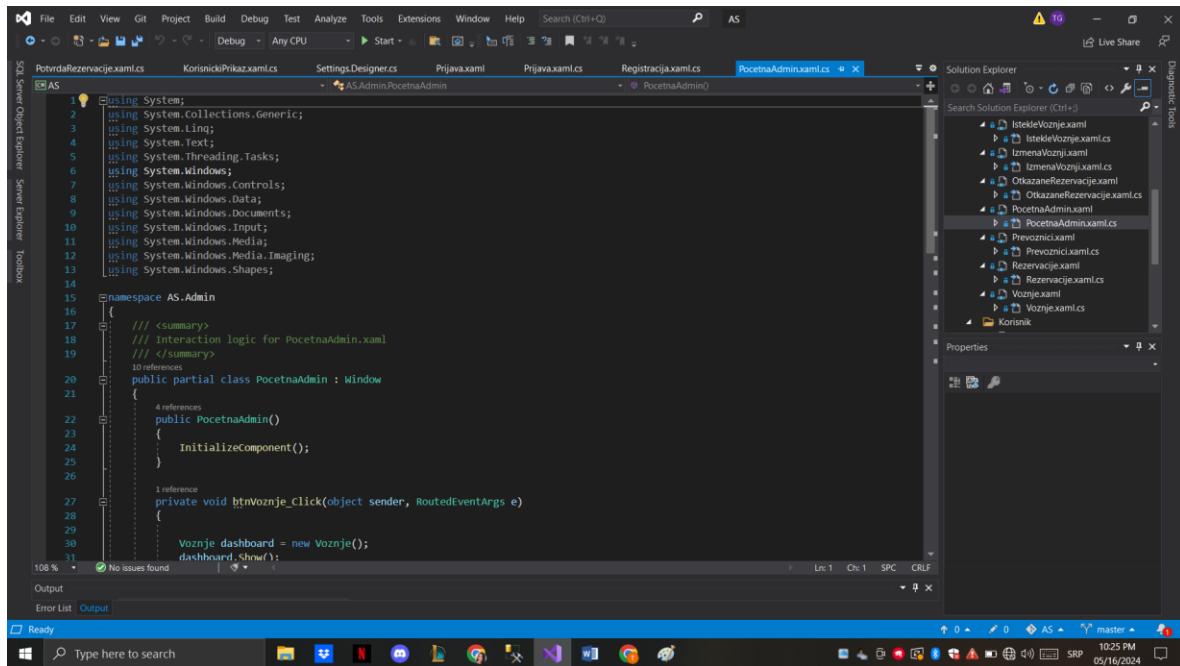
```

35     public Registracija()
36     {
37         InitializeComponent();
38     }
39
40     private void btnRegistracija_Click(object sender, RoutedEventArgs e)
41     {
42         try
43         {
44             if (sqlCon.State == ConnectionState.Closed)
45                 sqlCon.Open();
46             string query = "INSERT INTO Korisnik (ime, prezime, KorisnickoIme, Lozinka) VALUES(@ime, @prezime, @KorisnickoIme, @Lozinka)";
47             string query1 = "SELECT COUNT(*) FROM Korisnik WHERE KorisnickoIme = @KorIme";
48
49             SqlCommand sqlCmdKor = new SqlCommand(query, sqlCon);
50             sqlCmdKor.CommandType = CommandType.Text;
51             sqlCmdKor.Parameters.AddWithValue("@KorIme", KorisnickoImeTextBox.Text);
52             int count = Convert.ToInt32(sqlCmdKor.ExecuteScalar());
53             if (count == 1)
54             {
55                 MessageBox.Show("Nalog sa istim korisničkim imenom već postoji. Da bi ste se registrovali promenite korisničko ime.");
56                 OcistiPodatke();
57             }
58             else
59             {
60                 SqlCommand sqlCmd = new SqlCommand(query, sqlCon);
61                 sqlCmd.CommandType = CommandType.Text;
62                 sqlCmd.Parameters.AddWithValue("@KorisnickoIme", KorisnickoImeTextBox.Text);
63                 sqlCmd.Parameters.AddWithValue("@Lozinka", SifraPasswordBox.Password);
64                 sqlCmd.Parameters.AddWithValue("@ime", ImeTextBox.Text);
65                 sqlCmd.Parameters.AddWithValue("@prezime", PrezimeTextBox.Text);
66             }
67         }
68     }
69 
```

```
65     sqlCmd.Parameters.AddWithValue("@Prezime", PrezimeTextBox.Text);
66
67     int countNew = Convert.ToInt32(sqlCmd.ExecuteScalar());
68
69     MessageBox.Show("Uspešno ste se registrovali!");
70
71     Prijava dashboard = new Prijava();
72     dashboard.Show();
73     this.Close();
74 }
75
76 catch (Exception ex)
77 {
78     MessageBox.Show(ex.Message);
79 }
80 finally
81 {
82     sqlCon.Close();
83 }
84 }
85
86
87 private void btnNazad_Click(object sender, RoutedEventArgs e)
88 {
89     Prijava dashboard = new Prijava();
90     dashboard.Show();
91     this.Close();
92 }
93
94 }
```

Slike 19, 20, 21 – Registracija prozor

Kada se admin uspešno prijavi na svoj nalog otvara mu se prozor PocetnaAdmin. Ovaj prozor se sastoji od pet tastera. Klikom na bilo koji od njih se otvara određeni prozor. Prvi vodi na prozor za pregled vožnji, drugi na pregled rezervacija, treći na prozor za upravljanje podacima o vožnjama, a četvrti na prozor koji sadrži informacije o prevoznicima i omogućava njihovu izmenu. Peto dugme je dugme za odjavu, a klikom na njega se admin odjavljuje i otvara se prozor za prijavljivanje tj. Prijava prozor.



Slike 22, 23– PocetnaAdmin prozor

Prozor za upravljanje informacijama o vožnjama se sastoji od tabele sa podacima o vožnjama, kontrola za unos podataka i određenih tastera. Podaci koji se evidentiraju su: id vožnje, naziv prevoznika, datum polaska, vreme polaska, početna i krajnja destinacija. Kreiranje liste getPrevoznik() omogućava čitanje podataka iz baze u okviru combobox-a. Selektuje se naziv prevoznika iz tabele Prevoznik i zatim se podaci smještaju u combobox. Pored se nalazi dugme za dodavanje novog prevoznika u tabelu Prevoznik, u slučaju da se

