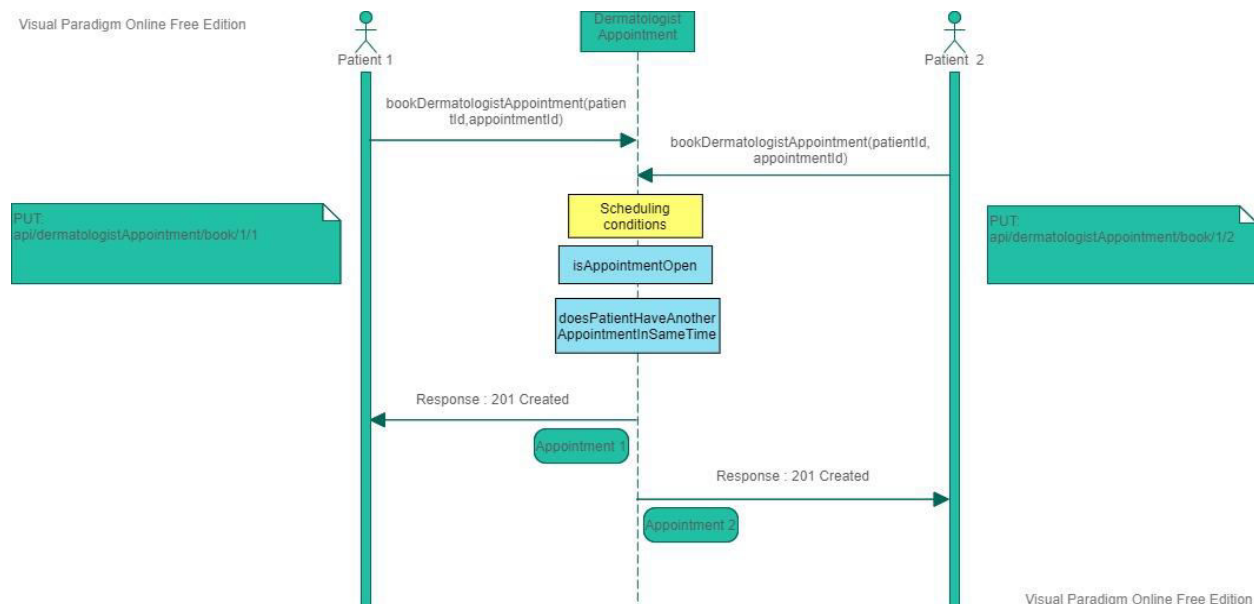


OPIS KONFLIKTNIH SITUACIJA

1. PREGLEDI KOJI SU UNAPRED DEFINISANI NE SMEJU BITI REZERVISANI OD STRANE VIŠE RAZLIČITIH KORISNIKA

Definisanim terminima dermatologa može pristupiti više korisnika u isto vreme. Konfliktna situacija se javlja ukoliko više korisnika pokuša da zakaze **isti termin** kod **istog dermatologa** u **isto vreme**. U ovoj situaciji, korisnik koji poslednji izmeni appointment u bazi će pregaziti prethodno postavljene vrednosti. Odnosno oba korisnika dobice poruku o uspesnom zakazivanju termina kod dermatologa u isto vreme što bismo svakako hteli da izbegnemo.



