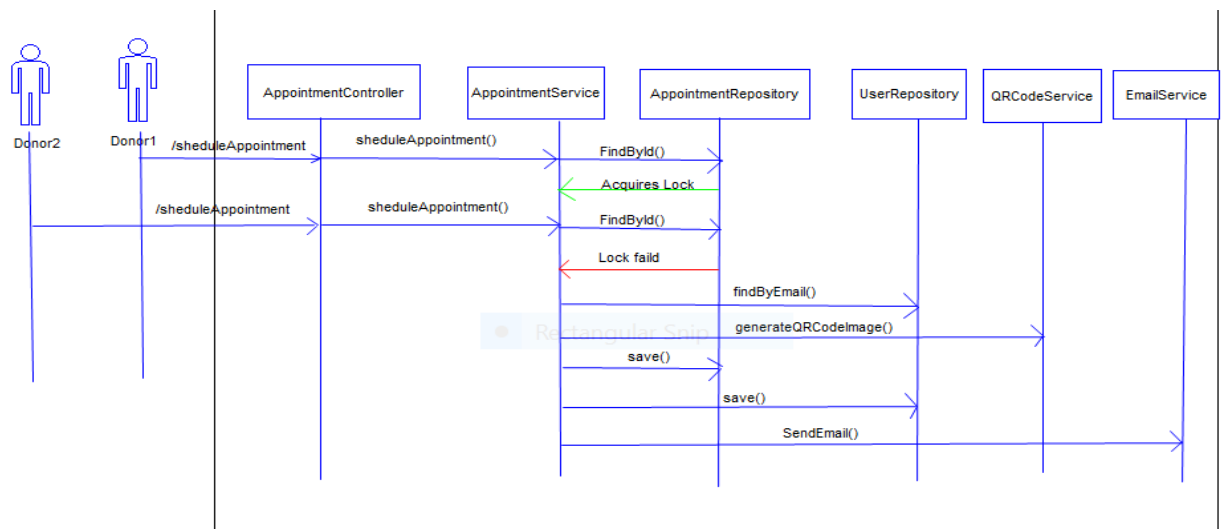


Konkurentni pristup resursima u bazi

1. Termini koji su unapred definisani ne smeju biti rezervisani od strane više različitih korisnika

Konfliktna situacija se može javiti ukoliko dva korisnika u isto vreme pokušaju da rezervisu predefinisani termin.



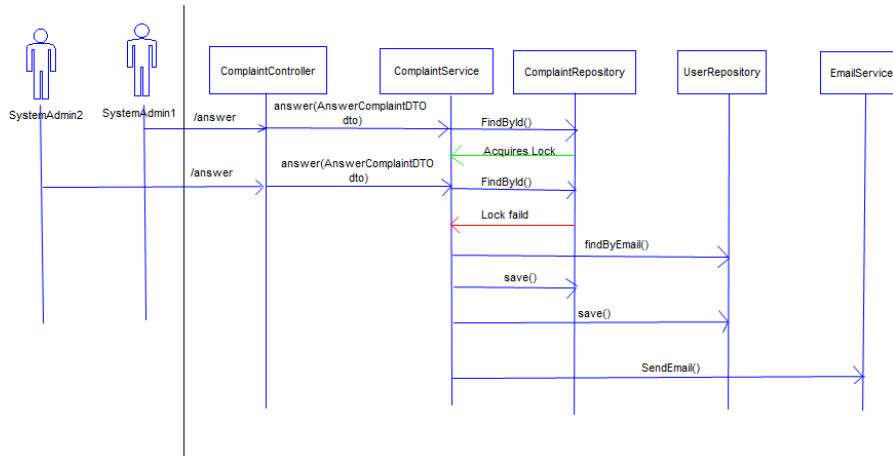
Koristi se Pessimistic Lock na objekat klase Appointment kako bi se sprecilo rezervisanje jednog termina od strane dva donora. U ovom slucaju prikazan je primer kada dva donora zele istovremeno da rezervisu predefinisane termine u centru za krv. Prvi donor uspesno rezervise termin dok je drugi onemogucen da to uradi.

