

Proračuni na osnovu zahteva

Dokumentacija projekta iz predmeta – Virtuelizacija
procesa

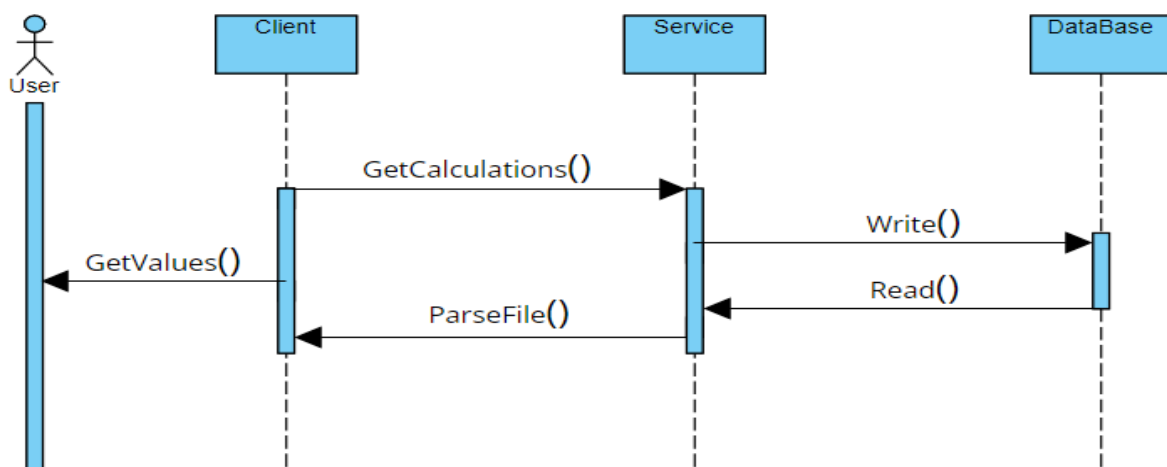
Članovi tima:

- Aleksa Bugarinović PR14/2020
- Aleksa Antonijević PR16/2020
- Jovan Bratić PR10/2020
- Luka Novaković PR113/2020

1. Uvod

Klijent želi aplikaciju za očitavanje potrošnje električne energije. Očitavanje se beleži u CSV datoteci, po satu , za tekući dan, a podaci o potrošnji se čitaju iz XML datoteke. Aplikacija se sastoji od servisa i klijentske aplikacije, koje komuniciraju putem WCF-a. Servisna aplikacija čuva pristigle podatke u bazi podataka koja je implementirana kao datoteka TBL_LOAD.xml. Kad servis primi CSV datoteku, izvrši se parsiranje datoteke i kreira se kolekcija objekata klase **Load**. Load predstavlja podatke za svaki sat iz CSV datoteke. Ukoliko se uoči greška za neki sat, kreiraće se **Audit** objekat koji se upisuje u bazu i prosledjuje klijentskoj aplikaciji u okviru rezultata.

2. Arhitektura projekta



Slika 1. Dijagram toka podataka

2.1. Opis toka podataka

Na početku, korisnik sačuva CSV datoteku na definisanoj lokaciji na računaru gde se nalazi klijentska aplikacija.

Klijentska aplikacija čita CSV datoteku sa podacima o potrošnji električne energije.

Klijentska aplikacija šalje CSV datoteku kao zahtev servisnoj aplikaciji putem WCF-a.

Servisna aplikacija prima zahtev i vrši parsiranje CSV datoteke, izvlačeći podatke o potrošnji za svaki sat.

Servisna aplikacija ažurira XML bazu podataka (TBL_LOAD.xml) sa novim podacima o potrošnji za svaki sat.

Ukoliko se uoči greška prilikom parsiranja podataka, servisna aplikacija kreira Audit objekat koji sadrži informacije o greški.

Servisna aplikacija šalje rezultate ažuriranja baze podataka i Audit objekte klijentskoj aplikaciji putem WCF-a.

Klijentska aplikacija prima rezultate ažuriranja baze podataka i Audit objekte.

Korisnik unosi naredbu "Get" sa odgovarajućim parametrima u konzolu klijentske aplikacije.

Klijentska aplikacija šalje zahtev servisnoj aplikaciji putem WCF-a.

Servisna aplikacija izvršava događaj proračuna na osnovu pristiglih opcija (parametara) i čita podatke iz XML baze podataka.

Servisna aplikacija izračunava minimalnu/maksimalnu potrošnju ili standardnu devijaciju na osnovu podataka za tekuci dan iz baze.

Servisna aplikacija kreira tekstualnu datoteku sa rezultatima proračuna.

Servisna aplikacija šalje tekstualnu datoteku klijentskoj aplikaciji putem WCF-a.

Klijentska aplikacija prima tekstualnu datoteku sa rezultatima i čuva je na definisanoj lokaciji.

Na konzoli klijentske aplikacije se ispisuje poruka o imenu i putanji kreirane tekstualne datoteke.

2.2. Opis dijagrama

Dijagram sa tokom podataka prikazuje interakciju između različitih komponenti sistema tokom prenosa CSV datoteka na server kao i generisanje TXT datoteka na serveru i vraćanja klijentu.

2.3. Opis komponenti

1.**Baza podataka (DataBase)**: XML datoteke koje čuvaju podatke o potrošnji i greškama. Svaka od njih predstavljena je zasebnom XML datotekom.

2.**Serverska aplikacija(Service)**: Obradjuje zahtev korisnika. Parsira CSV datoteke, ažurira bazu podataka, izračunava proračune u zavisnosti od zahteva klijenta, kreira i šalje rezultate korisnicima.

3.**Klijentska aplikacija(UI)**: Pruža korisnički interfejs za unos naredbi i prikaz rezultata. Komunicira sa serverskom aplikacijom preko WCF-a

4.Common: Sadrže definicije klasa Load i Audit koje se koriste za reprezentaciju podataka o potrošnji i greškama, sadrži interfejse za komunikaciju između komponenti.

3. Opis interfejsa

Klijentska aplikacija kreirana je kao konzolna aplikacija i ne zahteva korisnički interfejs

4. Opis tehnologija

Razvojni okvir: .NET Framework

Programski jezik: C#

Komunikacija: WCF (Windows Communication Foundation)

Baza podataka: XML baza podataka, sadrži XML datoteke u koje se upisuju podaci

5. Zaključak

Projekat je uspešno implementiran i omogućava generisanje TXT datoteka na osnovu XML baze podataka o potrošnji električne energije. Klijentska aplikacija komunicira sa servisom putem WCF-a, šalje CSV datoteke, a zatim prima generisane .txt datoteke. Mogući pravci budućeg istraživanja i proširenja zadatka mogu biti prelazak sa XML baze podataka na neku od popularnijih tipova baza podataka poput Oracle, MySQL, SQL Server, ili PostgreSQL kao i implementacija proračuna za tekucu sedmicu, mesec ili godinu.