

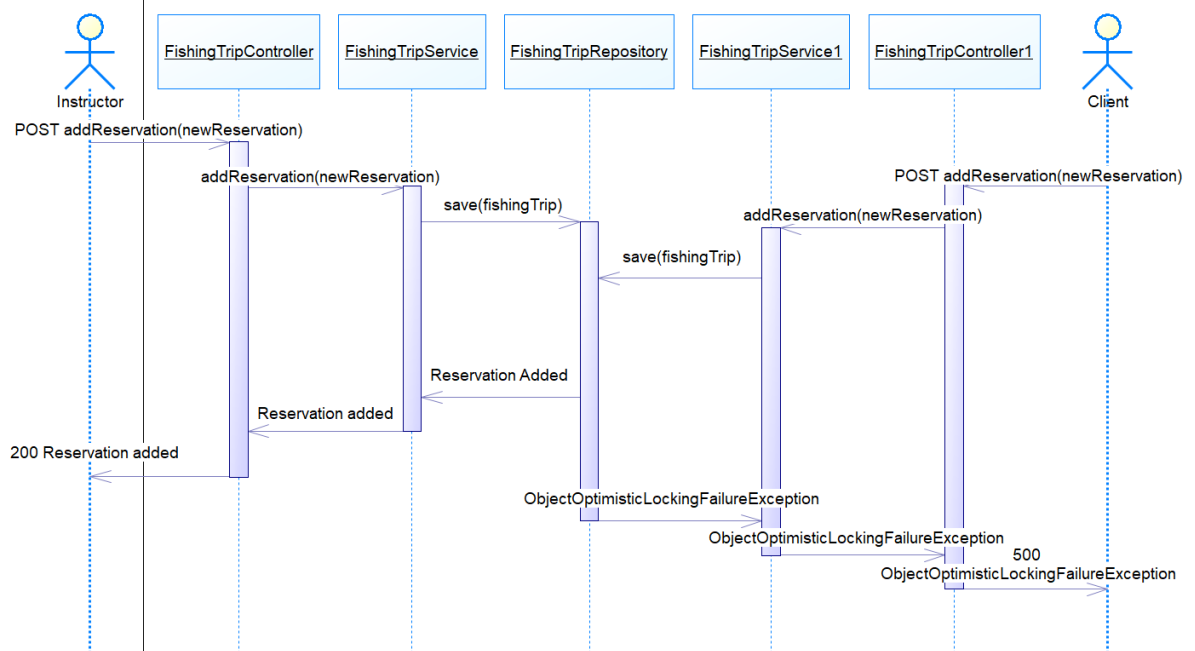
Student 3

Konkurentni pristup resursima u bazi

1. Instruktor ne može da napravi rezervaciju u isto vrijeme kad i drugi klijent

Problem: Instruktor može da zakaže rezervaciju klijentu na svoju uslugu, ali isto tako, i klijent može da zakaže rezervaciju na uslugu instruktora. Problem nastaje kad i instruktor, i klijent pokušaju da rezervišu istu uslugu u isto vrijeme, gdje bi se moglo dogoditi da se obje rezervacije upišu u bazu.

Tok zahtjeva do konfliktne situacije:

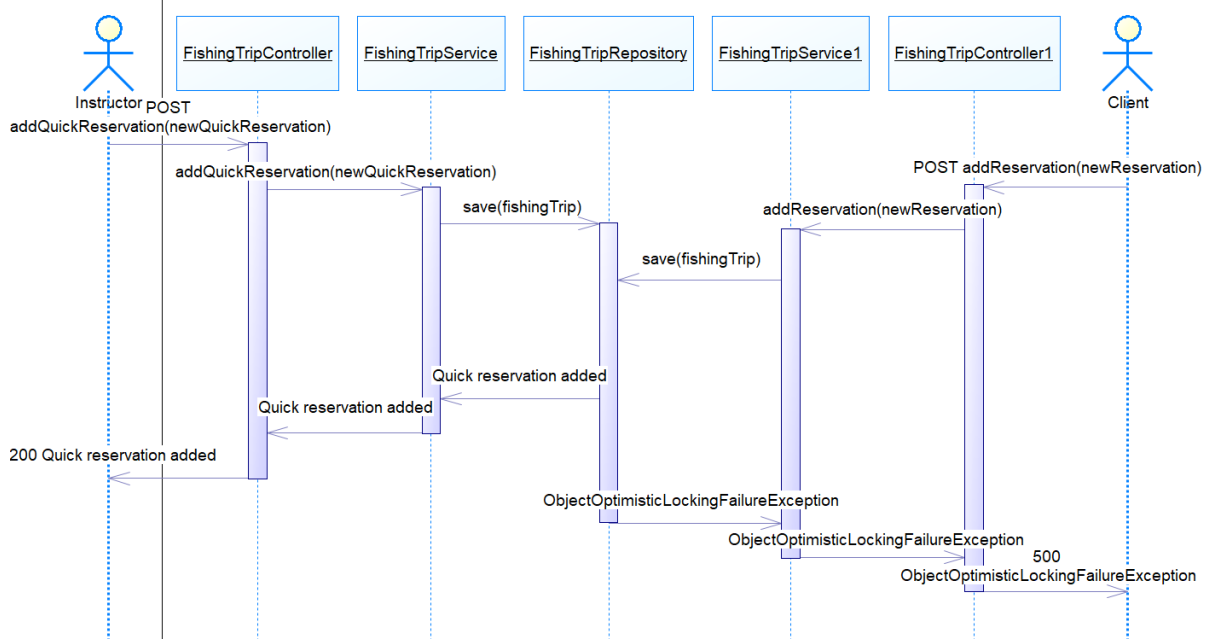


Rješenje: Konfliktna situacija riješena je upotrebom optimističkog zaključavanja. U model klasu FishingTrip dodato je polje int version sa anotacijom @Version. Tako, ako instruktor i klijent istovremeno rezervišu na istu uslugu, prvi korisnik će zakazati rezervaciju, polje version će se tada povećati za 1, te drugi korisnik dobiti poruku greške. Metoda addReservation servisne klase FishingTripService je anotirana sa anotacijom @Transactional.

2. Instruktor ne može da napravi akciju u isto vrijeme kad i drugi klijent vrši rezervaciju postojećeg entiteta

Problem: Instruktor može da kreira akciju za svoju uslugu, ali isto tako, i klijent može da zakaže rezervaciju na uslugu instruktora. Problem nastaje kad instruktor pokuša da kreira akciju, a klijent zakaže rezervaciju za istu uslugu u isto vrijeme, gdje bi se moglo dogoditi da se i akcija i rezervacija upišu u bazu.

Tok zahtjeva do konfliktne situacije:

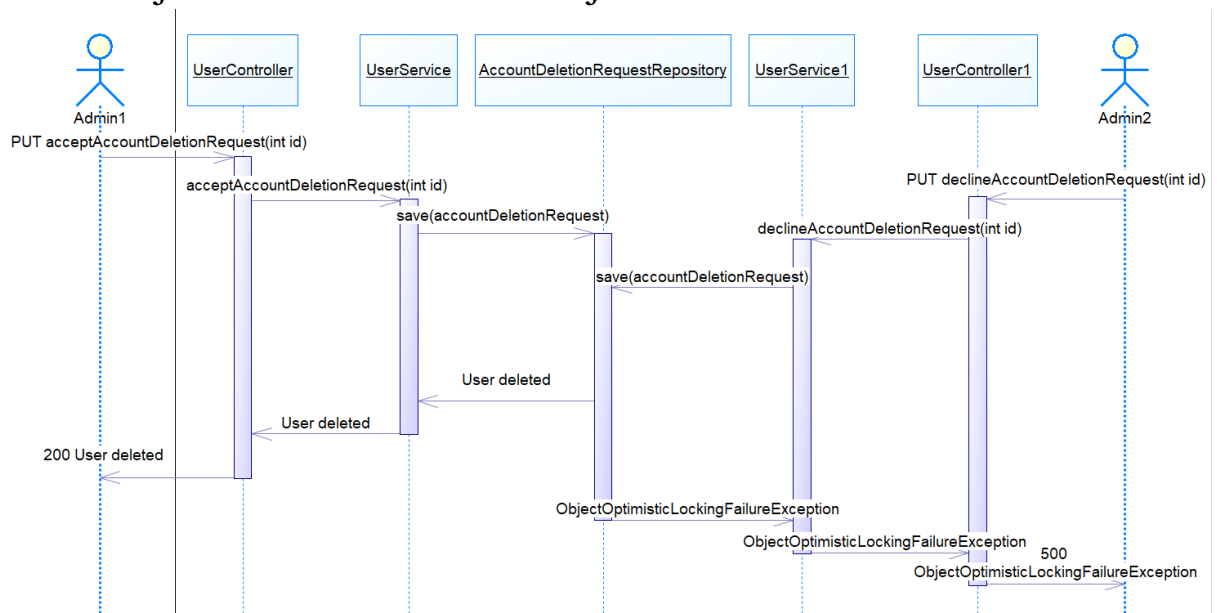


Rješenje: Konfliktna situacija riješena je upotrebom optimističkog zaključavanja. U model klasu FishingTrip dodato je polje int version sa anotacijom @Version. Tako, ako instruktor kreira akciju, a klijent rezerviše, na istu uslugu u isto vrijeme, prvi korisnik će biti uspješan, polje version će se tada povećati za 1, te drugi korisnik dobiti poruku greške. Metode addReservation i addQuickReservation servisne klase FishingTripService su anotirane sa anotacijom @Transactional.

