

Rešavane konfliktne situacije:

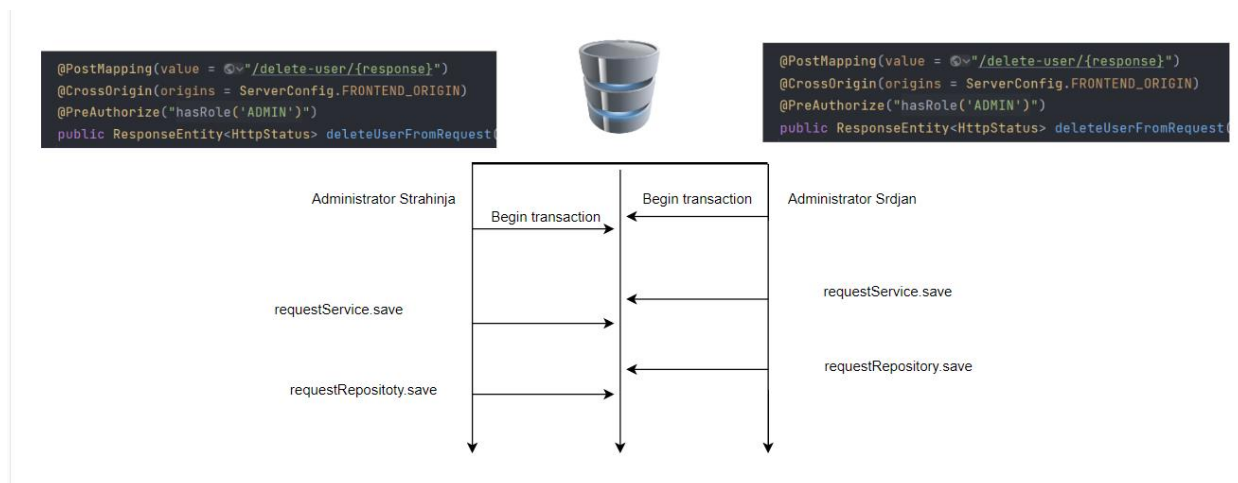
1. Na jedan zahtev za brisanje naloga može da odgovori samo jedan administrator sistema.
2. Na jednu žalbu može da odgovori samo jedan administrator sistema.
3. DODATNA SITUACIJA: Dva administratora sistema ne mogu da odgovore u isto vreme na zahtev za registraciju.

Rešenja:

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

Scenario problema: Jedan administrator sistema i drugi administrator sistema u isto vreme pokušavaju da odgovore na isti zahtev za brisanje naloga odnosno da pristupe istom entitetu. Oba administratora imaju mogućnost da odobre ili odbiju zahtev za brisanje naloga i ukoliko to rade u isto vreme, doći će do konkurentne situacije.

Recimo prvi admin će da odobri zahtev za brisanje naloga i sačuvati status (obrisan, **isDeleted** na **true**) korisnika u bazi i takođe ozančiti da je na taj zahtev odgovoreno postavivši **isAnswered** na **true** i drugi administrator će isto pokušati da odgovori na taj zahtev, samo nije svestan da je na taj zahtev već odgovoreno, jer ima staru verziju objekta svih zahteva. Prema tome, i drugi administrator će uspeti da odgovori na taj isti zahtev recimo suprotno od prvog, to sačuvati u bazi i time narušiti konzistentnost.



Rešenje problema: Za rešenje problema korišćeno je optimistično zaključavanje.

Dodata je kolona version u Request i boolean isChanged. Prilikom svakog odgovaranja na zahtev vrši se prvo menjanje boolean vrednosti is changed i njegovo čuvanje u bazu. Svakim pozivom save metode vršiće se inkrementiranje version atributa. Ukoliko dva

administratora pokušaju da odgovore na isti zahtev u isto vreme to neće biti moguće jer jedan neće imati odgovarajuću verziju version-a te neće biti izvršeno čuvanje druge transakcije. Tako da će se pri pokušaju odgovora na report nije najnovija verzija desiti *ObjectOptimisticLockingFailureException*.

