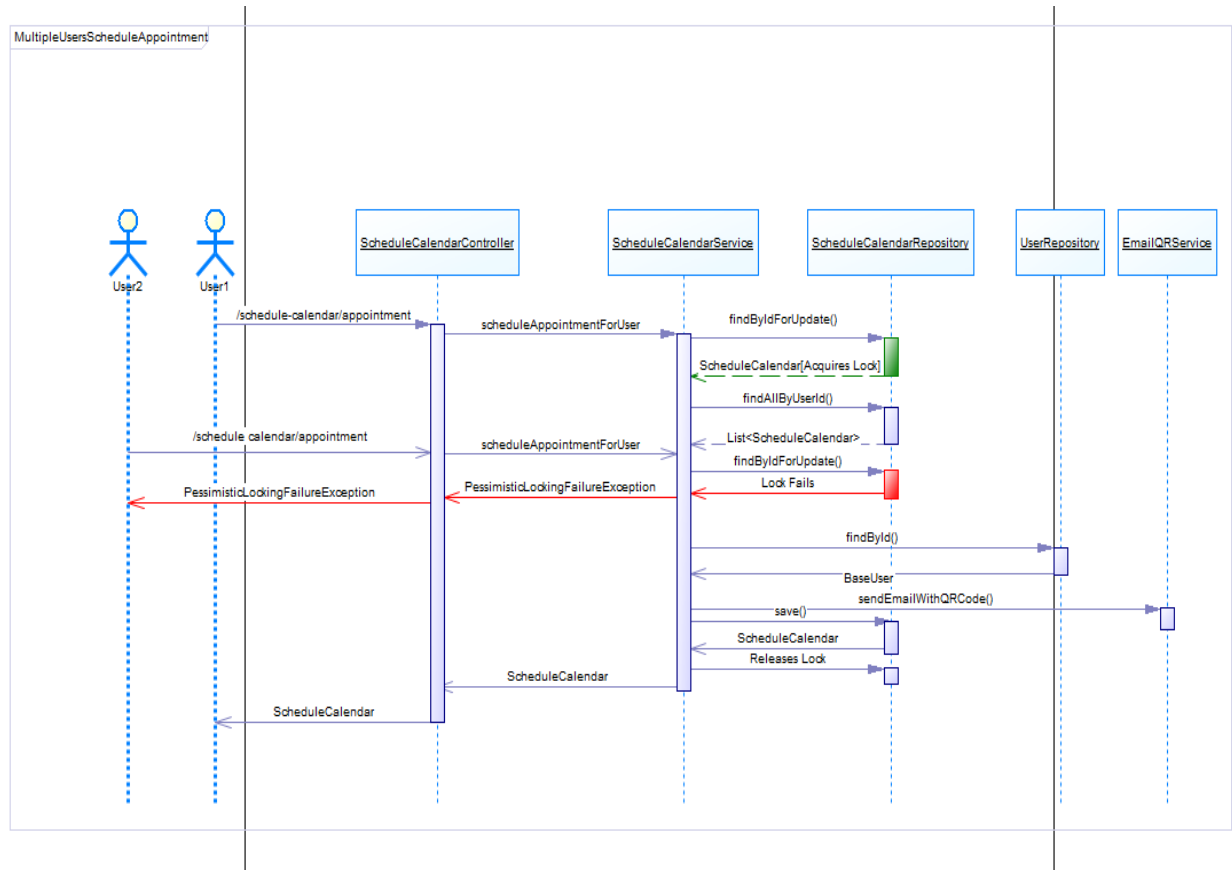


Konfliktne situacije - Boban

Opis

- više istovremenih korisnika aplikacije ne može da rezerviše termin koji je u međuvremenu postao nedostupan



