

PANEVROPSKI UNIVERZITET APEIRON
FAKULTET INFORMACIONIH TEHNOLOGIJA
BANJA LUKA

Seminarski rad

PROGRAM ZA VOĐENJE RENT-A-CAR AGENCIJE

Nastavni predmet: *Objektno-orjentisano programiranje*

Predmetni nastavnik:

Saša Salapura

Student:

Siniša Božić

192-20/RITP

Banja Luka, 2021.

SADRŽAJ

UVOD	1
1 PREZENTACIJA PROGRAMA	2
2 STRUKTURA PROGRAMA	6
3 STRUKTURA BAZE PODATAKA	7
4 KONEKCIJA NA BAZU PODATAKA I UPITI	9
5 KORIŠTENI METODI	10
ZAKLJUČAK	12
LITERATURA	13

UVOD

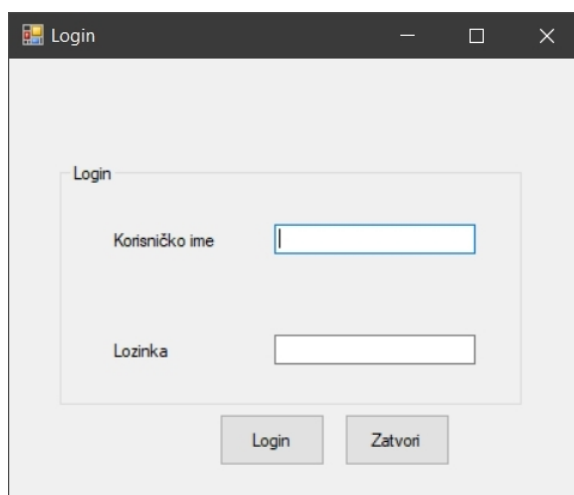
U predmetnom seminarskom radu je prezentovan program koji može da služi za evidenciju rada rent-a-car agencije, odnosno iznajmljivanje vozila. Program je izrađen u C# koristeći Visual Studio i sastoji se od tehnologije Windows Forms, odnosno korisnički interfejs je vizuelno izrađen i elementi interfejsa su programirani na određeni način. Inače, prirodan odabir za aplikaciju sa korisničkim interfejsom je C# budući da je to nativni programski jezik Windows operativnom sistemu¹. Sam programski jezik C# je novijeg datuma, nastao nakon Java programskog jezika i vrlo mu je sličan. Program je objektno-orjentisanog tipa, sa napomenom da je takva organizacija programskog koda rezultat korištenja razvojnog okruženja. Napredne OOP tehnike kao što su enkapsulacija i polimorfizam nisu korištene već je programski kod unutar klasa organizovan u metodima. Metodi su void tipa, odnosno implicitno ne vraćaju vrijednost i koriste se za predviđeno funkcionisanje grafičkog interfejsa odnosno za prikaz i manipulaciju podataka u njemu.

Program omogućava evidentiranje svih segmenata funkcionisanja rent-a-car agencije i jedino bi mu eventualno trebalo dodati mogućnost eksporta podataka ili štampanja da bi mogao zadovoljiti potrebe stvarne agencije koja iznajmljuje vozila.

¹ Podržan je i u drugim operativnim sistemima.

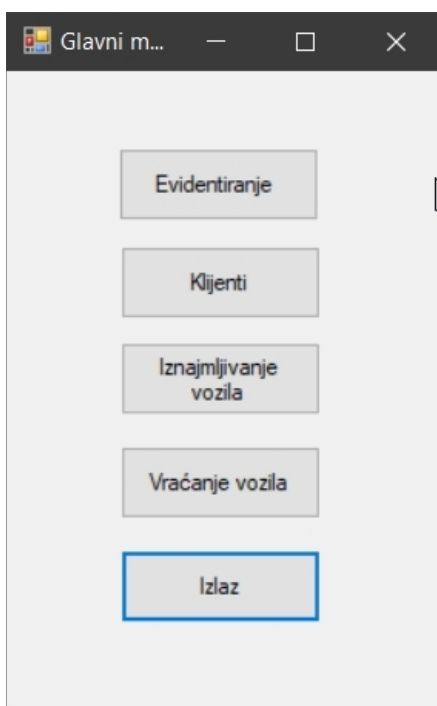
1 PREZENTACIJA PROGRAMA

Program je izrađen u C# programskom jeziku, uz korištenje Visual Studio 2019 razvojnog programskog okruženja. Program zapravo služi kao front-end za bazu podataka, jer manipuliše sa SQL upitima koji manipulišu podacima na nivou baze podataka. Kao baza korištena je MS SQL.



