

Konflikte situacije i njihova rješenja – Dragan Mirković, student 1

- 1) Više istovremenih klijenata ne mogu da naprave rezervaciju istog entiteta u isto (ili preklapajuće) vreme
- 2) Više istovremenih klijenata ne mogu da naprave rezervaciju istog entiteta na akciji u isto vreme
- 3) Klijent šalje više istovremenih zahtjeva za otkazivanje rezervacije

1)

Prilikom rezervisanja entiteta za odgovajući termin, dostupnost entiteta treba da bude uvijek ispravna i da se ne može desiti da dva ili više klijenata rezervišu isti termin za isti entitet i da dođe do preklapanja. Za rješavanje ovog problema odabrano je optimističko zaključavanje. U klasi “RentingEntity”, koja predstavlja apstraktnu klasu koju naslijeđuju klase “Ship”, “VacationHouse”, “FishingClass”, dodat je integer atribut “version” sa anotacijom “@Version”, koji hibernate koristi za kontrolu toka verzije objekta. Uz to dodata je i “@Transactional” notacija iznad metode “makeClientReservation” u servisu “ClientService”. Ova metoda prati verziju iznajmljujućeg objekta i ako je došlo do njegove izmjene prilikom rezervisanja baca se izuzetak “ObjectOptimisticLockingFailureException” nakon čega se baza vraća u prethodno stanje, a klijent se obavještava da pokuša kasnije.

