# Dokumentacija

Tehnologija korišćena u izradi klijentske strane ovog softvera, WPF(Windows Presentation Foundation) sa primenom MVVM(Model View ViewModel ) obrasca. Serverski deo je razvijen u WCF(Windows Communication Foundation) tehnologiji i oslanja se na Entity Framework za rad sa bazom podataka.

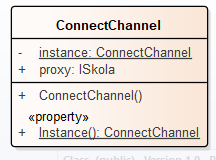
Korišćeno je više vrsta programskih obrazaca. Korišćeni su sledeći obrasci:

* **Singleton**
* **Prototype**
* **Factory**
* **Observer**
* **Command**
* **Proxy**

# Upotrebljeni obrasci

Singleton ili unikat obrazac obezbeđuje da klasa ima samo jednu instancu i omogućava globalni pristup toj instanci.

Upotrebom singletona memorija je bolje iskorišćena. Mane: sakriva zavisnosti, sama kontroliše svoj životni ciklus.



U projektu obrazac singleton je upotrebljen prilikom instanciranja objekta klase Konekcija koje vrše povezivanje između klijenta i servera.

Prototype ili prototip obrazac olakšava kloniranje postojeće

instance odrađenog objekta i sakriva njegovu kompleksnost. Mana: teže kloniranje kružnih referenci.

U projektu je iskorišćen radi dupliranja instance klase AutoSkola. Pritiskom na dugme “Dupliraj” instanca klase se duplira.

Factory ili fabrički metod se koristi kada želimo da prepustimo odgovornost kreiranja objekta nekoj podklasi.

Factory klasa definiše interfejs za kreiranje objekata, a sama implementacija kreiranja objekta je prepuštena podklasama.

U ovom projektu komunikacija između klijenta i servera se obavlja WFC tehnologijom koja ima ugrađen ovaj obrazac.

Proxy ili zastupnik obrazac omogućava da zamenski objekat upravlja pristupom stvarnom subjektu. WCF tehnologija se oslanja takođe i na ovaj obrazac.



Command ili komanda obrazac enkapsulira zahtev za

izvršenjem određene operacije u jedan objekat. Konkretne klase koje implementiraju komande obično imaju mnogo zajedničkih osobina koje se lokalizuju u osnovnu apstraktnu klasu.U projektu, obrazac je realizovan pomoću interfejsa

ICommand kog nasljeđuje svaka klasa koja je vezana za komandu. Nasljeđivanjem interfejsa implemetiraju se dve metode (CanExecute i Execute) i jedan događaj (CanExecuteChanged).

CanExecute metoda definiše da li akcija može da se izvrši. Execute metoda predstavlja samu akciju.

Observer ili posmatrač obrazac omogućava da se promena sadržaja u jednom elementu odmah pojavi i u svim delovima programa koji dati element prikazuju u nekom obliku.

MVVM uključuje ovaj obrazac. On je takođe realizovan pomoću interfejsa INotifyPropertyChanged

(System.ComponentModel). Nasleđuju ga ViewModel klase i implementiraju PropertyChangedEventHandler koji nosi naziv PropertyChange. Obezbeđuje se da prilikom promene vrednosti nekog polja ovaj event obavesti View da je došlo do promena kako bi se on ažurirao.



**Opis aplikacije**

Klijent-server aplikacija za opis rada auto skole, koja se sastoji iz tri dela :

1. **Console application (server)**
2. **Windows Presentation Foundation (klijent)**
3. **Class library (biblioteka)**

Serverski deo je razvijen u WCF (Windows Communication Presentation) tehnologiji i pruža klijentu usluge rada nad bazom podataka. Za realizaciju baze podataka korišćen je Entity Framework. Svaka tabela u bazi podataka predstavlja

