

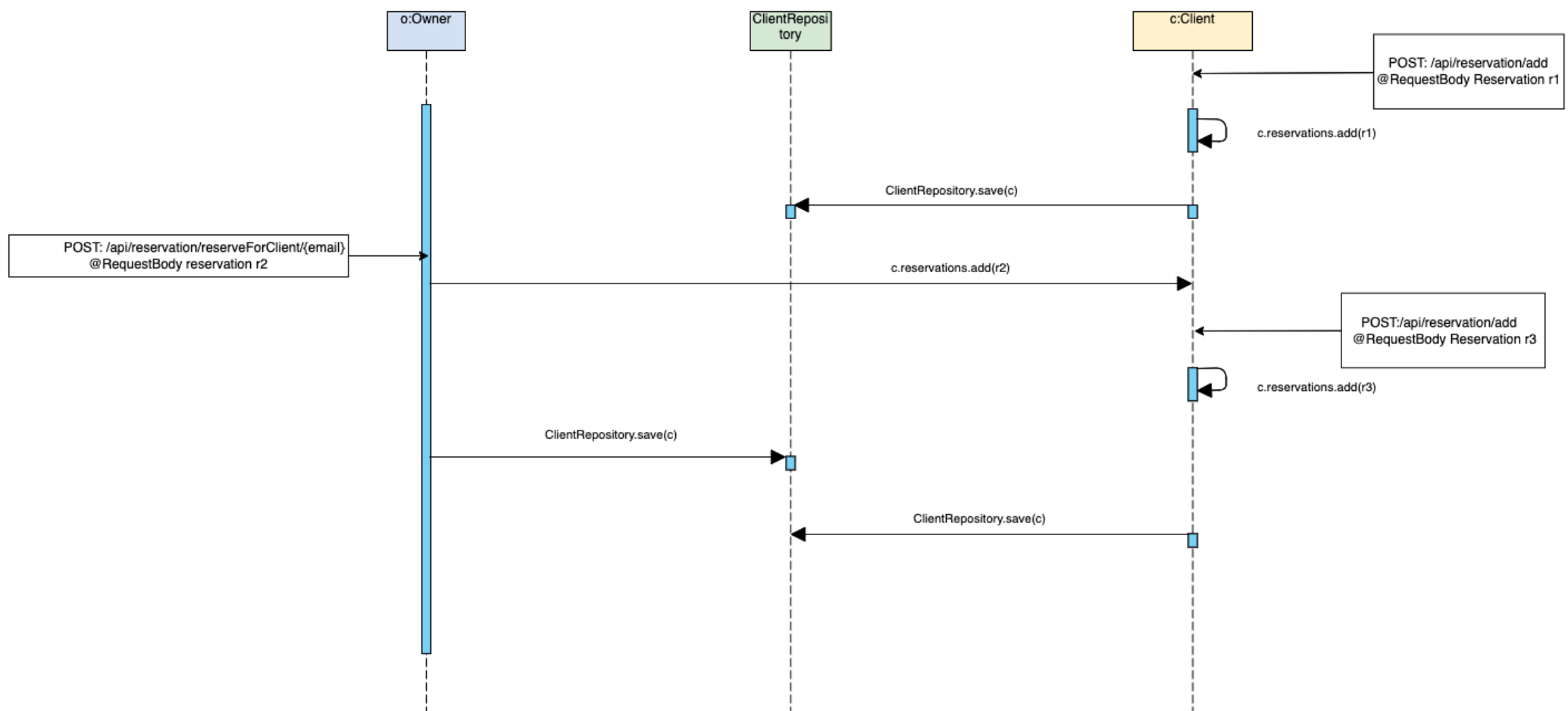
# Rešavanje konfliktnih situacija

Mičo Milić SW68/2019

1. Vlasnik vikendice/broda ili instruktor ne može da napravi rezervaciju u isto vreme kad i drugi klijent

Pri rezervaciji entiteta, njegovi termini dostupnosti moraju biti konzistentni, ne smeju se pojaviti dve ili više rezervacija za klijenta u istom terminu za isti entitet. Ako se desi da vlasnik kreira rezervaciju za klijenta u isto vreme kada i klijent kreira takvu rezervaciju može doći do konflikta gde bi se u ovom primeru napravile rezervacije r2 i r3 i dodale u klijentove rezervacije.

