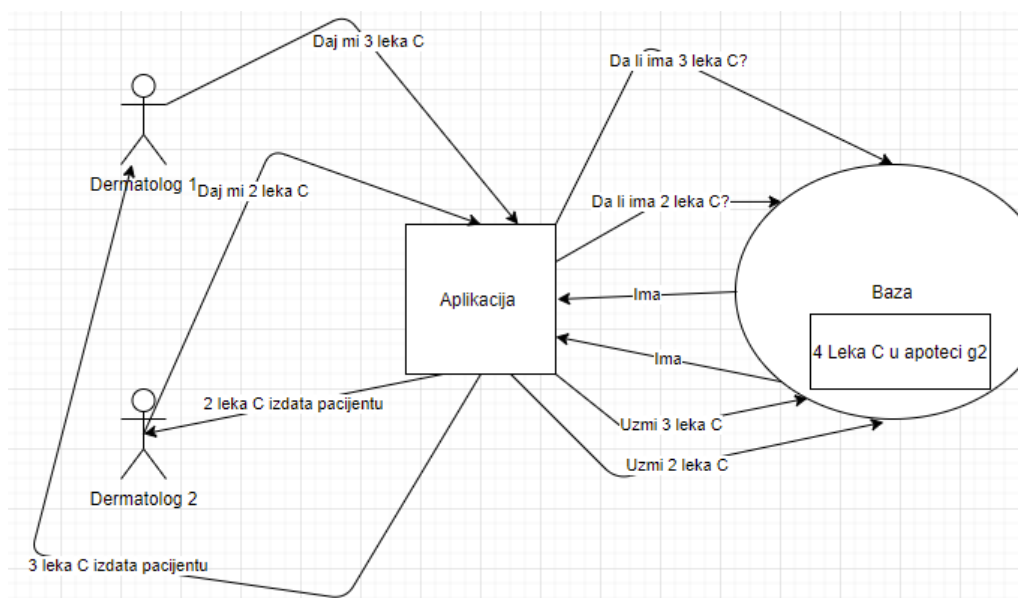


## Konkurentni pristup bazi –Student 3 – Filip Živanac SW-66-2018-tim07

### Scenario 1

Opis scenarija – Dva ili više farmaceuta/dermatologa salju zahtev za rezervaciju liste lekova na kraju pregleda. U toj listi se takođe nalaze i količine lekova. Apoteka ima određen broj datog leka u svojim zaliham. Ukoliko farmaceuti/dermatolozi u isto vreme salju zahtev za dat lek, a ukupna tražena količina prevazilazi zalihe apoteke može se dogoditi da svi zahtevi prođu jer se pri proveru količine ne očitavaju najsvežije informacije iz baze o stanju tog leka u apoteci. Prikaz neželjenog scenarija.



Rešenje – Potrebno je zaključati podatke o količini lekova pri obradi zahteva koji sadrži taj lek u određenoj apoteci. Međutim u tom sličaju može nastati problem u vidu deadlocka. Ukoliko zahtev A traži lekove(2, 3) redom a zahtev B traži lekove redom (3, 2) i simultano se obrađuju, A će zaključati lek 2 a B lek 3. Zatim A neće moći da pristupi leku 3 jer ga je B zaključao a B neće moći da pristupi leku 2 jer ga je A zaključao. Ovaj problem se rešava sortiranjem ulazne liste lekova po njihovim šiframa tako će se u ovom slučaju lista u zahtevu B sortirati u (2, 3). B neće moći da pristupi leku 2 jer ga je A zaključao i moraće da sačeka da se zahtev A obradi i otključa resurse.