potrebni podaci ne nalaze u bazi podataka. Prozor Prvoznik će biti opisan u nastavku rada. Funkcijom LoadDataGrid() se popunjava tabela pomoću naredbe SELECT, tj. DataGrid sa podacima iz baze podataka. Funkcija VoznjeDataGrid_SelectionChanged() omogućava da se klikom na neku vrstu u tabeli njeni podaci selektuju i prikažu u kontrolama za unos podataka, a na taj način se omogućava lakša izmena i brisanje podataka. PonistiUnos() briše sve vrednosti iz kontrola za unos kako bi se unele nove vrednosti prilikom snimanja novih podataka. Klikom na dugme dodaj se podaci uneti u kontrole za unos unose u tabelu Voznja pomoću INSERT INTO upita. Klikom na dugme obriši se briše selektovani red u tabeli, tj. određeni podaci za vožnju u tabeli podataka Voznja koji odgovaraju ID-u vožnje koji je selektovan u tabeli. Ovo se omogućava pomoću SQL upita DELETE. Klikom na dugme izmeni se menjaju podaci selektovani u tabeli podacima koji su uneti u kontrole za unos, a čiji IDVoznje odgovara ID-u vožnje u tabeli Voznja, što znači da se IDVoznje ne može izmeniti, nego samo ostali podaci. Izmena se omogućava pomoću SQL upita UPDATE. Klikom na dugme Isprazni kontrolе se poziva funkcija PonistiUnos() i brišu se svi podaci iz kontrola za unos teksta. Klikom na dugme nazad se otvara prozor PocetnaAdmin.

```

1  using System;
2  using System.Collections.Generic;
3  using System.Data;
4  using System.Data.SqlClient;
5  using System.Linq;
6  using System.Text;
7  using System.Threading.Tasks;
8  using System.Windows;
9  using System.Windows.Controls;
10 using System.Windows.Data;
11 using System.Windows.Documents;
12 using System.Windows.Input;
13 using System.Windows.Media;
14 using System.Windows.Media.Imaging;
15 using System.Windows.Shapes;
16
17 namespace AS.Admin
18 {
19     /// <summary>
20     /// Interaction logic for IzmenaVoznji.xaml
21     /// </summary>
22     public partial class IzmenaVoznji : Window
23     {
24         private int ID;
25
26         SqlConnection sqlCon = new SqlConnection(@"Data Source=DESKTOP-6TRB9F1\SQLEXPRESS;Initial Catalog=AutobuskaStanica;Integrated Security=True");
27
28         private List<string> getPrevoznik()
29         {
30             List<string> results = new List<string>();
31             DataSet Prevoznici = new DataSet();
32             try
33             {
34                 SqlDataAdapter da = new SqlDataAdapter("SELECT Ime FROM Prevoznici", sqlCon);
35                 da.Fill(Prevoznici);
36                 foreach (DataRow dr in Prevoznici.Tables[0].Rows)
37                 {
38                     results.Add(dr["Ime"].ToString());
39                 }
40             }
41             catch { }
42             return results;
43         }
44
45         private void button1_Click(object sender, RoutedEventArgs e)
46         {
47             if (ID != 0)
48             {
49                 string Ime = comboBox1.SelectedItem.ToString();
50                 string Voznik = comboBox2.SelectedItem.ToString();
51                 string Cilj = textBox1.Text;
52                 string Vreme = dateTimePicker1.Text;
53                 string Voznik2 = comboBox3.SelectedItem.ToString();
54                 string Cilj2 = textBox2.Text;
55                 string Vreme2 = dateTimePicker2.Text;
56
57                 string query = "UPDATE Voznje SET Voznik = '" + Voznik + "', Cilj = '" + Cilj + "', Vreme = '" + Vreme + "' WHERE ID = " + ID;
58                 string query2 = "UPDATE Voznici SET Ime = '" + Ime + "' WHERE ID = " + Voznik;
59
60                 SqlCommand cmd = new SqlCommand(query, sqlCon);
61                 cmd.ExecuteNonQuery();
62
63                 SqlCommand cmd2 = new SqlCommand(query2, sqlCon);
64                 cmd2.ExecuteNonQuery();
65
66                 PonistiUnos();
67             }
68         }
69
70         private void button2_Click(object sender, RoutedEventArgs e)
71         {
72             if (ID != 0)
73             {
74                 string query = "DELETE FROM Voznje WHERE ID = " + ID;
75
76                 SqlCommand cmd = new SqlCommand(query, sqlCon);
77                 cmd.ExecuteNonQuery();
78
79                 PonistiUnos();
80             }
81         }
82
83         private void PonistiUnos()
84         {
85             ID = 0;
86             textBox1.Text = "";
87             textBox2.Text = "";
88             dateTimePicker1.Text = "";
89             dateTimePicker2.Text = "";
90             comboBox1.SelectedIndex = -1;
91             comboBox2.SelectedIndex = -1;
92             comboBox3.SelectedIndex = -1;
93         }
94
95         private void button3_Click(object sender, RoutedEventArgs e)
96         {
97             this.Close();
98             PocetnaAdmin admin = new PocetnaAdmin();
99             admin.Show();
100        }
101    }
102}

```

```

    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
    finally
    {
        sqlCon.Close();
    }
    foreach (DataRow row in Prevoznici.Tables[0].Rows)
    {
        results.Add((string)row["NazivPrevoznika"]);
    }
    return results;
}

1 reference
public IzmenaVoznji()
{
    InitializeComponent();
    UcitajDataGrid();
    PrevoznikComboBox.ItemsSource = getPrevoznik();
}

5 references
private void UcitajDataGrid()
{
    sqlCon.Open();
    SqlCommand cmd = new SqlCommand();
    cmd.CommandText = "SELECT * FROM Voznja WHERE cast (Datum as Date)= cast (GETDATE() as Date) OR (Datum > GETDATE())";
    cmd.Connection = sqlCon;
    SqlDataAdapter dataAdapter = new SqlDataAdapter(cmd);
    DataTable dataTable = new DataTable("Voznja");
    dataAdapter.Fill(dataTable);
    DataGrid1.ItemsSource = dataTable.DefaultView;
}

