

Resenje konfliktnih situacija - Student 3

Student : Jovan Jelicki RA107/2017

Konfliktne situacije :

1. Jedan dermatolog ne moze istovremeno da bude prisutan na vise razlicitih pregleda
2. Jedan farmaceut ne moze istovremeno da bude prisutan na vise razlicitih savetovanja
3. Pregledi koji su unapred definisani ne smeju biti rezervisani od strane vise korisnika
4. Vise istovremenih korisnika aplikacije ne moze da zakaze savetovanje u istom terminu kod istog farmaceuta (termini se ne smeju preklapati)
5. Davanje penala pacijentu

Pre nego sto se opisu date situacije, napomenuo bih da je **appointment** univerzalna klasa za savetovanje i preglede. Razlikujemo ih na osnovu EmployeeType polja (vrednosti : ROLE_pharmacist i ROLE_dermatologist) i employeeId i na taj nacin dobijamo konkretne preglede/savetovanja za datog usera.

Appointment sadrzi definisan **UniqueConstraint** koji cine 3 polja, a to su : examinerId, employeeType i periodStart. Jedan pregled traje **tacno sat vremena**. PeriodStart je polje tipa LocalDateTime, za koje se u bazi pamti datum i satnica odrzavanja pregleda, gde su minuti i sekundi uvek zaokruzeni na 0, cime smo dobili da se u svakom satu radnog dana moze odrzati tacno jedan pregled.

```
@Entity
@Table(uniqueConstraints={@UniqueConstraint(columnNames={"examinerId", "type", "periodStart"})})
public class Appointment {
```

Iznad svake metode u kojoj moze doci do konflikta, podrška u kodu bila je anotacija @Transactional, sa odredjenim parametrima u zavisnosti od zahteva za tu funkcionalnost.

1. i 3. Dermatolog ne može biti prisutan na istovremeno više različitih pregleda i unapred definisani pregledi ne mogu biti rezervisani od strane više korisnika.

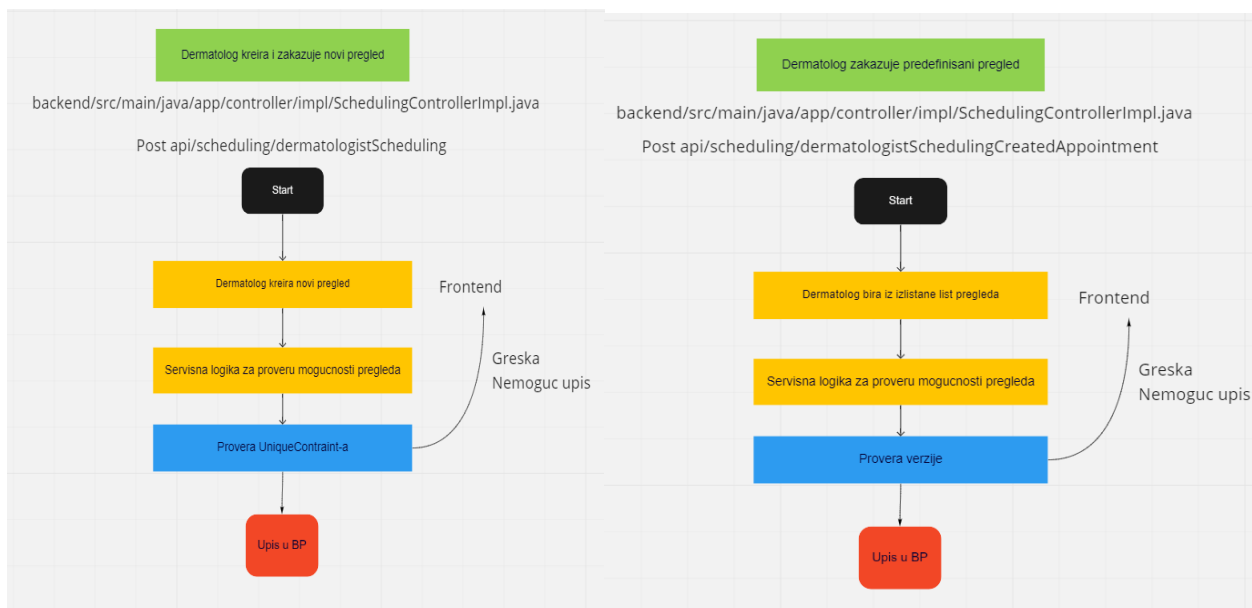
Posmatracemo date tačke u jednoj celini s obzirom da se odnose na zakazivanja pregleda kod dermatologa, i rešenje im je dosta slično.

