

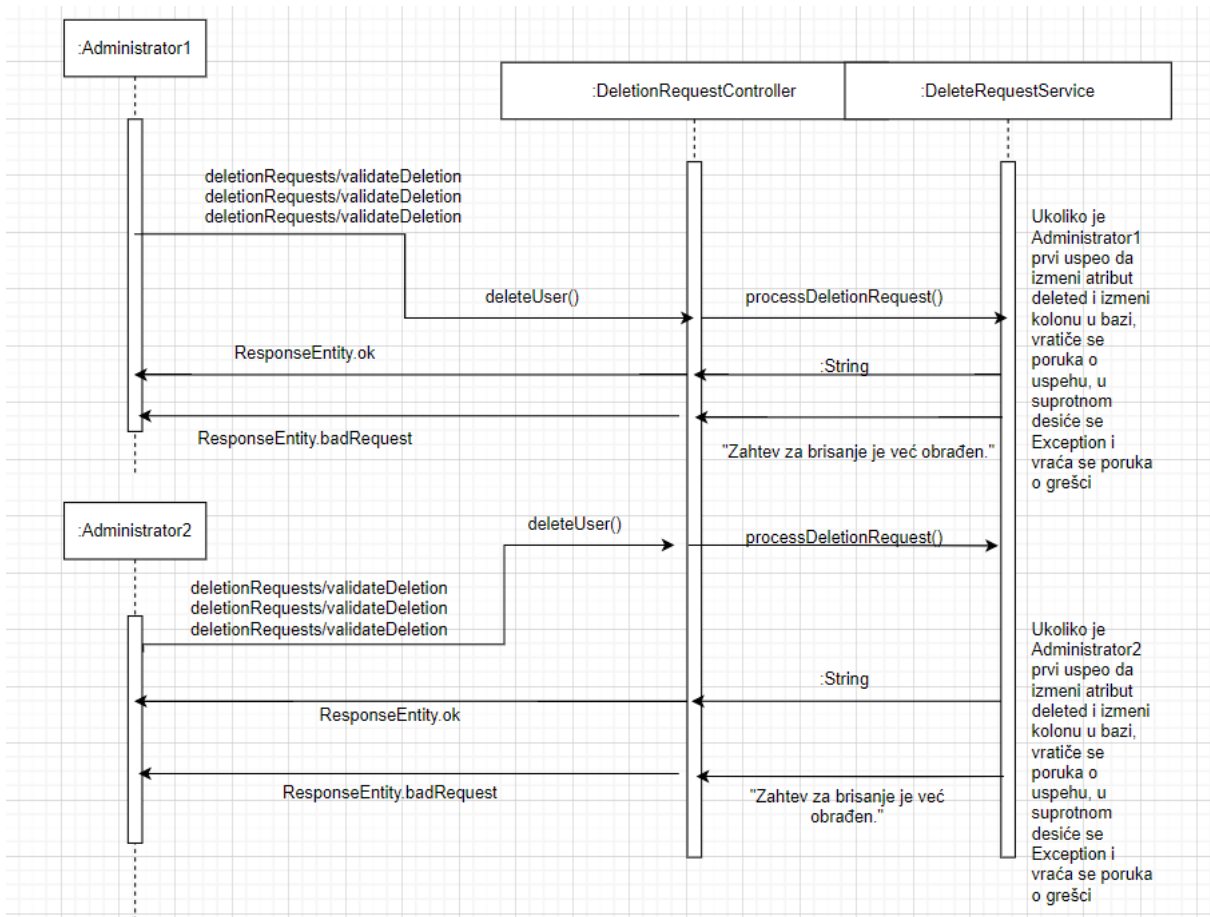
Student 3 - Dunja Stojanov SW 30/2019

1. Konflikt

a. Opis konfliktne situacije

Na jedan zahtev za brisanje naloga odgovara više administratora sistema

b. Tok zahteva



c. Rešenje

Za rešavanje ovog problema odabrano je optimističko zaključavanje. Razlog za to je što prilikom odgovaranja na zahtev za brisanje, zahtev je već kreiran i postoji u bazi. Ono što je potrebno odraditi jeste izmeniti kolonu *deleted* i iz tog razloga objektu *DeleteRequest* dodat je poseban atribut *Version* koji predstavlja mehanizam za rešavanje konflikata u konkurentnom pristupu entitetima baze.

