



# Virtuelizacija procesa

Evidencija prognozirane i ostvarene  
potrošnje električne energije

Luka Vidaković PR 137/2020

Nemanja Mijonić PR 138/2020

Olivera Čakan PR 71/2020

Milica Klincov PR 70/2020

# SADRŽAJ

1. Uvod .....	4
1.1 Svrha dokumenta .....	4
1.2 Cilj dokumenta .....	4
1.3 Publika dokumenta.....	4
2. Opis projekta .....	4
2.1 Kontekst .....	4
2.2 Cilj projekta.....	4
2.3 Opis funkcionalnosti aplikacije.....	5
3. Arhitektura aplikacije.....	5
3.1 Višeslojna arhitektura .....	5
3.2 Komponente aplikacije.....	6
3.2.1 Baza podataka.....	6
3.2.2 Servisni sloj.....	7
3.2.3 Korisnički interfejs - konzolna aplikacija.....	7
3.2.4 Common sloj.....	7
3.3 Komunikacija putem WCF-a.....	7
4. Import podataka.....	8
4.1 Uvoz podataka iz CSV datoteka.....	8
4.2 Validacija podataka.....	8
4.3 Kreiranje objekata klase Load.....	9
4.4 Upisivanje objekata u bazu podataka.....	9
4.5 Kreiranje objekta ImportedFile.....	9
5. Proračun odstupanja.....	10
5.1 Izračunavanje odstupanja.....	10
5.2 Odabir vrste odstupanja.....	10
5.3 Upisivanje odstupanja u bazu podataka.....	10
5.4 Aktiviranje događaja ažuriranja baze podataka.....	10
6. Model podataka.....	10
6.1 Klasa Load.....	11
6.2 Klasa ImportedFile.....	11
6.3 Klasa Audit.....	11
6.4 MessageType.....	11
7. Implementacija baze podataka.....	12
7.1 XML baza podataka.....	12
7.1.1 Tabele XML baze podataka.....	12
7.2 In-Memory baza podataka.....	12

8. Tehnički i implementacijski zahtjevi.....	13
8.1 Višeslojna arhitektura.....	13
8.2 Rad sa datotekama.....	13
8.3 Komunikacija putem WCF-a.....	13
8.4 Održavanje i nadogradnje.....	13
9. Zaključak.....	14

# 1. Uvod

## 1.1 Svrha dokumenta

Ova dokumentacija ima svrhu da pruži detaljan pregled projekta "Evidencija prognozirane i ostvarene potrošnje električne energije" i da služi kao vodič za razumijevanje arhitekture, funkcionalnosti i implementacije aplikacije. Dokumentacija opisuje različite aspekte projekta i pruža informacije potrebne za razvoj, održavanje i dalje proširenje aplikacije.

## 1.2 Cilj dokumenta

Cilj ovog dokumenta je pružiti jasnu i sveobuhvatnu dokumentaciju projekta. Dokumentacija će opisati osnovne funkcionalnosti aplikacije, arhitekturu projekta, tehnologije koje su korišćene, kao i moguće pravce budućeg istraživanja i proširenja zadatka. Ova dokumentacija će biti korisna za razvojni tim i buduće održavanje aplikacije.

## 1.3 Publika dokumenta

Ova dokumentacija je namenjena članovima projektnog tima, razvojnim inženjerima, programerima, testerima i svim ostalim osobama koje su uključene u razvoj, implementaciju i održavanje aplikacije. Takođe, dokumentacija može biti korisna i za buduće korisnike i istraživače koji žele razumjeti tehničke detalje i mogućnosti aplikacije.

# 2. Opis projekta

## 2.1 Kontekst

Projekat je razvijen za kompanije za prenos električne energije. Cilj aplikacije je omogućiti evidenciju i analizu prognozirane i ostvarene potrošnje električne energije. Aplikacija ima zadatak uvoziti podatke o planiranoj i ostvarenoj potrošnji iz CSV datoteka, izračunavati odstupanje između prognozirane i ostvarene potrošnje, te upisivati rezultate u bazu podataka. Ova evidencija omogućava korisnicima da prate i analiziraju potrošnju električne energije.

