

Razvoj višeslojnih aplikacija

Event Planner

Tim 15

PR 33/2020 Tatjana Kosić

PR 156/2020 Duška Damjanović

PR 85/2020 Dejan Dobrilović

PR 70/2020 Milica Klincov

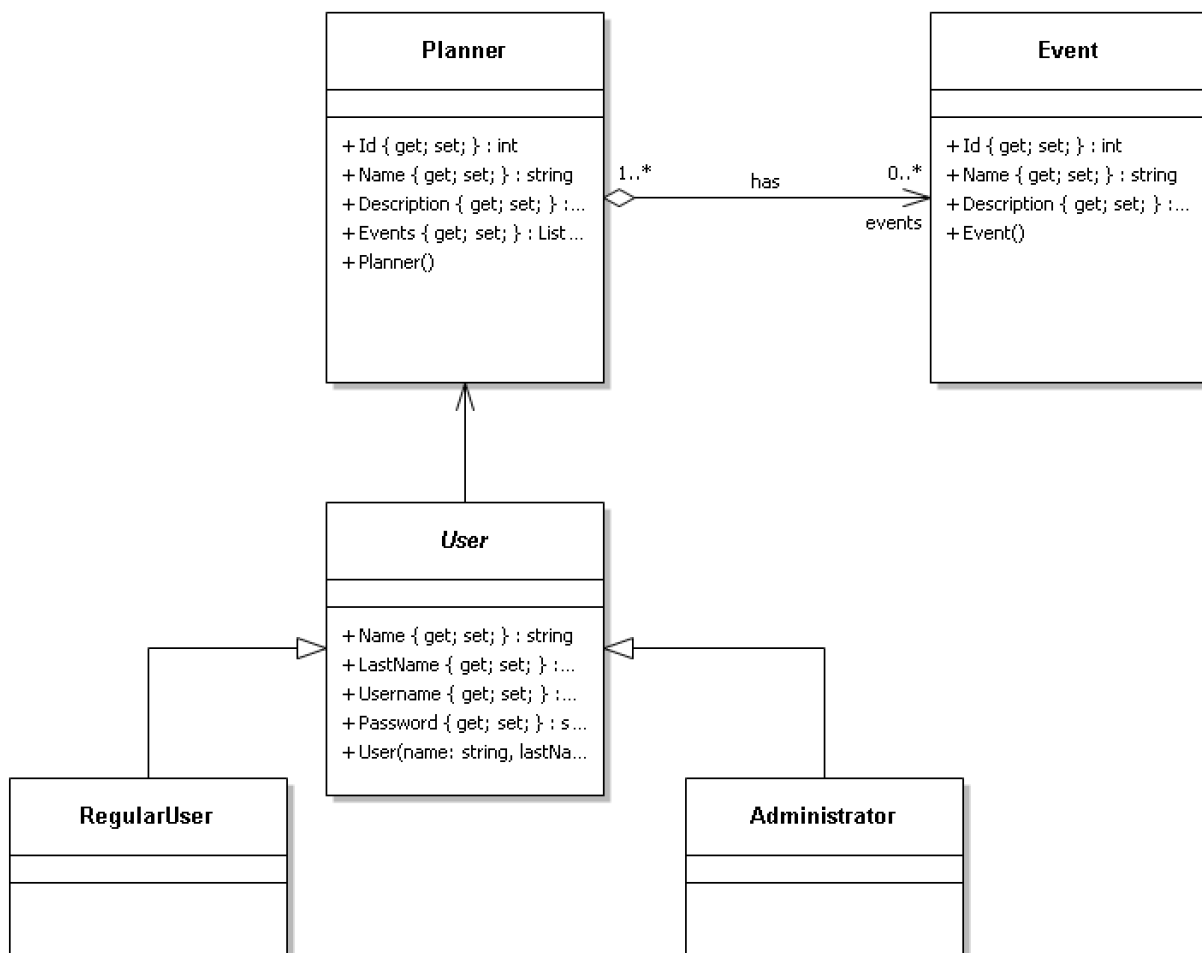
6.6.24.

Specifikacija i User manual

Projekat predstavlja klijent-server aplikaciju za rad sa zadatim modelom podataka na osnovu UML dijagrama. Aplikacija je sa klijentske strane implementirana u WPF tehnologiji (MVVM), dok serverski deo koristi WCF tehnologiju uz podršku Entity Framework-a za rad sa LocalDB bazom podataka koja je integrisana u Visual Studio 2019.