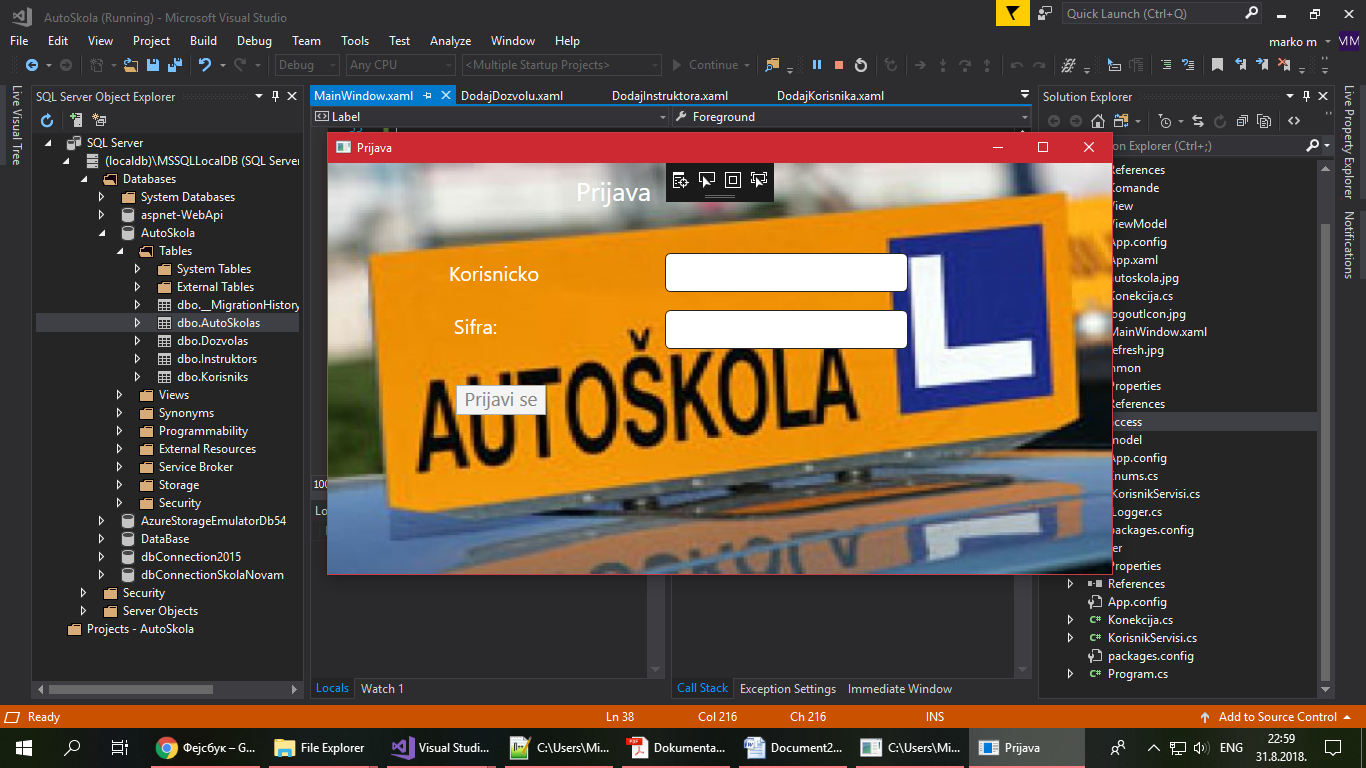
entitet koji odgovara C# klasi čiji atributi predstavljaju kolonu u tabeli. Za potrebe aplikacije kreirane su 4 tabele.

Postoje dva tipa korisnika koje je potrebno razlikovati, a to su administrator i korisnik. Razlika između njih je ta što administrator ima mogućnost izvršavanja svih mogućih akcija, dok je korisniku uskraćeno dodavanje novih korisnika.

Klijentski deo aplikacije je razvijen u WPF tehnologiji korišćenjem MVVM (Model View ViewModel) obrasca.

