Konfliktne situacije - Student 1

Stefan Kandić
stefan.kandic@yahoo.com

June 14, 2020

1 Odgovor na zahtev za pregled

Svaka klinika ima jednog ili više admina koji mogu koristiti aplikaciju u isto vreme. To može dovesti do neželjene situacije kada dva admina u isto vreme odgovaraju na isti zahtev. Odbijanje zahteva podrazumeva njegovo brisanje iz tabele sa svim zahtevima, dok prihvatanje zahteva podrazumeva brisanje iz tabele sa zahtevima i kreiranje datog zahteva u tabelu sa drugim potvrdjenim pregledima. Pošto odgovor na zahtev može zahtevati više od jedne operacije koja menja podatke u bazi ovde može doći do neželjenih situacija.

1.1 Reject - Approve

Prva neželjena situacija je Reject - Approve, tj. prvo jedan admin odbija zahtev, i zatim drugi admin, koji još uvek vidi taj zahtev u aplikaciji ga prihvata. Problem koji može nastati je to da iako zahtev koji je prvi stigao briše pregled, drugi zahtev ima sve informacije o pregledu i uspeva da ga potvrdi iako se on više ne nalazi u bazi. Na slici 1 možete videti tok tih zahteva.

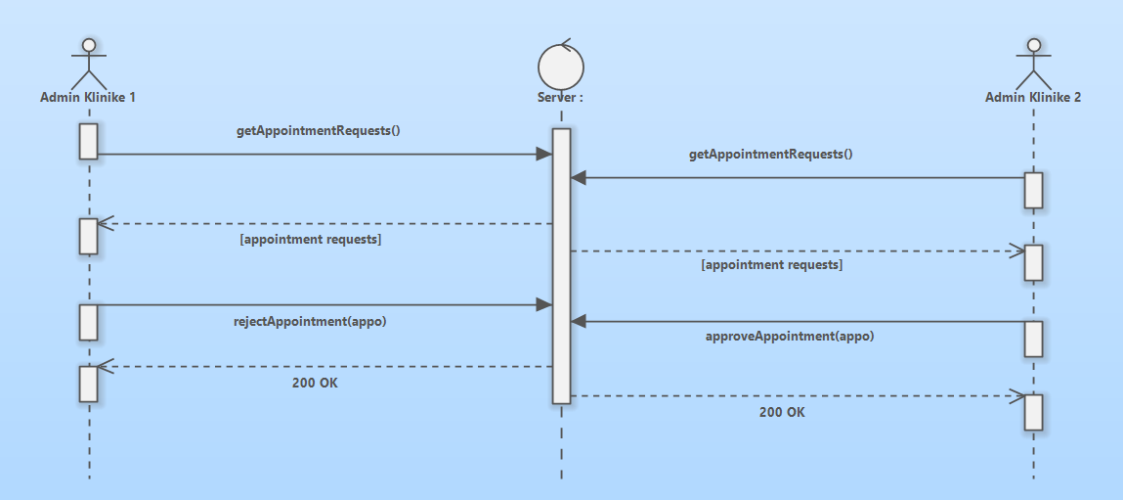

Figure 1: Tok zahteva za zakazivanje pregleda

1.2 Rešenje

Ovo je rešeno enkapsulacijom odgovora na zahtev Sequlelize transakcijom. To znači da iako bi dodavanje u potvrdjene preglede prošlo, brisanje će baciti grešku nakon čega će doći do rollback-a i vraćanje baze u

konzistentno stanje. Zahtev će biti odbijen, a admin koji je drugi poslao zahtev će dobiti poruku da je na zahtev već odgovoreno i osvežiće mu se zahtevi tako da vidi samo one koji nisu potvrđeni ili odbijeni.

1.3 Ostali slučaji

Situacije Reject - Reject, Accept - Reject, Accept - Accept, kao i slučaj kada dva pacijenta u isto vreme potvrđuju predefinisan pregled su dosta slični te nema potrebe da ih detaljnije objašnjavam. Sistem će prednost dati onom ko je bio prvi, dok će svi ostali dobiti grešku, kao i nove podatke iz baze.