Prilog 1: Početni izgled

Program počinje sa prikazom Login stranice, gdje se koristi korisničko ime i lozinka definisana u programskom kodu. Ukoliko se unesu pogrešni podaci, program odbija dalji pristup.



Prilog 2: Glavni meni

Glavni meni programa se sastoji od nekoliko pod-menija kojima se program razdvaja na segmente koji obrađuju ulazne podatke na predviđeni način, a to je:

- Evidentiranje:
Evidencija podataka o vozilima kojima raspolaže agencija.
- Klijenti:
Evidencija klijenata agencije.
- Iznajmljivanje vozila:
Evidencija iznajmljenih vozila.
- Vraćanje vozila:
Evidencija povrata vozila.

The screenshot shows a window titled "Evidentiranje". On the left, there is a form titled "Evidencija vozila" with the following fields: "Broj registracije" (containing "00003"), "Proizvođač" (empty), "Model" (empty), and "Dostupno" (a dropdown menu with "Da" selected). Below the form are two buttons: "Dodaj" and "Odustani". On the right, there is a table with the following data:

	Reg.br	Brend	Model	Dostup	Izmijeni	Izbriši
▶	0	Audi	A3	No	Izmijeni	Izbriši
	1	Lada	Samara	Da	Izmijeni	Izbriši
	2	Kia	Saphire	Da	Izmijeni	Izbriši

Below the table is a large grey rectangular area. At the bottom center of the window is a button labeled "Ažuriraj".

Prilog 3: Evidentiranje vozila

Kod evidencije vozila, korisnik unosi podatke o vozilu: svako vozilo ima svoj registracioni broj, potom podatke o proizvođaču, nazivu modela i padajući meni kojim se može odabrati da li je vozilo dostupno ili ne (na primjer, vozilo može biti u vlasništvu rent-a-car agencije ali ne mora nužno biti dostupno za iznajmljivanje, može biti na servisu). Na desnoj strani se tabelarno prikazuju uneseni podaci koji se mogu mijenjati po želji.

Klijenti

Podaci o klijentu

Klijent ID: 00004

Naziv klijenta:

Adresa:

Broj telefona:

Dodaj Odustani Ažuriraj

	Klijent ID	Naziv kl.	Adresa	Telefon	Izmijeni	Izbriši
▶	1	Mark	Boston	2323232	Izmijeni	Izbriši
	2	Dino	Fridrichsha...	234234	Izmijeni	Izbriši
	3	Tony	Bergamo	2345345	Izmijeni	Izbriši

Prilog 4: Evidentiranje klijenata

Evidentiranje klijenata je slično kao evidencija vozila. Tu su detaljni podaci o klijentima agencije i istovjetno se tabelarno prikazuju.

