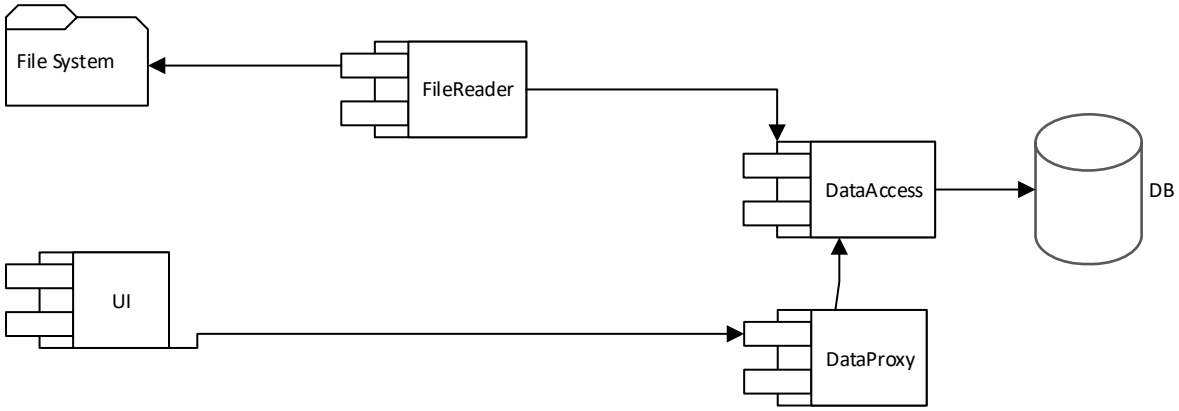
Implementacija sistema za pisanje i čitanje podataka o potrošnji električne energije uz optimizaciju čitanja korišćenjem DATA PROXY modula

Često čitanje podataka iz baze podataka predstavlja „skup“ proces sa stanovišta performansi. Data Proxy komponenta služi za kešitanje podataka u radnu memoriju radi bržeg čitanja. Ova komponenta „cenu“ izmešta iz procesora u radnu memoriju.

U ovom projektnom zadatku potrebno je implementirati module za upisivanje i čitanje ostvarene potrošnje električne energije na satnom nivou, tako da čitanje bude optimizovano korišćenjem Data Proxy komponente.

Implementaciono rešenje treba da sadrži module kao što sledi:



DB – baza podataka
DataAccess – modul pristupa bazi podataka. Služi za perzistenciju podataka.
DataProxy – modul keširanja podataka za čitanje.
FileReader – modul za čitanje podataka iz sistema datoteka i slanje na upis u bazu podataka
UI – Korisnički interfejs

Komponenta FileReader čita podatke o satnoj potrošnji električne energije za jedan dan za određeno geografsko područje. Podaci se čitaju iz sistema datoteka. Datoteke mogu biti u CSV ili XML formatu, po izboru razvojnog tima. Jedna datoteka sadrži podatke o satnoj potrošnji za jedno područje za jedan dan. Podaci koji se čitaju su sledeći:

- Timestamp (datum plus vreme)
- Potrošnja u mW/h
- ID geografskog područja

Ako fali potrošnja za neki sat u danu, odbacuje se upisivanje u bazu podataka i informacija o ovome se upisuje u log fajl.

Dobijeni podaci se u FileReader modulu pretvaraju u modelsku strukturu koja treba da bude definisana dizajnom aplikacije. Takvi podaci se prosleđuju DataAccess komponenti koja vrši upis podataka u bazu podataka. FileReader komponenta takođe preko DataAccess komponente vrši proveru da li za zadato geografsko područje i timestamp već postoji upisana potrošnja. Ako postoji, ta informacija se upisuje u log i preskače se upis u bazu podataka.

U UI komponenti zadaje se datumski opseg za koji se vrši čitanje. Ovaj podatak se prosleđuje DataProxy komponenti. Data proxy proverava da li upit za prosleđeni datum već postoji. Ako ne postoji, podaci se čitaju iz baze podataka,upisuju se u internu keš strukturu i prosleđuju se nazad UI komponenti. Ako upit postoji, podaci se čitaju iz keš strukture, bez kontaktiranja baze podataka, i prosleđuju se nazad UI komponenti. UI komponenta vrši ispis dobijenih podataka.

Keširani podaci u DataProxy modulu treba da egzistiraju najduže 2 sata, nakon čega se brišu.

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 nije potrebno da se vrši evidentiranje geografskih područja. Geografska područja treba da postoje u bazi podataka.

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 sofverske 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 60%
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