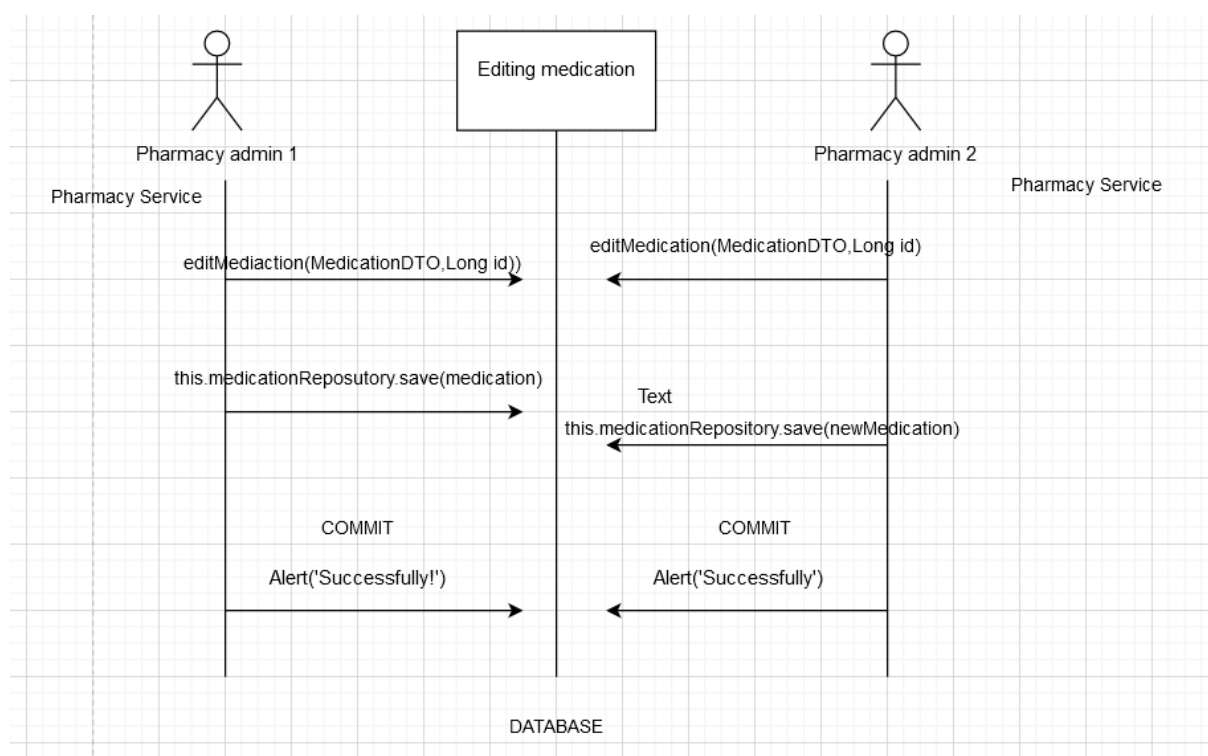


- **Editovanje lekova:**

Problem: Administrator apoteke 1# pristupa lekovima i menja ime leka br. 1. Administrator apoteke 2# pristupa lekovima istovremeno i menja ime leka br. 1. Prva pristigla transakcija menja objekat međutim samo se promene druge transakcije prihvataju. Administrator apoteke 1# nije obavešten o neuspelosti transakcije. Podaci prve transakcije su izgubljeni.

Tok izvršavanja:



Rešenje: Optimističko zaključavanje, kontrola verzije i postavljanje parametra transakcije na REPEATABLE\_READ.

