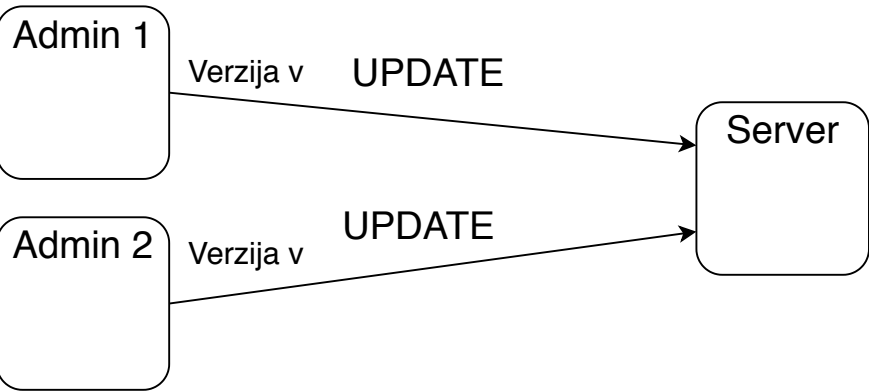


UOČENI PROBLEMI

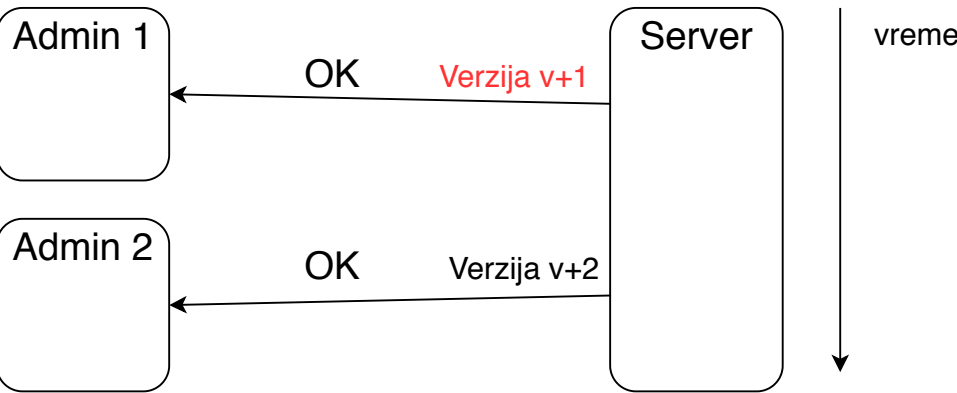
Ažuriranje profila klinike

Scenario: Dva administratora klinike u isto vreme šalju zahtev za ažuriranje svoje klinike.

Simptom: Obojici je javljeno da je ažuriranje uspešno, međutim jedan od njih vidi pogrešne podatke.



Jedan od zahteva će se izvršiti prvi, ažurirati podatke ali će to potencijalno pregaziti drugi zahtev koji će se izvršiti neposredno posle prvog.
Ipak, oba korisnika su dobili potvrđan odgovor i opravdano misle da su trenutno podaci u bazi oni koji su njima prikazani, iako onaj čije ažuriranje je pregaženo vidi stariju verziju.



Problem izgubljenog ažuriranja (lost-update) ne pravi štetu u ovom konkretnom slučaju, ali je loša praksa u opštem smislu. Trebalo bi da klijent i server budu u sinhronizaciji što je više moguće.

Ažuriranje nerezervisane sobe

Analogno prethodnom.

Ažuriranje nekorišćenog tipa pregleda

Analogno prethodnom.

