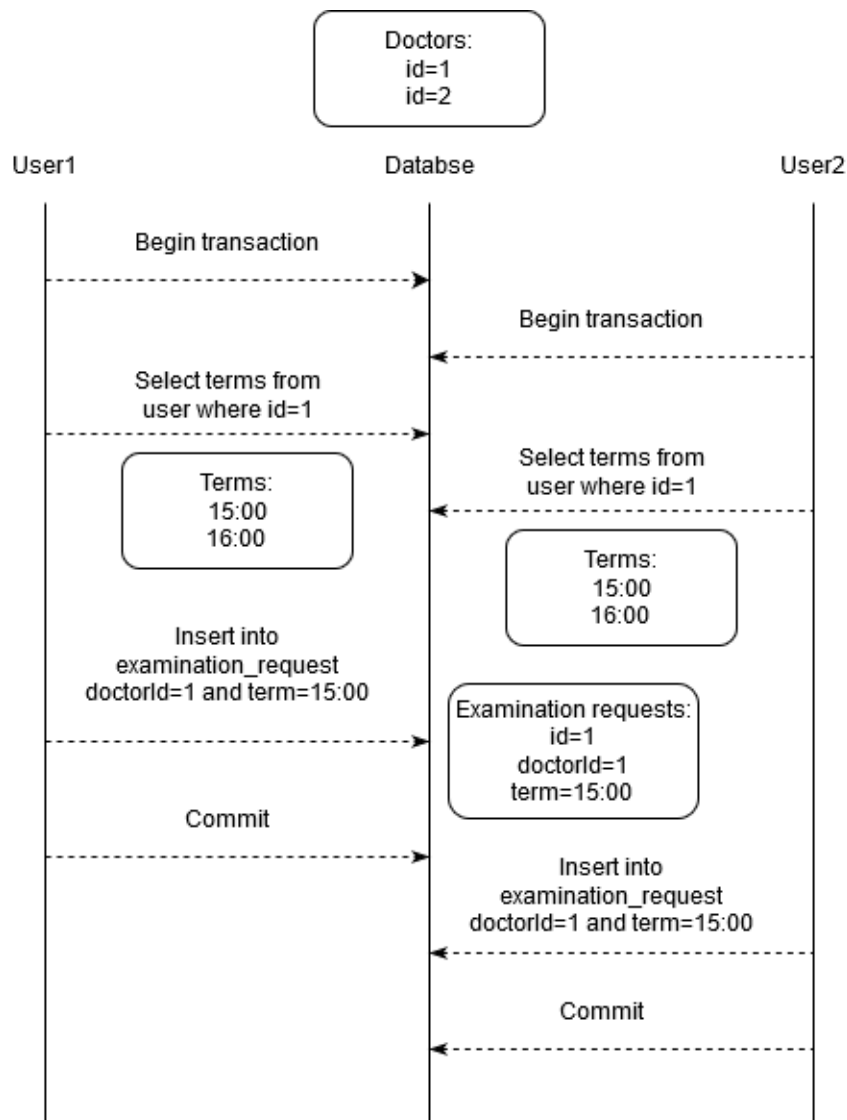


KONKURENTAN PRISTUP RESURSIMA U BAZI

Student 1

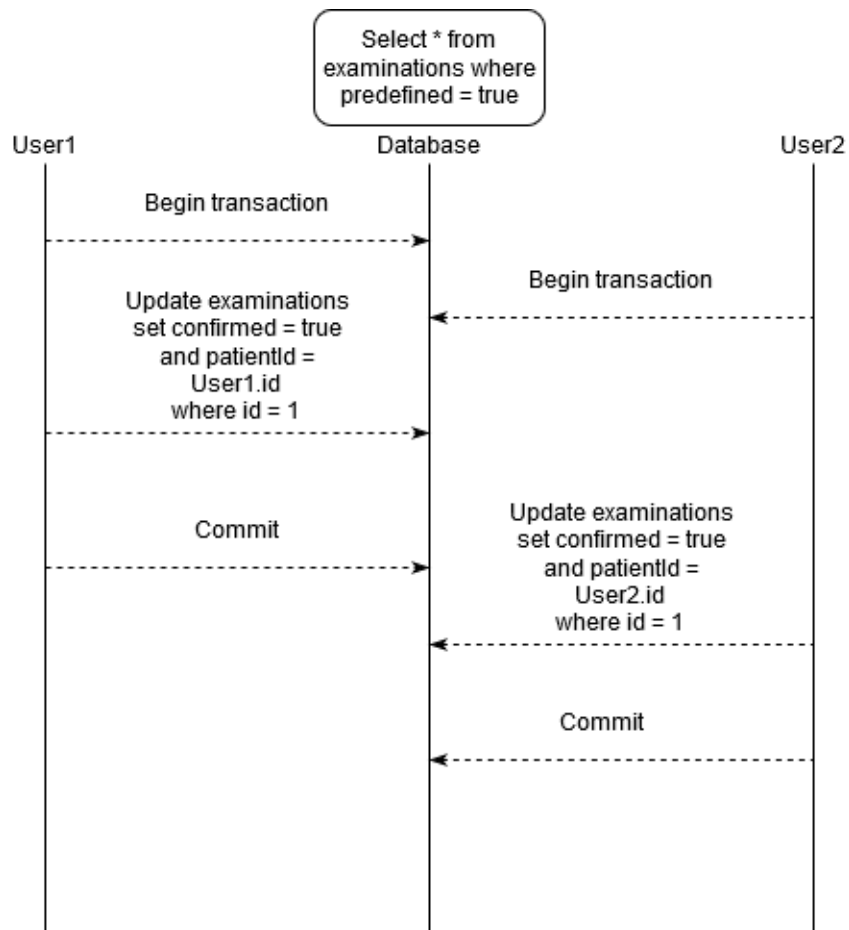
Situacija 1: Više korisnika aplikacija pokušavaju da zatraže upit za pregled u istom terminu kod istog doktora. Doktor ne može da bude prisutan na oba termina i samim tim dolazi do konfliktne situacije. Potrebno je sprečiti ovakvu situaciju tako što će se jedan od korisnika obavestiti da se u međuvremenu stanje promenilo i nije moguće zatražiti upit za konkretani termin.



Navedena situacija je rešena zaključavanjem baze uz pomoć mehanizma pesimističnog zaključavanja. Konkretno je iskorišćen *PESSIMISTIC_READ* koji nam

