

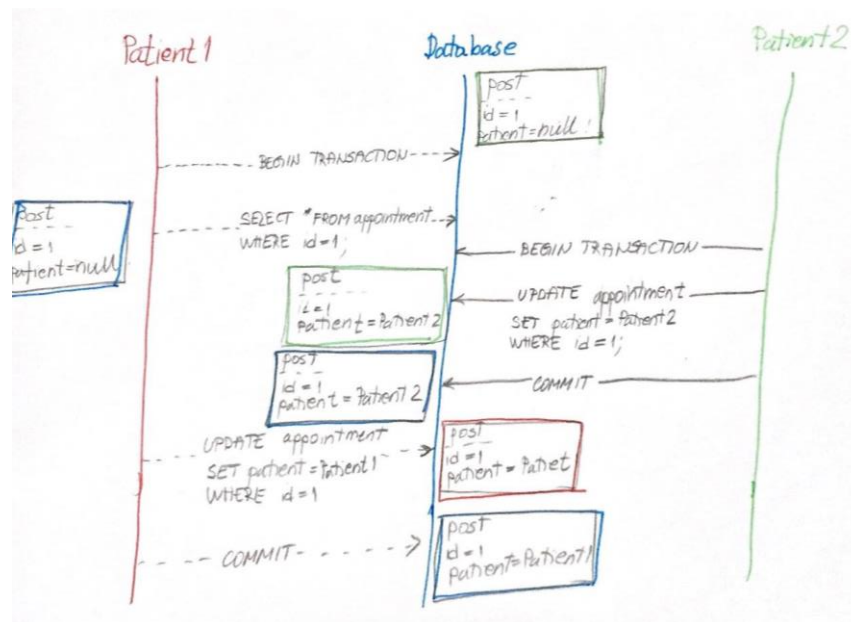
4.4. Konkurentni pristup resursima u bazi:

1. Rezervisanje brzih pregleda (student 1)

1.1. Opis konfliktne situacije

U bazi postoje unapred definisani pregledi, koji su definisani od strane admina klinike. Pacijenti te klinike imaju mogucnost rezervisanja unapred definisanih pregleda jednim klikom. Sistem ne sme da dozvoli da dva pacijenta u isto vreme rezervisu isti pregled jer ce doci do nekonzistentnosti podataka tj. druge izmene ce pregaziti prve. Prvi pacijent koji je zatrazio rezervaciju pregleda mora biti i pacijent koji je dobio tu rezervaciju.

