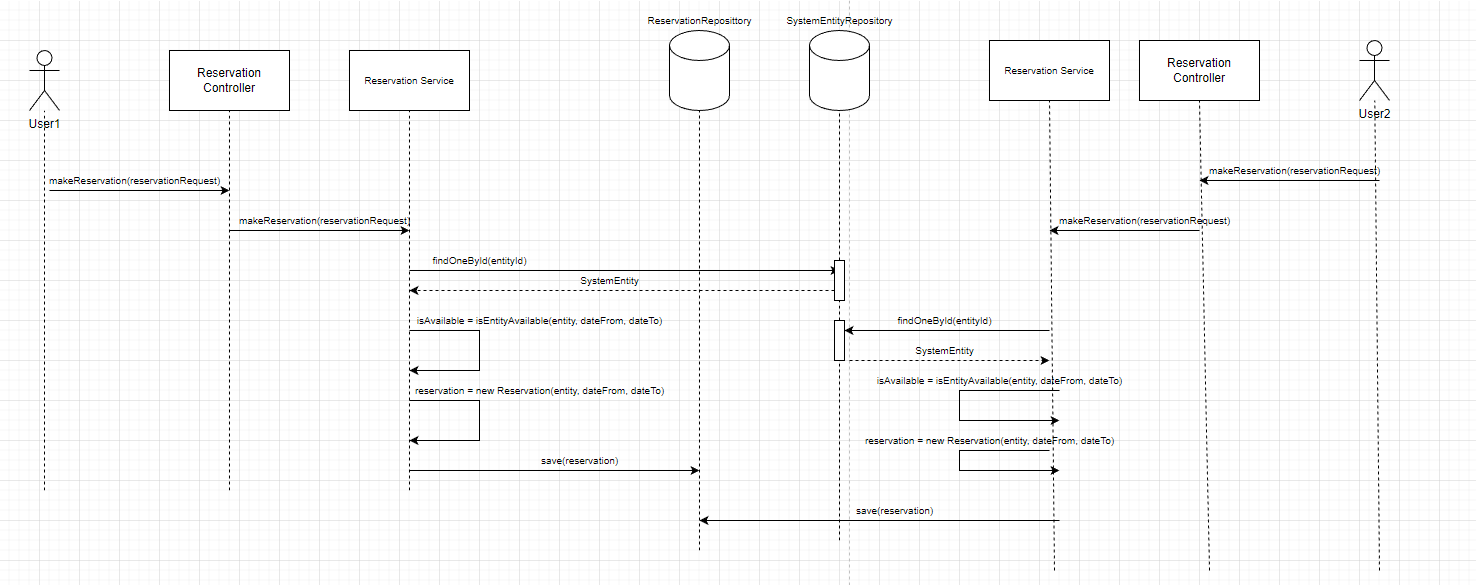
## Konkurentni pristup resursima u bazi

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

**Opis problema:**

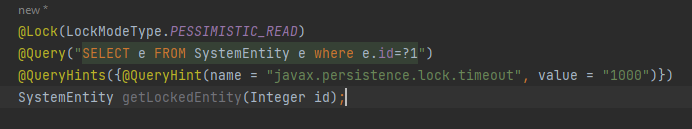
Mogući problem jeste situacija u kojoj više korisnika želi da rezerviše isti entitet u isto ili preklapajuće vreme. Ukoliko bi dva ili više korisnika u isto vreme pokušali da rezervišu određeni entitet, postojala bi mogućnost da oba korisnika dobiju povratnu informaciju da je entitet slobodan u željenom vremenu, te bi postojala opasnost od dualne zauzetosti jednog entiteta. Dijagram opisane situacije može se videti na Slika 1.



Slika 1 Rezervacija entiteta – tok zahteva

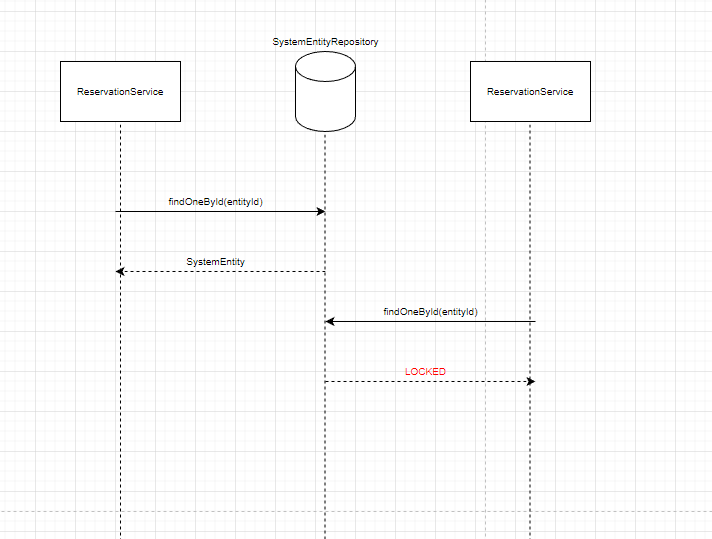
**Rešenje:**

Problem se može rešiti pomoću pesimističkog ili optimističkog zaključavanja baze. Za ovaj primer odlučio sam se za pesimističko zaključavanje baze na find metode SystemEntityRepository. Dodatno, u klasi ReservationService, metoda makeReservation je označena @Transactional anotacijom, Slika 2.



