

Konfliktne situacije – student 1 – Mihajlo Mitić RA145/2018

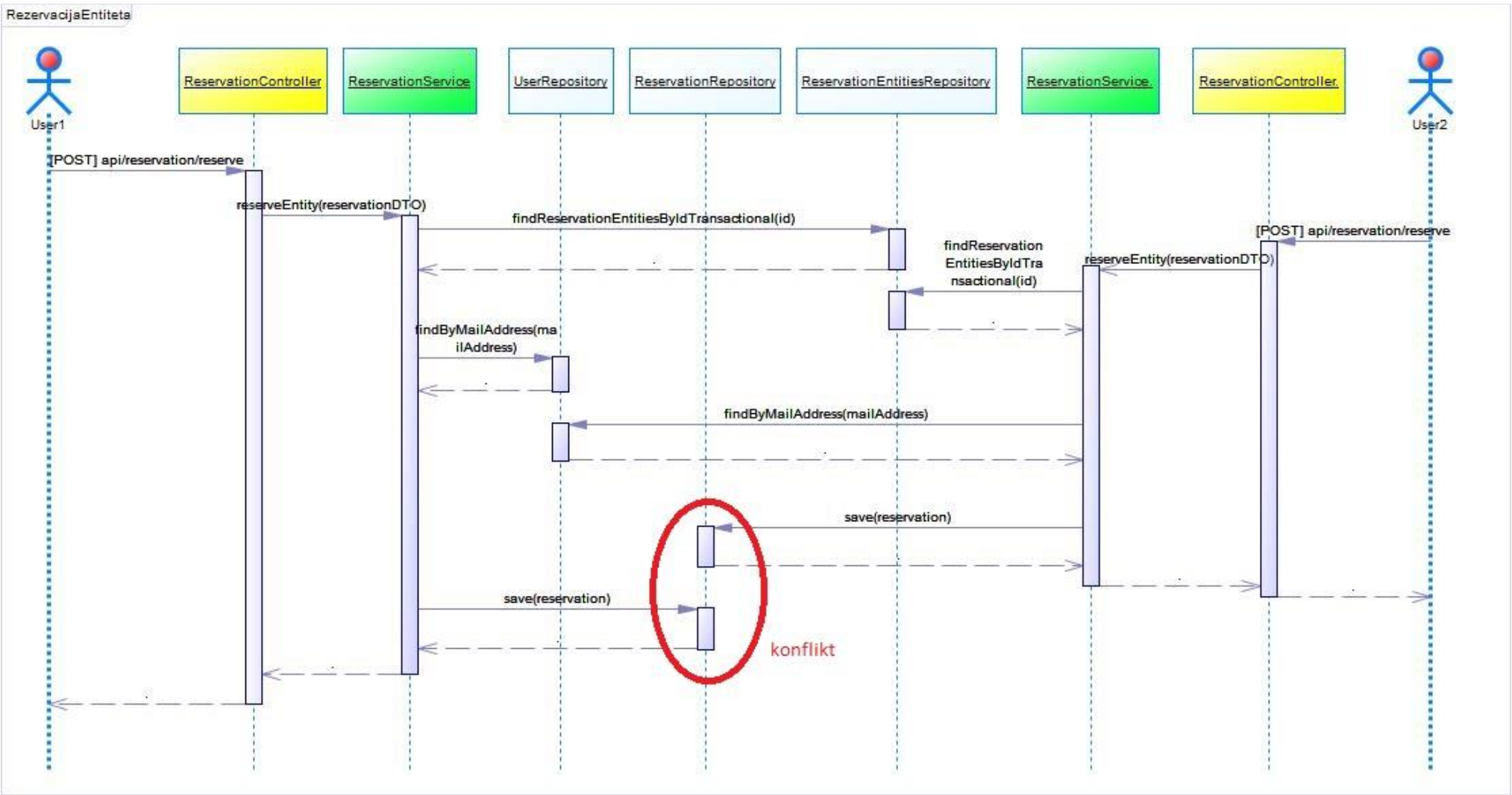
1. Više istovremenih klijenata ne mogu da naprave rezervaciju istog entiteta u isto (ili preklapajuće) vreme