Slika toka koji izaziva opisanu konfliktnu situaciju

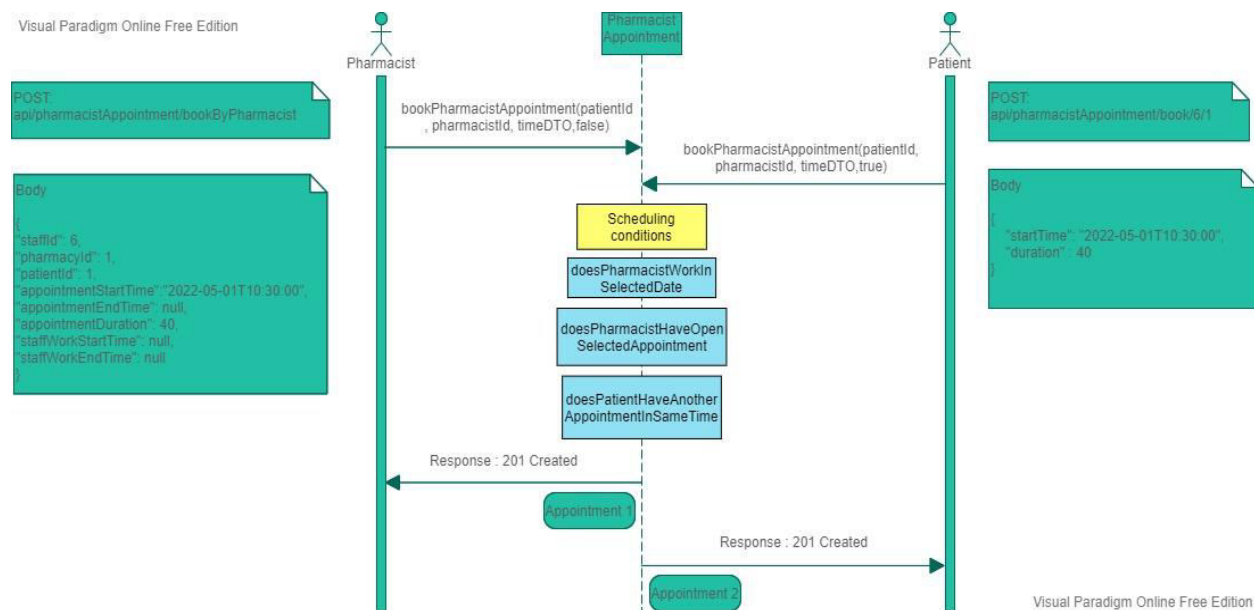
Pristup koji je upotrebljen za resavanje ove konfliktne situacije je optimističko zaključavanje. Kako ne bi dolazilo do konflikta, entitet Appointmenta (roditelj entiteta DermatologistAppointment) je verzionisan dodavanjem anotacije **@Version** na atribut "private Long Version". Pre završetka svake obrade zakazivanja termina dobavlja se trenutna verzija i ukoliko se ona poklapa sa verzijom koja je preuzeta na početku obrade, omogućeno je zakazivanje. U suprotnom korisnik koji pokuša cuvanje stare verzije entiteta, izazvace izuzetak **ObjectOptimisticLockingFailureException**, te ćemo ga na taj način obavestiti da je doslo do greske i da ne može da rezervise izabrani

termin. Dodata je i anotacija **@Transaction** iznad metode **bookDermatologistAppointment** cime obezbedjujemo da se baza vati u prethodno stanje u slucaju da dodje do gore napomenutog izuzetka.

Takodje postoji i metoda **bookDermatologistAppointmentTest** koja služi za testiranje gore navedene situacije, a test koji oponasa konflikt naziva se **testOptimisticLockingScenarioConflict1Student3**.

2. JEDAN FARMACEUT NE MOŽE ISTOVREMENO DA BUDE PRISUTAN NA VIŠE RAZLIČITIH SAVETOVANJA

Konfliktna situacija u kojoj farmaceut ima **vise zakazanih savetovanja u isto vreme**, javlja se zbog mogucnosti da vise korisnika zakaze isti termin savetovanja kod istog farmaceuta u isto vreme ili da farmaceut zeli da zakaze ponovni pregled pacijentu u istom trenutku kada i neki drugi pacijent zeli da zakaze termin.



Slika toka koji izaziva opisanu konfliktnu situaciju

Konfliktna situacija je resena pesimistickim zakljucavanjem odredjenog reda u tabeli Pharmacist. Prvi korisnik koji pristupi resursu ce ga i zakljucati i uspesno završiti transakciju, dok ce sledeci korisnik morati da saceka odredjeno vreme kako bi ponovo pokusao da zakaze. To smo postigli tako sto je iznad metode **findLockedById** dodata anotacija **Lock(LockModeType.PESSIMISTIC_WRITE)** i anotacija za timeout. Kada prva transakcija završi upis u tabelu, sledeci korisnik nastavlja svoju transakciju i ukoliko