## 2.2 Cilj projekta

Glavni cilj projekta je razviti aplikaciju koja će omogućiti klijentu da efikasno upravlja evidencijom potrošnje električne energije. Cilj je automatizovati proces uvoza podataka iz CSV datoteka, obezbediti tačno i pouzdano praćenje odstupanja između prognozirane i ostvarene potrošnje, te omogućiti korisnicima pregled i analizu ovih podataka. Takođe, cilj je obezbediti fleksibilnost aplikacije kako bi se mogla prilagoditi promjenama i proširenjima u budućnosti.

## 2.3 Opis funkcionalnosti aplikacije

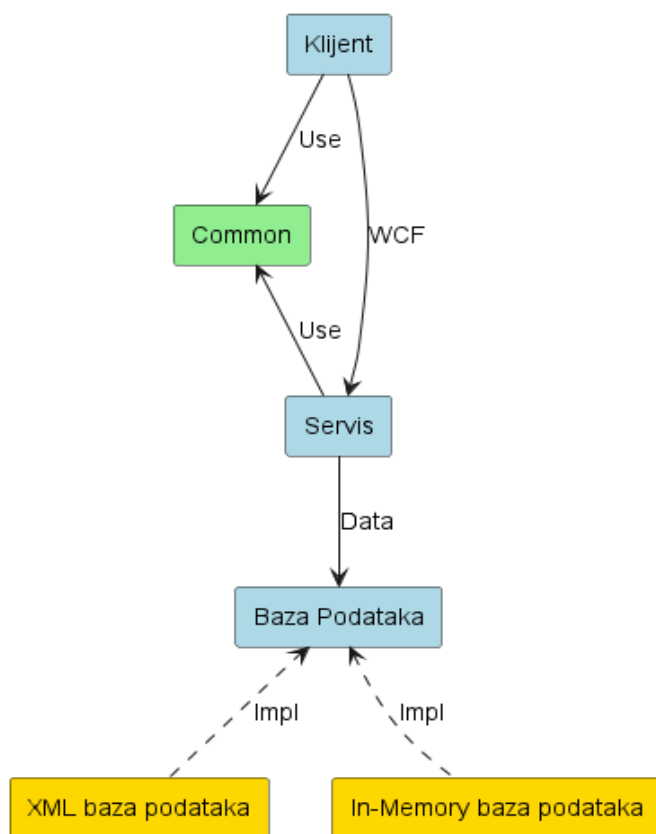
Aplikacija pruža niz funkcionalnosti koje omogućavaju efikasno praćenje i analizu potrošnje električne energije. Ove funkcionalnosti uključuju:

- Uvoz podataka: Korisnicima se omogućava uvoz podataka o prognoziranoj i ostvarenoj potrošnji električne energije iz CSV datoteka. Korisnici mogu izabrati datoteke putem korisničkog interfejsa, a aplikacija automatski čita podatke iz datoteka i vrši validaciju.
- Kreiranje objekata klase Load: Za svaki sat u uvezenim CSV datotekama kreira se objekt klase Load koji sadrži podatke o prognoziranoj i ostvarenoj potrošnji za taj sat. Ako već postoji objekt sa istim datumom i satom, vrši se ažuriranje podataka.
- Praćenje podataka o prognozi i ostvarenju: Aplikacija prati podatke o prognoziranoj i ostvarenoj potrošnji električne energije za svaki sat. Korisnici mogu pregledati ove podatke i pratiti trendove potrošnje.
- Proračun odstupanja: Nakon što su podaci učitani, aplikacija vrši proračun odstupanja između prognozirane i ostvarene potrošnje po satu. Odstupanje se može izračunati kao apsolutno procentualno odstupanje ili kvadratno odstupanje, u zavisnosti od podešavanja. Rezultati proračuna se beleže i upisuju u bazu podataka.
- Upis u bazu podataka: Aplikacija podržava upisivanje podataka u bazu podataka. Podaci o prognoziranoj i ostvarenoj potrošnji, kao i rezultati proračuna odstupanja, se upisuju u odgovarajuće tabele u bazi podataka.
- Auditiranje: Ako neka CSV datoteka nije validna, tj. ne odgovara broju sati u danu, kreira se Audit objekt koji se upisuje u bazu podataka. Audit objekt sadrži informacije o grešci prilikom uvoza datoteke.
- Tehnički zahtevi: Aplikacija je implementirana u višeslojnoj arhitekturi i koristi WCF za komunikaciju između klijentske aplikacije i servisa. Rad sa datotekama je optimizovan kako bi se održavala memorija, a poslovna logika se obavlja na servisnoj strani aplikacije. Podaci se mogu upisivati u XML bazu podataka ili In-Memory bazu podataka, u zavisnosti od podešavanja. Takođe, aplikacija podržava upotrebu događaja i delegata za aktiviranje ažuriranja baze podataka.
- Budući razvoj: U zaključku dokumenta se razmatraju mogući pravci budućeg istraživanja i proširenja projekta. Ovo može uključivati dodavanje novih funkcionalnosti, optimizaciju performansi, podršku za dodatne vrste podataka ili integraciju sa drugim sistemima.