Iznajmljivanje

Iznajmljivanje

ID vozila: 2

ID klijenta: 3

Naziv klijenta: Tony

Naknada: 120

Datum početka: subota , 18. decembar 202

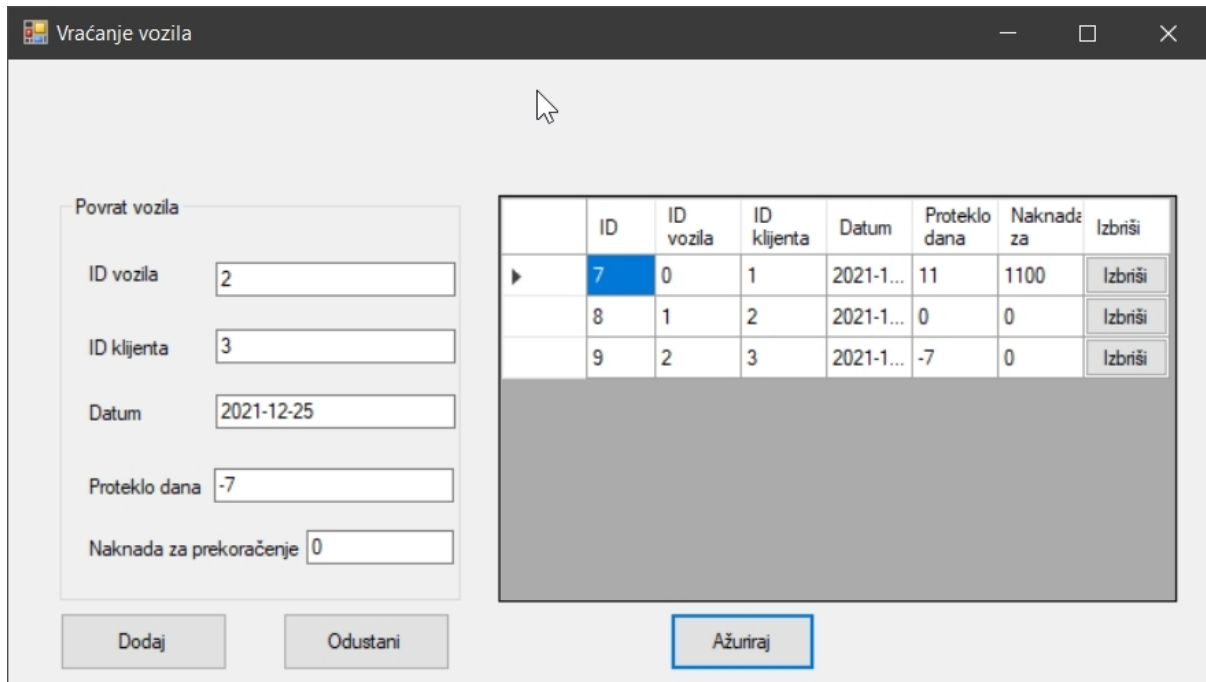
Krajnji datum: subota , 25. decembar 202

Dodaj Dostupno Da Ažuriraj

	ID	ID vozil	ID klijer	Ime klijenta	Naknada	Početni datum	Krajnji datum
▶	16	0	1	Mark	100	2021-12-01	2021-1
	17	1	2	Dino	80	2021-12-16	2021-1
	19	2	3	Tony	120	2021-12-18	2021-1

Prilog 5: Registrovanje iznajmljivanja

Dio programa koji evidentira iznajmljivanje je nešto složeniji jer kombinuje podatke iz više tabela. Tako korisnik može da izabere dato vozilo iznajmljeno datom klijentu, da odredi naknadu po danu i vremenski početak i predviđeni kraj iznajmljivanja. Tako definisano iznajmljivanje se evidentira tabelarno sa desne strane.



Vraćanje vozila

Povrat vozila

ID vozila:

ID klijenta:

Datum:

Proteklo dana:

Naknada za prekoračenje:

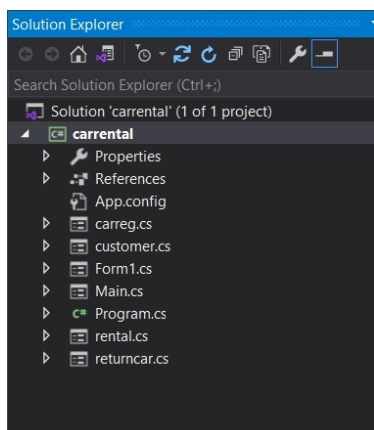
	ID	ID vozila	ID klijenta	Datum	Proteklo dana	Naknada za	Izbriši
▶	7	0	1	2021-1...	11	1100	<input type="button" value="Izbriši"/>
	8	1	2	2021-1...	0	0	<input type="button" value="Izbriši"/>
	9	2	3	2021-1...	-7	0	<input type="button" value="Izbriši"/>

Prilog 6: Registrovanje vraćanja vozila

Segment vraćanja vozila obuhvata podatke iz prethodnih tabela, te na osnovu identifikacionih brojeva vozila i klijenata shodno trenutnom datumu evidentira vraćanje vozila, te obračunava eventualnu naknadu za prekoračenje. Na prilogu 6. vidimo da je protekli broj dana iznajmljivanja negativan, što je zato jer je klijent vratio vozilo prije predviđenog roka vraćanja, što se takođe može desiti.

2 STRUKTURA PROGRAMA

U C#, program je organizovan unutar namespacea (može ih biti i više a mogu biti i hijerarhijski organizovani). Tako je predmetni program sadržan unutar namespace imena carrental, a obuhvata nekoliko klasa koji su zasebni fajlovi. Svaki od tih fajlova je linkovan u glavnom meniju (pogledati prilog br. 2), odnosno programski kod je podijeljen na više funkcionalnih dijelova, koji međusobno komuniciraju, što je jedna od odlika objektno-orijentisanog programiranja. Svaka od klasa je zasebno programirana, ali se sastoji od vrlo sličnog koda koji je jednom napisan a korišten više puta, sa eventualnim prilagođavanjima.



