

Virtuelizacija procesa

Tehnička dokumentacija
“Evidencija prognozirane i ostvarene
potrošnje električne energije”

Strahinja Perić PR20/2020

Sadržaj:

1. Uvod
2. Arhitektura projekta sa tokom podataka
3. Opis interfejsa
4. Tehnologije
5. Zaključak

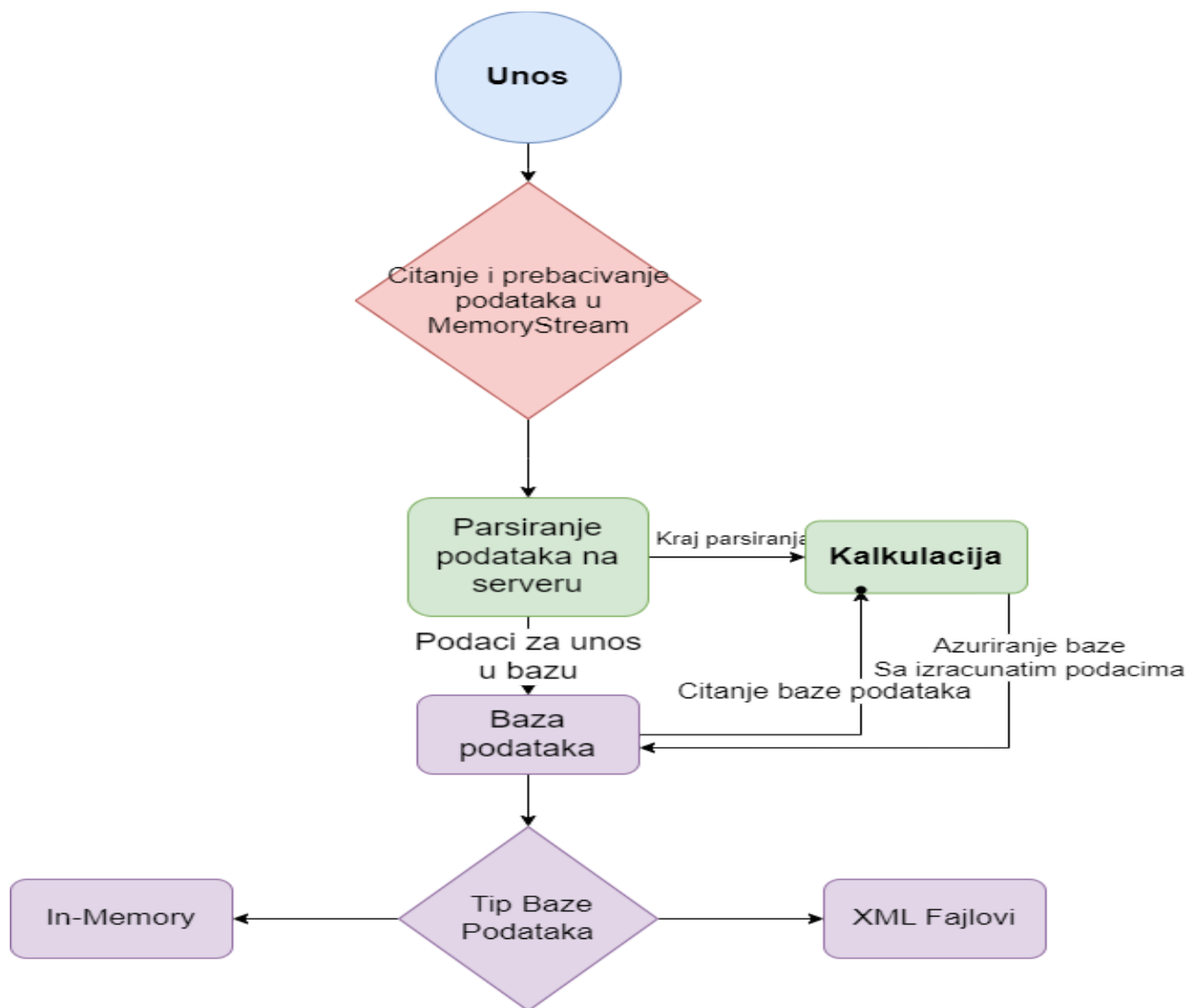
1. Uvod

Aplikacija “Evidencija prognozirane i potrošene električne energije” se bavi evidencijom prognozirane i potrošene električne energije u bazu podataka. Takođe vrši izračunavanja apsolutnog procentualnog odstupanja ili kvadratnog odstupanja u zavisnosti od toga kako je izabrano u konfiguracionom fajlu. U konfiguracionom fajlu može da se izabere i tip baze podataka in-memory ili xml fajlovi. In-memory baza podataka se briše prilikom isključivanja aplikacije. Unos podataka se vrši putem csv fajlova, koji se nalaze u posebnom direktorijumu. Direktorijumi moraju biti različiti za fajlove sa prognoziranim podacima od direktorijuma koji sadrži fajlove sa izmerenim podacima. Za više informacija o pravilima cuvanja i izgleda csv fajlova pogledajte UserManual.

