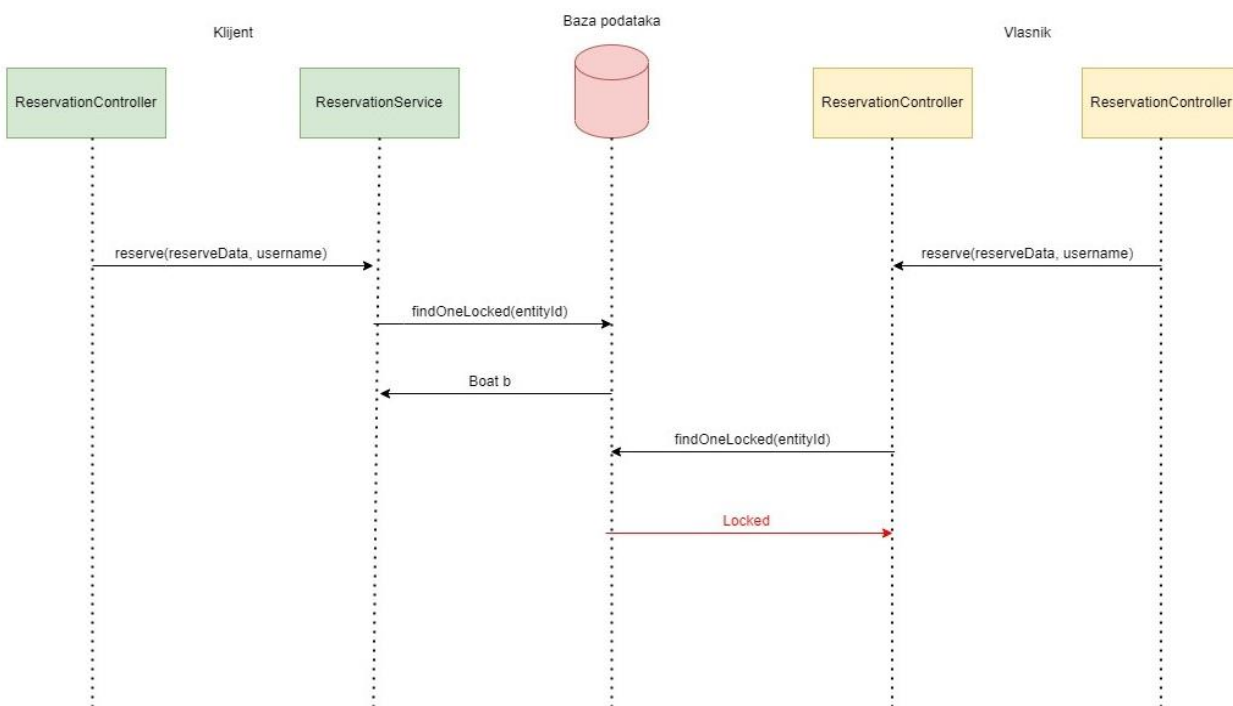


Konkurentni pristupi resursima u bazi

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

Opis problema: Vlasnik vikendice/broda ili instruktor u dogovoru sa klijentom može da napravi novu rezervaciju za tog klijenta. Pored toga, svi klijenti mogu da vrše rezervacije za vikendice, brodove ili avanture. U situaciji kada vlasnik kreira novu rezervaciju za klijenta, ako rezervacija prođe validaciju biće upisana u bazu. Ali kada bi u toku kreiranja klijent izvršio rezervaciju za isti entitet u preklapajućem terminu, validacija rezervacije bi mogla proći pre upisa vlasnikove rezervacije i samim tim bi se izvršio upis klijentove rezervacije što bi dovelo do konfliktne situacije.



Rešenje problema: Za rešenje ovog problema korišćeno je pesimističko zaključavanje resursa. U metodi za rezervaciju (ista metoda za vikendice, brodove i avanture) se dobavlja entitet koji se rezerviše kada se i zaključava sve dok rezervacija ne prođe validaciju, i u zavisnosti od ishoda validacije bude ili ne bude upisana u bazu. Ako u toku tog procesa dođe neki drugi zahtev za rezervaciju entiteta, korisnik će dobiti poruku da rezervacija nije moguća jer postoje preklapajući termini.

Zaključavanje entiteta vikendice i broda u repozitorijumima.

