

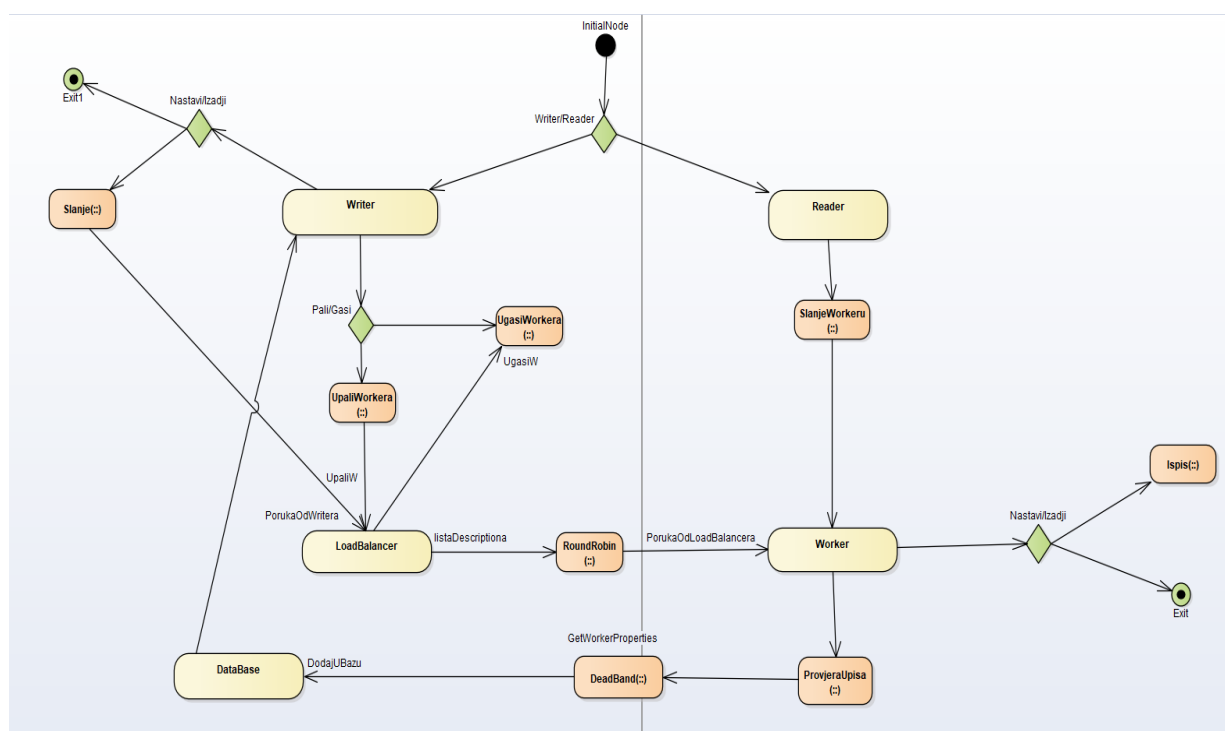
Projektni zadatak

Load Balancer

Uvod :

U projektu je simuliran rad i komunikacija Load Balancer modula sa Writer, Worker i Reader komponentom. Podaci se čuvaju u Microsoft SQL bazu podataka.

Način funkcionisanja komponenti :