2. Arhitektura projekta sa tokom podataka

Aplikacija je napisana u više slojnoj arhitekturi. Sadrži četiri dela:

- Korisnički sloj - konzolna aplikacija koja služi za komunikaciju sa korisnikom
- Servisni sloj - sloj aplikacije koji sadrži glavnu logiku aplikacije, zadužen je za parsiranje fajlova, proračun podataka i slanje upita ka bazi podataka
- Baza podataka - implementirana kao In-memory baza ili xml baza podataka. Služi za skladištenje podataka i davanje podataka servisnom sloju za proračune.
- Common - class library u kojem se nalaze definicije svih zajedničkih interfejsa za sva tri ostala sloja.



Tok podataka u aplikaciji je vrlo jednostavan. Korisnik unosi absolutnu putanju do direktorijuma gde se fajlovima nalaze. Fajlovi se šalju servisnom sloju putem MemoryStream-a. Servisni sloj obrađuje fajlove izvlači podatke i pravi upit prema bazi podataka. Baza podataka proverava koji tip baze je izabran i na osnovu toga skladišti podatke. Nakon što su svi fajlovi iz direktorijuma učitani servisni sloj kreće sa izračunavanjem apsolutnog procentualnog odstupanja ili kvadratnog odstupanja. Servisni sloj šalje zahtev bazi za sve dostupne podatke. Kada ih primi, izračuna potrebne stavke i šalje bazi azurirane podatke. I na kraju baza opet proverava u koju bazu podataka treba da upise azurirane podatke.

3. Opis interfejsa

Najbitniji interfejs za aplikaciju je `IDataBase`, u kojem su opisane sve funkcije vezane za komunikaciju sa bazom podataka. Unutar tog interfejsa se nalaze funkcije za:

- Dodavanje podataka - `AddLoad`, `AddAudit` i `AddImportedFile`
- Vraćanje liste podataka - `ReadLoad`
- Ažuriranje baze - `UpdateLoads`

Sve funkcije za dodavanje imaju slične parametre, klasu objekta koji treba da se doda u bazu i tip baze podataka gde se dodaje objekat. Od te tri funkcije najkomplikovanija je `AddLoad`. Najkomplikovanija je iz razloga što nakon na primer podataka prognoziranih vrednosti sledeći put kada unosim podatke potrebno je proveriti da li objekat sa tim datumom i vremenom već postoji i ako postoji potrebno je samo taj objekat ažurirati, što nije problem kod In-memory baze podataka već kod XML baze.

`ReadLoad` funkcija samo čita bazu podataka i vraća listu objekata `Load` koji predstavljaju podatke koje se nalaze u bazi,

`UpdateLoads` funkcija prima listu objekata klase `Load` koji treba da se upisu u bazu podataka. Funkcija upisuje novu listu objekata preko stare liste ili starog fajla i tako se azurira baza sa najnovijim podacima.

4. Tehnologije

Tehnologija korišćena za komunikaciju između slojeva aplikacije je WCF. Za čitanje podataka je korišćen `FileStream` a za prenos fajlova između korisničkog sloja i servisnog sloja `MemoryStream`. Sve vreme se vodilo računa o održavanju memorije korišćenjem `Dispose` pattern-a. In-memory baza je implementirana kao `Dictionary` gde je ključ id reda gde se čuva podataka a vrednost je objekat klase podatka. Za pisanje i čitanje

xml fajlova korišćena je klasa XmlDocument koja dolazi uz standardnu biblioteku c#-pa.

5. Zaključak

Aplikacija nije u potpunosti gotova. Treba poboljšati mehanizme za rukovanjem greškama koje korisnik može da napravi nemarnim unosom ili lošim pravljenjem csv fajla. Dodati bolji korisnički interfejs u kome se podešavanja aplikacije mogu nameštati bez menjana app.config fajla. Promeniti bazu podataka sa in-memory ili xml na SQL bazu podataka radi preglednosti i lakše manipulacije podacima. Takođe treba napraviti sistem ubacivanja podataka direktno iz baze podataka a ne preko csv fajlova.