```

```

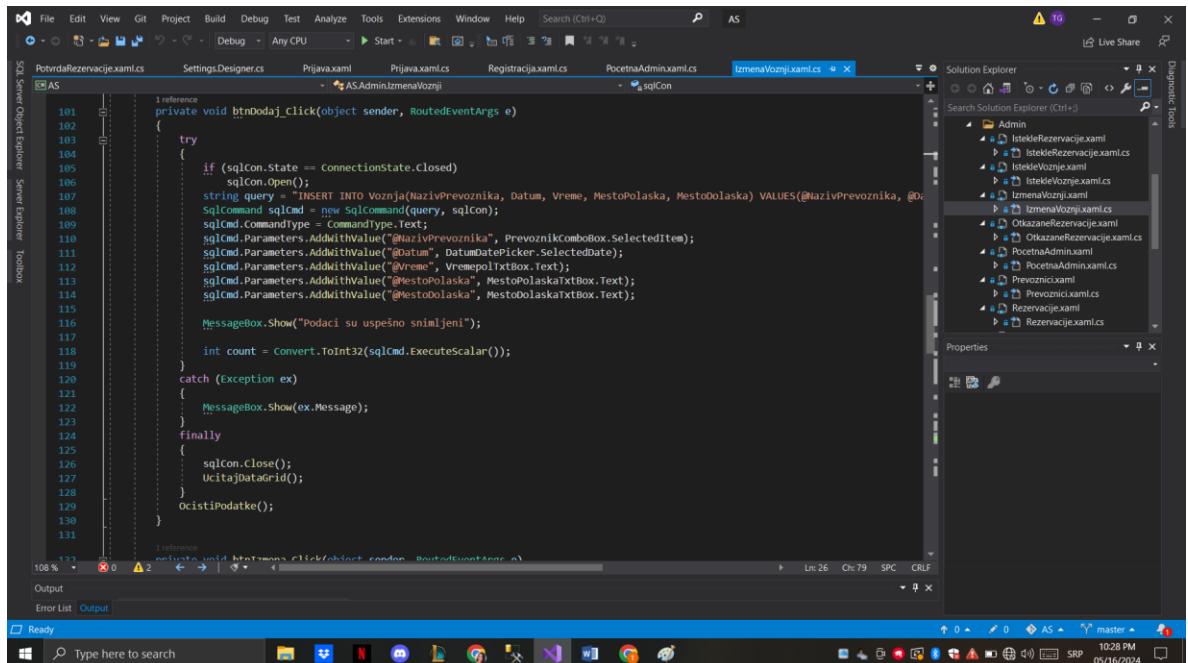
    }

4 references
private void OcistiPodatke()
{
    PrevoznikComboBox.SelectedIndex = -1;
    DatumDatePicker.Text = "";
    VremePol(textBox.Text = "");
    MestoPolaskaTextBox.Text = "";
    MestoBolaskaTextBox.Text = "";
}

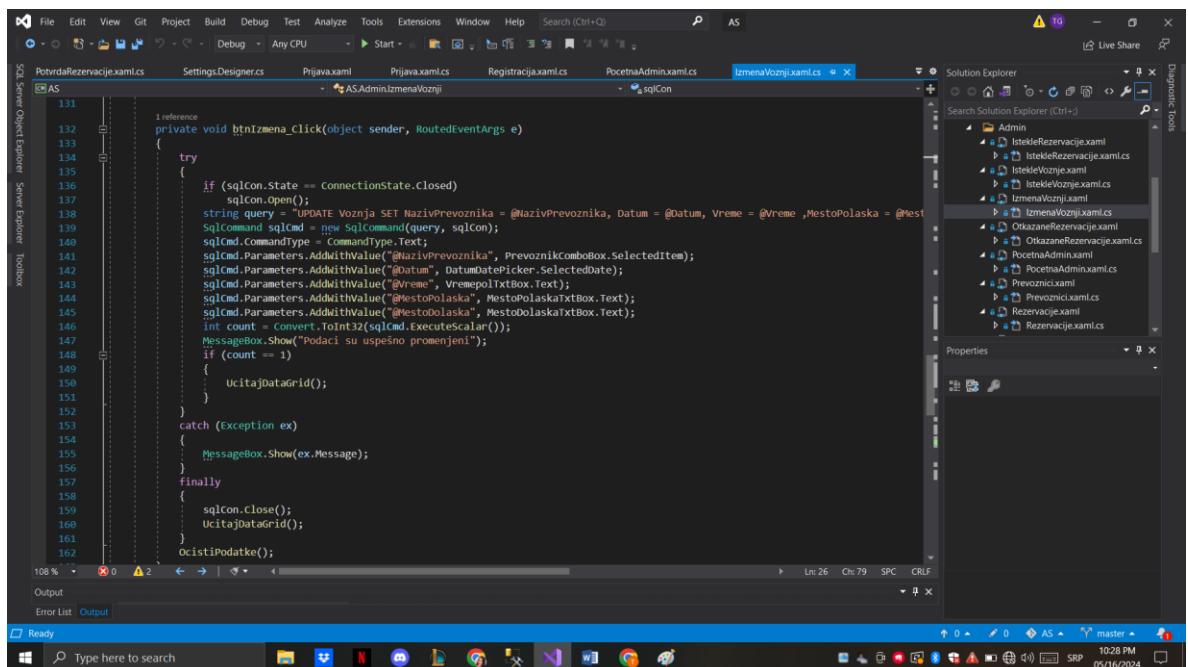
1 reference
private void VoznjeDataGrid_SelectionChanged(object sender, SelectionChangedEventArgs e)
{
    DataGrid dg = sender as DataGrid;
    DataRowView dr = dg.SelectedItem as DataRowView;
    if (dr != null)
    {
        ID = int.Parse(dr["idvozne"].ToString());
        PrevoznikComboBox.SelectedValue = dr["NazivPrevoznika"].ToString();
        DatumDatePicker.Text = dr["Datum"].ToString();
        VremePol(textBox.Text = dr["Vreme"].ToString());
        MestoPolaskaTextBox.Text = dr["MestoPolaska"].ToString();
        MestoBolaskaTextBox.Text = dr["MestoBolaska"].ToString();
    }
}

1 reference
private void btnDodaj_Click(object sender, RoutedEventArgs e)
{
}

```



```
101     private void btnDodataj_Click(object sender, RoutedEventArgs e)
102     {
103         try
104         {
105             if (sqlCon.State == ConnectionState.Closed)
106                 sqlCon.Open();
107             string query = "INSERT INTO Voznja(NazivPrevoznika, Datum, Vreme, MestoPolaska, MestoDolaska) VALUES(@NazivPrevoznika, @Datum, @Vreme, @MestoPolaska, @MestoDolaska)";
108             SqlCommand sqlCommand = new SqlCommand(query, sqlCon);
109             sqlCommand.CommandType = CommandType.Text;
110             sqlCommand.Parameters.AddWithValue("@NazivPrevoznika", PrevoznikComboBox.SelectedItem);
111             sqlCommand.Parameters.AddWithValue("@Datum", DatumDatePicker.SelectedDate);
112             sqlCommand.Parameters.AddWithValue("@Vreme", VremepoljTextBox.Text);
113             sqlCommand.Parameters.AddWithValue("@MestoPolaska", MestoPolaskaTextBox.Text);
114             sqlCommand.Parameters.AddWithValue("@MestoDolaska", MestoDolaskaTextBox.Text);
115
116             MessageBox.Show("Podaci su uspešno snimljeni");
117
118             int count = Convert.ToInt32(sqlCmd.ExecuteScalar());
119         }
120         catch (Exception ex)
121         {
122             MessageBox.Show(ex.Message);
123         }
124         finally
125         {
126             sqlCon.Close();
127             UcitajDataGrid();
128         }
129     }
130
131     private void btnIzmene_Click(object sender, RoutedEventArgs e)
```



```
131
132     private void btnIzmene_Click(object sender, RoutedEventArgs e)
133     {
134         try
135         {
136             if (sqlCon.State == ConnectionState.Closed)
137                 sqlCon.Open();
138             string query = "UPDATE Voznja SET NazivPrevoznika = @NazivPrevoznika, Datum = @Datum, Vreme = @Vreme ,MestoPolaska = @MestoPolaska, MestoDolaska = @MestoDolaska WHERE Id = @Id";
139             SqlCommand sqlCommand = new SqlCommand(query, sqlCon);
140             sqlCommand.CommandType = CommandType.Text;
141             sqlCommand.Parameters.AddWithValue("@NazivPrevoznika", PrevoznikComboBox.SelectedItem);
142             sqlCommand.Parameters.AddWithValue("@Datum", DatumDatePicker.SelectedDate);
143             sqlCommand.Parameters.AddWithValue("@Vreme", VremepoljTextBox.Text);
144             sqlCommand.Parameters.AddWithValue("@MestoPolaska", MestoPolaskaTextBox.Text);
145             sqlCommand.Parameters.AddWithValue("@MestoDolaska", MestoDolaskaTextBox.Text);
146
147             int count = Convert.ToInt32(sqlCmd.ExecuteScalar());
148             MessageBox.Show("Podaci su uspešno promenjeni");
149             if (count == 1)
150             {
151                 UcitajDataGrid();
152             }
153         }
154         catch (Exception ex)
155         {
156             MessageBox.Show(ex.Message);
157         }
158         finally
159         {
160             sqlCon.Close();
161             UcitajDataGrid();
162         }
163     }
164
165     private void OcistiPodatke();
```

