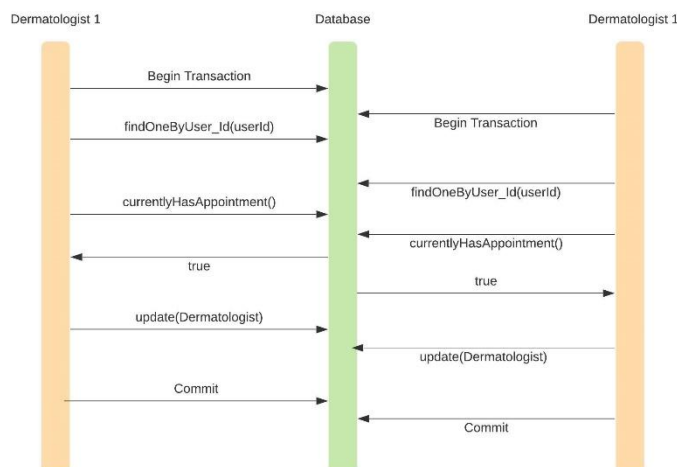


Konfliktne situacije – student 3 (Milijana Đorđević)

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

Opis konfliktne situacije: Nakon što se uloguje, dermatolog ima pristup svom radnom kalendaru. Iz radnog kalendara, klikom, on može započeti novi pregled ili može naznačiti da pacijent nije došao. Pošto naš sistem dozvoljava da se pregled započne i ranije nego što počinje - problem nastaje kada jedan isti dermatolog pokuša da započne više od jednog pregleda. Veći deo problema rešen je samom implementacijom. U klasu Dermatologist dodato je polje `currentyHasAppointment` čije se stanje mora proveriti upitom prilikom započinjanja pregleda. Ukoliko je polje `false`, pregled se može započeti, pa se polje postavlja na `true`. Međutim, i dalje postoji mogućnost da jedan isti dermatolog pokuša da započne dva različita pregleda u isto vreme – u tom slučaju dolazi do konflikta.

Crtež: Dermatolog započinje prvu transakciju, a zatim i drugu. Sa obe strane šalje se upit ka bazi za dobavljanje dermatologa preko njegovog id-ja, a zatim se proverava stanje polja `currentyHasAppointment`. Kako još uvek stanje polja nije izmanjeno ni sa jedne strane, `update` će se odraditi nad jednim te istim dermatologom dva puta i započinjanje pregleda za obe strane biće dozvoljeno. Ovako može doći do početka dva (ili više) pregleda odjednom i, samim tim, rad aplikacije nije ispravan.

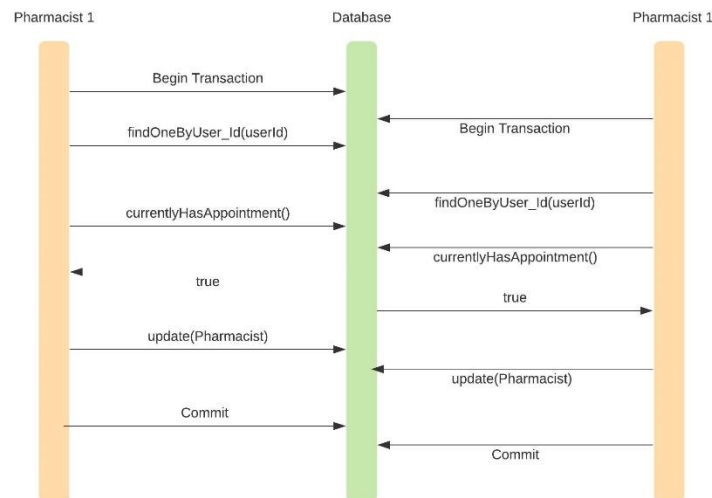


Rešenje: Kada dermatolog pokuša da započne pregled treba onemogućiti novo započinjanje pregleda za istog dermatologa dok se proverava i `update` stanja polja `currentyHasAppointment` ne završi. Nakon što dođe do `commita`, dermatolog će ponovo moći da pokuša proverava i menja polje `currentyHasAppointment`. Problem je rešen pomoću optimističkog zaključavanja. U klasi `Dermatologist` uveden je novi atribut `Long version` (pomoću anotacije `@Version`). Kada se u prvoj transakciji odradi `update` dermatologovog polja `currentyHasAppointment`, verzija će se inkrementovati za jedan pa će se u drugoj javiti izuzetak jer se verzije podataka ne poklapaju.

2. Jedan farmaceut ne može istovremeno da bude prisutan na više različitih savetovanja

Opis konfliktne situacije: Kao i kod dermatologa, nakon što se uloguje, farmaceut ima pristup svom radnom kalendaru. Iz radnog kalendara, klikom, on može započeti novi pregled ili može naznačiti da pacijent nije došao. Pošto naš sistem dozvoljava da se pregled započne i ranije nego što počinje - problem nastaje kada jedan isti farmaceut pokuša da započne više od jednog pregleda. Veći deo problema rešen je samom implementacijom. U klasu Pharmacist dodato je polje `currentyHasAppointment` čije se stanje mora proveriti upitom prilikom započinjanja pregleda. Ukoliko je polje `false`, pregled se može započeti, pa se polje postavlja na `true`. Međutim, i dalje postoji mogućnost da jedan isti dermatolog pokuša da započne dva različita pregleda u isto vreme – u tom slučaju dolazi do konflikta.

