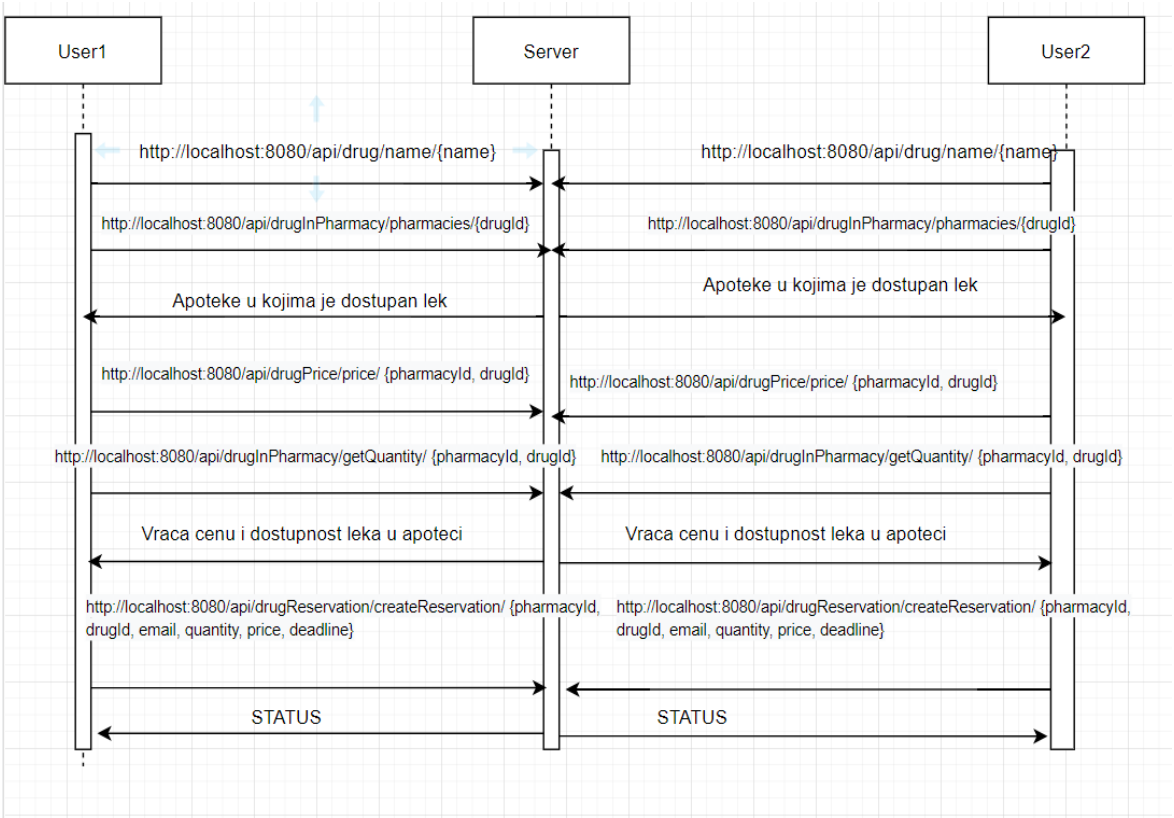
Konkurentni pristup resursima u bazi
Jovan Bosnić RA173-2017 ISA Student 1

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

Opis:
Korisnik na stranici za prikaz svih lekova bira lek koji zeli da rezervise, ukoliko je lek dostupan u bilo kojoj apoteci korisniku se prikazuje forma za reservisanje leka. U formi korisnik bira datum do kojeg ce preuzeti lek, kolicinu leka I zatim bira apoteku u kojoj zeli da rezervise lek nakon cega mu se prikazuje dostupsnost leka u zeljenoj apoteci. Ukoliko u apoteci ne postoji dovoljna kolicina leka pacijent ne moze da rezervise lek, ukoliko postoji pacijentu se prikazuje cena za jedan I ukupna cena koju treba da plati u zavisnosti od kolicine koju je korisnik uneo I pacijent moze da rezervise lek.
Do konfliktne situacije moze da dodje ukoliko se desi da vise korisnika u istoj apoteci rezervise isti lek a pritom su trazene kolicine vece nego dostupne.