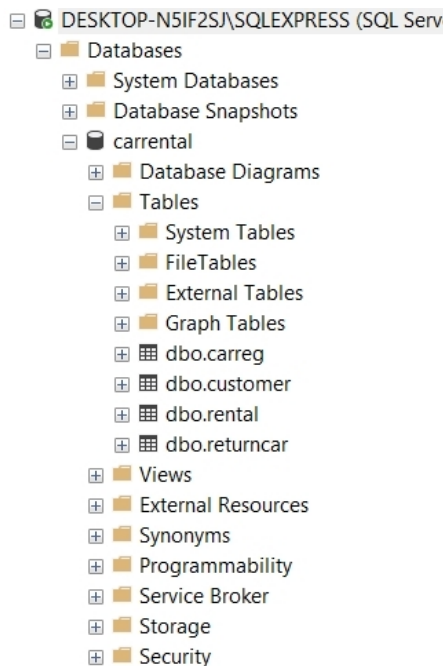
Prilog 7: Struktura programa i pripadajući fajlovi

```
10
11 namespace carrental
12 {
13     4 references
14     public partial class Main : Form
15     {
16         1 reference
17         public Main()
18         {
19             InitializeComponent();
20         }
21
22         1 reference
23         private void button1_Click(object sender, EventArgs e)
24         {
25             carreg c = new carreg();
26             c.Show();
27         }
28
29         1 reference
30         private void button2_Click(object sender, EventArgs e)
31         {
32             customer c = new customer();
33             c.Show();
34         }
35
36         1 reference
37         private void button3_Click(object sender, EventArgs e)
38         {
39             rental r = new rental();
40             r.Show();
41         }
42
43         1 reference
44         private void button4_Click(object sender, EventArgs e)
45         {
46             returncar r=new returncar();
47             r.Show();
48         }
49
50         1 reference
51         private void button5_Click(object sender, EventArgs e)
52         {
53             this.Close();
54         }
55     }
56 }
```

