Tim 1 - Load Balancer

Aplikacija Load Balancer

Program se zasniva na klijent/server komunikaciji preko WCF servisa. Klijentsku stranu predstavlja *Writer* dok serversku stranu predstavlja *LoadBalancer* koji prima zahteve od *Writer*-a, a zatim posao obrađivanja objekata Item strukture deli na objekte *Worker* komponente.

Writer komponenta LoadBalancer-u šalje slučajno generisane objekte Item-a uz zahtev za njihov upis u bazu podataka, a korisnik je u svakom trenutku u mogućnosti da pritiskom na <ENTER> zaustavi rad *Writer*-a i iz ponuđenog menija izabere zahtev:

MENU:

1. TURN OFF worker - zahtev za uništavanje postojećeg *Worker*-a

2. TURN ON worker - zahtev za pokretanje novog objekta komponente *Worker*

3. Write Item - zahtev za slanje objekta strukture Item i njegov upis u bazu podataka

- nakon odabira opcije 3, program korisniku nudi odabir koda (CODE) a zatim traži unos vrednosti Item-a (VALUE)

4. List items from interval - zahtev za dobavljanje liste Item-a koji su upisani u bazu u određenom intervalu

- nakon odabira opcije 4, program korisniku nudi odabir koda, očekuje unos id-a *Worker*-a kao i početak i kraj vremenskog intervala predstavljenog u formatu [format -> 2009 - 05 - 08 14:40:52]

5. Exit program - koji omogucava normalan zavrsetak rada aplikacije

Nakon što *Worker* pošalje zahtev pokrenut od strane korisnika, on nastavlja sa slanjem slučajno generisanih Item-a, sve dok korisnik ponovo ne pauzira rad *Writer*-a.

*Writer* pri svakoj akciji poziva operaciju za Log gdje se bilježe sve aktivnosti *Writera* u writerLog.txt fajl.

Struktura Item sastoji se iz polja Code i Value i kao takva šalje se preko WCF komunikacije.

Mogući kodovi za unos su:

CODE\_ANALOG,

CODE\_DIGITAL,

CODE\_CUSTOM,

CODE\_LIMITSET,

CODE\_SINGLENODE,

CODE\_MULTIPLENODE,

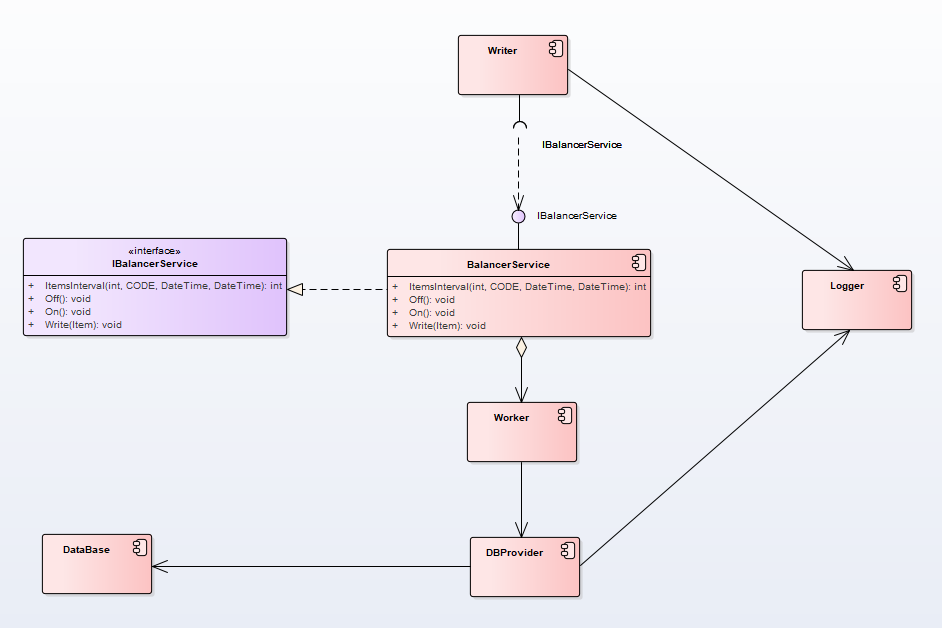
CODE\_CONSUMER,

CODE\_SOURCE.

