

Student 1-Konfliktne situacije

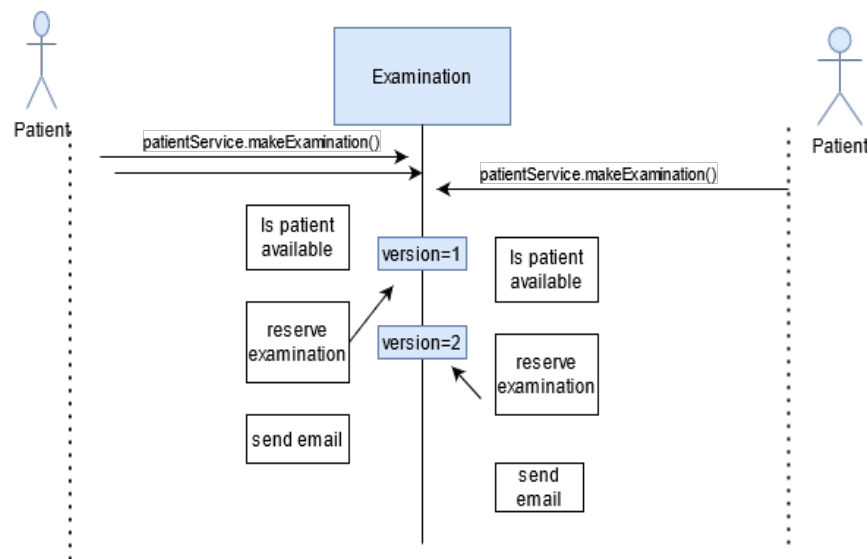
1. Pregledi koji su unaprijed definisani ne smiju biti rezervisani od strane vise razlicitih korisnika

Endpoint: /api/patients/makeExamination/{patientId}

Metoda : makeExamination(patientId,ExaminationId)

Opis:Prilikom rezervacije pregleda moze doci da 2 pacijenta zele da rezervisu isti pregled klikom na dugme Make Examination u priblizno isto vrijeme.U tom slucaju pregled ce biti rezervisan pacijentu koji je prvi rezervisao,a drugom ce se onemoguciti rezervacija.

Opis rjesenja:Rjesenje je postignuto pomocu optimistic locking-a,koristenjem anotacije @Version,u klasi Examination(dodato polje Long version sa anotacijom @Version).Pri cuvanju odgovora i mijenjanja Status polja na scheduled provjerava se verzija podatka koji se cuva.Ukoliko je verzija podatka ista kao pri učitavanju iz baze podaci ce se sacuvati i polje version ce se povecati, i to ce se desiti kada podatke na backend posalje prvi pacijent dok ce se pri slanju podataka na backend u slucaju drugog pacijenta provjeriti verzija i nece odgovarati jer je prvi pacijent vec rezervisao pregled.



2.Jedan lijek ne moze da se rezervise ako je u medjuvremenu postao nedostupan.

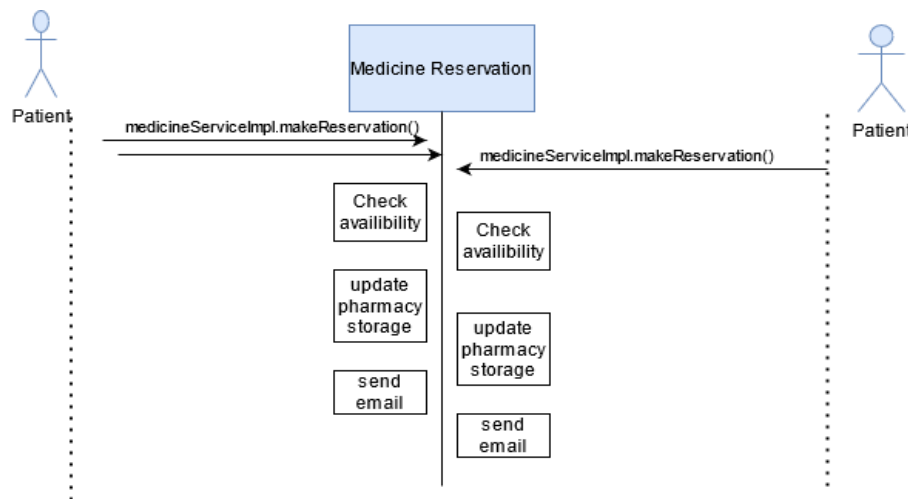
Endpoint :/api/medicines/makeReservation/{patientId}

Metoda: public MedicineReservationDTO makeReservation(MedicineReservationDTO medDto,String patientId) u MedicineServiceImpl

Opis:Prilikom rezervacije lijeka moguca je situacija da dodje do rezervacije lijeka koji je u medjuvremenu postao nedostupan.

Opis rjesenja: Rijeseno je verzionisanjem entiteta PharmacyStorage(dodato polje Long version sa anotacijom @Version).Kada se zapocne proces pokupi se trenutna verzija PharmacyStorage I odmah pre cuvanja ce se izmijeniti trenutna kolicina lijeka u

određenoj apoteci.