## 3.Arhitektura aplikacije

### 3.1 Višeslojna arhitektura

Aplikacija za evidenciju prognozirane i ostvarene potrošnje električne energije je dizajnirana kao višeslojna arhitektura koja omogućava jasno razdvajanje funkcionalnosti i odgovornosti. Ova arhitektura omogućava fleksibilnost, skalabilnost i lakše održavanje aplikacije.



Arhitektura se sastoji od sledećih slojeva:

- **Korisnički interfejs:** U našem slučaju, korisnički interfejs je implementiran kao konzolna aplikacija koja omogućava interakciju sa korisnikom. Korisnički interfejs šalje zahteve i prima odgovore od servisnog sloja.
- **Servisni sloj:** Servisni sloj predstavlja središnji deo aplikacije koji sadrži poslovnu logiku. Ovde se vrše sve obrade podataka, proračuni odstupanja, upis u bazu podataka i komunikacija sa korisničkim interfejsom. Servisni sloj koristi sloj baze podataka za čuvanje podataka i sloj Common za deljenje zajedničkih resursa.
- **Baza podataka:** Baza podataka je odgovorna za čuvanje podataka o prognoziranoj i ostvarenoj potrošnji, kao i rezultate proračuna odstupanja. Implementirana je kao XML baza podataka i In-Memory baza podataka. XML baza podataka sadrži XML datoteke u koje se upisuju podaci, dok se In-Memory baza podataka implementira kroz Dictionary strukturu podataka.
- **Common sloj:** Common sloj je zajednički sloj koji sadrži kôd koji je deljen između različitih slojeva aplikacije. Ovde se nalaze zajedničke klase, interfejsi i resursi koji se koriste u više delova aplikacije.

## 3.2 Komponente aplikacije

Aplikacija se sastoji od sledećih komponenti:

### 3.2.1 Baza podataka

Baza podataka se sastoji od XML baze podataka i In-Memory baze podataka.

**XML baza podataka:** XML baza podataka sadrži XML datoteke u koje se upisuju podaci o prognoziranoj i ostvarenoj potrošnji električne energije. Svaka tabela je implementirana kroz jednu XML datoteku. Ukoliko XML datoteka ne postoji, automatski se kreira prilikom upisa podataka.

**In-Memory baza podataka:** In-Memory baza podataka je implementirana kroz Dictionary strukturu podataka. Ova vrsta baze podataka se koristi za brži pristup podacima tokom izvršavanja aplikacije i omogućava efikasno čuvanje podataka u radnoj memoriji.

### 3.2.2 Servisni sloj

Servisni sloj je odgovoran za obradu podataka, proračun odstupanja i komunikaciju sa korisničkim interfejsom. Ovde se nalazi poslovna logika aplikacije.

- Obrada podataka: Servisni sloj prima podatke o prognoziranoj i ostvarenoj potrošnji električne energije iz korisničkog interfejsa i upisuje ih u bazu podataka. Takođe, vrši i validaciju podataka kako bi se osigurala njihova ispravnost.
- Proračun odstupanja: Servisni sloj vrši proračun odstupanja između prognozirane i ostvarene potrošnje. Na osnovu tih podataka se generišu izveštaji i statistike koje se prikazuju korisniku.
- Komunikacija sa korisničkim interfejsom: Servisni sloj prima zahteve od korisničkog interfejsa i šalje odgovore nazad. Komunikacija se odvija putem definisanih servisnih metoda.

### 3.2.3 Korisnički interfejs - konzolna aplikacija

Korisnički interfejs je implementiran kao konzolna aplikacija koja omogućava interakciju sa korisnikom. Korisnik može unositi podatke o prognoziranoj i ostvarenoj potrošnji električne energije i dobija povratne informacije. Uneti podaci se prosleđuju servisnom sloju radi dalje obrade.

