

Konfliktne situacije Student 1

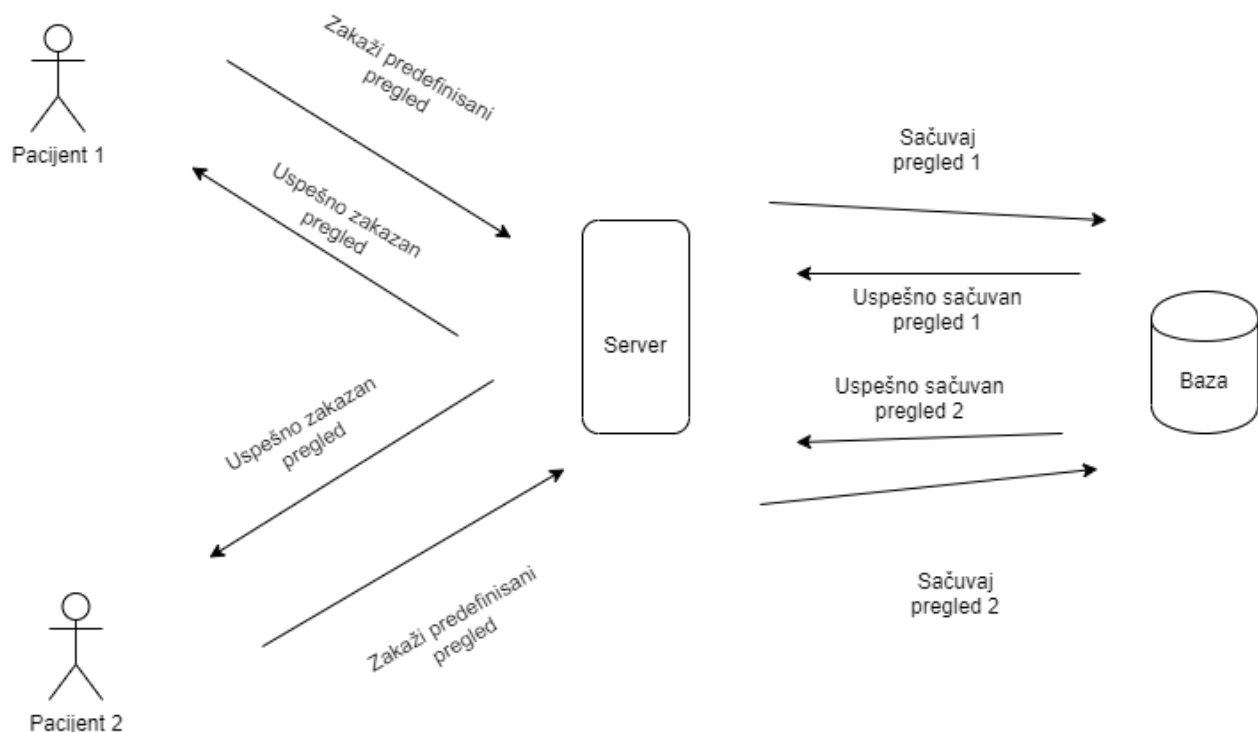
Matija Petrović
SW 33-2017

Konkurentno zakazivanje predefinisanih pregleda

Prilikom zakazivanja predefinisanih pregleda neke klinike može doći do konkurentnog pristupa bazi u slučaju da dva ili više ulogovanih pacijenata pokuša da zakaže isti pregled istovremeno.

U slučaju naše aplikacije, taj use case je implementiran tako što se postojeći od postojećeg PredefinedAppointment-a napravi Appointment kojem se doda pacijent i sačuva u bazu, dok se predefinisani pregled briše iz baze.

Do konflikta može doći ako dva isprepletena zahteva dodaju svaki sa svojim pacijentom dodaju Appointment u tabelu i na kraju izvršenja će se u bazi naći dva Appointment-a za jedan PredefinedAppointment.



