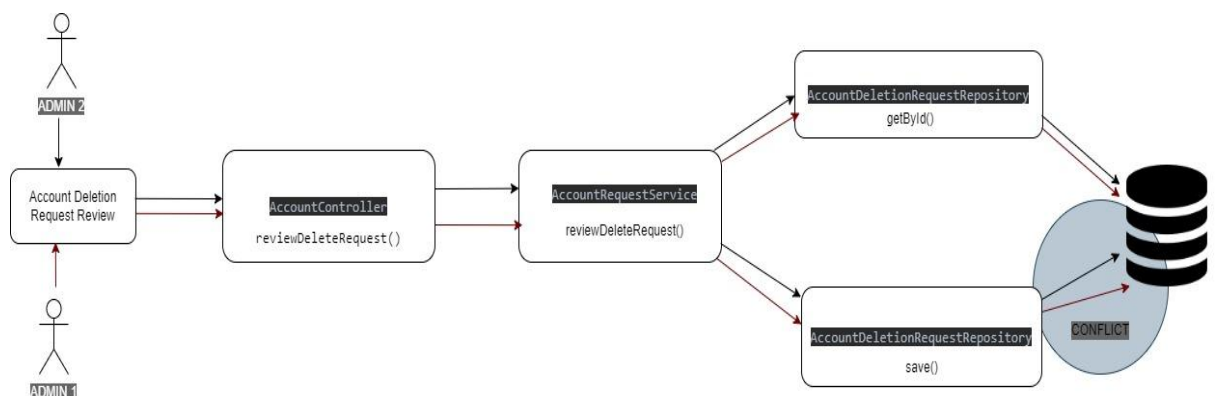


Konkurentni pristup bazi

1. Konfliktna situacija

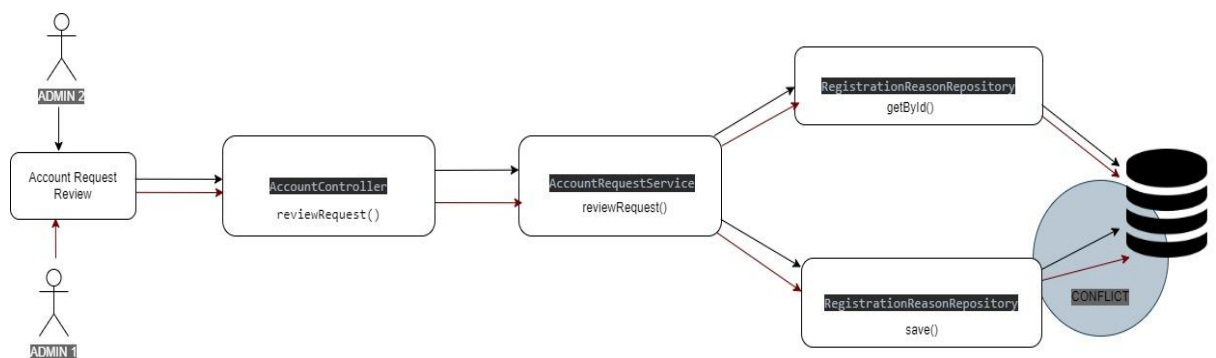
Kada korisnik podnese zahtjev za brisanje profila on se proslijeđuje svim administratorima sistema. Na jedan zahtjev treba da odgovori samo jedan administrator. Kada administrator pregleda taj zahtjev, zahtjev više nije vidljiv administratorima sistema za recenziju. Međutim, do konfliktne situacije dolazi kada jednom zahtjevu istovremeno pristupi više administara. U ovoj situaciji recenzija svakog sljedećeg koja bi pristizala bi pregazila recenzije prethodnog. Poslednja pristigla recenzija bi bila konačna, a klijent bi bio meglom obavješten o svakoj. Željeno ponašanje je da se uvaži prva recenzija koja je pristigla a svim ostalim administratorima koji pokušavaju da pregledaju da se vrati poruka da je zahtjev već pregledan. Dati problem je riješen korištenjem optimističkog zaključavanja. Ovako će se prilikom transakcije "version" povećati za jedan pa sledeća transakcija koja pokuša da sačuva promjene će prvo provjeriti vrijednost tog atributa u bazi sa onim u transakciji i baciti izuzetak ako ne bude poklapanje. Transakcija se neće izvršiti. Izabrani nivo izolacije je Read Committed. U klasi "AccountDeletionRequest" sam dodala Version polje koje se gleda prilikom izmjene samog zahtjeva.