### 3.2.4 Common sloj

Common sloj je zajednički sloj koji sadrži deljeni kôd, klase i resurse koji se koriste u različitim delovima aplikacije.

- Zajedničke klase: Ovde se nalaze klase koje se koriste u više slojeva aplikacije. Tu se nalaze modeli podataka, pomoćne funkcije, nabrojivi tipovi i slično.

## 3.3 Komunikacija putem WCF-a

Komunikacija između korisničkog interfejsa i servisnog sloja se ostvaruje putem WCF-a.

Korisnički interfejs i servisni sloj implementiraju WCF klijenta i servis, koji omogućavaju slanje zahteva i primanje odgovora.

Jednostavno proširivanje aplikacije u budućnosti se može postići na primer, mogu se dodati nove funkcionalnosti putem novih servisnih metoda ili se može promeniti način komunikacije bez uticaja na ostatak aplikacije.

Korišćenje WCF-a takođe omogućava skalabilnost aplikacije, jer se komunikacija može odvijati između različitih računara ili servera, omogućavajući raspodelu opterećenja i povećanje performansi.

U našem slučaju, WCF se koristi za komunikaciju između korisničkog interfejsa i servisnog sloja. Korisnički interfejs šalje zahteve za unosom podataka, dok servisni sloj vrši obradu tih zahteva i vraća odgovore sa rezultatima.

Komunikacija putem WCF-a pruža sigurnost, pouzdanost i skalabilnost, čineći je pouzdanim izborom za komunikaciju između slojeva aplikacije.

## 4. Import podataka

U cilju omogućavanja uvoza podataka o potrošnji električne energije iz CSV datoteka, razvijen je mehanizam za uvoz, validaciju i čuvanje tih podataka u odgovarajućim objektima klase.

### 4.1 Uvoz podataka iz CSV datoteka

Uvoz podataka iz CSV datoteka je osnovna funkcionalnost aplikacije koja omogućava korisnicima da unesu podatke o prognoziranoj i ostvarenoj potrošnji električne energije. CSV datoteke sadrže informacije o potrošnji po satu za određene vremenske periode.

Mehanizam uvoza podataka iz CSV datoteka je implementiran na sledeći način:

- CSV datoteka se parsira i podaci se izvlače.
- Korisnik bira CSV datoteku koju želi da uveze.
- Podaci se validiraju u skladu sa definisanim pravilima.
- Validirani podaci se koriste za kreiranje objekata klase Load.
- Kreirani objekti se upisuju u odgovarajuću bazu podataka.

### 4.2 Validacija podataka

Validacija podataka je bitan korak prilikom uvoza podataka iz CSV datoteka. Cilj validacije je osigurati da uneseni podaci budu tačni, konzistentni i odgovarajući definisanim pravilima.

Validacija podataka uključuje sledeće korake:

- Provera formata i strukture CSV datoteke.
- Provera ispravnosti vrednosti polja, kao što su datum, vreme i numeričke vrednosti.
- Provera da li su svi potrebni podaci prisutni.

Ukoliko se prilikom validacije otkriju greške, korisnik će biti obavešten o detaljima greške, a uvoz podataka će biti obustavljen.



### 4.3 Kreiranje objekata klase Load

Nakon uspešne validacije podataka, kreiraju se objekti klase Load koji predstavljaju pojedinačne unose o potrošnji električne energije. Svaki objekat klase Load sadrži informacije o datumu, vremenu i izmerenoj potrošnji električne energije.

Kreiranje objekata klase Load obuhvata sledeće korake:

- Izvlačenje informacija iz validiranih podataka, kao što su datum, vreme i potrošnja.
- Kreiranje objekta klase Load sa odgovarajućim vrednostima polja.
- Dodavanje kreiranog objekta u kolekciju objekata klase Load.

### 4.4 Upisivanje objekata u bazu podataka

Nakon uspešnog kreiranja objekata klase Load, sledeći korak je njihovo upisivanje u odgovarajuću bazu podataka. U ovom slučaju, koristimo XML bazu podataka za trajno skladištenje podataka o prognoziranoj i ostvarenoj potrošnji električne energije.

Proces upisivanja objekata u bazu podataka obuhvata sledeće korake:

- Otvaranje ili kreiranje odgovarajuće XML datoteke koja predstavlja tabelu u bazi podataka.
- Konverzija objekata klase Load u XML format.
- Upisivanje XML podataka u odgovarajuću tabelu u XML bazi podataka.
- Zatvaranje XML datoteke i čuvanje promena.

