

Konfliktne situacije, student 1, tim 7

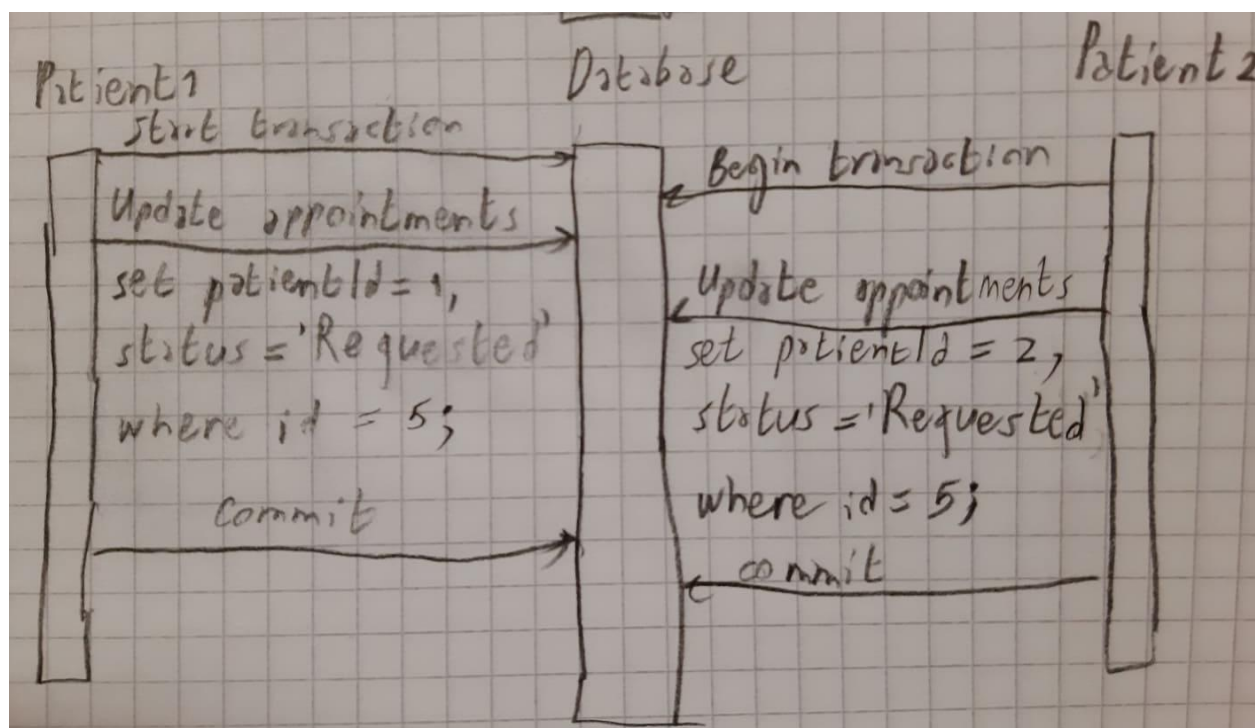
Situacija 1: Više pacijenata pokuša da istovremeno rezerviše isti predefinisani pregled.

Opis situacije: Jedna od stranica koje pacijent može da vidi jeste tabela koja sadrži spisak predefinisanih pregleda. Pacijent klikom na dugme može da rezerviše pregled za sebe. Konflikt može da nastane ako dva pacijenta pokušaju da istovremeno rezervišu isti predefinisani pregled.

