

# Konkurentni pristup bazi podataka

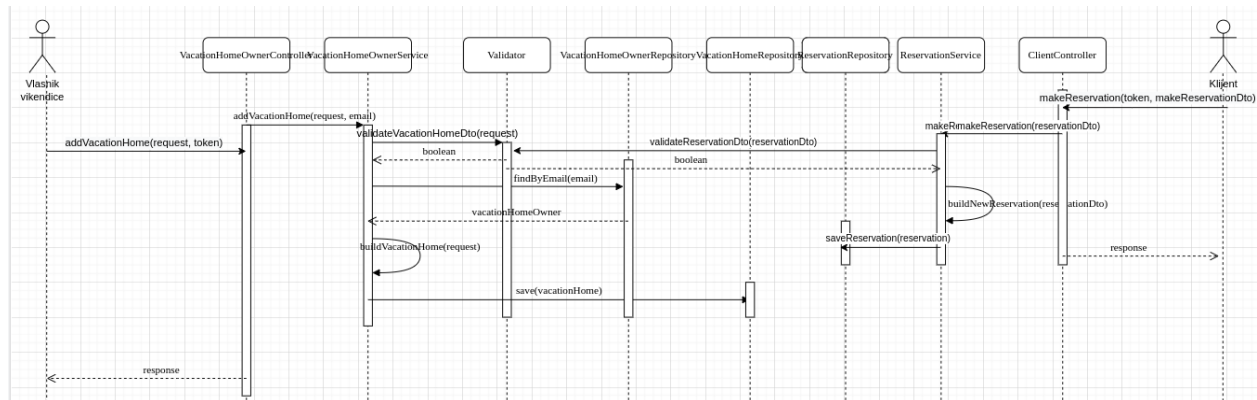
## Konfliktne situacije

Student: Vladan Mikić  
Broj indeksa: SW76/2019

### Situacija 1:

Vlasnik vikendice ili vlasnik broda pokušava da napravi rezervaciju za nekog klijenta dok u isto vreme neki drugi klijent pokušava da rezerviše tu uslugu. Posledica ove situacije je da se kreiraju dve rezervacije u istom terminu što nije ispravno.

**Tok akcije:** Vlasnik vikendice ili vlasnik broda pritiska dugme za dodavanje rezervacije u isto vreme kad neki drugi klijent pritisne dugme za rezervaciju tog entiteta. Serveru se šalju oba zahteva i ako prođu validaciju obe rezervacije se kreiraju i čuvaju u bazu.

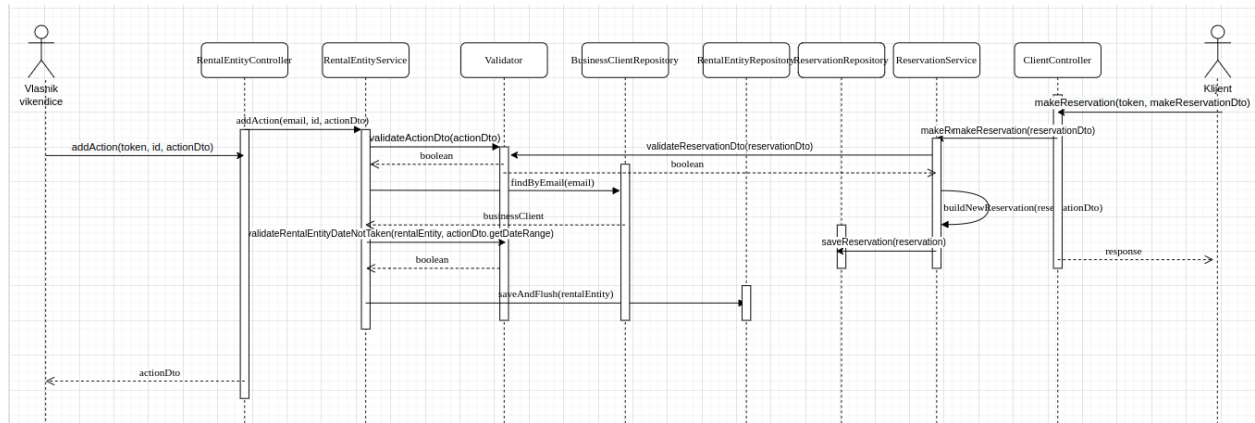


**Rešenje:** Problem je rešen upotrebom optimističkog zaključavanja. U klasi *RentalEntity*, koja je bazna klasa za sve entitete koji se mogu rezervisati, dodat je atribut *version* koji modeluje trenutnu verziju tog objekta. Prilikom kreiranja rezervacije doći će do promene tog objekta i samim tim i promene verzije, što će izazvati optimističko zaključavanje tog objekta. Nakon zaključavanja drugi zahtev za rezervisanje entiteta neće proći jer pokušava da izmeni staru verziju objekta. To izaziva *OptimisticLockingFailureException* koji biva uhvaćen u kontroleru.