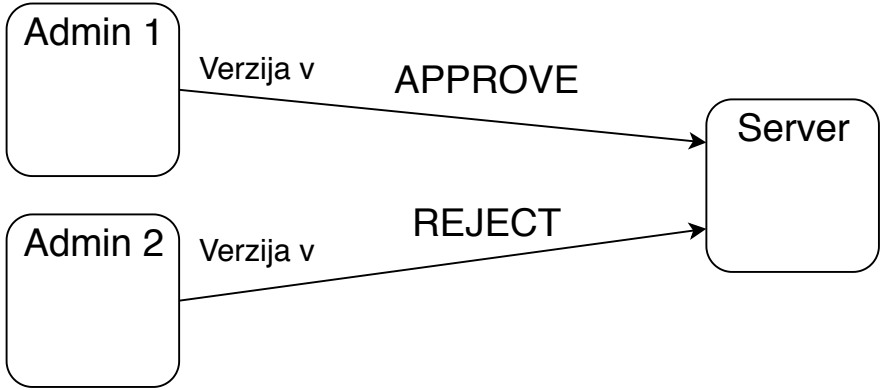
Ažuriranje slobodnog termina

Analogno prethodnom.

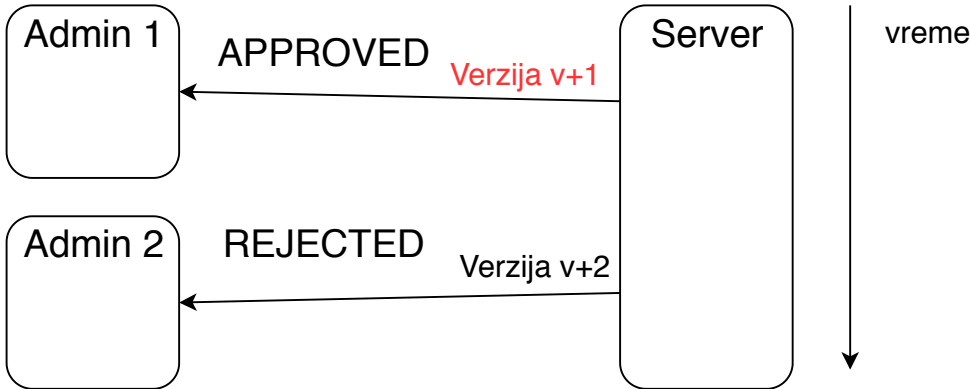
Odgovaranje na zahtev za pregled

Scenario: Dva administratora u isto vreme odgovaraju na zahtev za pregled. Prvi ga prihvata, drugi ga odbija.

Simptom: Prvom je javljeno da je prihvaćeno, drugom da je odbijeno dok je stvarnosti prošao samo jedan od tih zahteva.



Uzrok problema je isti kao u prethodnom primeru, međutim posledice u ovom slučaju mogu zaista biti štetne. Pacijent će dobiti oba obaveštenja, a svaki administrator klinike će smatrati da je njegova operacija uspešna i poslednja.



Odgovaranje na zahtev za odsustvo/godišnji odmor

Analogno prethodnom.

REŠENJA

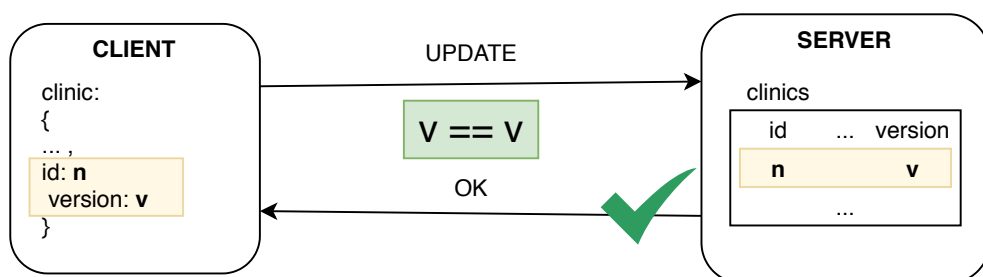
Optimistično zaključavanje

Važno je da korisnik uvek radi s najsvežijim podacima te je odlučeno da sistem odbija ažuriranje koje potiče od korisnika sa starijom verzijom istih.

Za sve funkcionalnosti ažuriranja je uveden mehanizam optimističnog zaključavanja redova. Sledećim tabelama je dodata kolona version koja označava verziju reda u toj tabeli. Nakon svakog ažuriranja reda se inkrementira ova kolona.

