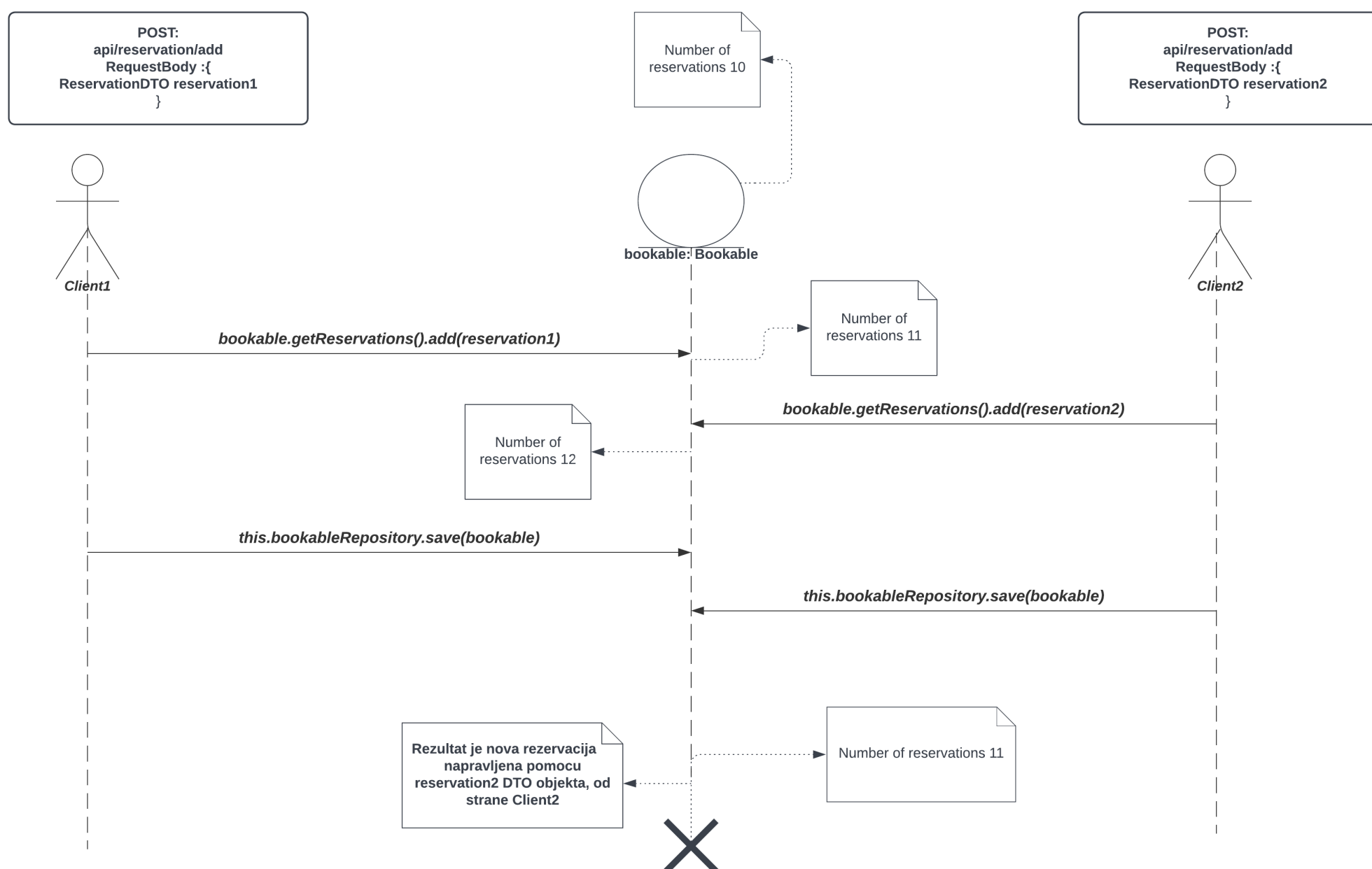


# Konkurentni pristup resursima u bazi

## 1. Rezervacija entiteta

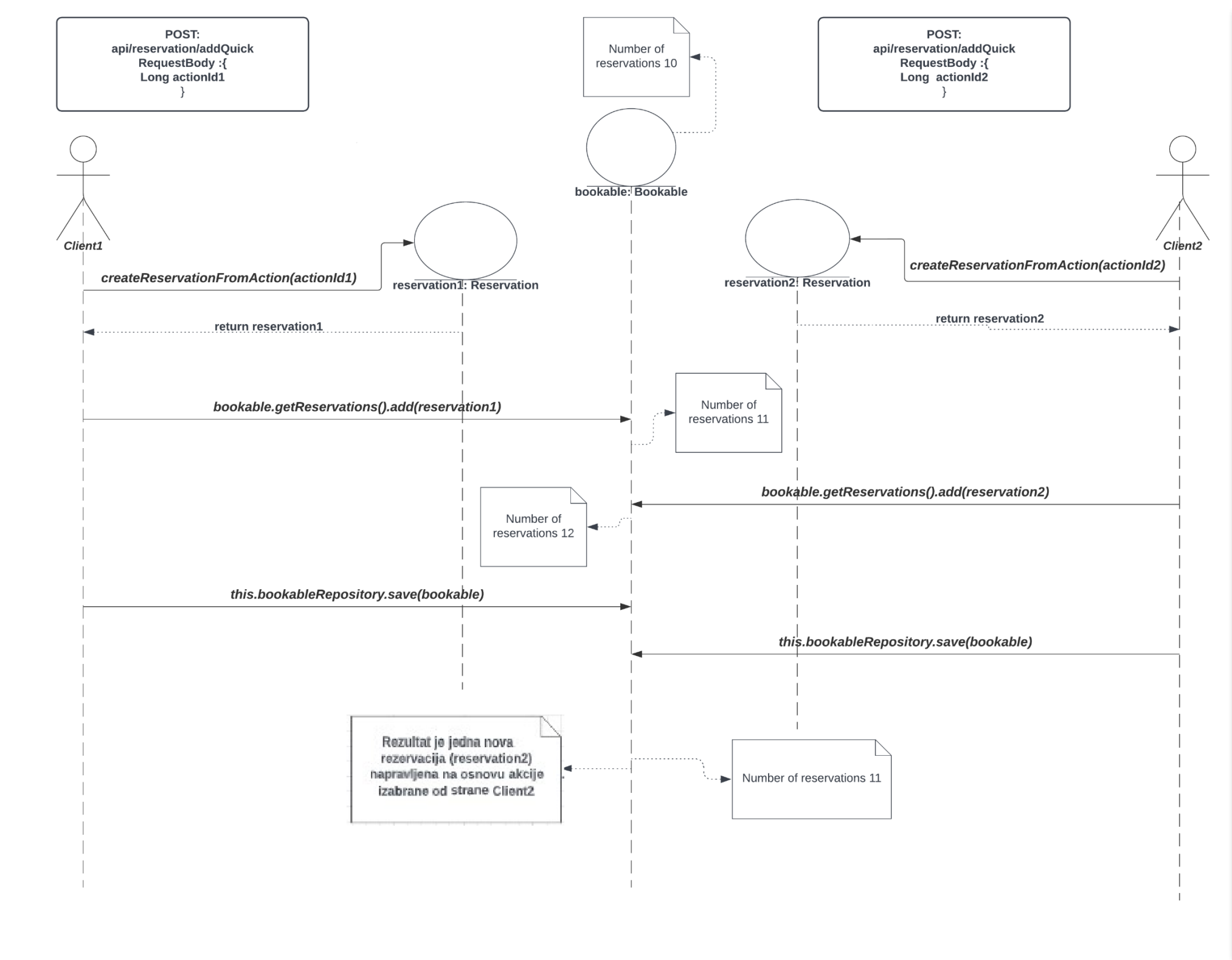
U projektu postoji više situacija gde može doći do konflikata. Jedna od njih jeste rezervacija istog entiteta u isto vreme od strane više klijenata. Ukoliko više klijenata u isto vreme pokuša da rezerviše isti entitet, doći će do nekonzistentnog stanja odnosno *Client1* će rezervisati entitet, dok će *Client2* takodje pokušati da rezerviše. Tada *Client1* neće biti svestan da, nakon čuvanja, njegova rezervacija nije sačuvana nego da se samo rezervacija drugog klijenta nalazi u bazi.



Ovaj problem je rešen upotrebom optimističkog zaključavanja. U klasi *Bookable* je dodat novi atribut *version* sa anotacijom *@Version*. Zatim je iznad metode *addReservation()* u servisu *ReservationService* dodata anotacija *@Transactional*. Kombinovanjem ove dve anotacije proverava se da li je došlo do promene verzije objekta u toku transakcije u odnosu na verziju koja je preuzeta na početku transakcije. Ukoliko se verzija promeni, izaziva se *ObjectOptimisticLockingFailureException* nakon kog se baza vraća u prethodno stanje.

## 2. Rezervacija entiteta na akciji

Takodje, do konflikta dolazi i kada više klijenata rezerviše isti entitet na akciji u isto vreme. Ukoliko više klijenata u isto vreme pokuša da rezerviše isti entitet na akciji, doći će do nekonzistentnog stanja odnosno *Client1* će