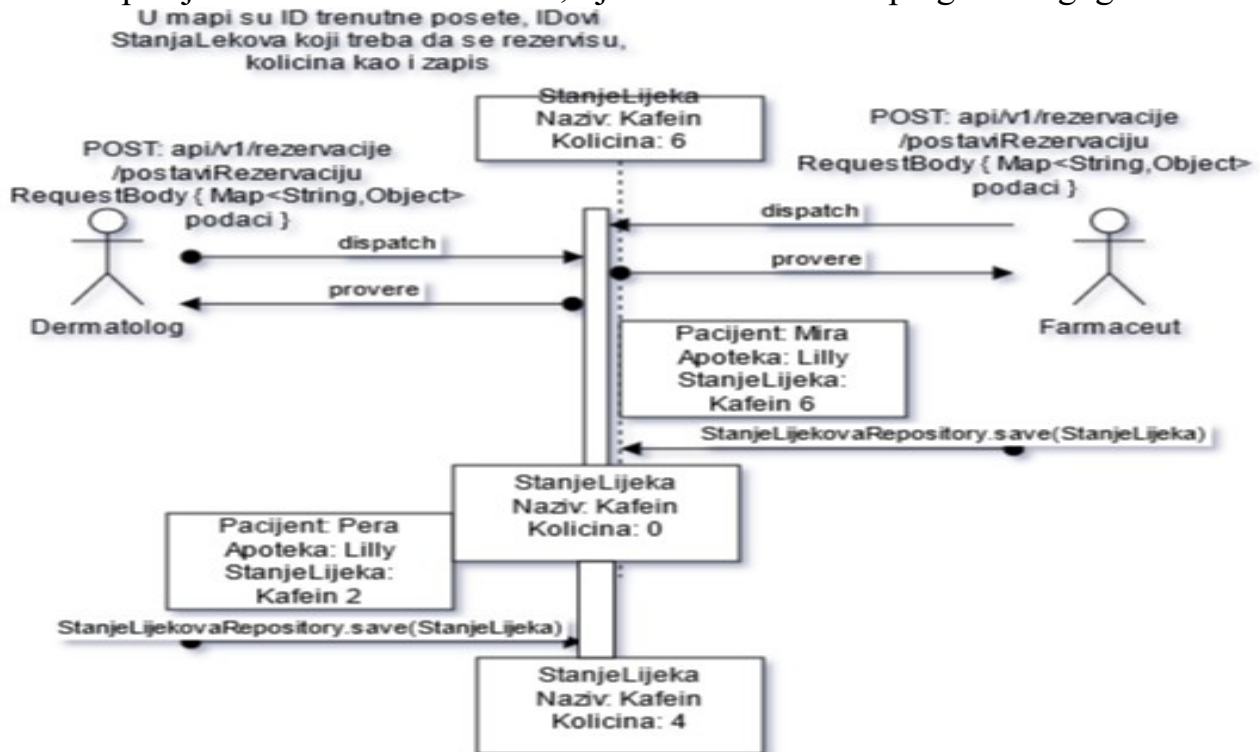


Rešavanje konfliktnih situacija

Dragan Arsić SW-5-2018

1. Prilikom čuvanja stanja leka pri završetku pregleda dolazi do nekonzistentnosti broja preostalih lekova

Kada imamo situaciju da kada 2 Radnika iz iste apoteke žele da prepisu isti lek nekom pacijentu dođe do data race-a, i jedna količina leka pregazi drugog.