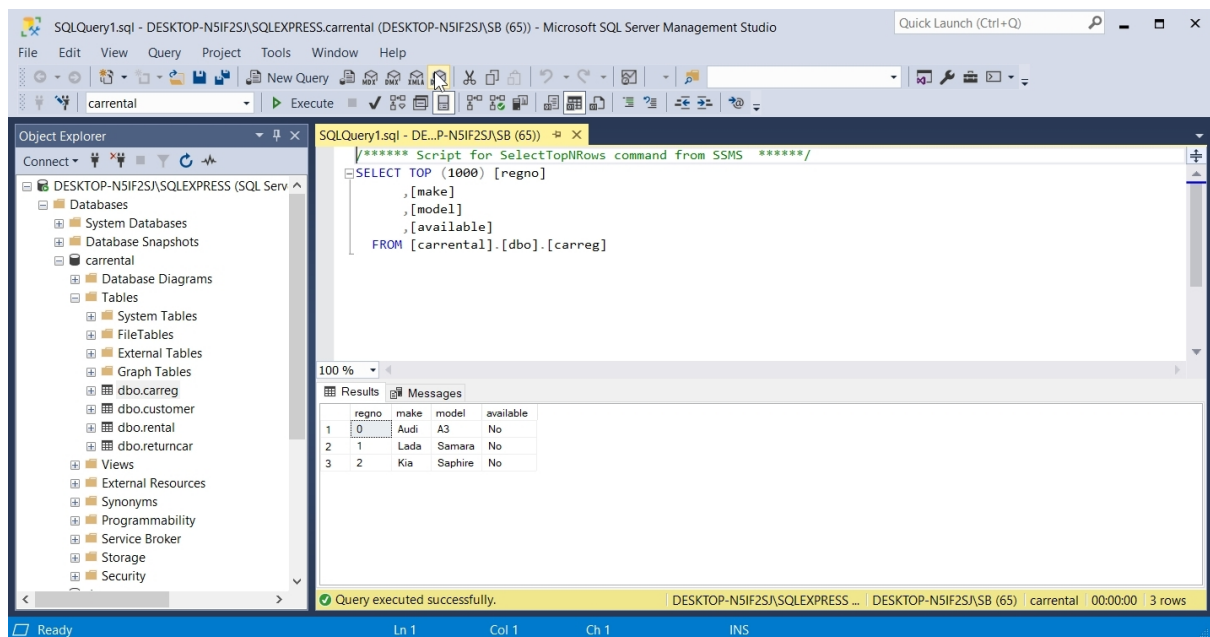
Prilog 8: Programski kod glavnog menija

3 STRUKTURA BAZE PODATAKA

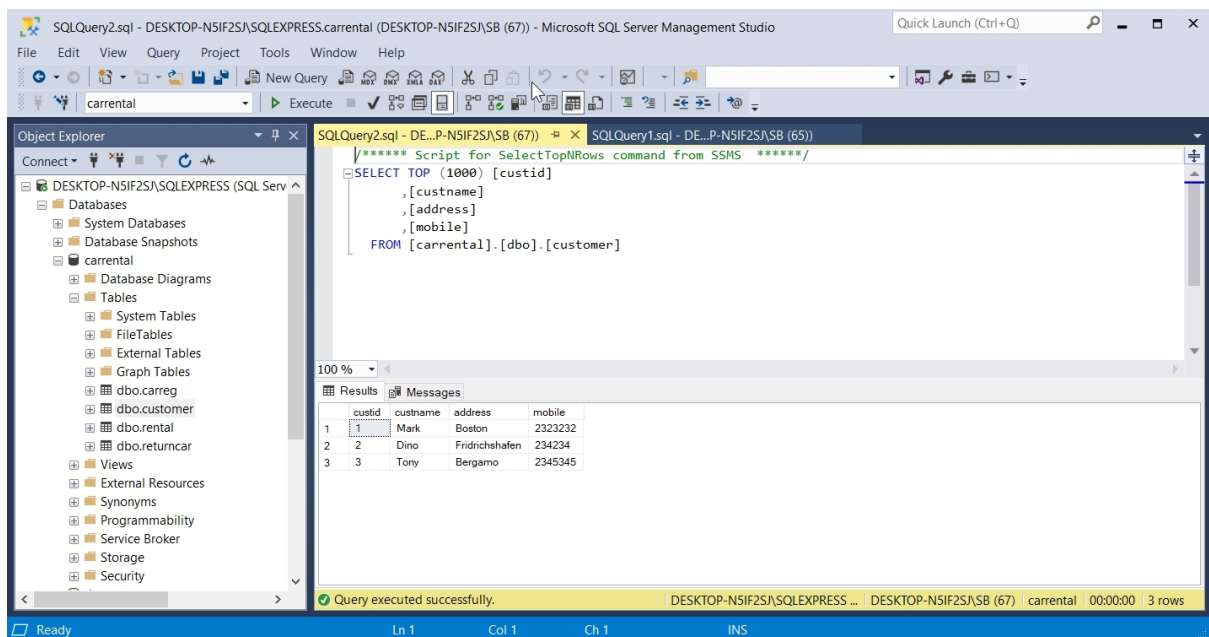
U MS SQL je kreirana baza podataka sa četiri pripadajuće tabelle.



