VIRTUELIZACIJA PROCESA

PROJEKTNI ZADATAK 1

Članovi tima:

- Luka Đelić PR60/2020
- Momčilo Micić PR69/2020
- Stefan Milovanović PR68/2020
- Dušan Paripović PR76/2020

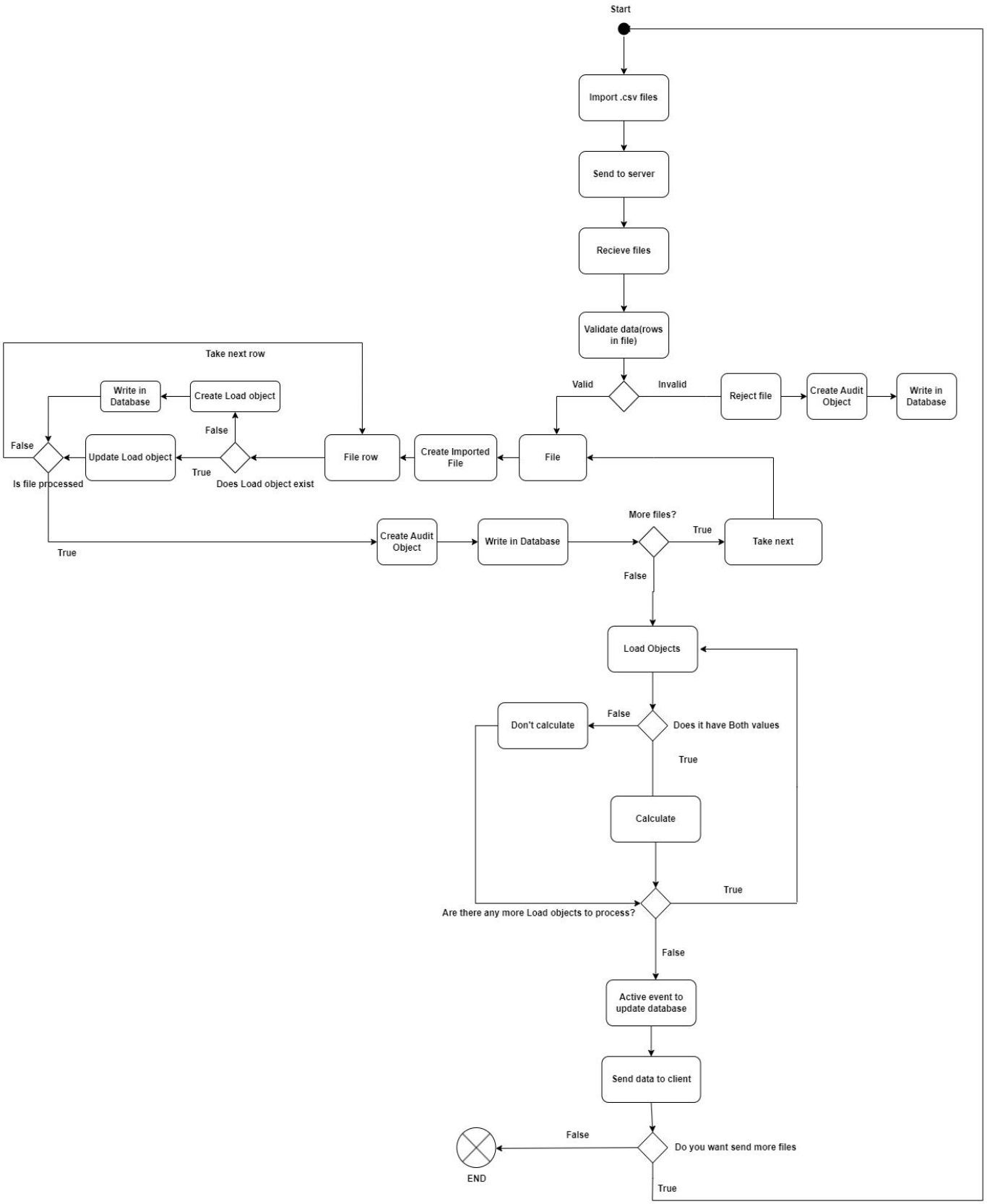
Profesor:

- Bojan Jelačić

Asistenti:

- Zorana Babić
- Zoran Pajić
- Zoran Janković

Katedra za elektroenergetiku i primenjeno softversko inženjerstvo



DIJAGRAM TOKA PODATAKA

OPIS PROJEKTNOG ZADATKA

Na osnovu dijagrama toka podataka u daljem nastavku teksta će biti izložen opis projektnog zadatka i eventualno razjašnjene neke nejasnoće koje su nastale prilikom čitanja dijagrama.