Implementirano u: PharmacyService

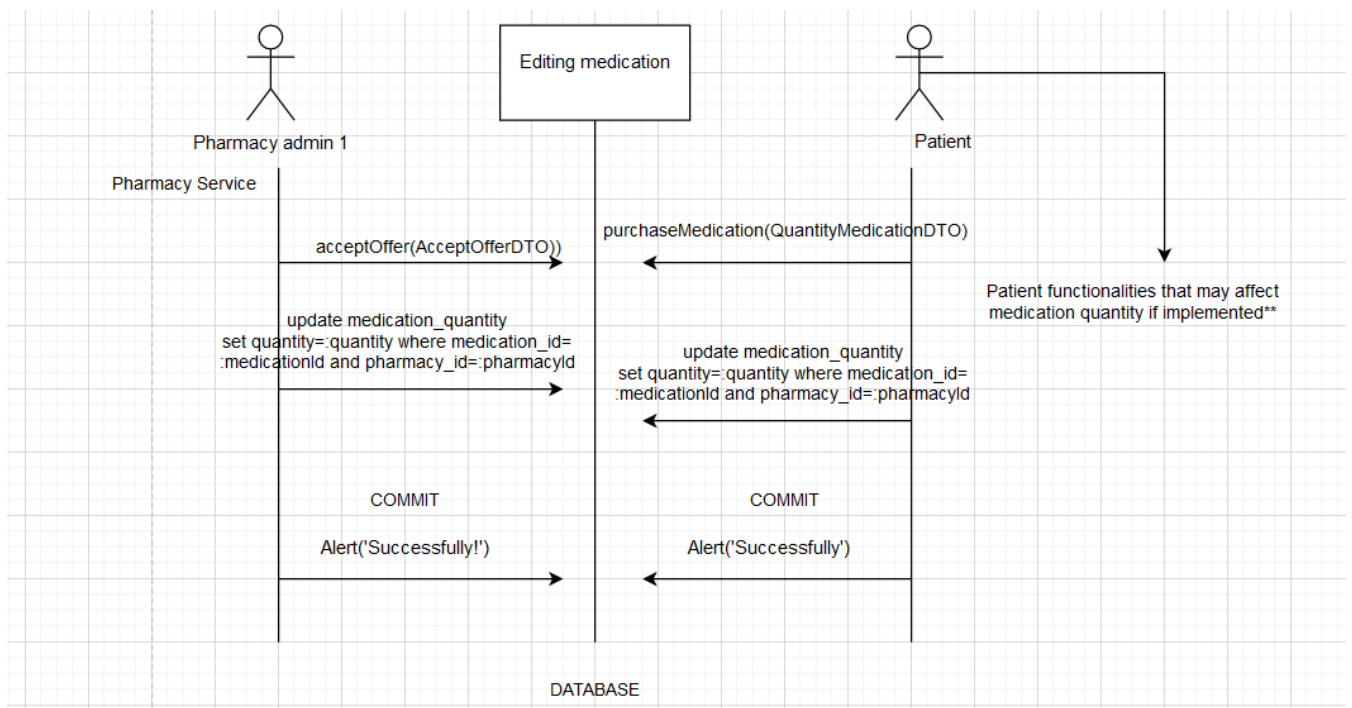
```

@Override
@Transactional(propagation = Propagation.REQUIRES_NEW, isolation = Isolation.REPEATABLE_READ, rollbackFor = Exception.class, readOnly = false)
public boolean editMedication(MedicationQuantityDTO dto, Long id) {
    // ...
}
    
```

- **Količina leka na stanju se mora ispravno ažurirati nakon prihvatanja ponude za narudžbenicu:**

Problem: Pristupanje količini leka od strane više korisnika, kao što su administrator apoteke kada prihvata narudžbenice, pacijent koji kupuje izvesnu količinu leka itd. Sve to može dovesti do manje ili veće količine leka koji se nalazi u apoteci, a koji nije tačan.

