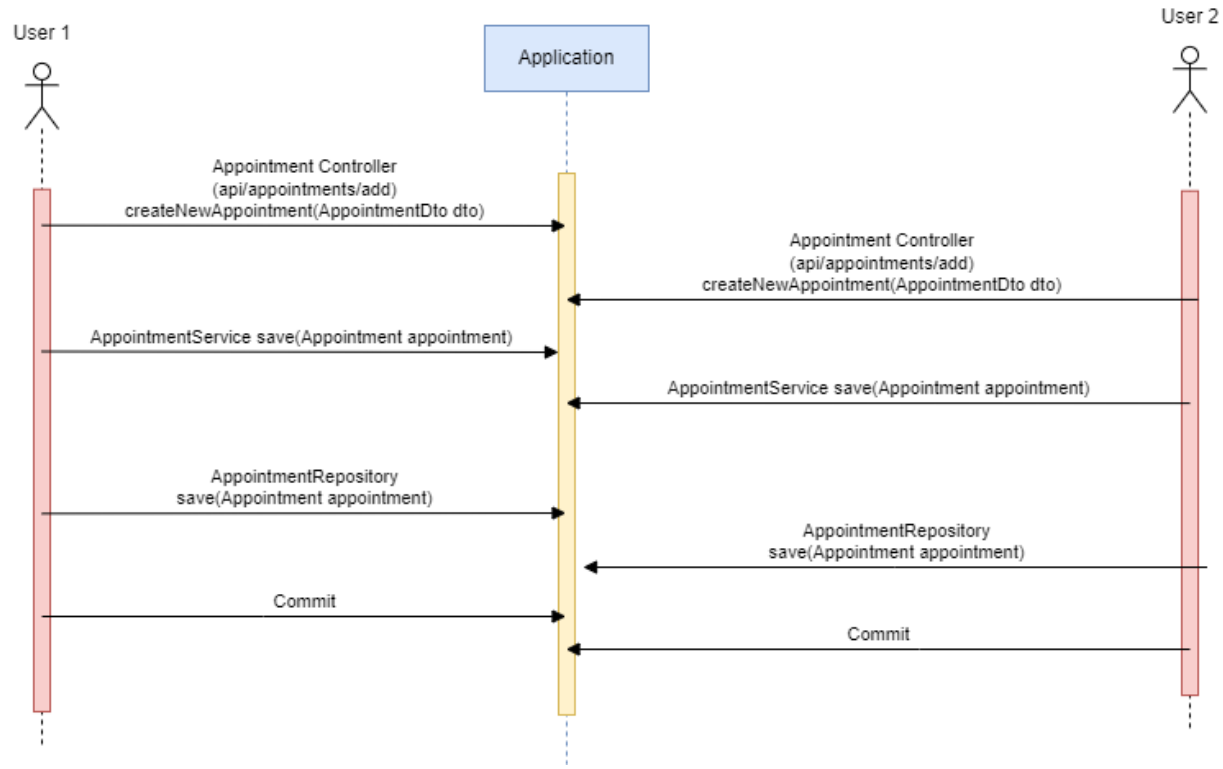
Konfliktne situacije – Student 3 (Darko Cokić RA 72/2019)

1) Jedan administrator centra/jedan predstavnik medicinskog osoblja ne može istovremeno da bude prisutan na više različitih pregleda

Opis konfliktne situacije: Medicinsko osoblje na svom naloga imaju mogučnost da unapred zakažu slobodne termine za davanje krvi. Do konflikta može doći ako dva korisnika, sa različitih uređaja, uloguju na isti nalog i pokušaju da naprave slobodan termin u isto ili čak preklapajuće vreme. To će dovesti do toga da se u bazi sačuvaju oba termina sa preklapajućim vremenom trajanja i da jedan zdravstevni bude u oba, što je fizički neizvodljivo.

Sequence dijagram situacije:



Opis rešenja: U ovom slučaju je korišćeno pesimističko zaključavanje. U *AppointmentRepository* klasi, metoda *findByBloodCenter_Id(Integer bloodCenterId)* je anotirana sa *@Lock(LockModeType.PESSIMISTIC_WRITE)* kako bi se transakcije izvršavale sekvencijalno i onemogućila drugim transakcijama da pišu i čitaju bazu sve dok se ona ne završi, i sa *@QueryHints({@QueryHint(name = "javax.persistence.lock.timeout", value ="0")})* kako bi mogla da baca *PessimisticLockingFailureException.* Takođe, metoda *save(Appointment appointment*) iz klase *AppointmentService* je anotirana sa *@Transactional(readOnly = false, propagation = Propagation.REQUIRES_NEW)* I implemntirana je tako da, ukoliko dođe do greške, baca SameMedicalStaffException.

