

Student 1 – konkurentni pristup

Identifikovane konfliktne situacije za funkcionalnosti koje implementira student 1 su sledeće:

1. Zakazivanje pregleda kod dermatologa od strane korisnika
2. Zakazivanje savetovanja kod farmaceuta od strane korisnika
3. Rezervacija lekova od strane korisnika

Pored navedenih funkcionalnosti, nije uočena ni jedna druga situacija gde bi se konflikti mogli javiti.

1. Zakazivanje pregleda kod dermatologa

Pri zakazivanju predefinisano pregleda kod dermatologa, postoji mogućnost da više korisnika pokuša da zakaže jedan termin istovremeno. U ovom slučaju može doći do konflikta, odnosno, moguće je da bi sistem pri proveru dostupnosti termina za jednog pacijenta utvrdio da je pomenuti termin dostupan, ali da pre uspešnog zakazivanja i za zahtev drugog pacijenta takođe utvrdi da je termin dostupan. U tom slučaju sistem bi za jednog korisnika zakazao termin i odmah nakon toga bi zakazao taj isti termin i za drugog korisnika, čime bi prvi termin bio „pregažen“, a oba korisnika bi dobili potvrdu o uspešnom zakazivanju. Ono što je olakšavajuća okolnost kod ovog konflikta je to što onom pacijentu čiji je termin pregažen, ne bi bio prikazan taj termin u listi predstojećih termina.

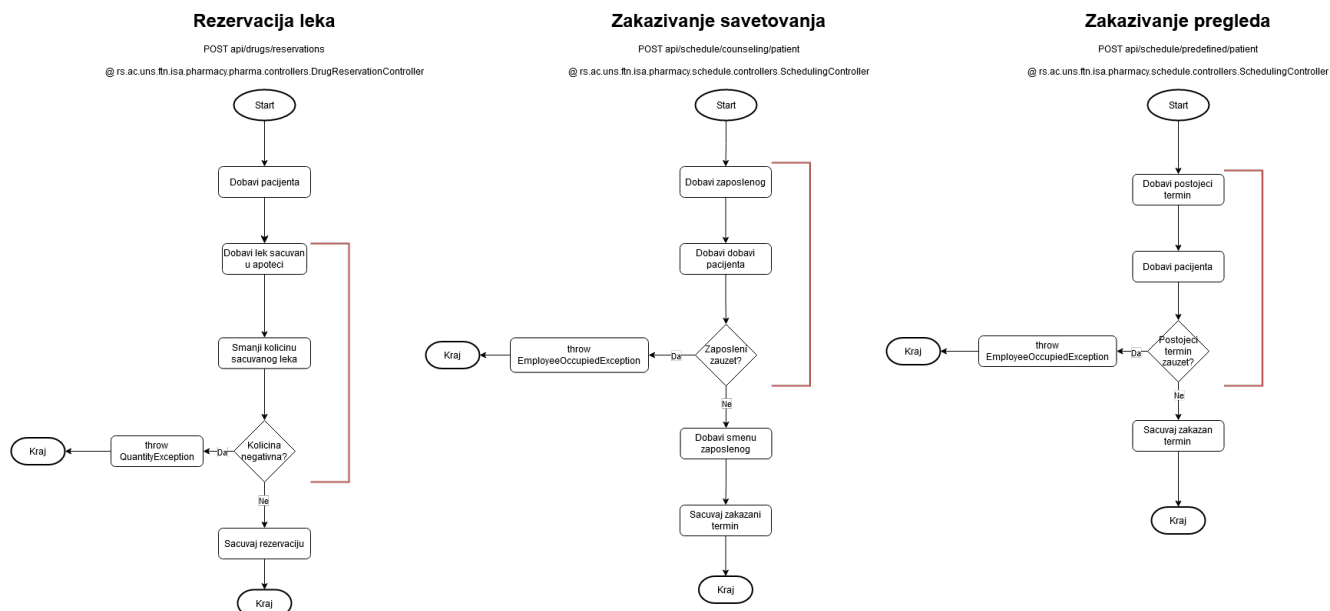
2. Zakazivanje savetovanja kod farmaceuta

