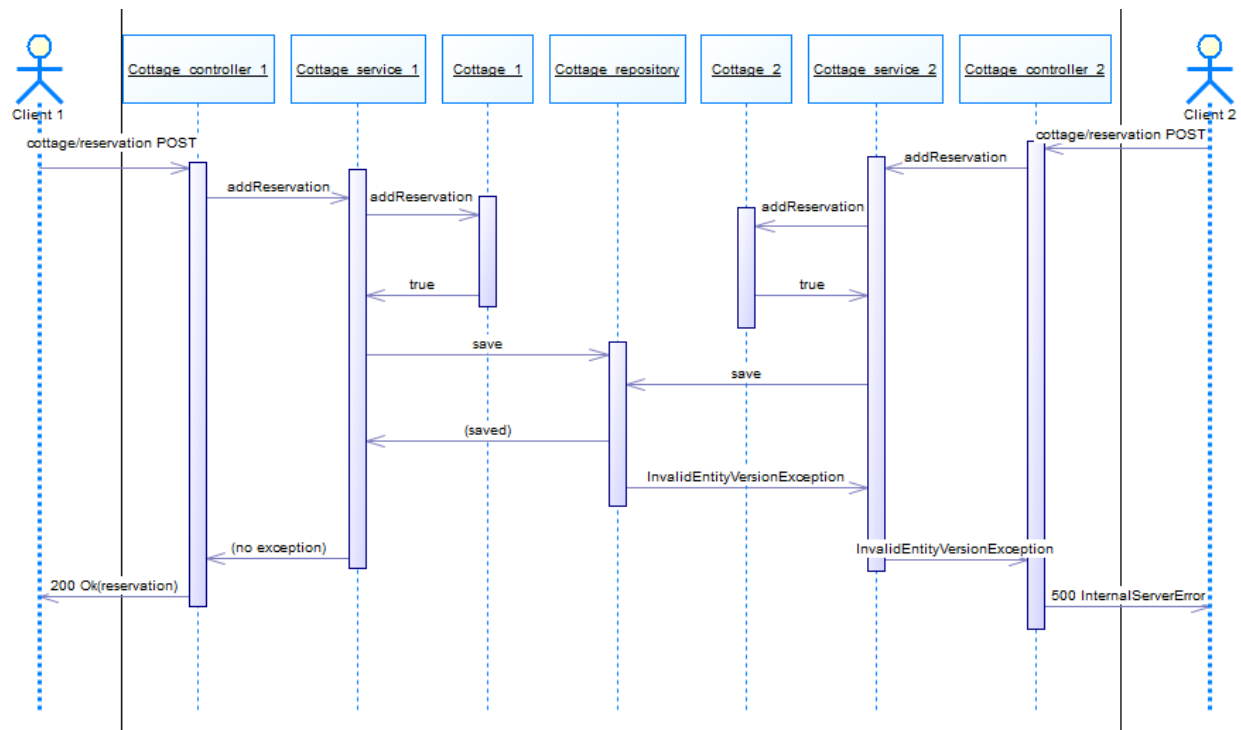


Konkurentni pristup bazi – Student 1

1. Više istovremenih klijenata ne mogu da naprave rezervaciju istog entiteta u isto (ili preklapajuće) vreme

Problem: Klijenti koji pokušaju da istovremeno zakažu rezervaciju istog entiteta u preklapajućem terminu, entitet će, u slučaju da ovaj problem nije rešen, zakazati obe rezervacije, i samim time će doći do konflikte situacije kada se oba klijenta pojave u vreme svoje rezervacije koja može da podrži samo jednog. Aplikacija mora da obezbedi da baza podataka nikada ne dođe u situaciju u kojoj je dozvoljeno upisati dve rezervacije u isto vreme, i u slučaju kada klijenti pošalju svoje zahteve u isto vreme, pa je na pri prijemu zahteva validno pravljenje druge, konfliktne rezervacije. (Napomena: Važi za sve entitete, rešenje prikazano u slučaju vikendice)