Rešenje

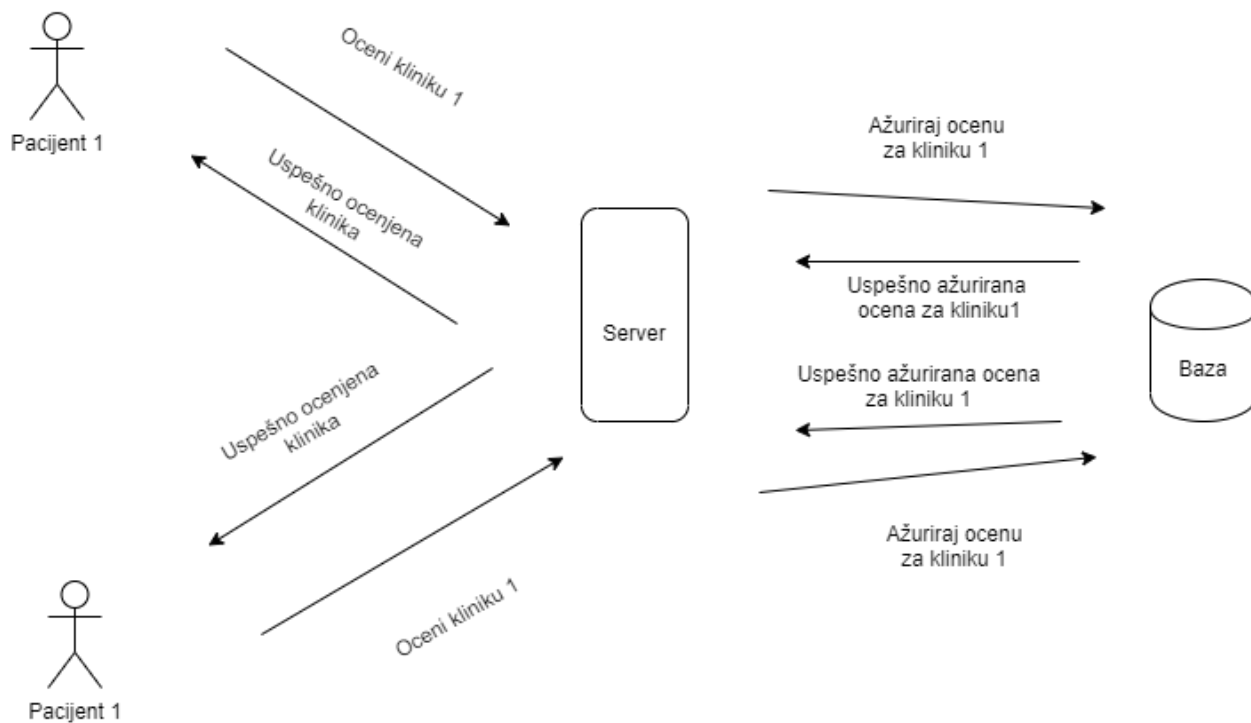
Način na koji se ova situacija može rešiti je kombinacijom obavljanja ove funkcionalnosti transakcijom i zaključavanjem (u našem slučaju pesimističkim) PredefinedAppointment-a koji se zakazuje.

Prilikom vršenja SELECT upita za PredefinedAppointment zaključavamo ga pomoću PESSIMISTIC_WRITE lock-a.

Nakon što prvi zahtev izvrši SELECT, drugi će biti blokiran dok se ne završi transakcija prvog zahteva i nakon toga, SELECT upit neće vratiti ništa i biće bačen RuntimeException koji će rollback-ovati transakciju drugog zahteva i klijentu će biti vraćena greška.

Konkurentno ocenjivanje klinike/doktora

S obzirom da u našoj aplikaciji jedan nalog može biti korišćen od više korisnika istovremeno, javlja se situacija u kojoj više pacijenata istovremeno proverava polje ocene klinike/doktora koje govori da li je pacijentu dozvoljeno da oceni kliniku/doktora (da li je obavio preglede za tu kliniku/doktora od kad je ostavio poslednju ocenu) i onda ažurira postojeću ocenu. Zahtev koji drugi stigne bi trebalo da ne bude u mogućnosti da ostavi ocenu jer prilikom ostavljanja ocene se polje dozvole postavlja na FALSE.



Rešenje

Rešenje ove situacije je se može postići zaključavanjem reda u bazi u kom se nalazi ocena pacijenta za kliniku/doktora, u našem slučaju PESSIMISTIC_WRITE lock-om, koji se postavlja prilikom izvršenja SELECT-a prvog zahteva. Nakon toga drugi zahtev je blokiran i nakon što prvi zahtev završi transakciju, drugi će naići na ocenu kojoj je dozvola za ocenu postavljena na FALSE i korisniku će biti vraćena greška.