

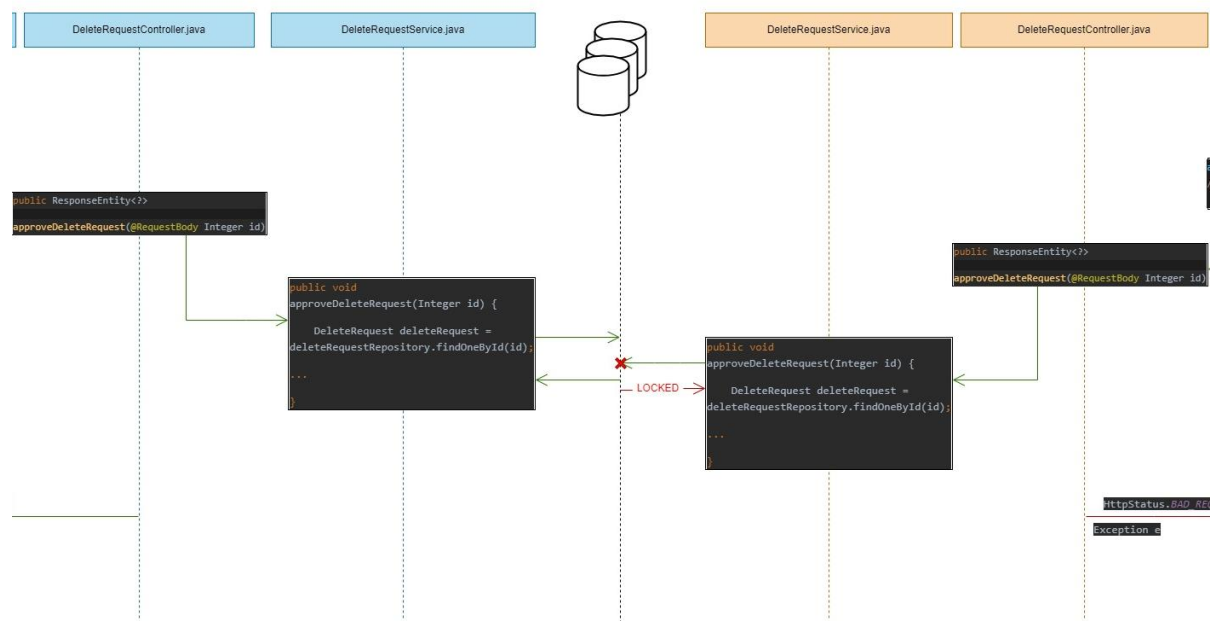
## 4.4 Konkurentan pristup resursima u bazi

Rešene konfliktne situacije:

Na jedan zahtev za brisanje naloga može da odgovori samo jedan administrator sistema

Opis situacije: Administratori odgovaraju na zahtev korisnika za brisanje naloga. Administrator može da potvrdi ili odbije zahtev. U oba slučaja, šalje se email korisniku o administratorovoj odluci. Potrebno je obezbediti da dva administratora ne mogu u isto vreme da odgovore na jedan isti zahtev kako ne bi došlo do različitog rukovanja tj. različitog odgovora na zahtev od strane njih.

Isečak crteža konfliktne situacije (cela slika u folderu ./Transakcije/Student3/ Student3-1.jpg):



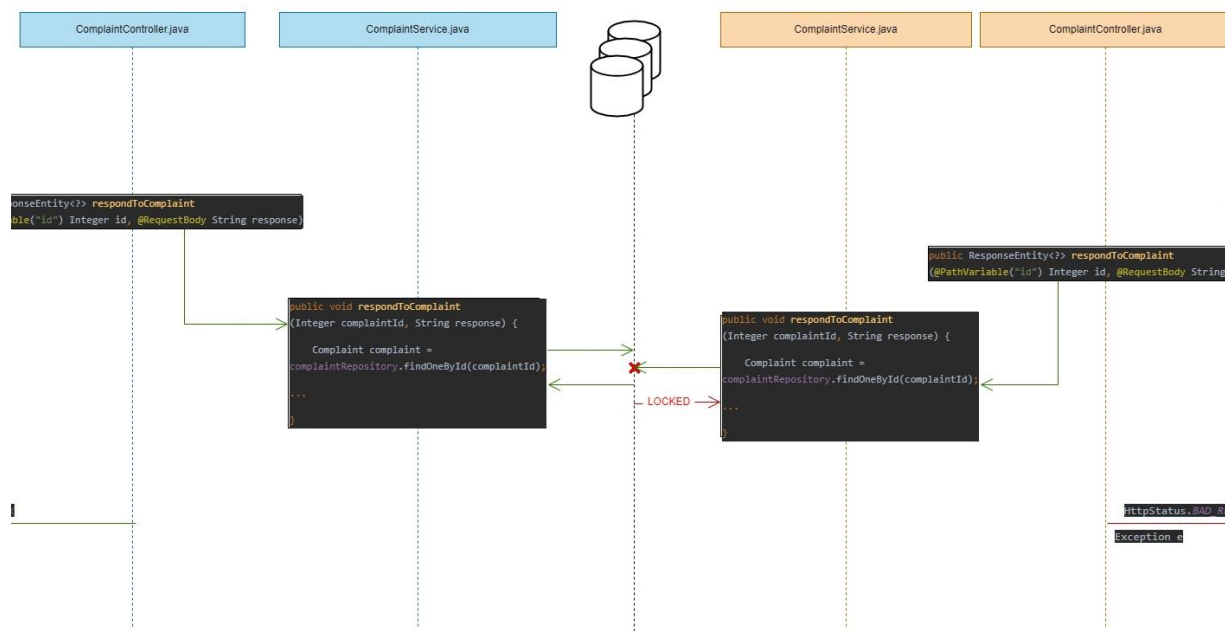
Rešenje: Postupak se nalazi u klasama *DeleteRequestController.java* (rukovanje izuzetkom), *DeleteRequestService.java* (poziva metodu `findOneById(Integer id)` repozitorijuma) i *DeleteRequestRepository.java* (gde je odrađeno pesimističko zaključavanje na nivou metode `findOneById(Integer id)`). Kako bi se izbegla konfliktna situacija, korišćeno je pesimističko zaključavanje jer se nakon odobravanja/odbijanja zahteva za brisanje entitet *DeleteRequest* odmah briše iz baze, pa je jedini način obezbeđivanje tog entiteta dok postoji, njegovo zaključavanje i zabrana da mu bilo ko drugi pristupi ako je zaključan od strane nekog drugog

