

Virtuelizacija procesa

Na predmetu rade:

- Zorana Babić
- Zoran Pajic
- Zoran Janković
- Bojan Jelačić

Evidencija prognozirane I ostvarene potrošnje elektricne energije

Grupa 1

Projekat uradili:

- Marija Stijelja PR13/2020
- Nikolina Bratić PR7/2020
- Miloš Medić PR101/2019
- Irina Kovač PR17/2020

Opis projektnog zadatka

Aplikacija vrši prikupljanje i analiziranje planirane i ostvarene potrošnje električne energije. Potrebno je pročitati csv datoteke koje sadrže ove informacije, čitamo csv datoteke koje klijent šalje serveru. Nakon uspešnog čitanja zahteva se obrada iščitanih vrednosti. Na osnovu pravila vrši se selekcija validnih vrednosti. Nakon toga potrebno je kreirati odgovarajući tip objekta. Objekat može biti: AUDIT, LOAD, IMPORTED. Audit objekat se kreira kada fajl ne sadrži informacije o svakom satu u toku dana(audit Error), takođe, kreiranje se vrši i kada su sve vrednosti validne(audit Info) i kada postoji određeni Warning u našem fajlu(audit Warning). Load objekat se kreira samo ako su svi zahtevi ispunjeni, odnosno, svaka measured datoteka ima korespondentnu forecast. Za svaki sat u toku dana kreirace se po jedan objekat Load.

Nakon uspešno kreiranog objekta potrebno je taj objekat upisati u bazu podataka. U zavisnosti od parametra u app config-u, upisivanje se vrši u InMemory ili u Xml bazu.

Upisani podaci se iščitavaju i nakon toga potrebno je izračunati i dodeliti Load objektu vrednost za apsolutnu ili kvadratnu devijaciju. Odluku koja će se vrednost dodeliti dobijamo iz app config-a.

Nove vrednosti upisujemo u bazu. Ako se sve uspešno izvršilo klijent će dobiti potvrdu od servera, odnosno svi objekti će biti isporučeni klijentu.

Arhitekturu (dijagram) projekta sa tokom podatak

Projekat sadrži:

- Client
- Server
- Common
- Base

Client : WPF(Windows Presentation Foundation) UI Framework koji kreira desktop klijentske aplikacije. Klijent bira putanju sa svog diska i sa te lokacije ako na njoj postoje csv fajlovi šalje fajlove serveru, korišćenjem Memory Stream-a. U glavnom prozoru(xaml) korisnik ostvaruje interakciju sa serverom. U code behind-u ovog prozora implementirane su sve funkcionalnosti.

Common : Class Library je deljena biblioteka koja se koristi u .Net-u. Ona dozvoljava da se korisne funkcionalnosti implementiraju u modulima koji se kasnije mogu koristiti.

Common sadrži : IFileHandler koji sadrži opis metoda, pomoću kojih je realizovana komunikacija između klijenta i servera(WCF).

Audit, Load I Imported klase sadrže potrebna polja, svojstva I pomoćne metode. FileCSV je pomoćna klasa čiji objekti će se prenositi u komunikaciji. Ona sadrži FileName(naziv fajla koji se prenosi), MemoryStream(sadržaj fajla).

Server : Console Application (.Net Framework 4.7.2)

Konzolna aplikacija koja vrši sve funkcionalnosti ovog sistema.

