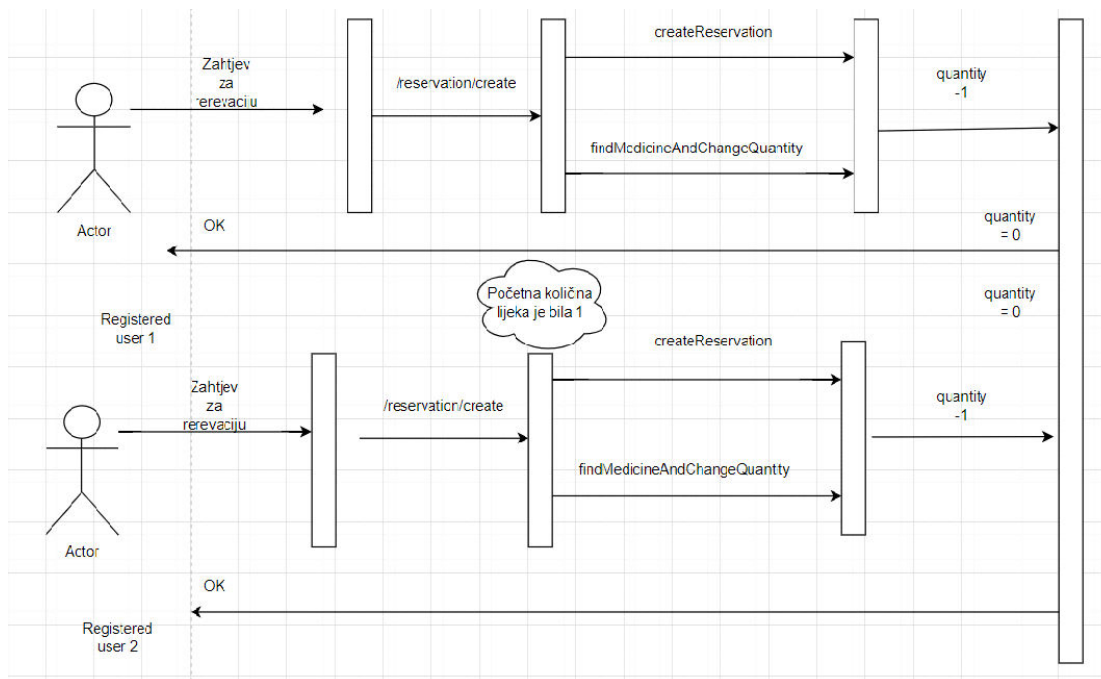


Konfliktne situacije - Student 1

1. Više istovremenih korisnika aplikacije ne može da rezerviše lek koji je u međuvremenu postao nedostupan

Do konfliktne situacije može doći ukoliko više korisnika u isto vrijeme pokuša da rezerviše isti lijek i pomenuti lijek u međuvremenu postane nedostupan.