1.2. Crtez



1.3. Resenje

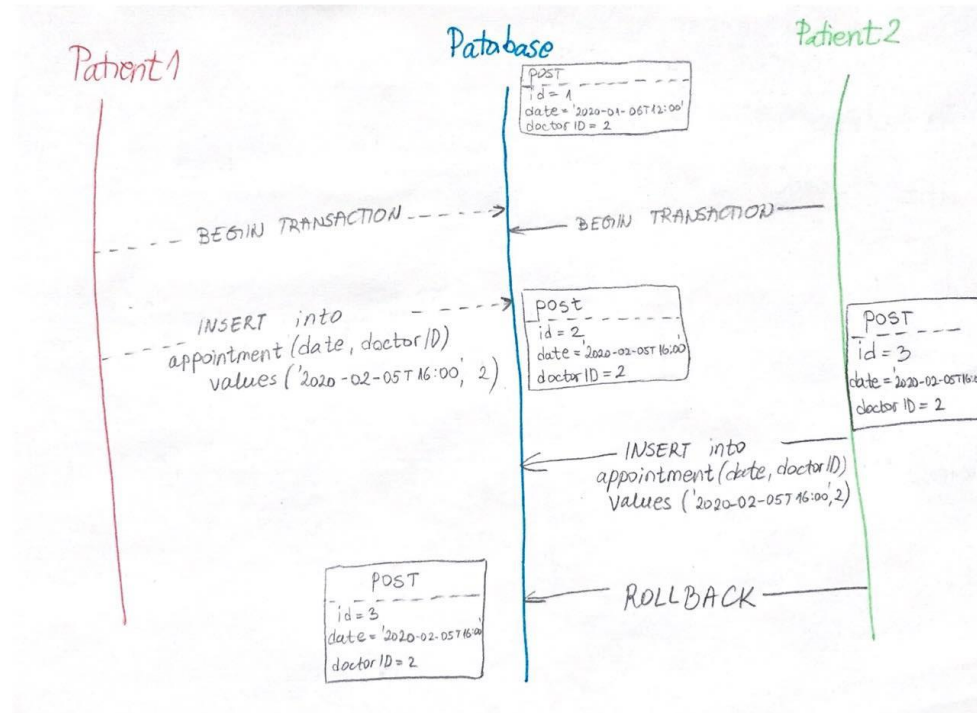
Dodato polje `@Version` u klasu Appointment.java kako bi se znala koja je verzija objekta nad kojom upit radi. Ukoliko se ne podudara verzija prvobitnog objekta i objekta sa kojim radimo u trenutku poredjenja, znaci da je u medjuvremenu objekat izmenjen (u nasem slucaju izmenjeno je polje patient unutar brzog pregleda). Koriscena izolacija je `UNCOMMITTED` kako bi se zastitilo od dolaska u konfliktnu situaciju

2. Više pacijenata ne može istovremeno da zatraži upit za pregled u isto vreme kod istog lekara (student 1)

2.1. Opis konfliktne situacije

Pacijent ima pravo da zatraži upit za pregled u određenoj klinici. Prilikom formiranja upita pacijent bira između ostalog i vreme i lekara. Zbog ovoga može doći do situacije da u isto vreme dva različita pacijenta zatraže upit u isto vreme kod istog lekara.

2.2. Crtez



2.3. Resenje

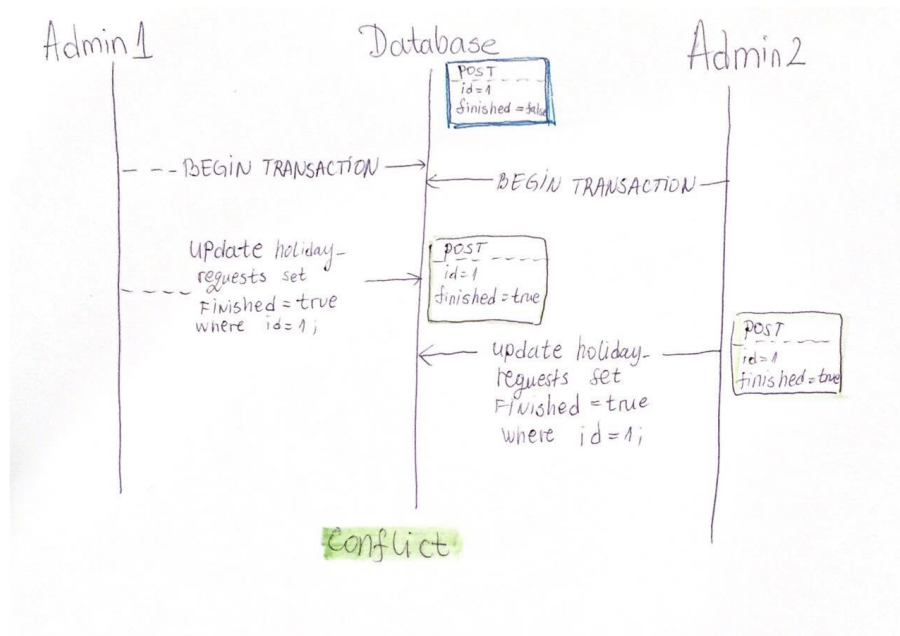
Posto je konflikt **dirty read** koristi se optimističko zaključavanje baze. Izolacija se stavlja na **COMMITTED**, jer ukoliko jedan pacijent rezervise pregled kod jednog doktora, a drugi pokuša da rezervise kod istog tog, desice se rollback, zato što je izmenjeno slobodno u koje je doktor slobodan.