Visual Paradigm Online Free Edition



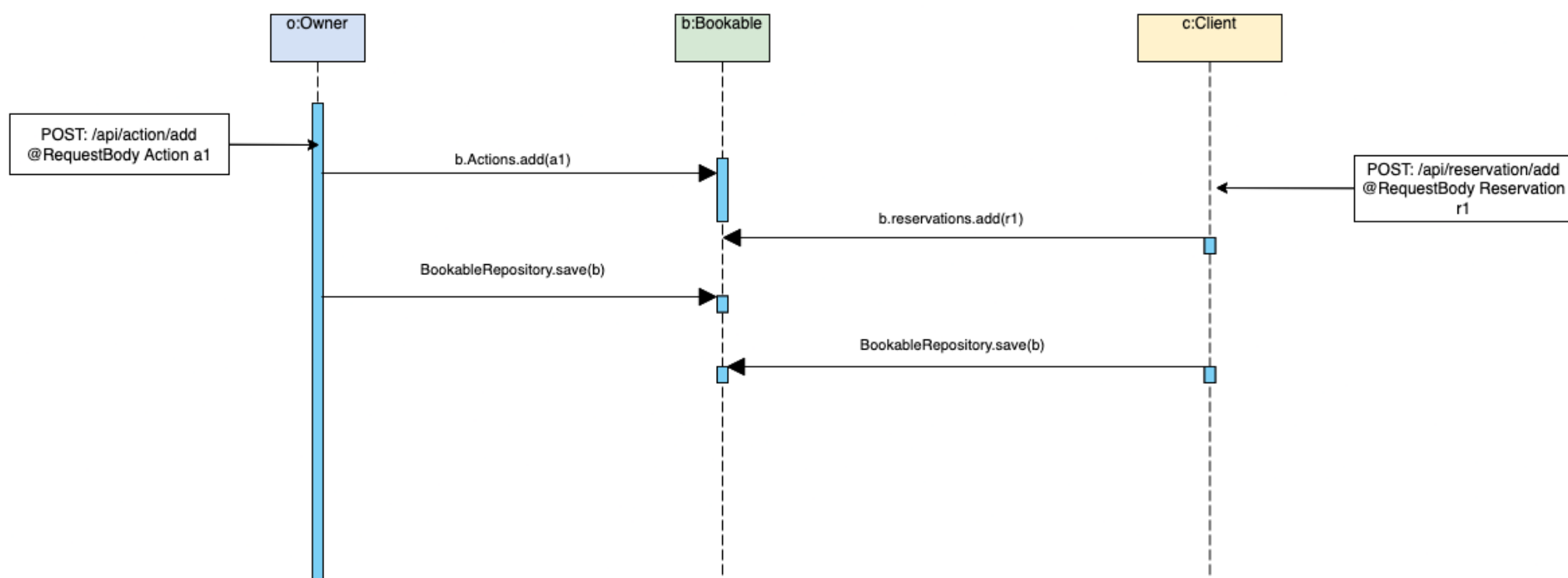
Visual Paradigm Online Free Edition

Konflikt se rešava tako što se u klasi "Person" koja je roditeljska klasa klase "Client" dodaje atribut "version" tipa integer sa hibernate anotacijom @Version pomoću čega se prati da li je objekat izmenjen u toku transakcije. Takodje metoda za rezervaciju na servisu anotirana sa @Transactional. Tokom rezervacije ukoliko dodje do promene objekta "Client" doći će do greške "ObjectOptimisticLockingFailureException" nakon čega se izvršava rollback, tj. transakcija se poništava. U ovom konkretnom primeru bi se rezervacija "r2" poništila dok bi se rezervacija "r3" sačuvala.

2. Vlasnik vikendice/broda ili instruktor ne može da napravi akciju u isto vreme kada i drugi klijent vrši rezervaciju postojećeg entiteta

Ako se desi da vlasnik kreira akciju za entitet u isto vreme kada i klijent vrši rezervaciju za taj entitet može doći do konflikta gde bi se u ovom primeru napravila akcija a1 iako je klijent pre njenog čuvanja dodao rezervaciju r1.