3. Na jedan zahtjev za brisanje naloga može da odgovori samo jedan administrator sistema

Problem: Svi korisnici sistema mogu da pošalju zahtjev za brisanje naloga. Administrator može (uz obrazloženje) da prihvati ili odbije taj zahtjev. Problem nastaje kada jedan administrator pokuša da prihvati, a drugi odbije isti zahtjev u isto vrijeme, gdje bi se moglo desiti da obje akcije prođu, što dovodi do toga da korisnik dobija dva maila, prvi u kome mu je zahtjev prihvaćen, a drugi u kome mu je odbijen.

Tok zahtjeva do konfliktne situacije:

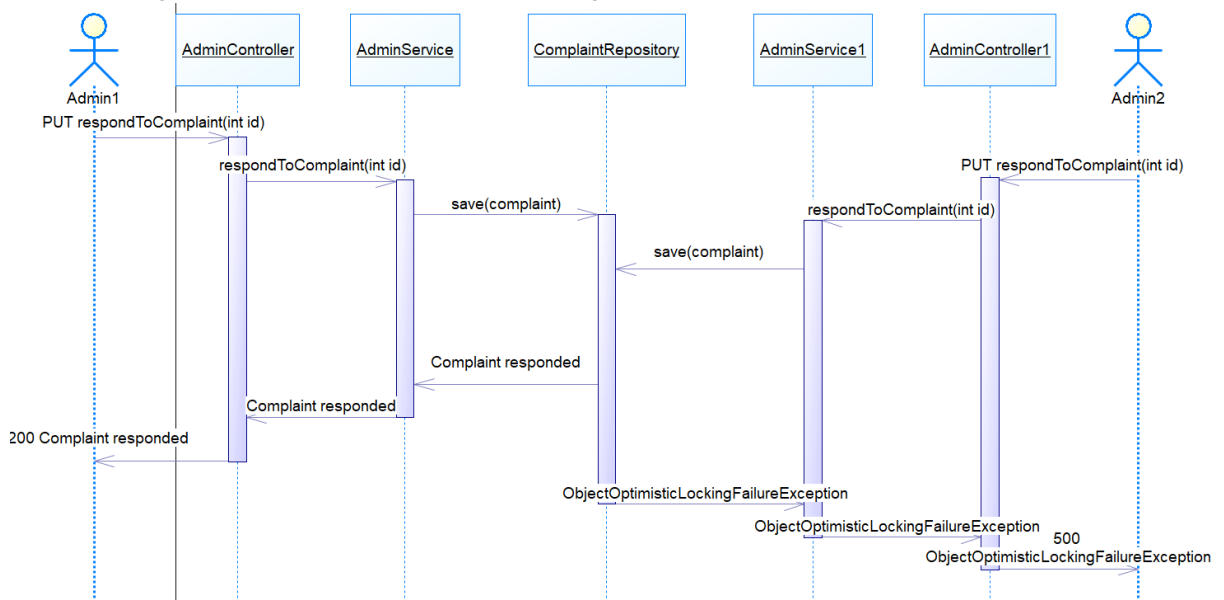


Rješenje: Konfliktna situacija riješena je upotrebom optimističkog zaključavanja. U model klasu AccountDeletionRequest dodato je polje int version sa anotacijom @Version. Tako, ako dva administratora istovremeno odgovaraju na isti zahtjev za brisanje naloga, prvi administrator će biti uspješan, polje version će se tada povećati za 1, te će drugi administrator dobiti poruku greške. Metode acceptAccountDeletionRequest i declineAccountDeletionRequest servisne klase UserService su anotirane sa anotacijom @Transactional.

4. Na jednu žalbu može da odgovori samo jedan administrator sistema

Problem: Klijenti šalju žalbe vezane za usluge sistema. Administrator može da odgovara na žalbe. Problem nastaje kada dva administratora pokušaju da odgovore na istu žalbu istovremeno, gdje bi se moglo desiti da obje akcije prođu, što dovodi do toga da korisnik i pružalac usluge dobiju po dva maila, koji mogu biti kontradiktorni.