3. Prihatanje i odbijanje zahteva za godisnji odmor (student 1)

3.1. Opis konfliktne situacije

Admini klinike imaju mogućnost da prihvataju i odbijaju zahteve za godišnji odmor medicinskog osoblja. Konflikt dolazi ukoliko dva ili više različitih admina prihvate ili odbiju isti zahtev za godišnji odmor. Jedan upit neće biti izvršen jer su podaci prethodno izmenjeni od strane drugog korisnika.

3.2. Crtez



3.3. Resenje

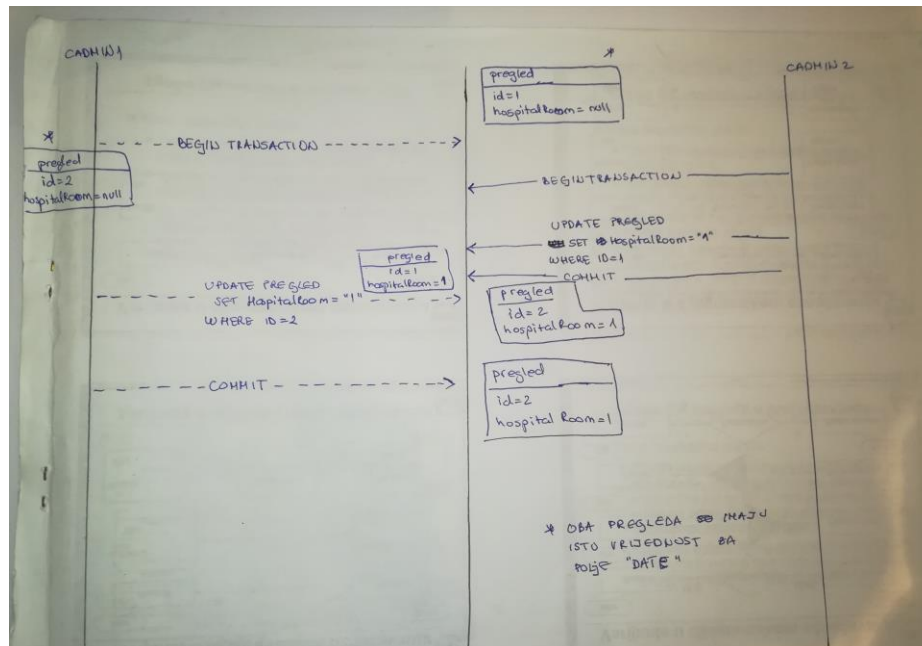
Konflikt se resave dodavanjem polja verzije u `HolidayRequest.java`. Ukoliko se pokuša izmena od strane dva admina, upravo zbog verzije neće moći obe operacije da se izvrše. Problem je `lost update` i resave se postavljanjem izolacije na `UNCOMMITTED`.

4. Prilikom odobravanja zahtjeva za pregled, jedna sala ne može biti rezervisana u isto vrijeme za različite preglede(student 2)

4.1. Opis konfliktne situacije

Jedna klinika može da ima više administratora klinike. Svaki administrator može da pristupi listi zahtjeva za pregled. Prilikom odobravanja zahtjeva za pregled bira se soba koja će biti dodijeljena pregledu. Ukoliko dva administratora pokušaju da odobre različit pregled zakazan u isto vrijeme u istoj sali, dolazi do konflikta.

