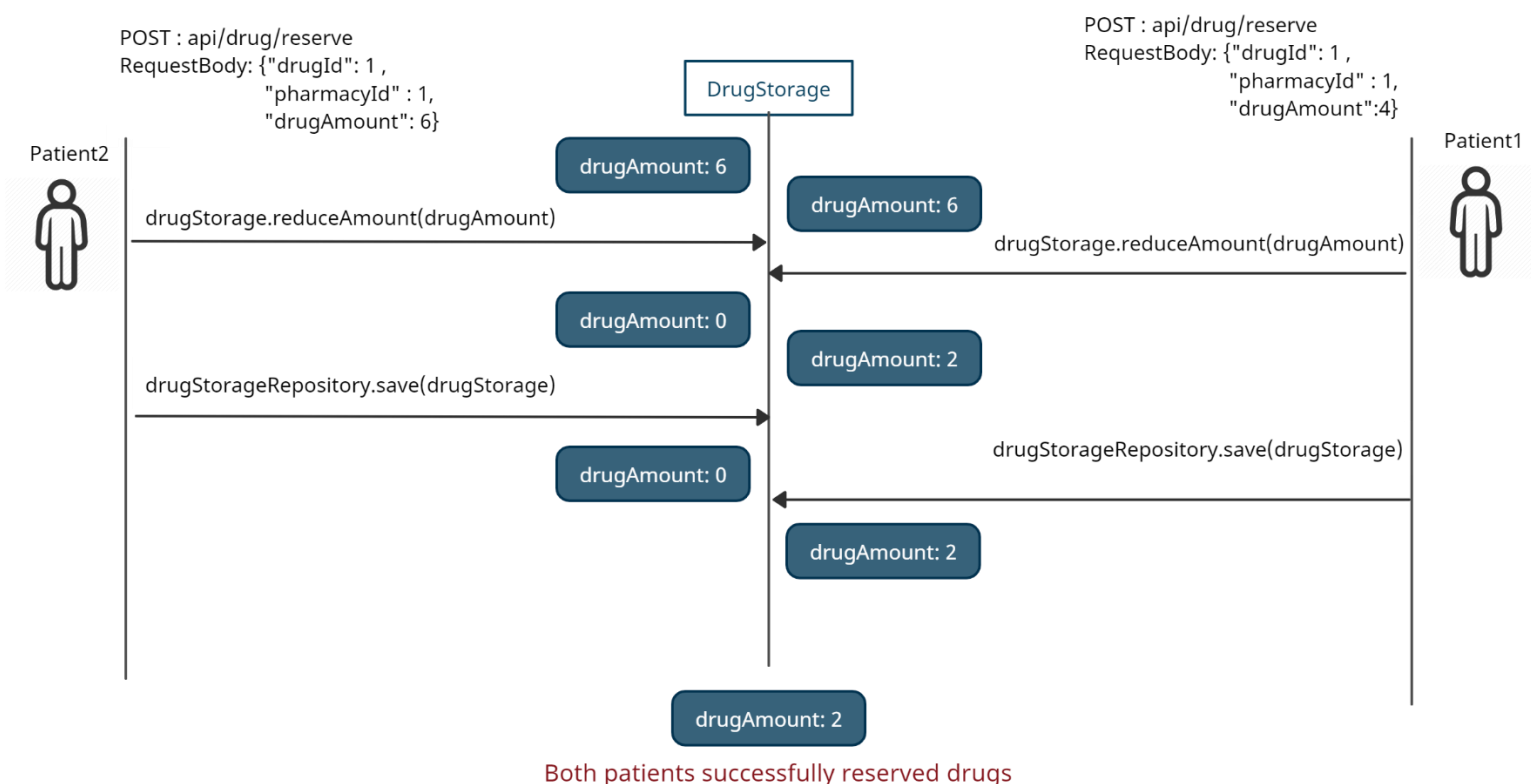


## Opis konfliktnih situacija

Dušan Petrović RA122/2017

### 1. Više istovremenih korisnika pokušavaju da rezervišu lek koji je u međuvremenu postao nedostupan

Kako je moguće da više pacijenata pristupi rezervaciji leka, ukoliko se desi da istovremeno pokušaju da rezervišu isti lek u istoj apoteci, može doći do konflikta i dovesti količinu leka na stanju u apoteci u nekonzistentno stanje.

