

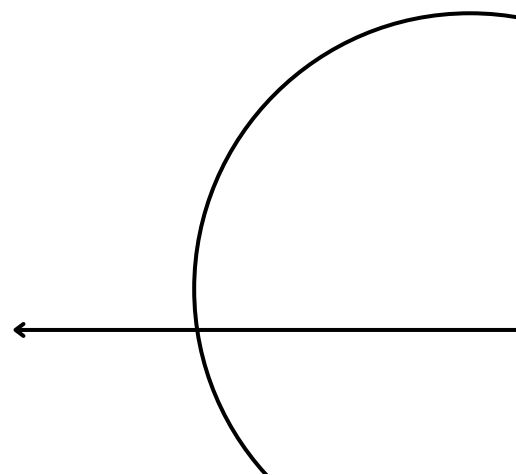


30. AUGUST 2023.

# EVIDENCIJA PROGNOZIRANE I OSTVARENE POTROŠNJE ELEKTRIČNE ENERGIJE

KATARINA MARINKOVIĆ

VIRTUELIZACIJA PROCESA





## OPIS PROJEKTA

Cilj projekta je razviti aplikaciju za evidenciju prognozirane i ostvarene potrošnje elektricne energije.

Zahtevi koje aplikacija mora ispuniti su:

- Uvoz podataka iz CSV datoteka odvojeno za prognoziranu i ostvarenu potrošnju, za svaki sat u danu
- Validacija datoteka i odbacivanje nevalidnih
- Kreiranje objekata Load za svaki sat i upis u bazu
- Racunanje odstupanja izmedju prognozirane i ostvarene potrošnje po formuli iz konfiguracije
- Cuvanje podataka u XML ili in-memory bazu podataka
- Korišćenje WCF servisa za komunikaciju izmedju klijenta i servisa
- Slojevi: baza, servisni sloj, konzolni intefrejs, common
- Izveštavanje i evidencija grešaka u audit tabeli
- Korišćenje dobrih praksi poput Dispose šablona za rad sa datotekama