Slika1. Activity diagram

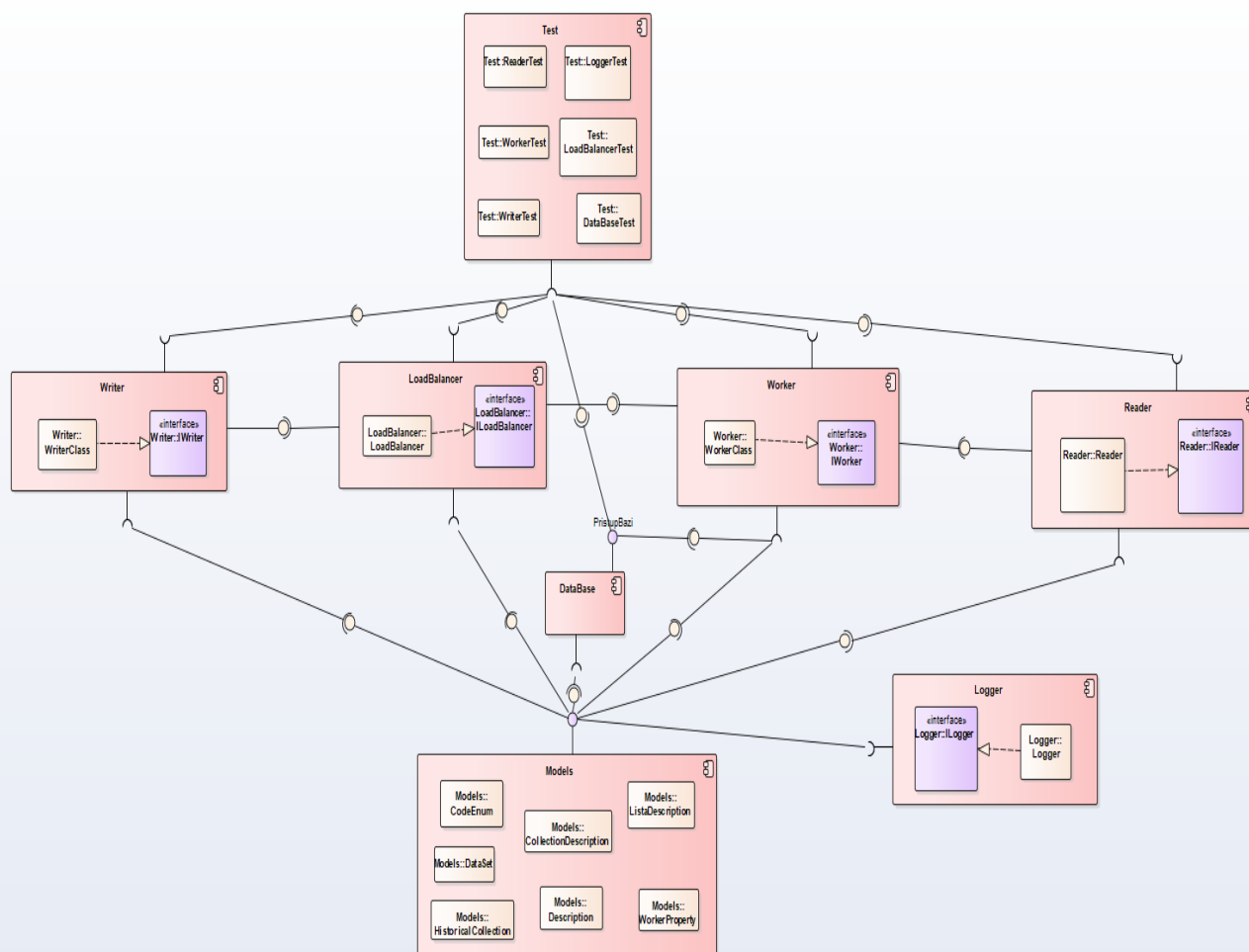
Writer komponenta je konzolni klijent. Ona u svom glavnom programu ima dva Thread-a. Prvi kreiran Thread šalje u pozadini podatke ka Load Balancer komponenti na svake 2 sekunde. Drugi kreiran Thread ispisuje poruku korisniku da unese podatak o paljenju nove Worker komponente ili o gašenju već upaljene Worker komponente. U našem sistemu imamo mogućnost da koristimo 4 Worker komponente. Prilikom iniciranja paljenja od strane Writera šalje se podatak o paljenju prvog slobodnog Workera, a prilikom iniciranja gašenja šalje se podatak o gašenju poslednjeg ugašenog Workera.

Load Balancer je Class library komponenta. Ona prima podatke od Writer komponente i stavlja te podatke u internu strukturu – listu Descriptiona. Zatim te podatke prosleđuje slobodnim Workerima na obradu. Na osnovu stanja iz klase "PaljenjeIgašenje" on zna koliko ima upaljenih Workera i njima po principu Round-Robin raspoređivanja šalje podatke iz bafera.

Worker komponenta je takodje Class library. Ona prima podatke od Load Balancer komponente. Primljene podatke prepakuje u svoju strukturu za rad (CollectionDescription). Zatim dobijene podatke stavlja u bafer koji je tipa lista CollectionDescription-a. Za svaki podatak koji se upiše u bafer se doda timestamp i ključ. Ključ je redni broj Workera koji je upisao podatak u bafer, a timestamp je vreme kad je podatak upisan. Zatim se proverava da li su pristigli podaci popunili strukturu za određeni DataSet. Ako jesu, podaci se dodatno proveravaju za uslov DeadBand od 2% (za sve osim CODE_DIGITAL) i ukoliko vrijednost izadje iz DeadBanda ona upisuje u bazu po određenom ID-u, ako ne ponovo se ceka na novo popunjavanje Data Seta i ponovnu provjeru pri upisu u bazu u vidu Dead Banda.