omogućava da čitamo podatke, ali je nemoguće izvršiti brisanje ili izmenu zaključenih podataka. Ovime smo omogućili da se samo jednom može rezervirati jedan termin kod jednog doktora. Na klijentskoj strani u trenutku rezervisanja, ponovo se dobave svi slobodni termini doktora i ukoliko se selektovani termin ne nalazi više u listi, u međuvremenu je neko zakazao taj termin, čime se obaveštava korisnik.

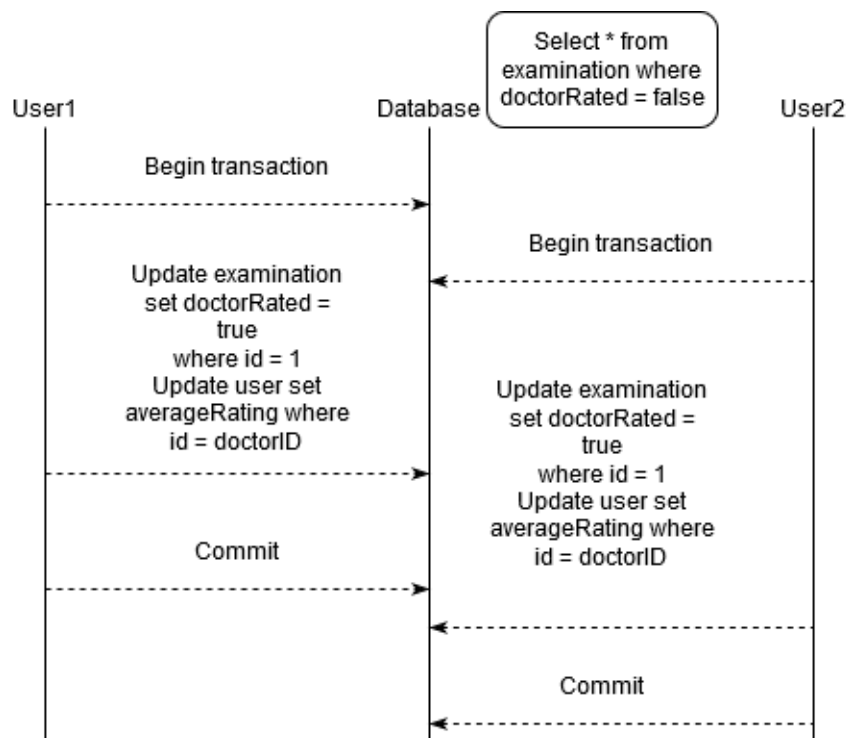
Situacija 2: Unapred definisani pregled ne smeju biti rezervisani od strane više korisnika. Jedan termin može biti rezervisan isključivo pa je moguće doći do konfliktne situacije. Potrebno je sprečiti ovakvu situaciju tako što će se jedan od korisnika obavestiti da se u međuvremenu stanje promenilo i nije moguće zakazati predefinisani pregled.