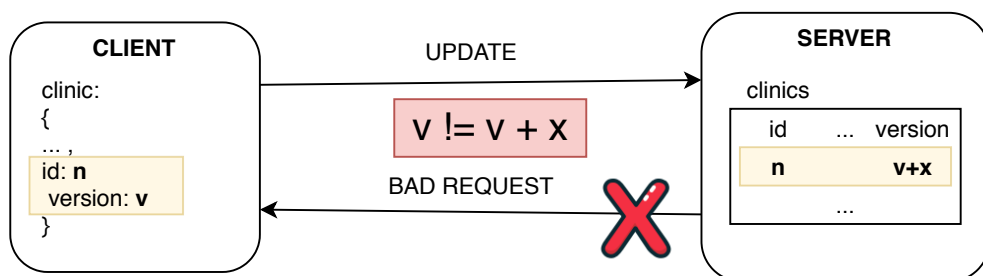
- clinics (klinike)
- free_appointments (slobodni termini)
- rooms (sobe)
- appointment_type (tip pregleda)
- appointment_request (zahtev za pregled)
- leave_request (zahtev za odsustvo/god. odmor)

Dakle, ispravan zahtev koji bi sistem prihvatio bi izgledao ovako nekako:



Verzije se poklapaju, dakle korisnik radi s najsvežijim podacima i ažuriranje se dozvoljava.

Neispravan zahtev bi izgledao ovako nekako:



Verzije se ne poklapaju, dakle korisnik radi sa starijom verzijom tog reda i ažuriranje je onemogućeno.

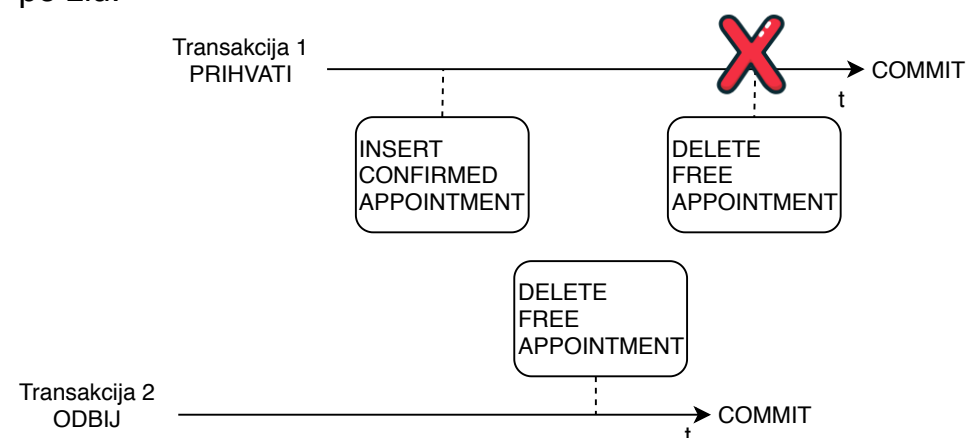
U ovom slučaju korisnik se obaveštava i učitavaju se najsvežiji podaci u aplikaciju. Kontrola se vraća njemu, te može da pokuša ponovo ukoliko želi.

Pakovanje upita bazi u transakcije

Ukoliko operacija uključuje više međuzavisnih upita ka bazi a važno je da uspeju sve ili nijedna primenjuje se mehanizam transakcije.

Prilikom prihvatanja zahteva za pregled prvo se kreira nov potvrđen pregled, a onda se briše zahtev. Prilikom odbijanja zahteva se prosto on obriše.

Tako su svi upiti ka bazi koji se tiču prihvatanja zahteva za pregled spakovani u transakciju koja će vratiti bazu na početno stanje (pre prvog upita unutar transakcije) ukoliko nešto pođe po zlu.



Zahtev za odbijanje se "ubacio" između dva upita u toku prihvatanja, te se postupak prihvatanja abortira i kreirani potvrđen pregled se briše.

Zahtev za odsustvo je jednostavniji jer se sastoji samo iz jednog upita: ažurira mu se status na prihvaćen ili odbijen. Ovo je rešeno optimističnim zaključavanjem.

