

## Problem: jedan farmaceut ne može istovremeno da bude prisutan na više različitih savetovanja

Po definiciji aplikacije koju pravimo, farmaceut ne sme da ima zakazana dva savetovanja u isto vreme. Vodeći se tom činjenicom, odabrali smo pristup po kome farmaceut može da započne savetovanje (*appointment*) samo za ona savetovanja koja se odvijaju u tom zadatom trenutku. Zahvaljujući ove dve činjenice, farmaceut može da obavlja samo jedno savetovanje u zadatom trenutku, i to jedino ukoliko je za to savetovanje definisano da počinje pre sadašnjeg trenutka i traje posle sadašnjeg trenutka.

Prilikom dodavanja bilo kog novog savetovanja, osim provere da li to savetovanje upada u radno vreme farmaceuta kod koga se odvija (počinje nakon početka njegovog radnog vremena i završava se pre kraja radnog vremena), vršimo proveru da li se taj termin poklapa sa bilo kojim drugim savetovanjem koje farmaceut već ima, bilo kojim drugim pregledom koji pacijent već ima ili bilo kojim zahtevom za odmor koji farmaceut ima.

Ukoliko bismo smatrali da je stanje u bazu uvek apsolutno tačno, ove provere bi nam garantovale da se dva termina ne mogu poklopiti, ali moramo obratiti pažnju i na transakcije koje se odvijaju prilikom dodavanja novog termina savetovanja (*appointment*).

Pošto sam vršio implementaciju zakazivanja kontrolnog pregleda od strane farmaceuta, predstavimo problem koji se tu može dogoditi. Ukoliko u trenutku kada farmaceut izabere termin za zakazivanje novog pregleda za trenutnog pacijenta nije postojao ni jedan drugi termin sa kojim bi se izabrani „preklapao“, ali za vreme popunjavanja termina drugi pacijent izabere isti termin i uspe da se upiše u bazu, dolazimo do nekonzistencije sa pravilom o jedinstvenom trenutnom terminu.

Ovom problemu smo mogli da pristupimo iz ugla pesimističke konkurentnosti baze, ali zbog konvencije o nekorišćenju tog pristupa uz alat Entity Framework za ASP.NET, odlučili smo se za sledeću proveru: Svaki put kada u servisu pozivamo funkciju *Create(appointment)*, osim standardne provere sa svim vrednostima iz baze, izvršićemo proveru i sa svim vrednostima novih ili izmenjenih objekata tipa

Appointment za koje su napravljene transakcije na čekanju, ali još uvek nisu poslati na bazu (Entity: *\_context.tbAppointments.Local*). Na ovaj način sprečavamo mogućnost da se 2 pregleda zakažu u isto vreme, da se njihovi termini preklapaju i da oba budu prihvaćena, jer će jedan od njih sigurno dobiti raniju transakciju te će naredni, ukoliko prvi zakazani pregled ne uspe da se ubaci u bazu, svakako biti odbijen zbog postojanja prvog pregleda u transakciji unutar Entity frejmworka.

Pacijent ili lekar, u zavisnosti od toga čiji je zahtev stigao malo kasnije, dobiće obaveštenje da njihov termin nije moguće zakazati (iako su na FRONTEND strani prošli provere saglasnosti sa drugim terminima), te će biti naterani da ponovo pošalju izmenjen zahtev za zakazivanje pregleda.

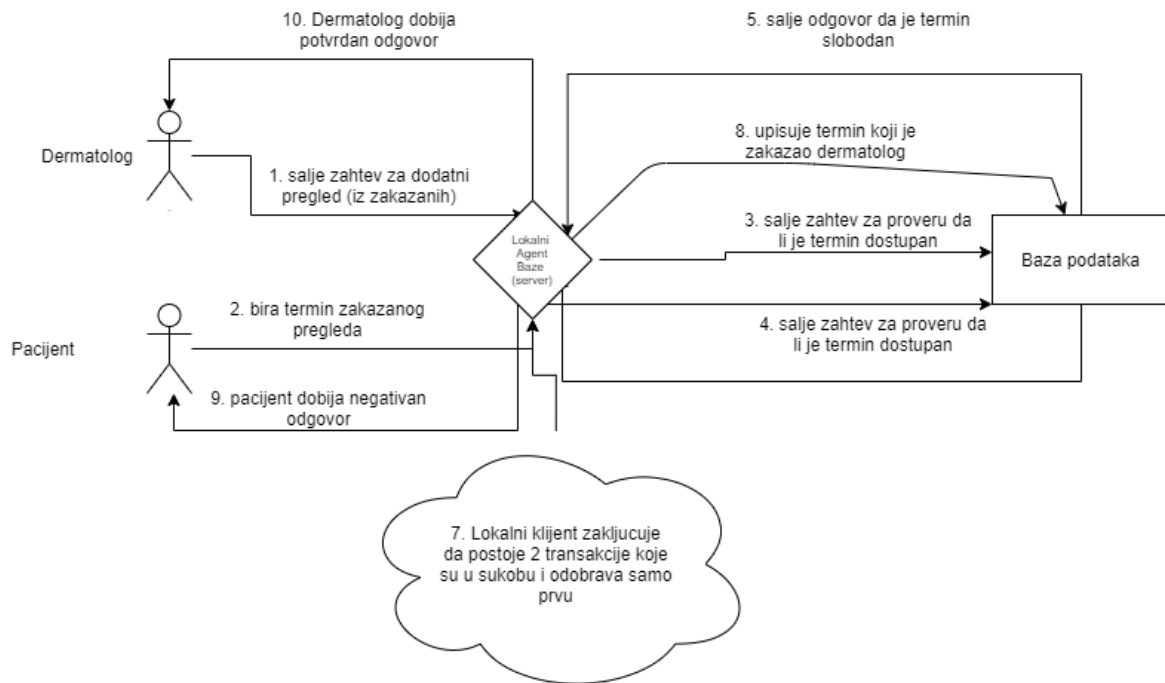
**Problem: jedan dermatolog ne može istovremeno da bude prisutan na više različitih pregleda**

Rešenje ovog problema konkurencije se većinski može svesti na rešenje prethodnog problema, uz jednu izmenu: dermatolog, osim slobodnog izbora za dodatni termin pregleda za pacijenta koga pregleda, može da izabere i jedan od unapred definisanih termina za njegove preglede – tako da osim funkcije *Create* dermatolog može da poziva i funkciju *Update*.