Visual Paradigm Online Free Edition

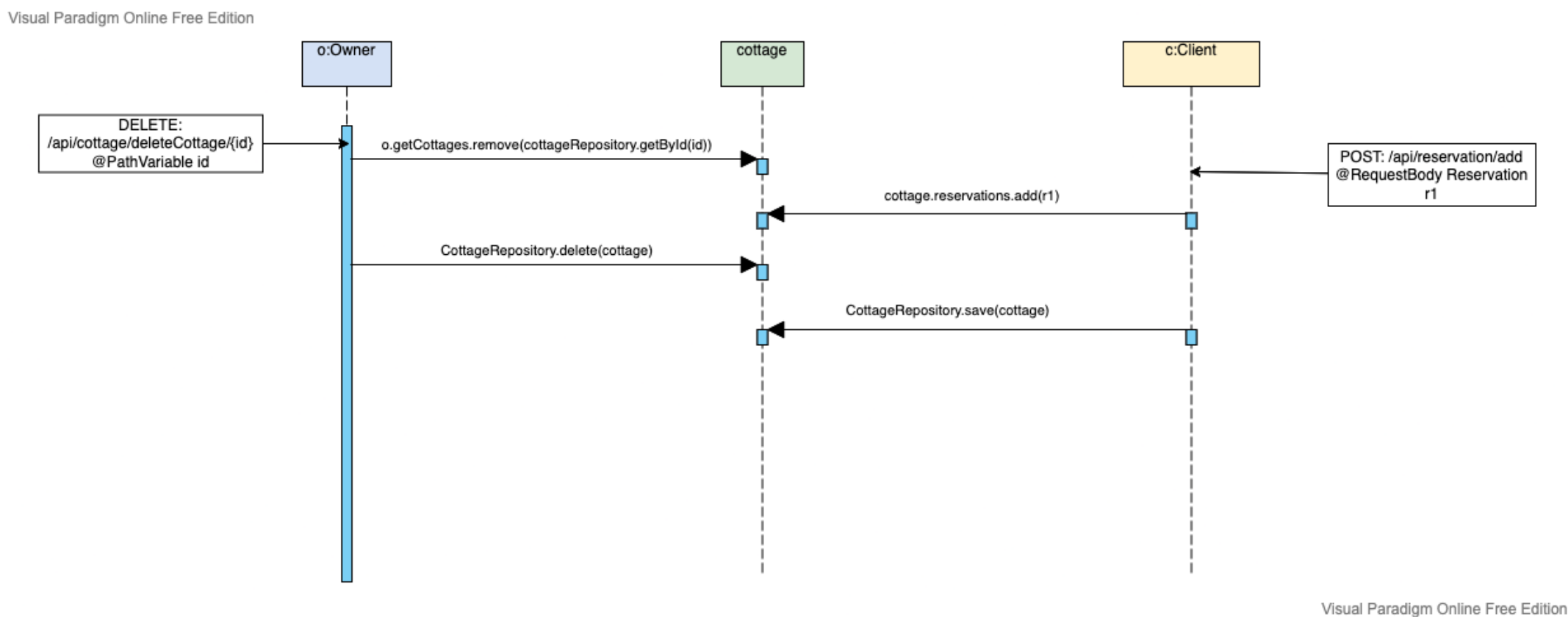


Visual Paradigm Online Free Edition

Konflikt se rešava tako što se u klasi "Bookable" dodaje atribut "version" tipa integer sa hibernate anotacijom @Version pomoću čega se prati da li je objekat tipa "Bookable" izmenjen u toku transakcije. Takodje metoda na servisu za dodavanje akcija je anotirana sa @Transactional. Tokom dodavanja akcije ukoliko dodje do promene objekta "Bookable" doći će do greške "ObjectOptimisticLockingFailureException" nakon čega se izvršava rollback, tj. transakcija se poništava. U ovom konkretnom primeru bi se rezervacija "r1" sačuvala dok bi se dodavanje akcije a1 poništilo.

3. Vlasnik vikendice/broda ili instruktor ne može da obriše entitet u isto vreme kada i drugi klijent vrši rezervaciju postojećeg entiteta

Ako se desi da vlasnik briše entitet u isto vreme kada i klijent vrši rezervaciju za taj entitet može doći do konflikta prilikom kog bi klijent napravio rezervaciju i ona bi se sačuvala za entitet koji je pre toga obrisao. U ovom primeru napravila bi se rezervacija r1 iako je cottage obrisano.



Konflikt se rešava tako što se u klasi “Bookable” dodaje atribut “version” tipa integer sa hibernate anotacijom `@Version` pomoću čega se prati da li je objekat tipa “Bookable” izmenjen u toku transakcije. Takodje, metoda na servisu za brisanje entiteta je anotirana sa `@Transactional`. Tokom brisanja entiteta ukoliko dodje do promene objekta “Bookable” doći će do greške “`ObjectOptimisticLockingFailureException`” nakon čega se izvršava rollback, tj. transakcija se poništava. U ovom konkretnom primeru bi se rezervacija “r1” sačuvala dok bi se brisanje vikendice poništilo.