Slično zakazivanju pregleda, postoji mogućnost da više korisnika pokuša da zakažu isti termin, ili termine koji se poklapaju. Sistem bi i u ovom slučaju utvrdio da u smeni zaposlenog postoji slobodan interval, te bi moglo da se dogodi da se jedan korisnik uspe da zakaže termin, ali da je za drugog korisnika sistem već utvrdio da je interval u smeni slobodan, pa bi i on uspeo da zakaže termin koji bi se preklapao sa terminom drugog korisnika. Za razliku od slučaja 1, ova situacija je mnogo gora, jer bi oba korisnika imala zakazane termine koji su preklapljeni.

3. Rezervacija lekova

Kada pacijent pošalje zahtev za rezervaciju određenog leka, sistem smanjuje količinu leka u apoteci, a zatim kreira rezervaciju. Količina ne može biti negativna pa korisnik ne može rezervirati lekove koji nisu na stanju. U koliko više korisnika pokuša da rezerviše isti lek iz iste apoteke, može se dogoditi da sistem za višestruke zahteve utvrdi da je količinsko stanje leka u bazi isto, te će izvršiti proveru dostupnosti leka nad istim brojem. Ova situacija ne mora imati ozbiljne posledice ukoliko postoji dovoljna količina leka za sve korisnike, ali kada to nije slučaj, količina leka u apoteci bi mogla da postane negativna.

Prikaz dijagrama toka procesa za navedene funkcionalnosti



Rešenje konfliktnih situacija:

1. Rezezervacija lekova

Kod rezervacije lekova, implementirano je optimističko zaključavanje čuvanih lekova u apoteci. Verzija se čuva u klasi `StoredDrug`, i time se sprečava da se vrši upis u bazu leka sa istom verzijom. Pri izmeni stanja leka u bazi broj verzije se povećava.

Izabrano je optimističko zaključavanje jer je u ovom slučaju bilo lako za implementaciju i ne želimo da zabranimo pristup lekovima čuvanim u apoteci.

2. Zakazivanje savetovanja kod farmaceuta

Kod zakazivanja savetovanja kod farmaceuta, implementirano je pesimističko zaključavanje farmaceuta kada se čitaju iz baze. Rezultat ovoga je da će sistem pri procesu zakazivanja zaključati dobavljenog farmaceuta kod kojeg se zakazuje termin i tako sprečiti čitanje, odnosno proveru dostupnosti farmaceuta od strane drugih niti.

Razlog za korišćenje pesimističkog zaključavanja je taj što se u bazi vrši upisivanje novog termina, prema tome nigde nije moguće voditi računa o verziji enititeta, pa optimističko zaključavanje nije moguće.

3. Zakazivanje pregleda kod dermatologa

Kod zakazivanja pregleda kod dermatologa, implementirano je pesimističko zaključavanje termina kada se čitaju iz baze. Rezultat ovoga je da će sistem pri procesu zakazivanja zaključati dobavljen postojeći termin, a otključaće ga tek po završetku transakcije, odnosno nakon što izvrši proveru i zakazivanje.

U ovom slučaju je bilo moguće raditi i optimističko zaključavanje, gde bi se verzija čuvala u predefinisanom terminu, ali je odabrano pesimističko zbog konzistentnosti sa drugim tipovima zakazivanja.

Za implementaciju navedenih tipova zaključavanja korišćene su anotacije

`@Lock(LockModeType.PESSIMISTIC_WRITE)` za pesimističko zaključavanje

(`rs.ac.uns.ftn.isa.pharmacy.schedule.repository.AppointmentRepository` i

`rs.ac.uns.ftn.isa.pharmacy.users.employee.repository.EmployeeRepository`) i `@Version` za optimističko zaključavanje

(`rs.ac.uns.ftn.isa.pharmacy.pharma.domain.StoredDrug`)