Ovaj proces se ponavlja za svaki kreirani objekat klase Load tokom uvoza podataka iz CSV datoteke. Na taj način, podaci o prognoziranoj i ostvarenoj potrošnji električne energije se trajno čuvaju u XML bazi podataka.

### 4.5 Kreiranje objekta ImportedFile

Prilikom uvoza podataka iz CSV datoteka, takođe se kreira objekat klase ImportedFile koji predstavlja informacije o uvezenoj datoteci. Ovaj objekat sadrži detalje kao što su naziv datoteke i ID.

Kreiranje objekta ImportedFile obuhvata sledeće korake:

- Izvlačenje informacija o datoteci kao što su naziv i ID.
- Kreiranje objekta ImportedFile sa odgovarajućim vrednostima polja.

Kreirani objekat ImportedFile se koristi za evidentiranje informacija o uvezenoj datoteci, što omogućava praćenje i analizu uvezenih podataka u budućnosti.

Ukupno gledano, mehanizam za import podataka omogućava korisnicima da unesu podatke o potrošnji električne energije iz CSV datoteka, vrši njihovu validaciju, kreira odgovarajuće objekte klase Load, upisuje ih u bazu podataka i evidentira informacije o uvezenoj datoteci.

## 5. Proračun odstupanja

U okviru sistema za praćenje potrošnje električne energije, važan korak je proračunavanje odstupanja u potrošnji u odnosu na predefinisane vrednosti ili ciljeve. Proračun odstupanja omogućava identifikaciju situacija gdje je potrošnja izvan očekivanih granica, što može ukazivati na potencijalne probleme ili neefikasnosti u potrošnji električne energije.

### 5.1 Izračunavanje odstupanja

Proračun odstupanja se vrši na osnovu upoređivanja stvarne potrošnje sa referentnim vrednostima. Stvarna potrošnja se dobija iz uvezenih podataka, dok se referentne vrednosti mogu odrediti na osnovu prethodnih perioda, planiranih vrednosti ili drugih relevantnih parametara.

### 5.2 Odabir vrste odstupanja

Prilikom proračuna odstupanja, potrebno je odabrati odgovarajuću vrstu odstupanja koja se želi analizirati. Vrsta odstupanja može biti vezana za ukupnu potrošnju električne energije.

### 5.3 Upisivanje odstupanja u bazu podataka

Nakon izračunavanja odstupanja, rezultati se upisuju u bazu podataka radi dalje analize i praćenja. Odstupanja se čuvaju u posebnoj tabeli koja sadrži informacije o vremenskom periodu, vrsti odstupanja, vrednosti odstupanja i drugim relevantnim podacima.

### 5.4 Aktiviranje događaja ažuriranja baze podataka

Aktiviranje događaja ažuriranja baze podataka predstavlja mehanizam koji omogućava automatsko ažuriranje baze podataka sa novim informacijama o odstupanjima. Kada se događaj aktivira, sistem automatski pokreće rutine za ažuriranje, čime se osigurava da baza podataka uvek sadrži najnovije podatke o odstupanjima.

Aktiviranje događaja ažuriranja baze podataka ima za cilj održavanje baze podataka ažurnom i pružanje najnovijih informacija o odstupanjima. Ovo omogućava korisnicima da imaju uvid u aktuelnu situaciju i da donose informisane odluke u vezi sa upravljanjem potrošnjom električne energije.

## 6. Model podataka

Model podataka definiše strukturu i organizaciju podataka koji se koriste u sistemu. Ovo obuhvata definisanje klasa i njihovih polja koje predstavljaju različite entitete i informacije koje se koriste u sistemu za praćenje odstupanja u potrošnji električne energije.

## 6.1 Klasa Load

Klasa Load predstavlja entitet koji sadrži informacije o odstupanjima u potrošnji električne energije za određenu mernu tačku. Ova klasa ima sledeća polja:

- ID - identifikator
- Timestamp – vremenski trenutak
- ForecastValue – prognozirana vrijednost
- MeasuredValue – izmjerena vrijednost
- AbsolutePercentageDeviation – apsolutno procentualno odstupanje
- SquaredDeviation – kvadratno odstupanje
- ForecastFileId - identifikator
- MeasuredFileId - identifikator

Klasa Load omogućava čuvanje i manipulaciju podacima o odstupanjima u potrošnji električne energije za dalju analizu.