2. Konfliktna situacija

Kada vlasnik broda/vikendice ili instruktor podnese zahtjev za otvaranje naloga na sajtu on se proslijeđuje svim administratorima sistema. Na jedan zahtjev treba da odgovori samo jedan administrator. Kada

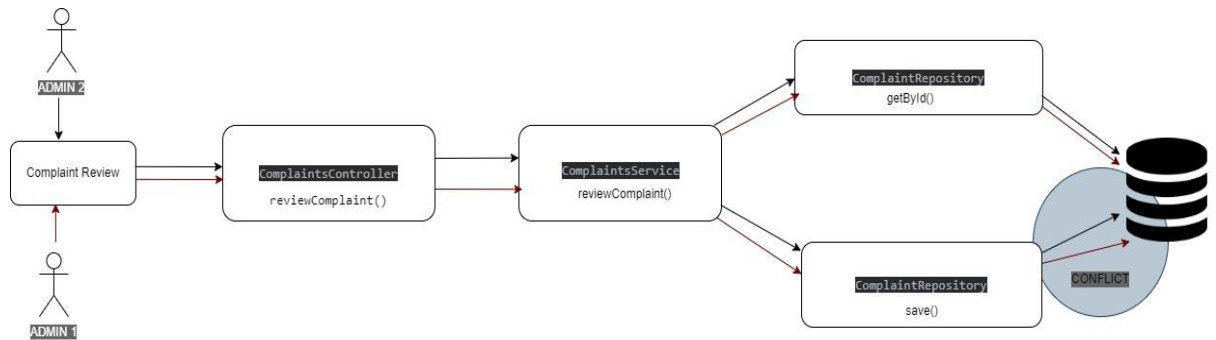
administrator pregleda taj zahtjev, zahtjev više nije vidljiv administratorima sistema za recenziju. Međutim, do konfliktne situacije dolazi kada jednom zahtjevu istovremeno pristupi više administratora. U ovoj situaciji recenzija svakog sljedećeg koja bi pristizala bi pregazila recenzije prethodnog. Poslednja pristigla recenzija bi bila konačna, a klijent bi bio meglom obavješten o svakoj. Željeno ponašanje je da se uvaži prva recenzija koja je pristigla a svim ostalim administratorima koji pokušavaju da pregledaju da se vrati poruka da je zahtjev već pregledan. Dati problem je riješen korištenjem optimističkog zaključavanja. Ovako će se prilikom transakcije “version” povećati za jedan pa sledeća transakcija koja pokuša da sačuva promjene će prvo provjeriti vrijednost tog atributa u bazi sa onim u transakciji i baciti izuzetak ako ne bude poklapanje. Transakcija se neće izvršiti. Izabrani nivo izolacije je Read Committed. U klasi “AccountRequest” sam dodala Version polje koje se gleda prilikom izmjene samog zahtjeva.



3. Konfliktna situacija

Kada klijent podnese žalbu ona se prosleđuje svim administratorima sistema na uvid. Na jednu žalbu treba da odgovori samo jedan administrator. Kada administrator pregleda taj zahtjev, žalba više nije vidljiva administratorima sistema za recenziju. Međutim, do konfliktne situacije dolazi kada jednoj žalbi istovremeno pristupi više administratora. U ovoj situaciji recenzija svakog sljedećeg koja bi pristizala bi pregazila recenzije prethodnog. Poslednja pristigla recenzija bi bila konačna, a klijent bi bio meglom obavješten o svakoj. Željeno ponašanje je da se uvaži prva recenzija koja je pristigla a svim ostalim administratorima koji pokušavaju da pregledaju da se vrati poruka da je žalba već pregledana. Dati problem je riješen korištenjem optimističkog zaključavanja. Ovako će se prilikom transakcije “version” povećati za

jedan pa sledeća transakcija koja pokuša da sačuva promjene će prvo provjeriti vrijednost tog atributa u bazi sa onim u transakciji i baciti izuzetak ako ne bude poklapanje. Transakcija se neće izvršiti. Izabrani nivo izolacije je Read Committed. U klasi `Complaint` sam dodala Version polje koje se gleda prilikom izmjene samog zahtjeva.



RA 29-2018 Petra Jović