

Statistika devijacije prognozirane i ostvarene potrošnje

Klijent je kompanija za prenos električne energije. Aplikacije treba da se bavi evidencijom statistike odstupanja ostvarene od prognozirane potrošnje električne energije po geografskim oblastima.

Slede korisnički zahtevi.

1. Uvoz podataka o prognoziranoj i ostvarenoj potrošnji

Aplikacija sadrži modul za uvoz podataka o planiranoj i ostvarenoj potrošnji električne energije. Uvoz se vrši iz CSV fajlova. Fajlovi se importuju izborom opcije korisničkog interfejsa. Naziv fajla se sastoji od tipa fajla i datuma. Između ova dva podatka nalazi se donja crta. Tip fajla može biti „prog“ za prognozirano potrošnju i „ostv“ za ostvarenu potrošnju. Datum je u obliku yyyy_mm_dd_hh . Primeri naziva fajlova su prog_2020_05_07.csv i ostv_2020_05_07.csv . Prvi fajl sadrži podatke o prognoziranoj potrošnji, a drugi fajl o ostvarenoj potrošnji, za dan 7. maj 2020. godine.

CSV fajl se može odnositi na jedno ili više geografskih područja. Pojedinačni satni podaci u CSV fajlu sadrže: sat na koji se potrošnja odnosi, iznos potrošnje u mW/h i šifra geografske oblasti. Za svaku geografsku oblast može biti onoliko redova koliko ima sati u danu (24, 23 ili 25). Ako za neku geografsku oblast broj sati ne odgovara broju sati u danu za koji se vrši unos, ceo fajl se odbacuje kao nevalidan, a informacija o tome se upisuje u log fajl. Log fajl sadrži podatke o uspešno izvršenim operacijama (status Info), o upozorenjima (status Warning) i o greškama (status Error). Log fajl može da bude pregledan u aplikaciji.

Učitani podaci se upisuju u bazu podataka. Pojedinačno se upisuju svi redovi pročitani iz CSV fajla, kao satni podaci o potrošnji električne energije za geografsko područje. Posebno se prate podaci o prognoziranoj i ostvarenoj potrošnji. Za svaki podatak o potrošnji prati se i ime fajla iz kojeg je podatak uvezen, kao i vreme uvoza fajla.

2. Proračun statističke devijacije između prognozirane i ostvarene potrošnje za vremenski period

Aplikacija sadrži modul za proračun devijacije između prognozirane i ostvarene potrošnje. Proračun se vrši na osnovu izabranog datumskog opsega i geografskog područja. Apsolutna vrednost odstupanja za jedan sat se izračunava po formuli:

$$\left| \frac{\text{ostvarena potrošnja} - \text{prognozirana potrošnja}}{\text{ostvarena potrošnja}} \times 100 \right|$$

Statističko odstupanje za izabrani vremenski opseg izračunava se na dva načina: putem apsolutne srednje devijacije i kvadratne devijacije. Apsolutna srednja devijacija predstavlja prosečno apsolutno dnevno odstupanje po satima za izabrani vremenski opseg. Kvadratna devijacija predstavlja koren proseka kvadratnih satnih odstupanja za izabrani vremenski opseg. Na korisničkom interfejsu istovremeno se ispisuju apsolutna i kvadratna devijacija, kao i lista apsolutnih vrednosti odstupanja u svakom satu izabranog perioda. Vremenski opseg se zadaje iz korisničkog interfejsa. Na korisničkom interfejsu postoji mogućnost eksporta dobijenih rezultata u XML fajl.

3. Evidencija geografskih područja

Geografsko područje sadrži ime geografskog područja i šifru geografskog područja (skraćeno ime). Šifra geografskog područja se nalazi u CSV fajlovima.

Kroz korisnički interfejs potrebno je da se vrši evidentiranje geografskih područja. Ukoliko se prilikom importa CSV fajla pročita geografsko područje sa šifrom koja ne postoji u bazi podataka, geografsko područje sa tom šifrom se upisuje u bazu podataka, sa nazivom koji je isti kao i šifra.

Tehnički i implementacioni zahtevi

1. U dizajnu i arhitekturi aplikacije potrebno je definisati moguće use case-ove, klase, aktivnosti objekata klasa, interakciju između objekata klasa i softverske komponente aplikacije.
2. Aplikacija treba da bude u multy-component arhitekturi. Aplikacija treba da sadrži najmanje sledeće komponente:
 - baza podataka
 - servisni sloj (opciono može da bude razdvojen na sloj pristupa bazi podataka i sloj poslovne logike).
 - korisnički interfejs (konzolna, Web ili desktop aplikacija)

Slojevi mogu da komuniciraju direktno, odnosno servisni sloj ne mora da egzistira na aplikativnom serveru.

Baza podataka može da bude implementirana kroz neki od SUBP (MS SQL Server, Oracle), kroz neki od embeded sistema za baze podataka (SQLite, MS Access) ili kroz XML.

3. Servisni sloj treba da bude pokriven unit testovima. Pokrivenost unit testova treba da bude najmanje 90%
4. Aplikacija treba da bude razvijana poštujući Agile/Scrum metodologiju razvoja, korišćenjem TFS-a

Kriterijum ocenjivanja

1. Dizajn I arhitektura rešenja
2. Korišćenje Scrum metodologije razvoja – definisanje User Story-a i taskova, planiranje i estimacija
3. Implementacija rešenja
4. CI ciklus
 - a. Build
 - b. UnitTestovi
 - c. Pokrivenost koda testovima