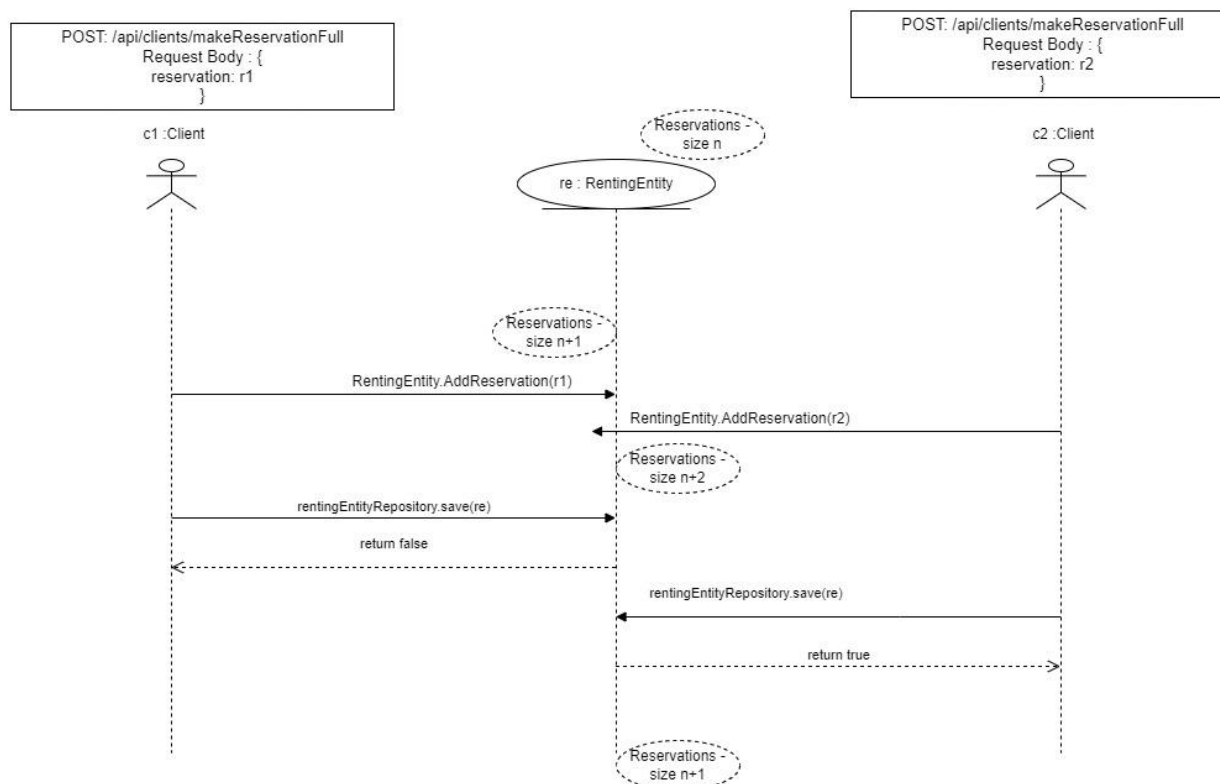


Figure 1 Dijagram konflikta 1

2)

Prilikom rezervisanja entiteta na akciji, dostupnost entiteta treba da bude uvijek ispravna i da se ne može desiti da dva ili više klijenata rezervišu isti entitet na akciji i da dođe do preklapanja. Za rješavanje ovog problema odabrano je optimističko zaključavanje. U klasi “RentingEntity”, koja predstavlja apstraktnu klasu koju naslijeđuju klase “Ship”, “VacationHouse”, “FishingClass”, dodat je integer atribut “version” sa anotacijom “@Version”, koji hibernate koristi za kontrolu toka verzije objekta. Uz to dodata je i “@Transactional” notacija iznad metode “makeQuickReservation” u servisu “ClientService”. Ova metoda prati verziju iznajmljujućeg objekta i ako je došlo do njegove izmjene prilikom rezervisanja baca se izuzetak “ObjectOptimisticLockingFailureException” nakon čega se baza vraća u prethodno stanje, a klijent se obavještava da je entitet već zauzet.

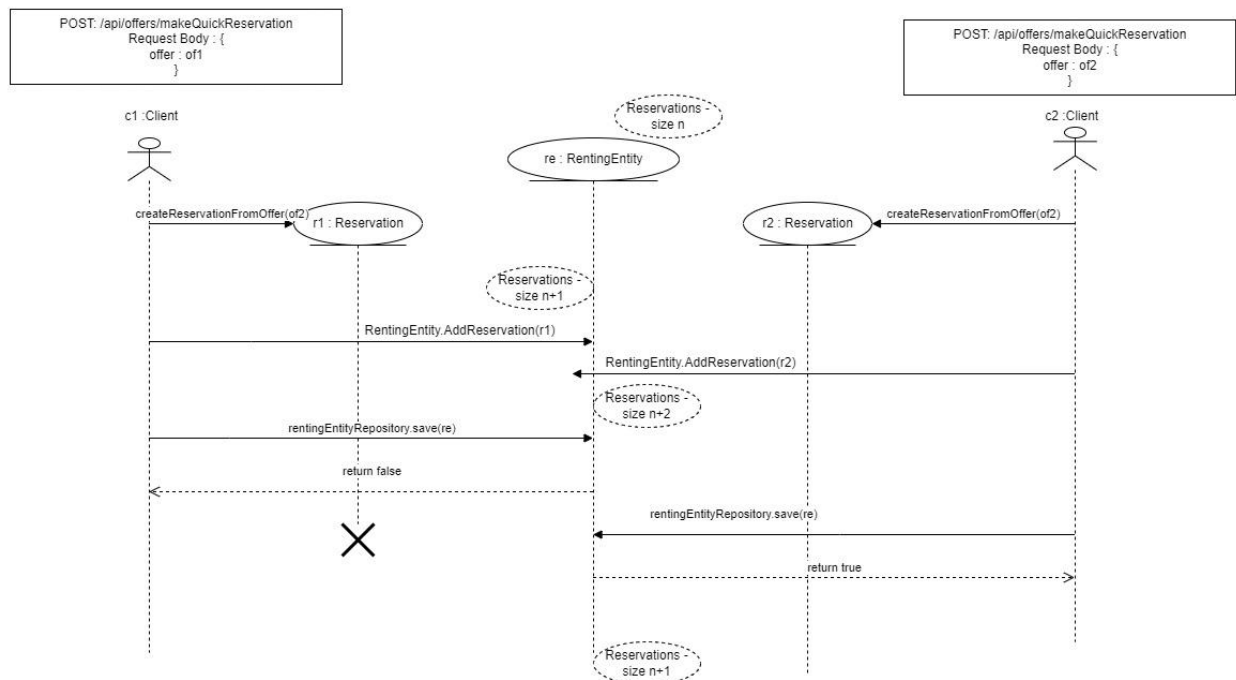


Figure 2 Dijagram konflikta 2

3)

