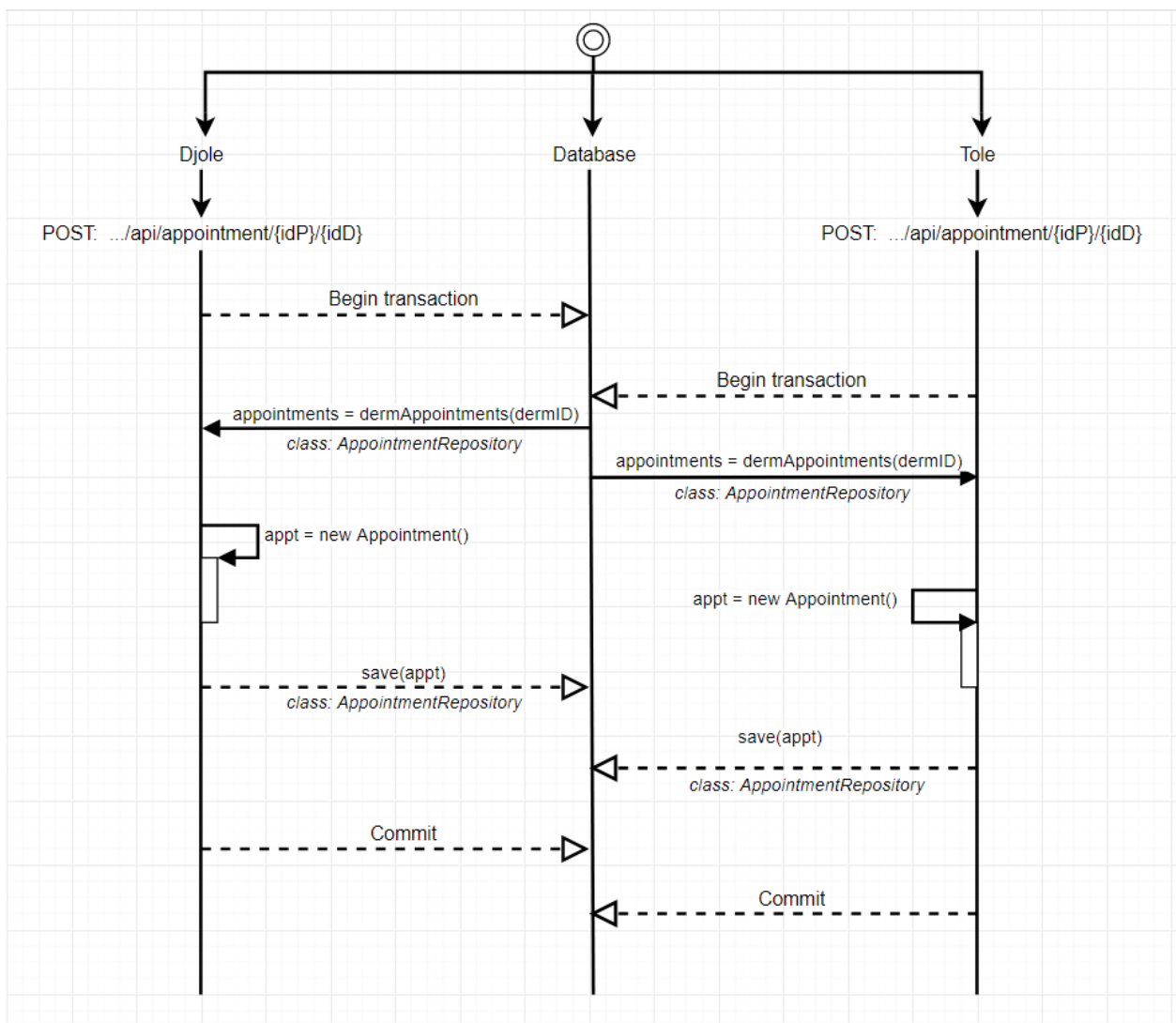


Konfliktna situacija 1: preklapanje termina pregleda dermatologa

- Ova konfliktna situacija se može javiti u dva slučaja – prilikom istovremenog zakazivanja pregleda dermatologa od strane više admina apoteke, ili u slučaju istovremenog zakazivanja pregleda dermatologa od strane samog dermatologa (nakon održanog pregleda) i admina apoteke.