Opis rešenja: Kada pacijent klikne na dugme za rezervaciju predefinisano pregleda, zahtev se šalje AppointmentController-u, u proceduru reservePredefined. Reserve predefined dalje zove AppointmentService metodu reserveAppointment, koja je tranzakciona. Metoda reserveAppointment prvo učitava odgovarajući AppointmentPOJO iz baze podataka, na osnovu id appointmenta, koji se prosleđuje sa front-enda. Prva provera je da li appointment još uvek postoji. Ako appointment sa biranim id ne postoji u bazi, ili ako mu je vrednost polja isDeleted jednaka true (appointment je logički obrisan), reserveAppointment vraća false, jer pacijent ne može da rezerviše appointment. Ako je status učitano appointmenta i dalje AVAILABLE, što u našem sistemu označava da je predefinisani pregled još uvek slobodan, status appointmenta preći će u stanje APPROVED, i dodeliće mu se pacijent, na osnovu id pacijenta prosleđenog zahtevom sa front-enda, i reserveAppointment vraća true, kao indikator uspeha. Ukoliko status appointmenta nije AVAILABLE, pretpostavimo da je drugi pacijent već zauzeo appointment, pa će reserveAppointment vratiti false. Boolean odgovor se dalje putem AppointmentControllera vraća front-endu. Ako je vraćeno true (tj. ako je pacijent uspešno rezervisao predefinisani pregled), dobiće modalni dijalog sa obaveštenjem o uspehu, a kada ga ugasi, biće preusmeren na stranicu sa pregledom svojih budućih pregleda, na kojoj se nalazi upravo rezervisani pregled. Ako je vraćeno false, pacijent dobija modalni dijalog sa obaveštenjem o neuspehu uz izvinjenje. Kada ga ugasi, ostaće na stranici sa predefinisanim pregledima, ali će na njoj vrednosti biti osvežene, da bi bolje predstavile tekuće stanje u sistemu.

Cela procedura reserve appointment tretira se kao jedna transakcija, sa nivoom izolacije READ_UNCOMMITTED, koji dozvoljava paralelno čitanje podataka iz tabele sa appointmentima drugim korisnicima, ali bez prava da išta menjaju u tabeli. Procedura proveri da li pacijentu trenutno može da dodeli appointment, ako može dodeli mu ga, i na kraju vrati true ili false, kao indikator uspeha ili neuspeha. U međuvremenu ostali korisnici imaju pravo da pregledaju sadržaj tabele s pregledima, koja obuhvata sve predefinisane preglede, preglede zahtevane od strane pacijenata ili lekara, preglede odobrene od strane administratora, preglede potvrđene od strane pacijenata i završene preglede. Nivo izolacije biran je da spreči da više pacijenata pokuša da istovremeno rezerviše isti predefinisani pregled, ali da pritom što manje uspori rad ostalim korisnicima. Tabela sa appointmentima je jedna od najpristupačnijih tabela u sistemu, svi tipovi korisnika osim administratora centra imaju neke funkcionalnosti vezane za nju, i pretpostavka je bila da bi bilo kakvo ograničenje prava na čitanje tabele značajno usporilo rad velikom udelu naših korisnika. Potencijalni nedostatak READ_UNCOMMITTED nivoa izolacije jeste taj što korisnici mogu da vide stanje sistema dok se ono menja, pre nego što se transakcija

potpuno završila. Ali u ovom konkretnom slučaju, gde je jedina izmena prelazak pregleda iz stanja dostupno u stanje nedostupno, drugi korisnici koji čitaju iz tabele u toku transakcije mogu ili da vide da je pregled dostupan iako će do kraja transakcije postati nedostupan, ili da vide da je nedostupan pre kraja transakcije. Prvi slučaj nije idealan, pacijent može da pomisli da mu je neki pregled dostupan iako ne bi trebao biti, ali ako pokuša da ga rezerviše za sebe sistem će mu javiti da je termin već zauzet. Pretpostavka je da se ovo neće često dešavati, zato što je mala verovatnoća da veliki broj pacijenata u istom trenutku pokuša da rezerviše isti pregled, i zato što je transakcija relativno kratka (tri if-a, jedan upit i jedan update) i ne bi smela dugo da traje. Drugi slučaj, kada je pregled već prešao u stanje zauzet, ali transakcija nije gotova, i drugi korisnik traži da čita tabelu, ne predstavlja problem, zato što transakcija svakako hoće zauzeti pregled, ali ovako će pregled delovati zauzeto \leq ukupno vreme transakcije ranije nego što bi formalno trebalo.