## KORIŠĆENE TEHNOLOGIJE

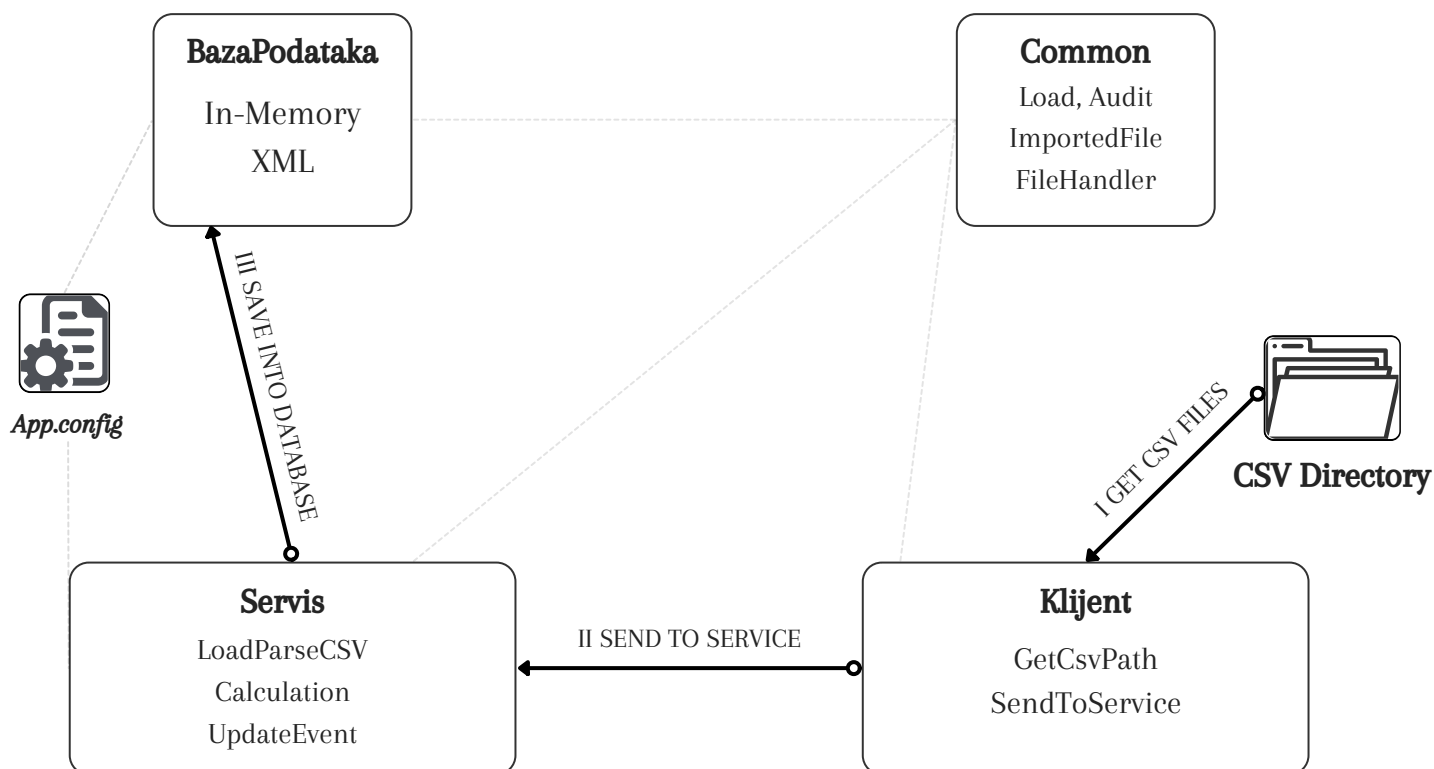
- C#/.NET
- WCF
- XML ili Dictionary baza
- Delegati i događaji za ažuriranje baze

## ARHITEKTURA

- **Konzolna aplikacija** - korisnicki interfejs za unos ulaznih fajlova
- **WCF servisi** - izloženi interfejsi za poslovnu logiku
- **Servisni sloj** - implementacija WCF servisa, sadrži poslovnu logiku
- **Baza podataka** - XML ili In-Memory baza
- **Common** - deljene klase i modeli podataka



## DIJAGRAM PROJEKTA



*Slika 1: Dijagram projekta sa tokom podataka*

## TOK PODATAKA

- Korisnik unosi putanju do CSV fajlova u konzolnoj aplikaciji
- Konzolna aplikacija poziva servis za uvoz podataka
- Servis cita CSV, parsira u objekte i poziva metodu na servisu da sacuva novo pristigle podatke u XML ili InMemory bazu
- Servis zatim poziva proracun da izracuna odstupanja
- Servis cita podatke iz baze, racuna odstupanja i aktivira dogadjaj ažuriranja da ažurira podatke o odstupanjima u bazi



## STRUKTURA PROJEKTA

### BazaPodataka

#### In

Ovo je klasa koja implementira **IBaza** interfejs i predstavlja **in-memory bazu podataka**. Koristi Dictionary kolekcije za cuvanje podataka o (**Loads**, **Audits**) i uvezenim datotekama (**ImportedFiles**).

Implementira metode za cuvanje podataka u recnicima, kao i metodu **SaveDb** za ažuriranje podataka o odstupanjima kroz događaj i delegat.

#### Xml

Ova klasa takodje implementira **IBaza** interfejs i predstavlja implementaciju **XML baze podataka**. Podaci se serijalizuju u XML datoteke za svaku tabelu.

**Save** metoda upisuje podatke u odgovarajuće XML datoteke i automatski kreira datoteke ako one ne postoje.

**SaveDb** ažurira podatke o odstupanjima (bila ona absolutna ili kvadratna) u XML dokumentu koji predstavlja tabelu za čuvanje podataka.

