Fakultet tehničkih nauka

Univerzitet u Novom Sadu

Zadatak 1

Tim 9

Virtuelizacija procesa

Predmetni projekat

Autori:

PR125/2020 Marko Lazić

PR126/2020 Filip Bogdanović

PR124/2020 Aleksandar Simić

PR122/2020 Veljko Kondić

Sadržaj

1.Uvod

Ova aplikacija sluzi za evidenciju prognozirane i ostvarene potrošnje električne energije, namenjena je klijentima koji su kompanije za prenos elektricne energije. Dokumentacija opisuje funkcionalnosti, zahteve i arhitekturu aplikacije, kao i tehnicke implementacione detalje.

2.Funkcionalni zahtevi

2.1 Uvoz podataka

Potrebno je da korisnik moze uvesti podatke o planiranoj i ostvarenoj potrošnji električne energije iz CSV datoteka.

Uvoz se vrši putem korisničkog interfejsa koji omogućava izbor CSV datoteka.

Naziv datoteke se sastoji od tipa datoteke (prog za prognoziranu potrošnju, ostv za ostvarenu potrošnju) i datuma u formatu yyyy_mm_dd_hh.

Podaci u datotekama se sastoje od sata i iznosa potrošnje u mW/h.

Za svaku nevalidnu datoteku se kreira Audit objekat koji se upisuje u bazu podataka sa odgovarajućom greškom.

Validni podaci se čuvaju u objektima klase Load i upisuju se u bazu podataka. Prate se podaci o prognoziranoj i ostvarenoj potrošnji.

Za svaku obrađenu CSV datoteku se kreira objekat ImportedFile i upisuje se u bazu podataka.

2.2 Proracun odstupanja

Nakon upisa podataka u bazu podataka, vrši se proračun odstupanja između prognozirane i ostvarene potrošnje po satu.

Proračun se vrši samo za one objekte koji imaju podatke o prognoziranoj i ostvarenoj potrošnji.

Može se izračunati apsolutno procentualno odstupanje ili kvadratno odstupanje, u zavisnosti od podešavanja u App.config datoteci.

Izračunata odstupanja se upisuju u bazu podataka za svaki red pojedinačno.

3.Arhitektura aplikacije

Komunikacija između klijentske aplikacije i servisa se obavlja putem WCF (Windows Communication Foundation) tehnologije.

Koristi se XML Baza ili In-Memory baza podataka u zaisnosti od podešavanja u App.config datoteci.

Rad sa datotekama se implementira uz održavanje memorije i korišćenje Dispose paterna.

Poslovna logika i parsiranje datoteka se obavljaju na servisnoj strani aplikacije.

Slanje datoteka između klijenta i servera se vrši korišćenjem MemoryStream objekata.

Aktiviranje događaja ažuriranja baze podataka se vrši korišćenjem Event-a i Delegate-a.

Aplikacija radi na .NET Framework – u verzije 4.7.2

Navedena dokumentacija daje pregled funkcionalnosti, zahteva i arhitekture aplikacije za evidenciju prognozirane i ostvarene potrošnje električne energije. Takođe, daje smernice za implementaciju, tehničke detalje i zahteve koji treba da se ispune. Detaljni koraci za korišćenje aplikacije, kao i detaljan opis arhitekture, treba da budu dostupni u posebnim dokumentima, kao što je User manual i dokumentacija arhitekture aplikacije.Radi održavanja memorije, koristi se “Dispose pattern”. Takođe koristimo i “Event” i “Delegate”, koji nam pružaju veću fleksibilnost u procesu generisanja datoteka.

Server i klijent su realizovani kao konzolne aplikacije.