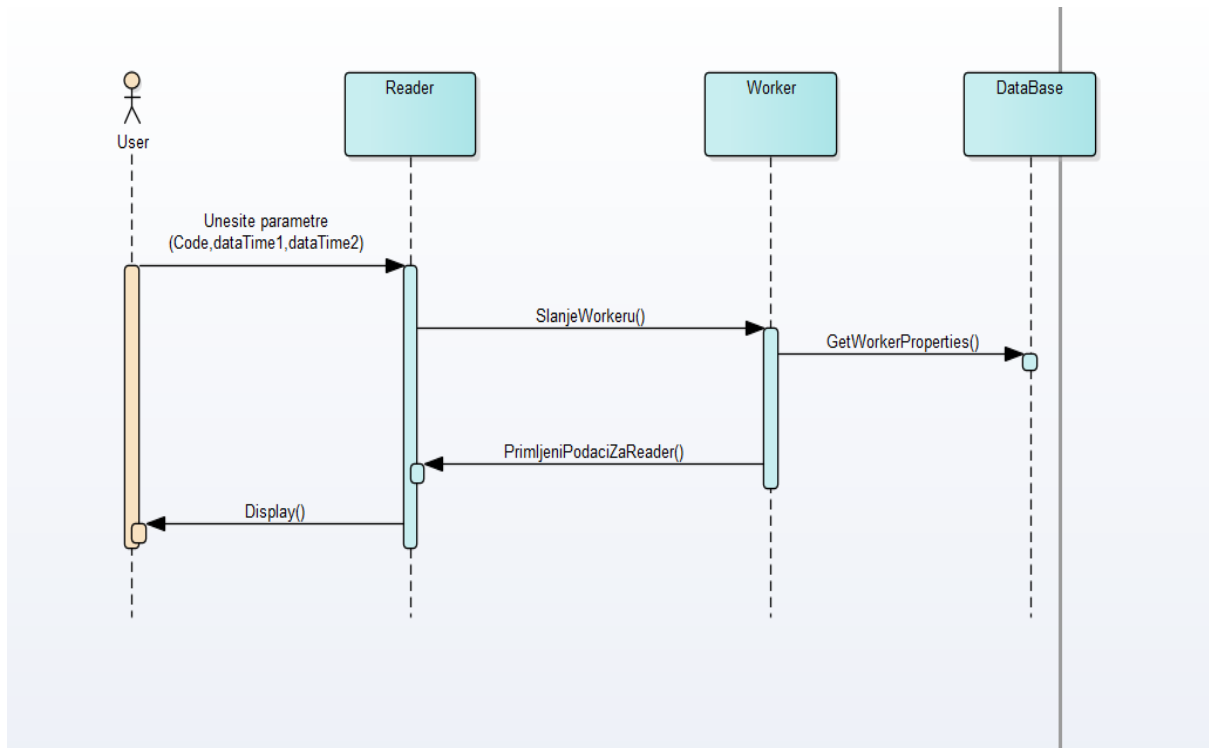
Reader komponenta je konzolni klijent. Ona na svom glavnom programu traži od korisnika da unese vremenski interval za traženi CODE za koji korisnik želi da vidi sve podatke koji su pristigli u određeni Worker. Reader komponenta šalje podatke ka Worker komponenti. Worker komponenta obrađuje podatke na osnovu dobijenih parametara i vraća te podatke ka Readeru. Reader zatim ispisuje te podatke na svoju konzolu.

Logger komponenta je takodje Class library. On prima podatke o svim akcijama koje se dešavaju u sistemu i upisuje ih u tekstualni fajl pod nazivom *logger.txt*.