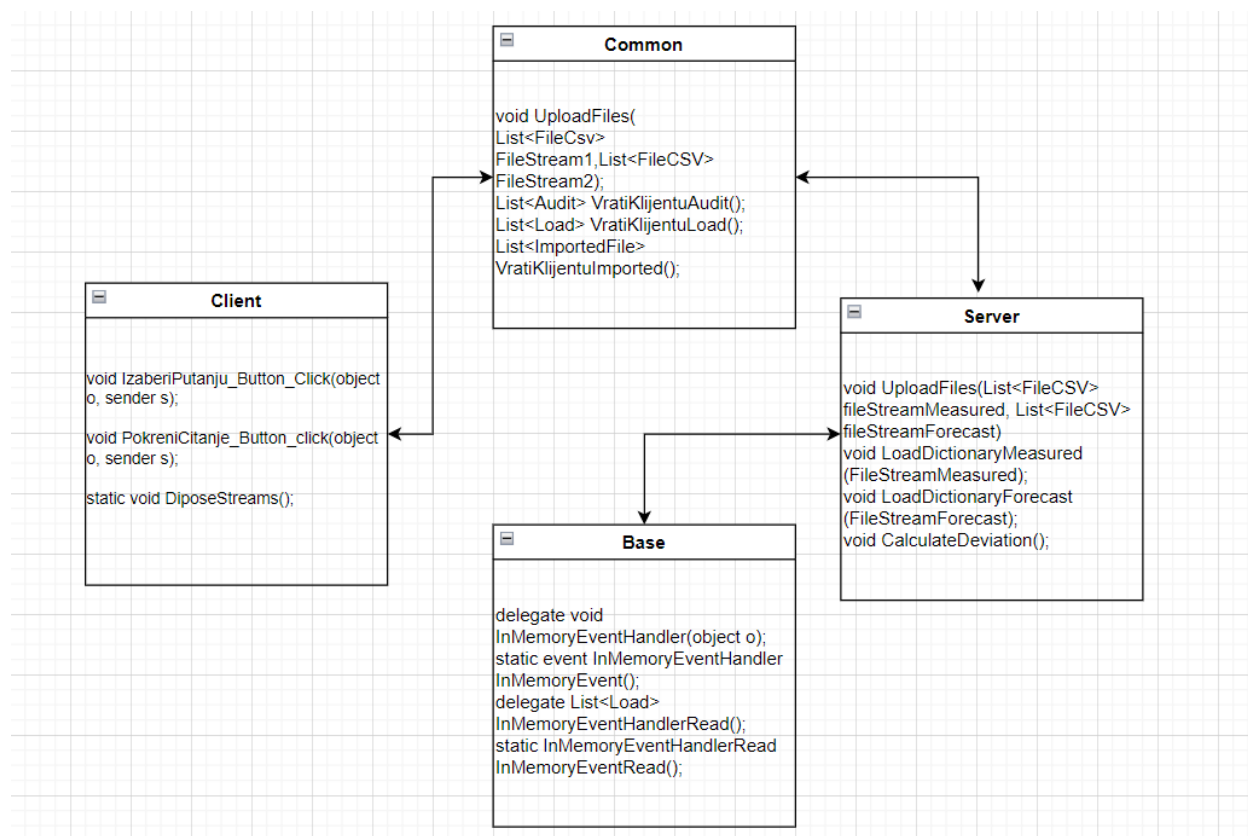
Klasa FileHandler implementira interfejs IFileHandler(metoda LoadFiles). Sva iščitavanja I analize su implementirani na ovoj strani.

Program.cs otvara ServiceHost za ovu komunikaciju na portu 4002(koristi TCP protokol I localhost adresu lokalnog računara).

