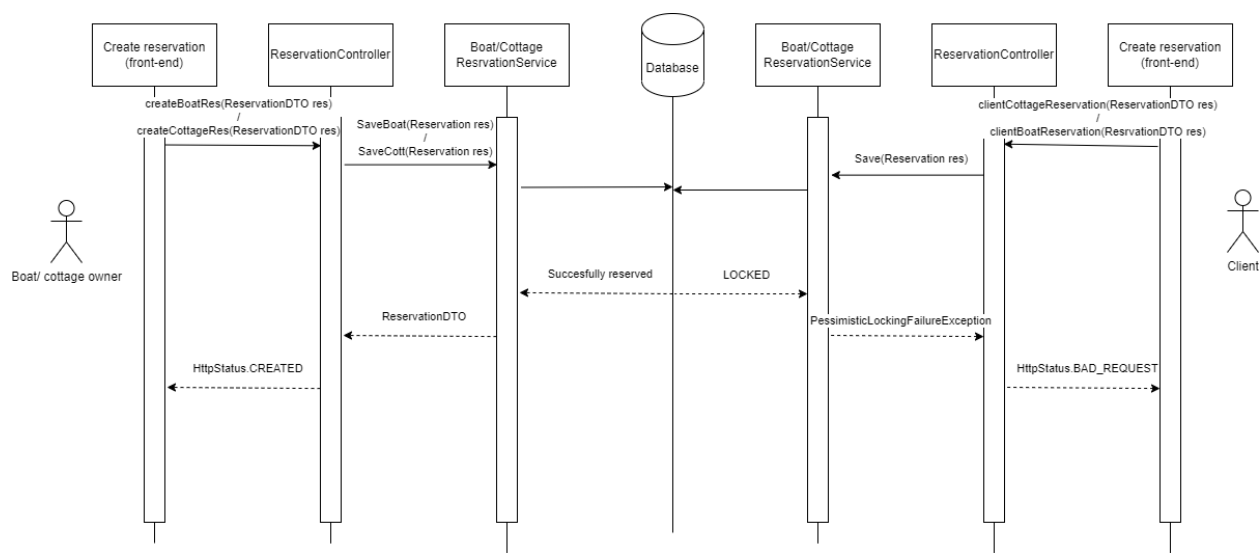


## 4.4. Konkurentni pristup resursima u bazi

*Konfliktna situacija 1:* Vlasnik vikendice/broda ne može napraviti rezervaciju istovremeno kada i drugi klijent.

Na slici ispod nalazi se dijagram sekvenci ovog problema.

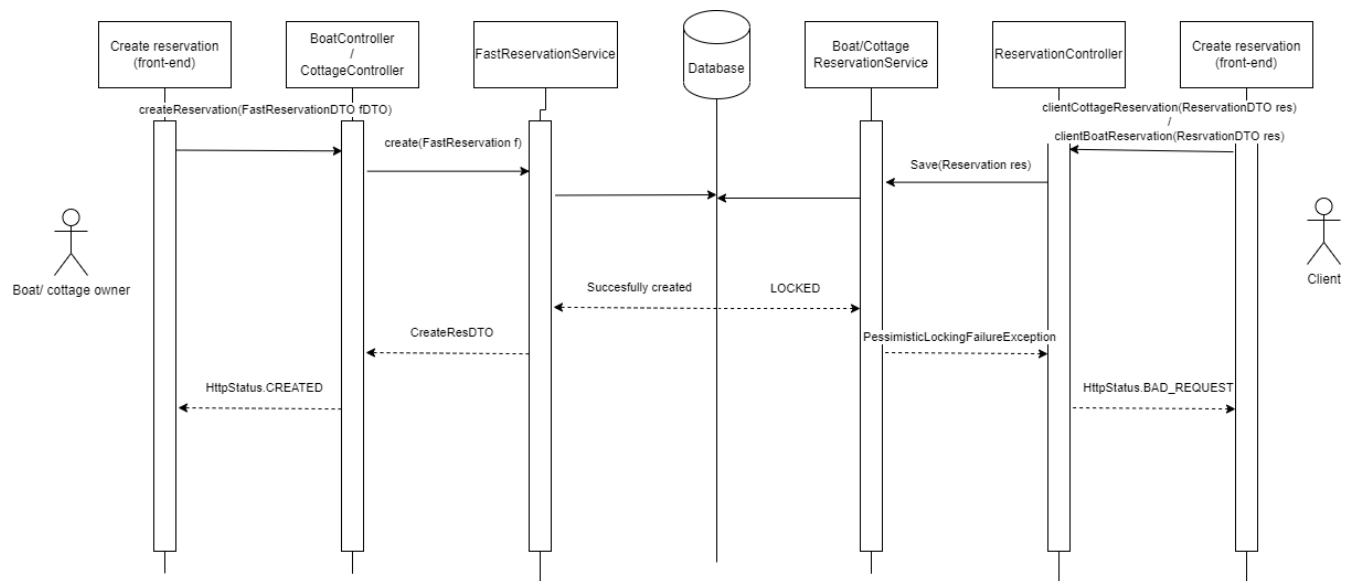


Predlog rešenja: U klasi kontrolera ReservationController nalaze se metode za kreiranje rezervacije od strane klijenta, i metode za kreiranje brze rezervacije (akcije) od strane vlasnika broda i vlasnika vikendice. Te metode pozivaju service BoatReservationService ili CottageReservationService, koje dalje pozivaju repozitorijume BoatRepository ili CottageRepository. Rukovanje izuzetkom u slučaju neuspelosti kreiranja rezervacije odrađeno je u klasi ReservationController. Metoda Save ima anotaciju @Transactional i poziva metodu repozitorijuma findLockedById(Long id). Isto tako, metode SaveBoat i SaveCottage proveravaju zauzetost resursa. U repozitorijumu se izvršava pesimističko zaključavanje. Tip zaključavanja koji se koristi je *PESSIMISTIC\_WRITE*.