Tok:
Prilikom odabira leka korisnik salje zahtev serveru, tacnije gadja **DrugController** metodu **findByName** kojoj se prosledjuje ime odabranog leka.Zatim se otvara forma za rezervisanje leka I salje se zahtev serveru za dobijanje apoteka u kojima je lek dostupan, odnosno **DrugInPharmacyController** metoda **findPharmaciesWithDrug** I prosledjuje se id leka. Klijent tada treba da odabere apoteku u kojoj zeli da rezervise lek nakon cega dobija informaciju o stanju leka u apoteci I cenu. Pri odabiru apoteke gadja se **DrugPriceController** metoda **findPrice** kojoj se prosledjuju id apoteke I id leka I ona vraca cenu leka u apoteci. Takodje se gadja I **DrugInPharmacyController** metoda **getQuantity** kojoj se prosledjuju id leka I id apoteke I ona vraca dostupnu kolicinu leka u apoteci. Zatim pacijent unosi kolicinu leka, a moze se uneti samo kolicina manja od trenutne u apoteci, datum do kada ce preuzeti lek I klikom na dugme za rezervaciju gadja se **DrugReservationController** metoda **createDrugReservation** koja prima email pacijenta, datum preuzimanja, id apoteke, kolicina koja se rezervise, id leka, I ukupna izracunata cena koja treba da se plati. Ova metoda vraca status o uspesnosti akcije I korisnik se redirektuje na svoj Home Page.