4.2. Crtez



4.3. Resenje

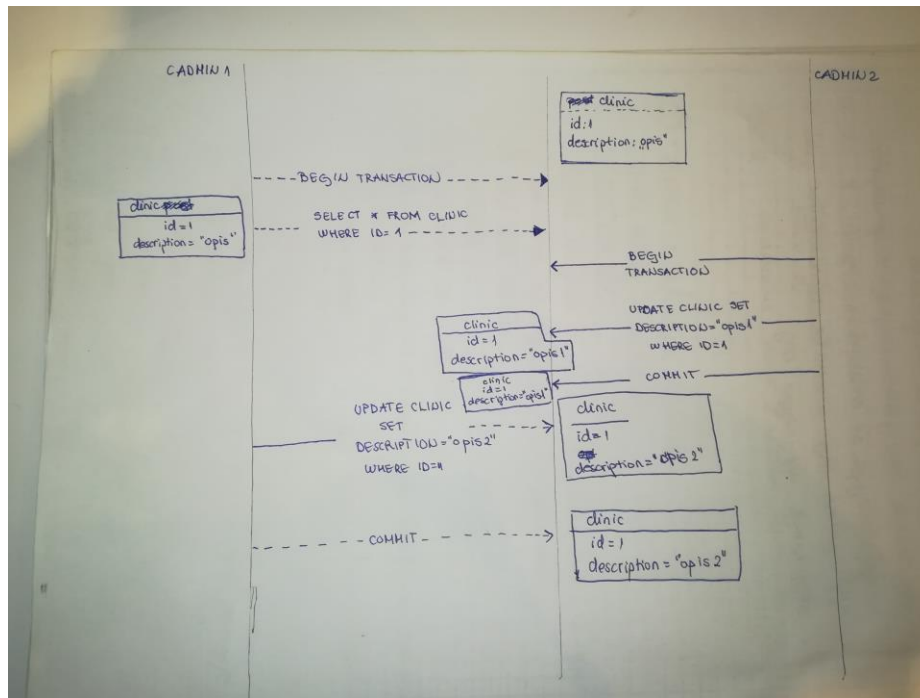
Za rjesenje je koristeno polje @Version dodato u tabelu HospitalRoom. Ukoliko u isto vrijeme dva administratora pokušaju da dodijele istu sobu nekom pregledu, samo onaj koji je prvi poslao upit će stvarno i rezervirati sobu za svoj pregled. Drugi administrator će dobiti obavještenje da se podaci o sobi trenutno mijenjaju i da mora da sačeka da bi radio sa validnim podacima. Ovo je omogućeno preko READ_UNCOMMITTED unutar oznake transakcije iznad metode za dodjeljivanje sobe pregledu.

5. Prilikom mijenjanja podataka o klinici (na primjer, dva administratora u isto vrijeme mijenjaju opis klinike) ne smije doći do istovremenog mijenjanja podataka (student 2)

5.1. Opis konfliktne situacije

Jedna klinika može da ima više administratora klinike. Jedna od konfliktnih situacija do kojih može doći radi ovoga jeste da dva administratora iste klinike mijenjaju podatke o toj klinici u isto vrijeme. Pošto drugi administrator nije svjestan o promjenama prvog, sistem ne bi trebalo da dopusti da on vrši dalje promjene nad podacima dok god se prve promjene ne izvrše. Pošto bi u ovom slučaju doslo do izmjene nad istim skupom podataka, ukoliko bi oba mijenjala u isto vrijeme promjene prvog bi bile pregazene.

