**Mladenka Kokanovic RA50/2017**

1. **Na pristigli zahtjev za godisnjim odmorom od strane farmaceuta moze odgovoriti samo jedan administrator apoteke**

* Opis situacije

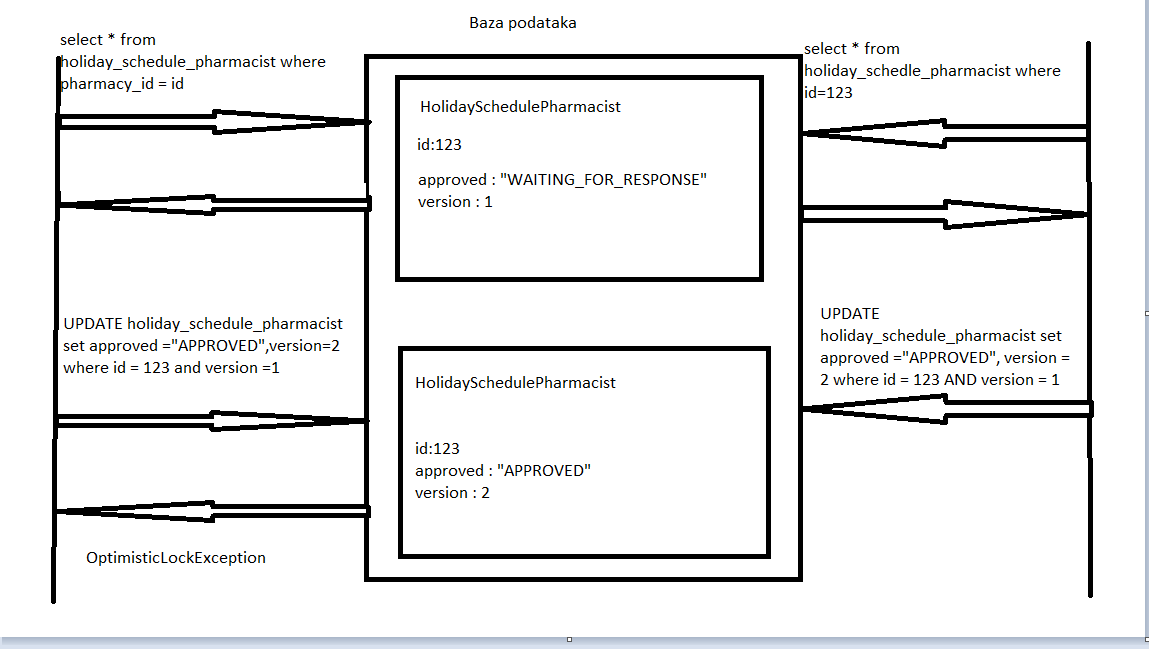
Administartoru apoteke su na raspolaganju svi pristigli zahtjevi za godisnjim odmorima koje on moze da odobri ili da odbije uz obrazlozenje o odbijanju. Klikom na dugme za prihvatanje se salje zahjev na server za azuriranje zahtjeva I mijenjanje vrijednosti odgovarajuce polje iz “WAITING\_FOR\_RESPONSE” u “APPROVED”. Azurirani podaci se cuvaju u bazi.

* Opis problema

Vise administratora apoteke, ili u ovom slucaju dva, kliknu na dugme za odobravanje zahtjeva za godisnjim odmoroom. Prihvatanje zahtjeva ce biti omoguceno samo jednom adminu apoteke jer je on to prvi uradio, dok ce drugom biti onemoguceno.

* Resenje problema

Resenje problema predstavlja upotreba oprimistic locking-a. U klasu modela HolidaySchedulePharmacist je dodato polje private Integer version sa anotacijom @Version. Data anotacija omogucava da se pri cuvanju azurirane informacije o odobravanju zahtjeva prvo provjeri trenutno stanje podataka u bazi. Ukoliko su podaci nepromijenjeni od trenutka ucitavanja baze, status odobravanja godisnjeg odmora se azurira, novi podaci se upisuju u bazu a vrijednost polja version se uvecava za jedan. Kada je rijec o pristupu od strane dva admina, ukoliko je prvi prije kliknuo za dugme za odobravanje godisnjeg odmora, njegov zahtjev brze stize do server I njegova izmjena sacuvati. Sa druge strane, za admina 2 ce pri provjeri trenutnog stanja u bazi dobiti inormacija o promijenjenim podacima jer je admin 1 u medjuvremenu odobrio zahtjev za godisnjim odmorom. Ova situacija dovodi do ObjectOptimisticLockingFailureException-a u klasi HolidaySchedulePharmacist, kada se u metodi approve iz HolidaySchedulePharmacistService pozove holidayScheduleRepository.save(holidaySchedulePharmacist).



1. **Jedan dermatolog ne moze istovremeno da bude prisutan na vise razlicitih pregleda**

* Opis situacije

Administartor apoteke je u mogucnosti da definise termine za preglede kod dermaotologa koji su zaposleni u njeogovoj apoteci. Termini se difinisu u okviru radnog vremena dermatologa i jedan dermatolog, u jednom terminu, moze biti pristan na samo jednom pregledu. Izborom dermatologa, datuma, pocetnog vremena, cijene i trajanja i klikom na dugme za zakazivanje se salje zahtjev na server za cuvanje definisanog termina u tabeli examination\_schedule.

* Opis problema

Vise administratora apoteke, ili u ovom slucaju dva, kliknu na dugme za zakazivanje termina pregleda kod dermatologa u isto vrijeme. Zakazivanje termina ce biti omoguceno samo jednom adminu apoteke, dok ce drugom biti onemoguceno.

* Rjesenje

Resenje ovog problema predstavlja upotreba Pessimistic Lock-a tako sto prvi korisnik koji pristupi resursu, zakljuca ga, a ostali korisnici cekaju dok on ne zavrsi. Dodata je metoda findOneById DermatologistRepository koja ce, za razliku od metode findById, da zakljuca resurs koji vrati pomocu anotacije @Lock. Koristen je pristup PESSIMISTIC\_READ koji onemogucava korisnicima koji nisu prvi pristupili da rade save ili npr upadate metodu, medjutim mogu da obave citanje resursa.

1. **Jedan farmaceut ne moze istovremeno da bude prisutan na vise razlicitih savetovanja**

* Opis situacije

Pacijenti su u mogucnosti da zakazu termine za savjetovanje kod farmaceuta. Izborom farmaceuta, datuma i zeljenog vremena i klikom na dugme za zakazivanje se salje zahtjev na server za zakazivanje savjetovanja u tabeli consulting.

* Opis problema

Vise pacijenata, ili u ovom slucaju dva, kliknu na dugme za zakazivanje savjetovanja kod farmaceuta u isto vrijeme. Zakazivanje savjetovanja ce biti dozvoljeno samo jednom pacijentu dok ce drugom biti zabranjeno.

* Rjesenje

Resenje ovog problema predstavlja upotreba Pessimistic Lock-a tako sto prvi korisnik koji pristupi resursu, zakljuca ga, a ostali korisnici cekaju dok on ne zavrsi. Dodata je metoda findOneById u PharmacistRepository koja ce, za razliku od metode findById, da zakljuca resurs koji vrati pomocu anotacije @Lock. Koristen je pristup PESSIMISTIC\_READ koji onemogucava korisnicima koji nisu prvi pristupili da rade save ili npr. upadate metodu, medjutim mogu da obave citanje resursa. Pomocu vrijednosti value=”10” je korisniku dodijeljen dati vremeski period da obavi zeljenu akciju.