## 6.2 Klasa ImportedFile

Klasa ImportedFile predstavlja entitet koji predstavlja informacije o uvezenim datotekama koje sadrže podatke o odstupanjima. Ova klasa ima sledeća polja:

- ID: Identifikator uvezenog fajla
- FileName: Naziv fajla

Klasa ImportedFile omogućava praćenje informacija o uvezenim datotekama i povezivanje sa odgovarajućim odstupanjima.

## 6.3 Klasa Audit

Klasa Audit predstavlja entitet koji čuva informacije o ažuriranju baze podataka i promenama koje su se dogodile u sistemu. Ova klasa ima sledeća polja:

- ID: Identifikator zapisa u auditu
- Timestamp: Vremenski trenutak akcije
- MessageType: Tip poruke
- Message: Poruka

## 6.4 MessageType

MessageType je enumeracija koja definiše različite vrste poruka koje se koriste u sistemu. Ove vrste poruka mogu obuhvatati informacije o statusu uvoza, greškama ili drugim relevantnim informacijama. Ova enumeracija obezbeđuje jasno definisane vrednosti za vrste poruka koje se koriste u sistemu.

Model podataka omogućava organizovanje i strukturiranje informacija u sistemu, omogućavajući efikasno čuvanje i manipulaciju podacima o odstupanjima u potrošnji električne energije.

## 7. Implementacija baze podataka

### 7.1 XML baza podataka

XML baza podataka se koristi za čuvanje podataka o odstupanjima u potrošnji električne energije. XML format omogućava fleksibilno skladištenje i organizaciju podataka. U okviru XML baze podataka, definišemo sledeće tabele:

#### 7.1.1 Tabele XML baze podataka

- Tabela "Loads": Ova tabela sadrži informacije o odstupanjima u potrošnji električne energije. Polja tabele uključuju ID, Timestamp ID, Timestamp, ForecastValue, MeasuredValue, AbsolutePercentageDeviation, SquaredDeviation, ForecastFileId, MeasuredFileId
- Tabela "ImportedFiles": Ova tabela čuva informacije o uvezenim datotekama koje sadrže podatke o odstupanjima. Polja tabele uključuju ID, FileName.
- Tabela "Audits": Ova tabela čuva informacije o ažuriranju baze podataka. Polja tabele uključuju ID, Timestamp, Message, MessageType.

### 7.2 In-Memory baza podataka

In-Memory baza podataka se koristi za brzo i efikasno čuvanje i pristupanje podacima u sistemu. Ova vrsta baze podataka omogućava čitanje i pisanje podataka direktno u memoriji računara, što rezultira poboljšanom performansom. U okviru In-Memory baze podataka, koristimo sledeće mehanizme za organizaciju podataka:

- Kolekcija "Loads": Ova kolekcija čuva objekte klase Load koji predstavljaju odstupanja u potrošnji električne energije.
- Kolekcija "ImportedFiles": Ova kolekcija čuva objekte klase ImportedFile koji predstavljaju informacije o uvezenim datotekama.
- Kolekcija "Audits": Ova kolekcija čuva objekte klase Audit koji predstavljaju informacije o ažuriranju baze podataka.

Implementacija baze podataka omogućava efikasno i pouzdano čuvanje i pristupanje podacima o odstupanjima u potrošnji električne energije, kao i praćenje promena i aktivnosti u sistemu. XML baza podataka omogućava fleksibilnost u organizaciji podataka, dok In-Memory baza podataka obezbeđuje brz pristup podacima i poboljšanu performansu.

## 8. Tehnički i implementacijski zahtevi

### 8.1 Višeslojna arhitektura

Sistem za praćenje odstupanja u potrošnji električne energije implementiran je koristeći višeslojnu arhitekturu. Ova arhitektura se sastoji od sledećih slojeva:

- Korisnički interfejs: Predstavlja konzolnu aplikaciju koja omogućava korisniku unos podataka i prikaz rezultata.
- Servisni sloj: Ovaj sloj sadrži logiku aplikacije za obradu podataka, kao i pristup bazi podataka. Implementiran je kao WCF servis koji omogućava komunikaciju sa korisničkim interfejsom.
- Baza podataka: Sistem koristi XML bazu podataka za čuvanje informacija o odstupanjima u potrošnji električne energije.