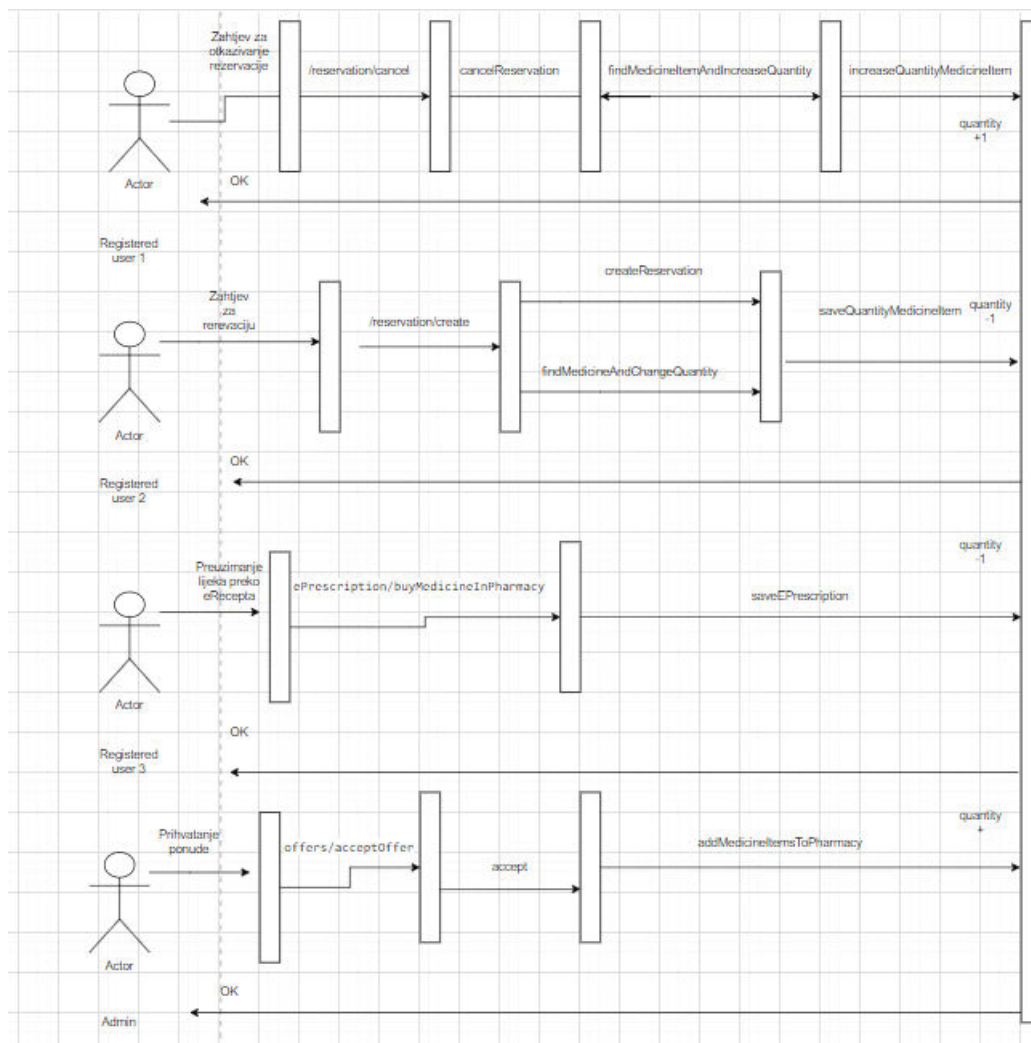
Korisnik pretragom lijeka dobija listu apoteka u kojima je lijek dostupan, te on bira apoteku i datum za rezervaciju. Iako je lijek bio dostupan prilikom selekcije, ukoliko 2 korisnika istovremeno pokušaju da naprave zahtjev za rezervacijom istog lijeka u istoj apoteci, čija je količina 1, dolazi to konflikte situacije na slici. Da bi se riješio ovaj konflikt potrebno je zabraniti korisniku, čij zahtjev stigne kasnije, da napravi željene izmjene.

Konflikt je riješen dodavanjem anotacije @Version na atribut Long version u klasi Medicineltem. Ovim putem je izvršeno verzionisanje entiteta Medicineltem.

Korisnik čij se zahtjev prvi obradi će izmjeniti verziju entiteta i njegova rezervacija će biti prihvaćena. Kada drugi korisnik pokuša da napravi izmjene sa starom verzijom entiteta doći će do greške `OptimisticLockingFailureException`. Da bi baza ostala u konzistentnom stanju dodata je i anotacija `@Transactional` iznad metode `createReservation` kako bi se izvršio rollback svih promjena ukoliko dođe do izuzetka.

2. Količina leka na stanju se mora ispravno ažurirati nakon rezervacije leka od strane pacijenta, otkazivanja rezervacije leka, izdavanja leka preko eRecepta, prihvatanja ponude za narudžbenicu itd.

