

Konkurentni pristup resursima u bazi

Milica Popović SW11/2019

Student 2

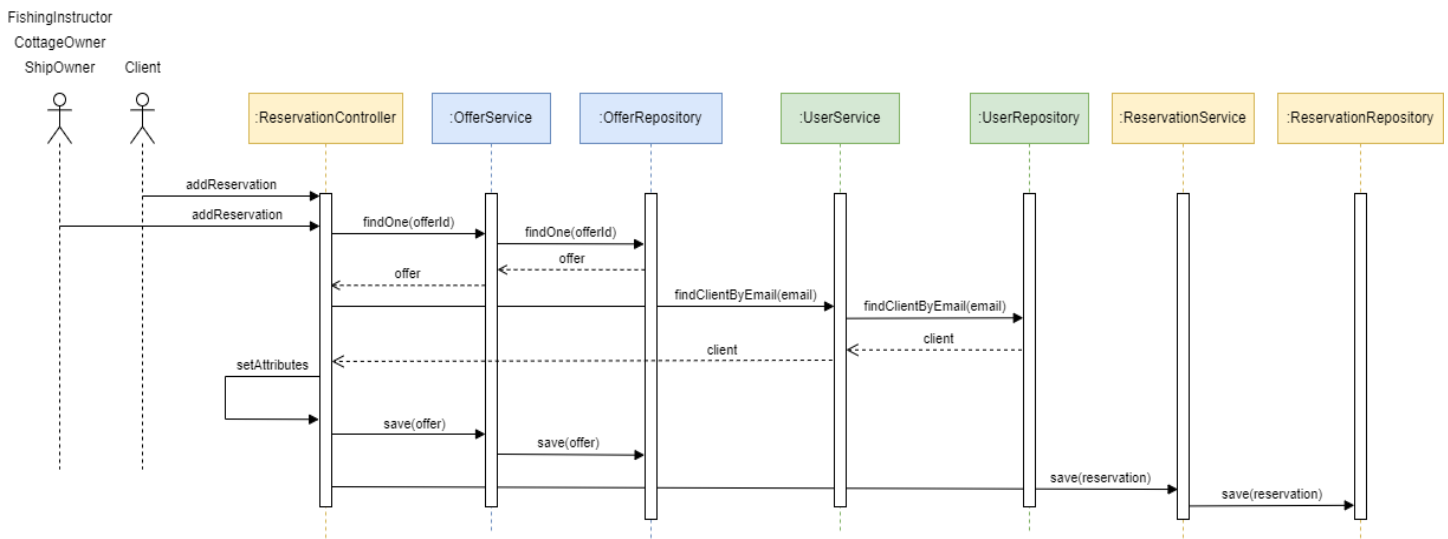
Konfliktna situacija 1

Opis problema

Potrebno je rešiti konfliktnu situaciju u okviru koje vlasnik vikendice/broda ili instruktor pravi rezervaciju u isto vreme kad i klijent. Ova situacija je problematična, jer može dovesti do toga da se u istom terminu za istu ponudu kreira više rezervacija, što nije dozvoljeno.

Model situacije

Vlasnik (vlasnik vikendice, broda ili instruktor pecanja) i klijent istovremeno iniciraju kreiranje rezervacije. Tada se poziva metoda *addReservation* iz klase *ReservationController*, te se iz baze dobavljaju ponuda i klijent za koje se rezervacija pravi. Nakon postavljanja svih potrebnih atributa novokreirane rezervacije, ona se čuva u bazu. Problem je u tome što se sada može desiti da su za ponudu kreirane dve rezervacije u istom terminu.



Rešenje

Navedeni problem je rešen upotrebom optimističkog zaključavanja ponude (vikendice, broda ili avanture), tako što je u klasi *Offer* dodat atribut *version*, koji predstavlja verziju objekta. Klasi *Offer* je takođe dodata promenljiva koja predstavlja brojač rezervacija. Kada se nova rezervacija doda, brojač se inkrementira i promena se sačuva u bazu, što dovodi do povećanja vrednosti atributa *version*. Zahvaljujući tome, u slučaju da klijent i vlasnik pokušaju napraviti rezervaciju u isto vreme, samo će se jedna sačuvati. Prilikom pokušaja druge izmene doći će do *OptimisticLockingFailureException*-a. Nije izabrano pesimističko zaključavanje, jer bi ono više narušilo performanse aplikacije.

