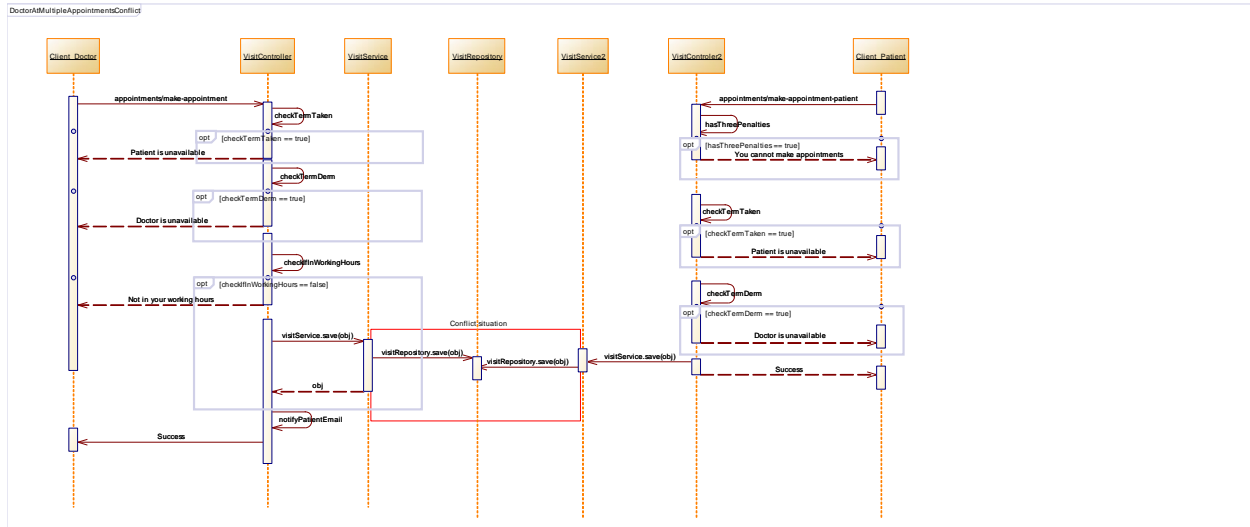


Internet softverske arhitekture

Konfliktne situacije – student 3

Filip Volarić

Jedan dermatolog ne može istovremeno da bude prisutan na više različitih pregleda



Analogno za farmaceuta na više različitih savetovanja i za zakazivanje savetovanja od strane pacijenata.

Rešenje konflikta:

Problem nastaje ukoliko se obrađuju 2 zahteva paralelno. Vrš se provere nad dobavljenim podacima i vrši se čuvanje u bazu. Kako bi podaci bilo pravovremeno dobavljeni i konzistentni potrebno je omogućiti da dok se vrše date provere, do momenta čuvanja u bazu, relacija bude nedostupna za dobavljanje. Pristup je odabran zbog sigurnosti da do ove situacije neće doći.

Klasa VisitService je anotirana sa `@Transactional(readOnly = true)`

```
@Service
@Transactional(readOnly = true)
public class VisitService implements IService<Visit>{
```

A metoda save je anotirana sa `@Transactional(readOnly = false)`.

```

@Transactional(readonly = false)
@Override
public Visit save(Visit obj) {
    return visitRepository.save(obj);
}

```

Što znači da će se operacija dobavljanja podataka uzrokovati da u tom trenutku ne može da se izvrši druga operacija dobavljanja dok se ne sačuva podatak u bazu. Rešenje je neoptimalno, ali je izabrano zbog sigurnosti. U odnosu na dijagram, logika validacije podataka (da li su pacijent ili doktor zauzeti, ili je termin van doktorovog radnog vremena) je izmeštena iz metode kontrolera u save metodu servisa.

```

@Transactional(readonly = false)
@Override
public Visit save(Visit obj) {
    if(!checkTermTaken(obj))
        return null;

    if(!checkTermDerm(obj, obj.getDoctorId()))
        return null;

    if(!checkIfInWorkingHours(obj)) {
        return null;
    }
    return visitRepository.save(obj);
}

```

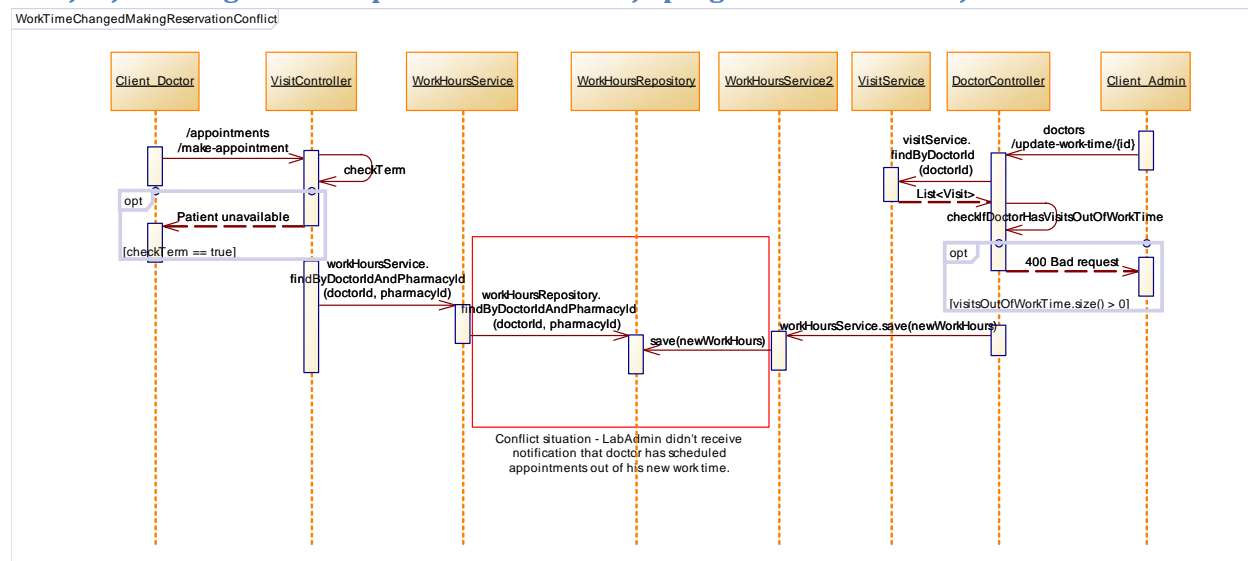
Lock mehanizam koristimo u Repository sloju.

```

@Lock(LockModeType.PESSIMISTIC_READ)
List<Visit> findByPatientIdEquals(Long patientId);

```

Menjanje radnog vremena prilikom zakazivanja pregleda ili savetovanja



Rešenje konflikta

Potrebno je obe operacije (kreiranje novog pregleda ili savetovanja, i update radnog vremena) izvršiti atomično. Ukoliko se desi situacija prikazana na dijagramu, doktor neće biti obavešten da doktor ima zakazan termin van radnog vremena, odnosno doktor neće biti obavešten da mu je promenjeno radno vreme u zavisnosti od toga koji će zahtev prvi upisati u bazu. Potrebno je izmeniti metodu `save()` klase `WorkHoursService` tako da se logika provere vrednosti obavlja unutar nje. Takođe je potrebno anotirati je sa `@Transactional`.

```

@Transactional
@Override
public WorkHours save(WorkHours obj) {
    // ...
}

```

```

@Lock(LockModeType.PESSIMISTIC_READ)
List<WorkHours> findByDoctorIdAndPharmacyId(Long doctorId, Long pharmacyId);

```