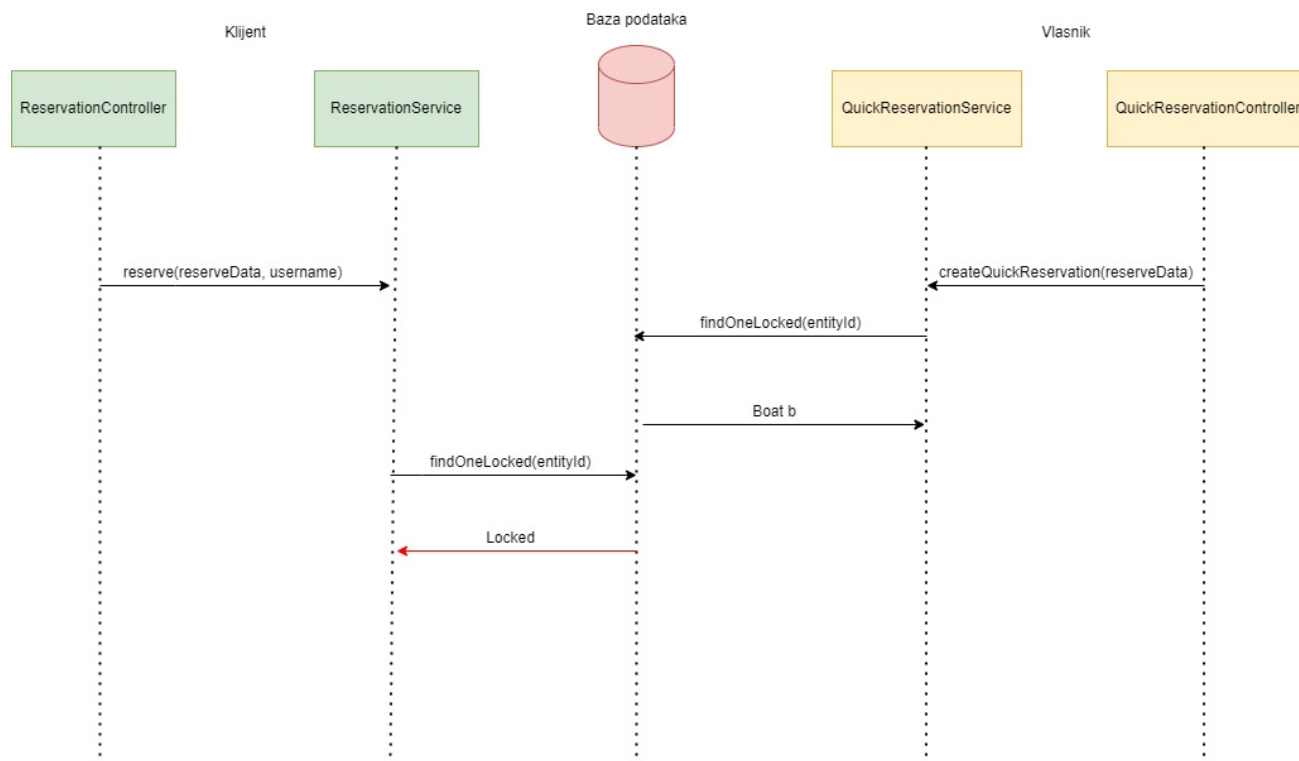
```
@Lock(LockModeType.PESSIMISTIC_WRITE)
@Query(value = "SELECT c FROM Cottage c WHERE c.id = :id")
@QueryHints({@QueryHint(name = "javax.persistence.lock.timeout", value = "0")})
Cottage findOneLocked(@Param("id") Integer id);
```

```
@Lock(LockModeType.PESSIMISTIC_WRITE)
@Query(value = "SELECT b FROM Boat b WHERE b.id = :id")
@QueryHints({@QueryHint(name = "javax.persistence.lock.timeout", value = "0")})
Boat findOneLocked(@Param("id") Integer id);
```

Klasa ReservationService je anotirana sa @Transactional. Anotacija je na nivou servisa.

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

Opis problema: Vlasnik vikendice/broda ili instruktor može da napravi novu akcijsku rezervaciju. Akcija sadrži predefinisani period trajanja, cenu, popust i broj osoba. Prilikom kreiranja brze rezervacije vrši se validacija da li entitet(vikendica,brod ili avantura) već ima rezervacija u tom periodu. Ako je validacija uspešna, akcija će biti upisana u bazu. Ali u slučaju da posle validacije, klijent rezerviše isti entitet i rezervacija se upiše u bazu, kreiranje akcije će takođe biti izvršeno iako su periodi preklapajući.U toj situaciji dolazi do konflikta.