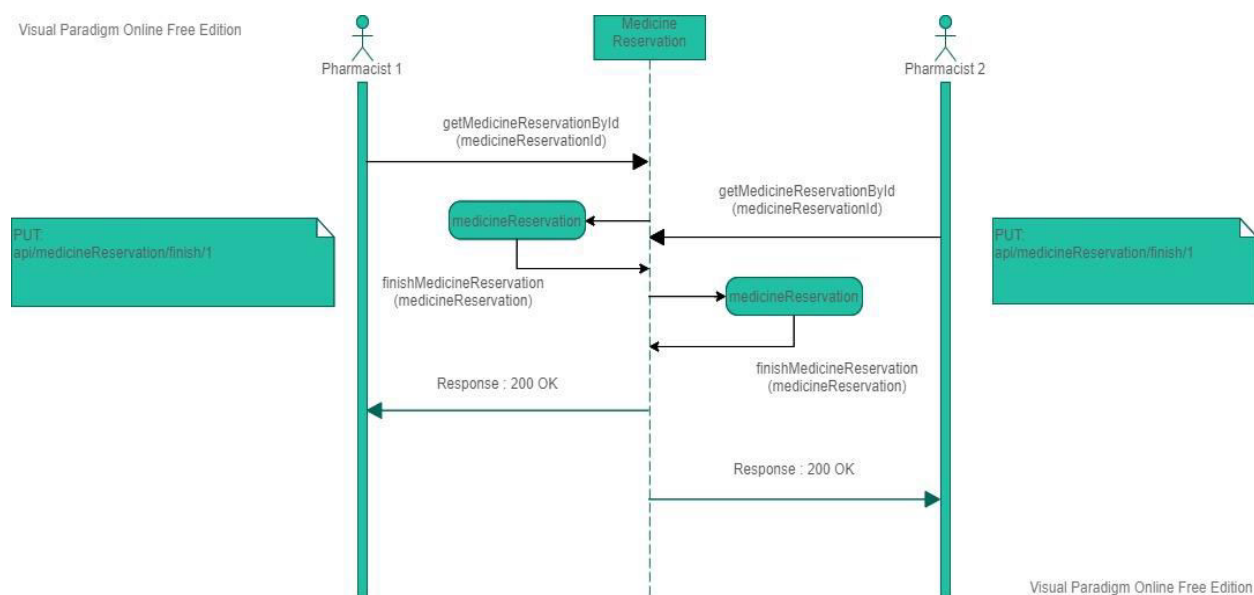
se desi da vec postoji savetovanje sa izabranim terminom, korisniku ce se ispisati poruka o greski.

Postoji i metoda **bookPharmacistAppointmentTest** koja služi za testiranje gore navedene situacije, a test koji oponasa konflikt naziva se **testPessimisticLockingScenarioConflict2Student3**.

3. DVA FARMACEUTA NE SMEJU ISTOVREMENO DA IZDAJU REZERVISANI LEK

Hipoteticki postoji mogucnost da rezervisani lek u **istoj apoteci** bude izdat **od dva razlicita farmaceuta istovremeno**, u tom trenutku ce doći do konfliktne situacije gde ce rezervacija biti 2 puta postavljena na **FINISH**. Korisnik ce preuzeti dva leka, a stanje leka u apoteci je umanjeno za jedan.

Primer: Moguce je da korisnik dodje u apoteku u kojoj rade dva farmaceuta, pokaze broj rezervacije jednom farmaceutu, zatim ode do drugog farmaceuta I njemu pokaze istu rezervaciju, nakon toga ponovo ode kod prvog farmaceuta I kaze da zeli da mu se izda rezervisani lek, takodje isto kaze I drugom farmaceutu. U ovoj situaciji pacijentu ce biti izdata dva leka, umesto jednog koliko je zapravo rezervisao.



Slika toka koji izaziva opisanu konfliktnu situaciju

Pristup koji je upotrebljen za resavanje ove konfliktne situacije je optimisticko zakljucavanje. Kako do ove situacije ne bi doslo, verzionisali smo entitet Medicine Reservation, dodavanjem anotacije **@Version** na atribut "private Long version". Na ovaj naci ce farmaceut koji prvi izmeni status rezervacije I upise u bazu, promeniti verziju

izabrane rezervacije leka. Drugi farmaceut koji je u medjuvremenu pokupio staru verziju rezervaciju iz baze prilikom izmene statusa i upisa u bazu dobice izuzetak `ObjectOptimisticLockingFailureException`. Da bi se baza vratila u prethodno stanje, kod pojave izuzetka `ObjectOptimisticLockingFailureException`, dodata je anotacija `@Transaction` iznad metode `finishMedicineReservation`.

`BookPharmacistAppointmentTest` je metoda koja služi za testiranje gore navedene situacije, a test koji oponasa konflikt naziva se `testOptimisticLockingScenarioConflict3Student3`.