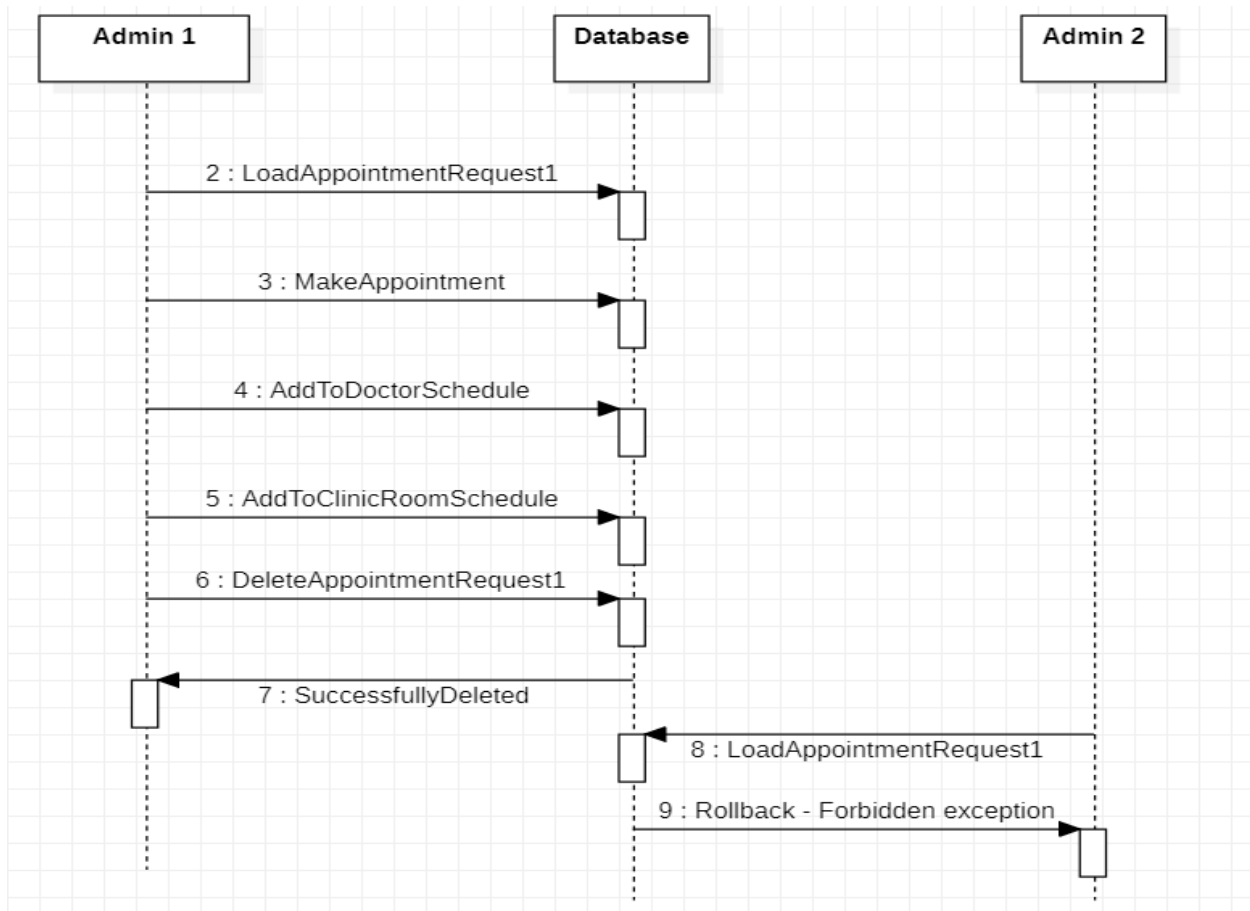


## Konfliktne situacije – student 2

Nikola Kabašaj, sw50/2017

### 1. Konfliktna situacija zauzimanje sale

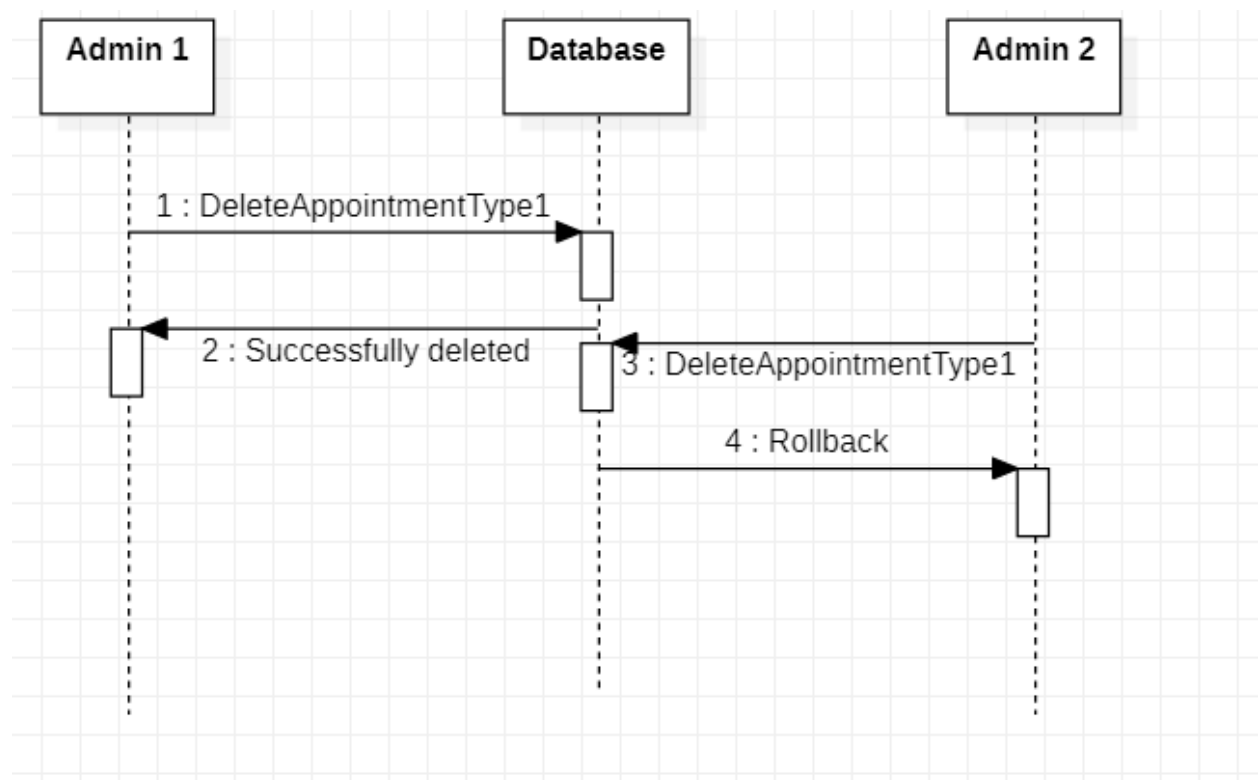
Opis problema: Administrator klinike ima mogućnost zakazivanja sobe za preglede na osnovu zahteva za pregled. Za traženo vreme pregleda administrator u sistemu pretražuje sve slobodne sale za traženi termin. Neku od ponuđenih sala bira klikom, i šalje se zahtev serveru. Potrebno je da server kreira novi pregled u bazi, zauzimaju se termini kod doktora i sobe, i zahtev za pregled se briše iz baze. Problem nastaje kada dva administratora rezervišu sobu u isto vreme za isti traženi zahtev. Tada se bazi pristupa tako što se za dva puta za isti zahtev kreira pregled, i dva puta se u kalendar doktora i sobe stavljaju isti podaci.



Rešenje problema: Problem je rešen pesimističkim zaključavanjem, što nam omogućava da prilikom završetka prve transakcije u potpunosti, druga transakcija bude poništena (*rollback*). Ovo je postignuto tako što je u metodi servisu postavljena anotacija *@Transactional*, što omogućava posmatranje akcija unutar nje kao akcije transakcije. Metoda koja se poziva u repozitorijumu baze prilikom provere da li zahtev koji se briše postoji u bazi ima anotaciju *@Lock(LockModeType.PESIMISTIC\_WRITE)* koja omogućava zaključavanje ove metode za transakcije koje pokušaju u međuvremenu da joj pristupe.

## 2. Brisanje tipa pregleda

Opis: Administrator klinike ima opciju brisanja tipa pregleda. Prilikom ove transakcije server pokušava da obriše tip pregleda, tako da nijedan doktor ne sadrži specijalizaciju koja se poklapa sa tipom pregleda koji se briše, što je urađeno logičkom proverom. Potrebno je onemogućiti brisanje tipa pregleda ukoliko dva administratora pokušaju da ga obrišu u isto vreme.

