



# *DOKUMENTACIJA PROJEKTOG ZADATKA*

## *P1 - LOAD BALANCER*

*TIM-21:*

*Nikola Bajagić PR43-2019*

*Nemanja Petrović PR44-2019*

*Nemanja Malinović PR108-2019*

*Jovana Lažetić PR111-2019*

# 1. Komponente sistema

## 1.1. *Writer*

Writer (klijent) je komponenta koja služi za upisivanje i slanje novih podataka Load Balancer (server) komponenti, kao i opcije paljenja i gašenja Worker-a sa kojima Load Balancer radi. Šalje podatke o očitanim vrednostima sa kućnih brojila. Podatak koji se šalje sadrži informaciju o ID-u komponente, vrednosti i meseca kada je vrednost očitana.

## 1.2. *Load Balancer*

Load Balancer (server) je komponenta koja raspoređuje poslove i služi za manuelan upis merenja. Posедуje bafer u koji privremeno smešta podatke i nakon 10 prikupljenih vrednosti prosleđuje prvom Workeru koji je dostupan.

## 1.3. *Worker*

Worker je komponenta koja služi za obradu primljenih podataka koje je dobio Load Balancer. Komponenta koristi modul Database CRUD kako bi dobavila podatke iz baze podataka. Nakon obavljenog posla ova komponenta se nalazi u *“sleep”* stanju na određen vremenski period.

## 1.4. *Database CRUD*

Database CRUD komponenta koristi adapter kako bi uspostavlja komunikaciju sa bazom podataka. Omogućava operacije kreiranja, čitanja, ažuriranja i brisanja.

## 1.5. *Database Analytics*

Database Analytics komponenta koristi modul Database CRUD za izvlačenje svih potrebnih statistika iz baze podataka u vidu izveštaja. Omogućava prikaz izveštaja potrošnje po mesecima za tačno definisani grad, kao i prikaz izveštaja po mesecima za konkretno brojilo.

## 2. Tehnologije i opis sistema

Realizacija *Client - Server* arhitekture izvedena je uz pomoć *Socket Programming* tehnologije za mrežno programiranje. Load Balancer komponenta (*Server*) prihvata zahtev za pristiglu konekciju od *Writera (Client)* i za svakog klijenta na strani servera pokreće novu nit koja opslužuje klijenta. Nit postoji sve dok postoji i komunikacija između klijenta i servera. Nakon obustave komunikacije, nit se terminira i prestaje da postoji. Za kreiranje niti koristi se *Thread* tehnologija.

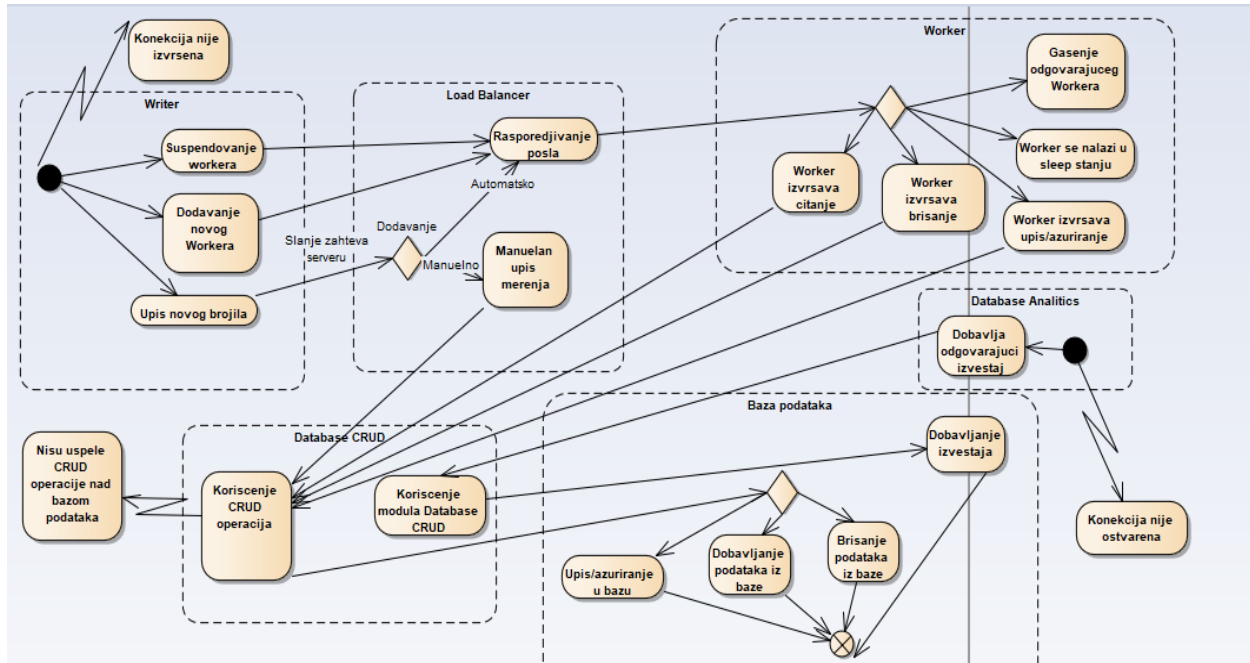
Svaki *Worker* predstavlja jednu nit koja postoji na strani servera i ima zadatak da pomoću *DatabaseCRUD* komponente upiše merenja u bazu podataka. Kako bi se podaci između *Load Balancer* komponente i *Workera* razmenili, neophodno je da postoji red u koji se podaci upisuju. Za ovu namenu iskorišćena je *Queue* tehnologija koja predstavlja linearnu strukturu podataka gde se podaci smeštaju po principu *First In First Out*. Naime, kada se u buffer-u *Load Balancera* nađe deset zahteva, zahtevi se šalju prvom slobodnom *Workeru* upisujući iste u njegov red čekanja. Svaki *Worker* se zapravo izvršava u beskonačnoj petlji gde postoji blokirajuća operacija *queue.get()* koja čeka da se u redu čekanja pojavi zahtev. Nakon punjena reda čekanja, *Worker* poziva metodu za izvršavanje posla. Nakon svakog obrađenog zahteva trenutna *Worker* nit se uspavljuje na pet sekundi kako bi se simuliralo upošljavanje drugih niti za pristiglih drugih deset zahteva. Za ovu namenu upotrebljena je *time.sleep()*.

