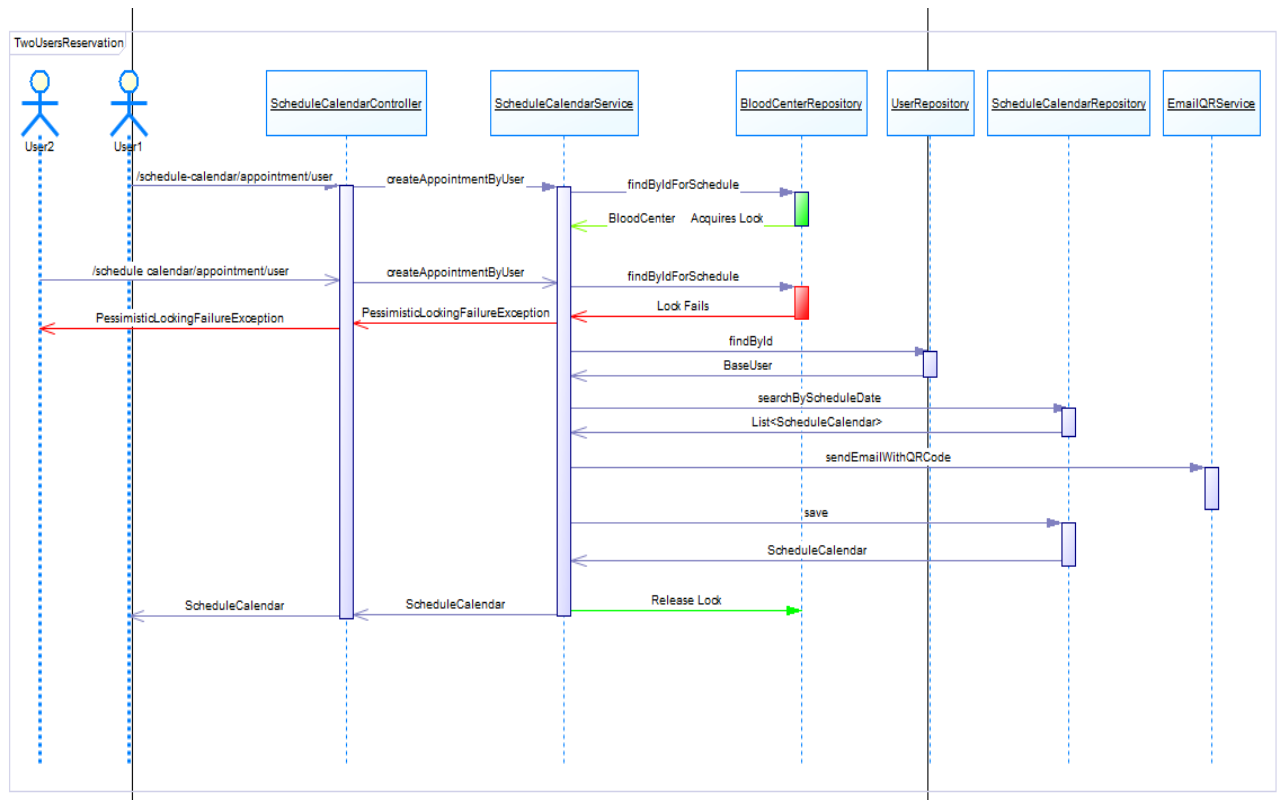


- više istovremenih korisnika aplikacije ne može da rezerviše termin koji je u međuvremenu postao nedostupan



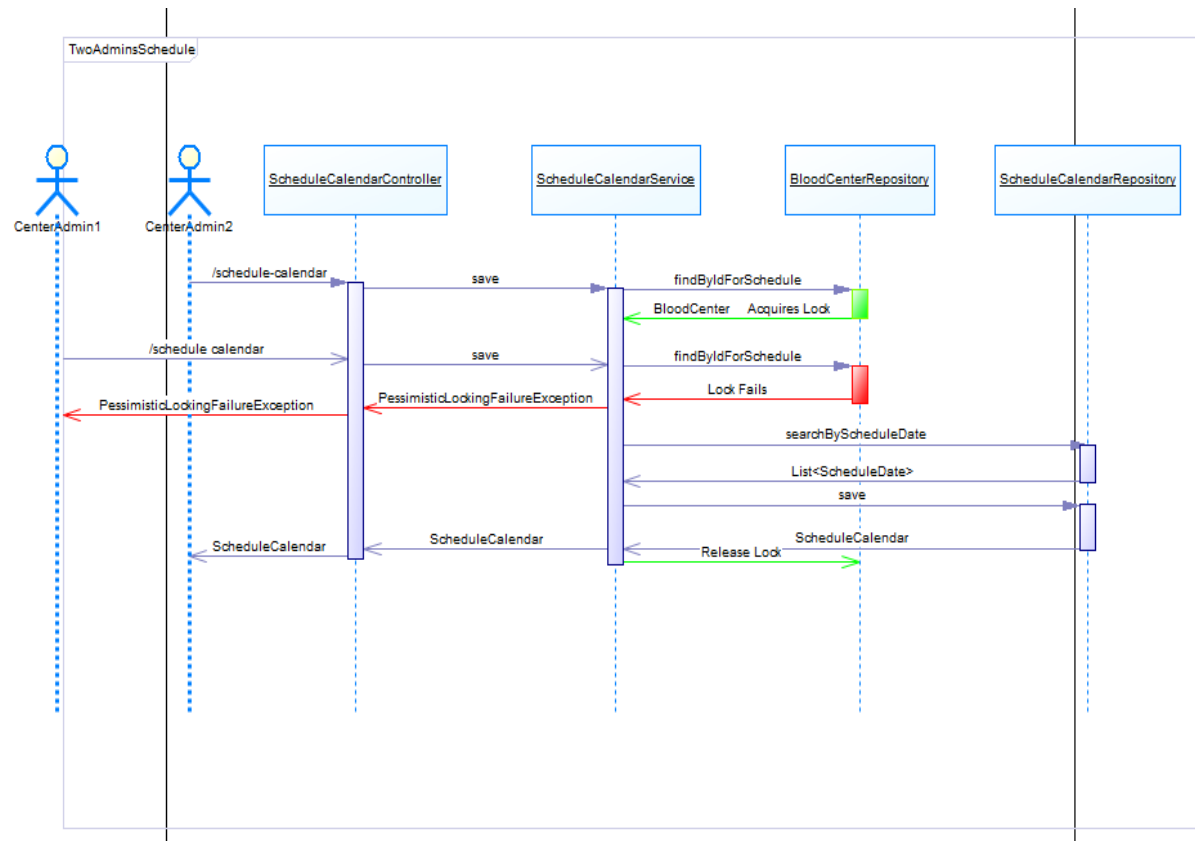
- Koristi se Pessimistic Lock na objekat od BloodCenter klase kako bi se onemogućilo istovremeno rezervisanje termina u istom centru za davanje krvi. U ovom slučaju prikazan je primer kada dva korisnika zele istovremeno da rezervisu termine u centru za krv. Prvi korisnik uspešno rezervise termin dok je drugi onemogućen da to uradi.
- 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 primer se nalazi u *ScheduleCalendarTwoUsersTest.java* fajlu.

