

RES Projekat: Load Balancer

DOKUMENTACIJA

Filip Ilić PR 1/2019

Marko Milenković PR 2/2019

Andrija Ristić PR 6/2019

Jovan Miladinović PR 10/2019

1. Tehnologije

Zadatak je odradjen upotrebom programskog jezika Python, verzija 3 ili novija. Za komunikaciju izmedju komponenti je koriscen modul socket.

```
def create_socket(self,adresa):  
    sock = socket.socket(socket.AF_INET,socket.SOCK_STREAM)  
    sock.bind(adresa)  
    sock.listen(1)  
    return sock
```

Slika 1. Pravljenje serverskog socket-a

Modul threading se koristi kako bi bio podrzan paralelan rad vise klijentskih komponenti.

```
os_thread = Thread(target=self.start_worker_app, args=(file_name,))  
os_thread.start()
```

Slika 2. Pokretanje nove niti

Za bazu podataka je koriscen .mdf fajl koji je postavljen na lokalnu instancu Microsoft SQL Server Express, verzije 2019.

SQL Server Express se moze preuzeti sa sledeceg linka:

(<https://docs.microsoft.com/en-us/sql/database-engine/configure-windows/sql-server-express-localdb?view=sql-server-ver15>)

Modul pyodbc se koristi kako bi se povezali za bazu postavljenu na SQL Serveru.

```
connection_str = (  
    r'DRIVER=ODBC Driver 17 for SQL Server;'   
    r'SERVER=.\SQLEXPRESS;'   
    r'Trusted_Connection=yes;'   
    r'DATABASE=Brojila;'   
)
```

Slika 3. Konekcioni string

```
cnxn = pyodbc.connect(connection_str)
```

Slika 4. Konekcija na bazu podataka

Modul se instalira unošenjem jedne od komandi u **Command Prompt**:

```
python -m pip install pyodbc
```

```
py -m pip install pyodbc
```

```
pip install pyodbc
```

Modul freezegun se koristi za testiranje datuma i vremena. Instalira se unošenjem jedne od sledećih komandi u **Command Prompt**:

```
python -m pip install freezegun
```

```
py -m pip install freezegun
```

```
pip install freezegun
```

2. Model podataka i relacije u bazi podataka

Unutar baze podataka postoje 2 tabele, **Informacije** i **Potrosnja**. Tabela **Informacije** sadrži podatke kao što su ID, ime, prezime, ime ulice, broj ulice, poštanski kod i grad o svim prijavljenim brojlama. Primarni ključ (primary key) je ID brojila.

Id	Ime	Prz	Ulica	UBroj	Pbroj	Grad
1	Ratko	Acimovic	PARTIZANSKI BAZA	29	21137	Novi Sad
2	Miroslav	Bajcetic	DJURE JAKSICA	76	21411	Novi Sad
3	Vladislav	Veselinov	CETINJSKA	33	21203	Novi Sad
4	Jovica	Galipovic	SVETOZARA MILETICA	67	21203	Novi Sad
5	Vojislav	Dacevic	VUKA KARADZICA	48	21242	Novi Sad

