



Fakultet tehničkih nauka
Univerzitet u Novom Sadu

Kreiranje CSV datoteka na osnovu XML baze podataka

Virtuelizacija procesa

Zadatak 2.

PR97/2020 Milorad Maksić

PR103/2020 Anabela Zonai

PR87/2020 Nevena Ćulibrk

PR95/2020 Kristina Sretenović

Uvod

Dokumentacija pruža detaljan pregled aplikacije za kreiranje CSV datoteka na osnovu XML baze podataka, koja je razvijena za potrebe klijenta, kompanije za prenos električne energije. Aplikacija ima za cilj generisanje jedne ili više CSV datoteka sa podacima o prognozama i ostvarenjima potrošnje električne energije, koristeći XML datoteku kao ulaz.

U ovoj dokumentaciji će biti predstavljeni korisnički zahtevi, poslovna logika i model podataka koji čine osnovu aplikacije. Takođe, opisane su tehnički i implementacioni zahtevi, kao i preporuke za rad sa bazom podataka i datotekama u budućem istraživanju.

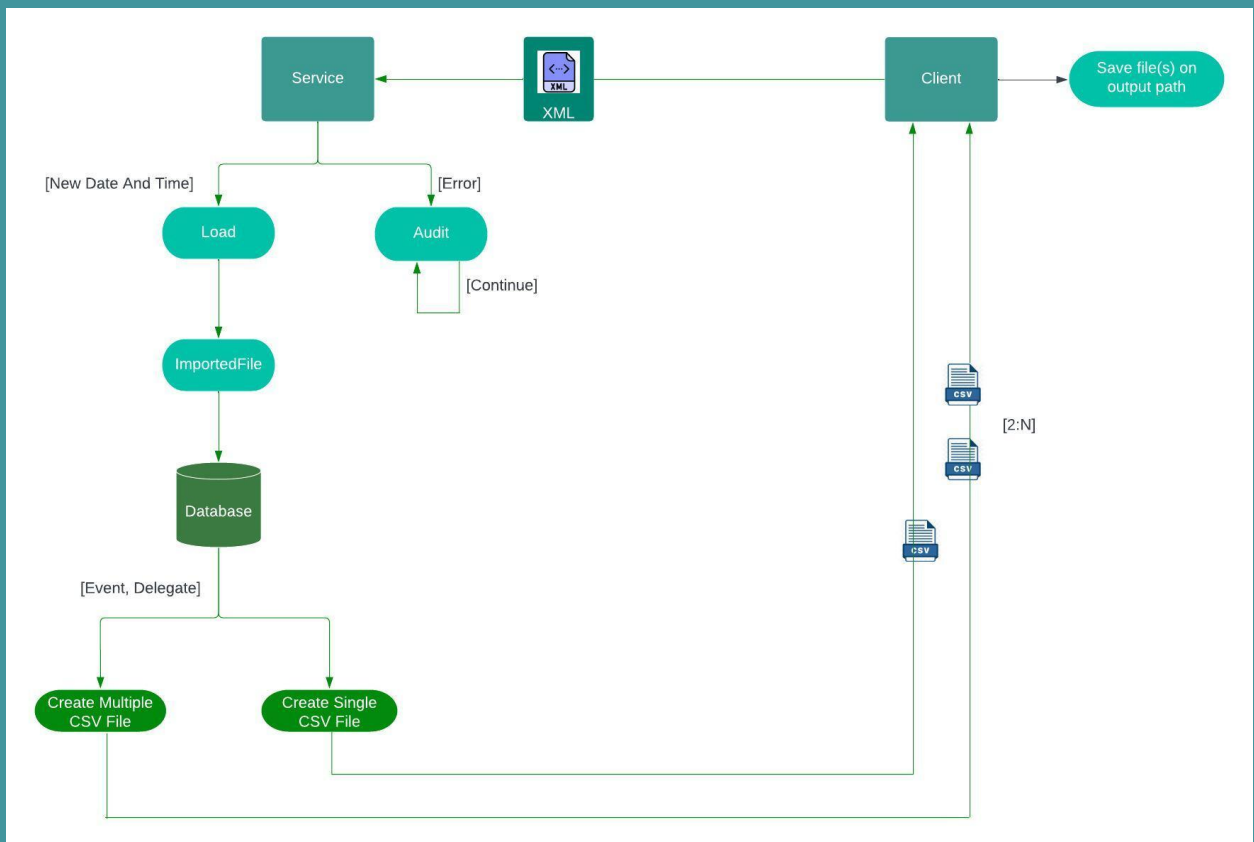
Opis projektnog zadatka

Arhitektura aplikacije je organizovana višeslojno, uključuje bazu podataka implementiranu kao In-Memory baza podataka (DataBase), servisni sloj (Service) i korisnički interfejs (Client) u obliku konzolne aplikacije. Komunikacija između klijentske aplikacije i servisa ostvaruje se putem WCF-a (Windows Communication Foundation), pružajući siguran i pouzdan prenos podataka. Klijent je dužan da preuzme fajl "LOAD_DATA.xml", smesti ga u folder i zatim unese ispravnu putanju do njega.

Važan deo aplikacije je i obrada XML datoteke, gde se na osnovu pristigle datoteke generišu objekti klase "Load" koji predstavljaju podatke o prognoziranoj i ostvarenoj potrošnji električne energije po datumima i satima. Ukoliko se desi greška u podacima, kreira se objekat "Audit" koji se takođe upisuje u bazu podataka. Nakon obrade svih podataka, formira se događaj koji pokreće kreiranje jedne ili više CSV datoteka, u zavisnosti od podešavanja definisanih u konfiguracionoj datoteci (App.config). Ako je podešeno na "True", kreira se više CSV datoteka, u suprotnom, samo jedna sa svim podacima. Podatke je potrebno pročitati iz fajla koji se otvara putem Microsoft Excel-a.

Radi održavanja memorije, koriste se "Dispose pattern" u okviru "Garbage collector-a". Takođe, koristimo "Event" i "Delegate", koji nam pružaju veliku fleksibilnost u procesu generisanja datoteka.