Slika 2 Pristupanje entitetu u bazi

Ovime dobijamo željeni efekat gde korisnik koji kasnije pristupi entitetu koji želi da rezerviše, mora da sačeka da drugi korisnik završi svoju „transakciju“, Slika 3.

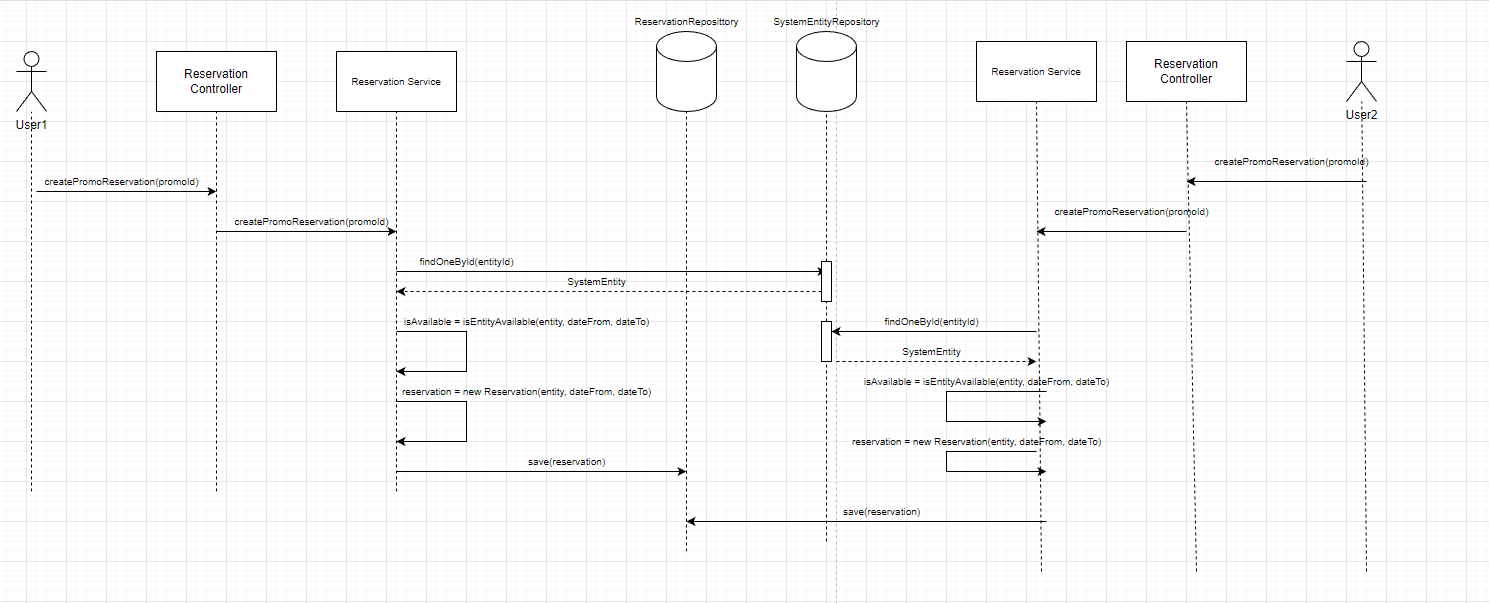


Slika 3 Pristupanje entitetu - tok zahteva

## Konflikt 2 - Više istovremenih klijenata ne mogu da naprave rezervaciju istog entiteta na akciji u isto vreme

**Opis problema:**

Mogući problem jeste situacija u kojoj više korisnika želi da iskoristi promociju istog entiteta. Ukoliko bi dva ili više korisnika u isto vreme pokušali da iskoriste istu promociju, postojala bi mogućnost da oba korisnika dobiju povratnu informaciju da je entitet slobodan u željenom vremenu, te bi postojala opasnost od dualne zauzetosti jednog entiteta. Dijagram opisane situacije može se videti na Slika 4.