Implementacija – Metoda koja zaključava resurse je sledeća

```
@Lock(LockModeType.PESSIMISTIC_WRITE)
@Query("Select dp from DrugPharmacy dp where dp.drug=?1 and dp.pharmacy=?2")
public DrugPharmacy findByDrugAndPharmacy(Drug d, Pharmacy ph);
```

a ona je metoda repozitorijuma za pristup leku u nekoj apoteci. Servisna metoda koja nju poziva je

```
@Transactional(readOnly = false, propagation = Propagation.REQUIRES_NEW)
@PostMapping(path="/submitReport/{id}/{report}", consumes="application/json")
public ResponseEntity submitReport(@PathVariable long id, @PathVariable String report, @RequestBody List<AppDrugDTO> drugs) {
```

a ona je anotirana sa transactional da bi se označio opseg zaključavanja. To je metoda klase

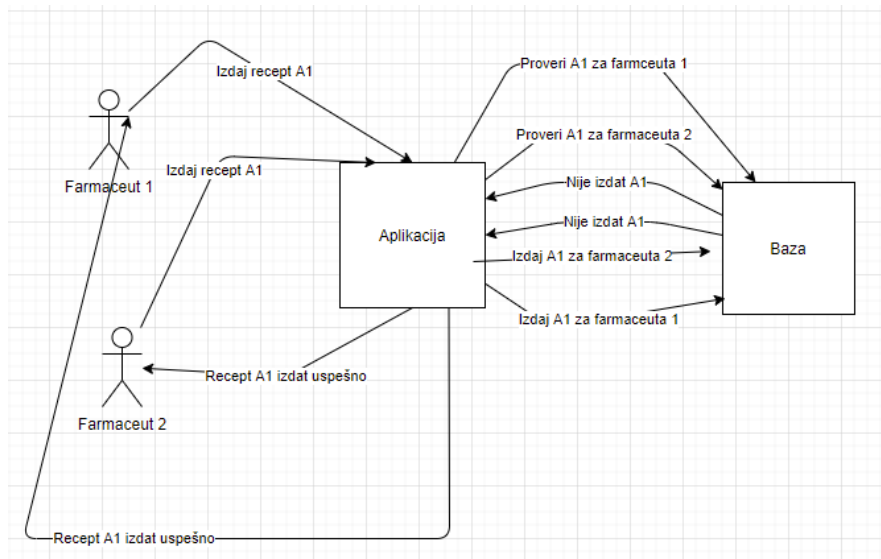
```
@Service  
public class AppointmentService {
```

 koja je anotirana sa anotacijom Service.

U RequesteBody- ove metoda nalazi se lista lekova sa njihovim količinama koja stiže sa klijentske strane i ona se sortira. Koristi se pessimistic lock da preventivno zaključa resurse u bazi jer se proveravanje količine lekova mora raditi sa najsvežijim podacima i zato je takođe zabranjeno čitanje eventualno starih podataka.

## Scenario 2

Opis scenarija – Dva ili više farmaceuta žele da izdaju istu rezervaciju lekova u isto vreme međutim rezervacija se može izdati samo jednom. Ukoliko je rezervacija već izdata ona se više ne može izdati a farmaceutu se šalje poruka o odbijanju. Problem nastaje u tome što se proverava da li je rezervacija izdata vrši pre promene njenog stanja tako da se očitavaju stari podaci. Prikaz neželjenog scenarija.



Rešenje – Potrebno je zaključati podatke o rezervaciji pri obradi zahteva za rezervaciju. To mora izvršiti metoda koja prvi vadi te podatke iz baze.

Implementacija – Metoda koja zaključava podatke je sledeća

```
@Query("SELECT r from Reservation r where r.id = ?1")  
@Lock(LockModeType.PESSIMISTIC_WRITE)  
// @QueryHints({@QueryHint(name = "javax.persistence.l  
public Reservation findOneById(long id);
```

a ona je metoda repozitorijuma za pristup rezervacijama. Servisna metoda koja nju poziva je

```
@PostMapping(path = "/issueReservation/{code}")  
@Transactional(readOnly = false, propagation = Propagation.REQUIRES_NEW)  
public ResponseEntity issueReservation(@PathVariable long code) {
```

a ona je anotirana sa transactional da bi se označio opseg zaključavanja. Čitanje je zabranjeno da se ne bi očitali eventualno stari podaci. Issue Reservation je metoda klase

```
@Service
```

```
@RestController
```

```
@RequestMapping("/reservation")
```

```
public class ReservationService {
```

koja je anotirana sa Service.

Kao argument metode šalje se kod rezervacije koja želi da se izda. Pesimističkim zaključavanjem sprečavamo čitanje starih podataka preventivnim zaključavanjem. A ukoliko neka metoda želi pristup istim podacima moraće da sačeka da se ova metoda izvrši.