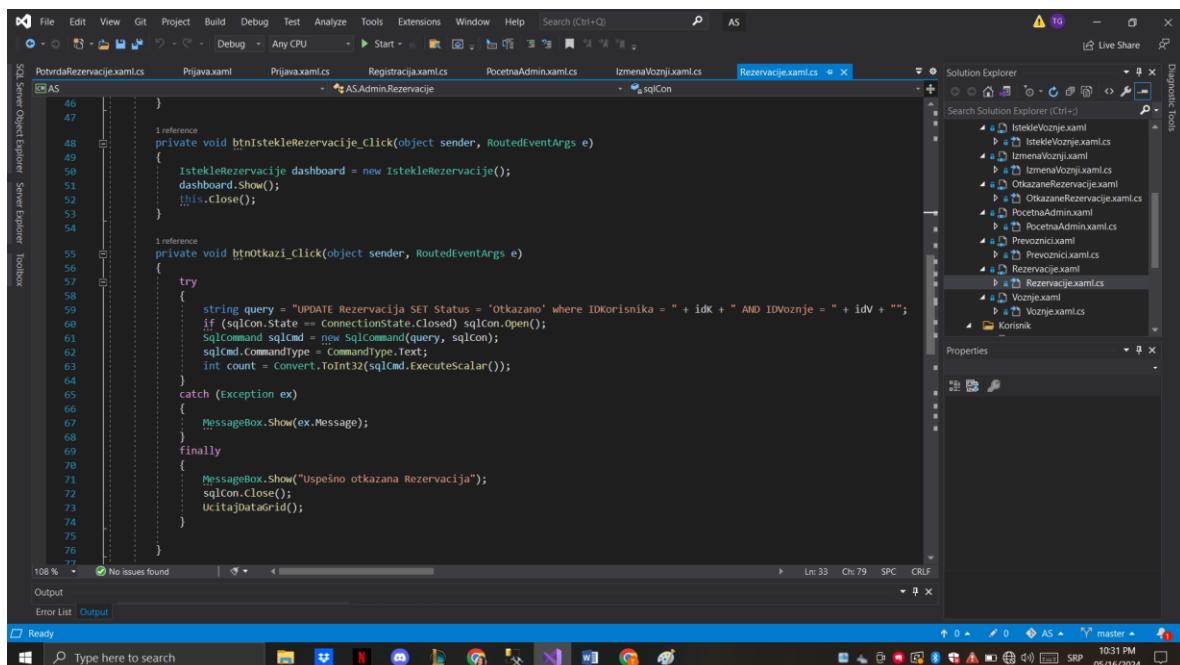
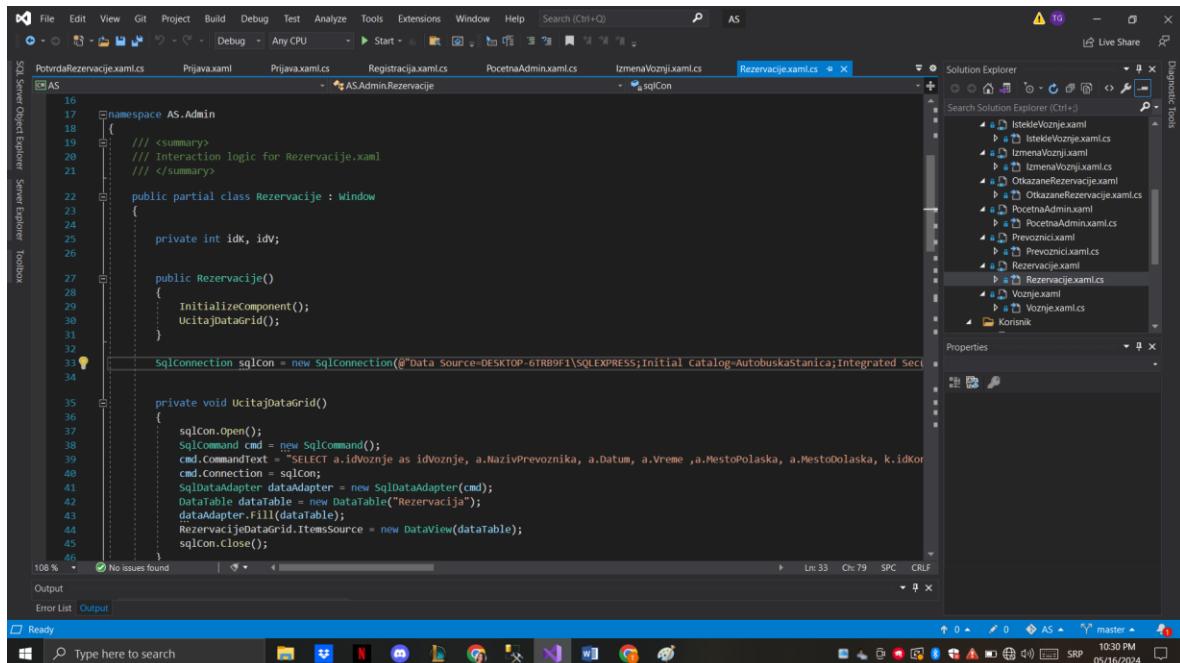
```

168     {
169         if (sqlCon.State == ConnectionState.Closed)
170             sqlCon.Open();
171         string query = "DELETE FROM Voznja WHERE idvozanje = " + ID + "";
172         SqlCommand sqlCmd = new SqlCommand(query, sqlCon);
173         sqlCmd.CommandType = CommandType.Text;
174         int count = Convert.ToInt32(sqlCmd.ExecuteScalar());
175         MessageBox.Show("Podaci su uspešno obrisani");
176     }
177     catch (Exception ex)
178     {
179         MessageBox.Show("Nije moguće obrisati podatke o vožnji zbog postojanja jedne ili više rezervacija za ovu vožnju.");
180     }
181     finally
182     {
183         sqlCon.Close();
184         ucitaDataGrid();
185     }
186     OcistiPodatke();
187 }
188
189 // Reference
190 private void btnNazad_Click(object sender, RoutedEventArgs e)
191 {
192     Voznje dashboardd = new Voznje();
193     dashboardd.Show();
194     this.Close();
195 }
196
197 // Reference
198 private void btnObrisiselekciju_Click(object sender, RoutedEventArgs e)
199 {
200     OcistiPodatke();
}