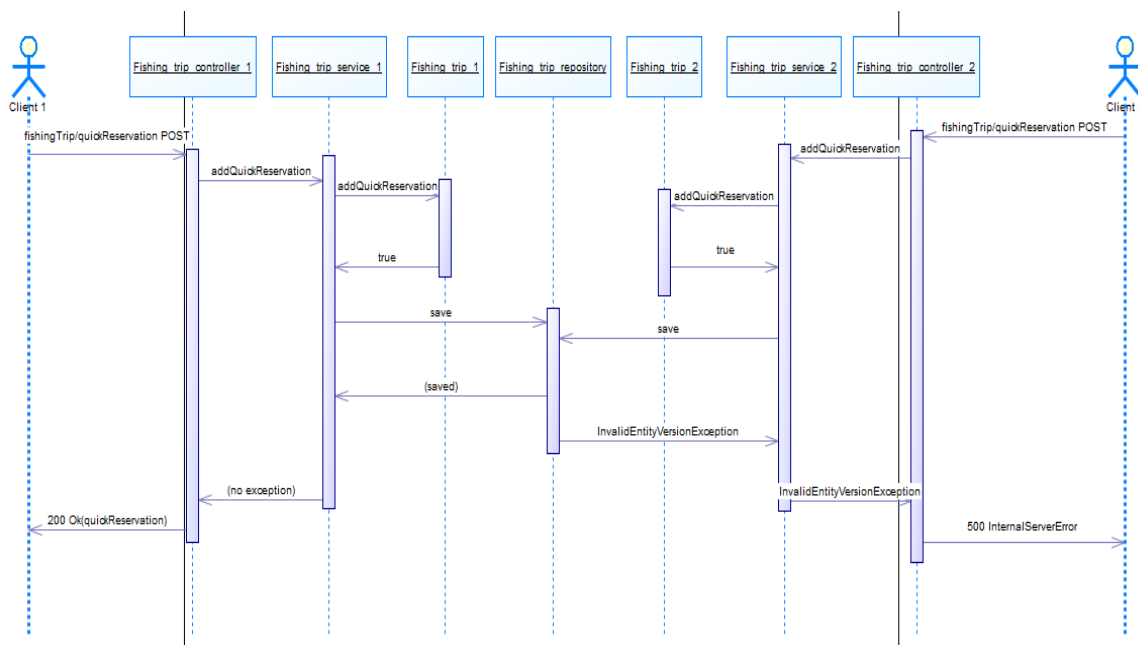


Rešenje: Konfliktna situacija je rešena upotrebom optimističkog zaključavanja. Optimističko zaključavanje implementirano je uvođenjem verzija vikendica u bazi. Verzionisanje je izvedeno tako što se u klasu *SaleEntity*, koju nasleđuje klasa *Cottage*, dodalo polje *version*, sa anotacijom *javax.persistence.Version*. Oba klijenta poziva *addReservation* metodu interfejsa *CottageService* (implementacija u *CottageServiceImpl*), sa anotacijom *org.springframework.transaction.annotation.Transactional*, sa default osobinom *Propagation.REQUIRED*. Na ovaj način, ako oba klijenta pokušaju da obezbede termin rezervacije u isto vreme, drugom klijentu će se prikazati poruka da pokuša

opet (Uz pretpostavku da je prvi klijent trenutak ranije pokrenuo obezbeđivanje, u suprotnom, prvom klijentu se prikazuje poruka).

2. Više istovremenih klijenata ne mogu da naprave rezervaciju istog entiteta na akciji u isto vreme

Problem: Klijenti koji pokušaju da istovremeno zakažu brzu rezervaciju istog entiteta na akciji u preklapajućem terminu, entitet će, u slučaju da ovaj problem nije rešen, zakazati obe rezervacije, i samim time će doći do konflikte situacije kada se oba klijenta pojave u vreme svoje rezervacije koja može da podrži samo jednog. Aplikacija mora da obezbedi da baza podataka nikada ne dođe u situaciju u kojoj je dozvoljeno upisati dve rezervacije u isto vreme, i u slučaju kada klijenti pošalju svoje zahteve u isto vreme, pa je na prijem zahteva validno pravljenje druge, konfliktne rezervacije. (Napomena: Važi za sve entitete, rešenje prikazano u slučaju instruktora pecanja i njegovih časova)



Rešenje: Konfliktna situacija je rešena upotrebom optimističkog zaključavanja. Optimističko zaključavanje implementirano je uvođenjem verzija vikendica u bazi. Verzionisanje je izvedeno tako što se u klasu *FishingTrip*, dodalo polje *version*, sa anotacijom *javax.persistence.Version*. Oba klijenta poziva *addReservation* metodu interfejsa *FishingTripService* (implementacija u *FishingTripServiceImpl*), sa anotacijom *org.springframework.transaction.annotation.Transactional*, sa default osobinom *Propagation.REQUIRED*. Na ovaj način, ako oba klijenta pokušaju da obezbede termin rezervacije u isto vreme, drugom klijentu će se prikazati poruka da pokuša opet (Uz pretpostavku da je prvi klijent trenutak ranije pokrenuo obezbeđivanje, u suprotnom, prvom klijentu se prikazuje poruka).