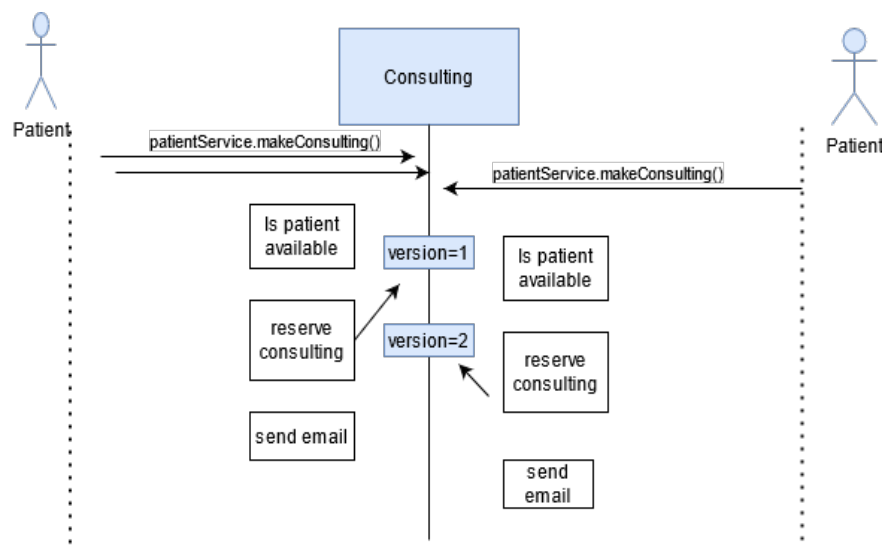


3. Vise istovremenih korisnika aplikacije ne može da zakaze savjetovanje u istom terminu kod istog farmaceuta

Endpoint: `:/api/patient/makeConsulting/{patientId}`

Metoda: `makeConsulting(patientId, consultingId)`

Opis: Rješenje je postignuto pomoću optimističkog locking-a, korištenjem anotacije `@Version`, u klasi `Consulting` (dodato polje `Long version` sa anotacijom `@Version`). Pri cuvanju odgovora i mijenjanja Status polja na `scheduled` provjerava se verzija podatka koji se čuva. Ukoliko je verzija podatka ista kao pri učitavanju iz baze podaci će se sačuvati i polje `version` će se povećati, i to će se desiti kada podatke na backend pošalje prvi pacijent dok će se pri slanju podataka na backend u slučaju drugog pacijenta provjeriti verzija i neće odgovarati jer je prvi pacijent već rezervisao pregled.



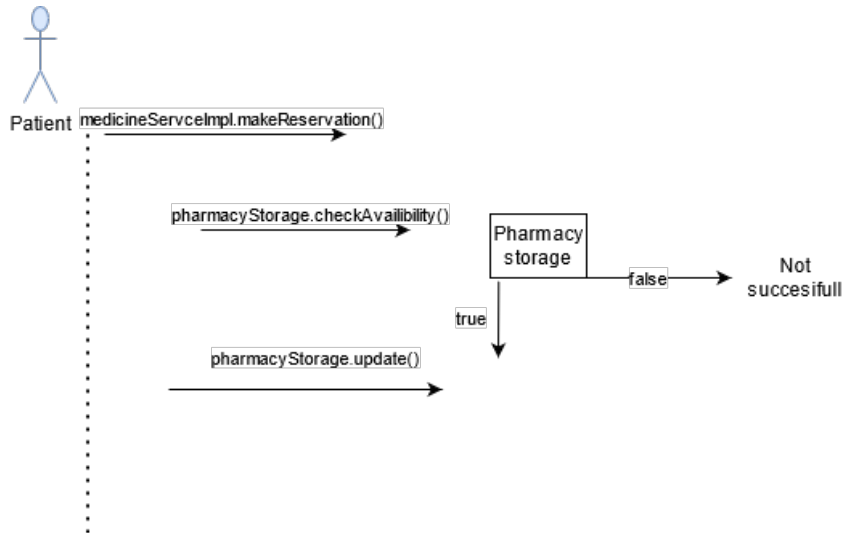
4. Količina lijeka na stanju se mora ispravno ažurirati nakon rezervacije lijeka od strane pacijenta

Endpoint: `/api/medicines/makeReservation/{patientId}`

Metoda: `public void updateStorage(Long medId, Long phId)` u `PharmacyStorageService`

Opis: Prilikom rezervacije lijeka od strane pacijenta potrebno je pravilno ažurirati stanje lijeka u magacinu apoteke iz koje se lijek rezervise

Opis rjesenja: Optimisticko zakljucavanje metode jer dolazi do ažuriranja tabele (pharmacy storage) koja nam daje informaciju o tome da li je lijek dostupan, uz pomoc `@Transactional` anotacije, `readOnly` je `false`, jer metoda mijenja podatke.



5. Pretraga apoteka

Endpoint: `/api/pharmacies/searchPharmacies`

Metoda: `public List<Pharmacy> searchPharmacies(SearchPharmacyDTO spDto)`,

Opis: U toku pretrage apoteka može doći do dodavanja nove ili izmjene postojeće. Pacijent neće uočiti promjene. Phantom read problem.

Rješenje: `@Transactional` anotacija, gdje je `readOnly` postavljen na `true`, zbog isčitavanja podataka