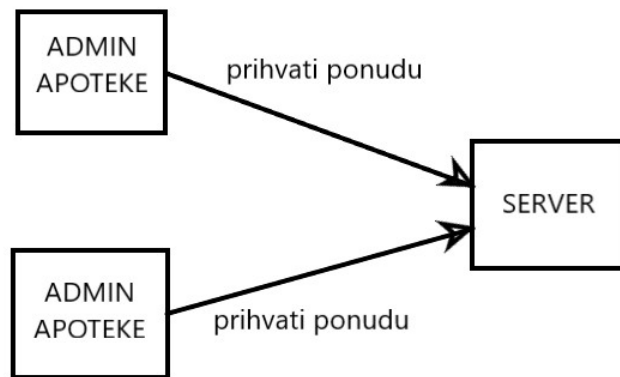


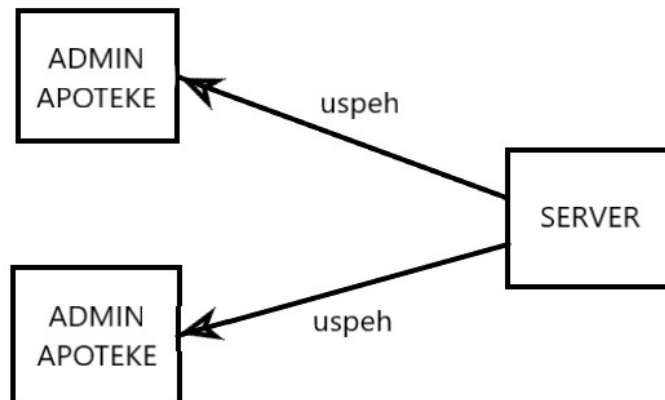
## Konflikti: Student 2

### Slučaj 1

Više administratora apoteke istovremeno pokušava da prihvati ponudu za narudžbenicu koju su kreirali, ali lekovi u narudžbenicama se preklapaju.



Server obaveštava administratore apoteke da je akcija prihvatanja ponude završena uspešno. Međutim, može doći do toga da se količina leka u apoteci ne ažurira pravilno, odnosno da prihvatanje ponude od strane jednog administratora pregazi promene izazvane prihvatanjem ponude drugog administratora.

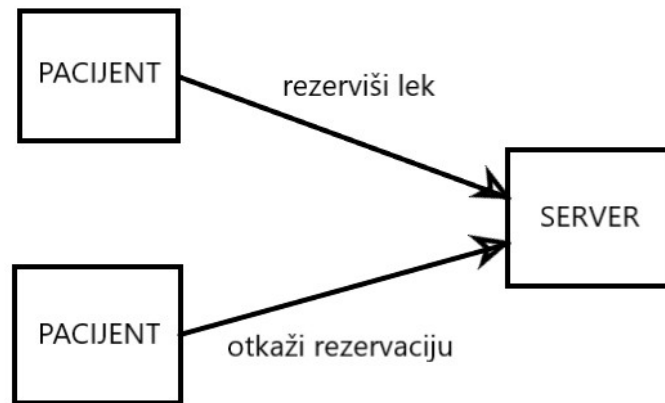


S obzirom da oba korisnika koriste iste podatke, potrebno je da se ažuriranje količine leka izvrši kao transakcija.

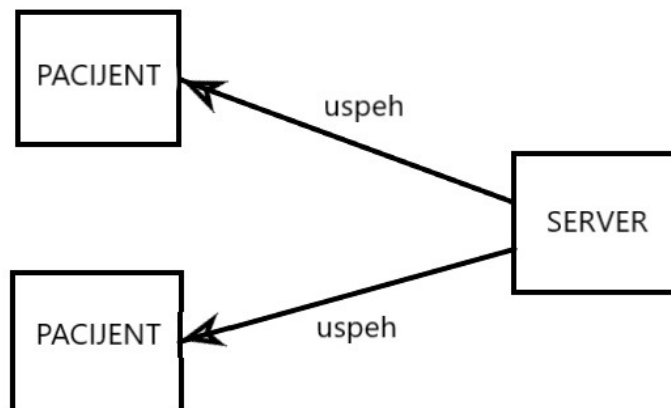
Metoda pod nazivom `prihvatiPonudu` u okviru klase `MagacinService` vrši prihvatanje ponude i izvršava se kao transakcija. Pesimističkim zaključavanjem se zaključavaju lekovi narudžbenice kako bi se obezbedio konkurentan pristup resursu i pravilno ažurirale količine lekova.

