Jelena Purić RA 215-2018 Student 3 – ISA PROJEKAT

KONKURETAN PRISTUP RESURSIMA U BAZI

KONFLIKTNE SITUACIJE

Po specifikaciji zadaci studenta 3 su bili da riješi sljedeće konfilktne situacije:

1. na jedan zahtev za brisanje naloga može da odgovori samo jedan administrator sistema
2. na jednu žalbu može da odgovori samo jedan administrator sistema

Dodatne konfliktne situacije koje su rješavane su:

1. na jedan zahtjev za registraciju prilikom odbijanja korisnika može da odgovori samo jedan administrator sistema
2. na zahtjev instruktora/vlasnika kuće/vlasnika broda da se klijentu dodijeli jedan penal može da odgovori samo jedan administrator sistema

3. više korisnika (tipa: instruktor pecanja) ne može u isto vrijeme da se registruje sa istom email adresom

Konfilktne situacije po specifikaciji 1 i 2:

Opis:

1. Administrator sistema može da odgovori na zahtjeve za brisanje naloga od strane instruktora pecanja, vlasnika vikendice ili broda, klijenta, kao i da ih obriše. Kada bi imali situaciju da dva administratora odgovaraju istom klijentu u istom trenutku na zahtjev za brisanje naloga, oba odgovora bi prošla kao validni i samim tim bi nastao problem u bazi koji odgovor bi bio proslijedjen. Ovaj problem dodavanje nove torke u bazu u isto vrijeme dovodi do nevalidnog stanja u bazi.

2. Administrator sistema može da odgovori na žalbe klijenata povodom usluga instruktora pecanja, vlasnika vikendice ili broda. Kada bi imali situaciju da dva administratora odgovaraju istom klijentu u istom trenutku na žalbu, oba odgovora bi prošla kao validni i samim tim bi nastao problem u bazi koji odgovor bi bio proslijedjen. Ovaj problem dodavanje nove torke u bazu u isto vrijeme dovodi do nevalidnog stanja u bazi.

Solucija:

Oba navedan problema možemo riješiti tako što u klasama servisa postavimo @Transactional anotacije iznad metode deleteUserAccount(UserDTO userDTO) za 1. problem, a za 2. problem iznad metoda responseToComplaint(AdventureComplaintsDTO dto), responseToComplaint(BoatComplaintsDTO dto), responseToComplaint(HomeComplaintsDTO dto) i u sklopu njih odradimo rukovanje izuzecima u skladu sa kojim će korisniku biti prikazane odgovarajuće poruke.

```
@Override
@Transactional(readOnly = false, propagation = Propagation.REQUIRES_NEW)
public User deleteUserAccount(UserDTO userDTO) throws Exception
{
```

