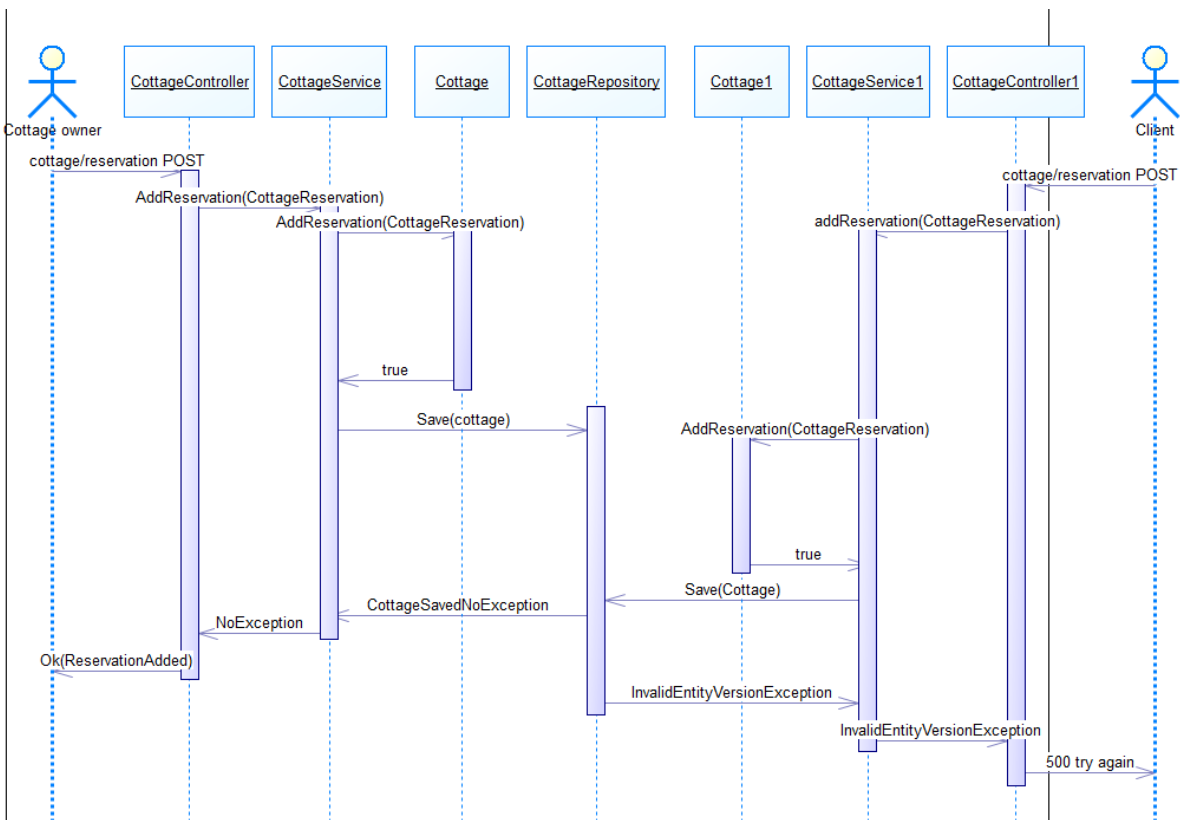


Konkurentni pristup bazi – Student 2

(Napomena, sve što se odnosi na vikendicu, odnosi se i na čamac)

1. Rezervisanje vikendica

Vikendicu mogu da rezervišu klijenti i vlasnik te vikendice. Zbog ovoga, postoji mogućnost da klijent i vlasnik pokušaju da rezervišu istu vikendicu u isto ili preklapajuće vreme. U ovom slučaju, dolazi do konfliktne situacije. Na slici 1 prikazan je dijagram sekvenci slučaja u kom vlasnik vikendice i klijent žele da rezervišu vikendicu u isto vreme, u istom terminu. U prikazanom slučaju, vlasnik vikendice je pokrenuo zakazivanje par trenutaka ranije.