5.2. Crtez



5.3. Resenje

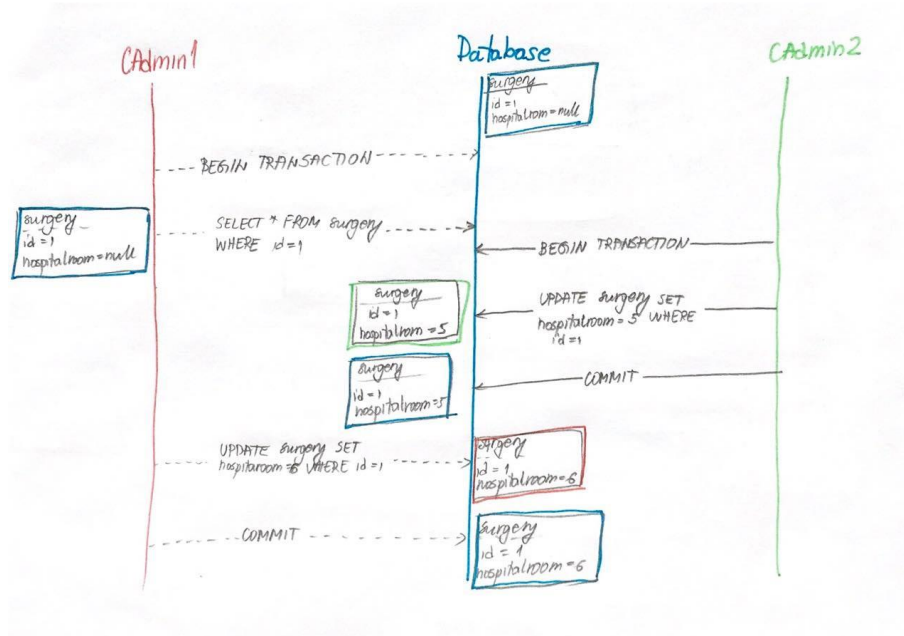
Unutar tabele klinika dodato je jos jedno polje, sa oznakom @Version . Ovim poljem moguće je izvršiti poredjenje verzije podataka s kojim radi korisnik ,sa podacima koji su u bazi. Ukoliko jedan administrator u tom trenutku salje izmjene ,potrebno je onemogućiti drugog da posalje svoje. Ovo je uradjeno dodavanjem oznake transakcije iznad metode za izmjenu podataka o klinici, unutar klase ClinicService.

6. Rezervacija sala za operacije (student 3)

6.1. Opis konfliktne situacije

Prilikom rezervacija sala može doći do konflikta ukoliko različiti admini za istu operaciju odaberu različitu salu. Jedna operacija ne može imati različite sale. Transakcije se tu koriste kako bi zaštitile rezervaciju sala koje ima operacija, problem je **lost update**. Update transakcije će biti pregazen, zato što druga transakcija nije bila svesna prethodne izmene.

6.2. Crtez



6.3. Resenje

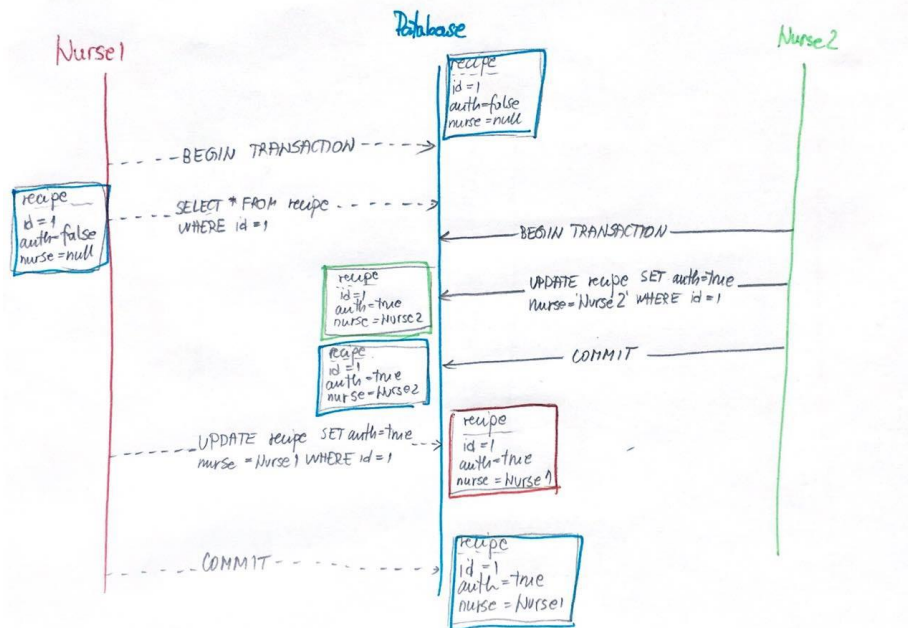
Transakcije su dodate pri snimanju u bazu. Korisceno je **optimisticko zakljucavanje**. Dodato je novo polje `@Version` koji broji izmene. Ukoliko prilikom update-a izmene nisu dobre transakcija ce vratiti rollback. Koriscena izolacija je **READ_UNCOMMITTED** kako bi se resio problem lost update.

