

Konkurentni pristup resursima u bazi

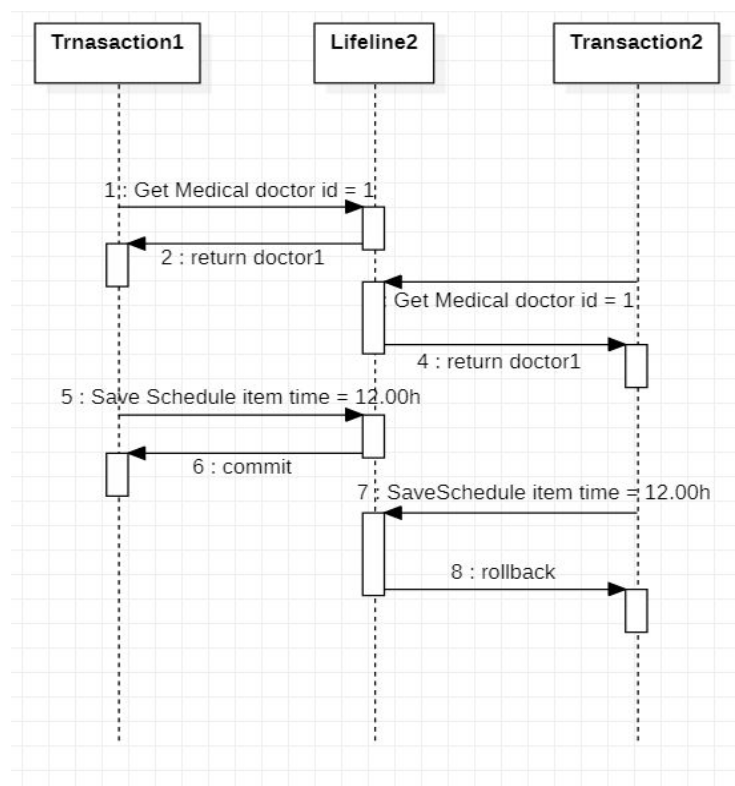
Bodroža Jovan,
SW 44/2017,
bodroza.joca@gmail.com

1. Jedan lekar ne može istovremeno da bude prisutan na više različitih operacija

A. Opis konfliktne situacije

Konfliktnost ove funkcionalnosti izražava se prilikom rezervisanja sala za operaciju. Naime, prilikom odabira sale za operaciju potrebno je izabrati doktore koji su obavezni da prisustvuju pomenutoj operaciji. Funkcionalnost odabira sale izvršava admin određene klinike. Kako je naša aplikacija namenjena za istovremenu upotrebu od strane više korisnika, moguć je scenario u kom više klinik admina istovremeno zahtevaju da zauzmu istog doktora za različite operacije. Naravno, u trenutku odabira pomenuti doktor je slobodan. Međutim ukoliko jedan od pomenutih zahteva uspe da zauzme posmatranog doktora, ostali zahtevi koji žele da ga zauzmu za isti vremenski period bivaju odbačeni.

B. Crteže tokova zahteva klijenta i odgovora servera



C. Opis načina na koji su rešavane uočene situacije

Rešenje analizirane konfliktne situacije u našem slučaju oslanja se na samu arhitekturu rešenja. Naime, radni kalendar doktora se formira tako što se kreiraju stavke radnog kalendara (*schedule items*) koje sadrže *doctor_id* i vreme kada se

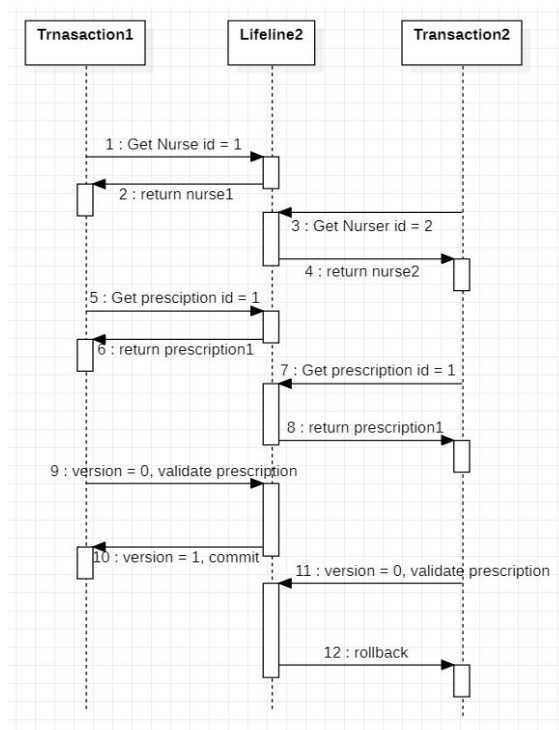
operacija izvršava (*date_time*). Ukoliko kombinaciju ovih atributa posmatramo kao jedinstvenu celinu (*unique constraint*) uočavamo da tada neće biti moguće da baza podataka za istog doktora sadrži stavke kalendara sa identičnim vremenima. Ukoliko posmatramo iznad priloženu sliku možemo videti da *Transaction1* i *Transaction2* pokušavaju da kreiraju stavku kalendara sa identičnim podacima (*doctor_id* = 1, *date_time* = DD.MM.YYYY 12:00h). Kako *Transaction1* prva uspeva da kreira stavku kalendara za određenu operaciju, ona će biti i odobrena. *Transaction2* neće uspeti da upiše u kalendar stavku sa identičnim podacima tako da će rezultati njenog rada biti opozvani (*rollback*).

2. Više medicinskih sestara istovremeno pokušava da overi isti recept

A. Opis konfliktne situacije

Konfliktnost ove funkcionalnosti izražava se prilikom overavanja recepata za određenu kliniku. Kako je naša aplikacija namenjena za istovremenu upotrebu od strane više korisnika, moguć je scenario u kom više medicinskih sestara istovremeno pokušava da overi isti recept, međutim, recept poseduje informaciju o isključivo jednoj medicinskoj sestri koja ga je overila. Na osnovu pomenutih činjenica uočavamo da je u ovakvoj, konfliktnoj situaciji neophodno doneti odluku o tome koja sestra će overiti posmatrani recept.

B. Crteže tokova svih zahteva klijenta i odgovora server



C. Opis načina na koji su rešavane uočene situacije

Rešenje analizirane konfliktne situacije u našem slučaju oslanja se na upotrebu optimističkog zaključavanja. Metoda optimističkog zaključavanja se realizuje tako što se odgovarajućem entitetu dodaje polje (*version*) koje označava verziju posmatranog entiteta (reda u tabeli relacione baze podataka). Ukoliko verzija određene transakcije (verzija u radnoj memoriji) odgovara verziji entiteta u tabeli, tada se toj transakciji odobrava izmena posmatranog polja. Na osnovu iznad priložene slike možemo videti da *Transaction1* uspeva da izvrši operaciju izmene nad entitetom na osnovu čega se povećava vrednost polja *version*. Nakon toga *Transaction2* pokušava da izvrši izmenu polja istog entiteta ali ne uspeva zbog toga što je verzija koja se nalazi u radnoj memoriji transakcije različita od verzije entiteta. U našem slučaju analizirani entitet jeste *Prescription* klasa kojoj je potrebno postaviti polje *medical_nurse*;