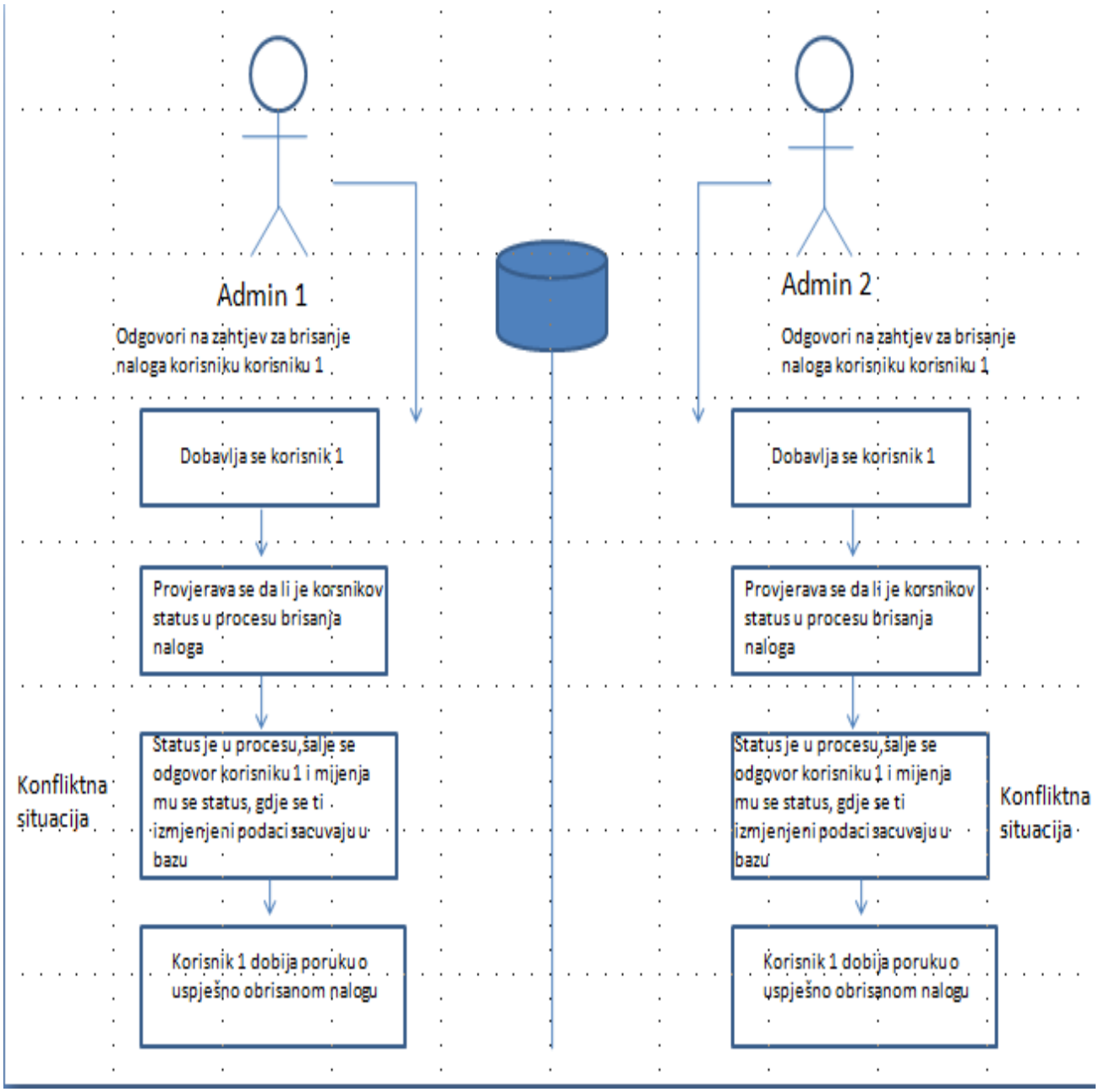
```
@Override
@Transactional(readOnly = false, propagation = Propagation.REQUIRES_NEW)
public AdventureComplaints responseToComplaint(AdventureComplaintsDTO dto) throws Exception{
```

```
@Override
@Transactional(readOnly = false, propagation = Propagation.REQUIRES_NEW)
public BoatComplaints responseToComplaint(BoatComplaintsDTO dto) throws Exception{
```