Pristup

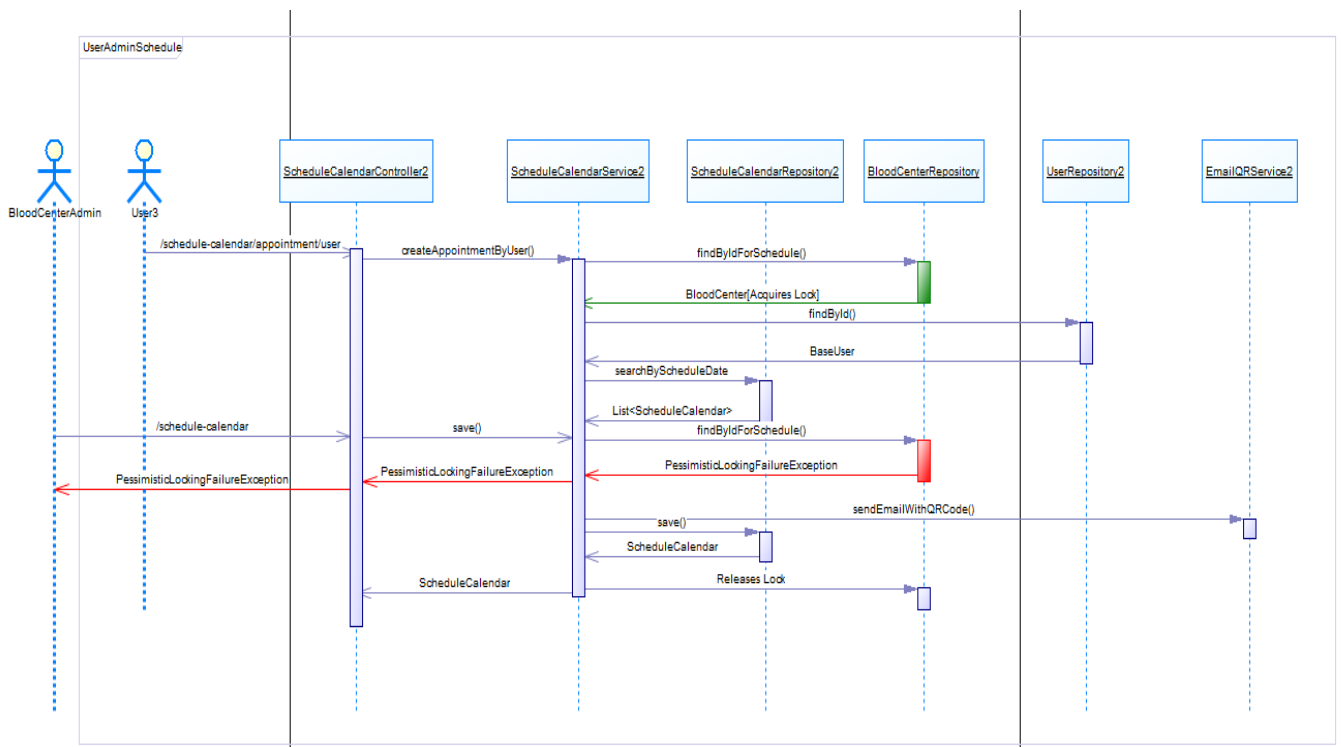
- resavano je preko Pessimistic Lock-a koji je stavljen za ScheduleCalendar objekat kada se trigeruje `findByIdForUpdate()` metoda. Pristup je odabran kako bi se sprecila istovremena izmena istog objekta.
- Implementacija u kodu:

```
@Query("SELECT sc FROM ScheduleCalendar sc WHERE sc.id = :id")
@Lock(LockModeType.PESSIMISTIC_WRITE)
@QueryHints({ @QueryHint(name = "javax.persistence.lock.timeout",
value = "0") })
Optional<ScheduleCalendar> findByIdForUpdate(@Param("id") Long id);
```

- Test se nalazi u fajlu `ScheduleCalendarPesimisticTest.java`

Opis

- administrator centra ne može unapred definisati termin u isto ili preklapajuće vreme za koje i korisnik kreira rezervaciju termina