7. Overavanje receptata (student 3)

7.1. Opis konfliktne situacije

Konflikt nastaje ukoliko dve sestre overe isti recept u isto vreme. Trebalo bi da se spreči jedno overavanje. Transakcija omogućava da se u bazu zabeleži samo jedna sestra da je overila taj recept, na drugom update-u ce se detiti rollback.

7.2. Crtez



7.3. Resenje

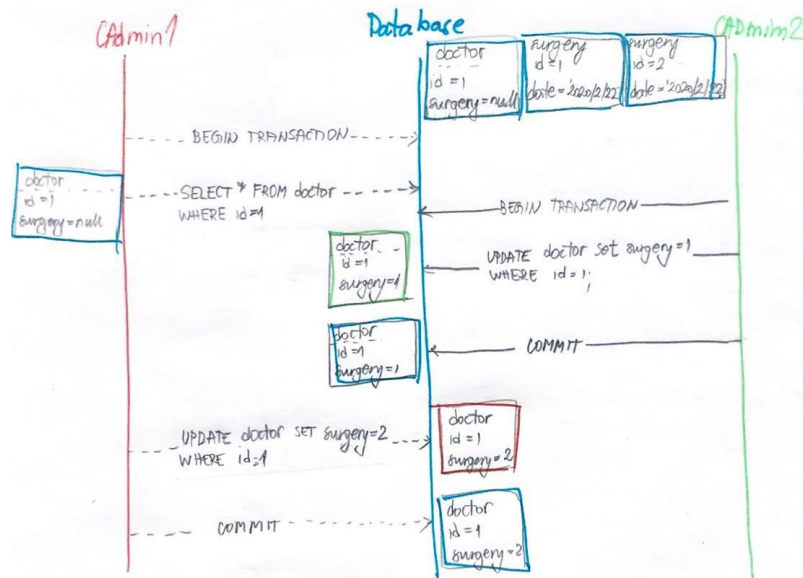
Konflikt se resava dodavanjem polja **@Version** u klasu **Recipe** i koristi se optimisticko zakljucavanje. Ukoliko druga sestra pokusa da izmeni taj isti recept tj. da ga overi, vratice joj se rollback upravo zbog bolja version, koji se povecava svaki put kad se taj recept pokusa menjati.

8. Jedan lekar ne moze da bude prisutan na vise razlicitih operacija(student 3)

8.1. Opis konfliktne situacije

Prilikom odabira sale, biraju se I lekari koji treba da prisustvuju toj operaciji koja je selektovana. Ukoliko se za dve razlicite operacije, koji su u isto vreme, odabere isti doktor onda dolazi do konflikta. Akcije druge transakcije ce pregaziti prvu. Problem koji se desava je **lost update**.

8.2. Crtez



8.3. Resenje

Doktoru se dodaje novo polje @Version koje prati verzije. Koristi se izolacija UNCOMMITTED jer moramo da procitamo da je neka operacija dodata doktoru I pre nego sto se ta transakcija zavrsi. Da se ne procita onda bi dtruga transakcija koja dolazi mlsila da je tad doktor slobodan u to vreme.