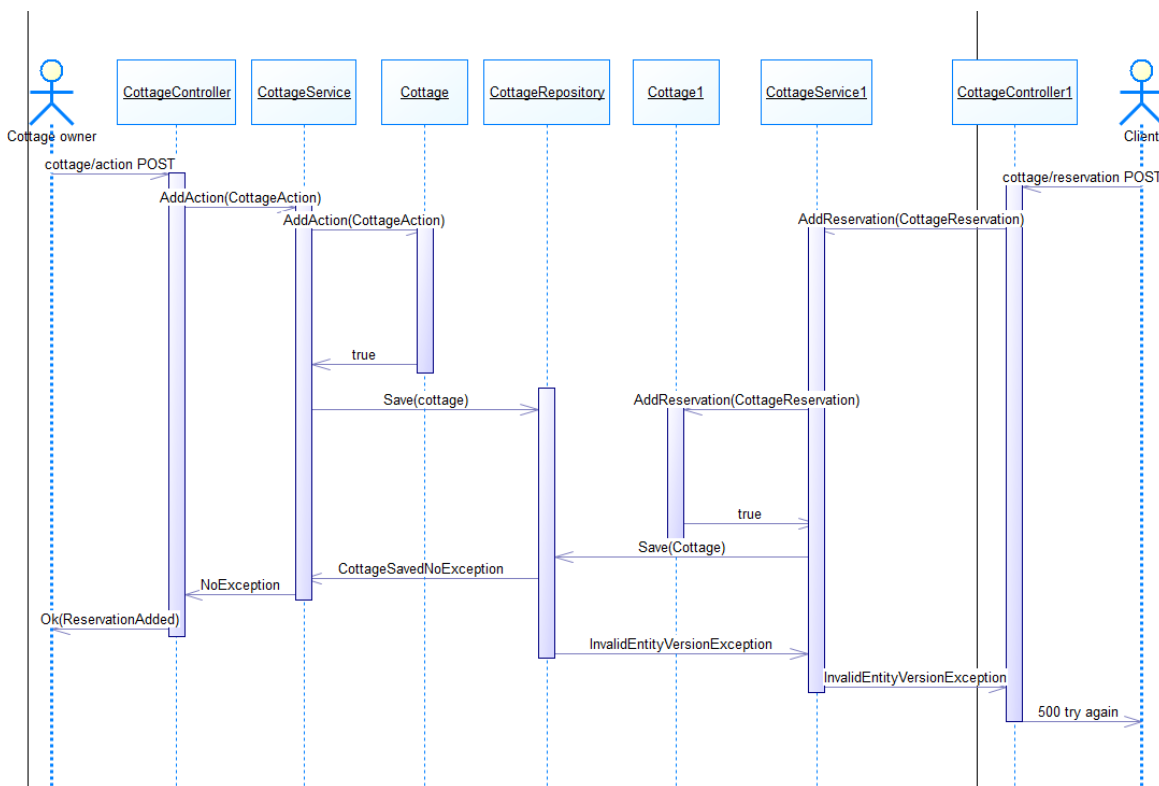


Slika 1. Istovremeno rezervisanje vikendica od strane vlasnika i klijenta

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. I vlasnik vikendice i klijent pozivaju AddReservation metodu interfejsa CottageService (implementacija u CottageServiceImpl). Ova metoda ima anotaciju org.springframework.transaction.annotation.Transactional, sa default osobinom Propagation.REQUIRED. Na ovaj način, ako vlasnik i klijent rezervišu vikendicu u isto vreme, klijentu će se prikazati poruka da pokuša opet. (Uz pretpostavku da je vlasnik trenutak ranije zakazao, u suprotnom, vlasniku se prikazuje poruka).

2. Akcije i rezervisanje

Vlasnik vikendice može da obezbedi termin za akciju, dok, u isto vreme, klijent može da rezerviše termin te vikendice. Zbog ovoga, postoji mogućnost da klijent i vlasnik pokušaju da rezervišu isti termin vikendice u isto ili preklapajuće vreme. U ovom slučaju, dolazi do konfliktne situacije. Na slici 2 prikazan je dijagram sekvenci slučaja u kom vlasnik vikendice i klijent žele da obezbede isti termin vikendice, u isto vreme. U prikazanom slučaju, vlasnik vikendice je pokrenuo obezbeđivanje termina par trenutaka ranije.