### Situacija 2:

Vlasnik vikendice ili vlasnik broda pokušava da napravi akciju za neku uslugu dok u isto vreme neki drugi klijent pokušava da rezerviše tu uslugu. Posledica ove situacije je da se kreira akcija u istom periodu i za istu uslugu koju je klijent već rezervisao..

**Tok akcije:** Vlasnik vikendice ili vlasnik broda pritiska dugme za dodavanje akcije u isto vreme kad neki drugi klijent pritisne dugme za rezervaciju tog entiteta. Serveru se šalju oba zahteva i ako prođu validaciju kreiraju se i čuvaju u bazu.

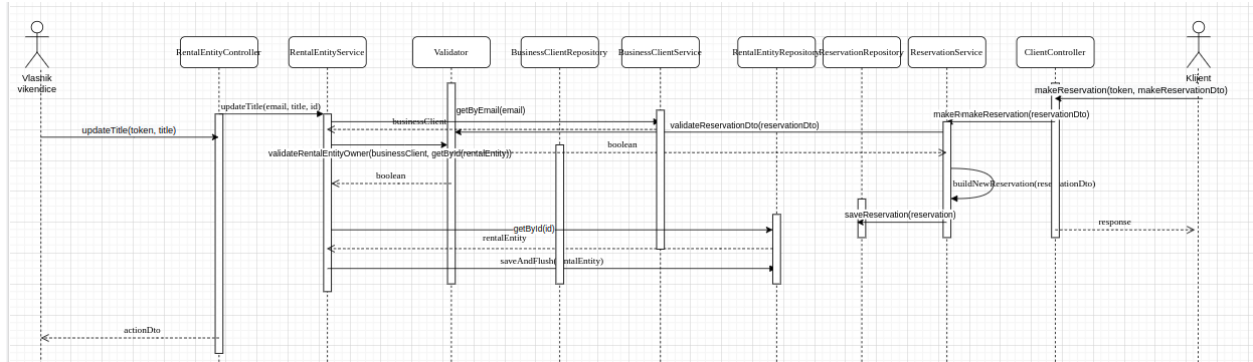


**Rešenje:** Problem je rešen upotrebom optimističkog zaključavanja. U klasi *RentalEntity*, koja je bazna klasa za sve entitete koji se mogu rezervisati, dodat je atribut *version* koji modeluje trenutnu verziju tog objekta. Prilikom kreiranja rezervacije ili akcije doći će do promene tog objekta i samim tim i promene verzije, što će izazvati optimističko zaključavanje tog objekta. Nakon zaključavanja drugi zahtev za rezervisanje entiteta ili pravljenje akcije neće proći jer pokušava da izmeni staru verziju objekta. To izaziva *OptimisticLockingFailureException* koji biva uhvaćen u kontroleru.

### Situacija 3:

Vlasnik vikendice ili vlasnik broda pokuša da izmeni ili obriše uslugu dok u isto vreme neki drugi klijent pokuša da rezerviše tu uslugu. Posledica ove situacije je da klijent rezerviše uslugu sa neočekivanim promenama.

**Tok akcije:** Vlasnik vikendice ili vlasnik broda pritiska dugme za čuvanje promene ili brisanje usluge u isto vreme kad neki drugi klijent pritisne dugme za rezervaciju tog entiteta. Serveru se šalju oba zahteva i ako prođu validaciju kreira se rezervacija sa neočekivanim parametrima jer je ta usluga izmenjena.



**Rešenje:** Problem je rešen upotrebom optimističkog zaključavanja. U klasi *RentalEntity*, koja je bazna klasa za sve entitete koji se mogu rezervisati, dodat je atribut *version* koji modeluje trenutnu verziju tog objekta. Prilikom promene ili brisanja tog objekta doći će do promene verzije, što će izazvati optimističko zaključavanje tog objekta. Nakon zaključavanja zahtev za rezervisanje entiteta neće proći jer pokušava da izmeni staru verziju objekta. To izaziva *OptimisticLockingFailureException* koji biva uhvaćen u kontroleru.