Aplikacija omogućava prijavu i odjavu korisnika, inicijalizaciju podataka, manipulaciju podacima (dodavanje, izmena, brisanje, dupliranje), kao i osvežavanje prikaza. Korišćena je i log4net biblioteka za beleženje kritičnih događaja.

Model podataka:



Početni izgled aplikacije:

1. Prijava i odjava (incijalno postoji jedan administrator sa kredencijalima 'admin', 'admin'):

A login form with a light gray background. It features two input fields: the first is labeled 'Username' with a person icon to its left, and the second is labeled 'Password' with a key icon to its left. Below these fields is a blue button with the text 'Login' in white.

2. Dashboard koji vide korisnici:

1

2

3

4

5

6

7

8

9

10

11

11

Planer1

OpisPlanera1

19

Planer43

OpisPlanera43

20

Planer1

OpisPlanera1

22







Planer23454

OPiss

1. Izmena profila korisnika
2. Prikaz planera i event-a
3. Dodavanje novog korisnika (opcija vidljiva samo administratoru)

4. Logovanje kritičnih događaja
5. Odjava korisnika
6. Pretraga po unetom parametru
7. Dodavanje planera
8. Dodavanje event-a u planeru
9. Izmena planera
10. Dupliranje planera
11. Brisanje planera

3. Dashboard za event-e:

11	Planer1	OpisPlanera1				
Id	Name	Description	1	2		
10	EventNoviii90	Noviii				
12	Noviovi	425				

1. Izmena event-a
2. Brisanje event-a

4. Logovanje:

```

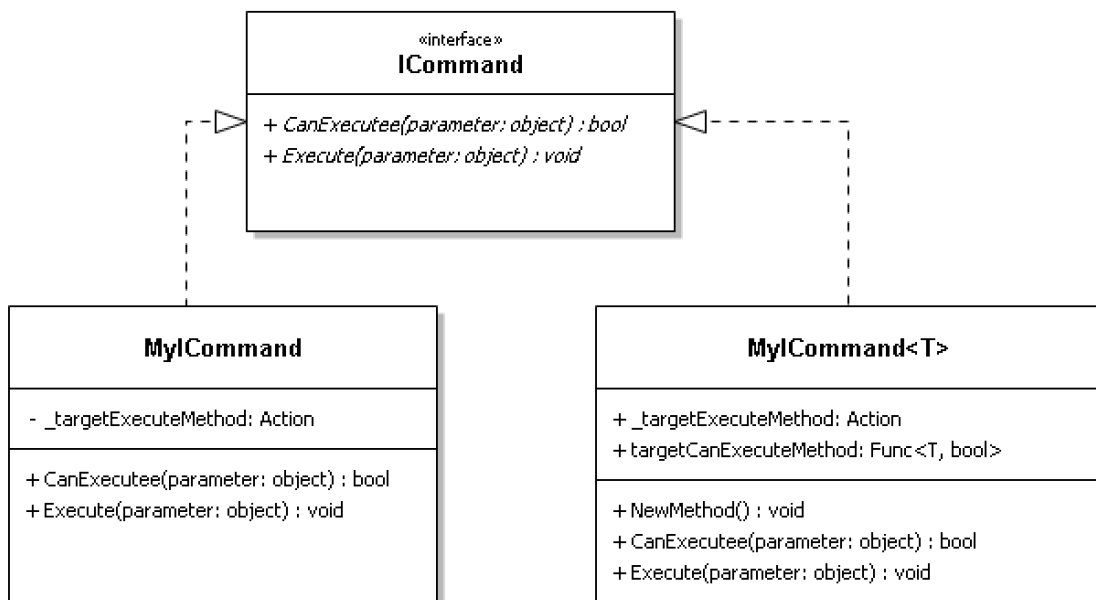
2024-06-04 15:20:04,644 DEBUG Client.ViewModel.LoginViewModel - Created factory
2024-06-04 15:20:05,255 INFO Client.ViewModel.LoginViewModel - Login successful marko|1234
2024-06-04 15:20:59,403 DEBUG Client.Model.MessageHost - Message received
2024-06-04 15:20:59,466 DEBUG Client.Model.MessageHost - Message sent: Common.Planner
2024-06-04 15:21:02,302 DEBUG Client.Model.MessageHost - Message received
2024-06-04 15:21:02,306 INFO Client.ViewModel.DashboardViewModel - Edited planner: 11
2024-06-04 15:21:02,323 DEBUG Client.Model.MessageHost - Message received
2024-06-04 15:21:02,325 DEBUG Client.Model.MessageHost - Message sent: Common.Planner
2024-06-04 15:21:02,326 DEBUG Client.Model.MessageHost - Message sent: Edit
2024-06-04 15:21:02,327 INFO Client.ViewModel.PlannersViewModel - Planner edited: id=Common.Planner
2024-06-04 15:22:09,181 DEBUG Client.Model.MessageHost - Message received
2024-06-04 15:22:09,236 DEBUG Client.Model.MessageHost - Message sent: Common.Planner
2024-06-04 15:22:39,821 DEBUG Client.Model.MessageHost - Message received
2024-06-04 15:22:39,824 INFO Client.ViewModel.DashboardViewModel - Edited planner: 20
2024-06-04 15:22:39,842 DEBUG Client.Model.MessageHost - Message received
2024-06-04 15:22:39,844 DEBUG Client.Model.MessageHost - Message sent: Common.Planner
2024-06-04 15:22:39,844 DEBUG Client.Model.MessageHost - Message sent: Edit
2024-06-04 15:22:39,845 INFO Client.ViewModel.PlannersViewModel - Planner edited: id=Common.Planner
2024-06-04 15:23:24,444 DEBUG Client.ViewModel.LoginViewModel - Created factory
2024-06-04 15:23:25,811 INFO Client.ViewModel.LoginViewModel - Login successful admin|admin
2024-06-04 15:23:29,079 DEBUG Client.Model.MessageHost - Message received
2024-06-04 15:23:29,091 DEBUG Client.Model.MessageHost - Message sent: Common.Planner