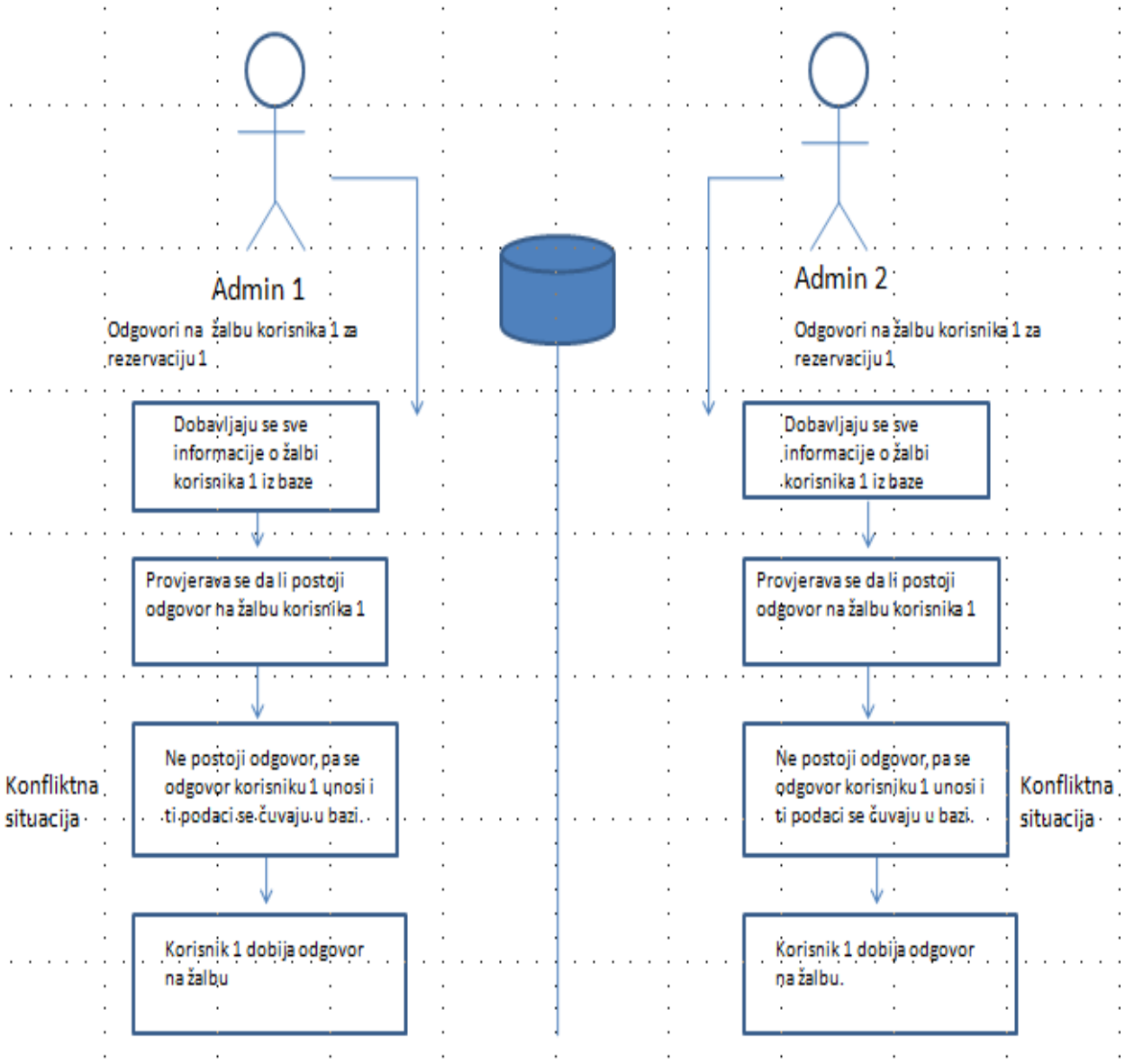
```
@Override
@Transactional(readOnly = false, propagation = Propagation.REQUIRES_NEW)
public HomeComplaints responseToComplaint(HomeComplaintsDTO dto) throws Exception{
```

Dijagram toka:

-odgovor na brisanje naloga



-odgovor na žalbu



Dodatne konfliktne situacije:

Opis:

1. Administrator sistema može da odgovori na zahtjeve za registraciju od strane instruktora pecanja, vlasnika vikendice ili broda, klijenta, ukoliko želi da ih odbije. Kada bi imali situaciju da dva administratora odgovaraju istom korisniku u istom trenutku na zahtjev za registraciju, oba odgovora bi prošla kao validni i samim tim bi nastao problem

u bazi koji odgovor bi bio proslijedjen. Ovaj problem dodavanje nove torke u bazu u isto vrijeme dovodi do nevalidnog stanja u bazi.

2. Administrator sistema može da dodijeli penal klijentu na zahtjev u izvještaju instruktora pecanja, vlasnika vikendice ili broda ukoliko je podnosilac izvještaja ostavio loš komentar. Kada bi imali situaciju da dva administrator u isto vrijeme dodijeluju penal istom klijentu na osnovu istog izvještaja, oba odgovora bi prošla kao validni i samim tim bi nastao problem u bazi koji odgovor bi bio proslijedjen. Ovaj problem dodavanje nove torke u bazu u isto vrijeme dovodi do nevalidnog stanja u bazi.

3. Kada korisnik(instruktor pecanja) pravi nalog sa email adresom, prvo se izvršava provera o jedinstvenosti unesenog email-a, u slučaju da je email zauzet korisniku će biti javljena greška u vidu poruke da adresa već postoji. U situaciji da dva korisnika istovremeno pokušavaju da pošalju zahtev za registraciju sa iste email adrese, može se desiti da oba zahteva prodju, što bi dovelo do nevalidnosti u bazi.