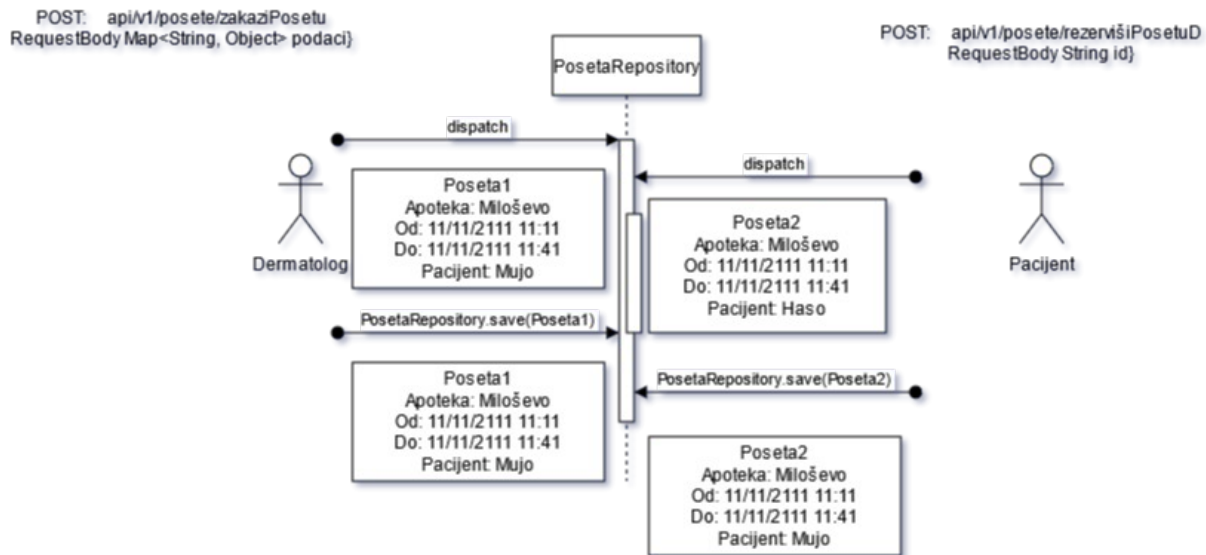


Slika 1. Konfliktna situacija pri završavanju pregleda

Konfliktna situacija je rešena verzionisanjem **StanjaLijeka**. Ukoliko dođe do konflikta gde dva Radnika mjenjaju stanje istog leka u isto vrijeme, proverice se trenutna verzija StanjaLijeka i desiće se **Exception ObjectOptimisticLockingFailureException**. Anotacija **@Transactional** rešava problem gde u slučaju višestrukih poziva *save* operacije, tokom dešavanja izuzetka, poništava prethodne izmene u bazi.

2. Dermatolog ne može biti na više različitih pregleda prisutan

Kada imamo situaciju da Dermatolog vrši pregled jednog pacijenta i tokom pregleda pokuša da zakaže drugi termin sa pacijentom dok u to isto vreme drugi Pacijent zakaže pregled kod njega, dolazi do data race-a.

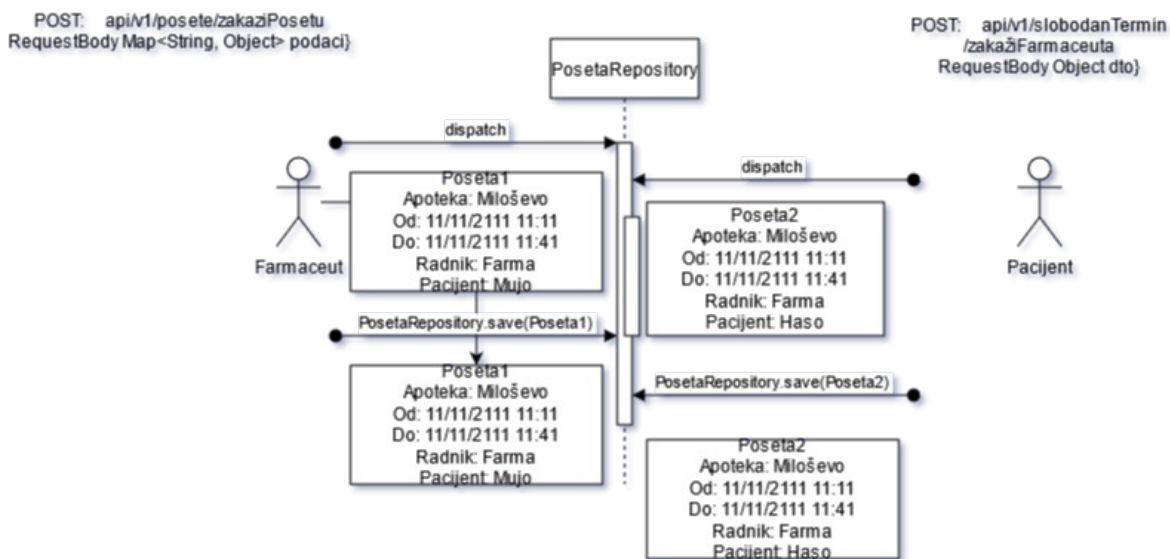


Slika 2. Konfliktna situacija pri zakazivanju posete kod Dermatologa

Konfliktna situacija je rešena identično kao prethodna, dodamo **Dermatologu** **Version** atribut pomoću koga se pre sačuvavanja proverava da li je došlo do izmene **Dermatologa**, ako jeste, diže se **ObjectOptimisticLockingFailureException**, takođe uz stavljanje „Transactional“ možemo odmah nakon Exception-a da vratimo bazu u prethodno stanje.

