

Konkurentni pristup – Student 1

Konfliktna situacija br. 1

Opis konfliktne situacije:

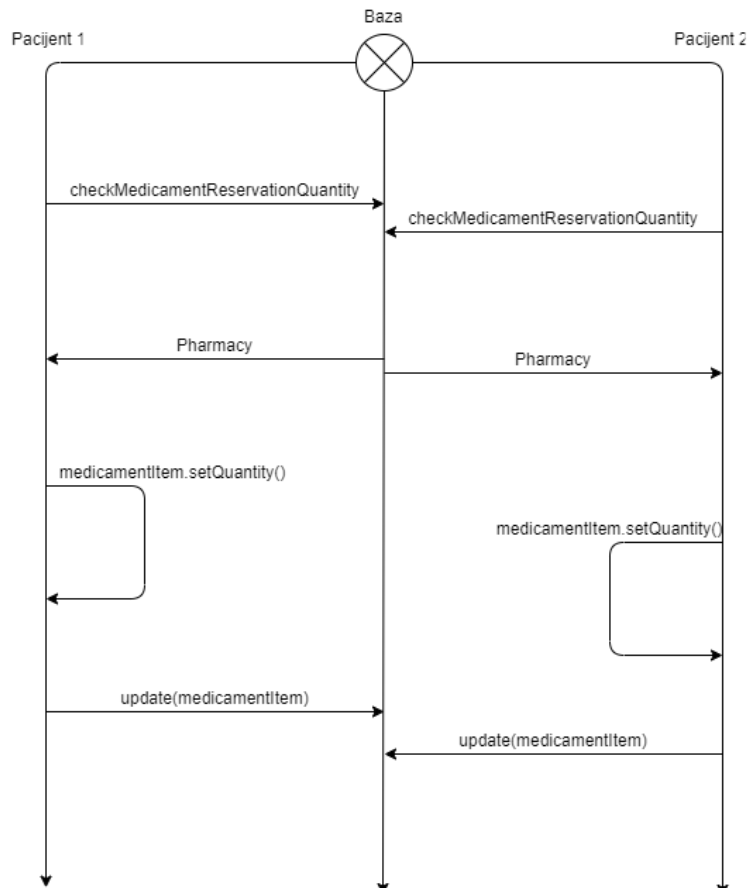
Više pacijenata ne može da rezerviše lekove u isto vreme u istoj apoteci, jer to može dovesti do nekonzistentnog stanja lekova.

Crtež zahteva i odgovor klijenta:

Dva pacijenta mogu da izvrše rezervaciju leka pri čemu dolazi do konfliktne situacije.

1. *Endpoint*: POST/api/patients/reserve
2. *Funkcija*: checkMedicamentReservationQuantity(PrescriptionMedicament, pharmacyId)

Kreira se novi PrescriptionMedicament koji predstavlja rezervaciju korisnika, vrše se provere za rezervaciju (količina pozitivan broj, penala ima manje od 3). Zatim se poziva metoda checkMedicamentReservationQuantity koja proverava da li ima dovoljno leka na stanju i, ako ima, prilagođava količinu leka. Ako je neko već pokušao da promeni stanje leka, novom klijentu to neće biti dozvoljeno i dobiće poruku o neuspešnoj rezervaciji.



Rešenje:

Problem je rešen pomoću optimističkog zaključavanja, tako što je u entitetu *MedicamentItem* dodat atribut *version* koji govori da li je neko već pokušao da promeni stanje leka u datom trenutku. Ako jeste, promena količine se neće izvršiti. Takođe, metodi u servisu koja proverava stanje leka dodata je anotacija **@Transactional**.