Opis konfliktne situacije:

Kada se korisnik uloguje na sistem, on ima mogućnost rezervisanja entiteta nakon pretrage istih. Problem nastaje kada 2 korisnika istovremeno pokušaju rezervaciju istog entiteta. Oba korisnika će dobiti informaciju da je željeni entitet dostupan. I nakon potvrde rezervacije, u bazi će se sačuvati obe rezervacije klijenata, istog entiteta. Problem je delimično rešen implementacijom, tako što pre poziva samog čuvanja rezervacije, proveravamo da li je slučajno neko već zauzeo. Međutim ovo rešenje nije rešilo problem istovremenog rezervisanja.

Slika:

Na slici je prikazano rešena konfliktna situacija.



Rešenje:

Ovaj problem je rešen zabranjivanjem pristupa jednom korisniku, uz pomoć pesimističkog zaključavanja rezervacionog entiteta (vikendice, broda, instruktora pecanja).

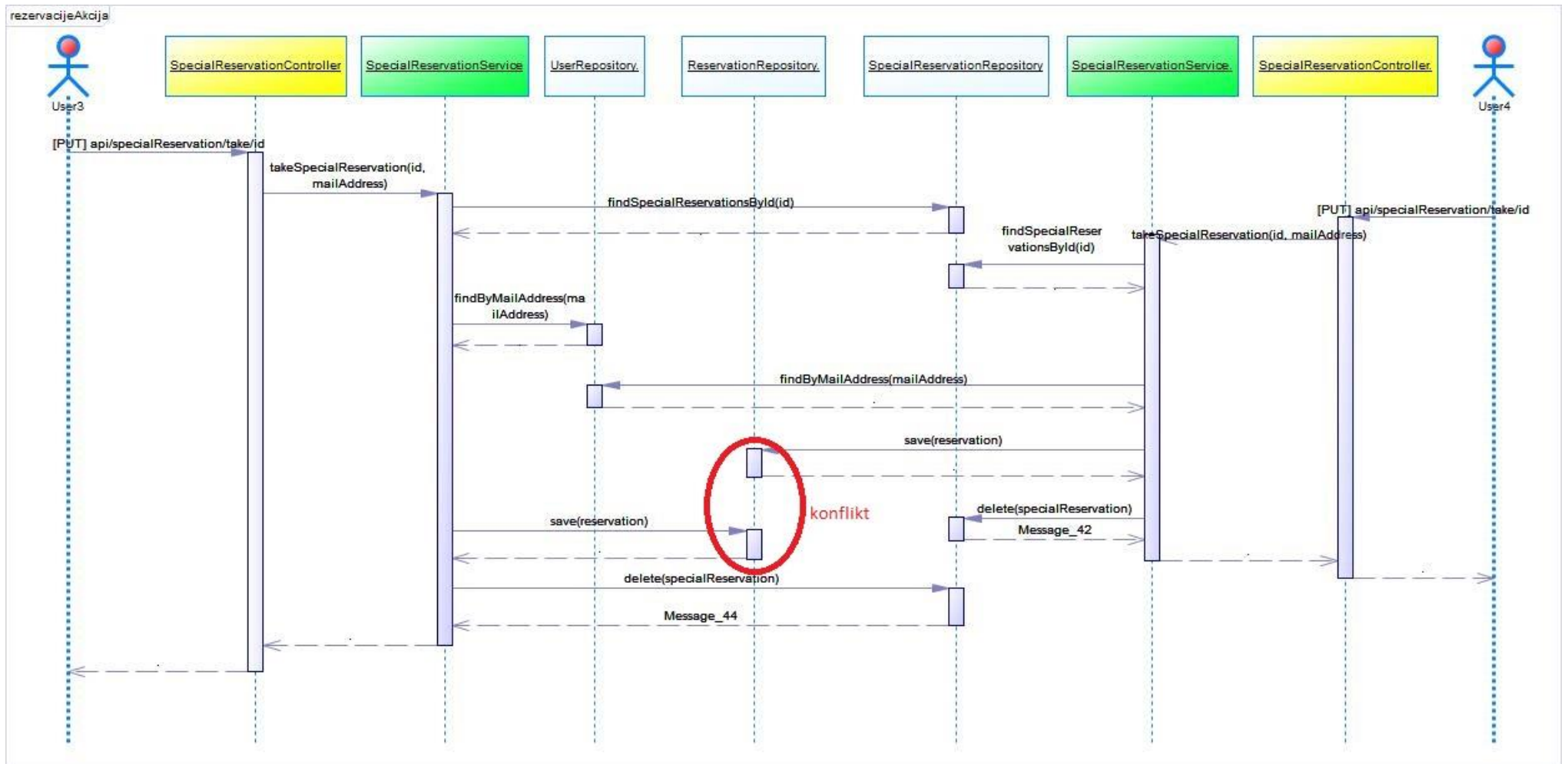
Prvobitna metoda `findReservationEntitiesById` je zamenjena metodom `findReservationEntitiesByIdTransactional` u kojoj je postavljena anotacija `@Lock(LockModeType.PESSIMISTIC_WRITE)` za pesimistično zaključavanje.

