

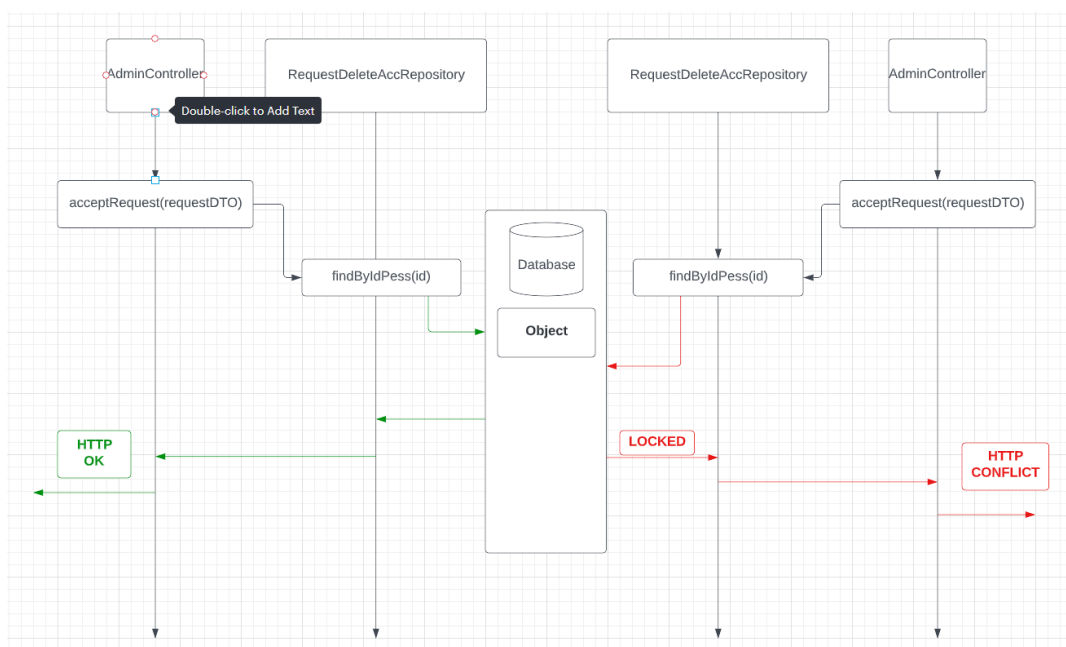
4.4 Konkurentan pristup resursima iz baze

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.

