Opšti zadatak

Potrebno je razviti klijent-server aplikaciju za rad sa zadatim modelom podataka na osnovu UML dijagrama. Model podataka se sastoji od instanci primarne (*zelene*) klase i instanci pomoćnih klasa. Očekuje se primena bar 4 obrasca.

Klijentska aplikacija

Klijentsku aplikaciju razviti u WPF tehnologiji, oslanjajući se na MVVM obrazac. Pri tome, treba omogućiti:

* Prijavu i odjavu korisnika.
* Inicijalno postoji jedan administrator pod imenom ‘admin’, sa lozinkom ‘admin’.
* Administrator ima pravo da dodaje druge korisnike, kao i da izvršava sve ostale akcije. Može biti više administratora.
* Svaki korisnik može da izmeni svoje osnovne podatke: ime i prezime.
* Prikaz svih instanci primarne (*zelene*) klase, sa pomoćnim prikazima instanci ostalih klasa.
* Pretragu primarnih (*zelenih*) podataka po zadatom uslovu.
* Dodavanje novih instanci primarnih i pomoćnih podataka po potrebi.
* Izmenu primarnih, uključujući i liste pomoćnih podataka po potrebi.
* Brisanje primarnih, uključujući i liste pomoćnih podataka po potrebi.
* Dupliranje primarnih podataka.
* Poništavanje i ponovno izvršavanje komande izvršene nad instancama primarnih podataka (undo/redo).
* Osvežavanje prikaza, za slučaj da je drugi korisnik napravio izmene. Osvežavanje vršiti na jedan od načina:
  + po potrebi,
  + na vremenski interval ili
  + na zahtev.
* Omogućiti rad više instanci klijentskog programa, gde mogu biti prijavljeni različiti korisnici. Testiranje izvršiti sa podizanjem dva klijenta na jednom računaru, gde mogu da se prijave različiti korisnici, a koriste usluge istog servisa.
* Svi klijenti mogu da vide i menjaju sve podatke.
* Ukoliko dođe do konflikta pri radu sa primarnim podacima (izmena ili brisanje), potrebno je izbaciti pitanje sa dve opcije:
  + Odbaciti svoje izmene, ili
  + Pregaziti tuđe izmene.

Serverska aplikacija

Serverski deo razviti u WCF tehnologiji, oslanajući se na *Entity Framework* za rad sa bazom podataka. Potrebno je omogućiti:

* Usluživanje više korisnika.
* Čuvanje svih podataka u bazi podataka.
* Podršku za potrebe klijentske aplikacije.

Napomena za implementaciju zadatka

Pored dobro urađenog zadatka, na konačnu ocenu projekta utiče i nivo kvaliteta dizajna (korišćenje programskih obrazaca), mnemoničko imenovanje klasa i promenljivih, korišćenje *lock* i *using* blokova, kao i dokumentovanja aplikacije (upotreba UML dijagrama klasa za svaki obrazac i druge bitne delove aplikacije).

Napomena za UML model

Dobijeni UML model proširiti podacima koji nedostaju:

* Dodati potrebna polja i property-je,
* Dodati klase ako treba.

Zadate podatke ne menjati:

* Ako je zadata klasa apstraktna, treba da ostane apstraktna,
* Ako je definisan neki deo relacije, treba ga uvažiti,
* Ako je definisana metoda, treba je implementirati.

Inicijalizacija podataka (obavezno za prolaz)

Napisati funkciju koja:

1. proverava da li postoje inicijalni podaci i
2. dodaje ih ukoliko ne postoje.

Dodatni bodovi (obavezno za maksimalnu ocenu)

Potrebno je:

* za klijentsku i serversku aplikaciju u tekstualnoj datoteci beležiti događaje koji su se odigrali. Za te potrebe preporučuje se biblioteka log4net.
* za klijentsku aplikaciju razviti grafičku, tabelarnu komponentu za obaveštavanje klijenta o izvršenim događajima koje je klijent inicirao. Svako obaveštenje za klijenta je novi red u tabeli. U ovoj implementaciji, obeshrabruje se prikaz obaveštenja putem dijaloga.

Za obe tačke očekuju se informacije o vremenu kada se događaj odigrao, tipu događaja (DEBUG, INFO, WARN, ERROR, FATAL) i poruci koja opisuje sam događaj.

Biblioteke

EF: <https://msdn.microsoft.com/en-us/library/aa937723(v=vs.113).aspx>

Log4net: <https://logging.apache.org/log4net/>

*\*Ukoliko postoje nedoumice sa zadatkom, konsultovati asistenta.*