```

Korišćeni obrasci i UML dijagrami

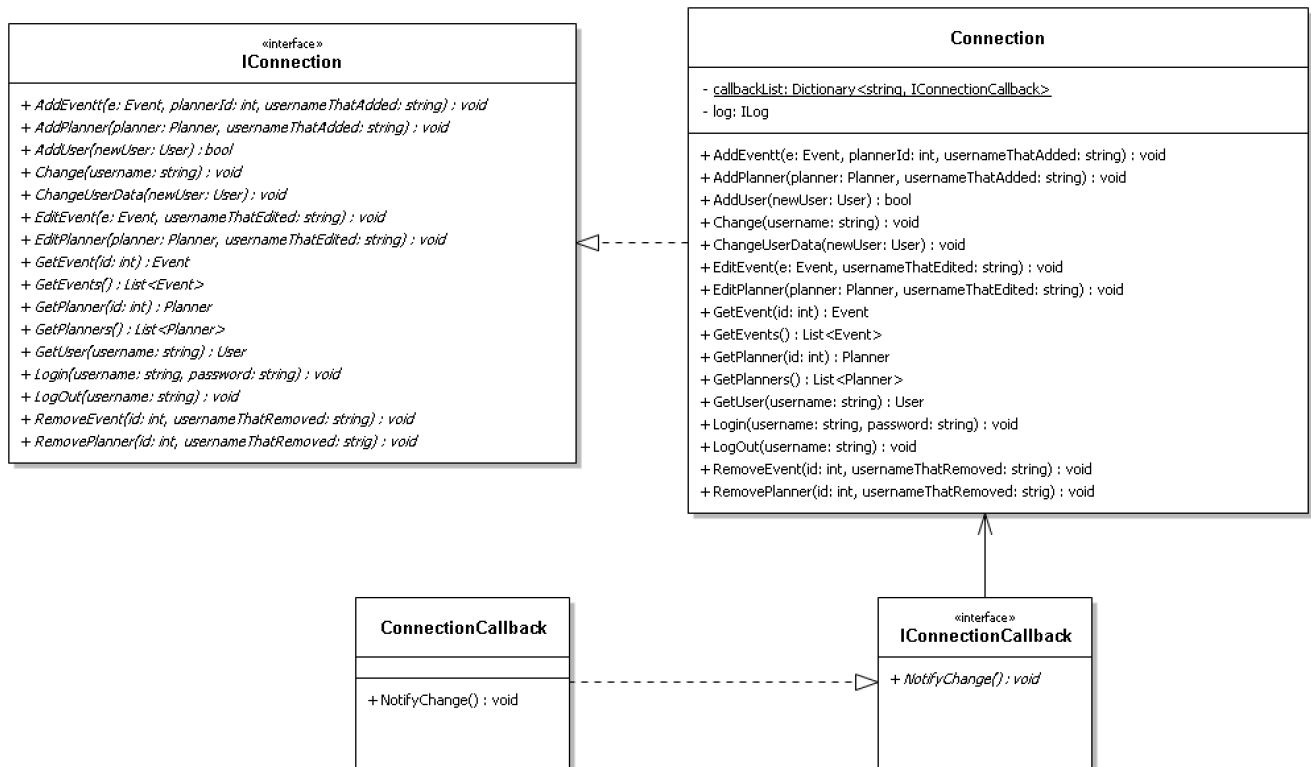
1. **Komanda** - enkapsulira zahteve ka servisu u objekte i može se koristiti:
 - kada želimo da objekat paramterizujemo operacijom koju treba da izvrši
 - za poništavanje akcije (Undo/Redo)
 - za upotrebu Command Binding-a za UI