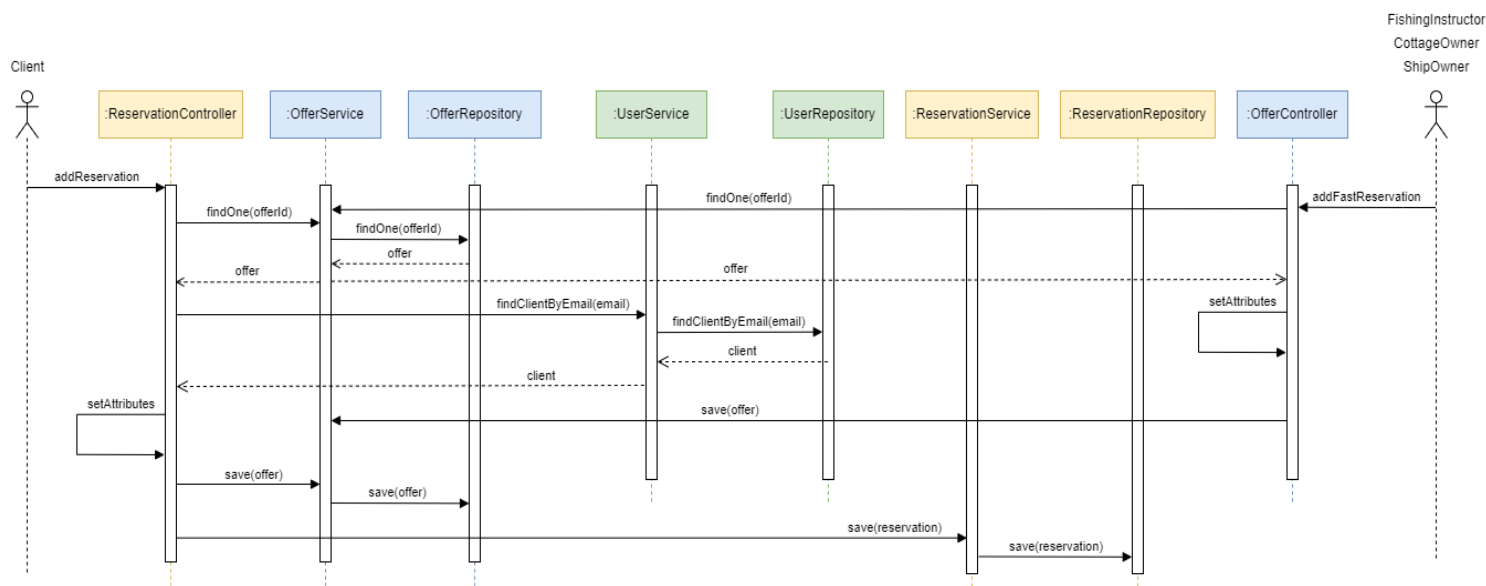
Konfliktna situacija 2

Opis problema

Potrebno je rešiti konfliktnu situaciju u okviru koje vlasnik vikendice/broda ili instruktor pecanja pravi akciju za brzu rezervaciju u isto vreme kad i klijent rezerviše ponudu. Ova situacija je problematična, jer može dovesti do toga da klijent rezerviše ponudu u terminu kada treba da važi akcija, ali ju je rezervisao po regularnoj ceni.

Model situacije

Vlasnik (vlasnik vikendice, broda ili instruktor pecanja) inicira kreiranje nove akcije za brzu rezervaciju u isto vreme kada klijent inicira kreiranje rezervacije (za istu ponudu). Prilikom kreiranja akcije za brzu rezervaciju, prvo se poziva metoda *addFastReservation* klase *OfferService*. Zatim se iz baze dobavlja ponuda, te se kreira nova akcija i ona se dodaje u kolekciju akcija za datu ponudu. Na kraju se izmenjena ponuda čuva u bazi. Dodavanje rezervacije klijenta se obavlja na isti način kao u prethodnoj situaciji. Ako se poklapaju termini klijentove rezervacija i novododane akcije, klijent neće videti tu izmenu i njegova rezervacija će biti po regularnoj ceni, iako bi trebalo da bude po akcijskoj.



Rešenje

Problem je rešen upotrebom optimističkog zaključavanja ponude (vikendice, broda ili avanture), tako što je u klasi *Offer* dodat atribut *version*, koji predstavlja verziju objekta. Klasi *Offer* je takođe dodata promenljiva koja predstavlja brojač rezervacija, kao i promenljiva koja predstavlja brojač akcija. U slučaju dodavanja nove rezervacije, brojač se inkrementira i promena se sačuva u bazu, što dovodi do povećanja vrednosti atributa *version*. Analogno tome, brojač akcija se inkrementira pri dodavanju nove akcije za brzu rezervaciju. Zahvaljujući tome, u slučaju da klijent pokuša napraviti rezervaciju u isto vreme kad vlasnik pokuša dodati novu akciju, samo jedna od tih promena će se sačuvati. Prilikom pokušaja druge izmene doći će do *OptimisticLockingFailureException*-a. Nije izabrano pesimističko zaključavanje, jer bi ono više narušilo performanse aplikacije.