Arhitektura (dijagram) projekta



Slika 1-arhitektura aplikacije

Opis dijagrama aktivnosti

Klijent šalje podatke serveru u vidu XML datoteke. Server prima i obrađuje podatke iz XML datoteke i priprema ih za dalju obradu. Ukoliko klijent pošalje nevalidne podatke, dolazi do greške (implementiranog Error-a) i tu se prekida trenutni tok podataka i prelazi u sledeći ciklus, tj. obrađuju se sledeće torke podataka (sledeći objekat). Na konzoli se ispisuje ID podatka kod kojeg je došlo do greške. U suprotnom, ako su poslani podaci ispravni onda server obrađuje podatke (Load), daje informacije o tome da su podaci obrađeni i dostavljeni (ImportedFile). Dostavljeni podaci se smeštaju u bazu podataka (Database) koja je implementirana u vidu ClassLibrary. U zavisnosti od Event-a i Delegate-a kreira se jedan ili više CSV fajlova, koji se potom šalju nazad klijentu, i on ih čuva u izlaznom (output) fajlu.

Opis komponenata aplikacije

Common - Realizovan je kao ClassLibrary koji sadrži klase i interfejs.

Klase:

1. Audit - koristi se u slučaju da se uoči greška, tako što se Audit objekat upisuje u bazu. Greška se javlja ako je podatak nevalidan, ako podaci nedostaju ili ako postoji greška u strukturi XML datoteke. Za Audit klasu je vezana i klasa MessageType iz foldera Enum gde su navedeni tipovi poruka: Info, Warning i Error.
2. ImportedFile - koristi se za opisivanje fajlova koji se obrade. Kreira se objekat ImportedFile i upisuje u bazu.
3. Load - opisuje obradu podataka. Kreira se objekat Load i upisuje u bazu.
4. FileData - klasa kojom se razmenjuju podaci između servera i klijenta. Ona sadrži MemoryStream: vrsta memorije u koju se mogu zapisati podaci kao da su u datoteci.

Interface:

IDataExport - služi za povezivanje klijenta i servera preko WCF-a.

Sadrži četiri funkcije:

1. ProcessFile() - služi za obradu fajla. Kao parametar prima objekat klase FileData
2. GetFiles() - služi da generiše fajlove na osnovu podataka u InMemory bazi.
3. GetAudits() - vraća listu svih audit zapisa iz InMemory baze.
4. GetImportedFile() - vraća listu svi imported datoteka iz InMemory baze.

DataBase - Realizovan kao ClassLibrary koji sadrži tri klase.