klijenta. U tom slučaju, dogodiće se izuzetak i administrator koji je izazvao konfliktnu situaciju biva obavešten o tome.

### Na jednu žalbu može da odgovori samo jedan administrator sistema

**Opis situacije:** Administratori odgovaraju na žalbe korisnika na entitet ili vlasnika/instruktora. Nakon odgovora na žalbu, administratorov odgovor se šalje i klijentu i oglašivaču na email i žalba se briše iz sistema jer nije više relevantna. Potrebno je obezbediti da dva administratora ne mogu u isto vreme da odgovore na jednu žalbu kako ne bi došlo do različitog odgovora na žalbu od strane njih.

Isečak crteža konfliktne situacije (cela slika u folderu ./Transakcije/Student3/ Student3-2.jpg):



**Rešenje:** Postupak se nalazi u klasama `ComplaintController.java` (rukovanje izuzetkom), `ComplaintService.java` (poziva metodu `findOneById(Integer id)` repozitorijuma) i `ComplaintRepository.java` (gde je odrađeno pesimističko zaključavanje na nivou metode `findOneById(Integer id)`). Kako bi se izbegla konfliktna situacija, kao i u prethodnom primeru, korišćeno je pesimističko zaključavanje jer se nakon odgovora na žalbu entitet `Complaint` odmah briše iz baze, pa je jedini način obezbeđivanje tog entiteta dok postoji, njegovo zaključavanje i zabrana da mu bilo ko drugi pristupi ako je zaključan od strane nekog drugog klijenta. U tom slučaju, dogodiće se izuzetak i administrator koji je izazvao konfliktnu situaciju biva obavešten o tome.

**Opis situacije:** Administratori mogu da odobre ili odbiju reviziju korisnika na neki entitet. Nakon odobravanja se ažurira ocena entiteta. Potrebno je osigurati se da ne mogu dva administratora u isto vreme da odgovore na reviziju, da ne bi došlo do nerelevantnih ocena entiteta.

```

sequenceDiagram
    participant TC1 as RevisionController.java
    participant RS1 as RevisionService.java
    participant DB as Database
    participant TC2 as RevisionController.java
    participant RS2 as RevisionService.java

    TC1->>RS1: approveRevision(Integer id)
    RS1->>DB: findOneById(id)
    DB-->>RS1: 
    RS1-->>TC1: 
    TC2->>RS2: approveRevision(Integer id)
    RS2->>DB: findOneById(id)
    DB-->>RS2: 
    RS2-->>TC2: 
    Note over RS2: LOCKED
    TC2-->>TC2: HttpStatus.BAD_REQUEST
    
```

**Rešenje:** Postupak se nalazi u klasama *RevisionController.java* (rukovanje izuzetkom), *RevisionService.java* (poziva metodu `findOneById(Integer id)` repozitorijuma) i *RevisionRepository.java* (gde je odrađeno pesimističko zaključavanje na nivou metode `findOneById(Integer id)`). Kako bi se izbegla konfliktna situacija, kao i u prethodna dva primera, korišćeno je pesimističko zaključavanje jer se nakon revizije ocene entitet *Revision* odmah briše iz baze, pa je jedini način obezbeđivanje tog entiteta dok postoji, njegovo zaključavanje i zabrana da mu bilo ko drugi pristupi ako je zaključan od strane nekog drugog klijenta. U tom slučaju, dogodiće se izuzetak i administrator koji je izazvao konfliktnu situaciju biva obavešten o tome.