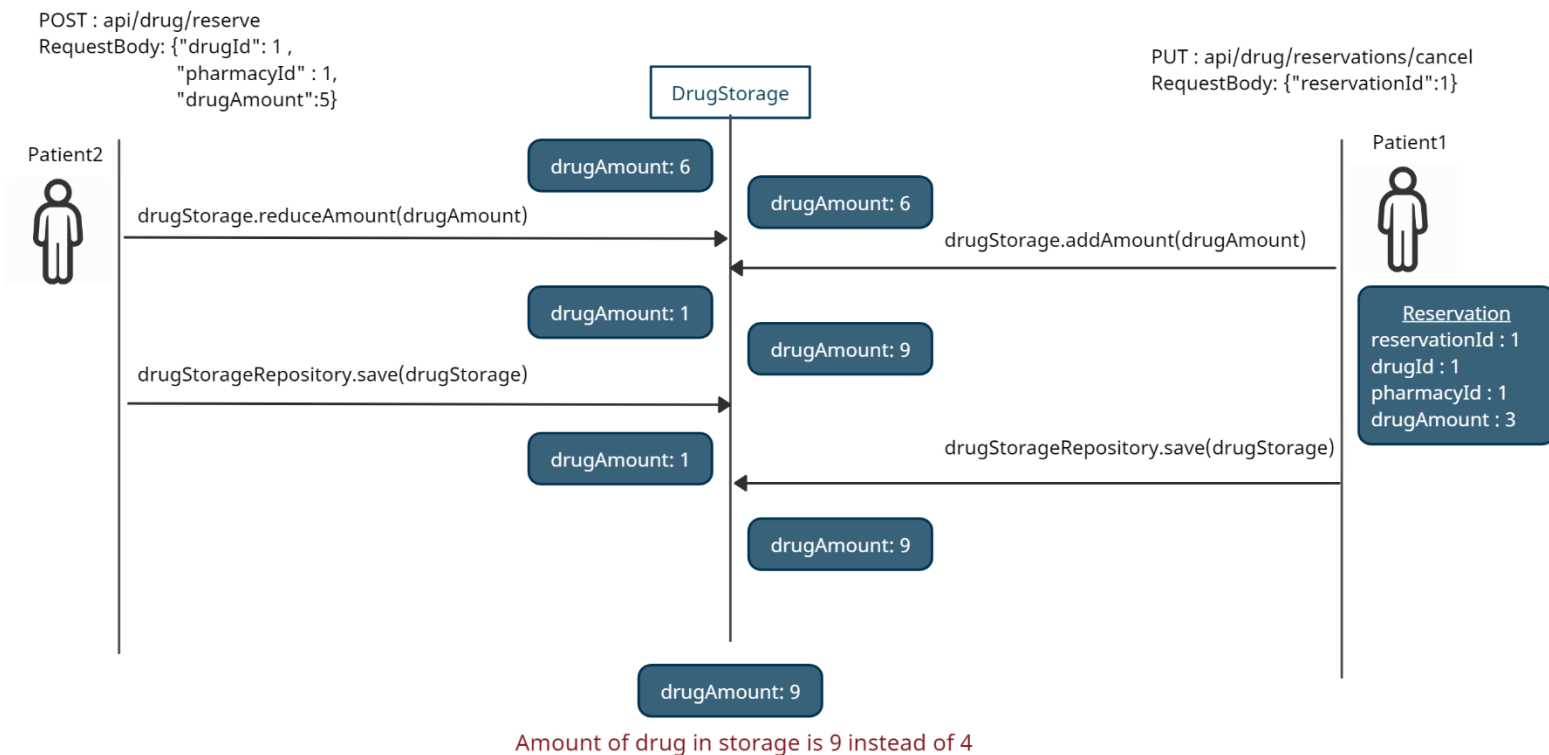


#### Primer konfliktne situacije

Konfliktna situacija je rešena verzionisanjem entiteta DrugStorage, gde će se pri pokušaju ažuriranja entiteta sa mesta koje nema najnoviju verziju entiteta desiti izuzetak ([ObjectOptimisticLockingFailureException](#)). Pacijentu će se ispisati poruka o grešci i neće moći istovremeno da izmeni stanje istog leka u istoj apoteci kao neki drugi pacijent. Sa anotacijom @Transactional obezbeđeno je da u slučaju višestrukih poziva operacije SAVE, u slučaju da se desi izuzetak, vratimo bazu u prethodno stanje.

## 2. Više korisnika ažurira stanje leka u isto vreme

Kako više korisničkih uloga ima mogućnost da menja količinu lekova na stanju, moguće je izazvati konflikte i dovesti količinu lekova u nekonzistentno stanje. Rezervacija leka od strane pacijenta, otkazivanje rezervacije leka, izdavanje leka preko eRecepta, prihvatanje ponude za narudžbenicu su neke od mogućih situacija gde može nastati konflikt.



