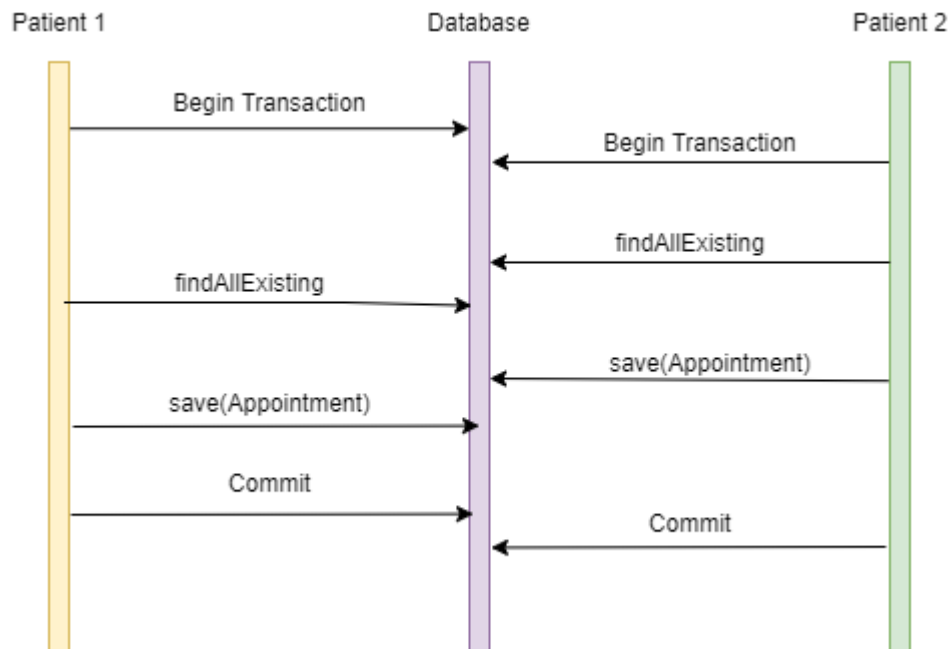


## Konfliktne situacije – student 1 (Eva Janković sw18/2017)

### 1) Više pacijenata istovremeno zakazuje pregled u isto vreme kod istog doktora

*Opis konfliktne situacije:* Nakon logovanja u aplikaciju pacijent može da na svojoj početnoj stranici odabere zakazivanje novog pregleda. Pacijent prvo bira datum i tip pregleda nakon čega mu se pokazuju klinike koje zadovoljavaju zadati kriterijum pretrage. Odabirom klinike prikazuju se doktori u klinici koji su specijalizovani za odabran tip pregleda i imaju slobodnih termina za odabran datum. Pacijent bira satnicu pregleda i prikazuju mu se informacije o odabranom pregledu. Nakon toga pacijent može da potvrdi informacije o pregledu i pošalje zahtev za pregled serveru, ili odustane od zakazivanja pregleda i vrati se na prethodnu stranicu. Do konflikta dolazi ukoliko dva ili više pacijenata pokušaju da zakažu pregled u istom terminu kod istog doktora. Takođe, do konflikta dolazi i ako pacijent i administrator klinike zakazuju pregled u istom terminu kod istog doktora.

*Crtež konfliktne situacije:* Pacijent 1 započinje transakciju, zatim pacijent 2 započinje transakciju i šalje upit ka bazi za pronalaženje preklapanja pregleda kod odabranog doktora za odabrano vreme. Nakon toga pacijent 1 šalje isti upit ka bazi i dobija iste podatke. Oba pacijenta kao odgovor od baze dobijaju da je doktor slobodan za zadat termin, zatim šalju upit za čuvanje pregleda u bazu i nakon toga komituju svoje izmene. Nakon obe transakcije se u bazi nalaze dva pregleda za istog doktora i isti termin i dolazimo do nedozvoljenog stanja baze aplikacije i neispravnog rada same aplikacije.