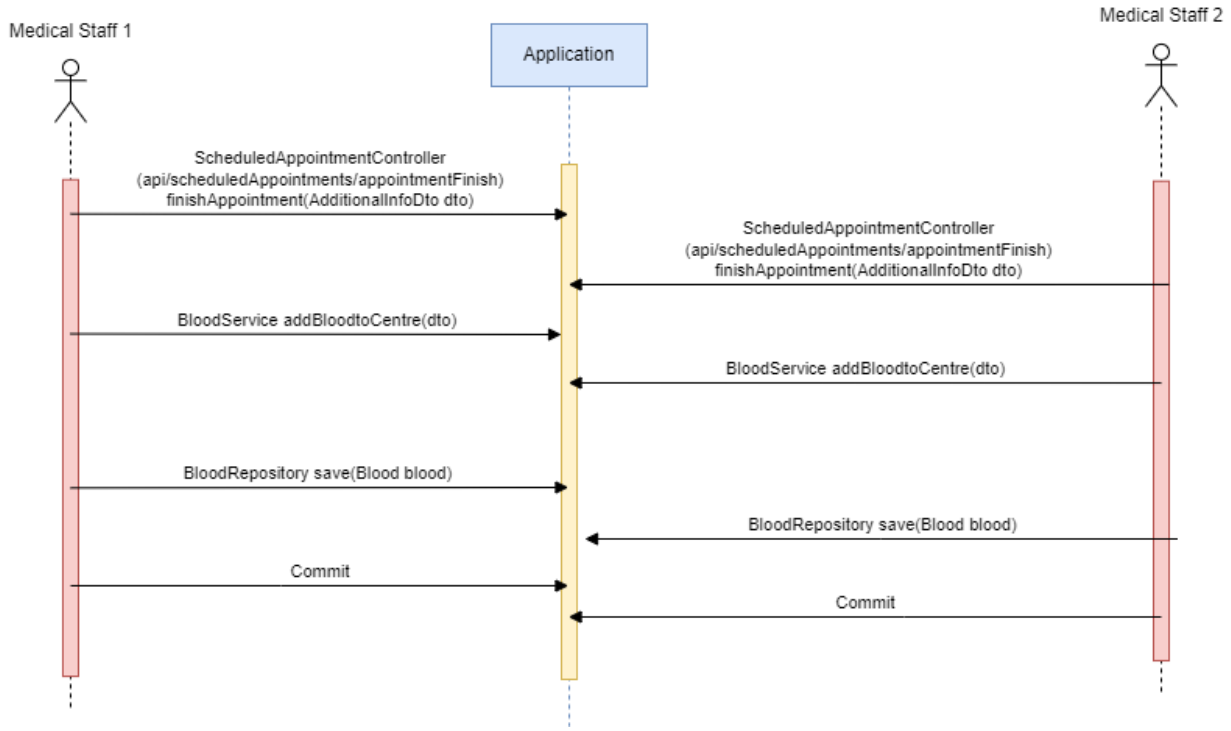
Testiranje: Ovde su koršćena dva testa. U prvom testu , učtaju se MedicalStaff i BloodCenter iz baze. Zatim se kreiraju dva objekta Appointment klase: a1 i a2, tako da imaju preklapajuće vreme i istog MedicalStaff-a. Zatim se kreiraju dve niti. Prvi nit pokušava da sačuva a1 u bazu, čime je zaključava. Druga nit će najpre biti uspavana na 40 milisekundi, pa će onda pokušati da sačuvati a2. Ovo joj neće poći za rukom jer je baza već zaključana, pa će biti bačen *PessimisticLockingFailureException.*

U drugom testu se na sličan način kao u prvom testu inicijalizuju podaci. Koriste se dve niti, i obe će pokušati da sčuvaju a1 i a2, respektivno. Ona koja prva krene će zaključati zaključati bazu i sačuvati termin, dok će ova druga bit blokirana i zatim uspavana na 3 sekunde, nakon čega će ponovo pokušati da sačuva termin. Pošto će biti preklapanja sa drugim terminom i imaće istog medicinskog radnika, javiće se *SameMedicalStaffException.*

2) Dva medicinska radnika pokušavaju da ažuriraju količinu iste krvne grupe

Opis konfliktne situacije: Medicinsko osoblje ima mogućnost da započne zakazani termin, unese potrbene podatke i da ih sumbit-uju, nakon čega se čuva izveštaj i ažurira se količina krvi u centru. Problem nastaje kada da ili više medicinska radnika nakon submit-ovanja ažuriraju količinu krvi iste grupe. Npr. ako oba medicinska radnika ažuriraju količinu A+ krvne grupe, a u centru ima 500 jedinica te krvi , može se desiti da jedan pročita 500 jedinica i poveća na 501, a da drugi isto pročita i poveća na 501, i time umesto da bude 502, bude 501 jedinica krvi, što se može jako loše odraziti na poslovanje datog centra.

Sequence dijagram situacije:



Opis rešenja: U ovom scenariju je korišćeno pesimističko zaključavanje. U *BloodRepository*-ju je metoda *getBloodByBloodCenterIdAndBloodType(int bloodCenterId, BloodType type);* anotirana sa @Lock(LockModeType.PESSIMISTIC_WRITE) kako bi zaključala bazu i time omogućila sekvencijalno izvšavanje i sa *@QueryHints({@QueryHint(name = "javax.persistence.lock.timeout", value ="0")})* kako bi bacala *PessimisticLockingFailureException.* U klasi *BloodService* se nalazi metoda *addBloodToCentre(AdditionalInfoDto additionalInfoDto)* koja je anotirana sa @Transactional(readOnly = false, propagation = Propagation.REQUIRES_NEW).

Testiranje: Ovde je korišćen jedan test. Najpre su inicijalizovani potrebni podaci za testiranje: jedan BloodCenter, dva MedicalStaff-a, dva RegistredUser-a, dva Appointment-a i dva ScheduledAppointment-a. Zatim se startuju dve niti, svaka inicijalizuje svoj AditiionalInfoDto. Druga nit biva uspavana na 40 milisekundi, dok prvi nit pokušava da ažurira stanje krvi. Nakon što se druga nit probudi, ona će pokušati da ažurira stanje krvi, ali, pošto je baza zaključana, javiće se *PessimisticLockingFailureException.*