Navedena situacija je rešena zaključavanjem baze uz pomoć mehanizma pesimističnog zaključavanja. Konkretno je iskorišćen *PESSIMISTIC_READ* koji nam omogućava da čitamo podatke, ali je nemoguće izvršiti brisanje ili izmenu zaključenih podataka. Ovime smo omogućili da se predefinisani pregled može zakazati samo jednom. Na klijentskoj strani u trenutku zakazivanja, ponovo se

dobave svi predefinisani pregledi i ukoliko se pregledne nalazi vise u listi, u medjuvremenu je neko zakazao, čime se obaveštava korisnik.

Dodatna situacija, student1: Jedan doktor ne može biti ocenjen u isto vreme od strane više pacijenata. Do konfliktne situacije dolazi jer će se desiti da ocena jednog korisnika bude „poništena“. U aplikaciji prosečna ocena se računa pri svakom ocenjivanju i ovakva situacija dovodi do prikaza pogrešne prosečne ocene doktora.



Navedena situacija je rešena zaključavanjem baze uz pomoć mehanizma optimističkog zaključavanja. Ovim mehanizmom dodajemo polje *Verzija* u modelu. U slučaju konkurentnog pristupa polje verzije se menja. Provera se vrednost polja verzije i ukoliko je ono promenjeno moramo da uhvatimo *OptimisticLockException* i možemo ponovo da pokušamo odraditi izmene.

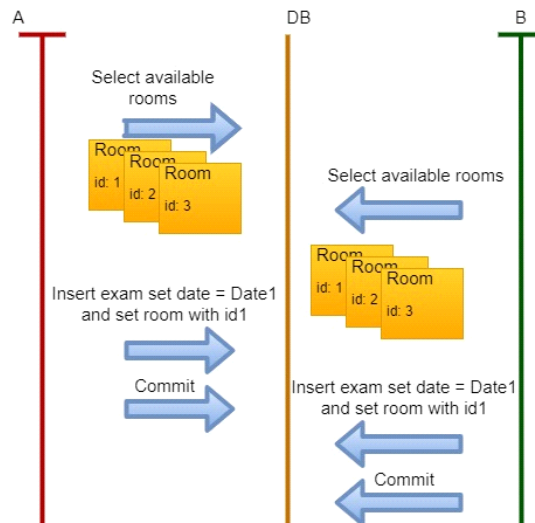
Student 2

Situacija 3:

Prilikom odobravanja zahteva za pregled/operaciju, ne može jedna sala da bude rezervisana u isto vreme za različite preglede/operacije. Prilikom postupka izbora sale za operaciju/pregled, adminu klinike se na raspolaganje stavljaju sve sale slobodne u specificiranom terminu. U slučaju u kom su dva admina pristupila

postupku izbora u kom je transakcija prvog krenula pre nego što se transakcija drugog izvršila, tada na raspolaganju za izbor sobe za svoje preglede/operacije u istom terminu imaju iste sobe. Ako bi kojim slučajem obojica odabrala istu sobu, napravila bi se dva pregleda/operacije u istom terminu sa istim sobama.

