



UNIVERZITET U NOVOM SADU
FAKULTET TEHNIČKIH NAUKA U
NOVOM SADU



Razvoj višeslojnih aplikacija u elektroenergetskim sistemima

Projektna Dokumentacija

Autor: Srđan Stankov - PR140/2016

Asistent: Mihailo Vasiljević

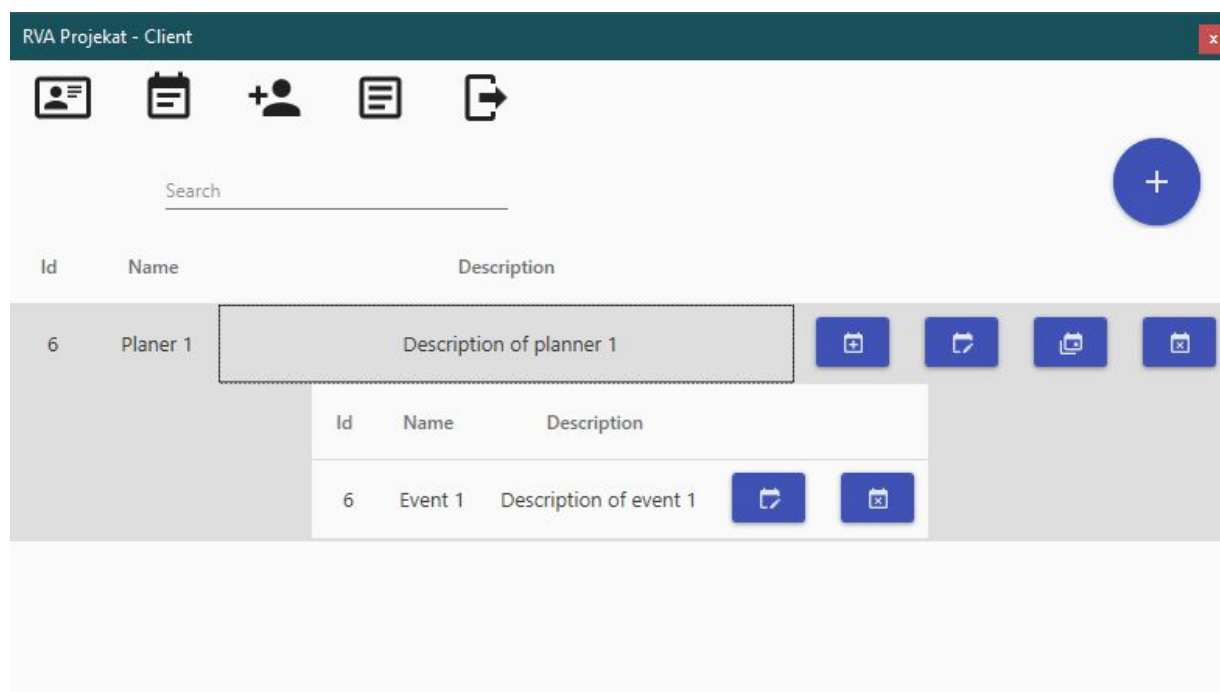
Profesor: Darko Čapko

Kratko uputstvo za upotrebu

Pre samog uputstva za upotrebu, neophodno je napomenuti da je potrebno prvo pokrenuti serversku aplikaciju, a zatim proizvoljan broj instanci klijentske aplikacije.

Prilikom pokretanja klijentske aplikacije, neophodno je se prijaviti / registrovati (na sistemu postoji trenutno jedan administrator sa kredencijalima *admin,admin*).

Nakon uspešnog prijavljivanja / registracije otvara se glavni prozor (Slika 1.) sa prikazom primarnih i sekundarnih objekata sa sledećim opcijama :



Slika1. Glavni prozor

1. Dugme za izmenu podataka trenutno prijavljenog korisnika
2. Dugme za prikaz svih primarnih i sekundarnih podataka
3. Dugme za dodavanje novog korisnika (prikazuje se samo administratoru)
4. Dugme za prikaz izvršenih korisnikovih akcija
5. Dugme za odjavu sa sistema
6. Polje za pretragu primarnih objekata
7. Dugme za dodavanje novog primarnog objekta
8. Dugme za dodavanje novog sekundarnog objekta
9. Dugme za izmenu primarnog objekta
10. Dugme za dupliranje primarnog objekta
11. Dugme za brisanje primarnog objekta
12. Dugme za izmenu sekundarnog objekta
13. Dugme za brisanje sekundarnog objekta

Za izradu same aplikacije, upotrebljene su sledeće tehnologije i obrasci:

- Entity Framework
- WCF Duplex
- log4net biblioteka
- MaterialDesign
- WPF
- Data i Command Binding
- Obrasci:
 - MVVM
 - Komanda
 - Observer (modifikovan)
 - Singleton
 - Message (modifikovan)

