

**RES Projektni zadatak – Dokumentacija Tema:  
Projektni zadatak 4**

**Studenti:**

**Aleksić Milovan**

**Vasiljević Nemanja**

**Gvozdenov Branislav**

**Sremčević Milan**

**Asistent:**

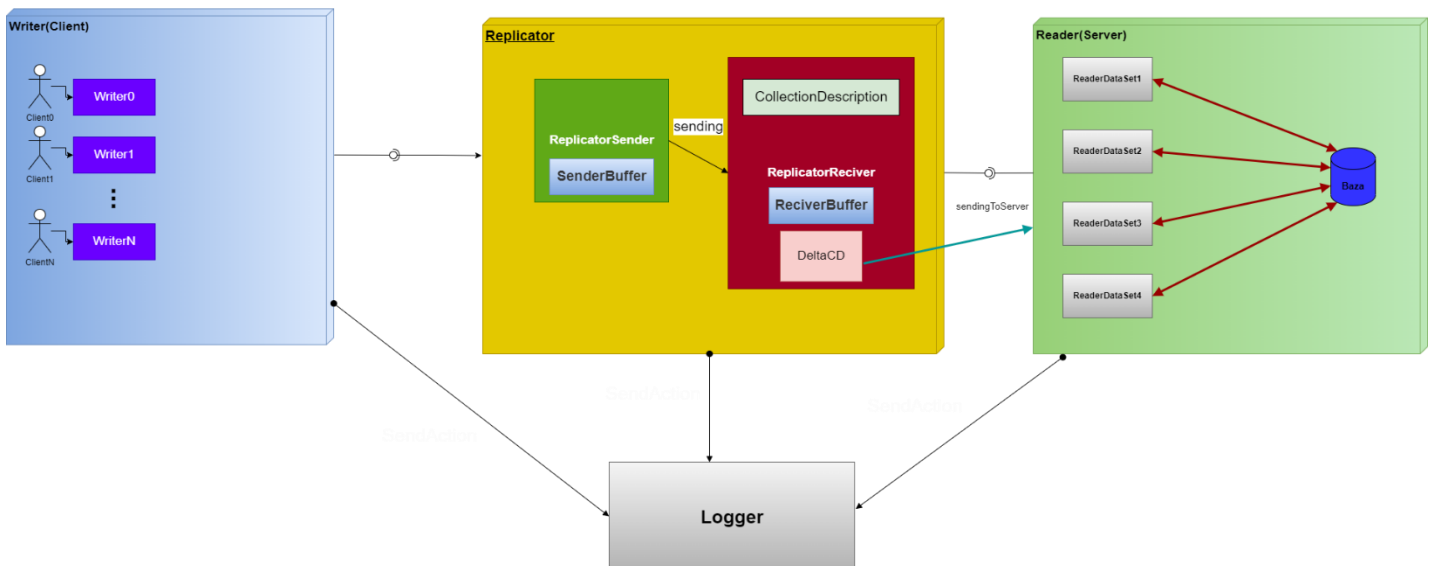
**Dejanović Stefan**

Ovaj projektni zadatak predstavlja rešenje koje se sastoji od 4 osnovne i 3 dodatne komponente.

**Komponente su:**

- **WriterComponent**
- **ClientComponent**
- **ReplicatorSenderComponent**
- **ReplicatorReceiverComponent**
- **ReaderComponent**
- **DataBaseComponent**
- **LoggerComponent**

**Component diagram:**



Čitav program je zbog svoje relativno male veličine i jednostavnosti implementiran kao konzolna aplikacija. Podaci su smesteni u DataModel.py fajlu, a čitavim procesom se komanduje iz Client.py fajla.

U narednom delu dokumentacije biće opisane osnovne metode klasa i podklase osnovnih komponenti:

**WriterComponent:**

Ova komponenta služi za kreiranje i upisivanje novih podataka u sistem kroz Replicator komponente. .

Ova komponenta sadrži sledeće metode:

- **SendData(s)** - Simulira slanje podatka na svake 2 sekunde.
- **SocketConnect()** - Pravi socket koji povezuje Writer komponentu sa ReplicatorSender komponentom.

**ClientComponent:**

Koristi se za komandovanje celim programom . Sastoji se od 6 metoda :

- **Menu()**- pruža korisniku meni iz kojeg može da bira jednu od sledećih funkcionalnosti :
  1. Paljenje novog Writer-a
  2. Gašenje Writer-a
  3. Istorijски izveštaj
  4. Izveštaj o poslednjim vrednostima
- **SendHistoricalRequest(s)** - služi za prosleđivanje podataka koji su potrebni bazi da vrati istorijски izveštaj vrednosti za prosleđeni kod.
- **RecieveHistoricalRequest()** - Kupi i prikazuje istorijски izveštaj iz baze.
- **SendCodeRequest(s)** - služi za prosleđivanje podataka koji su potrebni bazi da vrati izveštaj o poslednjim vrednostima za zadate kodove.
- **RecieveCodeRequest()** - Kupi i prikazuje izveštaj o poslednjim vrednostima za zadate kodove.

**ReplicatorSenderComponent:**

Ovo je relativno jednostavna komponenta koja služi za privremeno cuvanje podataka koje preuzme od Writer komponente I prosleđivanje ReplicatorReciever komponenti. Sadrži jednu klasu - **ClientThread** koja služi za prihvatanje novih klijentskih konekcija u posebnim nitima. Klasa poseduje jednu metodu:

- **Run(Self)** – služi za ostvarivanje konekcije sa Writer komponentom I prosleđivanje podataka Reader komponenti.

**ReplicatorRecieverComponent:**

Koristi se za primanje podataka od ReplicatorSender komponente, pakuje ih u objekat klase **CollectionDescription**, izvršava logiku prođenja da li su podaci za ažuriranje ili dodavanje. Prosleđuje listu objekata **CollectionDescription** klase Reader komponenti.

**ReaderComponent:**

Komponenta koja služi za komuniciranje sa bazom. Sadrži 10 metoda:

- **WriteData(data,database,db)** – služi za upisivanje podataka u bazu.
- **ReadData(code,database,db)** – služi za citanje podataka iz baze.
- **ReadHistory(historicalValue,database,db)** – služi za čitanje istorijskih vrednosti iz baze.
- **CalculateDifference(new,database,db)** – Vrš proračun na osnovu kojeg se razmatra da li će podatak biti upisan u bazu kao ažurirana vrednost.
- **SocketConnect()** – Kreira socket za komunikaciju sa ReplicatorReceiver komponentom.
- **ReceiveData(socket)** – služi za primanje podataka od strane ReplicatorReceiver komponente.
- **SendResponse(result)** – Vraca izveštaj Client komponenti.
- **WriteInDatabase(CDArray,db)** – upisuje u određenu bazu podatak u zavisnosti od toga koji je kod poslat od strane Writer komponente.
- **ReadLastValue(received,db)** – Vraća poslednje vrednosti iz baze za kodove koji su prosleđeni od strane klijenta.
- **ReadFromTableUsingTimeStamp(received,db)** – Vraća vrednosti koje su upisane u tabelu između datuma koje je korisnik prosledio.

**LoggerComponent** predstavlja trivijalnu komponentu koja služi za kreiranje **Log.txt** fajla u kome će se skladištiti informacije o radu sistema, kao i za samo skladištenje istih.

**DataBaseComponent** predstavlja pomoćnu komponentu koja služi za upravljanje bazom podataka.