Prilikom slanja zahtjeva za otkazivanje rezervacije, potrebno je da se zahtjev izvrši jednom i da ne dođe do višestrukog otkazivanja. Za rješavanje ovog problema odabrano je optimističko zaključavanje. U klasi “Reservation”, koja predstavlja klasu rezervacije određenog iznajmljujućeg entiteta, dodat je integer atribut “version” sa anotacijom “@Version”, koji hibernate koristi za kontrolu toka verzije objekta. Uz to dodata je i “@Transactional” notacija iznad metode “cancelReservation” u servisu “ReservationService”. Ova metoda prati verziju iznajmljujućeg objekta i ako je došlo do njegove izmjene prilikom višestrukog otkazivanja baca se izuzetak “ObjectOptimisticLockingFailureException” i izvršava se samo 1 zahtjev.

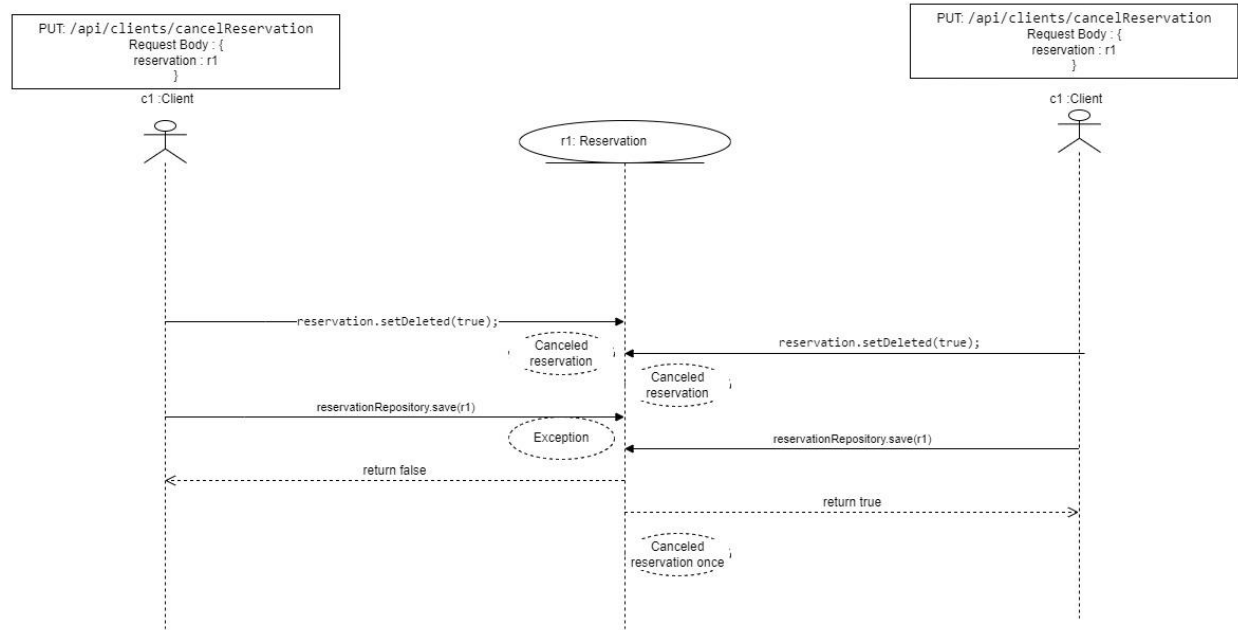


Figure 3 Diagram konflikta 3