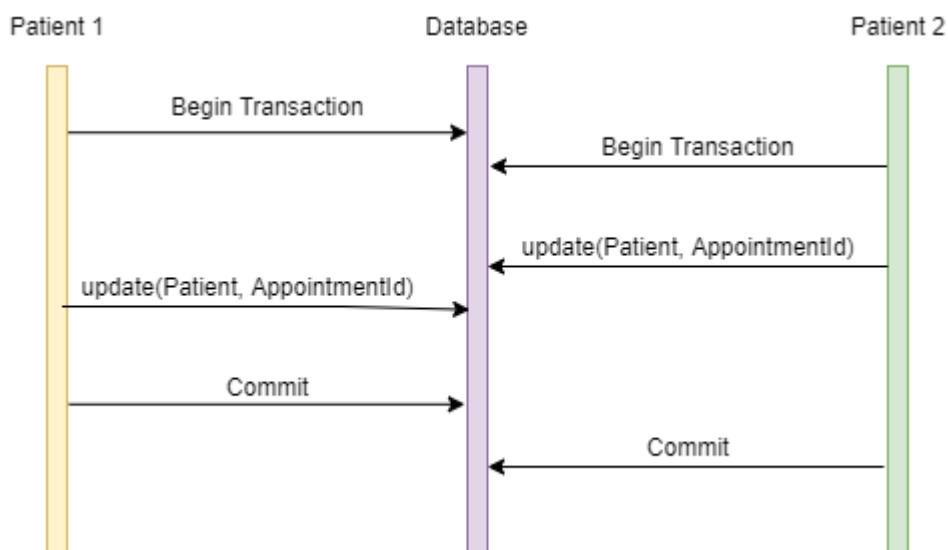
*Opis načina rešavanja konfliktne situacije:* Za rešavanje ove konfliktne situacije podrazumevani nivo izolacije aplikacije (READ\_COMMITTED) nije bio dovoljna. Bilo je potrebno podići nivo izolacije metode za zakazivanje pregleda na najviši nivo (SERIALIZABLE). U metodi za zakazivanje pregleda se vrši upit ka bazi

koji proverava da li za zadanog doktora postoji zakazan pregled/operacija koji se preklapa sa zadatim terminom. Ukoliko upit ka bazi ne vrati nijedan entitet doktor je slobodan za zadati vremenski period i administratorima klinike se šalje zahtev za pregled. Ukoliko tokom rada ove transakcije bilo koja druga transakcija pokuša da izmeni iste entitete, dolazi do greške i korisnik koji je pokrenuo drugu transakciju se obaveštava da pregled koji želi da zakaže nije više dostupan.

## 2) Više pacijenata istovremeno zakazuje isti predefinisani pregled

*Opis konfliktne situacije:* Administratori klinike mogu unapred da kreiraju predefinisane preglede za svoju kliniku. Predefinisani pregledi imaju odabranog lekara, tip pregleda, vreme početka, vreme kraja pregleda, datum pregleda i ordinaciju u kojoj će se održati pregled. Nakon logovanja u aplikaciju pacijent može na svom početnom ekranu da odabere pregled svih klinika koje se nalaze u okviru kliničkog centra. Odabirom klinike prikazuje se profil klinike sa osnovnim informacijama o klinici. Na profilu klinike pacijent može da odabere pregled svih predefinisanih pregleda. Problem nastaje ukoliko dva ili više korisnika istovremeno pokušaju da zakazu isti predefinisani pregled.

*Crtež konfliktne situacije:* Pacijent 1 započinje transakciju, zatim pacijent 2 započinje transakciju. Oba pacijenta rezervišu isti predefinisani pregled i komituju svoje izmene. Nakon završetka obe transakcije u bazi podataka se nalaze predefinisani pregled sa Pacijentom 2, što predstavlja pojavu LostUpdate-a, što dovodi bazu u nedozvoljeno stanje i aplikaciju u neispravno stanje.