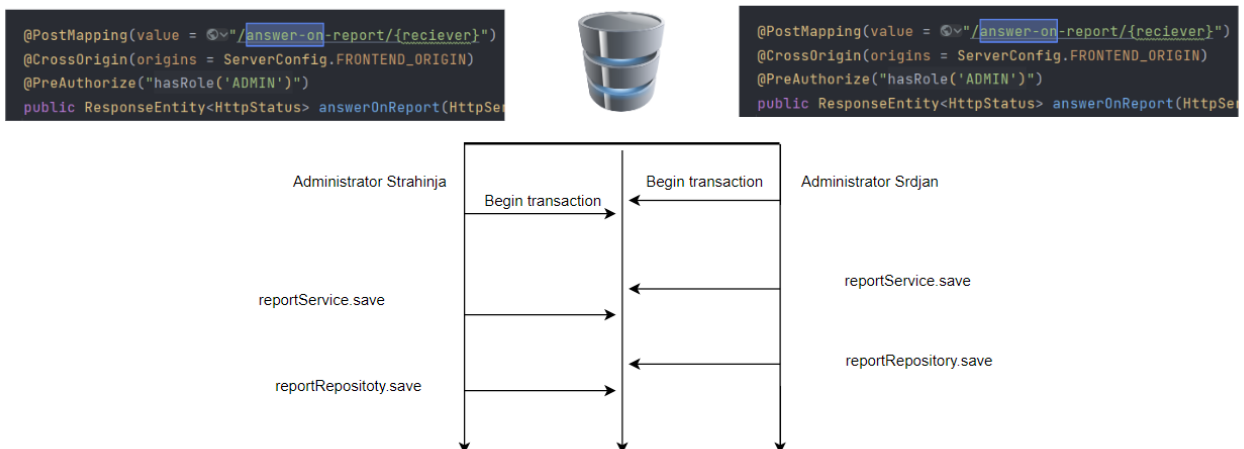
```
@Version
@Column(columnDefinition = "integer DEFAULT 0", nullable = false)
private Long version;

@Column(name = "is_changed", columnDefinition = "boolean DEFAULT false", nullable = false)
private boolean isChanged;
```

Takođe je dodata **@Transactional** anotacija nad onim metodama servisa koje rade sa odgovaranjem na zahteve.

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

Scenario problema: Jedan administrator sistema i drugi administrator sistema u isto vreme pokušavaju da odgovore da odgovore na istu žalbu za neki *Rental Service* odnosno da pristupe istom entitetu. Oba administratora imaju mogućnost da odgovore ili ignorišu žalbu i ukoliko to rade u isto vreme, doći će do konkurentne situacije. Recimo prvi admin će da odgovori na žalbu i označiti da je na tu žalbu odgovoreno postavivši ***isAnswered*** na ***true*** i drugi administrator će isto pokušati da odgovori na tu žalbu, samo nije svestan da je na tu žalbu već odgovoreno, jer ima staru verziju objekta svih žalbi. Prema tome, i drugi administrator će uspeti da odgovori na tu istu žalbu recimo različitim obraćanjem klijentu i onom ko izdaje taj *Rental Service*, to sačuvati u bazi i time narušiti konzistentnost.



**Rešenje problema:** Za rešenje problema korišćeno je optimistično zaključavanje. Dodata je kolona version u Report i boolean isChanged. Prilikom svakog odgovaranja na žalbu vrši se prvo menjanje boolean vrednosti is changed i njegovo čuvanje u bazu. Svakim pozivom save metode vršiće se inkrementiranje version atributa. Ukoliko dva administratora pokušaju da odgovore na istu žalbu u isto vreme to neće biti moguće jer jedan neće imati odgovarajuću verziju version-a te neće biti izvršeno čuvanje druge transakcije. Tako da će se pri pokušaju odgovora na report nije najnovija verzija desiti *ObjectOptimisticLockingFailureException*.

```

@Version
@Column(columnDefinition = "integer DEFAULT 0", nullable = false)
private Long version;

@Column(name = "is_changed", columnDefinition = "boolean DEFAULT false", nullable = false)
private boolean isChanged;

```

Takođe je dodata **@Transactional** anotacija nad onim metodama servisa koje rade sa odgovaranjem na žalbe.