2. Više istovremenih klijenata ne mogu da naprave rezervaciju istog entiteta na akciju u isto vreme

Opis konfliktne situacije:

Korisnik ima mogućnost rezervisanja entiteta na akciji. U okviru entiteta se prikazuju dostupne akcije. Klikom na odgovarajući link, korisnik potvrđuje svoju rezervaciju. Problem nastaje kada 2 korisnika u isto vreme pokušaju da rezervišu istu akciju. U bazi će doći do nekonzistentnog stanja, jer će biti sačuvana ista akcija 2 puta sa podacima za 2 različita korisnika.

Slika:



Rešenje:

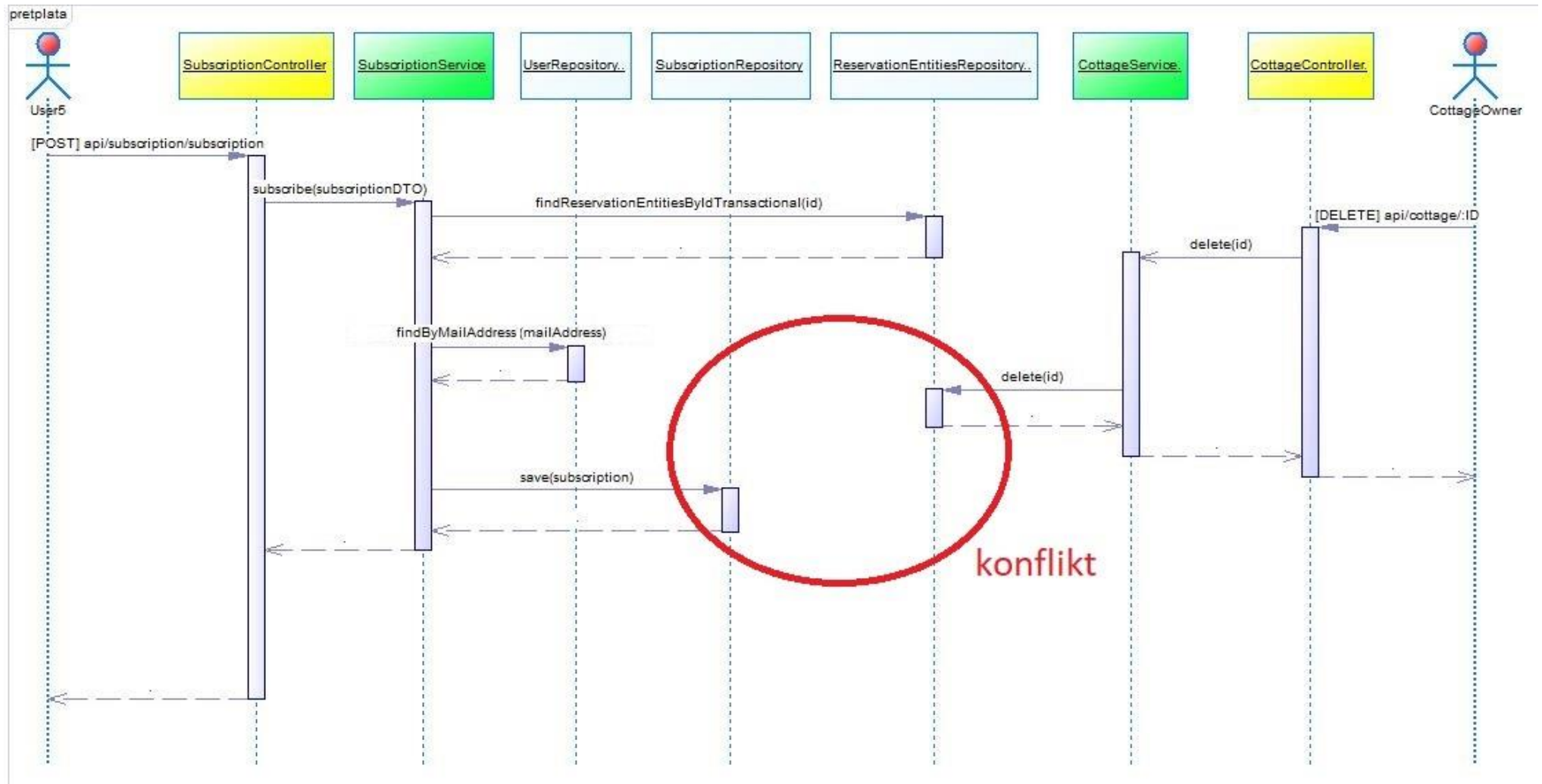
Problem je rešen pesimističkim zaključavanjem akcije (odnosno specijalne rezervacije u kodu). Metoda `findSpecialReservationsById` je anotirana sa `@Lock(lockMode = LockModeType.PESSIMISTIC_WRITE)`. Ovo znači da ni jedna druga transakcija neće moći da pristupi željenoj akciji, dok se transakcija, koja je prvobitno zauzela akciju, ne završi.

3. Klijent se ne može pretplatiti na entitet, ukoliko u isto vreme bude obrisani entitet od strane vlasnika entiteta

Opis konfliktne situacije:

Korisnik se može pretplatiti na bilo koji entitet klikom na određeni link. Konfliktna situacija nastaje kada u isto vreme korisnik pokuša da se pretplati na entitet, a vlasnik entiteta (u ovom slučaju vikendice) u isto vreme obriše isti entitet.

Slika:



Rešenje:

Ovaj problem je takođe rešen pesimističkim zaključavanjem. Ukoliko korisnik pokuša da izvrši pretplatu, zaključaće se entitet za rezervacije. Metoda `findReservationEntitiesByIdTransactional` je anotirana sa `@Lock(LockModeType.PESSIMISTIC_WRITE)` što omogućava zaključavanje entiteta, i zabranjuje vlasniku vikendice da obriše taj entitet dok se ne završi pretplata.

