

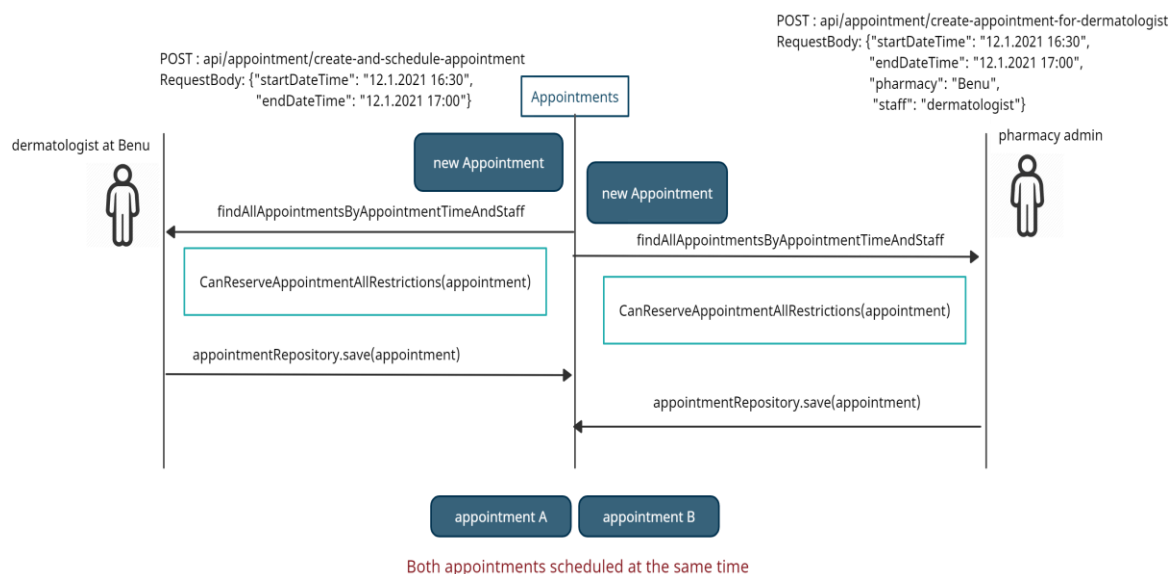
## Opis konfliktnih situacija

### Stefan Zindović RA132/2017

#### 1. Jedan dermatolog ne može istovremeno da bude prisutan na više različitih pregleda

Kako je moguće da dermatolog i administrator apoteke istovremeno kreiraju termin za novi pregled, ukoliko se desi da istovremeno pokušaju da kreiraju novi termin u istoj apoteci za istog dermatologa, može doći do konflikta i kreiranja dva termina za istog dermatolog u istoj apoteci u isto vreme.

#### Primer konfliktne situacije

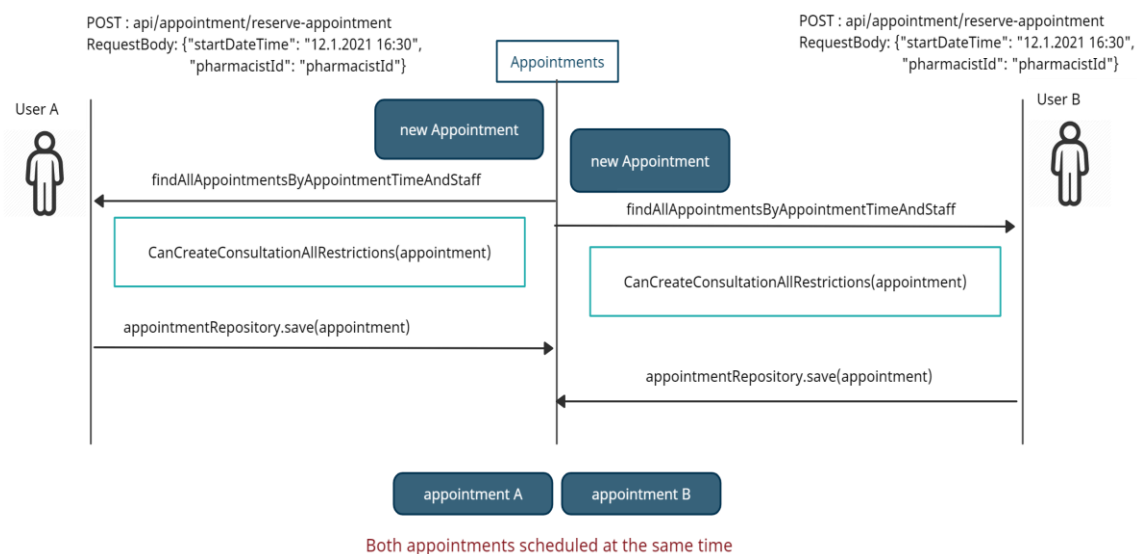


Konfliktna situacija je rešena verzionisanjem entiteta Staff nadtipa entiteta Dermatologist, gde na početku obrade povučemo trenutnu verziju entita, a ne kraju metodom save proverimo da li je neko updatovao entitet u međuvremenu. Na ovaj način kada dođe do izuzetka (`ObjectOptimisticLockingFailureException`) možemo obavestiti korisnika da je došlo do greške i nije moguće zakazati pregled u željenom terminu. Anotiramo metodu sa `@Transactional` kako imamo operaciju SAVE za Staff entitet za koga se pregled kreira i za novo kreirani pregled. Sa anotacijom `@Transactional` obezbeđeno je da u slučaju višestrukih poziva operacije SAVE, u slučaju da se desi izuzetak, vratimo bazu u prethodno stanje. Kako je vreme potrebno za izvršavanje navedenih metoda malo i kako ne očekujemo kreiranje novih pregleda za istog dermatologa u tako kratkom intervalu očekujemo da neće često dolaziti do izuzetka.