User Interface

Client

Nakon što se pritisne dugme import korisnik ima mogućnost da selektuje određeni direktorijum iz kog će biti uzeti podirektorijumi sa nazivom forecast i measured kao i sve .csv datoteke iz njih čiji naziv počinje sa forecast ili measured.

Korisnik šalje fajlove serveru na obradu pritiskom na dugme send.

Svaki put kada bi se korisnik odlucio da stisne dugme import pre nego što je izvršeno slanje prethodno odabrani direktorijum ce biti obrisan i bice ucitani nove .csv datoteke.

User Interface nakon Importovanja fajlova

U okviru ListView-a su prikazana imena svih učitanih fajlova.

Server

Nakon što je server prirmio datoteke kreće redom da proverava njihovu ispravnost.Važno je napomenuti da server radi sa fajlovima koji su u potpunosti tačni.

Provere koje su implementirane prilikom validacije datoteka:

1. Ispravnost Header-a
    - Header je **ispravan** ukoliko je definisan u obliku TIME_STAMP,FORECAST_VALUE ili TIME_STAMP,MEASURED_VALUE
2. Ispravnost vrednosti broja elemenata za svaku vrstu
    - Datoteka je **ispravna** ukoliko se u svakoj vrsti nalazi tačno 2 elementa prilikom čega u svakoj koloni se prvo nalazi time_stamp dok je u drugoj koloni forecast_value ili measured_value u zavisnosti od same datoteke.
3. Provera naziva datoteke sa TIME_STAMP-om za svaku vrstu

- Datoteka je **ispravna** ukoliko se datum koji je se nalazi u okviru njenog imena poklapa sa datumom iz time_stamp-a za svaku vrstu

4. Provera broja sati za određeni dan
    - Datoteka je **ispravna** ukoliko kada izuzmemo header imamo onoliko vrsta koliko treba da ima sati taj datum u godini.
    - Prilikom ove provere vođeno je računa i o danima prilikom kojih se sat pomera zbog letnjeg računanja vremena ili zimskog.

U slučaju da samo header nije dobar a da su sve ostale provere validne datoteka će se gledati samo što ce prilikom kreiranja Audit object-a(log object-a) stavlja poruka warning.

Ako se desilo da nisu prošle provere 2,3 ili 4 kreira se Audit object sa porukom Error i ta datoteka se celokupna odbacuje.

Obrada

Nakon izvršenih provera ako je datoteka ispravna kreće se automtski u njenu obradu.

Obrada se vrši tako šte se uzima vrsta po vrsta iz datoteke kako bi se izvukle potrebne vrednosti i kreirao Load object koji se kreira samo prvi put za određeni time_stamp nakon toga se vrši azuriranje vec postojećeg object-a.

Nakon sto je datoteka obrađena kreira se Audit object koji se upisuje u DataBase nakon cega se prelazi na obradu sledećeg fajla ako on postoji.

Proračuni

Uspešno kreirani Load objecti se učitavaju iz DataBase-a i vrši se provera da li su u okviru njih obe vrednosti ForecastValue i Measured Value kako bi proračun mogao uspešno da se obavi.

Vrsta proračuna se primenjuje u zavisnosti od toga kako je navedeno u App.config-u

Apsolutno procentualno odstupanje

$$\frac{|ostvarena\ potrošnja - prognozirana\ potrošnja|}{ostvarena\ potrošnja} x100$$

Kvadratno odstupanje

$$\left(\frac{ostvarena\ potrošnja - prognozirana\ potrošnja}{ostvarena\ potrošnja}\right)^2$$

Kada se izvrše proračuni za svaki Load za koji je to moguće aktivira se event za ažuriranje DataBase-a i podaci se vraćaju nazad client-u.