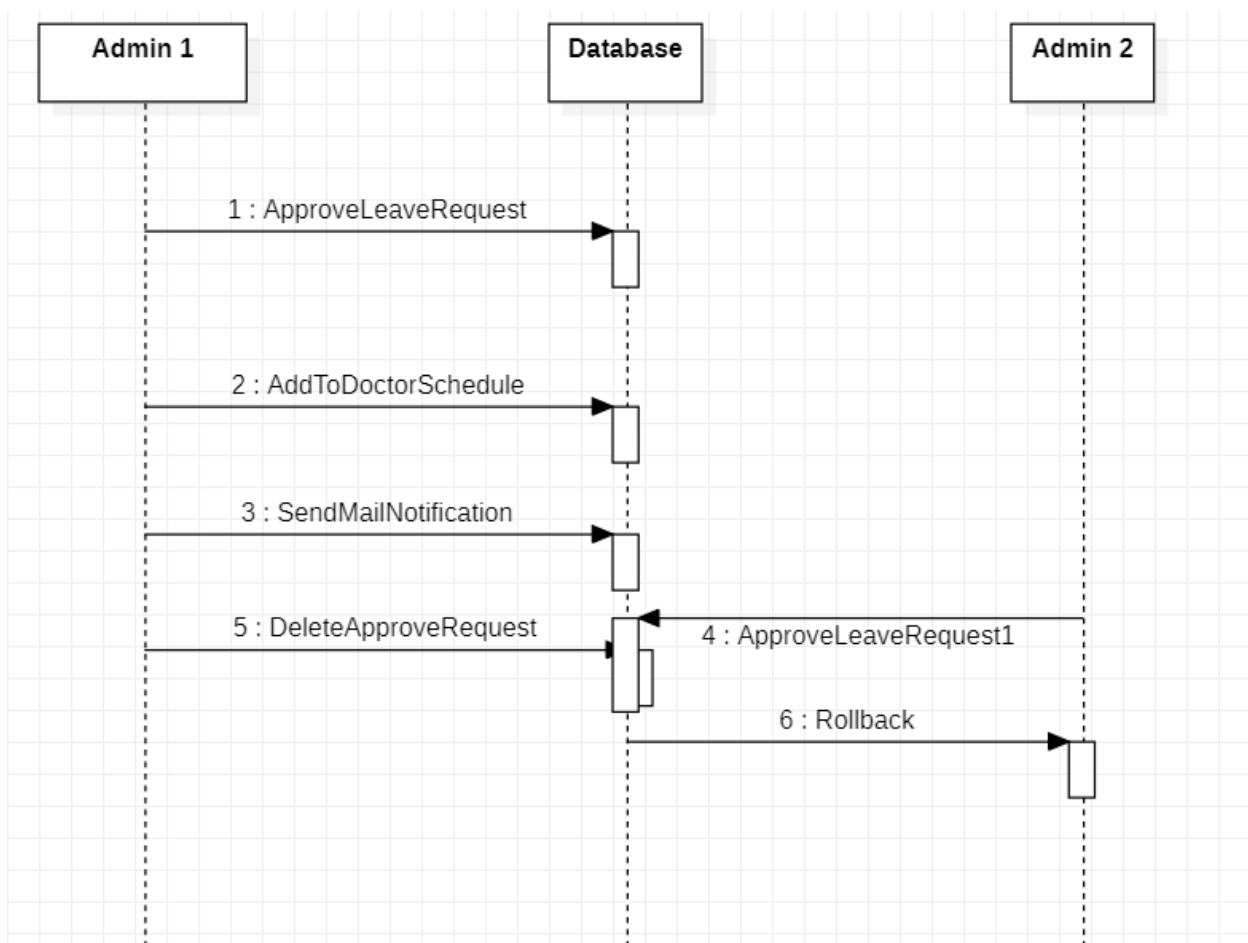


Rešenje problema: Metoda koja izvršava brisanje tipa pregleda dobija anotaciju *@Transactional*, a sama metoda repozitorijuma dobija anotaciju

@Lock(LockModeType.PESIMISTIC\_WRITE), čime prva pristigla transakcija (prvi pristigli zahtev za brisanje) zabranjuje brisanje svakoj sledećoj.

### 3. Konfliktna situacija potvrde zahteva za odsustvo/godišnji odmor

Opis problema: Administrator kliike ima mogućnost potvrde zahteva za odsustvo doktora, gde za tražene termine odsustva administrator jednim klikom potvrđuje taj zahtev, čime se remeti samo ponašanje pro. U ovom slučaju u bazu se dodaje zakazani termin u tabelu zakazanih termina doktora, briše se sam zahtev i šalje se mejl notifikacija da je zahtev prihvaćen. Ukoliko dva ulogovana administratora potvrde jedan zahtev u isto vreme, tada se dešava da se u tabelu odsustva doktora dva puta dodaje isti termin i dva puta mu je poslata mejl notifikacija za isti zahtev za odsustvo.



Rešenje problema: Problem se rešava pesimističkim zaključavanjem, tako što se metoda koja čita zahtev za odsustvo zaključa. Dodaje se anotacija *@Transactional* na metodu, tako da se njene akcije izvršavaju unutar jedne transakcije. Metoda koje čita iz baze zahtev za odsustvo ima anotaciju *@Lock(LockModeType.PESIMISTIC\_WRITE)* koja ne dozvoljava da obe transakcije pristupe zahtevu za odsustvo.