Konfliktna situacija 3

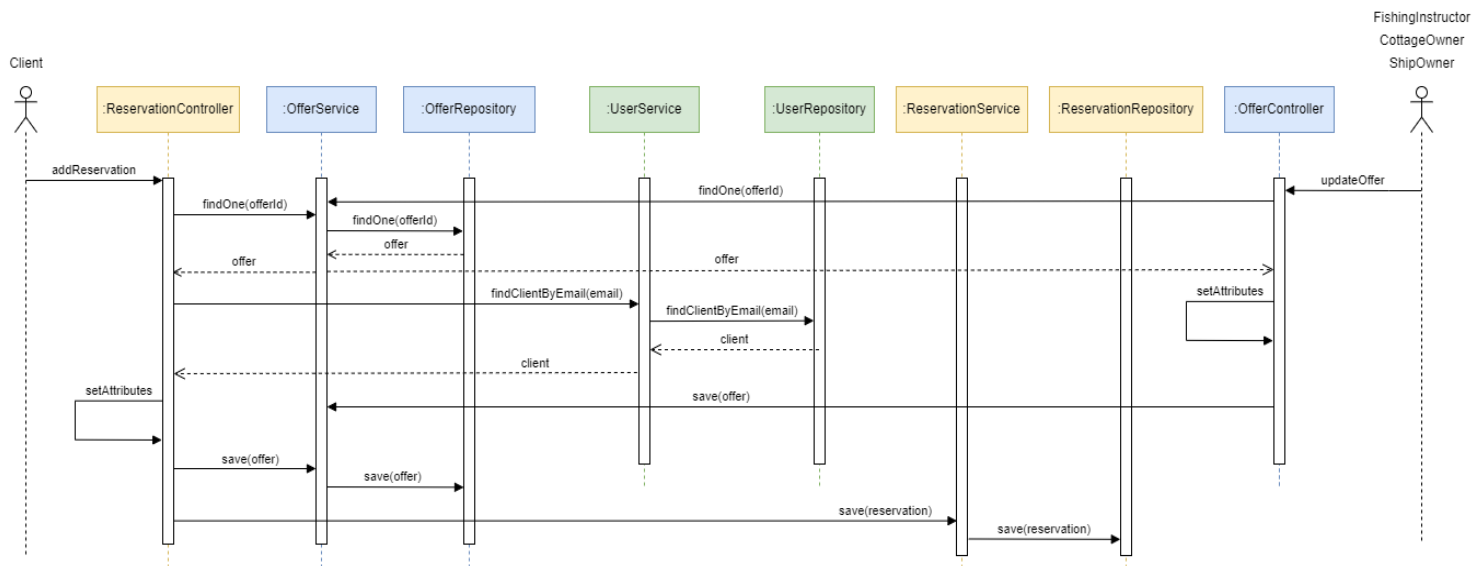
Opis problema

Potrebno je rešiti konfliktnu situaciju u okviru koje vlasnik vikendice/broda ili instruktor pecanja menja svoju ponudu u isto vreme kad i klijent rezerviše tu istu ponudu. Ova situacija je problematična, jer može dovesti do toga da klijent rezerviše ponudu u terminu kada ona više nije dostupna ili ima drukčiju cenu od

one koju će on da plati. Takođe, ponuda u tom slučaju neće imati sva svojstva za koja klijent smatra da ih ima. S druge strane, kada klijent obavi rezervaciju, vlasnik nema pravo da menja ponudu.

Model situacije

Vlasnik (vlasnik vikendice, broda ili instruktor pecanja) inicira izmenu ponude u isto vreme kada klijent inicira kreiranje rezervacije za istu ponudu. Prilikom izmene ponude, prvo se poziva metoda *updateOffer* klase *OfferService*. Zatim se iz baze dobavlja ponuda, te se postavljaju njena izmenjena svojstva. Na kraju se izmenjena ponuda čuva u bazi. Dodavanje rezervacije klijenta se obavlja na isti način kao u prethodne dve situacije. Na ovaj način je došlo do toga da je klijent rezervisao ponudu sa nekim svojstvima koja više ne važe. U najgorem slučaju, može se desiti da ponuda nije više dostupna u periodu u kom je rezervisana ili da će je klijent platiti po ceni koja više nije važeća.



Rešenje

Problem je rešen upotrebom optimističkog zaključavanja ponude (vikendice, broda ili avanture), tako što je u klasi *Offer* dodat atribut *version*, koji predstavlja verziju objekta. Klasi *Offer* je takođe dodata promenljiva koja predstavlja brojač rezervacija. U slučaju dodavanja nove rezervacije, brojač se inkrementira i promena se sačuva u bazu, što dovodi do povećanja vrednosti atributa *version*. To će se desiti i prilikom izmene bilo kog atributa ponude od strane vlasnika. Zahvaljujući tome, u

slučaju da klijent pokuša napraviti rezervaciju u isto vreme kad vlasnik pokuša izmeniti ponudu, samo jedna od tih promena će se sačuvati. Prilikom pokušaja druge izmene doći će do *OptimisticLockingFailureException*-a. Nije izabrano pesimističko zaključavanje, jer bi ono više narušilo performanse aplikacije.