rezervisati akciju, dok će *Client2* takodje pokušati da rezerviše. Tada *Client1* neće biti svestan da, nakon čuvanja, njegova rezervacija nije sačuvana nego da se samo rezervacija akcije drugog klijenta nalazi u bazi. Ovaj problem je rešen upotrebom optimističkog zaključavanja. U klasi *Bookable* je dodat novi atribut *version* sa anotacijom *@Version*. Zatim je iznad metode *addQuickReservation()* u servisu *ReservationService* dodata anotacija *@Transactional*. Kombinovanjem ove dve anotacije proverava se da li je došlo do promene verzije objekta u toku transakcije u odnosu na verziju koja je preuzeta na početku transakcije. Ukoliko se verzija promeni, izaziva se *ObjectOptimisticLockingFailureException* nakon kog se baza vraća u prethodno stanje.



### 3. Recenzija eniteta

Prilikom ostavljanja recenzija potrebno je omogućiti da se jedan isti entitet ne recenzira od strane istog klijenta dva puta u istom trenutku. Ovaj problem je rešen upotrebom optimističkog zaključavanja. U klasi *Bookable* je dodat atribut *version* sa anotacijom *@Version*. Zatim je iznad metode *addReview()* u servisu *ReviewService* dodata anotacija *@Transactional*. Kombinovanjem ove dve anotacije proverava se da li je došlo do promene verzije objekta u toku transakcije u odnosu na verziju koja je preuzeta na početku transakcije. Ukoliko se verzija promeni, izaziva se *ObjectOptimisticLockingFailureException* nakon kog se baza vraća u prethodno stanje.