### 3. DODATNA SITUACIJA: Dva administratora sistema ne mogu da odgovore u isto vreme na zahtev za registraciju

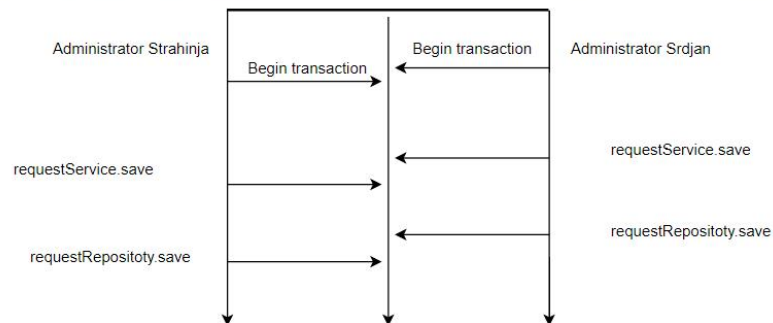
Scenario problema: Jedan administrator sistema i drugi administrator sistema u isto vreme pokušavaju da odgovore na isti zahtev za registraciju odnosno da pristupe istom entitetu. Oba administratora imaju mogućnost da odobre ili odbiju zahtev za registraciju i ukoliko to rade u isto vreme, doći će do konkurentne situacije. Recimo prvi admin će da odobri zahtev za registraciju i sačuvati status(aktivan, *isActive* na *true*)

korisnika u bazi i takođe označiti da je na taj zahtev odgovoreno postavivši **isAnswered** na **true** i drugi administrator će isto pokušati da odgovori na taj zahtev za registraciju, samo nije svestan da je na taj zahtev već odgovoreno, jer ima staru verziju objekta svih zahteva. Prema tome, i drugi administrator će uspeti da odgovori na taj isti zahtev recimo suprotno od prvog, to sačuvati u bazi i time narušiti konzistentnost.

```
@PostMapping(value = @RequestMapping("/registration/{response}")
@CrossOrigin(origins = ServerConfig.FRONTEND_ORIGIN)
@PreAuthorize("hasRole('ADMIN')")
public ResponseEntity<HttpStatus> registrationAllowing(
```



```
@PostMapping(value = @RequestMapping("/registration/{response}")
@CrossOrigin(origins = ServerConfig.FRONTEND_ORIGIN)
@PreAuthorize("hasRole('ADMIN')")
public ResponseEntity<HttpStatus> registrationAllowing(
```



**Rešenje problema:** Za rešenje problema korišćeno je optimistično zaključavanje. Dodata je kolona version u RentalService i boolean isChanged. Prilikom svakog dodavanja nove rezervacije vrši se prvo menjanje boolean vrednosti is changed i njegovo čuvanje u bazu. Svakim pozivom save metode vršiće se inkrementiranje version atributa. Isto tako se prilikom izmene podataka za svaki rental takođe vrši inkrementiranje atributa version. Tako da će se pri pokušaju ažuriranja rentala gde nije najnovija verzija desiti *ObjectOptimisticLockingFailureException*. Dodate su anotacije @Transactional nad metodama za izmenu rentala i kreiranje rezervacija.

```
@Version
@Column(columnDefinition = "integer DEFAULT 0", nullable = false)
private Long version;

@Column(name = "is_changed", columnDefinition = "boolean DEFAULT false", nullable = false)
private boolean isChanged;
```