Client izračunate vrednosti može da vidi u okviru novog

direktorijuma(results) koji se nalazi u okviru direktorijuma koji je selektovan prilikom slanja datoteka.

Datoteke koje su vraćene client-u su takođe .csv fajlovi koji imaju naziv result + datum za koji je vršen proračun.

OPIS INTERFEJSA

IFileHandling sadrži samo jednu metodu(SendFiles) prilikom koje se prosleđuje lista učitanih datoteka a kao povratnu vrednost client dobija listu datoteka sa proračunatim vrednostima koja se kasnije smešta u isti direktorijum odakle su učitane datoteke u okviru novog direktorijuma pod nazivom result.

KORIŠĆENE TEHNOLOGIJE

1. .NET Framework
2. Windows Communication Foundation (WCF)
3. Python skripta za kreiranje dodatnih test datoteka
    - https://github.com/dusvn/PythonScript
4. WPF Framework za izradu client UI

ZAKLJUČAK I MOGUĆI PRAVCI ISTRAŽIVANJA

Tema projekta predstavlja jednu veoma zanimljivu ideju postojanja mikroservisa koji treba da uz što manje opterećenje memorije vrši manipulaciju sa datotekama i odrađuje neki od dva proračuna.

Sama tematika projekta je odlična tako da ima dosta pravaca u kojima bi se ova ideja mogla razvijati.

Neki od mogućih dorada:

1. Ukoliko bi naš server radio konstantno i trebao da vrsi neke real time proračune morali bismo da se obezbedimo od

otpornosti na otkaze tj. da uzmemo u obzir postojanje server-a replikatora koji će biti u stanju da ukoliko dođe do ispada glavnog servera on vrati odgovore client-u u što kraćem vremenskom periodu uz minimalno kašnjenje.Naravno to ne mora da bude jedan replikator može ih biti više u zavisnosti od toga koliko je bitno da ne dolazi to otkaza prilikom korišćenja samog softvera ukoliko bi konkretno ove proračune zamenili sa nekim drugim proračunima koji zahtevaju dosta više vremena.

2. Čuvanje podataka bi moglo da se promeni iz InMemory Database u SQL bazu i tako ubrzamo dosta izvršavanje programa ako bi trebala da se izvrši neka pretraga objekata za koje će se vršiti određei proračuni sa jednim SQL upitom bi to veoma jednostavno izvršili i sam SQL developer bi uradio za nas optimizaciju u pozadini tako što bi preko b stabla vršio poklapanje podataka po nekom određenom polju.
3. Naravno uvek bolja opcija bi bila da podaci koji treba da se obrađuju budu smešteni na cloudu i da server automatski povlači podatke,vrši proračune i šalje podatke klijentu kako ne bi gubili vreme prilikom prenosa podataka i memorijskog zauzeća na samom računaru.Ako bi bili u pitanja milioni proračuna sa pretpostavkom da je svaka datoteka dobra to bi bilo mnogo povratnih datoteka za klijenta,ovako bi mogao da jednostavno povuce sa cloud-a(AWS,Microsoft Azure,IBM Cloud…) samo onaj proračun koji njega zanima i da podaci budu dostupni stalno sa bilo kog mesta u bilo koje vreme.
4. Uvođenje distribuiranog computing-a.U ovom nasem konkretnom primeru mi imamo jedan server koji bi vršio proračune,opet se osvrćemo na to šta bi se desilo ako bi se trebalo milon proračuna odraditi i tu nastupa ovo rešenje

prilikom čega bi naš server koji bi bio klaster računara rešavao jedan problem zajednički prilikom čega bi se gledalo da se ravnomerno rasporedi posao medju njima i tako dosta ubrzamo naše proračune.Naravno ako bi imali klaster računare za neke proračune opet bi mogli da se obezbedimo od otkaza sa još nekim klasterom kako se ne bi desilo da ukoliko naš klaster iz nekog razloga bude onemogućen da se velika moć i brzina obrade traži od jednog računara male moći.