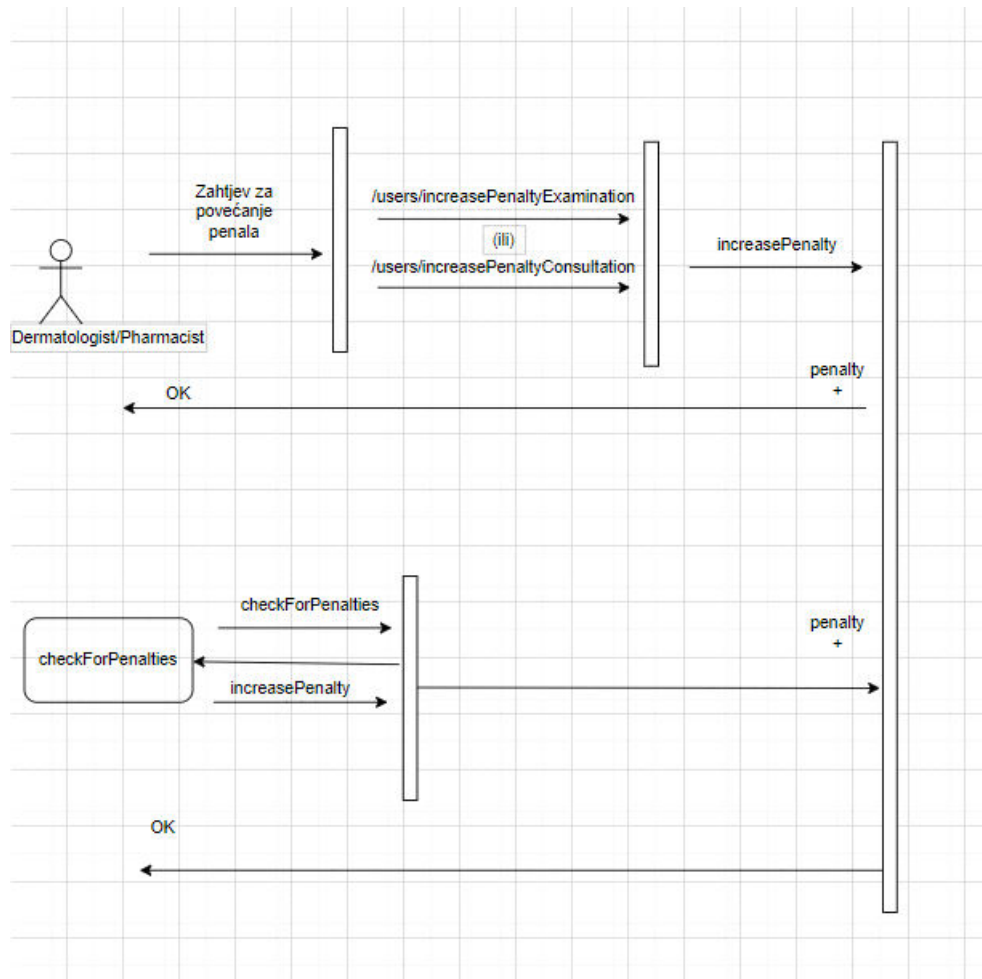
Potrebno je ažurirati lijekove na stanju prilikom svakog rada sa njima. Ovakva praksa je neophodna ne bi došlo do situacije da korisnici rukuju sa lijekovima koji u realnom sistemu nisu na stanju.



Ukoliko se stanje lijekova ne bi mijenjalo, došlo bi do situacije da korisnici ne mogu da preuzmu ili rezervišu lijekove koji su na stanju, ili u suprotnom slučaju, da rezervišu lijekove koji nisu na stanju. Za svaku od gore pomenutih funkcionalnosti implementirane su metode za povećavanje ili smanjivanje količine lijekova. Takođe, urađeno je verzionisanje entiteta MedicineItem u kom je i smještena informacija o količini, te je ukoliko više korisnika pokuša istovremeno da ažurira stanje u bazi, samo prvi korisnik će uspjeti da izmjeni stanje, a ostali će zbog nevalidne verzije entiteta dobiti izuzetak `OptimisticLockingFailureException`. Takođe iznad metoda `createReservation`, `cancelReservation`, `accept` i `buyMedicineInPharmacy` je dodana anotacija `@Transactional` kako bi bile poništene sve napravljene promjene ukoliko dođe do izuzetka.

3. Zabrana povećanja penala kod korisnika u isto vrijeme

Povećanje korisnikovih penala je moguće od strane farmaceuta i dermatologa u slučajevima kada se korisnik ne pojavi na pregledu. Takođe ovoj funkcionalnosti ima pristup i sistem koji provjerava da li korisnik ima rezervacije koje nije preuzeo, ni otkazao i dodajte mu po jedna penal za svaku nepreuzetu rezervaciju. Do konflikte situacije dolazi kada postoji više pokušaja da se promijeni broj penala.



Konfliktna situacije je riješena dodavanjem anotacije @Version iznad atributa version u entitetu User. Ovako je omogućeno da se prvi pristup entitetu izvrši i izmjeni entitet. Dok će u drugom pokušaju doći do izuzetka i rollback-a podataka zbog anotacije Transactional iznad metoda checkForPenalties, increasePenaltiesExamination i increasePenaltiesConsultation.