## Slučaj 2

Pacijenti otkazuju ili kreiraju različite rezervacije istog leka u isto vreme.



Server obaveštava pacijente da su uspešno otkazali, odnosno kreirali rezervaciju. Prilikom ovih promena nad količinom leka, može se desiti da se količina leka ne ažurira ispravno.

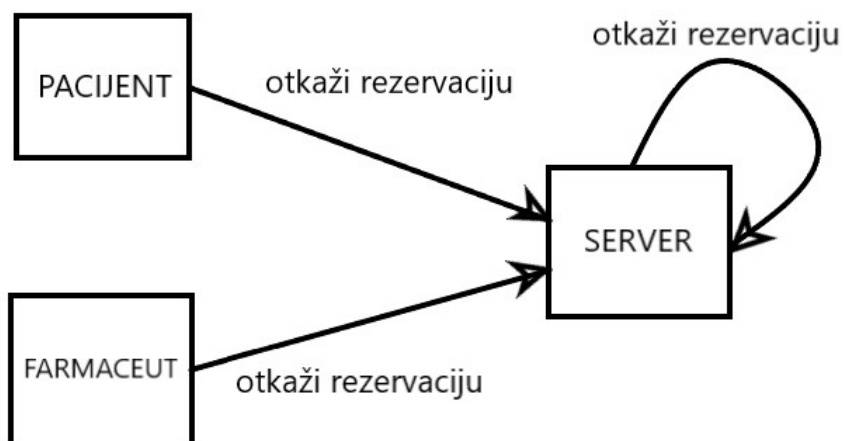


U aplikaciji je ovaj problem rešen zaključavanjem leka iz rezervacije i transakcijama u klasi RezervacijaService. Na ovaj način kada jedan korisnik radi sa lekom, zaključava ga za ostale korisnike, te ovo garantuje pravilno ažuriranje stanja leka u apoteci.

Pored akcijom pacijenta, otkazivanje rezervacije se izvršava u još dva slučaja:

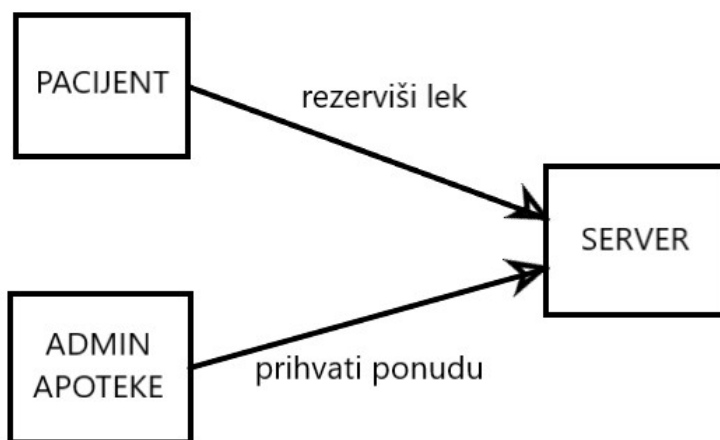
1. prilikom automatske provere da li postoje istekle rezervacije
2. kada farmaceut želi da proveriti status rezervacije, a ona je istekla

Ovi slučajevi su rešeni pesimističkim zaključavanjem kao i prethodno opisani, a sam problem možemo predstaviti na sledeći način:

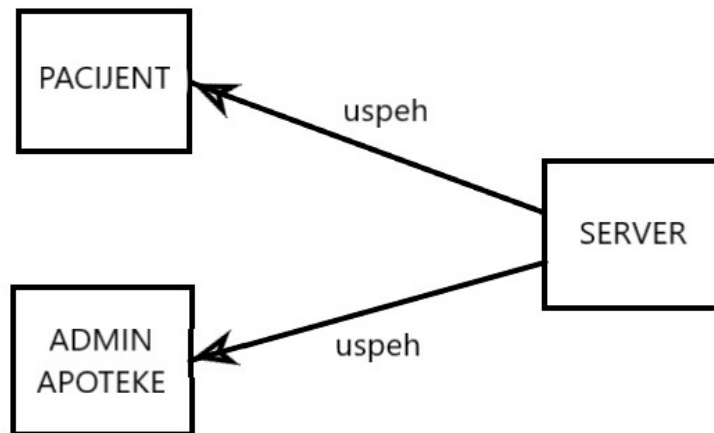


### Slučaj 3

U slučaju kada pacijent otkazuje ili kreira rezervaciju u isto vreme kada administrator prihvata ponudu za narudžbenicu može doći do konflikta kada je u pitanju ažuriranje količine leka u apoteci.