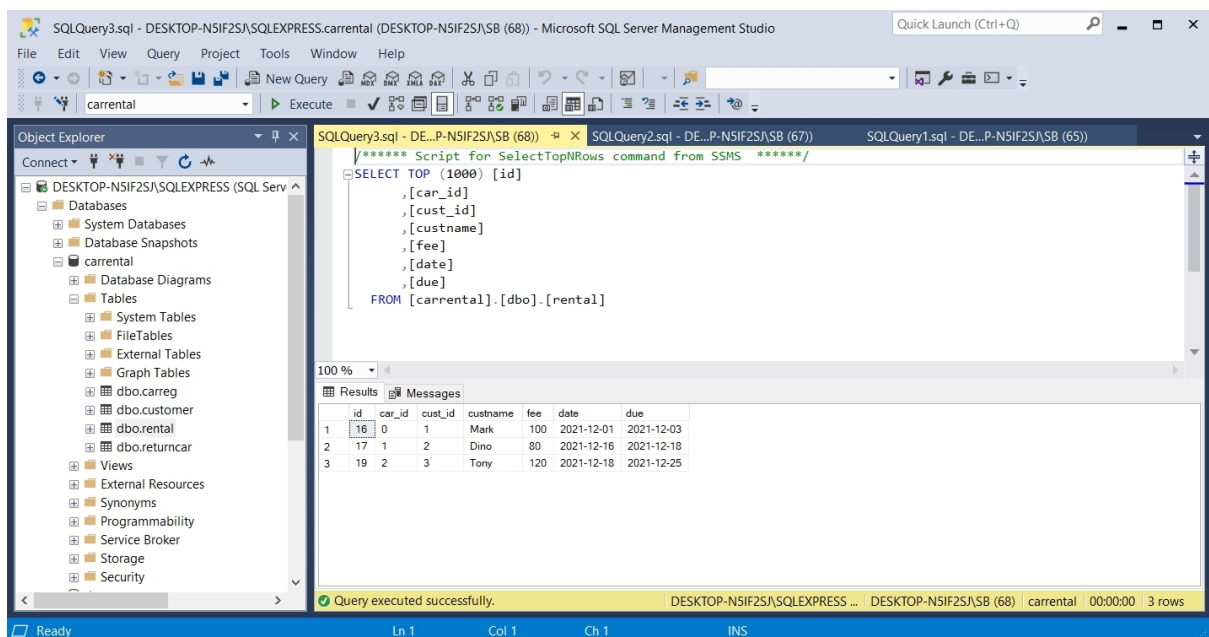
Prilog 9: Struktura baze podataka



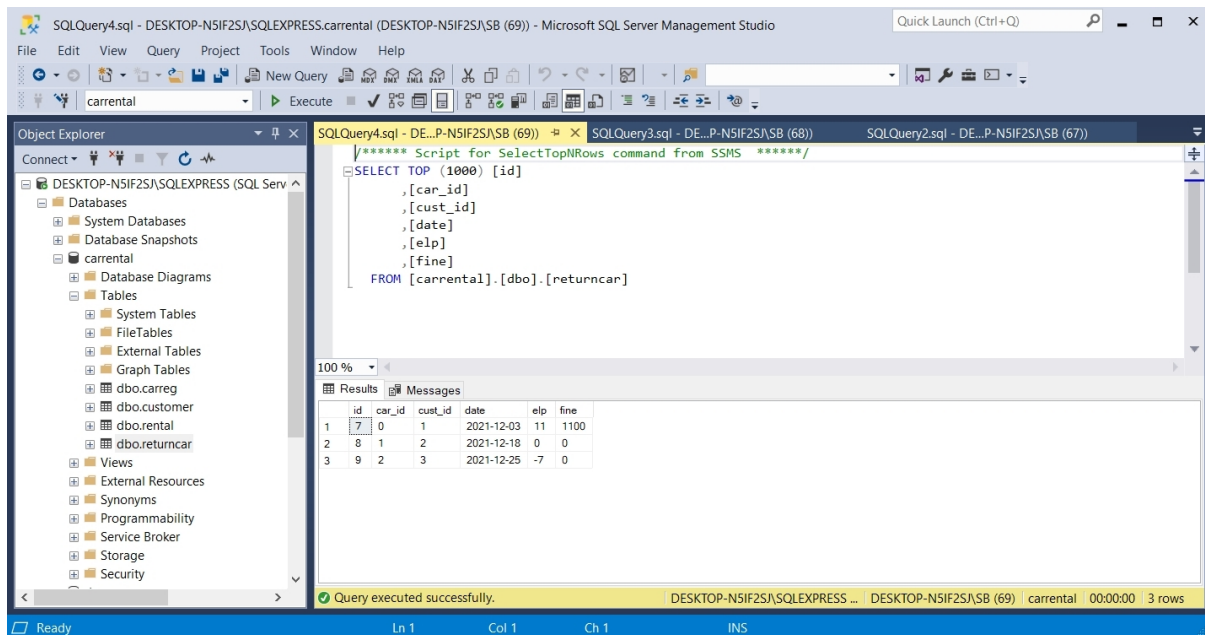
Prilog 10: Izgled tabelle gdje se evidentiraju vozila



Prilog 11: Izgled tabele gdje se registruju klijenti



Prilog 12: Izgled tabele gdje se evidentira iznajmljivanje



Prilog 13: Izgled tabele gdje se registruje vraćanje vozila

4 KONEKCIJA NA BAZU I UPITI

Veza ka bazi podataka se uspostavlja korištenjem klase `SqlConnection`:

```
using System.Data.SqlClient;

SqlConnection con = new SqlConnection("Data Source=DESKTOP-N5IF2SJ\\
SQLEXPRESS; Initial Catalog=carrental; User ID=testadmin;
Password=123;");

SqlCommand cmd;

SqlDataReader dr;
```