## 2. Jedan farmaceut ne može istovremeno da bude prisutan na više različitih savetovanja

Kako više korisnika ima mogućnost da zakaže savetovanje u željenom terminu, moguće je izazvati konflikte i dovesti zakazane konsultacije u nekonzistentno stanje. Ukoliko dva različita korisnika pokušaju da zakažu konsultaciju kod istog farmaceuta u istom terminu, moguće je da će se za farmaceutu zakazati dva savetovanja u istom terminu.

### Primer konfliktne situacije

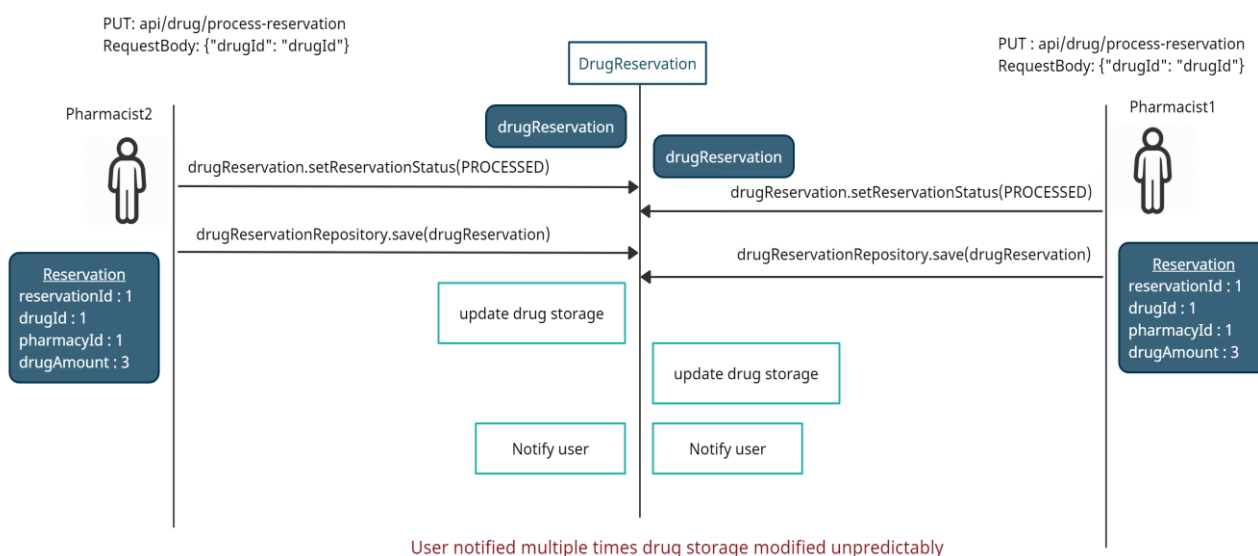


Konfliktna situacija je rešena verzionisanjem entiteta Staff nadtipa entiteta Pharmacist, gde na početku obrade povučemo trenutnu verziju entita, a ne kraju metodom save proverimo da li je neko updatovao entitet u međuvremenu. Na ovaj način kada dođe do izuzetka (`ObjectOptimisticLockingFailureException`) možemo obavestiti korisnika da je došlo do greške i nije moguće zakazati pregled u željenom terminu. Anotiramo metodu sa `@Transactional` kako imamo operaciju SAVE za Staff entitet za koga se pregled kreira i za novo kreirani pregled. Sa anotacijom `@Transactional` obezbeđeno je da u slučaju višestrukih poziva operacije SAVE, u slučaju da se desi izuzetak, vratimo bazu u prethodno stanje. Kako je vreme potrebno za izvršavanje navedenih metoda malo i kako ne očekujemo kreiranje novih pregleda za istog farmaceutu u tako kratkom intervalu očekujemo da neće često dolaziti do izuzetka.

### 3. Istovremeno izdavanje rezervisanog leka od strane dva farmaceuta

Rešavana je konfliktna situacija idavanje rezervisanog leka. Moguće je da dva farmaceuta istovremeno izdaju rezervisan lek pacijentu u tom slučaju stanje rezervacije se menja iz dva threada a i time okidaju i ostale metode koje čine obradu rezervacije na ovaj način može doći do umanjivanja duplo veće količine leka iz drug storagea. Takođe i sam korisnik će biti obavešten dva puta da je njegova rezervacija uspešno obrađena.

#### Primer konfliktne situacije



Konfliktna situacija je rešena verzionisanjem rezervacije leka, gde će se pri pokušaju ažuriranja sa mesta koje nema najnovije stanje entiteta desiti izuzetak (`ObjectOptimisticLockingFailureException`). Po dešavanju izuzetka korisniku će se ispisati poruka o grešci. Na ovaj način smo osigurali da dodatno procesiranje neće uticati na ostatak sistema.