Solucija:

Sva tri navedan problema možemo riješiti tako što u klasama servisa postavimo @Transactional anotacije iznad metode registrationDeclined(UserDTO userDTO) za 1. problem i u sklopu njega odradimo rukovanje izuzecima u skladu sa kojim će korisniku biti prikazane odgovarajuće poruke. Za 2. problem iznad metode strikeOnePenalty(Long id) u AdventureReviewServiceImpl,HomeReviewServiceImpl, i BoatReviewServiceImpl za 2. Problem. Za 3. problem iznad metode fishingInstructorRegistration(RegistrationDTO registrationDTO) izvršeno je pesimističko zaključavanje baze i dolazi do PessimisticLockingFailureException-a.

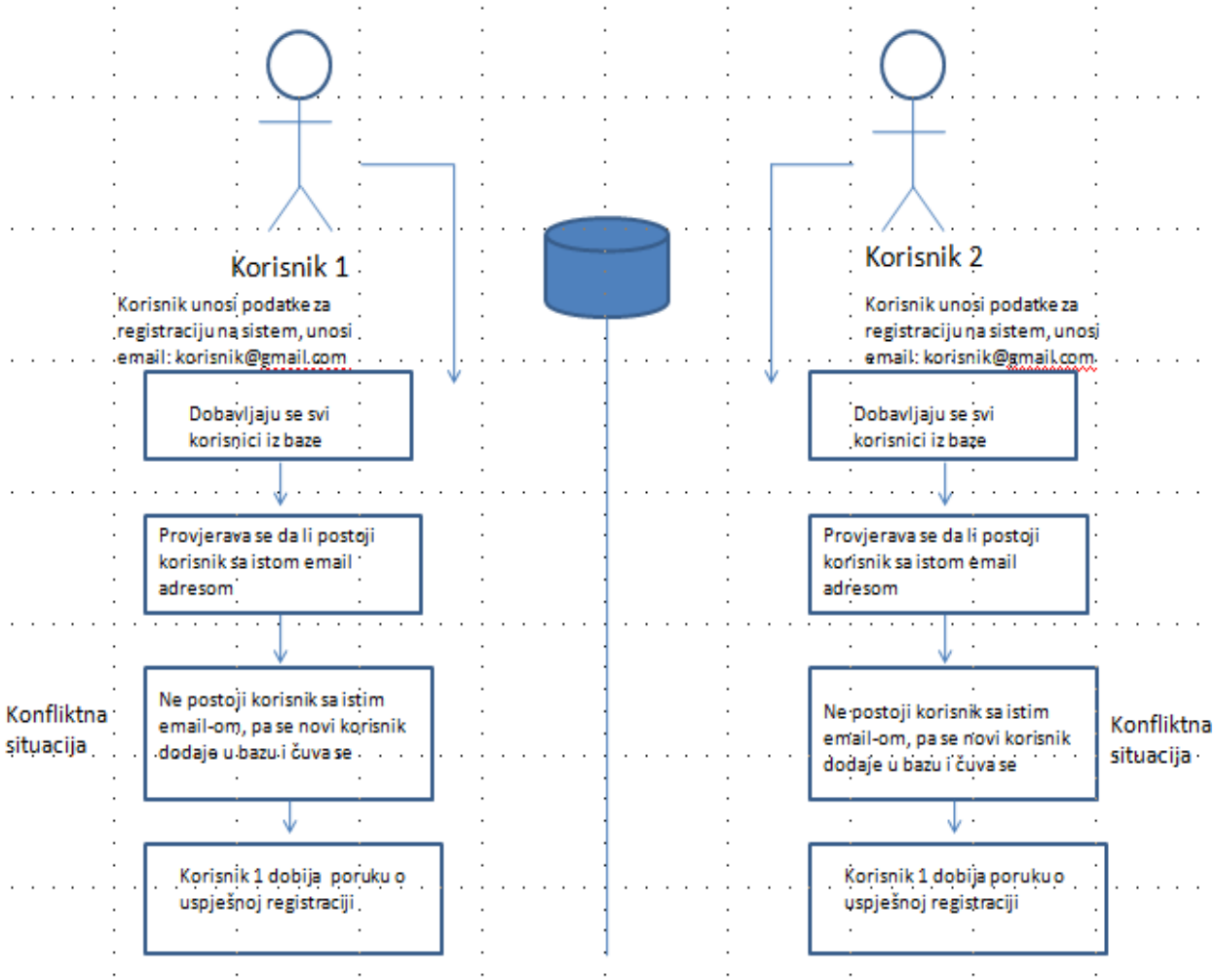
```
@Override
@Transactional(readOnly = false, propagation = Propagation.REQUIRES_NEW)
public User fishingInstructorRegistration(RegistrationDTO registrationDTO) {
```

```
@Override
@Transactional(readOnly = false, propagation = Propagation.REQUIRES_NEW)
public Boolean strikeOnePenalty(Long id){
```