### Primer konfliktne situacije

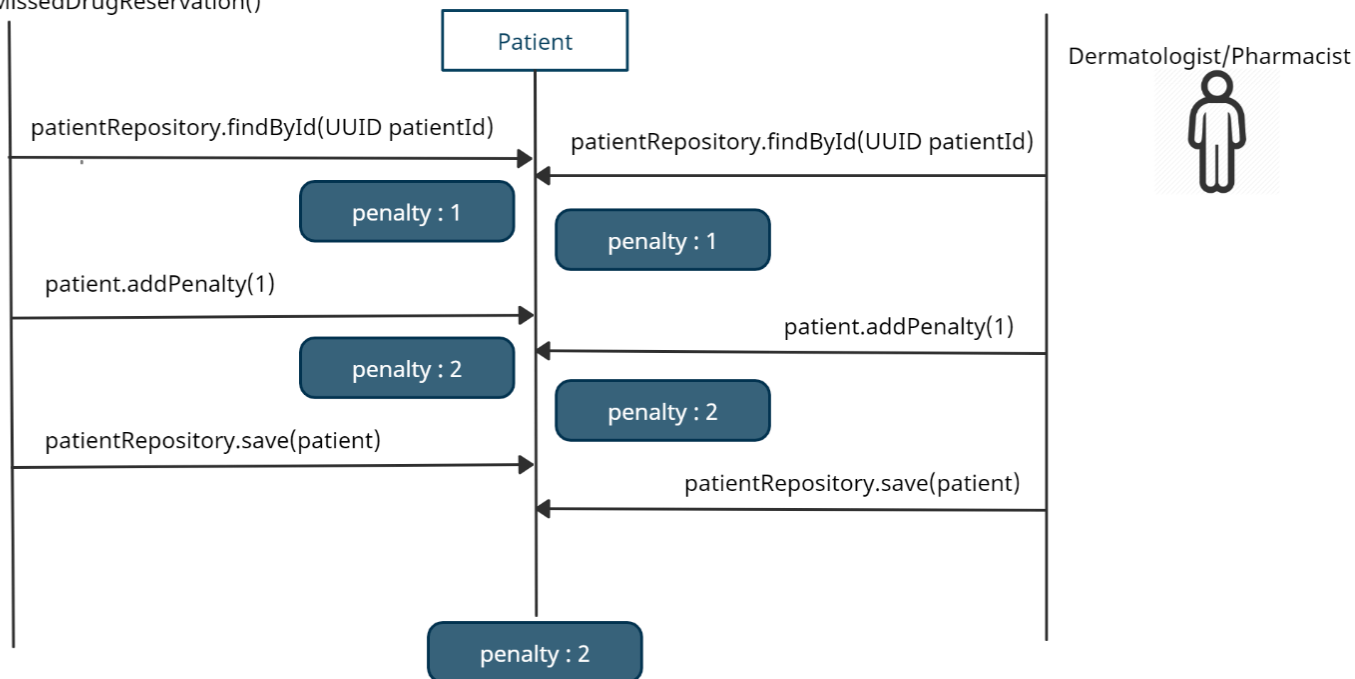
Konfliktna situacija je rešena verzionisanjem entiteta DrugStorage, gde se optimističkim zaključavanjem garantuje konzistentnost. U slučaju da više korisnika pokuša da istovremeno promeni stanje istog leka u istoj apoteci desiće se izuzetak (`ObjectOptimisticLockingFailureException`). Sa anotacijom `@Transactional` obezbeđeno je da u slučaju višestrukih poziva operacije SAVE, u slučaju da se desi izuzetak, vratimo bazu u prethodno stanje.

### 3. Istovremena modifikacija broja korisnikovih penala

Rešavana je konfliktna situacija davanja penala za pacijenta. Dermatolog i farmaceut imaju mogućnost davanja penala pacijentu ukoliko se on ne pojavi na zakazani pregled ili konsultaciju. Takođe potrebno je davati penale pacijentima u slučaju da ne preuzmu rezervisani lek do krajnjeg datuma. Ova metoda je implementirana tako da se automatski okida na svakih 5 minuta i daje penale pacijentima koji nisu preuzeli lek. Ovde može doći do konflikta i dovesti broj penala u nekonzistentno stanje.

@Scheduled(fixedRate = 300000, initialDelay = 2000)  
givePenaltyForMissedDrugReservation()

PUT : api/appointment/did-not-show-up



Number of penalties is 2 instead of 3

*Primer konfliktne situacije*

Konfliktna situacija je rešena verzionisanjem samog korisnika, gde će se pri pokušaju ažuriranja sa mesta koje nema najnovije stanje entiteta desiti izuzetak ([ObjectOptimisticLockingFailureException](#)). Po dešavanju izuzetka, ako ga je izazvao dermatolog/farmaceut, ispisaće mu se poruka o grešci, a u slučaju da je izazvala metoda koja se automatski okida, završiće, i nakon 5 minuta će se opet pokrenuti i imati priliku da da penale pacijentima. Sa anotacijom @Transactional obezbeđeno je da u slučaju višestrukih poziva operacije SAVE, u slučaju da se desi izuzetak, vratimo bazu u prethodno stanje.