Konfliktna situacija br. 2

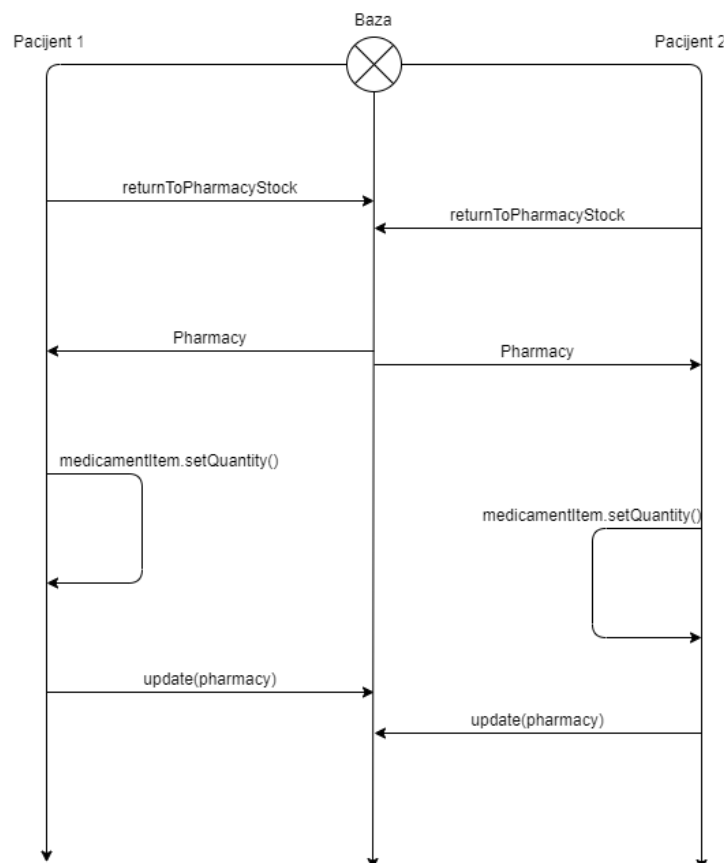
Opis konfliktne situacije:

Više pacijenata pokušava da u isto vreme poništi rezervaciju leka, što količinu leka u apoteci dovodi do nekonzistentnog stanja.

Crtež zahteva i odgovor klijenta:

Dva pacijenta su prijavljena i imaju listu rezervisanih lekova među kojima se nalaze oni koji su u istoj apoteci. Ukoliko oba u isto vreme pokušaju da otkazu njihove rezervacije, to dovodi do konfliktne situacije

1. *Endpoint*: DELETE/api/patients/cancelReservation/id
2. *Funkcija*: returnToPharmacyStock(Medicament, medicamentQuantity)



Rešenje:

Ovaj problem je rešen pomoću pesimističkog zaključavanja tako što je u metodu *getOneWithMedicamentsPatient* dodata anotacija **@Lock**(LockTypeMode.PESSIMISTIC_WRITE) koji stavlja korisnike koji ne mogu da pristupe podatku u određenom trenutku na čekanje sve dok ne budu imali pravo pristupa (kada trenutni klijent prestane da koristi podatke entiteta). Takođe, dodata je i **@Transactional** anotacija na metodu *returnToPharmacyStock*.

Konfliktna situacija br. 2

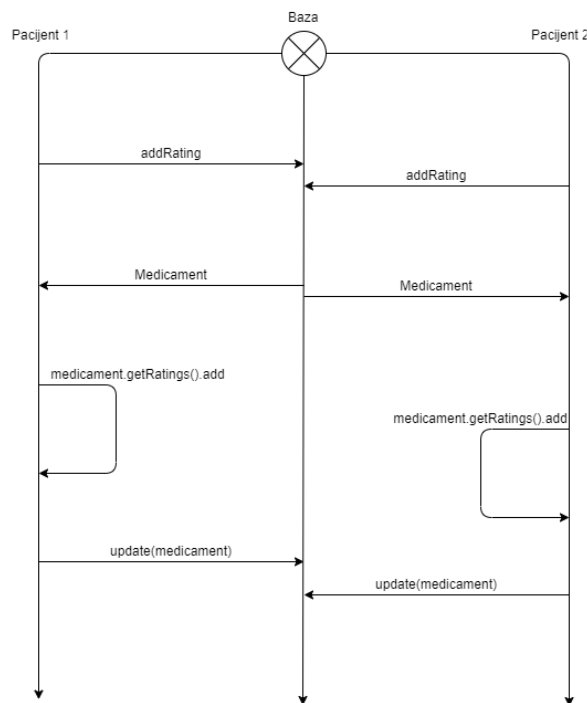
Opis konfliktne situacije:

Pacijent sa više od jednog uređaja pokušava da oceni isti lek i tako kreira više zahteva koji dovode do nekonzistentnog stanja rejtinga za taj lek.

Crtež zahteva i odgovor klijenta:

Pacijent je prijavljen i pokušava da na više uređaja oceni isti lek.

1. *Endpoint*: POST/api/patients/rating
2. *Funkcija*: *addRating*(Rating, medicamentId) u *medicamentService*



Rešenje:

Ovaj problem rešen je pomoću optimističkog zaključavanja gde je entitetu *Rating* dodat atribut *version* koji govori o tome da li neko koristi podatke atributa u datom trenutku. Takođe, dodata je i anotacija **@Transactional** u metodu *addRating* koja se nalazi u *MedicamentService*. Na ovaj način, pacijent će imati pravo da oceni apoteku samo jednom i svi ostali zahtevi će se zanemariti.