Osim problema sa preklapanjem termina (koji će biti rešen na isti način kao i prethodni primer), ovde dolazimo do novog problema sa konkurentnim pristupom bazi, a to je da se unapred definisan termin koji dermatolog bira da u njemu zakaže pregled izmeni od momenta kada ga dermatolog bira do momenta kada se on upisuje u bazu (na primer, upiše se neki pacijent ili administrator izmeni neke podatke). U tom slučaju bi se dogodilo da se zakazani termin nekog pacijenta koji je sistemu pristupio samostalno „pregazi“ izmenama dermatologa, što ne sme da se dogodi.

Kako rešavamo ovaj problem? Ovde ćemo pribeći sistemu optimistične konkurencije, te ćemo svaki put prilikom operacije *Update* proveravati i verziju u bazi reda koji želimo da izmenimo. Ukoliko se dogodi da se od trenutka kada smo mi taj red preuzeli i krenuli da menjamo verzija tog reda izmenila, naša promena neće biti dozvoljena i dermatolog će biti obavešten da se dogodila promena stanja

u bazi podataka, te da njegov zakazan termin neće moći da se prihvati. Dermatolog tada mora ponovo da pošalje izmenjen zahtev za rezervaciju termina.



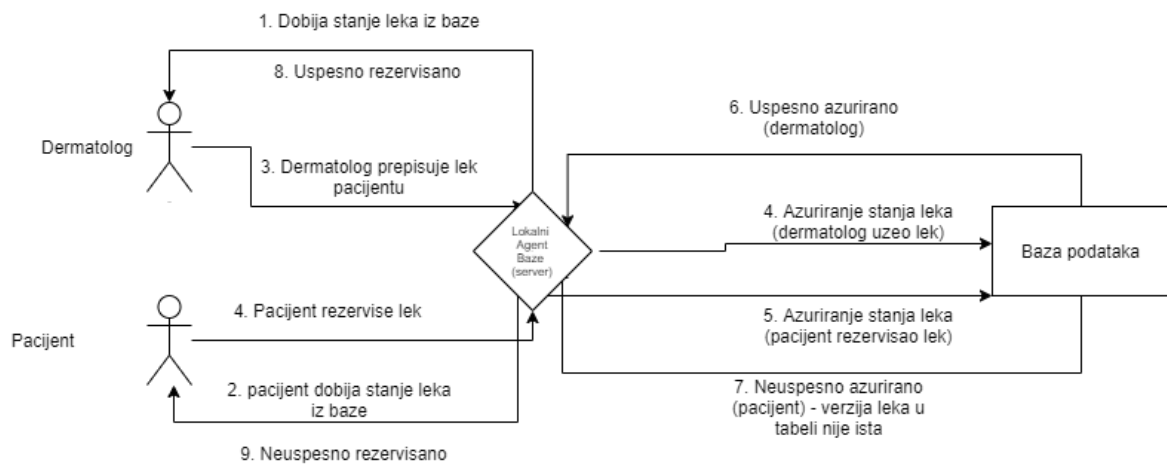
Slika 1: Dijagram komunikacije prilikom problema zakazivanja termina koji se poklapaju. Proces je isti i za slučaj kada farmaceut zakazuje dodatni termin savetovanja

Problem: lekar (dermatolog ili farmaceut) prilikom pregleda preporučuje i izdaje lek pacijentu

Problem sa operacijom *Update* evidentan je i na još jednom mestu. Prilikom pregleda, dermatolog ili farmaceut imaju mogućnost da pacijentu prepisu lek ili medicinsko sredstvo, te da mu ga odmah i izdaju, ukoliko je lek dostupan. Ovde može doći do više problema prilikom konkurentnog korišćenja baze, a jedan od njih je nestanak ili rezervacija baš tog leka prilikom čuvanja novog stanja leka u apoteci, nakon izdavanja leka pacijentu.

Ukoliko je lekar izabrao lek koji želi i koji je dostupan, a tog leka na stanju ima samo 1, ako se baš u trenutku između provere da li je lek dostupan i operacije *Update* (čuvanja nove količine leka u bazi) dogodi da neki drugi korisnik (drugi farmaceut ili dermatolog, ili čak pacijent prilikom rezervacije) uzme ili rezerviše, bez

konkurentne strategije došli bismo do nekonzistentnosti baze. Ovde bismo mogli da primenimo pesimističku konkurenciju, ali smo izabrali optimistički pristup, kako bismo izbegli zaključavanje baze i druge probleme koji se mogu dogoditi primenom pesimističkog pristupa. Naše rešenje je da ovom prilikom koristimo optimističku konkurenciju, te da prilikom operaciju *Update* proverimo verziju reda u tabeli koju želimo da izmenimo sa onom koju smo bili uvideli. Na ovaj način, ne može doći do izdavanja viška lekova, a lek će biti izdat onome ko prvi do njega dođe, dok će ostali korisnici biti odbijeni (naš lekar koji treba da izda lek).



Slika 2: Dijagram komunikacije prilikom problema uzimanja leka i rešenja optimističnim pristupom