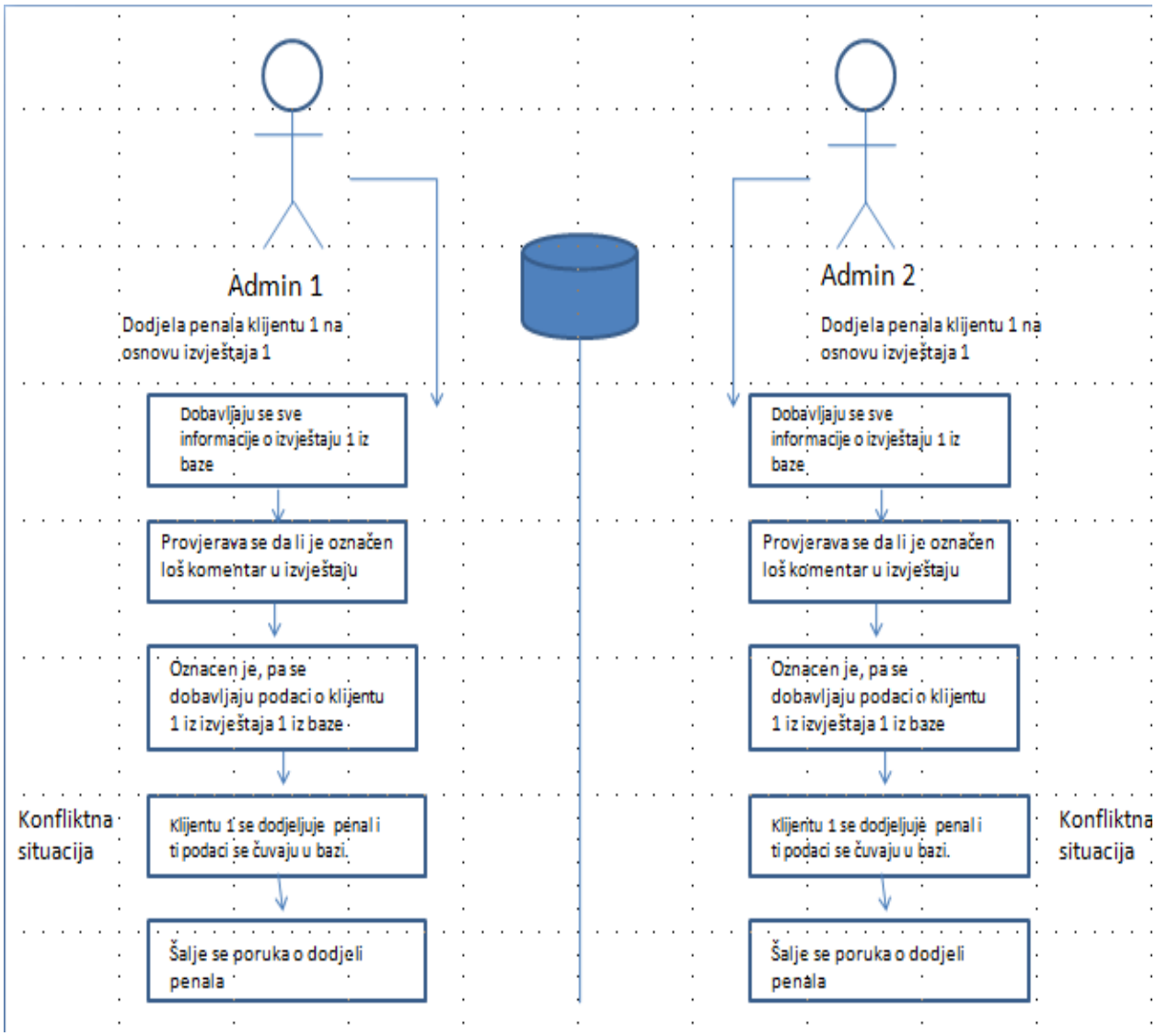
```java
@Override
@Transactional(readOnly = false, propagation = Propagation.REQUIRES_NEW)
public User registrationDeclined(UserDTO userDTO) throws Exception
{
```

Dijagram toka:

-registracija dva instruktora sa istim email-om u isto vrijeme



-dodjela penala



- **Odbijanje registracije**