Prilikom odgovaranja na zahtev za brisanje naloga, poziva se funkcija *processDeletionRequest*. Prvo se dobavi određeni zahtev i tom objektu se postavi atribut *deleted* na *true*, ove izmene potrebno je ponovo sačuvati u bazu podataka. Ažuriranje kolone u bazu obavlja se funkcijom *addDeleteRequest* i ona je transakciona. Ukoliko se atribut zahteva *Version* ne poklapa na odgovarajući način sa atributom *Version* zahteva koji pokušavamo trenutno sačuvati u bazu, desiće se

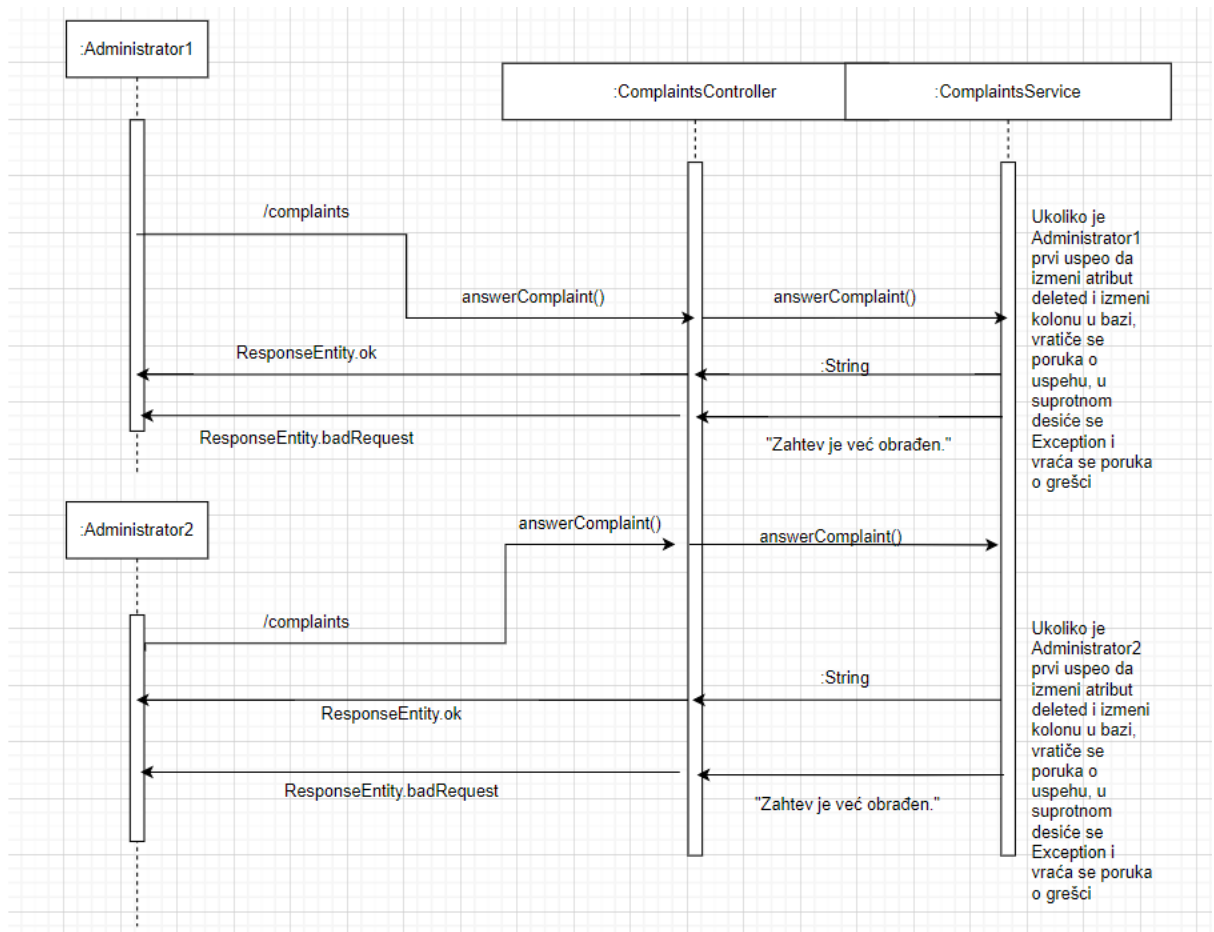
ObjectOptimisticLockingFailureException koji je potrebno uhvatiti i obavestiti administratora da je odabrani zahtev za brisanje naloga već obrađen.

2. Konflikt

d. Opis konfliktne situacije

Na jednu žalbu odgovora više administratora sistema

e. Tok zahteva



f. Rešenje

Za rešavanje ovog problema odabrano je optimističko zaključavanje. Razlog za to je što prilikom odgovaranja na žalbu, žalba je već kreirana i postoji u bazi. Ono što je potrebno odraditi jeste izmeniti kolonu *deleted* i iz tog razloga objektu *Complaint* dodat je poseban atribut *Version* koji predstavlja mehanizam za rešavanje konflikata u konkurentnom pristupu entitetima baze.