```

Slike 24, 25, 26, 27, 28– IzmenaVoznji prozor

Prozor Rezervacije omogućava pregled rezervacija i sastoje se od tabele sa podacima i dva dugmeta, jedno za brisanje rezrvacija, a drugo koje omogućava povratak na početnu stranicu PocetnaAdmin. Pomoću funkcije LoadDataGrid() se učitavaju podaci iz baze podataka u tabelu: id vožnje, naziv prevoznika, datum polaska, vreme polaska, početna, krajnja destinacija, id korisnika i ime i prezime korisnika koji su spojeni u kolonu sa nazivom putnik. Tabela Rezrvacije se sastoji od id-a vožnje iz tabele Voznja i id-a korisnika iz tabele Korisnik. To omogućava da korisnik kreira rezervaciju sa svog naloga za određenu destinaciju tj. vožnju. RezPregDataGrid_SelectionChanged() selektuje podatke id korisnika i id vožnje iz reda u tabeli koji je odabran. Potom klikom na dugme obriši briše rezervaciju tako da se vrednosti za id korisnika i id vožnje u bazi podataka i te izabrane vrednosti u tabeli poklapaju. Nakon uspešno obrisane rezrvacije prikazuje se poruka da je rezervacija uspešno obrisana.



Slike 29, 30 – Rezervacije prozor

Prozor Prevoznik omogućava tabelarni prikaz podataka u tabeli Prevoznik: naziv prevoznika i adresu prevoznika. Tabela se popunjava podacima iz tabele baze podataka pomoću funkcije LoadDataGrid(). Funkcijom PonistiUnos() se prazne kontrole za unos naziva i adrese. Klikom na dugme nazad se otvara prozor IzmenaVoznji, a klikom na dugme početna se

otvara prozor PocetnaAdmin. PrevoznikDataGrid_SelectionChanged() selektuje podatke o nazivu i adresi prevoznika i tim podacima puni kontrole za unos kako bi se omogućilo lakše brisanje i izmena podataka. Klikom na dugme dodaj se uneti podaci dodaju u tabelu Prevoznici pomoću INSERT INTO naredbe. Klikom na dugme obriši se briše onaj prevoznik čiji naziv odgovara nazivu prevoznika koji je selektovan u tabeli. Klikom na dugme izmeni adresu se menja adresa prevoznika tako što se menja adresa onom prevozniku čiji naziv odgovara nazivu koji je selektovan u tabeli. Klikom na dugme izmeni naziv se menja naziv onom prevozniku čija adresa odgovara adresi koja je selektovana u tabeli.

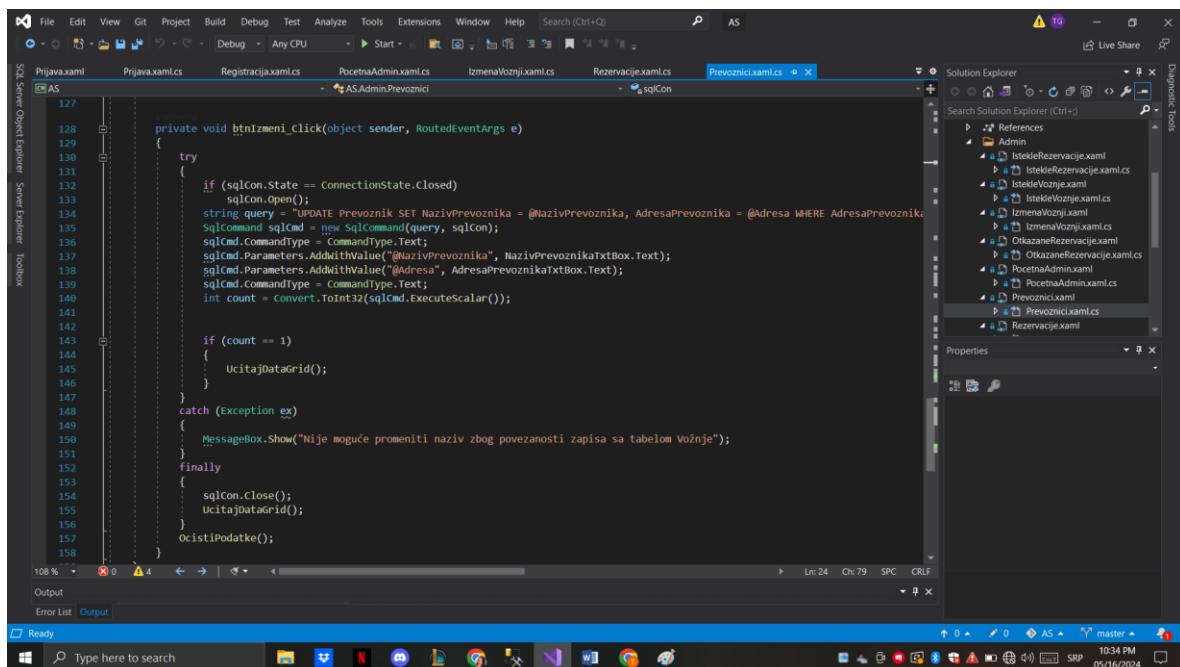
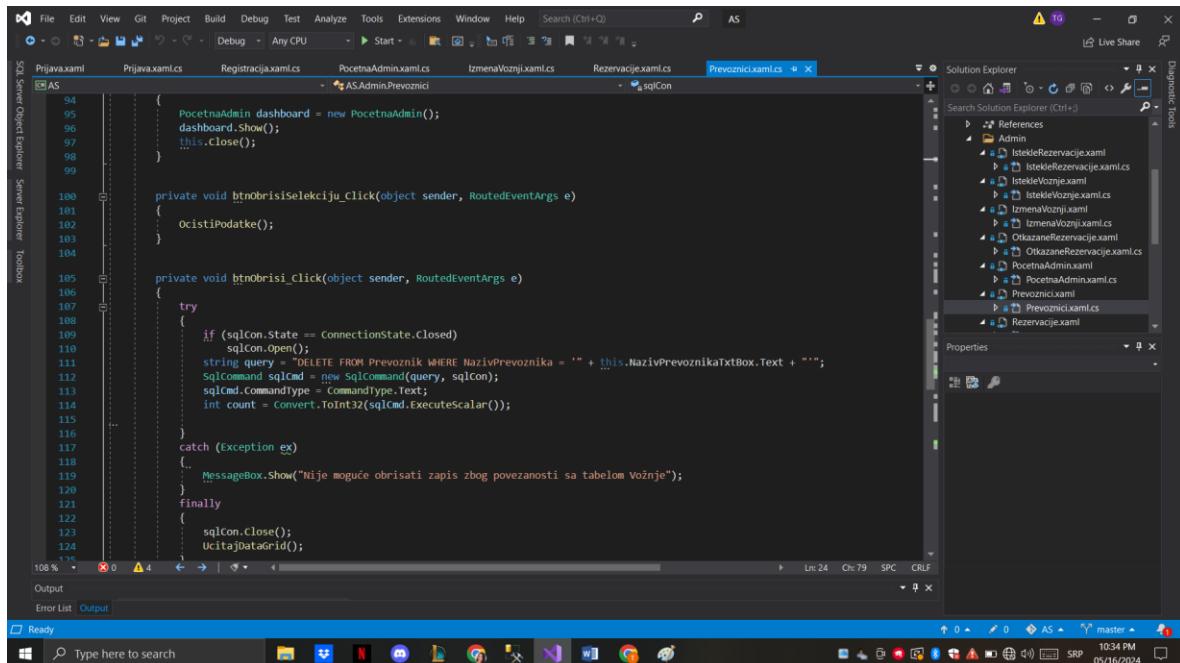
```

16  namespace AS.Admin
17  {
18      /// <summary>
19      /// Interaction logic for Prevoznici.xaml
20      /// </summary>
21      public partial class Prevoznici : Window
22      {
23          SqlConnection sqlCon = new SqlConnection(@"Data Source=DESKTOP-6TRB9F\SQLEXPRESS;Initial Catalog=AutobuskaStanica;Integrated Security=True");
24
25          1 reference
26          public Prevoznici()
27          {
28              InitializeComponent();
29              UcitajDataGrid();
30          }
31
32          5 references
33          private void UcitajDataGrid()
34          {
35              sqlCon.Open();
36              SqlCommand cmd = new SqlCommand();
37              cmd.CommandText = "SELECT * FROM Prevoznik";
38              cmd.Connection = sqlCon;
39              SqlDataAdapter dataAdapter = new SqlDataAdapter(cmd);
40              DataTable dataTable = new DataTable("Prevoznik");
41              dataAdapter.Fill(dataTable);
42              PrevoznikDataGrid.ItemsSource = new DataView(dataTable);
43              sqlCon.Close();
44          }
45      }