Potrebno je deklarirati nekoliko varijabli, među njima i varijablu koja vodi do date baze i sadrži parametre kako joj pristupiti. Ostali kod je standardni dio klase `SqlClient`. Baza podataka se pristupa komandama koje su inicijalizovane u varijabli. U zavisnosti od toga da li trebamo dobiti podatke iz baze, ili ih modifikovati, koristimo različite upite:

```
sql = "select regno from carreg order by regno desc";

sql = "insert into carreg(regno, make, model, available)
values(@regno, @make, @model, @available)";

sql = "update carreg setmake=@make,model=@model,available=
@available where regno=@regno";
```

Potom slijedi standardizovan kod koji SQL upitu pridružuje putanju ka bazi podataka, kako bi se upit mogao izvršiti. Konekcija se otvara a po izvršavanju zatvara.

```
cmd = new SqlCommand(sql, con); ;
con.Open();
dr = cmd.ExecuteReader();
con.Close();
```

Naravno, ove podatke mijenjamo u skladu sa tim kojoj tabeli pristupamo, koje podatke (redove ili kolone) želimo da dobijemo ili promijenimo.

5 KORIŠTENI METODI

Klase su organizovane korištenjem metoda koji grupišu kod koji ima određenu svrhu. Na primjer, izvršni dio klase `carreg` izgleda ovako:

```
public carreg()
{
    InitializeComponent();
    Autono();
    load();
}
```

Metod `InitializeComponent()` je automatski dodan od strane razvojnog okruženja, dok su metodi `Autono()` i `load()` rezultat korisničkog unosa. Metod `Autono()` služi za inkrementalno povećavanje broja pod kojim se registruju vozila u odjeljku Evidencija, dok metod `load()` učitava podatke iz baze podataka u tabelarnom dijelu na desnoj strani prozora. Ovi metodi su prisutni u svim klasama i prilagođeni su podacima kojima barata određena klasa.

```
public void Autono()
{
    sql = "select regno from carreg order by regno desc";
    cmd = new SqlCommand(sql, con); ;
    con.Open();
    dr = cmd.ExecuteReader();

    if(dr.Read())
    {
        int id = int.Parse(dr[0].ToString())+1;
        proid = id.ToString("00000");
    }
    else if (Convert.IsDBNull(dr))
    {

```

```

        proid = ("00000");
    }
    else
    {
        proid = ("00000");
    }

    txtregno.Text = proid.ToString();
    con.Close();
}

public void load()
{
    sql = "select * from carreg";
    cmd = new SqlCommand(sql, con);
    con.Open();
    dr = cmd.ExecuteReader();
    dataGridView1.Rows.Clear();

    while(dr.Read())
    {
        dataGridView1.Rows.Add(dr[0], dr[1], dr[2], dr[3]);
    }
    con.Close();
}

```

ZAKLJUČAK

U predmetnom radu prikazano je funkcionisanje programa koji evidentira sve bitne aspekte sa kojima se u svakodnevnom radu jedna rent-a-car agencija može susresti. Sve se dešava u bazi podataka a putem front-end Windows Forms aplikacije korisnik jednostavno manipuliše sa unosom i izmjenama podataka. Programski kod se sastoji od 7 klasa odnosno CS fajlova, koji su međusobno povezani i pristupaju bazi podataka. Program jednostavnim SQL upitima manipuliše podacima a korisnik ima mogućnosti evidentiranja vozila, klijenata, praćenje iznajmljivanja i povrata vozila te obračun naknade.

LITERATURA

- [1] Einstieg in C# mit Visual Studio 2019: Ideal für Programmieranfänger, Thomas Theis, Rheinwerk Computing, 2019
- [2] Murach's C#, Anne Boehm, Mike Murrach & Assoc. Inc., 2016
- [3] SqlConnection Class
<https://docs.microsoft.com/en-us/dotnet/api/system.data.sqlclient.sqlconnection?view=dotnet-plat-ext-6.0>, pristupano 18.12.2021
- [4] SQL Tutorial, <https://www.w3schools.com/sql/default.asp>, pristupano 18.12.2021
- [5] The C# Programming Yellow Book, Rob Miles, Independently published, 2018
- [6] User Interfaces in C#: Windows Forms and Custom Controls, Matthew MacDonald, Apress, 2002