1. *InMemoryAudit* - implementacija InMemory baze podataka za skladištenje audit zapisa. Omogućava osnovne operacije kao što su čitanje, dodavanje i ažuriranje audit zapisa.

2. *InMemoryImpFile* - implementacija InMemory baze podataka za skladištenje informacija o Imported datotekama.

Ova klasa omogućava osnovne operacije kao što su čitanje, dodavanje i ažuriranje informacija imported datotekama u InMemory bazi.

3. *InMemoryLoad* - implementacija InMemory baze podataka za skladištenje informacija o load objektima.

Ova klasa omogućava osnovne operacije kao što su čitanje, dodavanje i ažuriranje informacija o load objektima u InMemory bazi.

Sve funkcije koriste singleton obrazac kako bi se osiguralo da postoji samo jedna instanca baze podataka.

Klijent – implementiran je kao konzolna aplikacija koja će preko *App.config* da primi putanju fajla kojeg šalje serveru. Klijent prvo proverava da li se na datoj putanji zaista nalazi fajl, učitava ga, čita ga, i smešta u *MemoryStream*.

Sadrži sledeće klase:

1. *FileFetcher* – klasa služi za dobavljanja fajla sa putanje iz *App.config* i smeštanja u *MemoryStream*
2. *FileInUseChecker* – klasa služi da proveriti da li se fajl koristi (da li je otvoren, proverava i prava pristupa)
3. *FileSender* – klasa služi da pošalje podatke serveru preko proxy
4. *FileWatcher* - klasa služi da kreira output datoteku

Program.cs – Uspostavljanje veze sa serverom, i poziv funkcija iz gore navedenih klasa i za dobavljanje novih podataka.

Server - realizovan je kao konzolna aplikacija

Program.cs - kreira se instanca *ServiceHost* klase za pokretanje WCF servisa. Ako se upešno pokrene ispiše se poruka.

App.config - podešavamo parametre konfiguracije. Na osnovu key određujemo da li želimo da se kreira jedna ili više CSV datoteka.

DataExportService - servis za obradu pristigle XML datoteke od strane klijenta i generisanje CSV datoteka na osnovu tih podataka.

Nakon pristizanja XML fajla vrši se njegova obrada. Fajl se parsira i podaci se pretvaraju u objekte klase *Load*. Ukoliko su podaci validni, oni se upisuju u InMemory bazu podataka, a u suprotnom kreira se audit zapis sa greškom koja se takođe upisuje u InMemory bazu podataka. Kada se svi podaci obrade oni se zapisuju u *ImportedFile* i dodaju u InMemory bazu. Potom se generišu CSV datoteke na osnovu podataka iz InMemory baze. Na osnovu delegata, odnosno načina na koji je podešena konfiguracija generiše se više ili jedna

CSV datoteka. U servisu postoje i metode `GetAudits()` i `GetImportedFiles()` koje će klijentu proslediti određene podatke iz baze.

Opis tehnologija

Za implementaciju projekta “Kreiranje CSV datoteka na osnovu XML baze podataka” korišćene su sledeće tehnologije:

- Programski jezik C# korišćen je za implementaciju servisa i klijentske aplikacije
- WCF (Windows Communication Foundation) korišćen je radi ostvarenja komunikacije između servisa i klijentske aplikacije
- In-Memory baza podataka za čiju se implementaciju može koristiti struktura podataka poput Dictionary ili ConcurrentDictionary
- Konfiguracijske datoteke koje služe za čuvanje konfiguracijskih postavki, kao što su putanje datoteka ili postavke baze podataka, u našem slučaju je korišćena konfiguracijska datoteka poput App.config (u slučaju .NET okruženja).

Zaključak sa mogućim pravcima budućeg istraživanja

Proširenje zadatka "Kreiranje CSV datoteka na osnovu XML baze podataka" može obuhvatiti unapređenje validacije XML datoteka, dodatne analitičke funkcionalnosti, korišćenje trajne baze podataka, dodatne interfejse, kao i automatizaciju procesa.

Ovi pravci istraživanja i proširenja mogu poboljšati aplikaciju i omogućiti naprednije korišćenje podataka o potrošnji električne energije.

Reference

- <https://openai.com/blog/chatgpt>
- <https://www.eepsi.ftn.uns.ac.rs/group/virtuelizacija-procesa/discussion>
- <https://learn.microsoft.com/en-us/dotnet/api/system.io.memorystream?view=net-7.0>