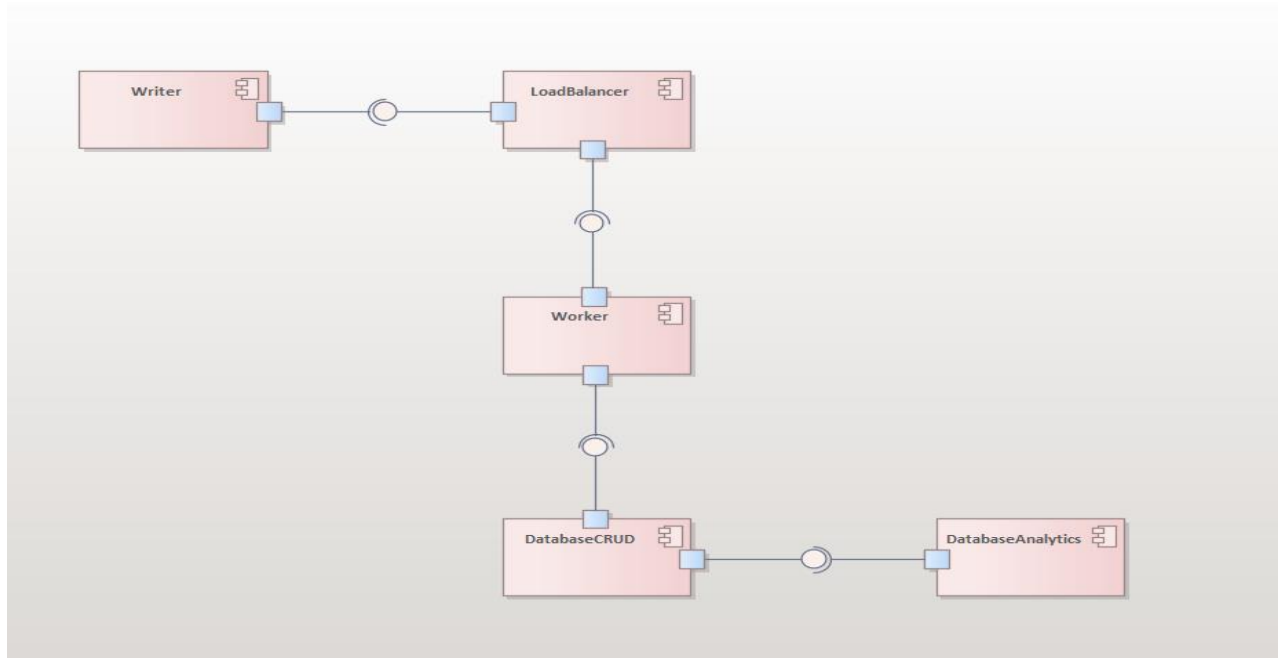
Slika 5. Tabela Informacije

U tabeli **Potrosnja** se nalaze podaci o potrošnji i datumu prijave potrošnje svih prijavljenih brojila, prilikom čega primarni ključ je ID brojila i datum prijave, dok strani ključ (foreign key) koji predstavlja vezu između ove dve tabele je ID brojila.

Id	Ptr	Datum
1	220.00	2000-10-01 11:24:59.000
1	220.00	2000-10-10 11:24:20.000
1	220.00	2000-10-10 11:24:59.000