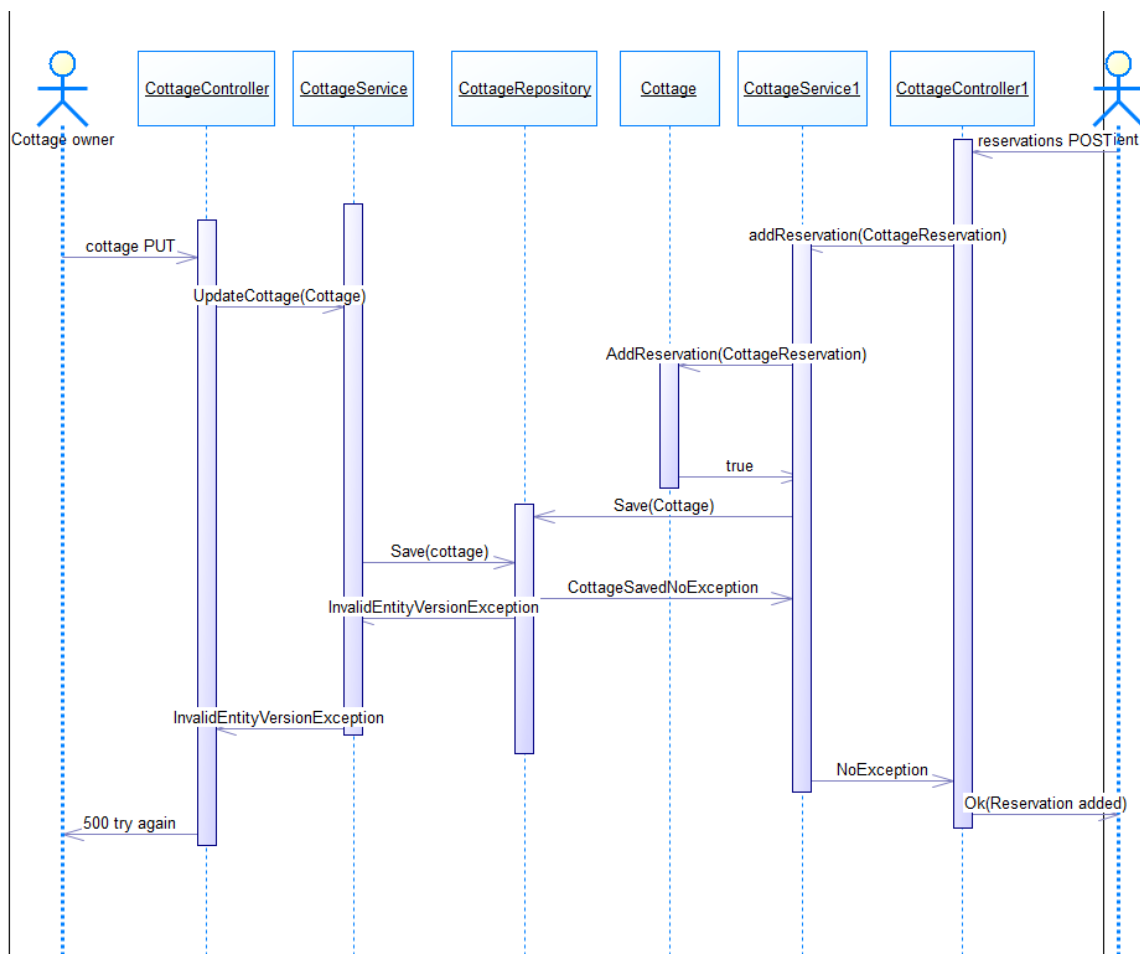
Slika 2. Istovremeno definisanje akcije i rezervisanje vikendice

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. Vlasnik vikendice poziva AddAction metodu interfejsa CottageService (implementacija u CottageServiceImpl), dok klijent poziva AddReservation metodu istog interfejsa. Obe metode imaju anotaciju org.springframework.transaction.annotation.Transactional, sa default osobinom Propagation.REQUIRED. Na ovaj način, ako vlasnik i klijent pokušavaju da obezbede termin vikendice u isto vreme, klijentu će se prikazati poruka da pokuša opet (Uz pretpostavku da je vlasnik trenutak ranije pokrenuo obezbeđivanje, u suprotnom, vlasniku se prikazuje poruka).

3. Ažuriranje podataka vikendice i rezervisanje vikendice od strane klijenta

Vlasnik vikendice ima mogućnost da ažurira podatke vikendice, pod uslovom da ta vikendica nije rezervisana u budućnosti, i da nema definisanih akcija u budućnosti. Klijent, takođe, može da rezerviše termin vikendice. Ukoliko vlasnik pokuša da ažurira vikendicu u isto vreme kada i klijent pokuša da rezerviše termin vikendice, dolazi do konfliktne situacije jer klijent može da kreira rezervaciju sa

nevažećom cenom, pravilima ponašanja i dodatnim uslugama. Na slici 3 prikazan je dijagram sekvenci slučaja kada vlasnik vikendice ažurira podatke vikendice u isto vreme kada klijent pokušava da rezerviše termin vikendice. U prikazanom slučaju, vlasnik je pokrenuo ažuriranje trenutak nakon klijenta.



Slika 3. Istovremeno ažuriranje i rezervisanje vikendice

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. Vlasnik vikendice poziva UpdateCottage metodu interfejsa CottageService (implementacija u CottageServiceImpl), dok klijent poziva AddReservation metodu istog interfejsa. Obe metode imaju anotaciju org.springframework.transaction.annotation.Transactional, sa default osobinom Propagation.REQUIRED. Na ovaj način, ako vlasnik pokušava da ažurira vikendicu par trenutaka pre nego što klijent rezerviše termin, izvršiće se ažuriranje vikendice, a klijentu će se prikazati poruka da pokuša ponovo. Ako klijent pokrene rezervaciju par trenutaka pre vlasnika, napraviće se rezervacija a vlasniku će se prikazati poruka da pokuša opet.