Komanda se oslanja na ugrađeni interfejs ICommand, a realizovana je u vidu 2 klase, od kojih je jedna generička.



1.1 Komanda

2. **WCF Duplex Observer** - mehanizam u kojem i klijent i server mogu da šalju zahteve (poruke) putem jednostavne full-duplex komunikacione linije. Server obaveštava klijente o izmenama tako što pozove metodu `NotifyChange`, `IConnectionCallback` interfejsa koju klijenti implementiraju.



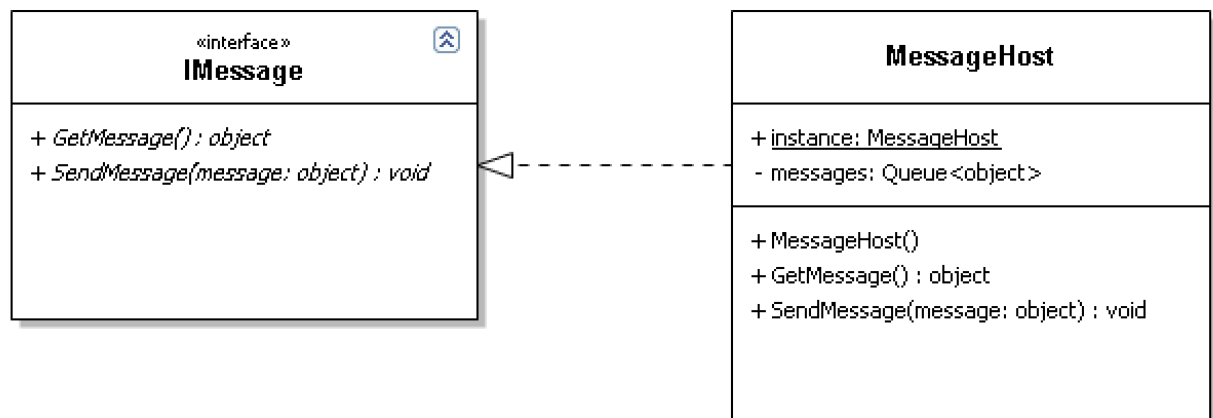
2.1 WCF Duplex Observer

3. **Singleton** - omogućuje nam da imamo samo jednu instancu date klase
Korišćen je za obaveštavanje MainWindowViewModel-a koji View da prikaže.



4. **Message** - obrazac definisan za WCF komunikaciju između 2 objekta, u ovom slučaju 2 prozora.

Klasa koja rukuje prijemom i slanjem poruka MessageHost je napravljena da bude u singleton obrascu jer je dele svi prozori.



4.1 Message

Korišćene tehnologije

- ◆ WPF
- ◆ WCF Duplex
- ◆ Entity Framework
- ◆ LocalDB
- ◆ log4net biblioteka
- ◆ Material Design
- ◆ Data i Command Binding
- ◆ MVVM
- ◆ Komanda, WCF Duplex Observer, Singleton, Message