Slučaj 1

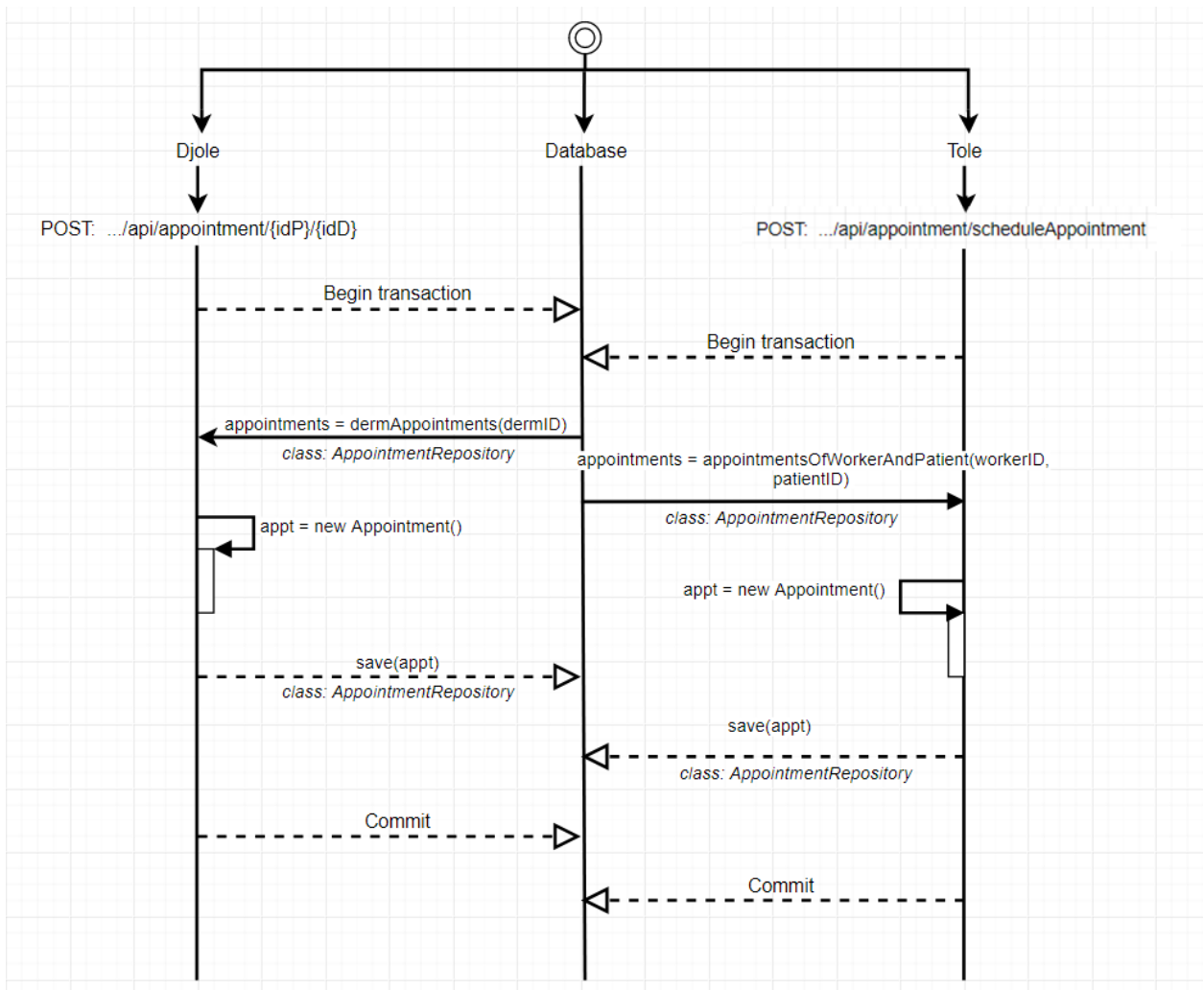


- Dva admina apoteke (*Djole* i *Tole*) u približno istom vremenskom trenutku pokušavaju da kreiraju pregled za određenog dermatologa. Do konflikta dolazi ukoliko obojica zakažu

preglede čija se vremena preklapaju – u tom slučaju bi dermatolog morao da prisustvuje na dva pregleda istovremeno.