```

```

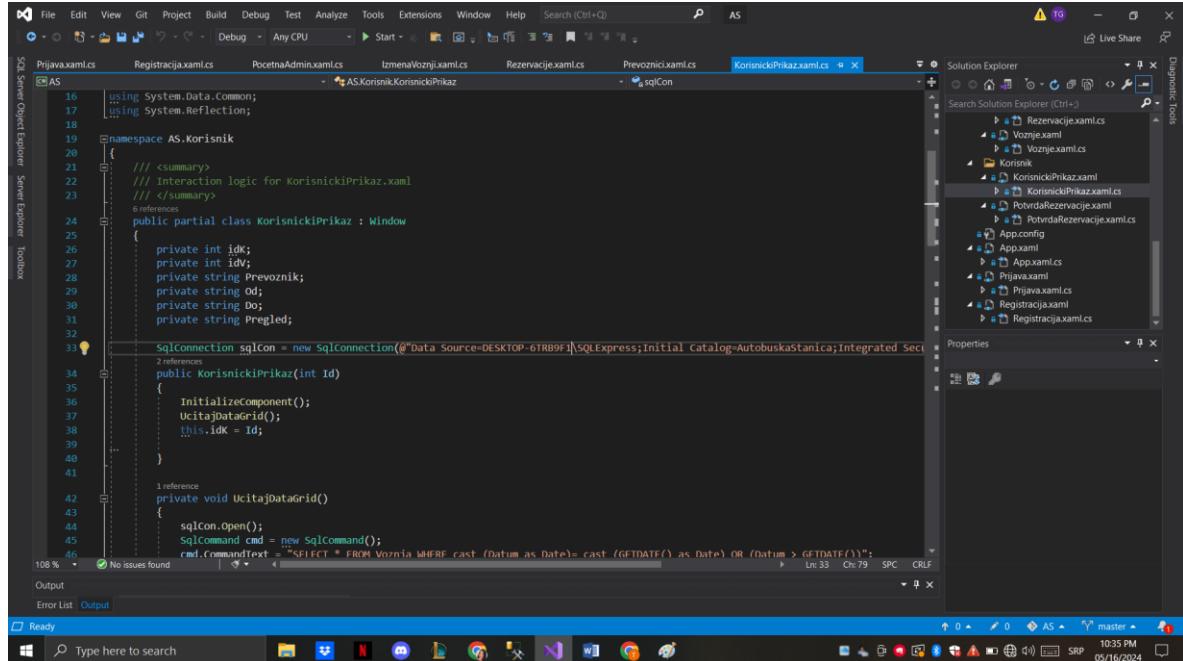
52      }
53
54      private void PrevoznikDataGrid_SelectionChanged(object sender, SelectionChangedEventArgs e)
55      {
56          DataGrid dg = sender as DataGrid;
57          DataRowView dr = dg.SelectedItem as DataRowView;
58          if (dr != null)
59          {
60              NazivPrevoznikaTextBox.Text = dr["NazivPrevoznika"].ToString();
61              AdresaPrevoznikaTextBox.Text = dr["AdresaPrevoznika"].ToString();
62          }
63      }
64
65
66      private void btnDodataj_Click(object sender, RoutedEventArgs e)
67      {
68          try
69          {
70              if (sqlCon.State == ConnectionState.Closed)
71                  sqlCon.Open(); string query = "INSERT INTO Prevoznik (NazivPrevoznika, AdresaPrevoznika) VALUES (@Naziv, @Adresa)";
72              ;
73              SqlCommand sqlCommand = new SqlCommand(query, sqlCon);
74              sqlCommand.CommandType = CommandType.Text;
75              sqlCommand.Parameters.AddWithValue("@Naziv", NazivPrevoznikaTextBox.Text);
76              sqlCommand.Parameters.AddWithValue("@Adresa", AdresaPrevoznikaTextBox.Text);
77              MessageBox.Show("Uspešno ste dodali prevoznika");
78              int count = Convert.ToInt32(sqlCmd.ExecuteScalar());
79          }
80          catch (Exception ex)
81          {
82              MessageBox.Show(ex.Message);
83          }
84      }
85  
```



