

Rešavanje problem konkurentnog pristupa resursima u bazi podataka

Problem kreiranja nove rezervacije od strane vlasnika vikendice/broda i instruktora

Problem konkurentnog pristupa resursima u bazi podataka čije rešavanje će biti prikazano u nastavku je problem kreiranja nove rezervacije od strane vlasnika vikendice/broda i instruktora u isto vreme kada klijent pravi rezervaciju za isti entitet. Prilikom dodavanja nove rezervacije postoji provera da li postoji rezervacija ili kreiranja specijalna akcija u odabranom vremenskom periodu za odabrani entitet. Ukoliko bismo smatrali da je stanje u bazi podataka uvek tačno ova provera bi nam garantovala da ne mogu postojati dve rezervacije za isti entitet čiji se period preklapa.

Pošto sam vršila implementaciju kreiranja nove rezervacije od strane vlasnika vikendice/broda i instruktora koja funkcioniše tako što vlasnik može da kreira novu rezervaciju za neki od svojih entiteta za goste koji trenutno borave u nekom od njegovih objekata, problem koji se ovde može javiti jeste taj što neki drugi klijent može da kreira rezervaciju za isti entitet u preklapajućem vremenskom periodu što bi dovelo do nekonzistencije sa pravilom o jedinstvenosti termina rezervacije entiteta.

U 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, stoga smo odlučili da problemu pristupimo na sledeći način : svaki put kad pozivamo funkciju *CreateView* u kontroleru gde osim standardne provere da li se odabrani termin rezervacije preklapa sa terminom neke od postojećih rezervacija, zatim i provera da li postoji neka specijalna akcija čiji se period realizacije preklapa sa odabranim terminom nove rezervacije, a dodata je provera za objekte tipa *CottageReservation/BoatReservation/AvantureReservation* za koje su napravljene transakcije koje su na čekanju ali još uvek nisu poslate na bazu (Entity : `_context.CottageReservation.Local` / `_context.BoatReservation.Local` / `_context.AdventureReservation.Local`). Na ovaj način sprečavamo da se kreiraju dve rezervacije u isto vreme čiji se period preklapa i da se obe rezervacije budu

prihvaćene, jer će jedna od rezervacija sigurno dobiti raniju transakciju, te će naredna rezervacija, ukoliko prva rezervacija ne uspe da se ubaci u bazu, svakako biti odbijena zbog postojanja prvog pregleda u transakciji unutar Entity Framework-a.

Klijent ili vlasnik vikendice/broda kao i instruktor, u zavisnosti od toga čiji je zahtev stigao kasnije biće obavešteni da njihova rezervacija nije prihvaćena, te će biti naterani da ponovo kreiraju rezervaciju sa promenjenim datumima.

Problem kreiranja nove akcije od strane vlasnika broda/vikendice i instruktora

Problem konkurentnog pristupa resursima u bazi podataka čije rešavanje će biti prikazano u nastavku je problem kreiranja akcije za brzu rezervaciju kada neki klijent kreira rezervaciju za isti entitet. Prilikom dodavanja nove akcije proverava se da li postoji neka rezervacija za odabrani entitet čiji se termin poklapa sa terminom za akciju, pored ove standardne provere kako bismo rešili gore pomenut problem potrebno je da prilikom kreiranja akcije proverimo da li je u toku kreiranje rezervacije od strane klijenta za isti entitet i da onemogućimo kreiranje akcije ukoliko se desio pomenuti slučaj.

Postupak kreiranja akcije je implementiran na sledeći način : vlasnik vikendice/broda ili instruktor bira entitet za koji želi da kreira akciju, i zatim popunjava formu gde unosi period važenja akcije, period realizacije akcije, i ostale elemente forme i zatim potvrđuje kreiranje akcije, problem koji se ovde javlja jeste taj što klijent može kreirati novu rezervaciju za isti entitet istovremeno. Klijente treba da pustimo da kreiraju rezervaciju pošto se veći profit ostvaruje prilikom standardne rezervacije nego prilikom akcije stoga ćemo zabraniti vlasnicima i instruktorima da kreiraju akciju ukoliko se istovremeno kreira nova rezervacija za isti entitet.

Svaki put kad pozivamo funkciju *CreateSpecialOffer* u kontroleru radi se provera da li postoji neka rezervacija za izabrani entitet čiji period se preklapa sa specijalnom akcijom, zatim provera da li postoji akcija čiji se period preklapa sa specijalnom akcijom koju vlasnik želi da kreira, i na kraju se nalazi provera da li postoji objekat tipa `CottageReservation/BoatReservation/AdventureReservation` za koje su napravljene transakcije koje su na čekanju ali još uvek nisu poslate na bazu (Entity : `_context.CottageReservation.Local` / `_context.BoatReservation.Local` /

_context.AdventureReservation.Local). Ukoliko postoji rezervacija koja je na čekanju za isti entitet za koji vlasnik ili instruktor želi da napravi specijalnu akciju tada puštamo klijenta da ostvari svoju rezervaciju dok vlasnik ili instruktor bivaju obavešten da nije uspelo kreiranje specijalne akcije.