U kodu je Pessimistic Lock stavljena na metodu FindById u AppointmentRepository, sto ce zakljucati red u tabeli Appointments i onemoguciti pristup dok se ne izvrši prva transakcija do kraja. Testirana je upotrebom dva thread-a u [ScheduleAppointmentTwoDonorsProblem](#).

```
new *
@Lock(LockModeType.PESSIMISTIC_WRITE)
@QueryHints({ @QueryHint(name = "javax.persistence.lock.timeout", value = "0") })
Optional<Appointment> findById(Long id);
```

2. Na jednu žalbu može da odgovori samo jedan administrator sistema

Konfliktna situacija nastaje ukoliko dva administratora sistema pokušaju da odgovor na istu žalbu. U bazu će biti sačuvan odgovor onog koji pobedi i prvi pristupi podacima, što nije dobar pristup.

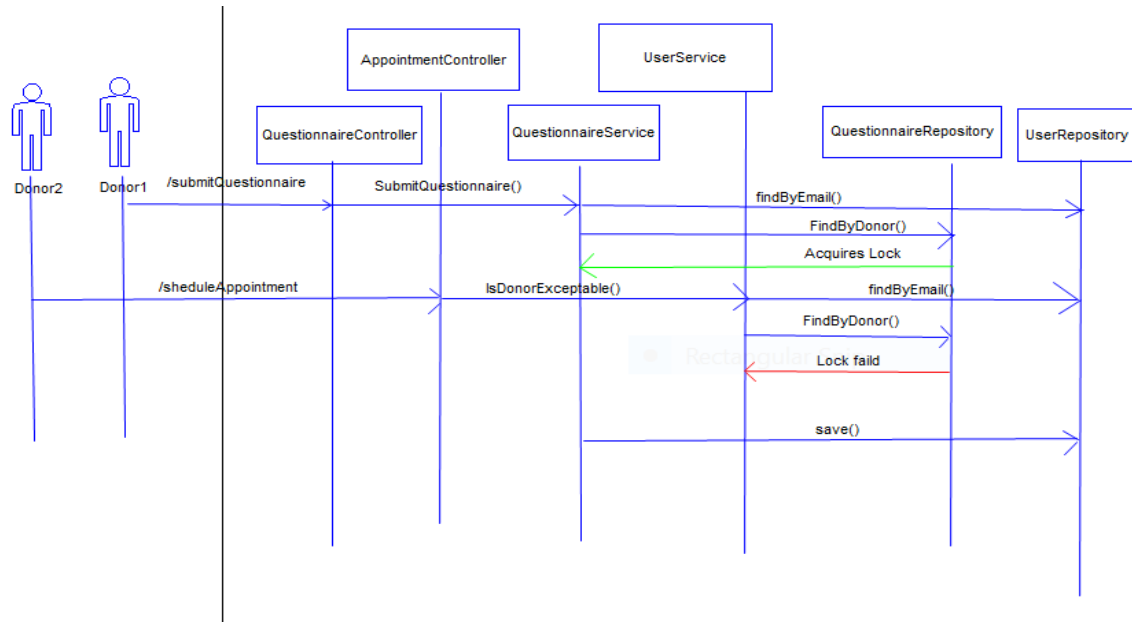


U kodu je Pessimistic Lock stavljena na metodu `findById` u `ComplaintRepository`, što će zaključati red u tabeli `Complaints` i onemogućiti pristup dok se ne izvrši prva transakcija do kraja. Testirana je upotrebom dva thread-a u `AnswerComplaintTwoAdminsProblem`.

```
new *
= @Lock(LockModeType.PESSIMISTIC_WRITE)
@QueryHints({ @QueryHint(name = "javax.persistence.lock.timeout", value = "0") })
Optional<Complaint> findById(Long id);
```

3. Dobavljanje informacije da li korisnik ima popunjen upitnik u trenutku kada ga popunjava

Prilikom zakazivanja predefinisano termina potrebno je proveriti da li korisnik ima popunjen upitnik, a to zahteva dobavljanje upitnika registrovanog donora. Istoj funkciji pristupa se i kada se upitnik popunjava pa moze doci do konfliktne situacije.



U kodu je Pessimistic Lock stavljena na metodu FindByDonorId u QuestionnaireRepository, sto ce zakljucati red u tabeli Questionnaires i onemoguciti pristup dok se ne izvrši prva transakcija do kraja. Testirana je upotrebom dva thread-a u [NewQuestionnaireShedulingAppointementProblem](#).

```
2 usages new *
@Lock(LockModeType.PESSIMISTIC_WRITE)
@QueryHints({ @QueryHint(name = "javax.persistence.lock.timeout", value = "0") })
Questionnaire findById(Long id);
```