Pregledi kod dermatologa mogu da se zakazu na dva načina : prvi je da dermatolog ili pacijent mogu da zakazu unapred definisane preglede (u ovom slučaju posmatracemo zakazivanje dermatologa) i drugi jeste da dermatolog u toku pregleda može na osnovu datuma i vremena pokušati da kreira novi pregled i automatski ga zakaze za datog pacijenta.

U prvom načinu dermatolog ima izlistane termine i odabirom jednog, šalje zahtev i pokušava da zakaze izabrani termin za pacijenta koji se nalazi u trenutnom pregledu. Problem može da nastane ukoliko određeni drugi korisnik pokušati da isti termin zakaze ili promeni, što bi dovelo do konflikta.

Ovaj problem je rešen pomoću **optimistic lock** za čuvanje pregleda, gde **appointment** klasa u sebi sadrži **version** atribut. Pregled neće biti promenjen ukoliko prosledjena verzija nije u skladu sa verzijom koja se nalazi sacuvana u bazi. Izvrsice se **rollback** i korisniku će se dati do znanja da je termin zauzet.

Drugi način, dermatolog pokušava da kreira novi pregled koji će automatski biti i zakazan. Konflikt može da nastane ukoliko **pharmacyAdmin** pokušati da kreira slobodan termin u tom periodu. Kako je već napomenuto koristen je **UniqueConstraint** i dati termin neće biti sacuvan, izvrsice se **rollback**.

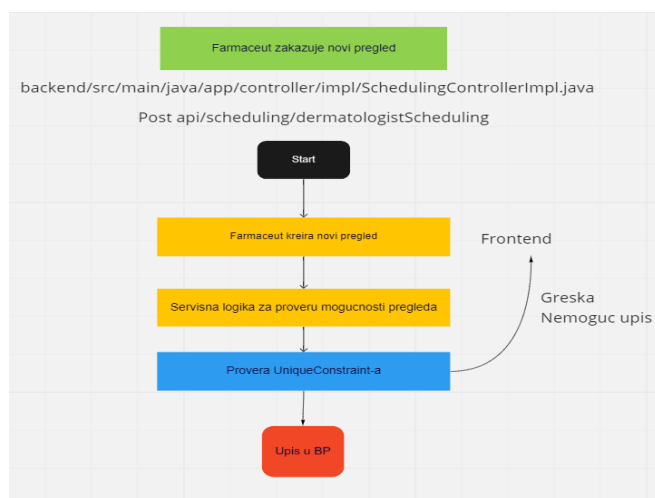


2. i 4. Jedan farmaceut ne moze istovremeno da bude prisutan na vise razlicitih savetovanja i vise istovremenih korisnika aplikacije ne moze da zakaze savetovanje u istom terminu kod istog farmaceuta

Kako su ove dve tacke prilicno slicne sa gledista farmaceuta, opisacemo ih u istoj celini.

Farmaceut moze u toku pregleda da zakaze novi za datog pacijenta. Zakazivanje se vrši na osnovu odabira datuma i satnice, gde se salje zahtev i pokusava da se napravi novi pregled i upise kao zakazan.

Konflikt moze da nastane ukoliko vise korisnika pokusa da zakaze isti termin kod istog farmaceuta. Medjutim, kako je vec napomenuto **appointment** sadrzi **UniqueConstraint** i upisace se samo jedan prosledjeni termin, dok ostali ce dobiti status o gresci, i izvorsice se **rollback**.



5. Davanje penala pacijentu

Davanja penala pacijentu je omoguceno u 2 pristupa.

Prvi pristup je **scheduled** metoda koja se poziva svakoga dana u 12 sati i proverava sve rezervacije lekova apoteke. Ukoliko je odredjena rezervacija istekla (pacijent je nije pokupio do termina kada je naveo), lekovi se vracaju na stanje apoteke, a datom pacijentu se penali uvecavaju za 1.

Drugi pristup jeste da ukoliko se pacijent ne pojavi na pregledu, farmaceutima i dermatolozima je omoguceno da ako je isteklo vreme pregleda pritisnu dugme "did not show up" i na taj nacin obeleze termin u odredjenom statusu i pacijentu dodele jedan penal. Farmaceuti i dermatolozi nisu u prinudi da to pritisnu odma nakon isteka, vec im je dozvoljeno da to izvrse kada god zele nakon isteka pregleda.

Kako bi se sprecila konfliktna situacija da se bodovi ne uracunaju u validnoj vrednosti, poniste ili cak ne promene, koristena je **optimistic lock** za cuvanje datog pacijenta.

Pacijent u sebi sadrzi polje **version**, i time je omogućeno da ce se profil pacijenta izmeniti samo ukoliko je data verzija koja dolazi na upis odgovarajuca verziji koja se nalazi u bazi.