Slika 4 Rezervacija akcije entiteta – tok zahteva

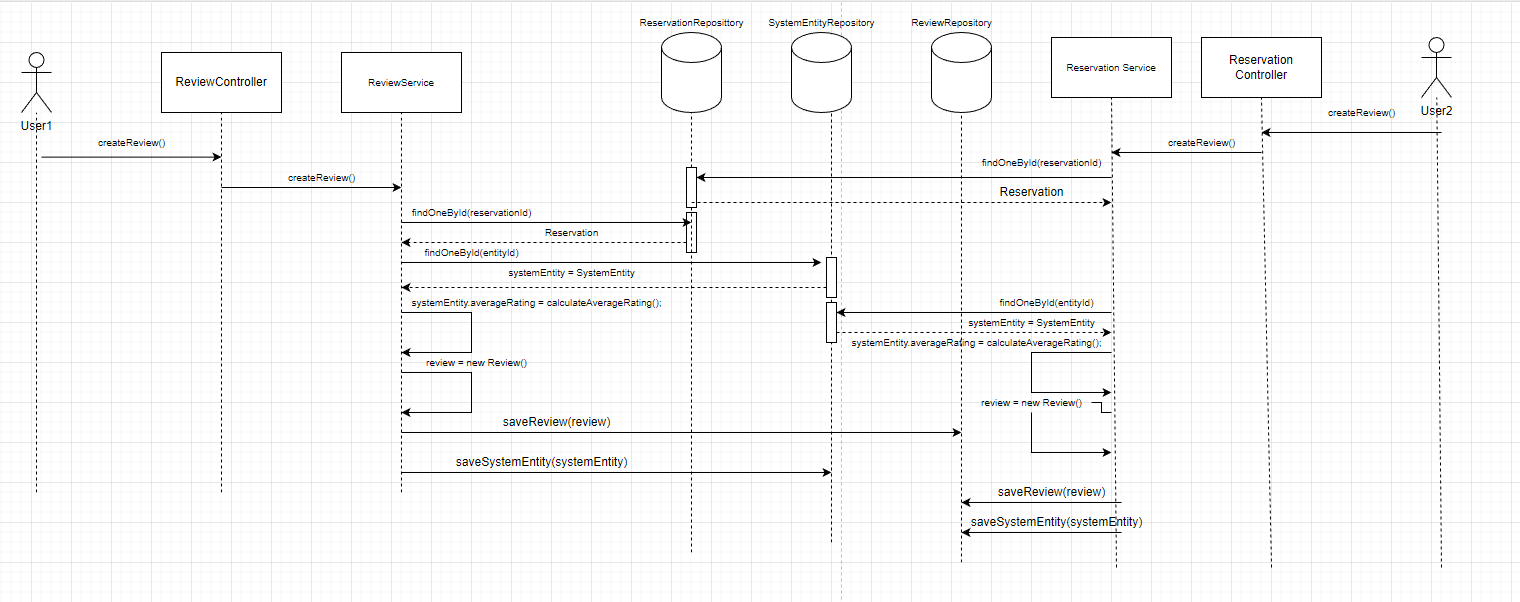
**Rešenje:**

Problem se može rešiti pomoću pesimističkog ili optimističkog zaključavanja baze. Za ovaj primer odlučio sam se za pesimističko zaključavanje baze na find metode SystemEntityRepository. Dodatno, u klasi ReservationService, metoda createPromoReservation je označena @Transactional anotacijom. Rešenje pristupa bazi je već rešeno u prvom problemu, Slika 2. Željeno rešenje je dijagramom opisano na Slika 3.

Napomena da se ovakvim rešavanjem rešava i problem preplitanja akcija sa običnim rezervacijama.

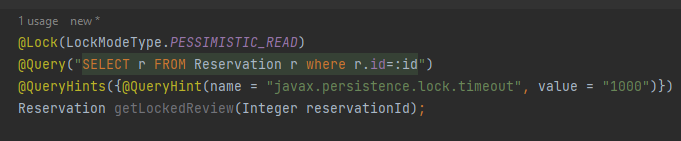
## Konflikt 3 - Više klijenata ne mogu istovremeno oceniti isti entitet u isto vreme

**Opis problema:**

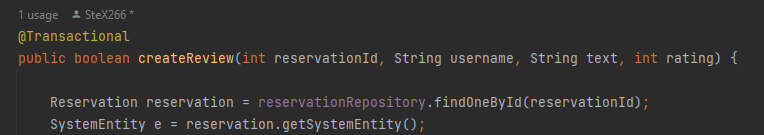
Mogući problem jeste situacija u kojoj više korisnika želi da oceni isti entitet u isto vreme. Ukoliko bi dva ili više korisnika u isto vreme pokušali da ocene isti entitet, postojala bi opasnost da se loše izračuna nova prosečna ocena entiteta gde bi se računala samo jedna od N ocena.

Slika 5 Ocenjivanje entiteta - tok zahteva

**Rešenje:**

Za rešenje ovog problema sam koristio pesimističko zaključavanje resursa. Sagledao sam model kreiranja ocena entiteta i svaki entitet je vezan za rezervaciju koja se prethodno očitava iz baze. Pri svakom pozivu servisne metode za ocenjivanje entiteta vrši se zaključavanje rezervacija, sve dok se ne izvrši logika za preračunavanje srednje ocene entiteta i sačuvaju promene. Dodattno je metoda za kreiranje ocene u servisu anotirana sa @Transactional.

Slika 6 Pristupanje rezervacijama



Slika 7 Servisna metoda za kreiranje ocene