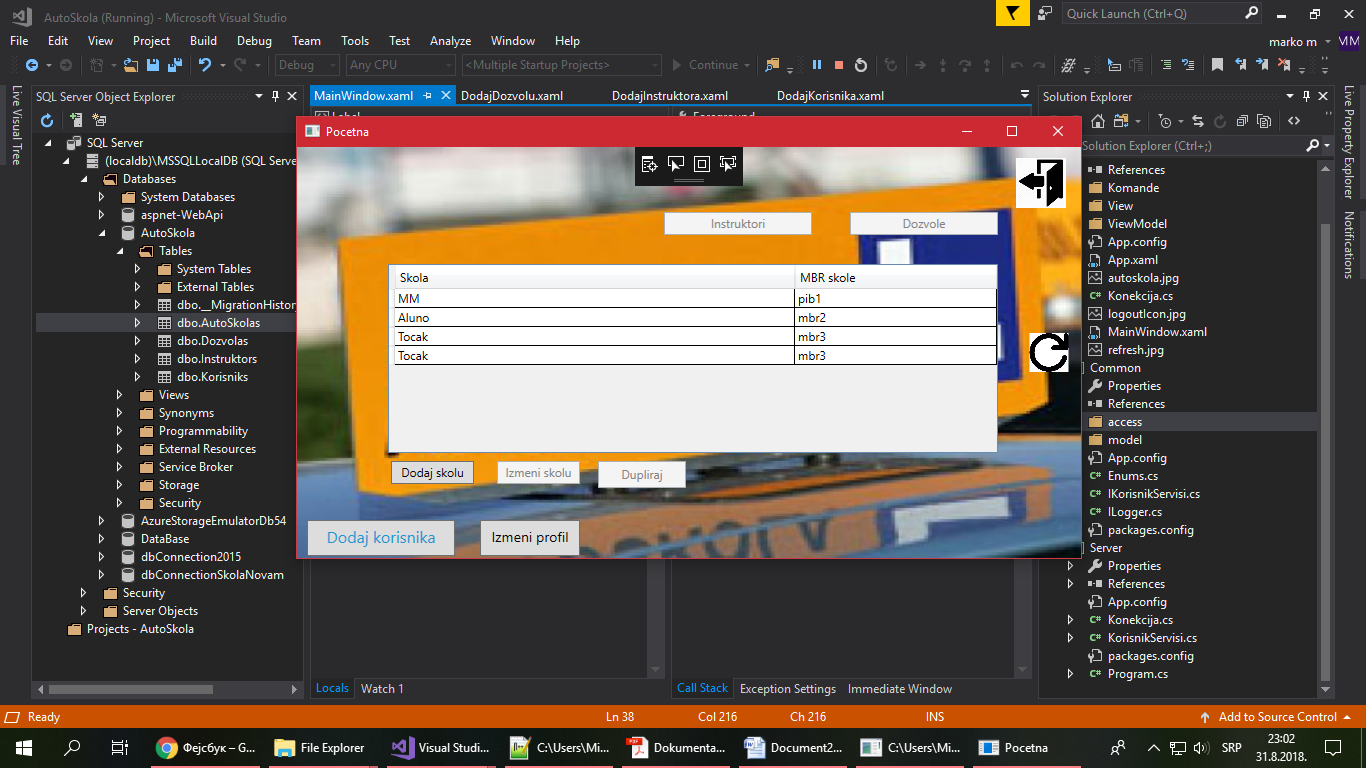
Pri pokretanju aplikacije, korisniku se prikazuje prozor sa formom za logovanje i dugmetom Uloguj se. Ukoliko su

unešeni podaci ispravni, šifra odgovarajuća a korisničko ime već postoji u bazi, prikazuje se glavni prozor u kome se izvršavaju sve dalje aktivnosti u zavisnosti od tipa korisnika.



Glavni prozor na kome su prikazane sve auto škole koje su dodate u aplikaciju, sa određenim dugmićima koji izvršavaju

određene akcije.

