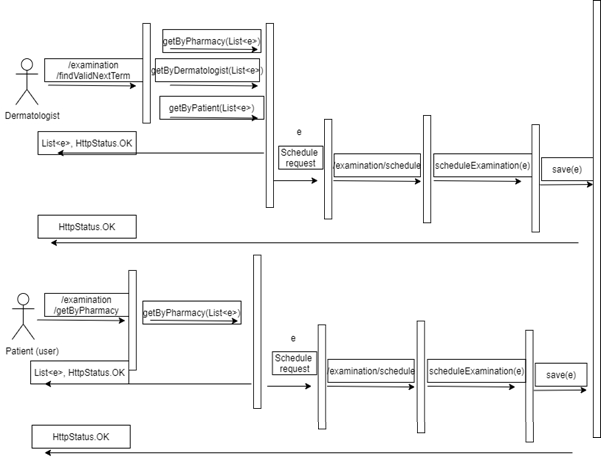
**Konkurentni pristup resursima u bazi. STUDENT 3**

1. **Jedan dermatolog ne može istovremeno da bude prisutan na više različitih pregleda.**

Dermatolog ima opciju da odabere novi pregled za pacijenta, ili da odabere iz liste definisanih pregleda. Konfliktna situacija ovdje nastaje u slučaju kada bi on i drugi korisnik (pacijent) u isto vrijeme odabrali isti, unaprijed definisan pregled i pokušali ga zakazati.

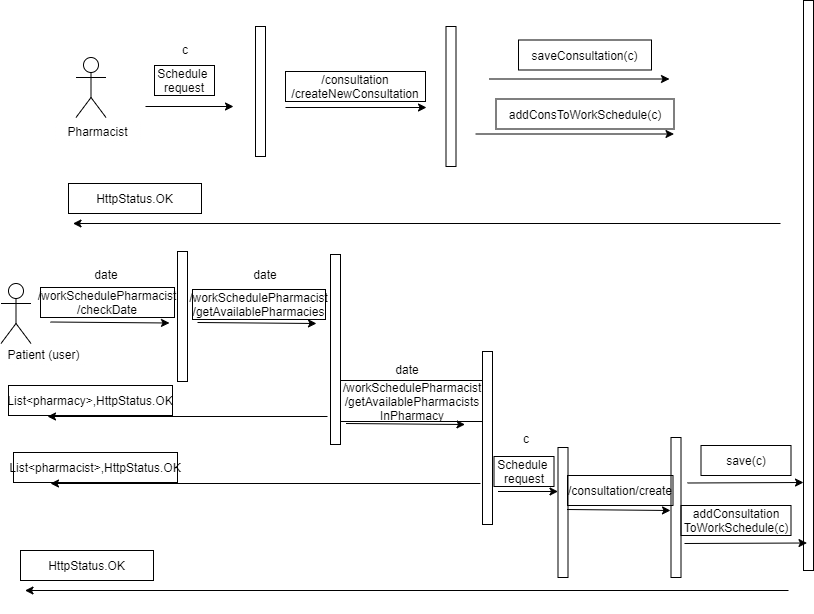


**Prikaz konfliktne situacije broj 1.**

Ovakav problem se rješava optimistiškim zaključavanjem. Dodajemo polje version (private Long Version) u entitet Examination i iznad njega anotaciju @Version. (Za to koristimo *import javax.persistence.Version;)*. Kada bi dermatolog(pacijent) pokušao prvi da zakaže pregled, pokupio bi trenutno stanje u bazi, odnosno trenutnu verziju, i ako bi se ta verzija poklopila i na kraju, zakazivanje bi bilo uspješno. Ako bi pacijent(dermatolog) tada pokušao zakazati isti termin, dok prvi korisnik nije završio zakazivanje, pokupio bi drugu verziju i javio bi se sljedeći izuzetak: ObjectOptimisticLockingFailureException. Naime, uvodimo @Transactional iznad metode za zakazivanje: scheduleExamination, i time obezbjeđujemo da se baza ne poremeti kada dođе do ovakvog izuzetka.

1. **Jedan farmaceut ne može istovremeno da bude prisutan na više različitih pregleda.**

Farmaceut ne može sam sebi zakazati više pregleda koji bi se desili u isto vrijeme. Međutim, pacijent bi mogao da rezerviše sebi novi pregled u istom trenutku dok to radi i farmaceut (ili neki drugi korisnik), što bi dovelo do konfliktne situacije, jer bi se 2 identična pregleda našla u random kalendaru farmaceuta.

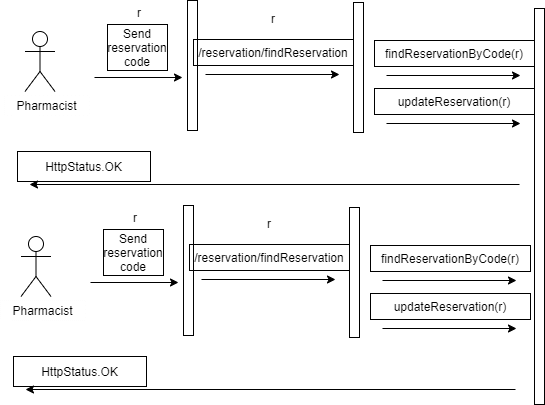


**Prikaz konfliktne situacije broj 2.**

Ovaj put koristimo pesimističko zaključavanje uz pomoć anotacije za zaključavanje reda u tabeli. (@Lock(LockModeType.PESSIMISTIC\_WRITE)). Time je omogućeno da dok jedan korisnik zakazuje, metoda bude zaključana (odnosno, nemoguće je upisati taj termin u tabelu).

1. **Više farmaceuta ne može u isto vrijeme izdati lijek koji je pacijent rezervisao.**

Kada bi dva farmaceuta (u istoj apoteci) pokušali izdati rezervaciju pod istim kodom u isto vrijeme, nastala bi konfliktna situacija. Pacijent bi dobio dva mejla za tu rezervaciju, odnosno 2 puta preuzeo istu, a u apoteci bi se smanjila količina jednom.



**Prikaz konfliktne situacije broj 3.**

Da bi riješili ovakvu konfliktnu situaciju, morali smo optimistički zaključati. Tačnije, u entitet Reservation dodajemo polje version(odnosno private Long Version) i iznad njega anotaciju @Version. Kada bi farmaceut pristupio rezervaciji u bazi, promijenio bi status rezervacije u FINISHED, a drugi bi u isto vrijeme pokupio staru verziju i time bi došlo do izuzetka (ObjectOptimisticLockingFailureException). Da bi bazu popravili, dodajemo anotaciju @Transactional iznad metode kojom tražimo željenu rezervaciju (findReservationByCode).