Slika 31, 32, 33, 34 – Prevoznici prozor

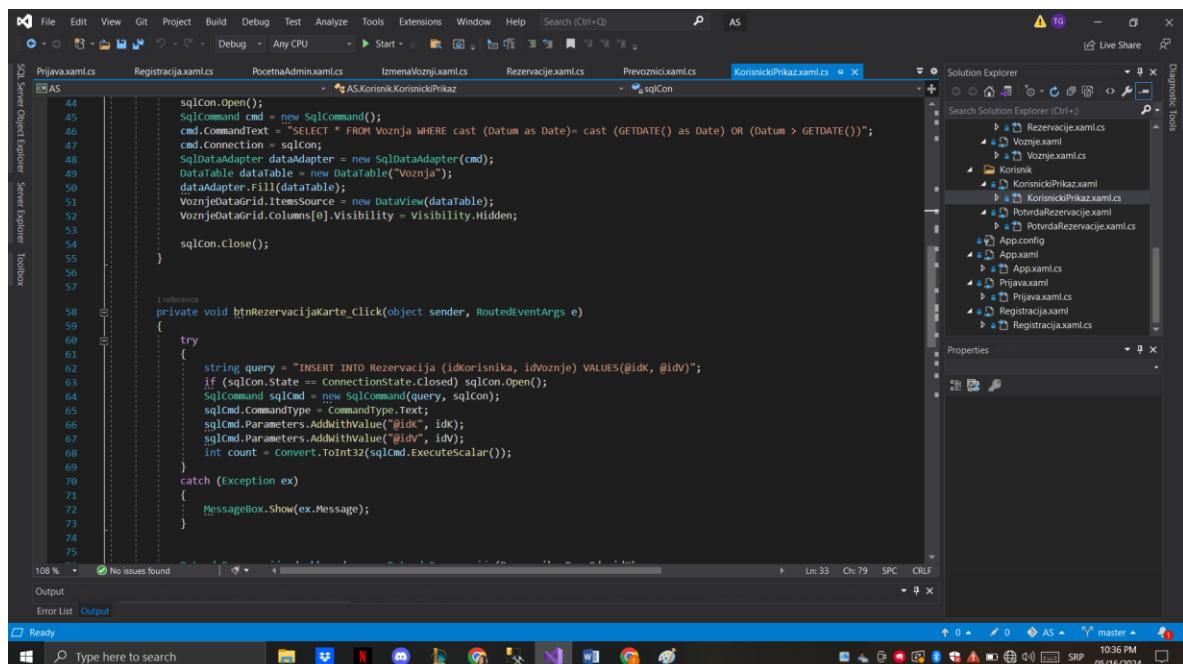
Prozor KorisnickiPrikaz jeste prozor koji se otvara kada se korisnik uspešno uloguje, tj. predstavlja korisnički pristup. Sastoji se od tabele u kojoj se nalaze podaci iz baze podataka: naziv prevoznika, datum polaska, vreme polaska, početna i krajnja destinacija. ID vožnje je sklonjen tako da se ne vidi u toj tabeli tako što je maksimalna širina kolone u kojoj se on nalazi postavljena na 0. VoznjaDataGrid_SelectionChanged() je funkcija koja omogućava da se klikom na određeni red sa podacima o vožnji prikaže naziv prevoznika,

početna i krajnja destinacija kako bi korisnik imao bolji uvid u podatke vožnje za koju želi da rezerviše kartu. Klikom na dugme rezerviši se upisuju podaci o rezervaciji u bazu podataka, a korisniku se otvara novi prozor koji predstavlja prikaz karte. Klikom na dugme za odjavu se korisnik vraća na stranicu za prijavu.



```

16  using System.Data.Common;
17  using System.Reflection;
18
19  namespace AS.Korisnik
20  {
21      /// <summary>
22      /// Interaction logic for KorisnickiPrikaz.xaml
23      /// </summary>
24      public partial class KorisnickiPrikaz : Window
25      {
26          private int idK;
27          private int idV;
28          private string Prevoznik;
29          private string od;
30          private string doj;
31          private string Pregled;
32
33          SqlConnection sqlCon = new SqlConnection(@"Data Source=DESKTOP-6TRB9E\SQLExpress;Initial Catalog=AutobuskaStanica;Integrated Security=True");
34
35          public KorisnickiPrikaz(int Id)
36          {
37              InitializeComponent();
38              UcitajDataGrid();
39              this.idK = Id;
40          }
41
42          private void UcitajDataGrid()
43          {
44              sqlCon.Open();
45              SqlCommand cmd = new SqlCommand();
46              cmd.CommandText = "SELECT * FROM Voznja WHERE cast(Datum as Date)= cast(GETDATE() as Date) OR (Datum > GETDATE())";
47              cmd.Connection = sqlCon;
48              SqlDataAdapter dataAdapter = new SqlDataAdapter(cmd);
49              DataTable dataTable = new DataTable("voznja");
50              dataAdapter.Fill(dataTable);
51              VoznjeDataGrid.ItemsSource = new DataView(dataTable);
52              VoznjeDataGrid.Columns[0].Visibility = Visibility.Hidden;
53
54              sqlCon.Close();
55          }
56
57
58          private void btnRezervacijaKarte_Click(object sender, RoutedEventArgs e)
59          {
60              try
61              {
62                  string query = "INSERT INTO Rezervacija(idkorisnika, idVoznje) VALUES(@idK, @idV)";
63                  if (sqlCon.State == ConnectionState.Closed) sqlCon.Open();
64                  SqlCommand sqlCmd = new SqlCommand(query, sqlCon);
65                  sqlCmd.CommandType = CommandType.Text;
66                  sqlCmd.Parameters.AddWithValue("@idK", idK);
67                  sqlCmd.Parameters.AddWithValue("@idV", idV);
68                  int count = Convert.ToInt32(sqlCmd.ExecuteScalar());
69              }
70              catch (Exception ex)
71              {
72                  MessageBox.Show(ex.Message);
73              }
74          }
75      }
    
```



```

44
45          sqlCon.Open();
46          SqlCommand cmd = new SqlCommand();
47          cmd.CommandText = "SELECT * FROM Voznja WHERE cast(Datum as Date)= cast(GETDATE() as Date) OR (Datum > GETDATE())";
48          cmd.Connection = sqlCon;
49          SqlDataAdapter dataAdapter = new SqlDataAdapter(cmd);
50          DataTable dataTable = new DataTable("voznja");
51          dataAdapter.Fill(dataTable);
52          VoznjeDataGrid.ItemsSource = new DataView(dataTable);
53          VoznjeDataGrid.Columns[0].Visibility = Visibility.Hidden;
54
55          sqlCon.Close();
56
57
58          private void btnRezervacijaKarte_Click(object sender, RoutedEventArgs e)
59          {
60              try
61              {
62                  string query = "INSERT INTO Rezervacija(idkorisnika, idVoznje) VALUES(@idK, @idV)";
63                  if (sqlCon.State == ConnectionState.Closed) sqlCon.Open();
64                  SqlCommand sqlCmd = new SqlCommand(query, sqlCon);
65                  sqlCmd.CommandType = CommandType.Text;
66                  sqlCmd.Parameters.AddWithValue("@idK", idK);
67                  sqlCmd.Parameters.AddWithValue("@idV", idV);
68                  int count = Convert.ToInt32(sqlCmd.ExecuteScalar());
69              }
70              catch (Exception ex)
71              {
72                  MessageBox.Show(ex.Message);
73              }
74          }
75      }
    
```

```
AS.KorisnickiPrikaz.xaml.cs
70     catch (Exception ex)
71     {
72         MessageBox.Show(ex.Message);
73     }
74
75     PotvrdaRezervacije dashboard = new PotvrdaRezervacije(Prevoznik, Do, Od, idk);
76     dashboard.Show();
77     this.Close();
78 }
79
80
81     private void btnOdjava_Click(object sender, RoutedEventArgs e)
82     {
83         Prijava dashboard = new Prijava();
84         dashboard.Show();
85         this.Close();
86     }
87
88     private void VoznjeDataGrid_SelectionChanged(object sender, SelectionChangedEventArgs e)
89     {
90         DataGrid dg = sender as DataGrid;
91         DataRowView dr = dg.SelectedItem as DataRowView;
92         if (dr != null)
93         {
94             idv = int.Parse(dr["idvozanje"].ToString());
95             Prevoznik = dr["NazivPrevoznika"].ToString();
96             Od = dr["MestoPolaska"].ToString();
97             Do = dr["MestoDoLaska"].ToString();
98         }
99
100
101
102
103 }
```