- Kako su oba korisnika dobila iste rezultate prilikom pravljenja upita o postojećim pregledima dermatologa (koji su potrebni da klijent ne bi zakazao pregled koji se preklapa sa nekim već postojećim), klijent *Tole* nije svestan da je klijent *Djole* pre njega potencijalno zakazao sastanak u novom, slobodnom terminu koji se preklapa sa njegovim. U tom slučaju, nakon obavljanja *commit*-a obe akcije zakazivanja, u bazi podataka će se nalaziti podaci o nova dva pregleda čija će se vremena preklapati, što predstavlja konflikt.
- Kao rešenje ovog konflikta korišćen je pristup pesimističnog zaključavanja. Kako se radi o konfliktu koji nastaje prilikom dodavanja novih objekata u bazu, optimistični pristup nije od pomoći - u ovoj situaciji nije bitna verzija pojedinačnih sastanaka sačuvanih u bazi, s obzirom da se po specifikaciji projekta ne vrši nikakvo njihovo ažuriranje. Stoga, korišćen je pesimistički pristup prilikom čega dolazi do zaključavanja i čitanja i ažuriranja sastanaka dermatologa (*PESSIMISTIC_WRITE*, funkcija *AppointmentRepository::dermAppointments(Long workerID)*). Na taj način, vrši se zaključavanje pristupa postojećim sastancima dermatologu, te će zbog toga jedna od transakcija dva korisnika sa slike (u ovom slučaju korisnika *Tole*) morati biti zaustavljena dok prva ne oslobodi resure, čime će drugi klijent dobiti informaciju o kreiranju pregleda od strane prvog, te neće zakazati pregled u tom terminu.

Slučaj 2



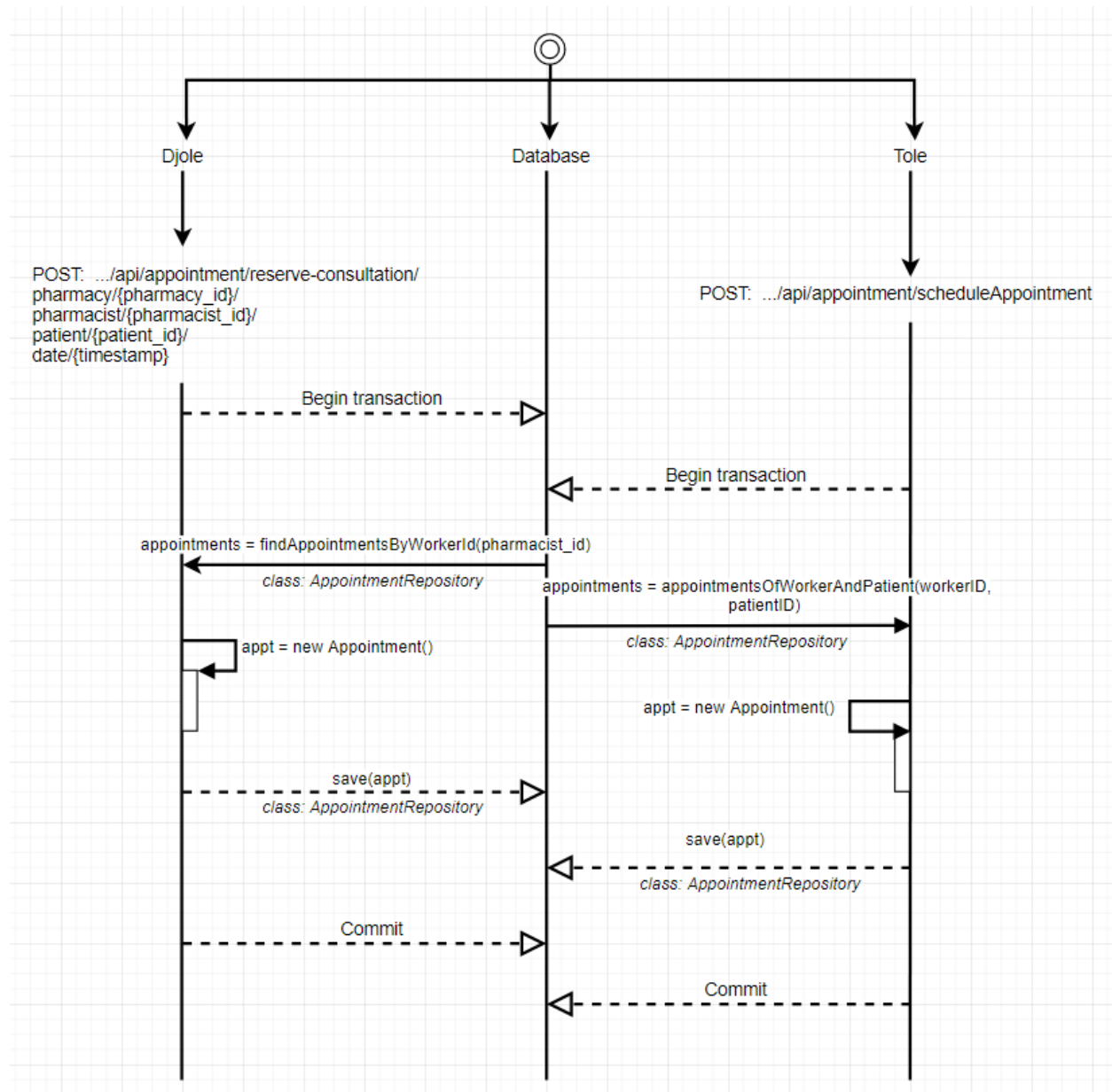
- Admin apoteke (*Djole*) i dermatolog (*Tole*) u približno istom vremenskom trenutku pokušavaju da kreiraju pregled za dermatologa (u ovom slučaju klijent *Tole*). Do konflikta dolazi ukoliko obojica zakažu pregleda čija se vremena preklapaju – u tom slučaju bi dermatolog morao da prisustvuje na dva pregleda istovremeno.
- U ovom slučaju, oba korisnika prave upit ka bazi da bi dobili podatke o postojećim pregledima dermatologa, kako ne bi kreirali novi pregled koji se preklapa sa već postojećim. Oba korisnika dobijaju isti skup podataka sastanaka dermatologa. U slučaju klijenta dermatologa, dobivljaju se takođe i pregledi pacijenta (da se novi pregled ne bi preklapao sa već postojećim klijentovim).
- Kako su oba korisnika dobila iste rezultate nakon upita o postojećim pregledima dermatologa (koji su potrebni da klijent ne bi zakazao pregled koji se preklapa sa nekim već postojećim), klijent *Tole* nije svestan da je klijent *Djole* pre njega potencijalno zakazao