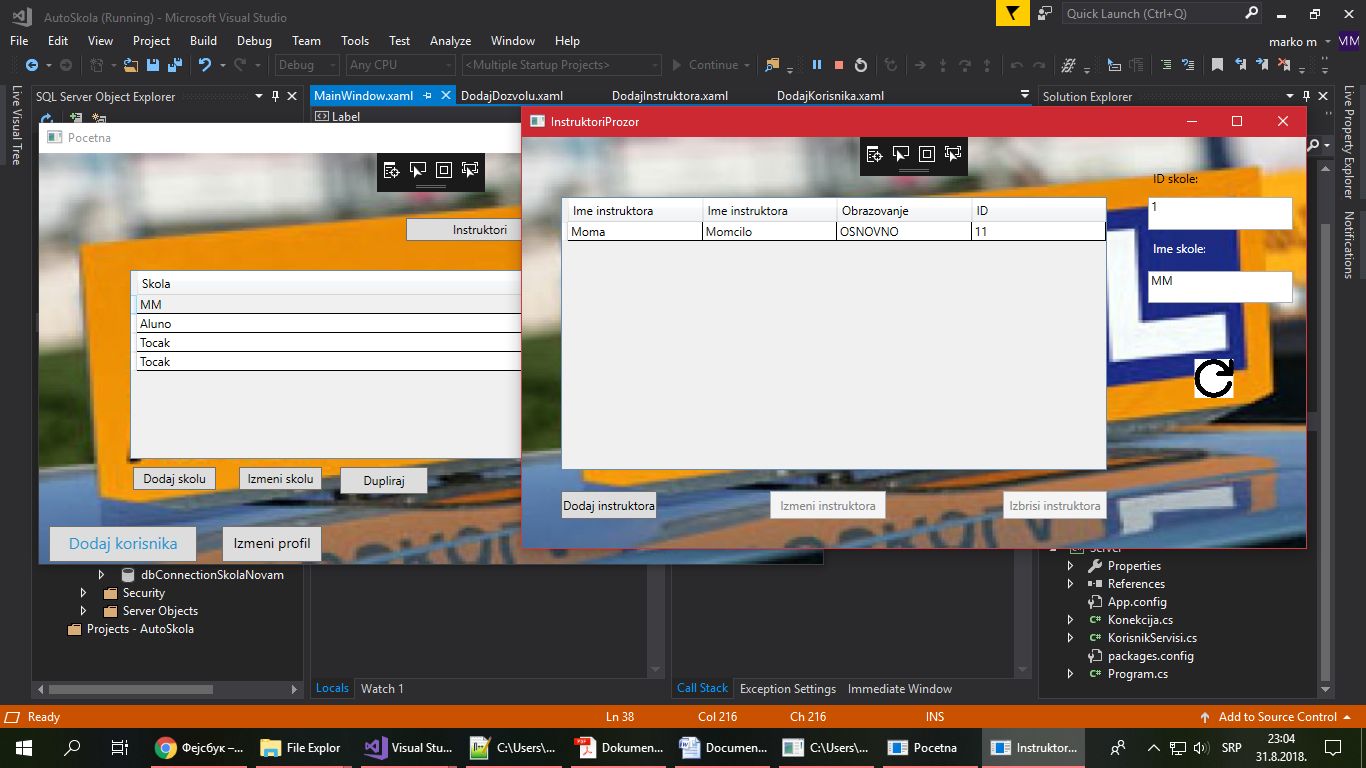


Korisnik može da doda novu školu, može da

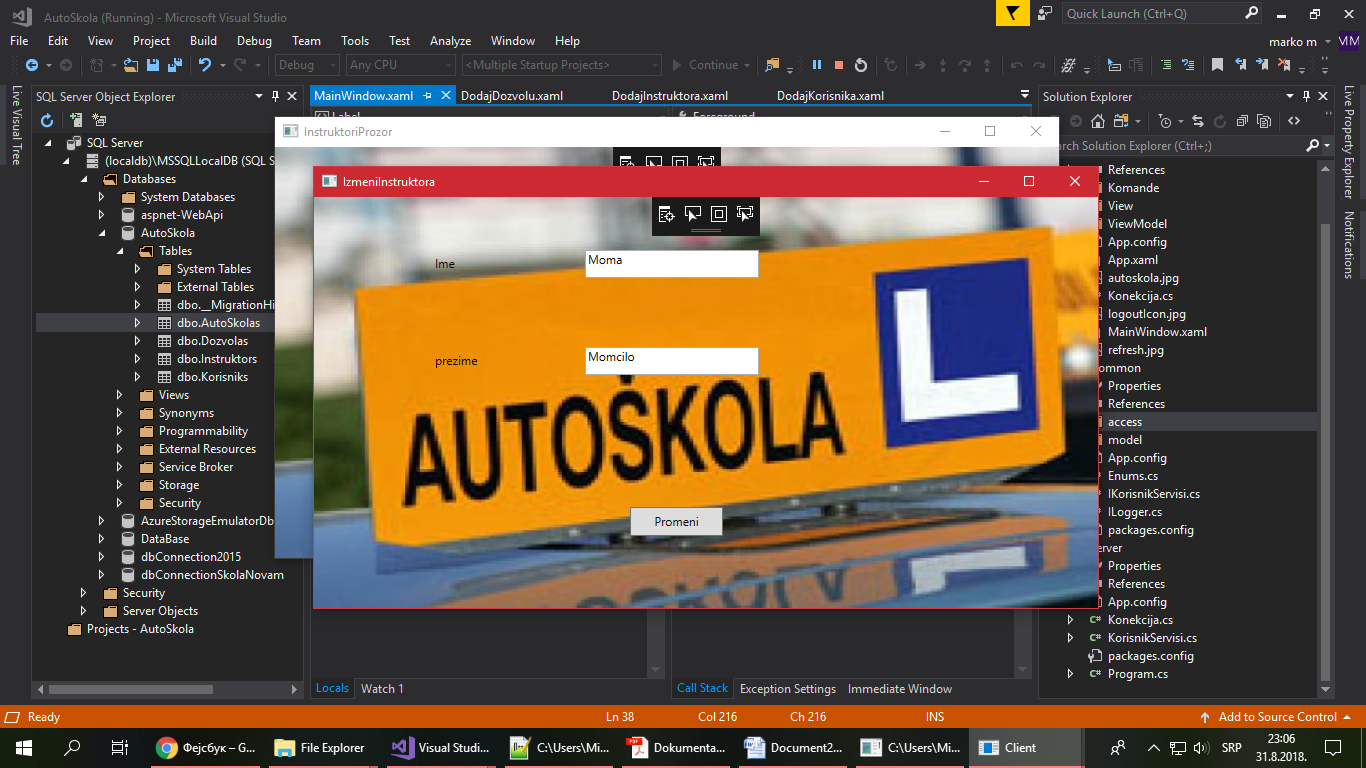
promeni svoje podatke, prikaže sve instruktore određene auto škole, i sve dozvole za koje je moguće polagati u auto školi. Administrator pored svih tih mogućnosti ima još dodatnu opciju da doda novog korisnika.