Prilikom odgovaranja na žalbe, poziva se funkcija *answerComplaint*. Prvo se dobavi određena žalba i tom objektu se postavi atribut *deleted* na *true*, ove izmene potrebno je ponovo sačuvati u bazu podataka. Ažuriranje kolone u bazu obavlja se funkcijom *saveComplaint* ona je transakciona. Ukoliko se atribut zahteva *Version* ne poklapa na odgovarajući način sa atributom *Version* zahteva koji pokušavamo trenutno sačuvati u

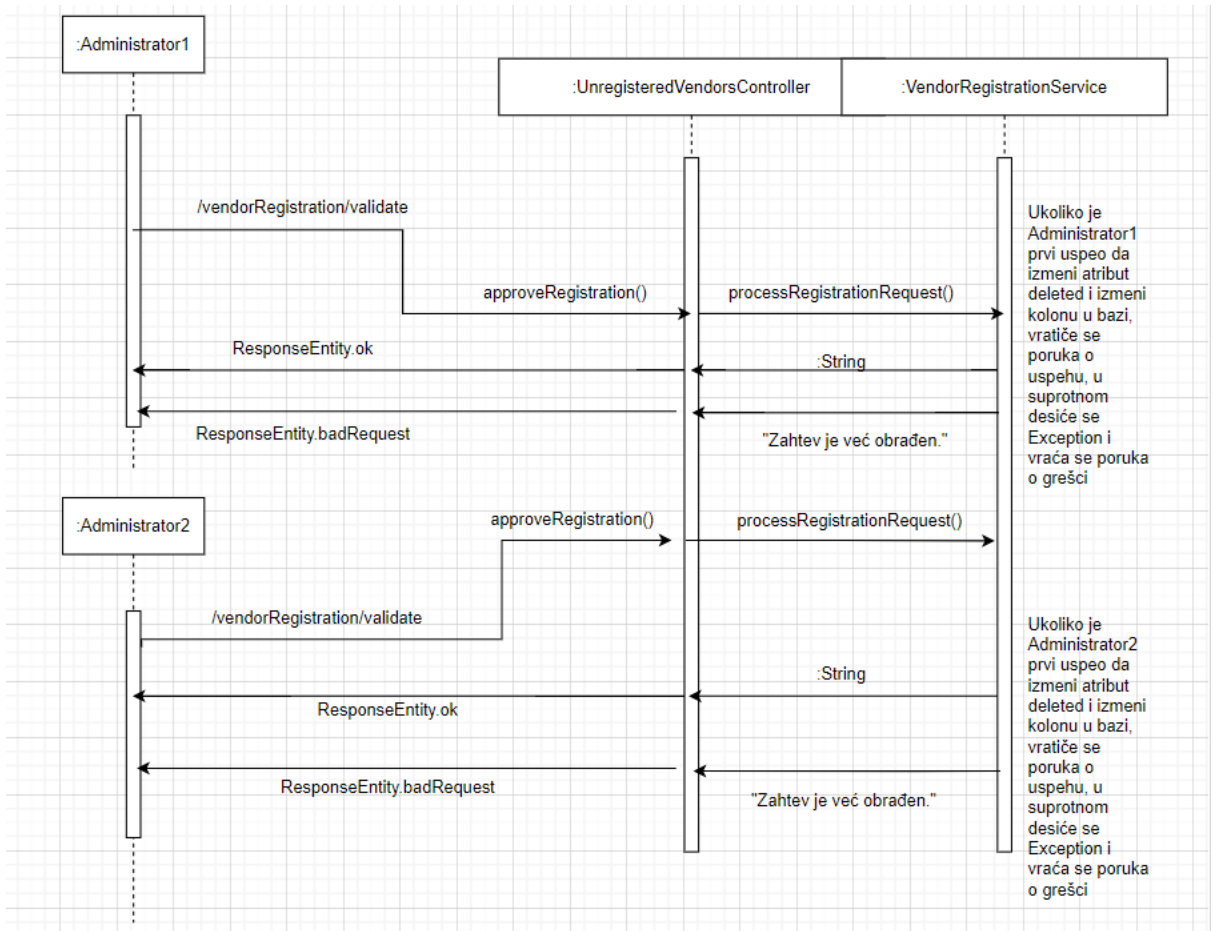
bazu, desiće se *ObjectOptimisticLockingFailureException* koji je potrebno uhvatiti i obavestiti administratora da je odabrana žalba već obrađena.

3. Konflikt

g. Opis konfliktne situacije

Na jedan zahtev za registracijom odgovara više administratora sistema

h. Tok zahteva



i. Rešenje

Za rešavanje ovog problema odabrano je optimističko zaključavanje. Razlog za to je što prilikom odgovaranja na zahtev za registracijom, zahtev je već kreiran i postoji u bazi. Ono što je potrebno odraditi jeste izmeniti kolone *deleted* i *response* i iz tog razloga objektu *RegistrationRequest* dodat je poseban atribut *Version* koji predstavlja mehanizam za rešavanje konflikata u konkurentnom pristupu entitetima baze.

Prilikom odgovaranja na zahtev za registracijom, poziva se funkcija *processRegistrationRequest*. Prvo se dobavi određeni zahtev i tom objektu se postavi odgovarajući *response*, kao i atribut *deleted* na *true*, ove izmene potrebno je ponovo sačuvati u bazu podataka. Ažuriranje kolone u bazu obavlja se funkcijom *addRegistrationRequest* ona je transakciona. Ukoliko se atribut zahteva *Version* ne poklapa na odgovarajući način sa atributom *Version* zahteva koji pokušavamo trenutno

sačuvati u bazu, desiće se *ObjectOptimisticLockingFailureException* koji je potrebno uhvatiti i obavestiti administratora da je odabrani zahtev za registracijom već obrađen.