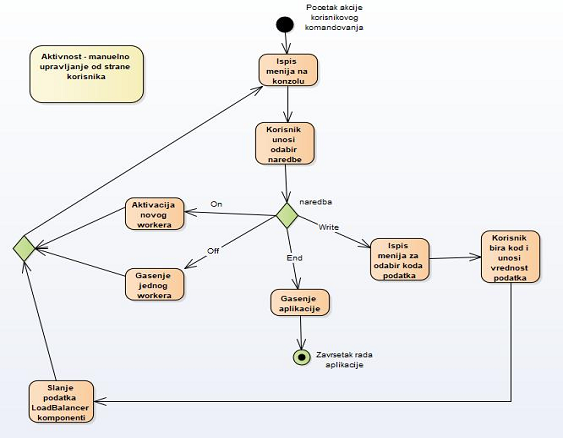
Podaci unutar baze podataka će biti sortirani po *DataSet*-u. DataSet 1 predstavljaće tabelu "One" u koju će biti smješteni podaci sa CODE\_ANALOG i CODE\_DIGITAL kodom, DataSet 2 tabelu "Two" u koju će biti smješteni podaci sa CODE\_CUSTOM i CODE\_LIMITSET, DataSet 3 tabelu "Three" koja će se vezati za CODE\_SINGLENODE i CODE\_MULTIPLENODE kodove i DataSet 4 tabelu "Four" u koju će biti smješteni podacu sa kodovima CODE\_CONSUMER i CODE\_SOURCE.

Serverska strana odnosno *LoadBalancer* komponenta paralelno prima zahteve od klijentske strane i raspoređuje posao upisa u bazu na aktivne komponente klase *Worker*. Ukoliko *LoadBalancer* primi zahtev za upis *Item*-a u bazu, privremeno prosleđeni Item smešta u lokalni *buffer*, dok ga *Worker* ne preuzme radi upisa. *Buffer* je na serverskoj strani organizovan u okviru struktura *Description* i *ListDescription*. *Description* struktura sastoji se od polja int Id, int DataSet i List<Item> listItem. Svi Item-i unutar liste jednog objekta *Description* strukture moraju da pripadaju istom DataSet-u. ListDescription struktura u sebi sadrži jedno polje koje je tipa List<*Description*> u kojoj se nalazi lista od 4 *Description*-a svih *DataSetova* i na taj način LoadBalancer već pri samom pristizanju Item-a na serversku stranu pravi raspodelu. U paraleli *LoadBalancer* pokreće sve aktivne *Worker* objekte po *Round* *Robin* principu i predaje im Item-e iz buffera odnosno *ListDescription* strukture kako bi *Worker* zatim pristupio bazi i uneo sve *Item*-e iz *LD* strukture. Upis i obrada se zasnivaju na provjeri *Deadband*-a, zatim na upisu struktura u bazu podataka. *Deadband* predstavlja odstupanje vrijednosti koju korisnik želi da upiše od vrijednosti iz baze podataka za dati Code. Ukoliko se nova vrijednosti razlikuje za 2% ili više od trenutnih iz baze za dati kod, tada je uslov za upis ispunjen i upis se može izvršiti. Ukoliko se vrši unos *Itam*-a koji sadrži CODE\_DIGITAL vrednost polja *Code* tada provera *DeadBanda* nije potrebna. Pored vrijednosti i koda u tabelu se upisuje i ID *Worker*-a koji je izvršio obradu i vrijeme kada se upis u bazu izvršio. Program sadrži komponentu *Logger*, koja je zadužena za logovanje odnosno upisivanje svih događaja u tekstualni fajl.

Component diagram :



Activity diagram 1:



Activity diagram 2:

Diagram, schematic

Description automatically generated

Članovi tima: Teodora Ruvčeski PR101-2018

Nebojša Gordić PR5-2018