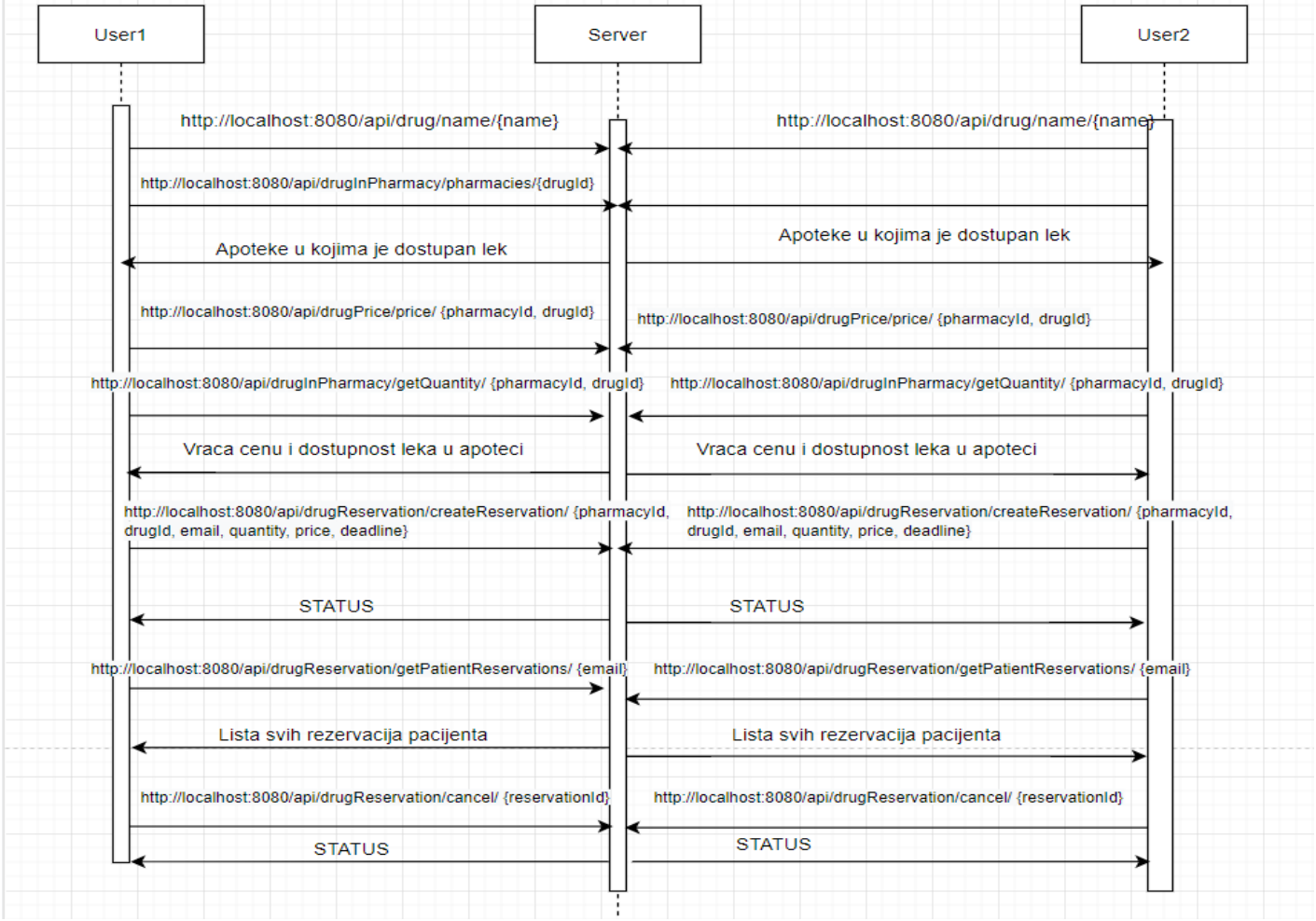


Resenje:
Ovaj problem sam resio dodavanjem *@Transactional(readOnly=false, propagation = Propagation.REQUIRES_NEW)* anotacije iznad metode za kreiranje rezervacije, metoda je **createDrugReservation** u klasi **DrugReservationServiceImpl**. Metoda za cuvanje kolicine leka **save** u **DrugInPharmacyServiceImpl** je takodje anotirana sa @Transactional, metoda **findPharmacyById** klase **PharmacyServiceImpl** koja se poziva unutar createDrugReservation je takodje oznacena kao transakciona @Transactional, kao I metoda **findById** u klasi **DrugServiceImpl**. Tako su sve metode koje se pozivaju unutar createDrugReservation transakcione, na ovaj nacin je omoguceno da lek ne bude rezervisan ukoliko je nestalo kolicine u izabranoj apoteci. Jer je tako sa readOnly omoguceno da se izmene podaci, a sa REQUIRES_NEW je omoguceno da se kreira nova transakcija I ako neka ima tekuca vec. Da sam koristio resenje sa verzijom ili zakljucavanjem resursa, onda ni korisnici koji imaju pravo da rezervisu lek, odnosno leka ima dovoljno za sve korisnike koji istovremeno pokusavaju da ga rezervisu ne bi mogli da izvrse tu akciju jer bi samo prvi uspeo da azurira stanje. Na ovaj nacin svi koji imaju pravo mogu da azuriraju, dok onaj koji pokusa da rezervise a leka nema dovoljno, nece moci to da uradi.

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

Opis:
U ovoj situaciji moze doci do konflikta ukoliko vise korisnika istovremeno zeli da izmeni kolicinu leka u apoteci, da l novom rezervacijom, otkazivanjem ili slicno. Moze se desiti da vise korisnika istovremeno vrsi update nad istim resursom.

Tok:
Postupak rezervisanja leka je objasnjen u prvom primeru, postupak otkazivanja leka je takav da se pacijentu na svom home page-u prikazuju sve rezervacije, odnosno gadja se **DrugReservationController** metoda **findDrugReservationByPatient** I prima parametar email pacijenta, ova metoda vraca sve rezervacije pacijenta, aktivne, otkazane, sve.Iz liste svojih rezervacija pacijent otkazuje zeljenu ukoliko na to ima pravo(ima vise od 24h do preuzimanja) I tako gadja **DrugReservationController** metodu koja prima id rezervacije I rezervacija se otkazuje. Nakon toga se vraca kolicina rezervisanog leka u apoteku gde je lek rezervisan. Do konfliktne situacije moze doci ukoliko vise korisnika istovremeno rezervise ili otkazuje isti lek u istoj apoteci, tacnije menja se kolicina leka.