### 8.2 Rad sa datotekama

Sistem podržava uvoz podataka iz CSV datoteka koje sadrže informacije o odstupanjima u potrošnji. Prilikom uvoza, vrši se validacija podataka kako bi se osigurala ispravnost i konzistentnost. Nakon validacije, podaci se čuvaju u bazi podataka.

### 8.3 Komunikacija putem WCF-a

Komunikacija između korisničkog interfejsa i servisnog sloja ostvarena je putem Windows Communication Foundation (WCF) tehnologije. WCF omogućava sigurnu i pouzdanu komunikaciju između klijenta (korisničkog interfejsa) i servera (servisnog sloja), omogućavajući prenos podataka i izvršavanje operacija.

### 8.4 Održavanje i nadogradnje

Sistem za praćenje odstupanja u potrošnji električne energije je dizajniran i implementiran sa ciljem olakšavanja održavanja i nadogradnji. Sistem pruža fleksibilnost u dodavanju novih funkcionalnosti i poboljšanja. Kod je organizovan i dokumentovan na način koji olakšava razumevanje i prilagođavanje sistema u skladu sa budućim zahtevima i promenama u tehnologiji.

Tehnički i implementacijski zahtevi osiguravaju da sistem za praćenje odstupanja u potrošnji električne energije bude pouzdan, skalabilan i efikasan. Komunikacija putem WCF-a obezbeđuje sigurnu i pouzdanu razmenu podataka, dok podrška za rad sa datotekama omogućava fleksibilnost u uvozu podataka.

Pored toga, sistem takođe ispunjava sledeće tehničke i implementacijske zahteve:

- Performanse: Sistem je optimizovan kako bi pružio brz odziv i efikasno rukovao sa velikim količinama podataka. Kroz efikasnu implementaciju algoritama i struktura podataka, obezbeđuje se visoka performansa obrade i manipulacije podacima.
- Skalabilnost: Sistem je skalabilan i može se prilagoditi rastućim zahtevima. Omogućava horizontalno skaliranje dodavanjem više instanci servera kako bi se povećala obrada zahteva i opterećenje.
- Pouzdanost: Sistem je projektovan sa namerom da bude pouzdan i otporan na greške. Koristi se mehanizam za oporavak od grešaka kako bi se obezbedila kontinuirana dostupnost i pouzdanost sistema.
- Dokumentacija: Prateća dokumentacija koja detaljno opisuje arhitekturu, funkcionalnosti i upotrebu.

Ovi tehnički i implementacijski zahtevi osiguravaju da sistem za praćenje odstupanja u potrošnji električne energije bude stabilan, bezbedan i pouzdan. Sistem je prilagodljiv za buduće promene i proširenja, omogućavajući efikasno održavanje i unapređenje funkcionalnosti u skladu sa potrebama korisnika i industrijskim standardima.

## 9. Zaključak

U okviru ovog projekta razvijen je sistem za praćenje odstupanja u potrošnji električne energije koji ima za cilj pružanje efikasnog i preciznog nadzora nad potrošnjom električne energije u različitim okruženjima. Kroz dokumentaciju projekta detaljno smo opisali svrhu, ciljeve, funkcionalnosti, arhitekturu, tehnologije i implementaciju sistema.

Implementirana višeslojna arhitektura omogućava skalabilnost, fleksibilnost i efikasnu komunikaciju između različitih komponenti sistema. Upotreba WCF-a kao tehnologije za komunikaciju pruža pouzdanu i sigurnu razmenu podataka između korisničkog interfejsa i servisnog sloja.

Održavanje sistema je ključno za njegovu pouzdanost i optimalno funkcionisanje. Planirane aktivnosti održavanja uključuju redovno praćenje, ažuriranje softvera i rešavanje problema.

Mogući pravci budućeg istraživanja i proširenja zadatka uključuju integraciju sistema sa pametnim mernim uređajima i IoT tehnologijama radi automatizacije procesa prikupljanja podataka, kao i razvoj naprednih analitičkih algoritama za predviđanje i optimizaciju potrošnje električne energije.

Ukupno gledano, ovaj projekat predstavlja uspešnu implementaciju sistema za praćenje odstupanja u potrošnji električne energije, koji pruža vredne informacije i alate za efikasno upravljanje energijom.