sastanak u novom, slobodnom terminu koji se preklapa sa njegovim. U tom slučaju, nakon obavljanja *commit*-a obe akcije zakazivanja, u bazi podataka će se nalaziti podaci o nova dva pregleda čija će se vremena preklapati, što predstavlja konflikt.

- Kao rešenje ovog konflikta korišćen je pristup pesimističnog zaključavanja. Kako se radi o konfliktu koji nastaje prilikom dodavanja novih objekata u bazu, optimistični pristup nije od pomoći - u ovoj situaciji nije bitna verzija pojedinačnih sastanaka sačuvanih u bazi, s obzirom da se po specifikaciji projekta ne vrši nikakvo njihovo ažuriranje. Stoga, korišćen je pesimistički pristup prilikom čega dolazi do zaključavanja i čitanja i ažuriranja sastanaka dermatologa (*PESSIMISTIC_WRITE*) u okviru klase *AppointmentRepository* u funkcijama *dermAppointments (Long workerID)* i *appointmentsOfWorkerAndPatient(Long workerID, Long patientID)*. Iako se ne radi o istom upitu, obe funkcije će pokušati da pristupe istom skupu podataka – sastancima dermatologa sa prosleđenim ID-om (druga funkcija će pokušati da pristupi i pregledima pacijenta). Na taj način, vrši se zaključavanje pristupa postojećim sastancima dermatologu, te će zbog toga jedna od transakcija dva korisnika sa slike (u ovom slučaju korisnika *Tole*) morati biti zaustavljena dok prva ne oslobodi resure, čime će drugi klijent dobiti informaciju o kreiranju pregleda od strane prvog, te neće zakazati pregled u tom terminu.

Konfliktna situacija 2: preklapanje termina konsultacija farmaceuta

Ova konfliktna situacija se može javiti u dva slučaja – prilikom istovremenog zakazivanja konsultacija kod farmaceuta od strane više pacijenata, ili u slučaju istovremenog zakazivanja konsultacija kod farmaceuta od strane pacijenta i farmaceuta. Kako je prva konfliktna situacija već rešena (*Student2.pdf*), ovde će biti analizirana druga situacija.