Posao koji *Worker* treba da uradi, oslanja se na *DatabaseCRUD* klasi. *DatabaseCRUD* mora da ostvari konekciju sa bazom podataka kako bi se CRUD operacije izvršile. Za ostvarivanje konekcije koristi se tehnologija odnosno adapter *MYSQL-connector-python*. Izabran je *Oracle-ov MySQL* sistem za upravljanje relacionom bazom podataka, kao i *MySQL WorkBench* kao interfejs za prikaz baze podataka i tabela unutar iste.

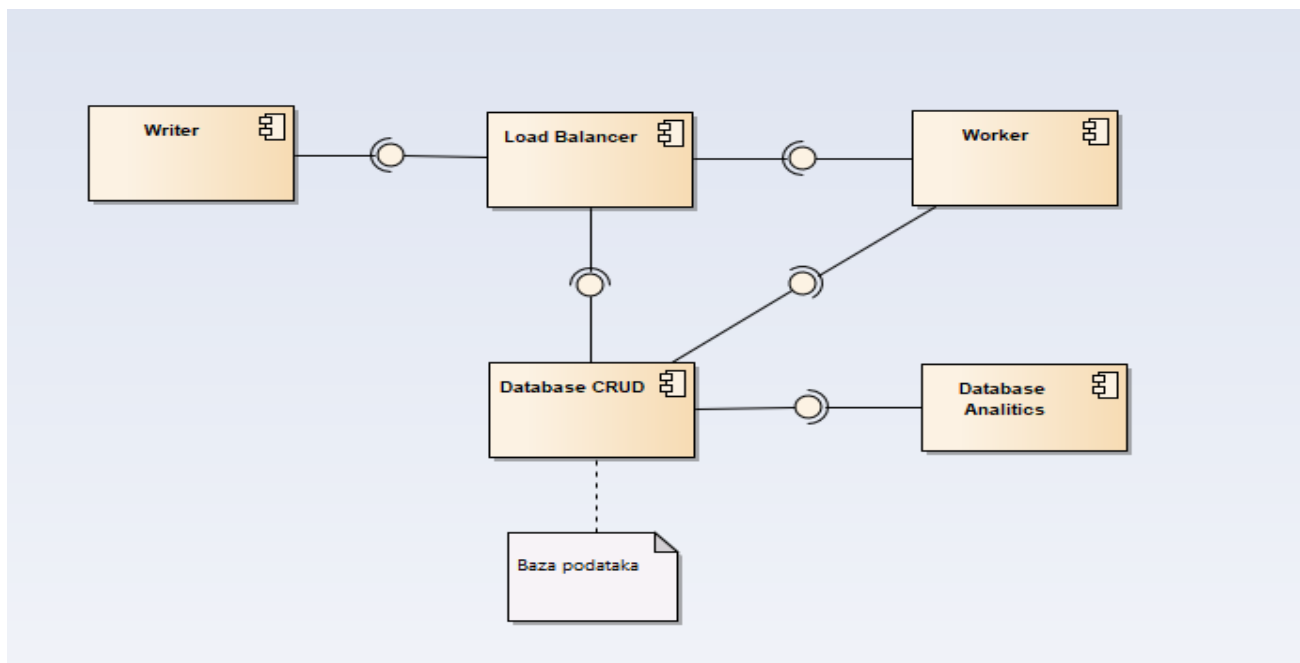
*Database Analytics* komponenta koristi *DatabaseCRUD* modul za dobijanje rezultata o mesečnoj potrošnji po brojilu i po gradu. Kako podaci dobijeni od *DatabaseCRUD* modula su zapravo torke, neophodno je iste isparsirati u vrednosti koje za korisnika predstavljaju čitljiv podatak. Za tu namenu koristi se *Parser* koji uzima torke i od njih pravi izveštaj koji kasnije klasa zadužena za ispis koristi za prikazivanje izveštaja.

### 3. Dijagrami sistema

#### 3.1 Activity dijagram



#### 3.2 Component dijagram



#### 4. Sema baze podataka