Tok zahtjeva do konfliktne situacije:

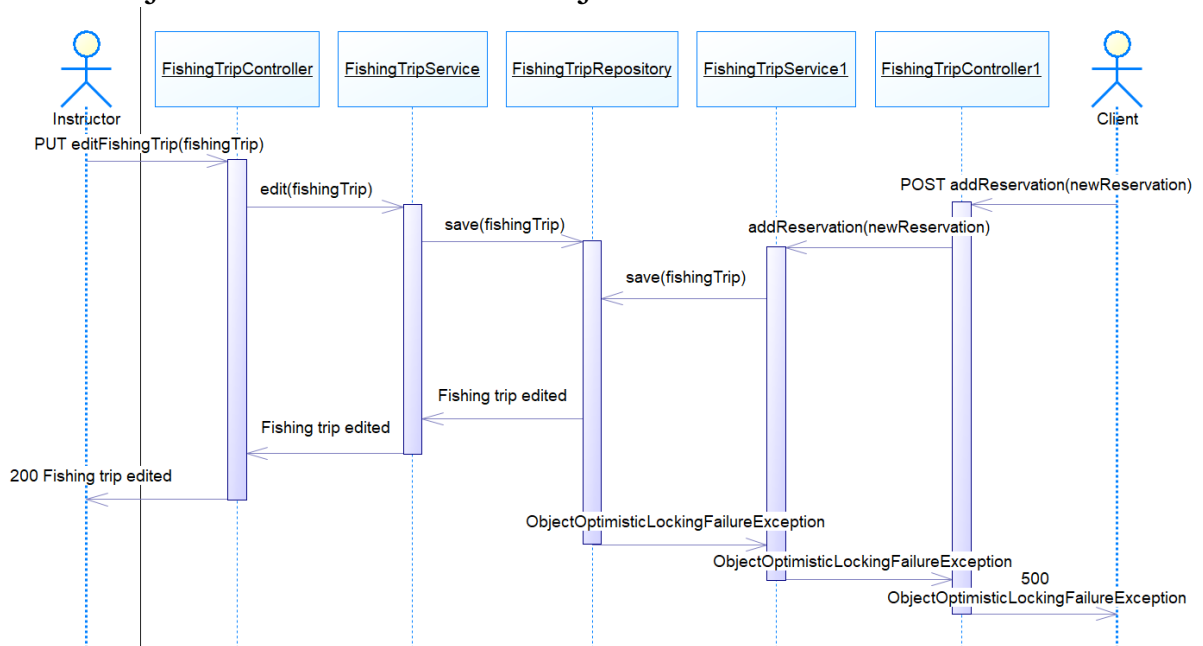


Rješenje: Konfliktna situacija riješena je upotrebom optimističkog zaključavanja. U model klase Complaint i ComplaintFishingInstructor dodato je polje int version sa anotacijom @Version. Tako, ako dva administratora istovremeno odgovaraju na istu žalbu, prvi administrator će biti uspješan, polje version će se tada povećati za 1, te će drugi administrator dobiti poruku greške. Metode respondToComplaint i respondToComplaintFishingInstructor servisne klase AdminService su anotirane sa anotacijom @Transactional.

5. Instruktor ne može da mijenja svoju uslugu u isto vrijeme kad i klijent vrši rezervaciju na nju

Problem: Instruktor može da mijenja podatke za svoju uslugu, ali isto tako, i klijent može da zakaže rezervaciju na uslugu instruktora. Problem nastaje kad instruktor pokuša da izmjeni podatke usluge, a klijent zakaže rezervaciju za istu uslugu u isto vrijeme, gdje bi se moglo dogoditi da se i izmjenjeni podaci o usluzi i rezervacija upišu u bazu, što bi na primjer moglo dovesti do toga da rezervacija sadrži neke dodatne pogodnosti koje usluga ne pruža.

Tok zahtjeva do konfliktne situacije:



Rješenje: Konfliktna situacija riješena je upotrebom optimističkog zaključavanja. U model klasu FishingTrip dodato je polje int version sa anotacijom @Version. Tako, ako instruktor mijenja podatke, a klijent rezerviše, na istu uslugu u isto vrijeme, prvi korisnik će biti uspješan, polje version će se tada povećati za 1, te drugi korisnik dobiti poruku greške. Metode addReservation i edit servisne klase FishingTripService su anotirane sa anotacijom @Transactional.