Dijagram 1

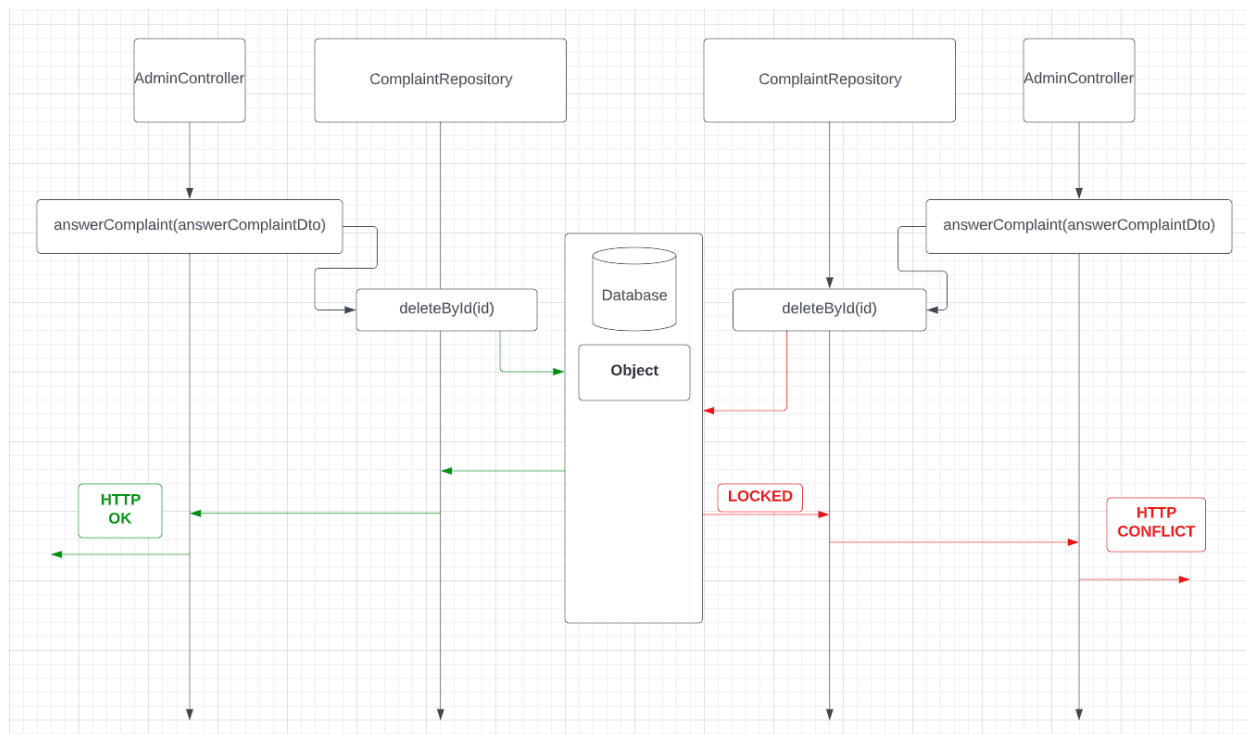


Rešenje: Postupak se nalazi u klasama *AdminController* (poziva metodu `findByIdPess(id)` repozitorijuma) i *RequestDeleteAccRepository.java* (gde je odrađeno pesimističko zaključavanje na nivou metode `findByIdPess(id)`). Kako bi se izbegla konfliktna situacija, korišćeno je pesimističko zaključavanje jer se nakon odobravanja/odbijanja zahteva za brisanje entitet *RequestDeleteAcc* 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.

Dijagram 2

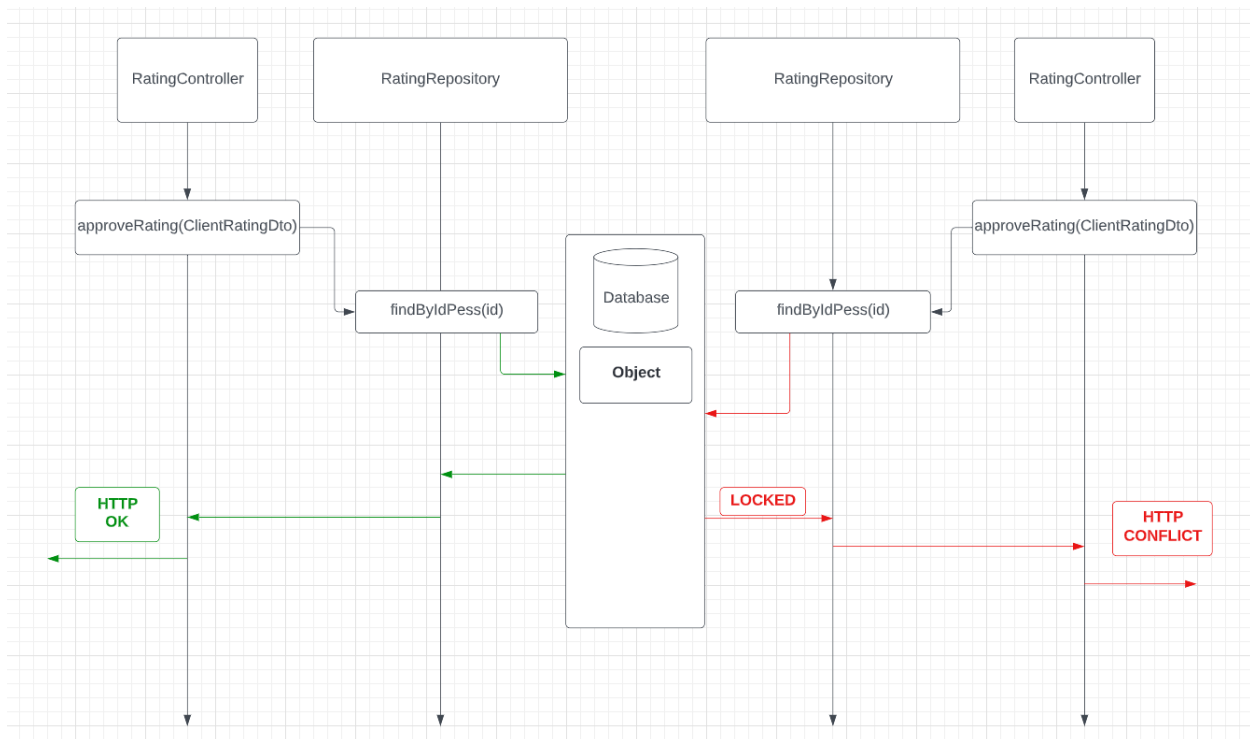


Rešenje: Postupak se nalazi u klasama *AdminController* (poziva metodu `deleteById(id)` repozitorijuma) i *ComplaintRepository* (gde je odrađeno pesimističko zaključavanje na nivou metode `deleteById(id)`). Kako bi se izbegla konfliktna situacija, kao i u prethodnom primeru, korišćeno je pesimističko 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.

Jednu ocenu može da odobri/odbije samo jedan administrator sistema

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

Dijagram 3



Rešenje: Postupak se nalazi u klasama *RatingController* (poziva metodu `findByIdPess(id)` repozitorijuma) i *RatingRepository.java* (gde je odrađeno pesimističko zaključavanje na nivou metode `findByIdPess(id)`). Kako bi se izbegla konfliktna situacija, kao i u prethodna dva primera, korišćeno je pesimističko zaključavanje jer se nakon revizije ocene entitet *Rating* 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.