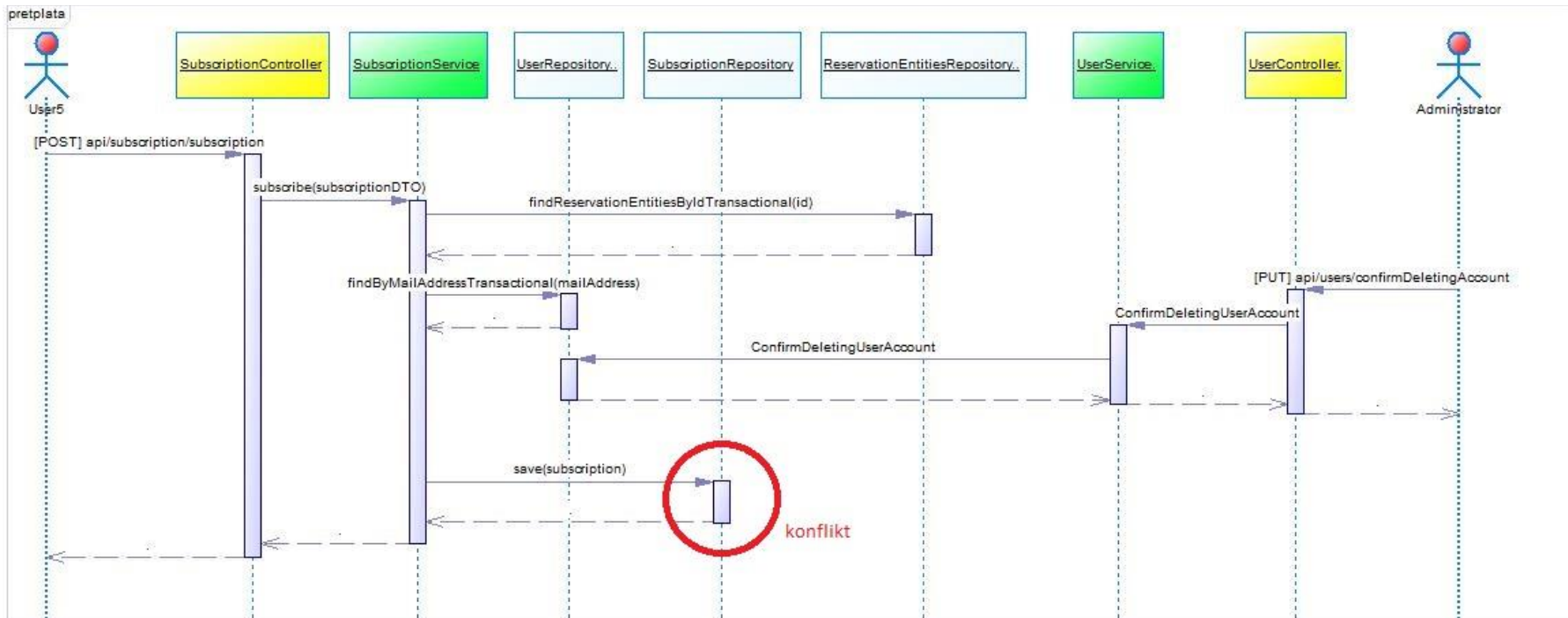
4. Klijent se ne može pretplatiti na entitet, ukoliko u isto vreme bude obrisao njegov nalog od strane administratora

-- napomena: funkcionalnosti administratora još uvek nisu implementirane, ali je u nastavku opisano kao da je implementirano.

Opis konfliktne situacije:

Korisnik se može pretplatiti na bilo koji entitet klikom na određeni link. Konfliktna situacija nastaje kada u isto vreme korisnik pokuša da se pretplati na entitet, a administrator prihvati zahtev za brisanje naloga (samim tim se briše nalog korisnika). U bazi za pretplate će se sačuvati id nepostojećeg korisnika.

Slika:



Rešenje:

Ovaj problem je takođe rešen pesimističkim zaključavanjem. Ukoliko korisnik pokuša da izvrši pretplatu, zaključaće se njegov nalog. Metoda `findByMailAddressTransactional` je anotirana sa `@Lock(LockModeType.PESSIMISTIC_WRITE)` što omogućava zaključavanje naloga korisnika, i zabranjuje administratoru da prihvati zahtev za brisanje dok korisnik ne završi pretplatu.