Slika 6. Tabela Potrosnja

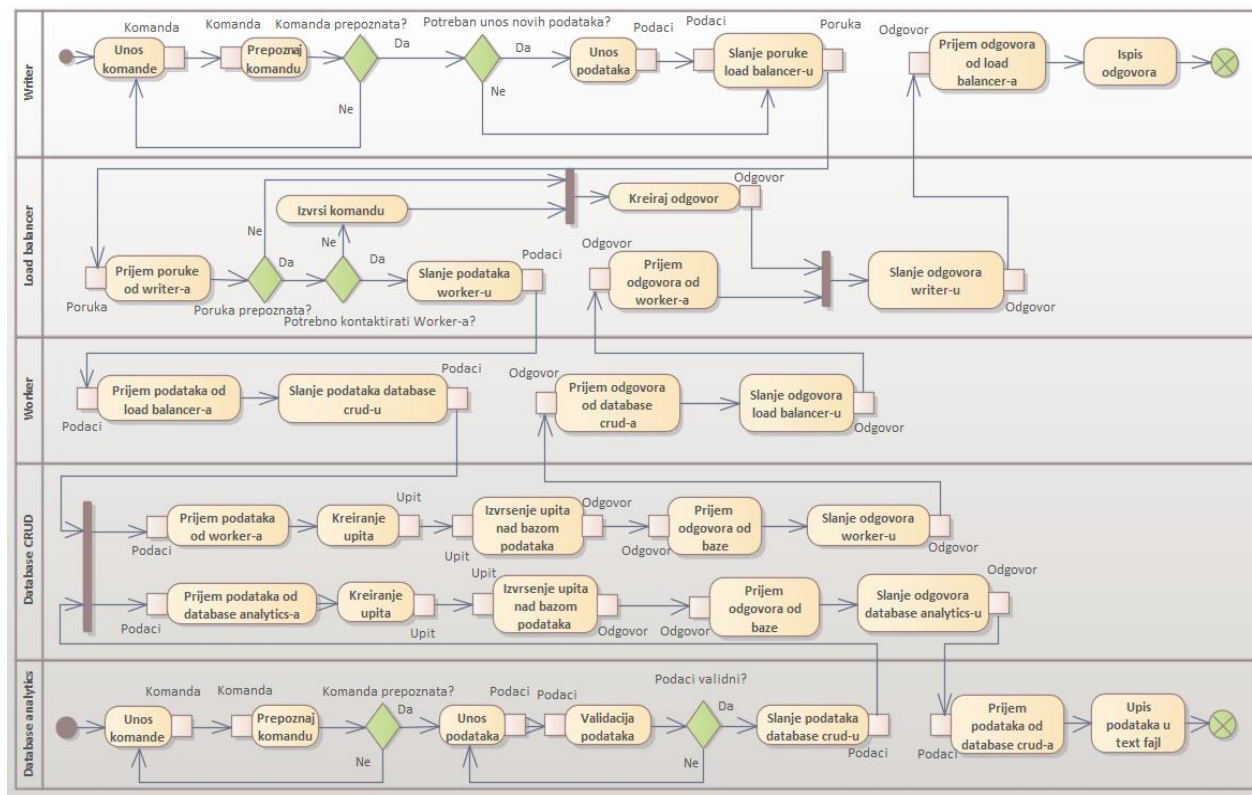
3. Arhitektura sistema



Slika 7. Component diagram

Writer komponenta može isključivo da komunicira sa load balancer-om preko porta 20000, koji može da prepozna 4 različite komande koje dolaze od writer-a. Po potrebi, load balancer šalje podatke worker komponenti kako bi oni bili sačuvani u bazu podataka. Komunikacija između worker-a i load balancer-a se odvija preko porta 21000. Worker podatke upisuje u bazu posredstvom Database CRUD komponente, koja jedina direktno komunicira sa bazom podataka. Database Analytics komponente izvršava upite nad bazom podataka preko Database CRUD komponente. Database CRUD za komunikaciju sa worker-om koristi port 22000, dok za Database Analytics koristi port 23000.

4. Tok podataka i korišćenje aplikacije



Slika 8. Activity diagram

Zadatak aplikacije je da implementira funkcionalnost čuvanja očitanih vrednosti sa kućnih brojala, i dobavljanje statistike o potrošnji po mesecima za određeni grad, odnosno o potrošnji po mesecima za određeno brojilo.

Da bi aplikacija radila potrebno je da pre pokretanja writer komponente već budu pokrenuti load balancer i Database CRUD. Korisnik preko konzole zadaje komandu koju writer šalje load balancer-u. Moguće komande su : sačuvaj vrednost, upali worker-a, ugasi worker-a, kraj rada. Po izboru komande sačuvaj potrebno je uneti id brojila i očitanu vrednost, dok druge komande ne zahtevaju dodatan unos. Po prijemu komande load balancer je izvrši , a ukoliko je lokalni bafer pun, on šalje podatke worker komponenti. Worker komponenta uspostavi vezu sa Database CRUD komponentom, i prosledi joj podatke koji trebaju da se upišu u bazu podataka. Po upisu u bazu, Database CRUD salje potvrdu worker-u da su podaci

upisani, koji vraća odgovor load balancer-u. U zavisnosti od komande koju je izvršio, load balancer izgeneriše odgovor koji vrati writer komponenti, gde se vrši ispis odgovora.

Paralelno je moguće izvršavati upite nad bazom podataka preko Database Analytics komponente, uz uslov da je Database CRUD komponenta već pokrenuta. Klijent bira jedan od dva podržana upita, nakon čega se podaci vezani za upit šalju Database CRUD komponenti, koja izvrši upit nad bazom, i vrati rezultat Database Analytics-u, koji taj rezultat sačuva u tekstualni fajl.