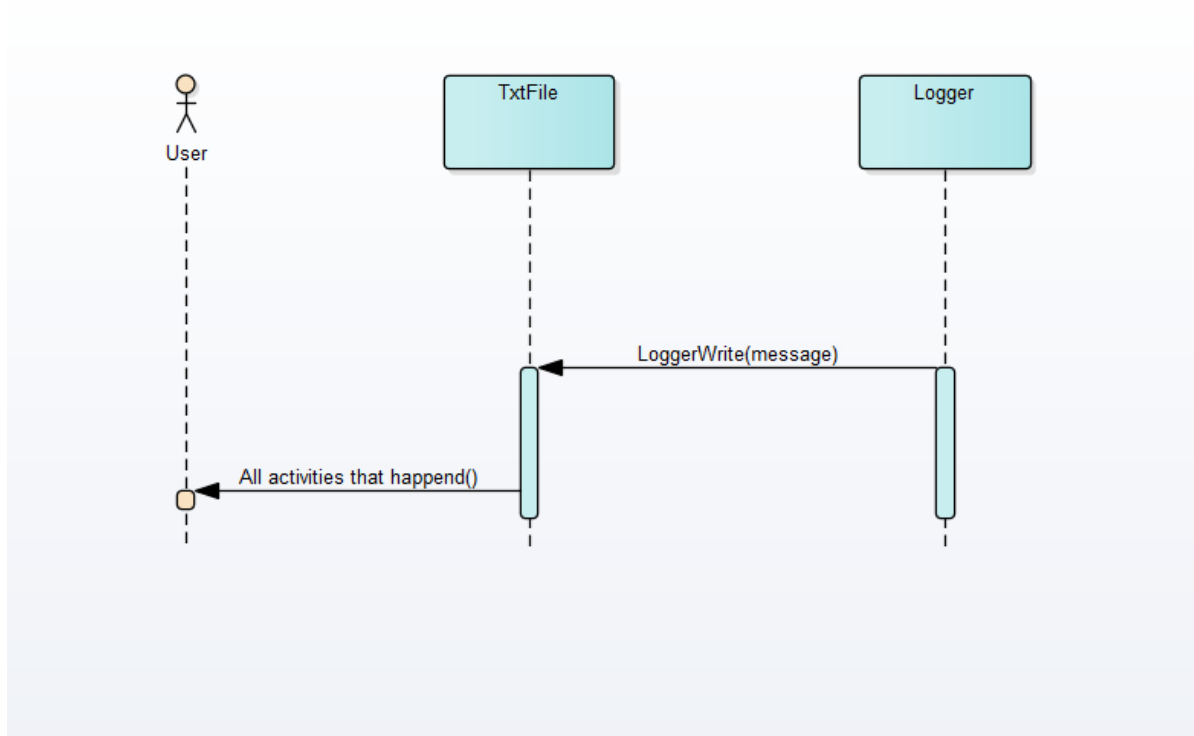


Slika2. Component diagram

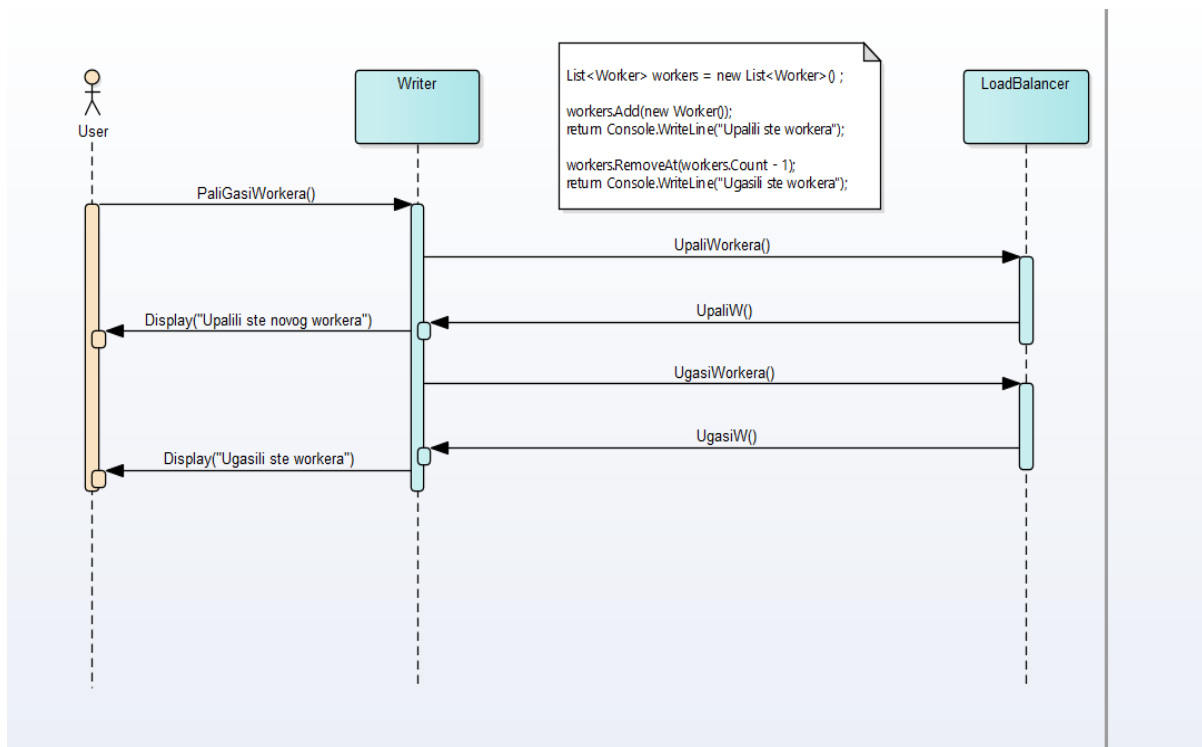
User Stories:



User story 1: As a user I want to be able to filter data by Code and time interval



User story 2: As a user I want to be able to see activities in logger.txt



User story 3: As user I want to be able to turn off or turn on some of workers

Testovi:

Projekat gde se nalaze svi testovi aplikacije, koji pokrivaju cijelu logiku sistema. Koristi se Nunit i Moq.

DataBase :

Sastoji se od modela baze podataka, i klase koja omogućava pristup istoj. Koristi EntityFramework.

Models:

Dole prikazani podaci koji se koriste u cijeloj aplikaciji.