Primenjeni obrasci

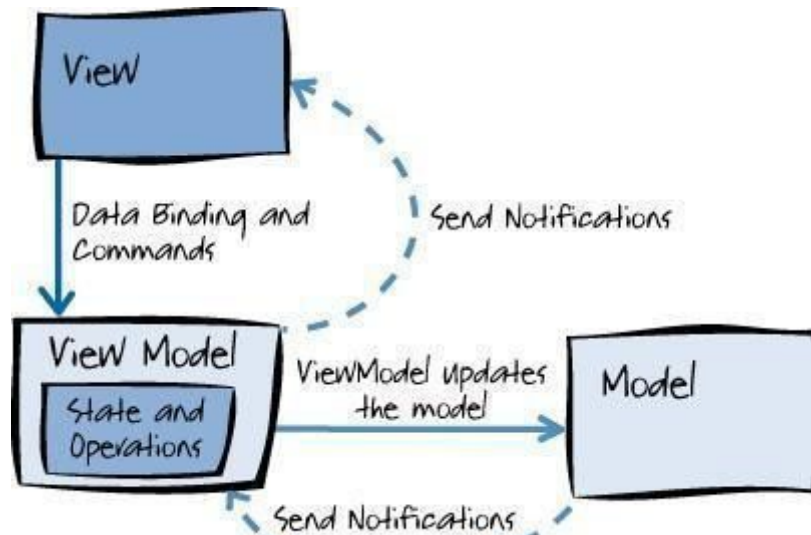
MVVM

Model-View-ViewModel(MVVM) obrazac se koristi za separaciju zaduženje u aplikaciji (Slika 2.). Sastoji se iz tri dela:

- Model - Model podataka, reprezentuje aplikativne podatke
- View - Prikaz podataka
- ViewModel - Interakcija korisnika sa podacima

Sam tok komunikacije je sledeći:

- Korisniku se prikazuje određeni View
- Korisnikove interakcije obrađuje ViewModel
- Sve izmene podataka ViewModel stavlja u Model



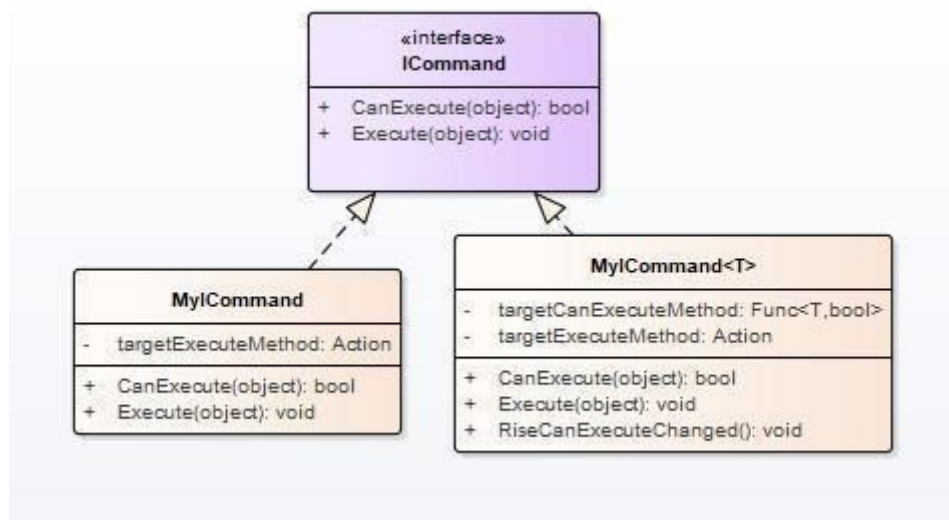
Slika 2. MVVM

Komanda

Komanda - enkapsulira zahteve ka servisu u objekte i može se koristiti:

- Kada želimo da objekat paramterizujemo operacijom koju treba da izvrši
- Za poništavanje akcija(Undo/Redo)

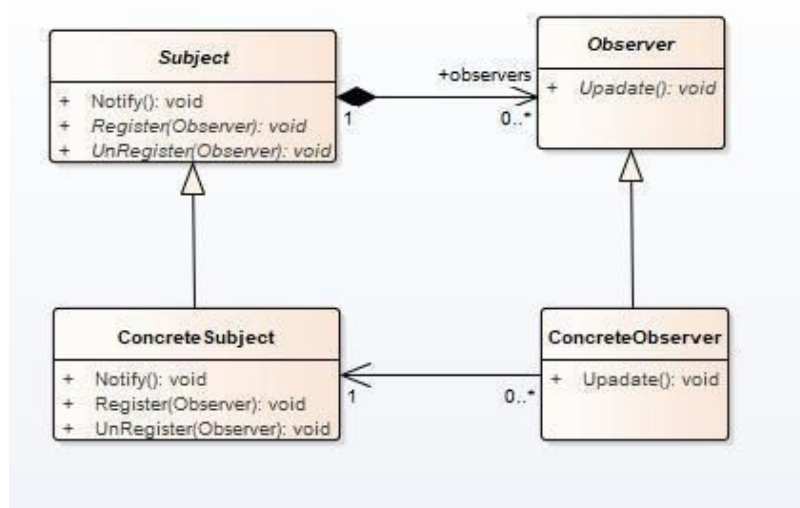
Komanda (Slika 3.) se oslanja na ugrađeni C# interfejs *ICommand* , i realizovana je u vidu dve klase, jedna od njih je generička. Namena ove vrste komandi je za upotrebu **Command Bindinga** za korisnicki interfejs.