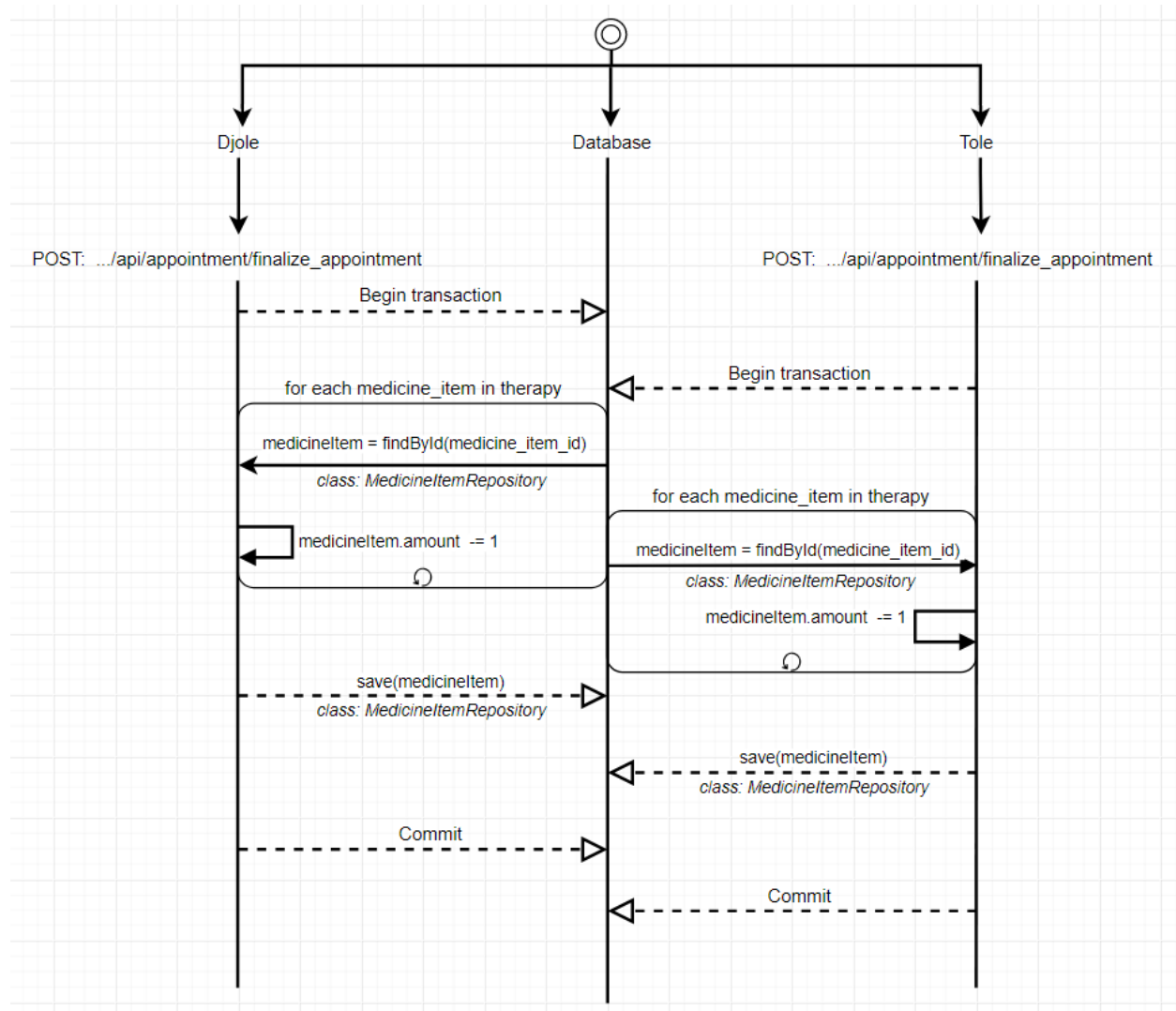


- Pacijent (*Djole*) i farmaceut (*Tole*) u približno istom vremenskom trenutku pokušavaju da kreiraju konsultacije za farmaceuta (u ovom slučaju klijent *Tole*). Do konflikta dolazi ukoliko obojica zakažu konsultacije čija se vremena preklapaju – u tom slučaju bi farmaceut morao da prisustvuje na dva sastanka istovremeno.