Pristup

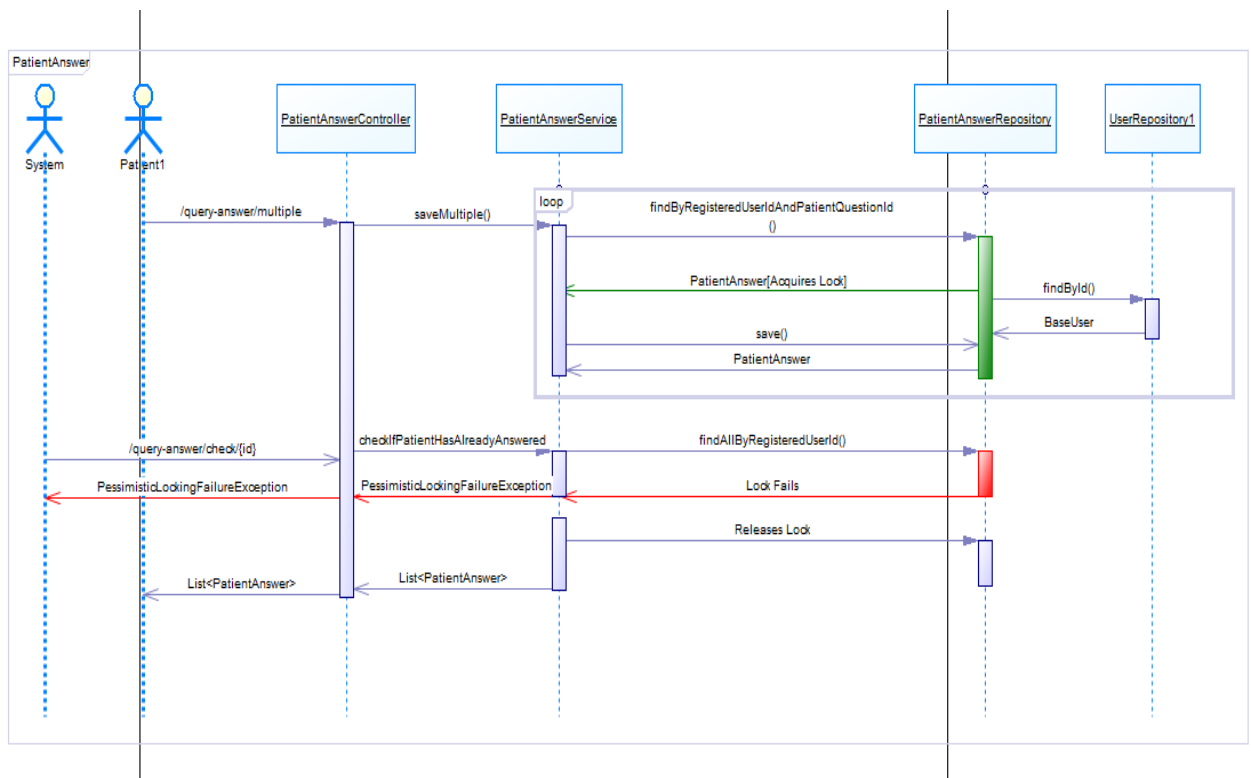
- resavano je preko Pessimistic Lock-a koji je stavljen za BloodCenter objekat kada se trigeruje *findByIdForSchedule()* metoda. Pristup je odabran kako bi se sprecila istovremeno kreiranje *ScheduleCalendar* objekata za isti *BloodCenter*.
- Implementacija u kodu:

```
@Query("select bc from BloodCenter bc where bc.id = :id")
@Lock(LockModeType.PESSIMISTIC_WRITE)
@QueryHints({ @QueryHint(name = "javax.persistence.lock.timeout", value =
"0") })
Optional<BloodCenter> findByIdForSchedule(Long id);
```

- Test se nalazi u fajlu *ScheduleCalendarUserAdminTest.java*

Opis

- ne može da se vrši provera da li je korisnik odgovorio na upitnik u isto vreme kada korisnik azurira odgovore na upitnik



Pristup

- Resavano je preko Pessimistic Lock-a koji je stavljen za PatientAnswer objekat kada se trigeruje `findByRegisteredUserIdAndPatientQuestionId()` ili `findAllByRegisteredUserId()` metoda. Pristup je odabran kako bi se sprecila istovremeno cekiranje da odgovora na upitnik i izmena odgovora upitnika.
- Implementacija u kodu:

```
@Lock(LockModeType.PESSIMISTIC_WRITE)
@QueryHints({ @QueryHint(name = "javax.persistence.lock.timeout", value = "0") })
List<PatientAnswer> findAllByRegisteredUserId(Long id);

@Lock(LockModeType.PESSIMISTIC_WRITE)
@QueryHints({ @QueryHint(name = "javax.persistence.lock.timeout", value = "0") })
PatientAnswer findByRegisteredUserIdAndPatientQuestionId(Long id, Long id1);
```

- Test se nalazi u fajlu *PatientAnswerTest.java*