3. Farmaceut ne može biti na više različitih savetovanja prisutan

Kada imamo situaciju da Farmaceut vrši savetovanje jednog pacijenta i tokom pregleda pokuša da zakaže drugi termin sa pacijentom dok u to isto vreme drugi Pacijent zakaže pregled kod njega, dolazi do data race-a.



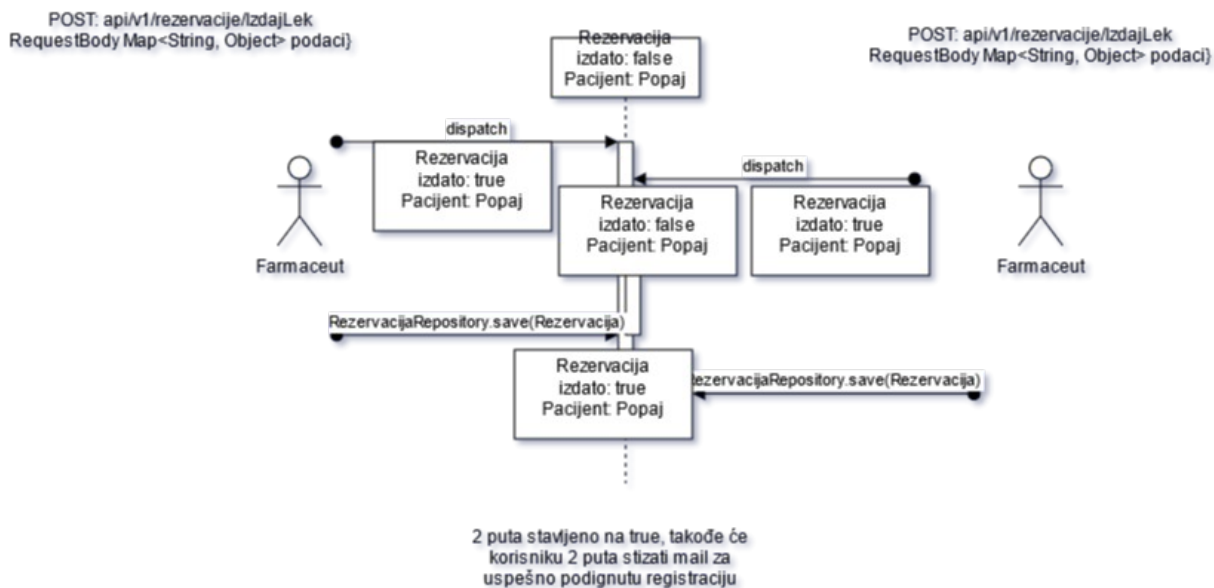
Slika 3. Konfliktna situacija prilikom rezervisanja Savetovanja

Konfliktna situacija je rešena identično kao prethodna, dodamo **Farmaceutu** Version atribut pomoću koga se pre sačuvavanja proverava da li je došlo do izmene Farmaceuta, ako jeste, diže se **ObjectOptimisticLockingFailureException**, takođe uz stavljanje „Transactional“ možemo odmah nakon Exception-a da vratimo bazu u prethodno stanje.

4. Duplo iskorišćenje Rezervacije ne bi trebalo biti moguće

Iako je situacija malo nerealna, postoji mogućnost duplog iskorišćenja Rezervacije – uzimanja svih lekova ponovo.

U konfliktnoj situaciji moraju učestvovati 2 Farmaceuti iz iste Apoteke koji izdaju jednom pacijentu lek u istom trenutku – dovodeći rezervaciju do data race-a.



Slika 4. Konfliktna situacija prilikom izdaje rezervacije

Konfliktna situacija je rešena identično kao prethodna, dodamo **Rezervaciji** **Version** atribut pomoću koga se pre sačuvavanja proverava da li je došlo do izmene Rezervacije, ako jeste, diže se **ObjectOptimisticLockingFailureException**, takođe uz stavljanje „Transactional“ možemo odmah nakon Exception-a da vratimo bazu u prethodno stanje – mada u ovom slučaju nije neophodno.