Slika 3. Prva vrsta komandi

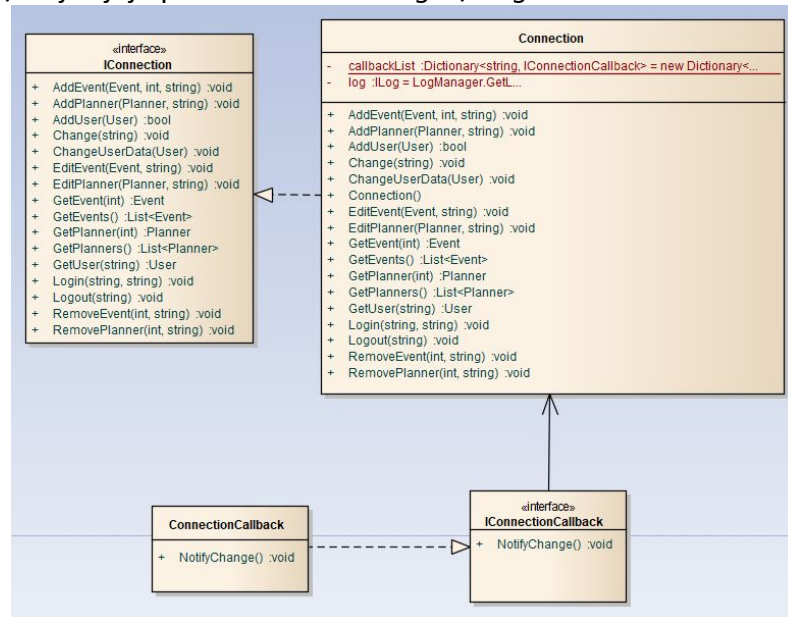
Observer

Observer - realizuje vezu jedan prema više , i omogućuje da zavisne klase saznaju za promene nastale u primarnoj klasi (Slika 4.). Implementirano je pomocu C# ugrađenog mehanizma event-a.



Slika 4. Standardni observer

U projektu je upotrebljen *WCF Duplex* mehanizam , tako da obe strne mogu da šalju zahteve - poruke putem jednostavne **full-duplex** komunikacione linije. Server obaveštava klijente o izmenama tako sto poziva metodu *NotifyChange*, *IConnectionCallback* interfejsa koji klijenti implementiraju (Slika 5.). Klijenti se prijavljuju / odjavljuju pomoću metoda *Login / Logout*.



Slika 5. WCF Duplex Observer.

Singleton

Singleton - omogućuje nam da imamo samo jednu instancu date klase i njen globalni pristup. Koristi se kada želimo da imamo samo jednu instancu date klase. Sama instanca odgovorna je za svoje postojanje.

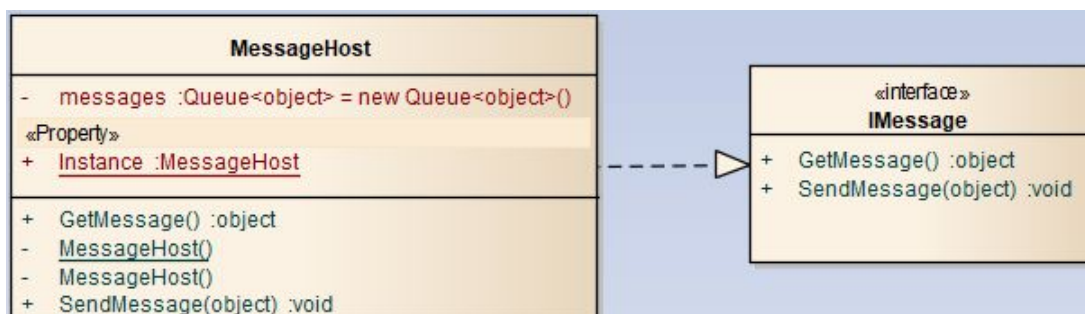
Singleton obrazac je primenjen za obavestavanje MainWindowViewModel-a, na klijentskoj strani, koji View da prikaze (Slika 6.).



Slika 6. Singleton

Message

Message - omogućuje nam da saljemo poruke izmedju 2 objekta, u ovom slucaju 2 prozora. Klasa koja rukuje prijemom i slanjem poruka *MessageHost* je napravljena da bude u singleton obrascu jer je svi prozori dele. (Slika 7.)



Slika 7. Message