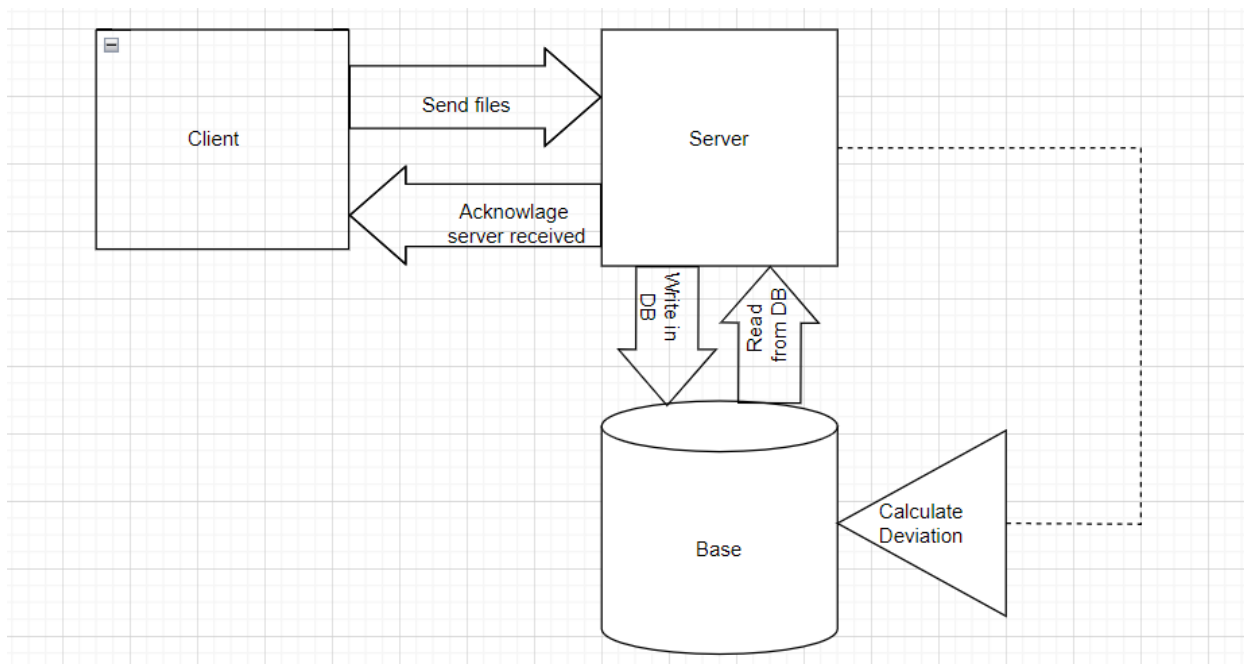
Base : Console Application (.Net Framework 4.7.2)

IXmlDataBase, IDatabase – interfejsi sa opisom svih potrebnih metoda za čitanje I pisanje u bazu.

XmlDataBase I DataBase implementiraju ova dva interfejsa.



Dijagram komponenti prikazuje osnovne metode u projektu, kao I međusobne veze između komponenti.



Dijagram aktivnosti. Prikazan je tok informacija slanje, citanje i upis fajlova, kao i računanje devijacije.

Opis interfejsa sa opisom osnovnih funkcionalnostima

XmlDataBase – vrši sve potrebne operacije nad XML bazom, upis i čitanje audit, imported i load objekata u xml fajl.

DataBase - vrši sve potrebne operacije nad InMemory bazom, upis i čitanje audit, imported i load objekata u rečnike.

IFileHandler – LoadFiles, prihvata memory stream-ove sa klijenta i konvertuje ih u čitljive csv fajlove.

Opis tehnologija koje su korišćene

Za komunikaciju između klijenta i server korišćen je WCF(Windows Communication Foundation) – Framework za izgradnju serverski orijentisanih aplikacija. Korišćenjem WCF-a omogućeno je slanje poruka sa jednog serverskog endpoint-a na drugi.

Korišćen je .Net Framework 4.7.2 , odnosno biblioteka koja omogućava postizanje raznih zadataka uključujući manipulaciju stringovima, kolekcijama podataka, konekciju na bazu podataka i pristup fajlovima.

Zaključak sa mogućim pravcima budućeg istraživanja i proširenjem zadatka

U mogućem proširenju zadatka bilo bi poželjno implementirati replikaciju podataka čime bi se obezbedila sigurnosna kopija u slučaju otkaza servisa. To bi obezbedili uvođenjem replikatora koji replicira podatke posle određenog vremenskog interval. Nakon otkaza servisa treba implementirati klijentsko povezivanje na servis koji menja ispali servis.