- U ovom slučaju, oba korisnika prave upit ka bazi da bi dobili podatke o postojećim konsultacijama farmaceuta, kako ne bi kreirali nove konsultacije koje se preklapaju sa već postojećim. Oba korisnika dobijaju isti skup podataka sastanaka farmaceuta. U slučaju klijenta farmaceuta, dobavljaju se takođe i pregledi pacijenta (da se novi sastanak ne bi preklapao sa već postojećim klijentovim).
- Kako su oba korisnika dobila iste rezultate nakon upita o postojećim konsultacijama farmaceuta (koji su potrebni da klijent ne bi zakazao konsultacije koje se preklapaju sa nekim već postojećim), klijent *Tole* nije svestan da je klijent *Djole* pre njega potencijalno zakazao sastanak u novom, slobodnom terminu koji se preklapa sa njegovim. U tom slučaju, nakon obavljanja *commit*-a obe akcije zakazivanja, u bazi podataka će se nalaziti podaci o nova dve konsultacije čija će se vremena preklapati, što predstavlja konflikt.
- Kao rešenje ovog konflikta korišćen je pristup pesimističnog zaključavanja. Kako se radi o konfliktu koji nastaje prilikom dodavanja novih objekata u bazu, optimistični pristup nije od pomoći - u ovoj situaciji nije bitna verzija pojedinačnih sastanaka sačuvanih u bazi, s obzirom da se po specifikaciji projekta ne vrši nikakvo njihovo ažuriranje. Stoga, korišćen je pesimistički pristup prilikom čega dolazi do zaključavanja i čitanja i ažuriranja sastanaka farmaceuta (*PESSIMISTIC_WRITE*) u okviru klase *AppointmentRepository* u funkcijama *findAppointmentsByWorkerId* (*Long pharmacist_id*) i *appointmentsOfWorkerAndPatient* (*Long workerID, Long patientID*). Iako se ne radi o istom upitu, obe funkcije će pokušati da pristupe istom skupu podataka – sastancima farmaceuta sa prosleđenim ID-om (druga funkcija će pokušati da pristupi i pregledima pacijenta). Na taj način, vrši se zaključavanje pristupa postojećim sastancima farmaceuta, te će zbog toga jedna od transakcija dva korisnika sa slike (u ovom slučaju korisnika *Tole*) morati biti zaustavljena dok prva ne oslobodi resure, čime će drugi klijent dobiti informaciju o kreiranju sastanka od strane prvog, te neće zakazati konsultacije u tom terminu.

Konfliktna situacija 3: kreiranje terapije za lek i menjanje njegovog stanja

Ova konfliktna situacija se javlja prilikom istovremenog kreiranja terapije za pacijente od strane više dermatologa ili farmaceuta (*endpoint* je isti za oba tipa korisnika). Prilikom ove akcije, može se desiti da oba zahteva sadrže bar jedan isti lek, a da se na kraju obe transakcije stanje tog leka promeni samo za 1 (umesto za 2 kao što bi trebalo).



- Dva dermatologa/farmaceuta (*Djole* i *Tole*) u približno istom vremenskom trenutku pokušavaju da kreiraju terapiju koja se sastoji od lekova, pri čemu se bar jedan lek nalazi u obe terapije. Do konflikta dolazi prilikom promene stanja leka, odnosno oba korisnika će videti istu količinu stanja leka X, i unutar svojih transakcija će je smanjiti za 1, a zatim novo stanje $X - 1$ upisati u bazu, čime će se stanje leka nakon obe terapije smanjiti za 1, umesto za 2, što dovodi do konflikta i greške u sistemu.
- Rešenje koje je iskorišćeno za rešavanje ovog problema je optimistično zaključavanje. Ovo rešenje je iskorišćeno jer se ova konfliktna situacija javlja u relativno retkim situacijama

(samo ukoliko dva ili više dermatologa/farmaceuta naprave zahtev za terapijom u sličnom vremenskom trenutku sa bar jednim zajedničkim lekom), te *rollback* u slučaju nepoklapanja verzija neće predstavljati veliki problem. Takođe, upravo zbog pomenutog ređeg javljanja ove situacije, nema potrebe zaključavati u potpunosti lekove apoteke pomoću pesimističnog zaključavanja, pogotovo uzevši u obzir da će time biti onemogućen pristup lekovima apoteke od strane neke druge, češće i ključnije operacije za sistem kao što je rezervacija lekova. Stoga, verzionisanje u okviru optimističnog pristupa se pokazalo kao najbolji pristup.

- Korišćenjem ovog pristupa, oba korisnika sa slike (*Djole* i *Tole*) će moći da pristupe lekovima dodeljenim terapiji, i da im smanje količinu za 1. Međutim, usled optimističkog pristupa i verzionisanja, nakon prvog *commit*-a doći će do izmene verzije leka apoteke, te će prilikom *commit*-ovanja druge transakcije doći do nepoklapanja verzija, što će dovesti do greške koja će sprečiti i onemogućiti pojavu ovog konflikta.