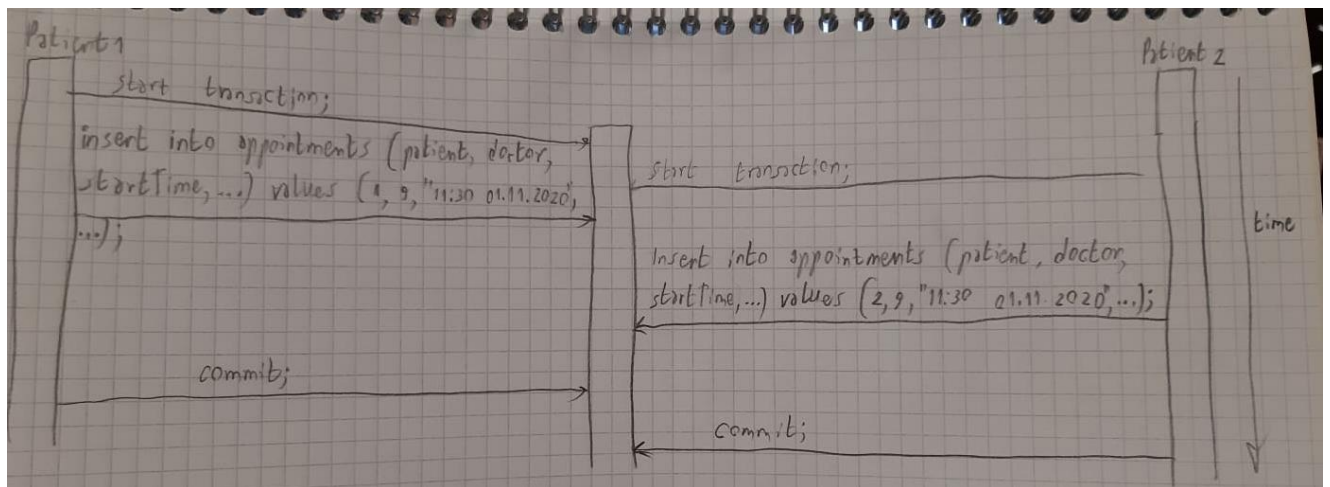
Situacija 1: Više pacijenata pokuša da istovremeno rezerviše isti predefinisani pregled.

Opis situacije: Jedna od stranica koje pacijent može da vidi jeste tabela koja sadrži spisak predefinisanih pregleda. Pacijent klikom na dugme može da rezerviše pregled za sebe. Konflikt može da nastane ako dva pacijenta pokušaju da istovremeno rezervišu isti predefinisani pregled.

Opis rešenja: Kada pacijent klikne na dugme za rezervaciju predefinisnog pregleda, zahtev se šalje AppointmentController-u, u proceduru reservePredefined. Reserve predefined dalje zove AppointmentService metodu reserveAppointment, koja je tranzakciona. Metoda reserveAppointment prvo učitava odgovarajući AppointmentPOJO iz baze podataka, na osnovu id appointmenta, koji se prosleđuje sa front-enda. Prva provera je da li appointment još uvek postoji. Ako appointment sa biranim id ne postoji u bazi, ili ako mu je vrednost polja isDeleted jednaka true (appointment je logički obrisani), reserveAppointment vraća false, jer pacijent ne može da rezerviše appointment. Ako je status učitaniog appointmenta i dalje AVAILABLE, što u našem sistemu označava da je predefinisani pregled još uvek slobodan, status appointmenta preći će u stanje APPROVED, i dodeliće mu se pacijent, na osnovu id pacijenta prosleđenog zahtevom sa front-enda, i reserveAppointment vraća true, kao indikator uspeha. Ukoliko status appointmenta nije AVAILABLE, pretpostavimo da je drugi pacijent već zauzeo appointment, pa će reserveAppointment vratiti false. Boolean odgovor se dalje putem AppointmentControllera vraća front-endu. Ako je vraćeno true (tj. ako je pacijent uspešno rezervisao predefinisani pregled), dobiće modalni dijalog sa obaveštenjem o uspehu, a kada ga ugasi, biće preusmeren na stranicu sa pregledom svojih budućih pregleda, na kojoj se nalazi upravo rezervisani pregled. Ako je vraćeno false, pacijent dobija modalni dijalog sa obaveštenjem o neuspehu uz izvinjenje. Kada ga ugasi, ostaće na stranici sa predefinisanim pregledima, ali će na njoj vrednosti biti osvežene, da bi bolje predstavile tekuće stanje u sistemu.