Crtež: Farmaceut započinje prvu transakciju, a zatim i drugu. Sa obe strane šalje se upit ka bazi za dobavljanje farmaceuta preko njegovog id-ja, a zatim se proverava stanje polja `currentyHasAppointment`. Kako još uvek stanje polja nije izmanjeno ni sa jedne strane (postavljeno je na `false`), update će se odraditi nad jednim te istim farmaceutom dva puta i započinjanje pregleda za obe strane biće dozvoljeno. Ovako može doći do početka dva (ili više) pregleda odjednom i, samim tim, rad aplikacije nije ispravan.

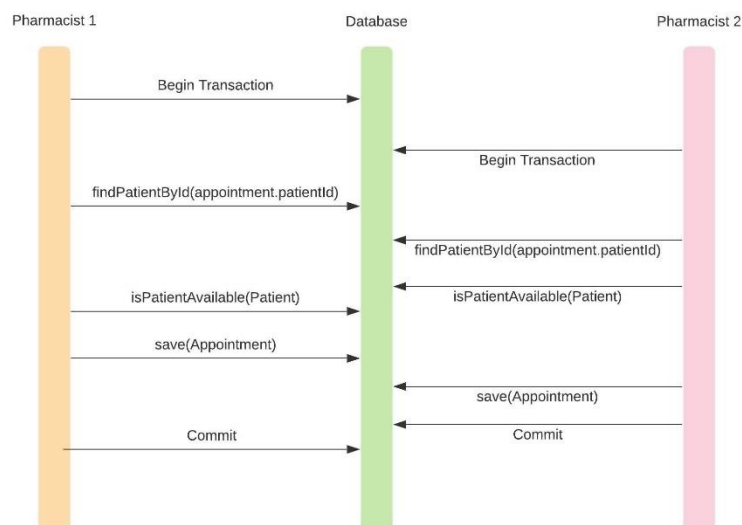


Rešenje: Kada farmaceut pokuša da započne pregled treba onemogućiti novo započinjanje pregleda za istog farmaceuta dok se proverava i update stanja polja `currentyHasAppointment` ne završi. Situacija je rešena na isti način kao i kod dermatologa - pomoću optimističkog zaključavanja. U klasi Pharmacist uveden je novi atribut `Long version` (pomoću anotacije `@Version`). Kada se u prvoj transakciji odradi update farmaceutovog polja `currentyHasAppointment`, verzija će se inkrementovati za jedan pa će se u drugoj javiti izuzetak jer se verzije podataka ne poklapaju. Tako će druga (ili ostale) transakcije (koje pokušaju da započnu pregled menjanjem polja `currentyHasAppointment`) biti sprečene.

3. Jednom pacijentu se istovremeno ne sme zakazati više pregleda od strane farmaceuta ili dermatologa

Opis konfliktne situacije: Farmaceut nakon završenog savetovanja (ili pregleda – ako je u pitanju dermatolog) ima mogućnost da pacijentu zakaže novo savetovanje odabirom datuma i vremena. Tom prilikom proverava se da li pacijent ima već zakazanu obavezu. Problem nastaje kada dva različita farmaceuta istovremeno pokušaju da zakažu savetovanje jednom istom pacijentu u istom terminu. Ista situacija se javlja i ako su u pitanju dva dermatologa ili farmaceut i dermatolog.

Crtež: Farmaceut 1 započinje transakciju, zatim farmaceut 2 započinje transakciju. Oba farmaceuta vrše proveru da li je pacijent slobodan u zadato vreme. Kako termin još uvek nije zauzet, oba farmaceuta dobijaju odgovor da je pacijent slobodan. Nakon toga, oba farmaceuta čuvaju novo savetovanje u bazi za istog pacijenta u isto vreme. Ovako može doći do čuvanja dve različite obaveze za pacijenta (zakazane u isto vreme) i do neispravnog rada aplikacije.



Rešenje: Kako bi drugom farmaceutu trebalo zabraniti pristup pacijentu u trenutku kada prvi vrši proveru nad istim pacijentom – situacija je rešena pesimističkim zaključavanjem pacijenta. Metoda `findPatientById` u repozitorijumu za pacijenta biće ista kao obična metoda `findById`, ali će usput raditi i zaključavanje pacijenta koji se datom metodom izvuče iz baze (`LockModeType.PESSIMISTIC_WRITE`). Tako nijedna druga transakcija neće moći da pristupi istom pacijentu dok se prva transakcija ne završi. Ista stvar događa se i u slučaju dva dermatologa ili farmaceuta i dermatologa: javiće se izuzetak `PessimisticLock` ukoliko istovremeno pokušaju da zakažu pregled/savetovanje istom pacijentu i dobiće adekvatnu poruku da ne mogu zakazati pregled.