No issues found

Output

Error List Output

Ready

Type here to search

Solution Explorer

Properties

Diagnostic tools

Slike 35, 36, 37 – KorisnickiPrikaz prozor

Prozor PotvrdaRezervacije predstavlja prozor u kome se korisniku prikazuje naziv prevoznika i relacija za koju je rezervisao kartu. Ovaj prozor predstavlja potvrdu korisniku da je uspešno rezervisao kartu. Klikom na dugme kraj se korisnik vraća na prozor KorisnickiPrikaz odakle može da se odjavi.

```
21     {
22         public string Prevoznik, Od, Do;
23         public int idK;
24         public string Voznja;
25
26         private void btnNazad_Click(object sender, RoutedEventArgs e)
27         {
28             KorisnickiPrikaz dashboard = new KorisnickiPrikaz(idK);
29             dashboard.Show();
30             this.Close();
31         }
32
33         public PotvrdaRezervacije(string Prevoznik, string Od, string Do, int idK)
34         {
35             InitializeComponent();
36             this.Prevoznik = Prevoznik;
37             this.Od = Od;
38             this.Do = Do;
39             this.idK = idK;
40
41             Pregled();
42         }
43
44         private void Pregled()
45         {
46             Voznja = Do + " -> " + Od;
47
48             PrevoznikTextBox.Text = Prevoznik.ToString();
49             VoznjaTextBox.Text = Voznja.ToString();
50         }
51     }
```

Slika38 – PotvrdaRezervacije prozor

6. Korišćeni alati i softveri

Svi alati navedeni u tabeli 4, kao deo projektne ideje, korišćeni su i u finalnoj izradi softvera, po planu.

Tabela 4 – Alati

Sloj na kom se alat ili softver koristi	Alati
Sloj podataka	MSSMS sistem za rukovanje bazom podataka
	SQL server Management Studio za administraciju baze podataka
Aplikacioni sloj	Programski jezik C#, razvojno okruženje “Visual Studio“
Prezentacioni sloj	razvojno okruženje “Visual Studio“

7. Lista tabela, slika i listinga

7.1. Lista tabela

TABELA 1 – SPISAK ZAHTEVA KORISNIKA	4
TABELA 2 – SPECIFIKACIJA ZAHTEVA KORISNIKA	4
TABELA 3 – PROJEKTNE IDEJE I NJIHOVI OPISI	7
TABELA 4 – ALATI	35

7.2. Lista slika

SLIKA 1 – PRIJAVA	
8	
SLIKA 2 – GREŠKA LOGIN	8
9	
SLIKA 3 – REGISTRACIJA	
9	
SLIKA 4 – GREŠKA REGISTRACIJA	
9	
SLIKA 5 – USPEŠNA REGISTRACIJA	
10	
SLIKA 6 – POČETNA STRANA	
10	
SLIKA 7 – UPRAVLJANJE VOŽNJAMA	11
SLIKA 8 – PRIKAZ PODATAKA	
12	
SLIKA 9 – PREGLED VOŽNJI	
13	
SLIKA 10 – REZERVACIJE ADMIN	
14	
SLIKA 11 – REZERVACIJA KORISNIK	
15	
SLIKA 12 – KARTA	
15	
SLIKA 13 - DIJAGRAM BAZE PODATAKA	19
SLIKA 14 – LISTA KREIRANIH GRANA NA GITHUB-U	20
SLIKA 15, 16, 17, 18 – PRIJAVA PROZOR	
21 - 23	
SLIKA 19, 20, 21 – REGISTRACIJA PROZOR	24 - 25
SLIKA 22, 23 – POCETNA ADMIN PROZOR	26
SLIKA 24, 25, 26, 27, 28, 29 – IZMENA VOZNJI PROZOR	27
– 30	
SLIKA 30, 31 – REZERVACIJE PROZOR	
31	
SLIKA 32, 33, 34 – PREVOZNICI PROZOR	32 - 33
SLIKA 35, 36, 37 – KORISNICKI PRIKAZ PROZOR	34 - 35
SLIKA 38 – POTVRDAREZERVACIJE PROZOR	36

8. Literatura

- [1] Dokumentacija za akreditaciju studijskog programa: Informacione tehnologije – softversko inženjerstvo. Tehnički fakultet „Mihajlo Pupin“. Zrenjanin. 2015.
<http://tfzr.rs/Content/files/0/IT%20softversko%20inzenjerstvo%20osnovne%20studije.pdf>
- [2] S. Atanasijević. Životni ciklus razvoja softvera. Univerzitet u Kragujevcu. Kragujevac. 2022.
https://www.researchgate.net/publication/362001817_ZIVOTNI_CIKLUS_RAZVOJA_SOFTVERA
- [3] M. Preston. System Development Life Cycle. 2023.
<https://www.clouddefense.ai/blog/system-development-life-cycle#:~:text=What%20Are%20the%207%20Phases,testing%2C%20implementation%2C%20and%20maintenance.>