2 Odgovor na zahtev za registraciju

Administratori klinike takođe odlučuju o zahtevima za registraciju pacijenata, ali odgovor na ove zahteve se razlikuje od odgovora na zahteve za pregled. Naime, pri odobravanju ili odbijanju zahteva za registraciju dolazi samo do promene polja statusa zahteva za registraciju iz PENDING u APPROVED ili REJECTED. Ono što se može desiti je da administratori pošalju različite odgovore na zahtev koji će iz statusa PENDING preći u APPROVED a zatim u REJECTED ili obratno. Pacijent će dobiti dva kontradiktorna mejla o statusu svog naloga a u bazi će se nalaziti izmena administratora koji je poslednji poslao izmenu. Struktura toka zahteva je identična kao na slici 1.

2.1 Rešenje

Za razliku od prošlog primera, korišćenje samo transakcija neće biti dovoljno, jer formalno ne dolazi do narušenja konzistentnosti baze podataka, što se nje tiče zahtev za registraciju može proizvoljan broj puta menjati status. Zbog toga trebamo uvesti zaključavanje reda u bazi koji se menja. Naime, prvi zahtev koji stigne će zaključati red na početku transakcije, i tek nakon završetka transakcije će vratiti ključ. Svi ostali će biti blokirani dok je red zaključan, i nako što dobiju pristup videće da status reda više nije PENDING te da je zahtev već obrađen.

3 Jedinstvenost vrednosti u tabeli

Ograničenja kao što su to da više pacijenata ne može zatražiti pregled kod istog lekara u isto vreme, i da doktor u isto vreme ne može biti prisutan na više pregleda su rešena na isti način.
Naime, iako NodeJS iz ugla programera radi kao single-threaded aplikacija, u produkciji bi imali više pokrenutih instanci gde bi svaka imala konekciju sa bazom, i zbog toga bi provera ovih uslova na aplikativnom nivou bila potencijalno problematična jer bi se moglo desiti da prođe na više mašina ukoliko dođe do preplitanja upita između zahteva.

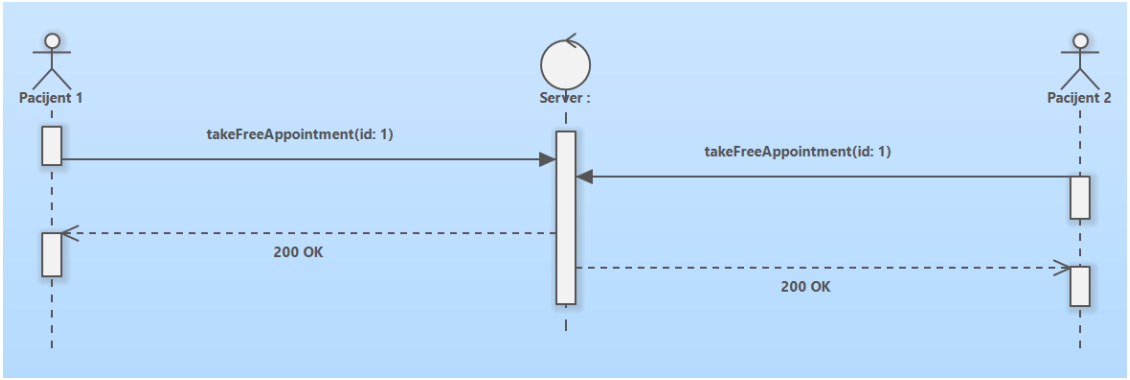

Figure 2: Zauzimanje slobodnog pregleda

3.1 Rešenje

Za instituciju kao što je klinički centar, navedeni problemi mogu imati ozbiljne posledice. Zbog toga su ova ograničenja ugradjena u samu bazu:

- svaki red u tabeli confirmed appointments mora imati jedinstvene vrednosti parova (doctorId, start), (patientId, start), (roomId, start)
- svaki red u tabeli appointment requests mora imati jedinstvenu vrednost parova (patientMedicalRecordId, start) što znači da pacijent ne može zatražiti više pregleda za isti vremenski termin. Primetite da kod zahteva može biti više doktora vezano za isti termin, to neće biti problem jer će admin klinike morati da promeni termin zahteva na termin kada je izabrani doktor slobodan.

Uprkos tome što imamo više konekcija ka našoj bazi, SQL baze izvršavaju operacije sekvencijalno, te bi samo prvo dodavanje u tabelu moglu da zadovolji ograničenje jedinstvene vrednosti, dok bi sva nakon njega uzrokovale grešku i tako održale bazu u konzistentnom stanju.