4. Opis interfejsa

1) Dictionary<int, Load> LoadDataFromCsv(string recievedMessage)

- učitava podatke iz unete csv datoteke, dodaje ih u listu učitanih Load elemenata u koliko oni ne postoje, ako postoje ažurira podatke

2) void LoadDatabaseEntry(List<Load> list)

- upisuje Load podatke u izabranu bazu podataka

3) void ImportedFileDatabaseEntry(List<ImportedFile> list)

- upisuje ImportedFile podatke u izabranu bazu podataka

4) void AuditDatabaseEntry(List<Audit> audits)

- upisuje Audit podatke u izabranu bazu podataka

5) void CalculateDeviation()

- računa odstupanja između prognozirane i ostvarene potrošnje na osnovu odabira u app.config

6) bool SendFiles(Stream stream)

- parsira datoteku poslatu od strane klijenta, zatim pozove upisivanje u fajl

7) Dictionary<int, Load> ReadXmlFile(string filePath)

- čita Load podatke iz xml fajla

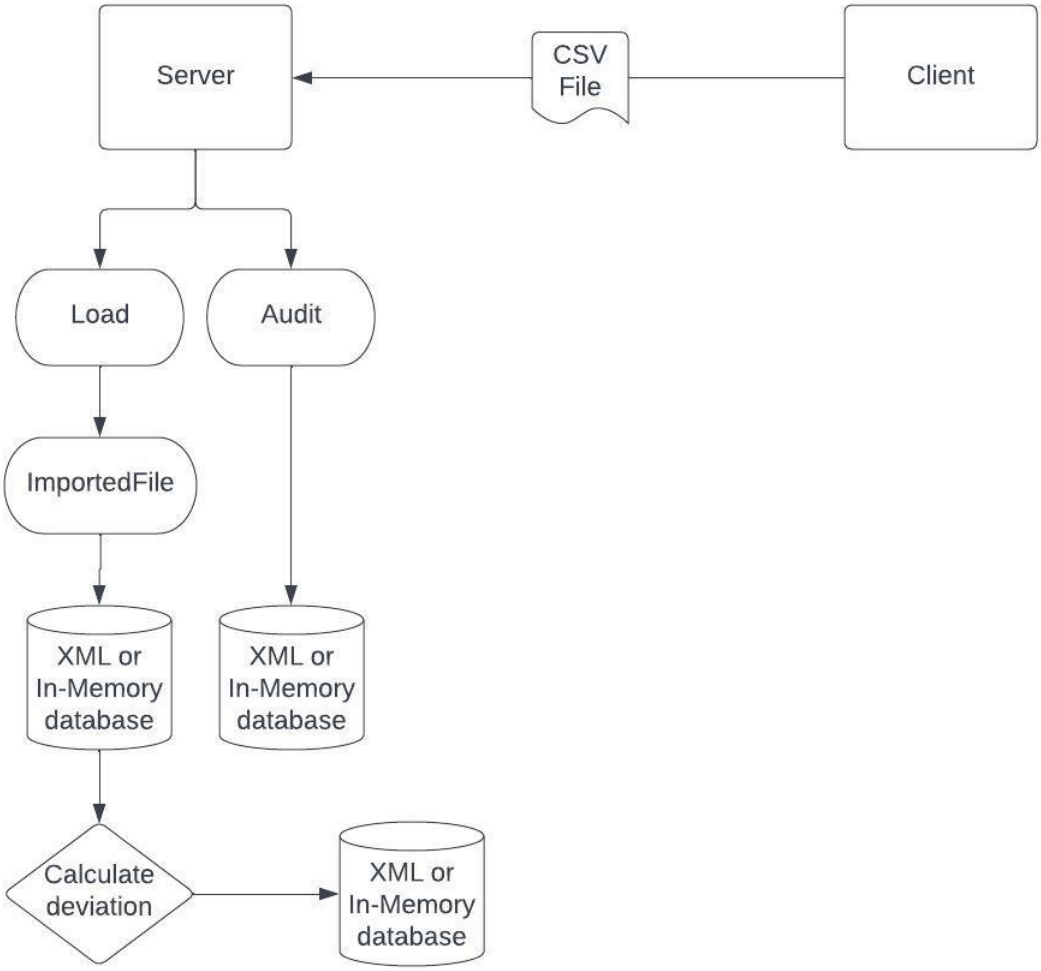
8) Dictionary<int, ImportedFile> ReadImportFile(string filePath);

- čita ImportedFile podatke iz xml fajla

9) Dictionary<int, Audit> ReadAuditFile(string filePath);

- čita Audit podatke iz xml fajla

5.Dijagram



Zaključak

Moguće je proširiti program na osnovu zahteva klijenta. Primena veštačke inteligencije bi značajno usavršila procenjivanje vrednosti na osnovu unetih podataka.