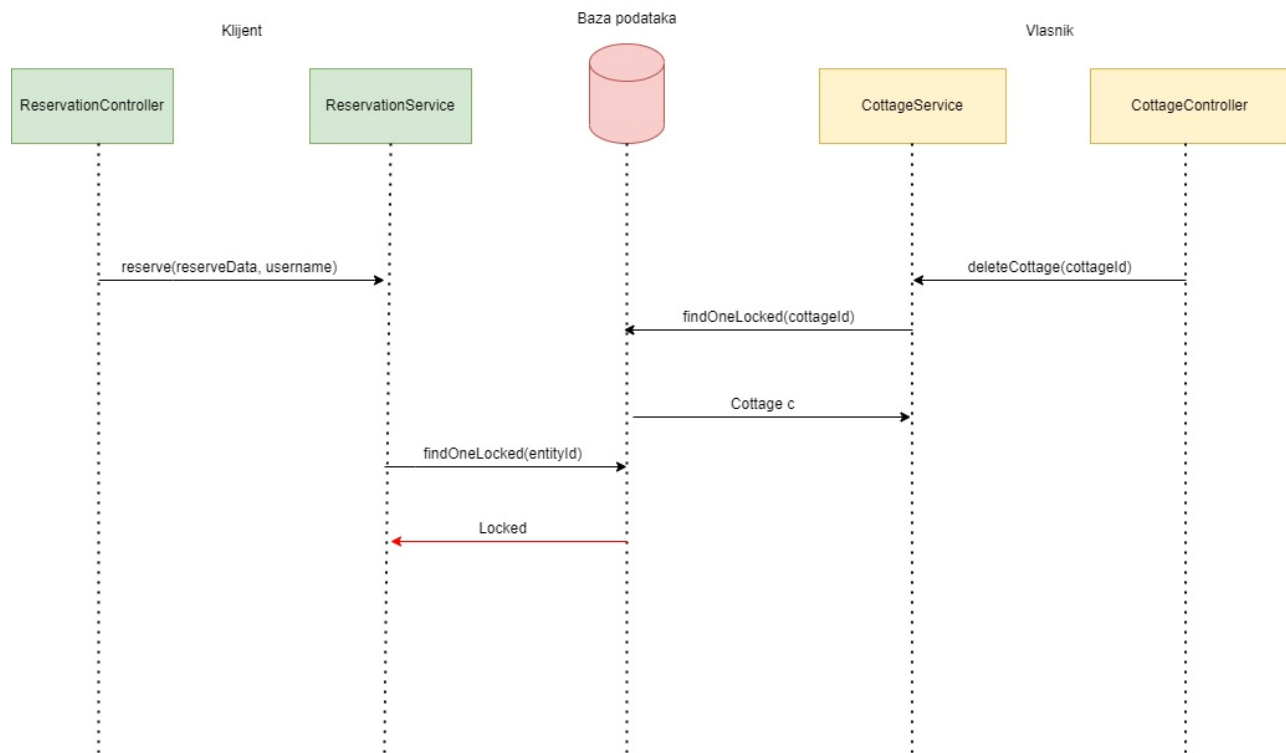


Rešenje problema: Za rešenje ovog problema korišćeno je pesimističko zaključavanje resursa. U metodi za kreiranje akcije se vrši zaključavanje entiteta za koji se pravi akcija sve dok kreiranje ne prođe validaciju, i u zavisnosti od ishoda validacije bude ili ne bude upisana u bazu. Ako u toku tog procesa dođe neki drugi zahtev za rezervaciju entiteta, korisnik će dobiti poruku da postoje preklapajući termini.

Zaključavanje entiteta je rešeno na isti način kao i u prethodnom primeru. U klasi QuickReservationService je dodata anotacija @Transactional.

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

Opis problema: Vlasnik vikendice/broda ili instruktor može da logički obriše entitet ako taj entitet nema nijednu rezervaciju u narednom periodu. Može doći do situacije da vlasnik započne brisanje i da zahtev prođe validaciju ali da u istom trenutku klijent kreira rezervaciju za isti entitet. Desiće se konflikt da je klijent napravio rezervaciju za obrisani entitet.



Rešenje problema: Za rešenje ovog problema korišćeno je pesimističko zaključavanje resursa. U metodi za brisanje se vrši zaključavanje entiteta sve dok brisanje ne prođe validaciju, i u zavisnosti od ishoda validacije bude ili ne bude logički obrisano. Ako je pre zaključavanja poslat zahtev za rezervaciju entiteta od strane klijenta, vlasnik će dobiti poruku da za entitet postoje buduće rezervacije.

Zaključavanje entiteta je rešeno na isti način kao i u prethodnom primeru. U klasama CottageService, BoatService i AdventureService su dodate anotacije @Transactional.