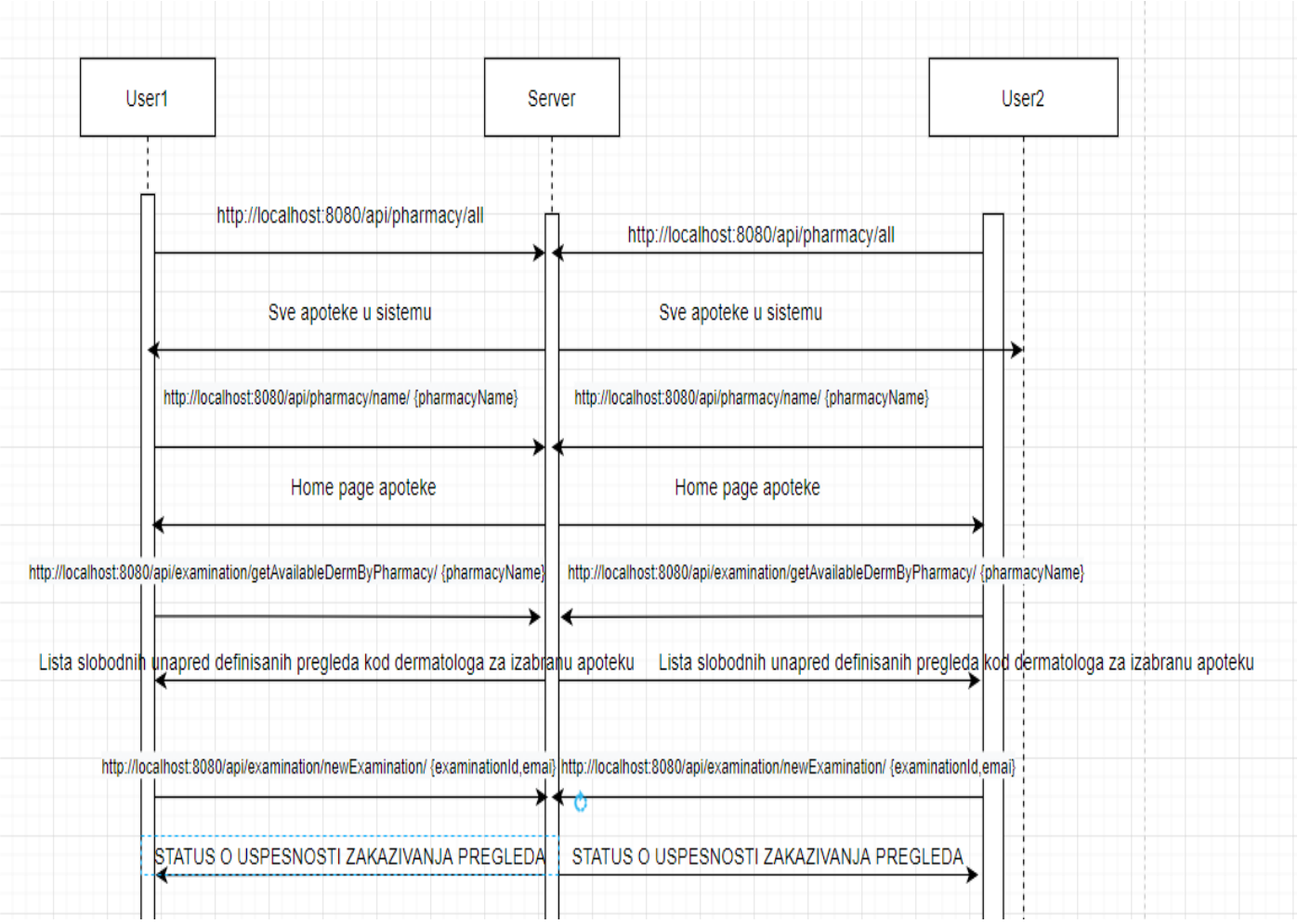
Resenje:

Kako bih resio potencijalnu konfliktnu situaciju iskoristio sam pesimisticku konkurenciju povezivanja na bazu. Zato sto update-ovanje kolicine leka u bazi moze da uradi korisnik koji prvi pristupi bazi, dok ostali moraju da sacekaju. Zato sam u klasu **DrugInPharmacyRepository** ubacio anotaciju @Lock(LockModeType.PESSIMISTIC_READ) I @QueryHints({QueryHint(name = "javax.persistance.lock.timeout", value ="2000")}) iznad metode **findDrugInPharmacy** I prvi korisnik koji pristupi resursu DrugInPharmacy ce jedini moci izvrsi rezervaciju ili otkazivanje I tako ce azurirati stanje leka u apoteci, dok ce ostali moci samo da citaju, bez prava izmene zakljucanog resursa u vremenskom periodu od 2 sekunde. Na ovaj nacin se omogucava da druga transakcija ne bude odmah odbijena nego da saceka 2 sekunde I uspesno obavi zeljenu akciju ukoliko ima pravo na to. U **DrugReservationServiceImpl** sam dodao notacije @Transactional I iznad metoda koje vrse zakazivanje I otkazivanje rezervacije leka, odnosno smanjuje kolicinu leka u apoteci I cuva stanje nakon rezervacije u bazi ili povecava kolicinu nakon otkazivanja rezervacije, metode su **`createDrugReservation`** i **`cancelReservation.`**

Zakazivanje vec definisanog pregleda kod dermatologa od strane vise pacijenata

Opis:
Pri zakazivanju vec unapred definisanih pregleda kod dermatologa od strane pacijenta, pacijent bira apoteku u kojoj zeli da zakaze pregled, zatim mu se otvara stranica sa listom termina I dermatologa koji su dostupni. Pacijent bira unapred definisani termin I zakazuje ukoliko u tom terminu nema vec zakazan pregled ili konsultovanje. Do konfliktne situacije moze doci ukoliko vise pacijenata u isto vreme pokusaju da zakazu isti pregled u istoj apoteci kod istog dermatologa.

Tok:
Na stranici za prikaz svih apoteka pacijent gadja **`PharmacyController`** metodu **`findAll`** koja mu vraca listu svih apoteka u sistemu. Nakon toga korisnik bira apoteku u kojoj zeli da zakaze pregled I gadja **`PharmacyController`** metodu **`findPharmacyByName`** koja mu vraca sve podatke o zeljenoj apoteci I redirektuje ga na home page apoteke. Zatim pacijent zahteva od **`ExaminationControllera`** listu svih dostupnih unapred definisanih pregleda kod dermatologa za zeljenu apoteku u metodi **`getAvailableDermatologistsByPharmacy`**. Nakon prikaza svih pregleda pacijent bira jedan pregled I gadja **`ExaminationController`** metodu **`scheduleExamination`** I zakazuje sebi pregled kod dermatologa ukoliko ima pravo na to.



Resenje:
Ovu konfliktnu situaciju sam resio optimistickom konkurencijom pristupa bazi, dodavanjem `@Version Long version` atributa u `Examination` klasu iz razloga sto prilikom kreiranja unapred definisanog pregleda pacijent je setovan na null. Pomocu version atributa se vrsi provera svaki put da li korisnik ima najnoviju verziju resursa iz baze, ukoliko ima dozvoljen mu je rad sa njim ukoliko nema nema pravo da menja resurs dok ne pokupi najnoviju verziju. Prilikom zakazivanja pregleda pacijent menja vrednost polja Patient u tabeli Examination I setuje ga na samog sebe. Na taj nacin sam omogucio da prvi pacijent koji klikne na zakazivanje pregleda to I uradi dok ostali nece moci da pristupe istom pregledu jer mu je verzija update-ovana I niko vise nece moci da zakaze isti pregled u istom terminu kod istog dermatologa. Takodje u `ExaminationControlleru` sam iznad metode scheduleExamination stavio anotaciju `@Transactional` jer se tu radi update pregleda.