Tok:

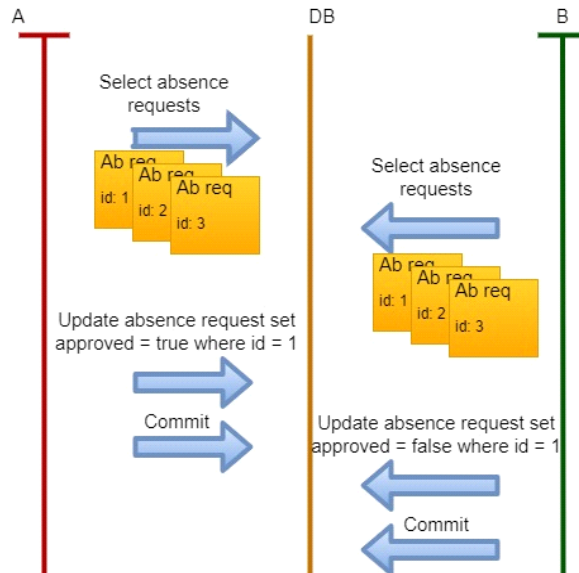


Rešenje:

Zaključati metodu čuvanja pregleda/operacije sa Pessimistic.Read čime se zaključava baza i upis u nju u istom trenutku, a slučaj kada dva ista admina dobiju iste sobe i žele da ih zakažu je rešen tako što su pre svakog pokušaja upisa novog pregleda/operacije sve dostupne sobe ponovo tražene iz baze, pa tek ako je izabrana soba i dalje među novim podacima može se pristupiti upisu novih pregleda/operacija.

Situacija 4: Prilikom odobravanja/odbijanja zahteva za godišnji odmor ili odsustvo, ne može jedan zahtev da bude odobren/odbijen od strane različitih administratora. Prilikom postupka odbijanja/prihvatanja adminima se prikazuju svi zahtevi iz te klinike. U slučaju kada dva admina imaju iste zaheve koje mogu da odbijaju/prihvataju u slučaju kada bi jedan prihvatio a drugi odbio, drugi admin bi pregazio akciju prvog.

Tok:



Rešenje:

Zaključati metodu čuvanja(u ovom slučaju ažuriranja) sa Pessimistic.Read čime se zaključava baza za upis/izmenu u istom trenutku, a slučaj kada dva ista admina dobiju na raspolaganje iste zahteve i pokušaju da ih odbiju/prihvate je rešen tako što se pre odbijanja/prihvatanja ponovo dobave svi zahtevi, pa samo ako je izabrani zahtev i dalje među neobrađenim može se pristupiti obradi.