### Common

#### IBaza

Ovo je interfejs koji definiše metode za rad sa bazom podataka. Sadrži metodu:

- **SaveDb** - za ažuriranje podataka o odstupanjima nakon parsiranja i obrade merenja

#### Load

Predstavlja entitet za evidenciju podataka o opterećenju. Sadrži svojstva kao što su ID, vremenska oznaka, prognozirana i izmerena vrednost i vrednosti odstupanja.

#### Audit

Entitet za evidenciju informacija i status obrade, parsiranja, cuvanja podataka. Sadrži svojstva kao što su ID, vremenska oznaka, tip poruke (Error, info, Warning) i sam sadržaj poruke.

#### ImportedFile

Entitet za evidenciju podataka o uvezenim datotekama. Sadrži ID i naziv datoteke koja je uspešno uvezena u aplikaciju.



## MessageType

Enumeracija koja definiše tipove poruka za Audit klasu (Info, Warning, Error).

## FileHandler

Pomocna klasa za rad sa datotekama i stream-ovima. Sadrži stream i naziv datoteke. Implementira Disposable nacin rada sa datotekama radi efikasnog korišćenja i oslobadjanja resursa zauzetih tokom obrade datoteka.

## KONZOLA

Konzolna aplikacija za unos podataka i prosledjivanje na servis. Korisnik unosi putanju do CSV datoteka nakon cega se datoteke pretvaraju u Memory Stream i šalju servisu na obradu.

## SERVIS

Service klasa implementira **IService** interfejs i sadrži sledece glavne funkcionalnosti:

- **LoadSaveDatabase** - Ucitava podatke iz CSV fajla iz poslatih sa klijenta putem memory stream-a, parsira ih i upisuje u odgovarajucu bazu podataka (XML ili In-Memory, u zavisnosti od željenog podešavanja u konfiguracionoj datoteci)
- **Calculation** - Vrší proračun odstupanja između prognozirane i ostvarene vrednosti na osnovu formula iz konfiguracione datoteke
- **SaveDelegate** - Delegat koji se koristi za ažuriranje podataka o odstupanjima u bazi podataka aktiviranje događaja ažuriranja

Konfiguracija iz **App.config** određuje koja baza se koristi i koja formula za odstupanje će biti primenjena prilikom izvršenja proračuna.

**Service** klasa predstavlja srž poslovne logike za obradu podataka o potrošnji energije i integriše sve delove sistema u skladnu celinu komunicirajući putem WCF tehnologije.



## **ZAKLJUCAK**

U ovom projektu je uspešno implementiran sistem za uvoz, obradu i cuvanje podataka o prognoziranoj i ostvarenoj potrošnji elektricne energije u skladu sa pocetnim zahtevima. Korišćenjem višeslojne arhitekture i WCF servisa postignuta je fleksibilna i proširiva platforma za dalji razvoj.

Podaci se uspešno učitavaju, validiraju i obrađuju kroz zasebne servise, pružajući mogućnost lakog dodavanja novih izvora podataka i servisnih metoda u budućnosti.

Praktična primena ovakvog rešenja omogućila bi elektroprivrednim preduzecima efikasnije upravljanje podacima i preciznije predviđanje potrošnje.

## **MOGUĆA PROŠIRENJA I POBOLJŠANJA**

- Dodavanje grafickog interfejsa umesto konzolne aplikacije
- Uvodjenje SQL ili NoSQL baze podataka za skalabilnost
- Napredna analitika i mašinsko učenje nad podacima za preciznije predikcije
- Integracija sa eksternim sistemima preko API-ja
- Dodavanje front-end portala za poslovne korisnike sistema
- Kontinualna isporuka i monitoring performansi i stabilnosti
- Bezbednosni aspekti (enkripcija, autorizacija, zaštita podataka)