Tok izvršavanja:

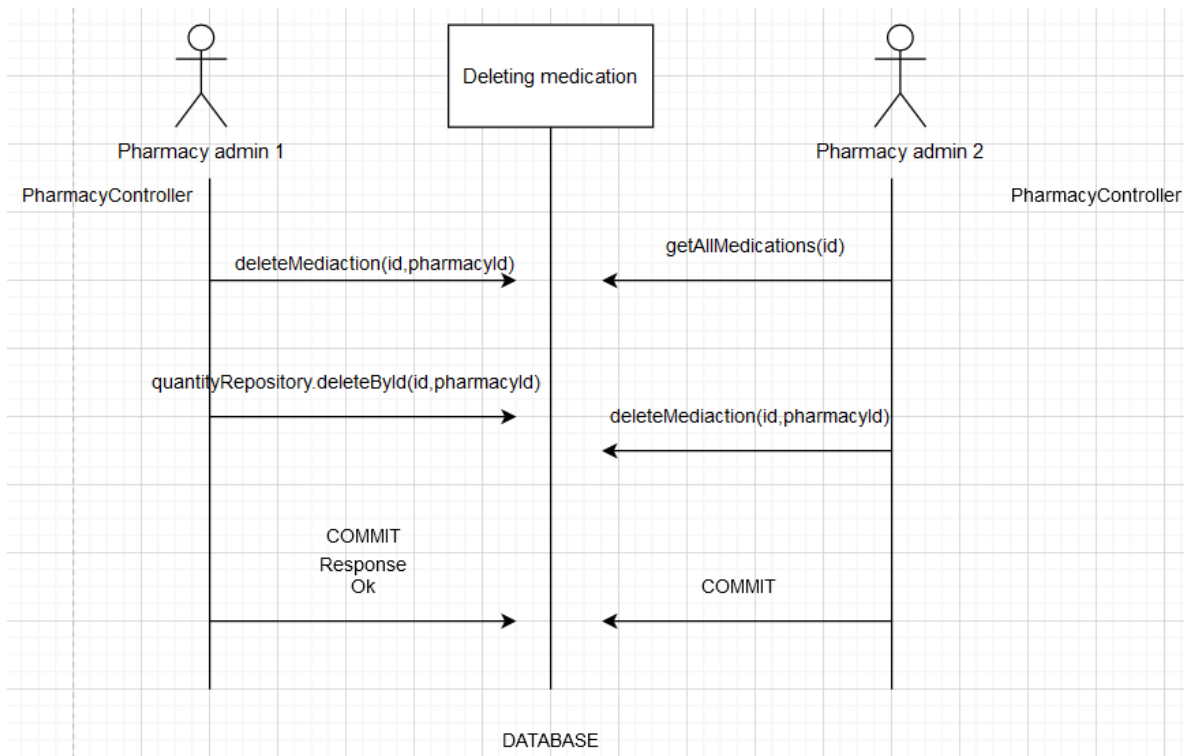


Rešenje: Postavljanje transakcije na metodu `Update()` koja se poziva u `acceptOffer()` metodi. Postavljanje optimističkog zaključavanja sa atributom `Version` kako bi se pratio tok verzije podatka.

- **Brisanje lekova:**

Problem: Administrator apoteke 1# brise lek iz liste lekova apoteke dok administrator apoteke 2# pokušava simultano čitati isti lek. Javlja se lost update jer se obrisani podatak vidi kao ishod.

Tok izvršavanja:



Rešenje: Optimističko zaključavanje sa version atributom u klasi i postavljanje izolacije transakcije na READ\_UNCOMMITTED.

Implementirano u: PharmacyController

```

@RequestMapping(method = RequestMethod.POST, path = "/delete-medication/{id}")
@ResponseStatus(HttpStatus.OK)
@Transactional(propagation = Propagation.REQUIRES_NEW, isolation = Isolation.READ_UNCOMMITTED, rollbackFor = Exception.class)
public boolean deleteMedication(@RequestBody Long medicationId, @PathVariable("id") Long id){
    Pharmacy pharmacy = this.pharmacyService.getById(id);
    boolean success = this.pharmacyService.deleteMedication(id, medicationId);
    return success;
}
  
```