Ovako je onemogućeno istovremeno rezervisanje istog broda ili iste vikendice od strane vlasnika i klijenta, međutim loša strana je to što će zakazivanje biti onemogućeno čak i ako se pokušavaju rezervisati različiti datumi, a i dalje isti entitet.

**Konfliktna situacija 2:** Vlasnik vikendice/broda ne može da kreira brzu rezervaciju (akciju) u isto vreme kada i klijent vrši običnu rezervaciju.

Na slici ispod nalazi se dijagram sekvenci ovog problema.

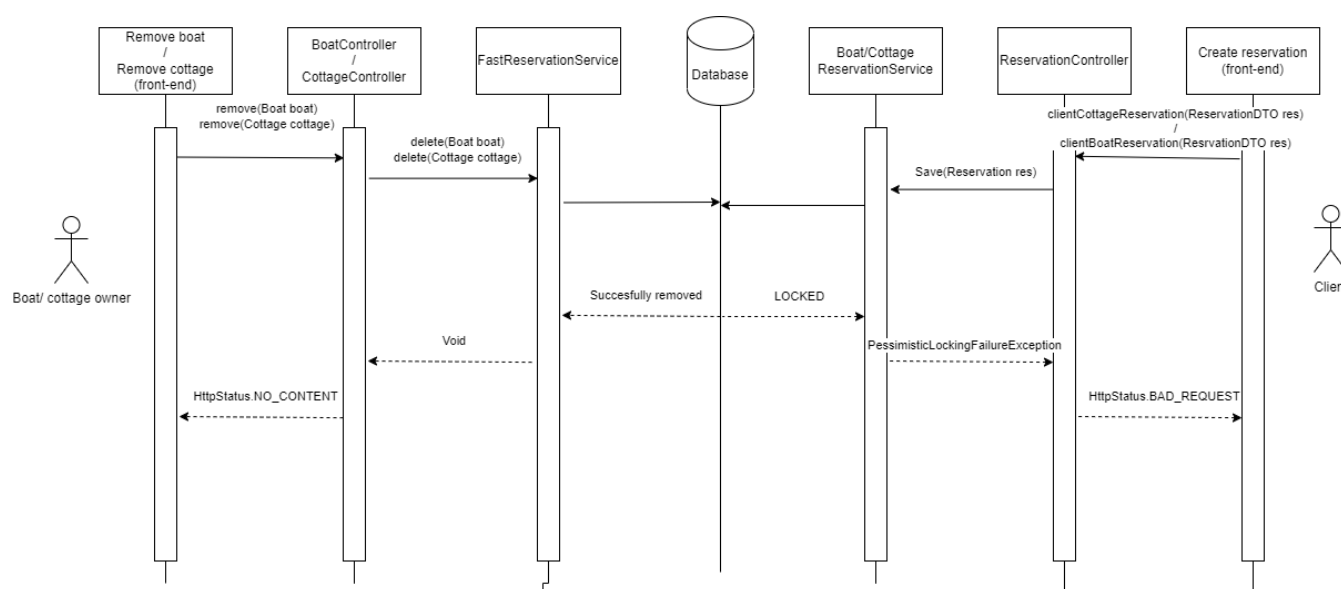


**Predlog rešenja:** U klasama kontrolera `ReservationController`, `BoatController` i `CottageController` nalaze se metode za kreiranje rezervacije od klijenta, kao i metode za uklanjanje broda ili vikendice od strane njihovih vlasnika. Te metode pozivaju servise `BoatService` i `CottageService`, koje dalje pozivaju repozitorijume `BoatRepository` ili `CottageRepository`. Rukovanje izuzetkom u slučaju neuspelosti kreiranja rezervacije odrađeno je u klasama `ReservationController`, `BoatController` i `CottageController`. Metoda `Save` ima anotaciju `@Transactional` (kao i metode `create (FastReservationCott fastReservation)` i `create (FastReservation fastReservation)`) i poziva metodu repozitorijuma `findLockedById(Long id)`. U njemu se izvršava pesimističko zaključavanje. Tip zaključavanja koji se koristi je `PESSIMISTIC_WRITE`.

Kao i u prvom primeru, onemogućeno je istovremeno zeuzeće istog broda ili iste vikendice, ali takođe ukoliko pokušaju da rezervišu različite termine, resursi će svakako biti zaključan.

**Konfliktna situacija 3:** Klijent ne može da rezerviše vikendicu ili brod u isto vreme kada vlasnik vikendice ili broda tu vikendicu obriše iz ponude.

Na slici ispod nalazi se dijagram sekvenci ovog problema.



**Predlog rešenja:** U klasama kontrolera **ReservationController**, **BoatController** i **CottageController** nalaze se metode za kreiranje rezervacije od strane klijenta. Te metode pozivaju service **FastReservationService** ili **FastReservationCottService**, koje dalje pozivaju repozitorijume **BoatRepository** ili **CottageRepository**. Rukovanje izuzetkom u slučaju neuspelosti kreiranja rezervacije odrađeno je u klasama **ReservationController**, **BoatController** i **CottageController**. Metoda `Save` ima anotaciju `@Transactional` i poziva metodu repozitorijuma `findLockedById(Long id)`. U njemu se izvršava pesimističko zaključavanje. Tip zaključavanja koji se koristi je `PESSIMISTIC_WRITE`. Takođe metode `delete(Boat boat)` i `delete(Cottage cottage)` imaju anotaciju `@Transactional`.

Tako je onemogućeno istovremeno brisanje broda ili vikendice i njihovo rezervisanje od strane klijenta.