Pored dodavanja korisnik može da duplira auto školu i da izbriše željenu auto školu.

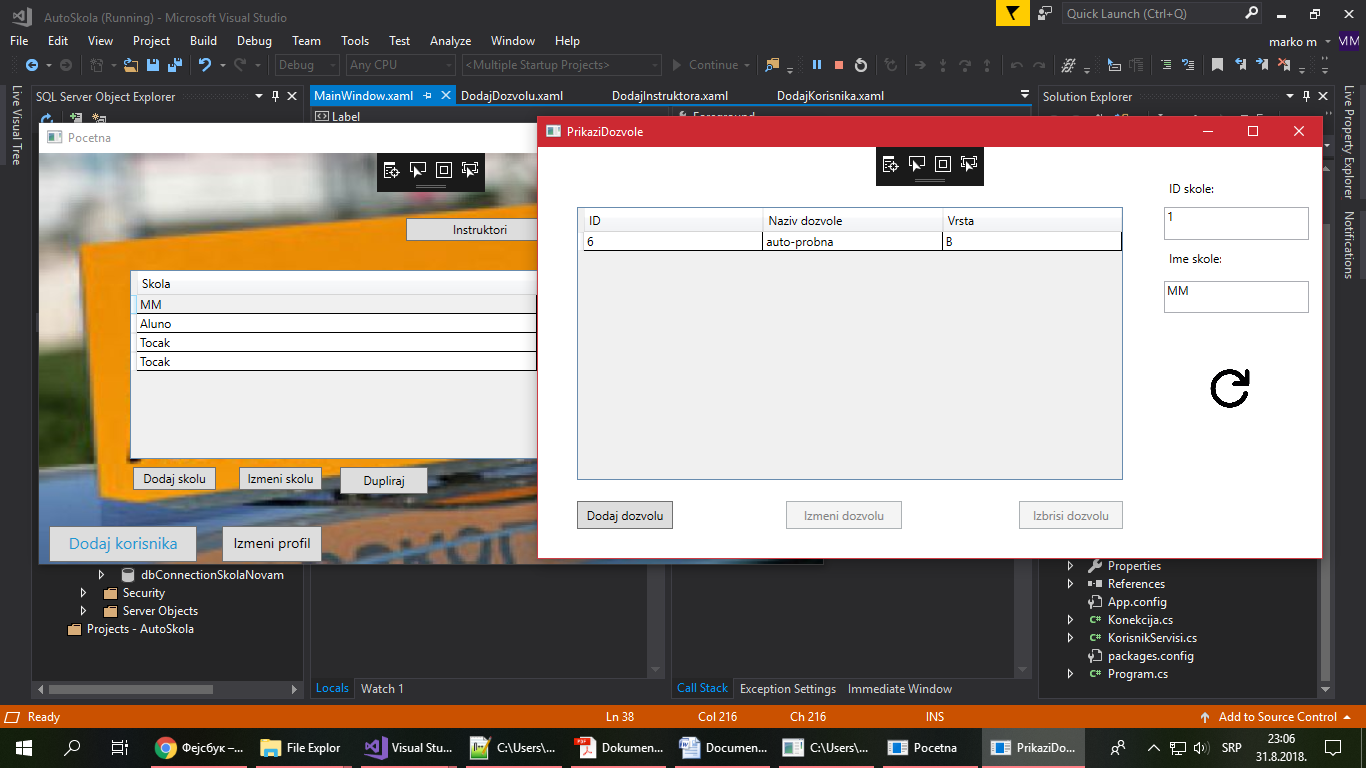
Kada izaberemo skolu I pritisnemo dugme Instruktori izlazi nam sledeći prozor na kome se mogu videti svi instruktori u auto školi:



Ovde možemo da dodajemo nove Instruktore, menjamo I brišemo postojeće.

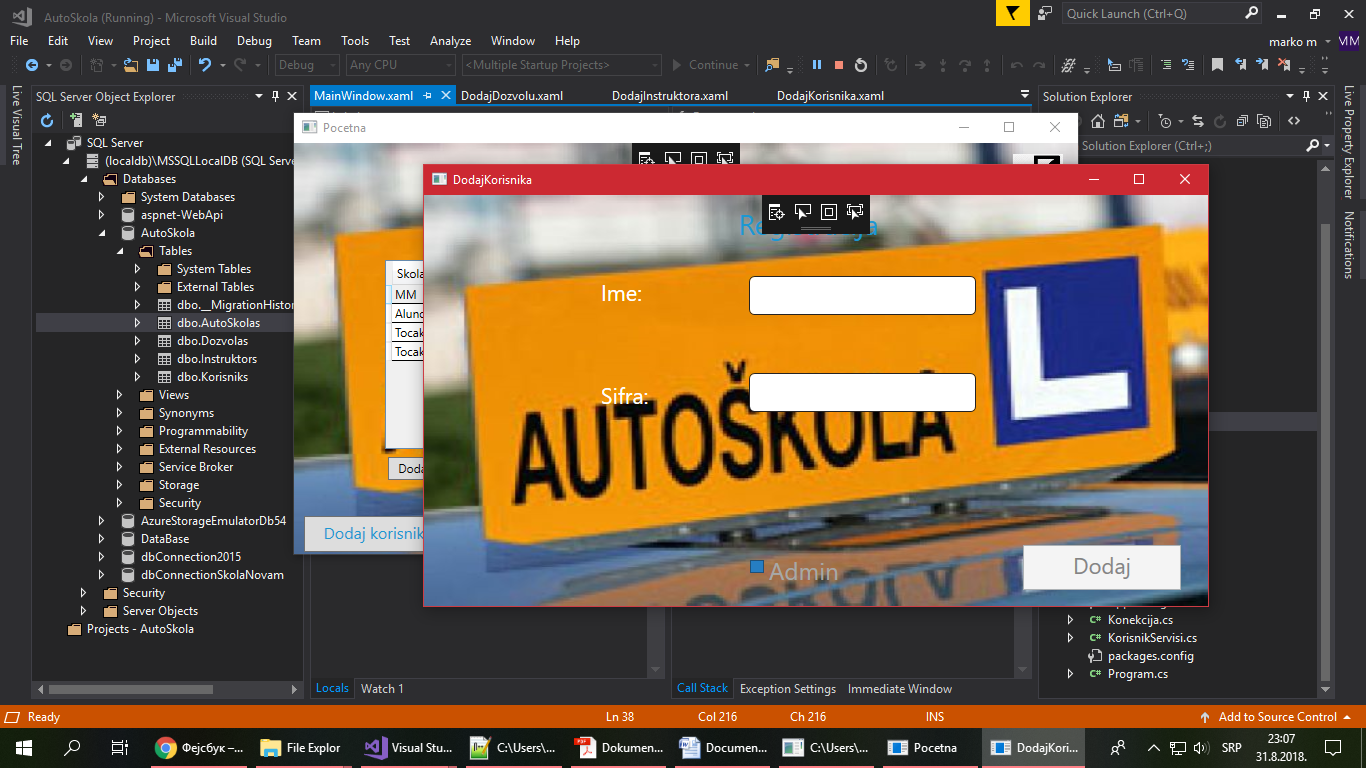


Kada pritisnemo dugme Dozvole izlazi nam sledeći prozor:

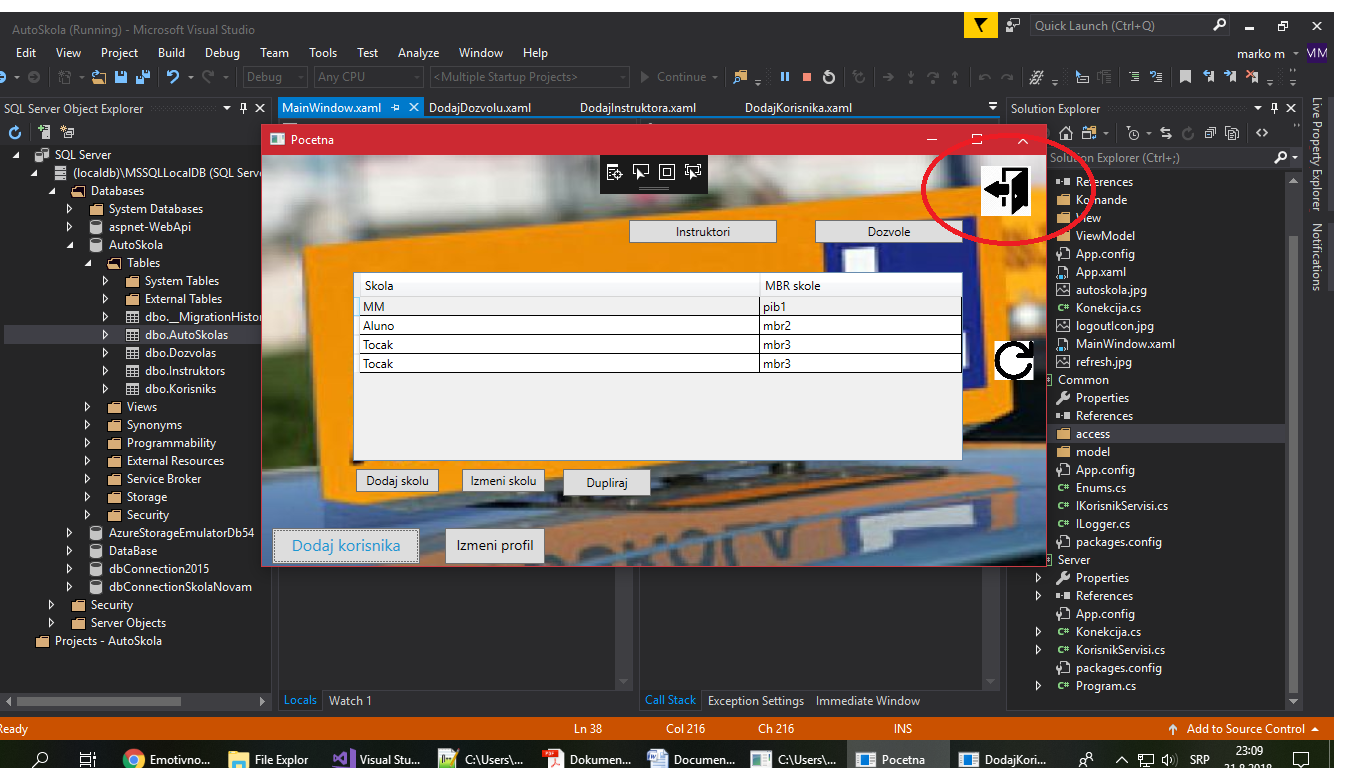


Ovde možemo da dodajemo nove dozvole, menjamo i brišemo postojeće.

Administratoru je omogućeno da dodaje nove korisnike, koji mogu takođe da budu administratori. Administrator unosi: korisničko ime, lozinku, a „čekiranjem“ određenog „combo box-a“ administrator bira da li će novi korisnik biti administrator ili običan korisnik. Dodavanje novog korisnika je prikazano na sledećoj slici:



Na pocetnom prozoru postoje dva dugmica u izgledu ikonica.



Prvi služi da se korisnik izloguje iz aplikacije i da se omogući prijavljivanje novog korisnika.

Ispod njega je drui sa kojim se ažuriraju promene u vidu novih auto skola i izmenama trenutnih.

Usluge WCF servisa čuvaju se u posebnoj biblioteci, Class Library, koja je referencirana od strane WCF server i klijent aplikacije.

**Marko Mijatović PR21/2015.**