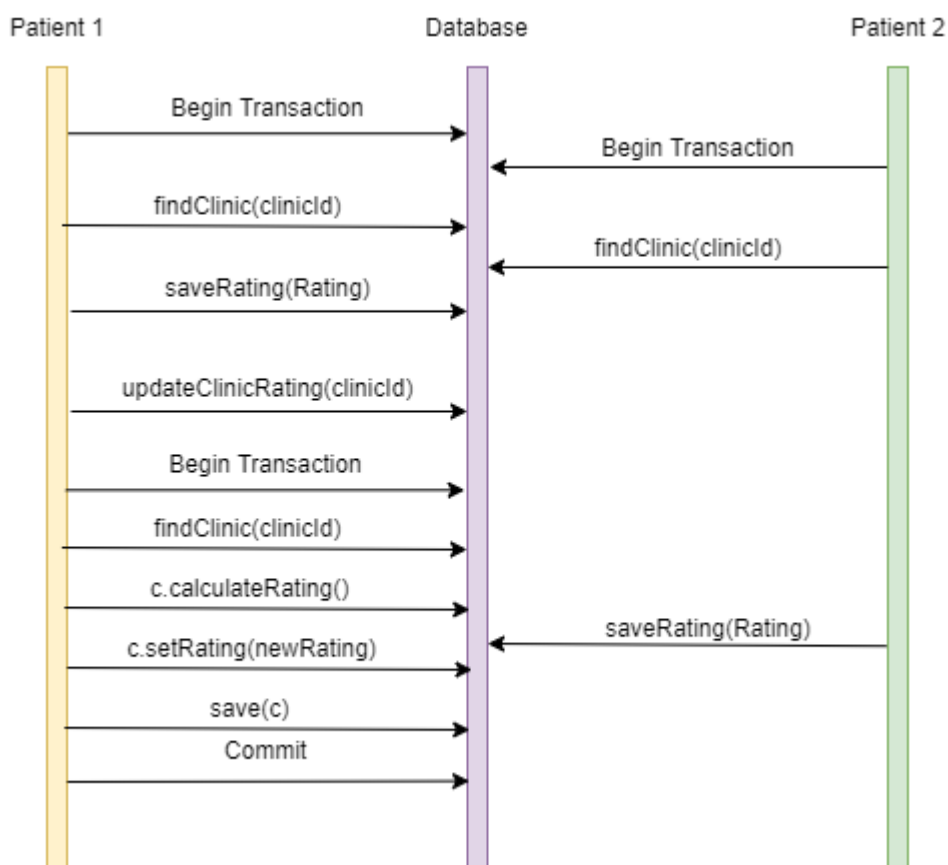
*Opis rešenja konfliktne situacije:* Kada jedan pacijent pristupi objektu klase Appointment potrebno ga je zaključati za sve druge transakcije koje žele da rade sa njim. Nakon komitovanja transakcije objekat se otključava i naredna transakcija može da ga preuzme. Način na koji je ovaj problem rešen je optimističko zaključavanje objekta klase Appointment. Da bi optimističko zaključavanje bilo moguće potrebno je u klasi Appointment uvesti novi atribut Long Version sa anotacijom @Version. Kada prva transakcija uzme i izmeni objekat Appointment (dodeli mu pacijenta) njegov Version atribut će se inkrementovati za jedan. Ukoliko naredna transakcija pokuša da izmeni objekat klase Appointment pre nego što je prethodna transakcija komitovala svoje promene dobiće izuzetak jer se verzija objekta u transakciji i

objekta u bazi ne podudaraju. Na ovaj način je rešen problem konkurentnog zakazivanja istog predefinisnog pregleda od strane dva ili više pacijenata.

### 3) Više pacijenata ocenjuje istu kliniku/doktora<sup>1</sup>

*Opis konfliktne situacije:* Pacijent kada se uloguje ima mogućnost da vidi sve doktore i klinike koje je do tada posetio, njihove ocene i, u slučaju da ih je već ocenio, ocene koje im je on dodelio. Kada dva ili više pacijenata istovremeno ocenjuju istu kliniku/doktora, može doći do konflikte situacije pri računanju ukupne ocene klinike/doktora (računa se na osnovu liste svih dodeljenih ocean toj klinici/doktoru).

*Crtež konfliktne situacije:* Pacijent 1 započinje transakciju i dobavlja kliniku/doktora i Pacijent 2 radi isto to. Pacijent 1 čuva novi Rating i započinje transakciju za osvežavanje ukupnr ocene klinike/doktora. Osvežavanje se sastoji iz sabiranja svih vrednosti ocena i deljenja tog broja sa ukupnim brojem ocena. Pacijent 2, u trenutku kada je završeno računanje zbira svih ocena ali pre deljenja sa ukupnim brojem ocena, dodaje novu ocenu za istu kliniku/doktora i time povećava ukupan broj ocena za tu kliniku/doktora. Time dolazimo do konflikta i ukupna ocena koja je izračunata ne odgovara pravom stanju, te je baza nekonzistentna, a aplikacija ne radi pravilno.



<sup>1</sup> Opisana situacija nije od velikog značaja jer ne utiče na rad aplikacije značajno (nema velike štete ako pacijent vidi nekorektnu ocenu klinike/doktora)

*Opis rešenja konfliktne situacije:* Kada jedan pacijent pristupi objektu klase Clinic/Doctor potrebno ga je zaključati za sve druge transakcije koje žele da rade sa njim. Nakon komitovanja transakcije objekat se otključava i naredna transakcija može da ga preuzme. Način na koji je ovaj problem rešen je optimističko zaključavanje objekta klase Clinic/Doctor. Da bi optimističko zaključavanje bilo moguće potrebno je u klasi Clinic/Doctor uvesti novi atribut Long Version sa anotacijom @Version. Kada prva transakcija uzme i izmeni objekat Clinic/Doctor (dodeli mu novu ocenu) njegov Version atribut će se inkrementovati za jedan. Ukoliko naredna transakcija pokuša da izmeni objekat klase Clinic/Doctor pre nego što je prethodna transakcija komitovala svoje promene dobiće izuzetak jer se verzija objekta u transakciji i objekta u bazi ne podudaraju. Na ovaj način je rešen problem konkurentnog ocenjivanja klinike/doktora od strane dva ili više pacijenata.