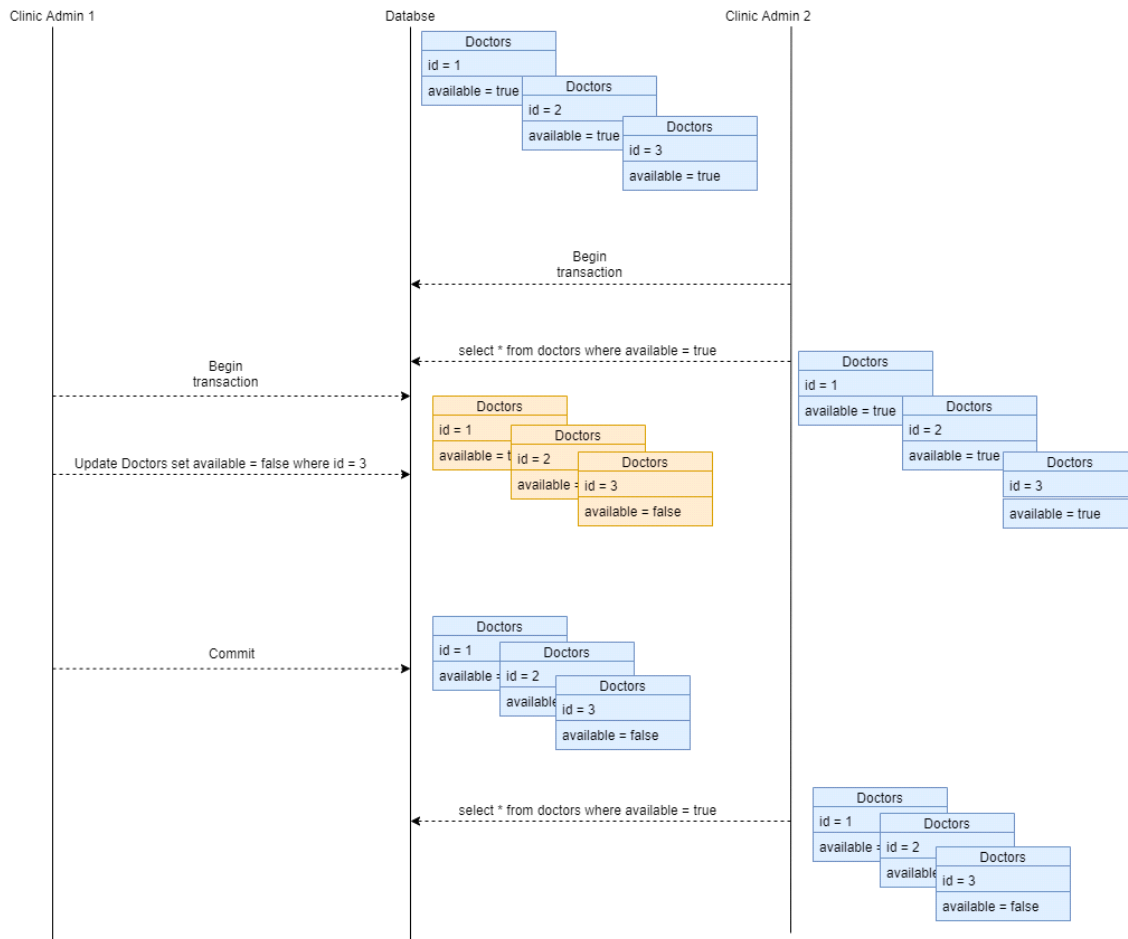
Student 3

Situacija: **Admin klinike bira dodatne doktore za operaciju.**

U ovoj situaciju problem nastaje kada Clinic Admin 2 zatraži listu svih slobodnih doktora za neku operaciju. Nakon izvršavanja upita on dobija tri doktora koji ispunjavaju zadati kriterijum (available = true). U međjuvremenu transakcija koju je pokrenuo Clinic Admin 1 u bazi radi update odnosno neki doktor više nije slobodan tj Clinic Admin 1 za doktora sa id-jem 3 setuje polje available na false i komituju se izmene. Nakon toga, kada Clinic Admin 2 izvrši identičan upit koji je izvršio i prvi put dobiće različite podatke za isti red u tabeli. Ovaj problem je poznat pod nazivom "Unrepeatable read" koji je rešen pomoću pesimističkog zaključavanja sa parametrom Pessimistic.READ koji rešava i gore navedeni problem pod nazivom "Dirty read".

Pessimistic.READ zapravo obezbeđuje da podaci ne budu izmenjeni između dva uzastopna čitanja istih, ali pritom ova vrsta zaključavanja ne blokira ostale

transakcije da čitaju podatke.



Situacija **overavanje recepta**:

Sestra Milica (Nurse 1) je pristupila stranici za overavanje recepata i učitala sve neoverene recepte. Želi da overi recept sa rednim brojem 5, koji je na početku transakcije imao vrednost validated = false. U bazi polje validated postaje true, ali izmene u transakciji još nisu komitovane. Zatim, sestra Ivana (Nurse 2) želi da overi neki recept, pristupa stranici na kojoj se prikazuju neovereni recepti. U tom mometu recept sa id-jem 5 ima vrednost validate = true i sestra Ivana ga neće imati prikazanog na svojoj stranici. U medjuvremenu transakcija koju je pokrenula sestra Milica postaje rollback-ovana. Ovde se stvara problem poznatiji pod nazivom "Dirty read", jer sestra Ivana dobija podatke koji zapravo ne postoje u

bazi, jer u momentu kada je sestra Ivana zatražila neoverene recepte, recept sa id-jem 5 je za polje validated imao vrednost true. Ovaj problem je rešen pomoću pesimističkog zaključavanja i to sa parametrom Pessimistic.READ