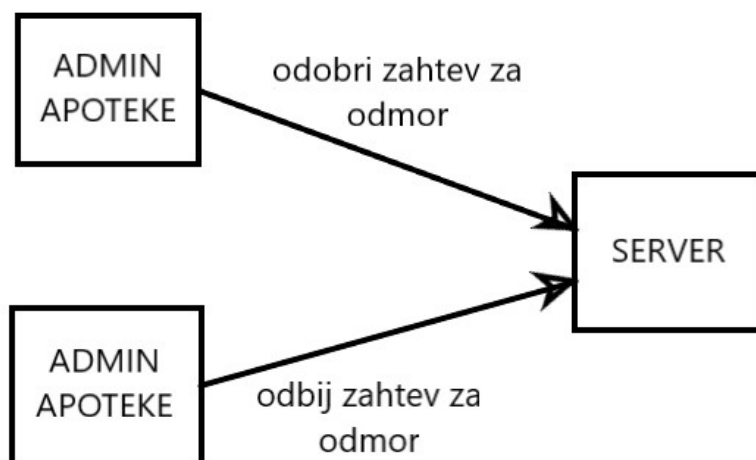
U ovom slučaju će kao i u prethodno opisanim slučajevima biti obavješteni korisnici da je željena akcija uspešno izvršena, ali postoji mogućnost da jedna akcija poništi efekat druge.



U aplikaciji je ovaj problem takođe rešen pesimističkim zaključavanjem i transakcijama u klasama RezervacijaService i MagacinService.

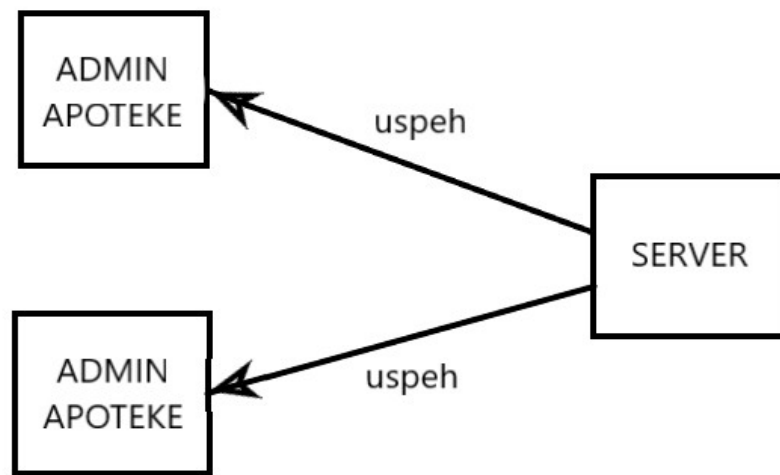
#### Slučaj 4

Još jedan primer konfliktne situacije je istovremeno odobravanje, odnosno odbijanje, istog zahteva za odmor zaposlenog od strane više administratora. Ovaj problem se može javiti prilikom rada sa zahtevima za odmor dermatologa i farmaceuta.



U ovom slučaju, oba administratora će biti obaveštena da su uspešno odobrili ili odbili zahtev za odmor zaposlenog. Problem je to što ukoliko se ne obezbedi konkurentan pristup resursu, može se desiti da jedan administrator odobri zahtev, a zatim da njegovo odobrenje bude pregaženo odbijanjem zahteva

od strane drugog administratora jer u trenutku rada sa zahtevom, oba administratora vide da je status zahteva OBRADA.



Ovaj problem je u aplikaciji rešen korišćenjem pesimističkog zaključavanja zahteva sa kojim se radi i transakcijama. Kada jedan od administratora pristupi zahtevu, on ostaje zaključan za ostale administratore kako ne bi mogli da utiču na status zahteva. Kada se zahtev obradi, otključava se, ali kada mu sledeći admin pristupi, status zahteva više neće biti OBRADA već PRIHVAĆEN ili ODBIJEN, te on ne može da mu ponovo menja status. Na ovaj način je u klasama `OdsustvoDermatologService` i `OdsustvoFarmaceutService` rešen opisani konflikt.