Cela procedura reserve appointment tretira se kao jedna transakcija, sa nivoom izolacije READ_UNCOMMITTED, koji dozvoljava paralelno čitanje podataka iz tabele sa appointmentima drugim korisnicima, ali bez prava da išta menjaju u tabeli. Procedura proveri da li pacijentu trenutno može da dodeli appointment, ako može dodeli mu ga, i na kraju vrati true ili false, kao indikator uspeha ili neuspeha. U međuvremenu ostali korisnici imaju pravo da pregledaju sadržaj tebele s pregledima, koja obuhvata sve predefinisane preglede, preglede zahtevane od strane pacijenata ili lekara, preglede odobrene od strane administratora, preglede potvrđene od strane pacijenata i završene preglede. Nivo izolacije biran je da spreči da više pacijenata pokuša da istovremeno rezerviše isti predefinisani pregled, ali da pritom što manje uspori rad ostalim korisnicima. Tabela sa appointmentima je jedna od najpristupanijih tabela u sistemu, svi tipovi korisnika osim administratora centra imaju neke funkcionalnosti vezane za nju, i pretpostavka je bila da bi bilo kakvo ograničenje prava na čitanje tabele značajno usporilo rad velikom udelu naših korisnika. Potencijalni nedostatak READ_UNCOMMITTED nivoa izolacije jeste taj što korisnici mogu da vide stanje sistema dok se ono menja, pre nego što se transakcija potpuno završila. Ali u ovom konkretnom slučaju, gde je jedina izmena prelazak pregleda iz stanja dostupno u stanje nedostupno, drugi korisnici koji čitaju iz tabele u toku transakcije mogu ili da vide da je pregled dostupan iako će do kraja transakcije postati nedostupan, ili da vide da je nedostupan pre kraja transakcije. Prvi slučaj nije idealan, pacijent može da pomisli da mu je neki pregled dostupan iako

ne bi trebao biti, ali ako pokuša da ga rezerviše za sebe sistem će mu javiti da je termin već zauzet. Pretpostavka je da se ovo neće često dešavati, zato što je mala verovatnoća da veliki broj pacijenata u istom trenutku pokuša da rezerviše isti pregled, i zato što je transakcija relativno kratka (tri if-a, jedan upit i jedan update) i ne bi smela dugo da traje. Drugi slučaj, kada je pregled već prešao u stanje zauzet, ali transakcija nije gotova, i drugi korisnik traži da čita tabelu, ne predstavlja problem, zato što transakcija svakako hoće zauzeti pregled, ali ovako će pregled delovati zauzeto \leq ukupno vreme transakcije ranije nego što bi formalno trebalo.



Situacija 3: Pacijent pokuša da pristupi profilu lekara u trenutku dok ga administrator klinike briše iz sistema

Opis situacije: Pacijentu je na na stranici sa profilom klinije prikazan spisak svih lekara koji rade u toj klinici. Klikom na profil lekara, pacijent prelazi na stranicu sa kratkim pregledom informacija o datom lekaru. Nezavisno od pacijenta, administrator klinike ima pravo da bilo kog lekara iz svoje klinike obriše. Pošto je brisanje logičko, kada pacijent pokuša da pristupi podacima o lekaru preko lekarovog id, moguće je da pacijent očita podatke o lekaru koji je neposredno pre upita obrisani.

Opis rešenja: Kada pacijent klikne na lekara čiji profil ga zanima, zahtev se prosleđuje u DoctorController, u proceduru getProfilePreview. Procedura dalje zove metodu getProfilePreview iz DoctorService. getProfilePreview prvo očita doktora iz baze, i to pomoću transakcione metode getIfNotDeleted, a zatim formatira odgovarajući DTO, koji se vraća DoctorController-ovoj metodi getProfilePreview, koja ga vraća pacijentu. GetIfNotDeleted je transakciona, i prilikom njenog poziva se potpuno zaključa tabela sa lekarima u bazi. Model brave je PESSIMISTIC_READ, koji blokira da bilo koja metoda čita vrednosti iz tabele dok je brava aktivna. Metoda prvo očita lekara iz baze na osnovu njegovog id, potom proveriti da li je lekar logički obrisani ili ne na osnovu vrednosti isDeleted polja, i ako lekar nije obrisani vrati DoctorPOJO koji opisuje očitaniog lekara, a ako jeste obrisani vrati null, i korisniku se vraća odgovor da fajl nije pronađen. Pesimističko zaključavanje nam ovde garantuje da će pacijenti moći da vide samo podatke o lekarima koji u trenutku pokretanja upita nisu obrisani. Nije idealna sa aspekta efikasnosti, zato što u toku transakcionog upita ni jedan drugi korisnik neće imati pravo pristupa tabeli sa lekarima. Ipak, pretpostavljamo da se lekarima neće toliko često pristupati kao pregledima, i da je ova transakcija dovoljno mala i kratka, te da performans sistema neće značajno patiti, a konzistentnost je osigurana.