- više administratora centra ne mogu unapred definisati termine u isto ili preklapajuće vreme

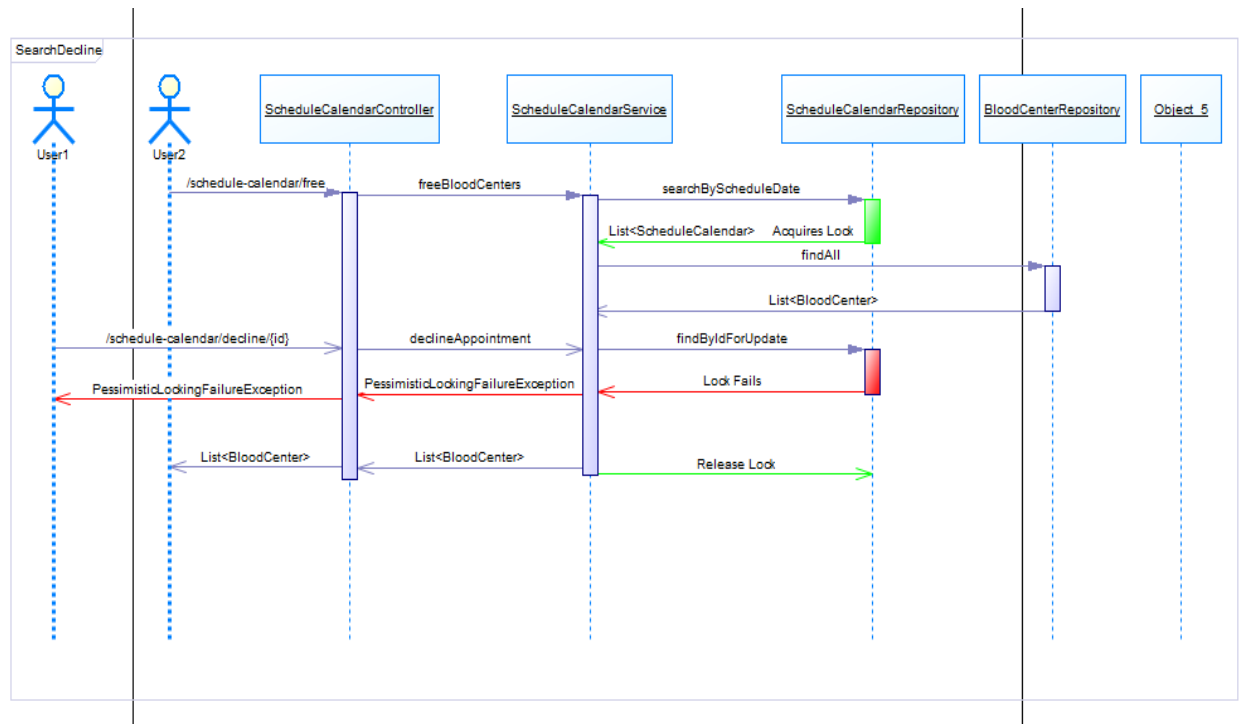


- Koristi se Pessimistic Lock na objekat od BloodCenter klase kako bi se onemogućilo istovremeno kreiranje slobodnih termina u istom centru za davanje krvi. U ovom slučaju prikazan je primer kada dva administratora zele istovremeno da kreiraju slobodne termine u centru za krv. Prvi administrator uspešno kreira slobodan termin dok se drugom onemogućava ta radnja
- 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 primer se nalazi u *ScheduleCalendarTwoAdminsTest.java* fajlu.

3. dobavljanje liste centara u isto vreme kada korisnik otkazuje termin



- U ovom slučaju imamo dva korisnika gde jedan želi da dobavi centre koji su slobodni za rezervaciju u specifikiranom vremenu, a drugi korisnik u to isto vreme otkazuje termin. Ovaj slučaj je implementiran na taj način da se koristi Pessimistic Lock na ScheduleCalendar objekte.
- Implementacija u kodu:

```

@Query("SELECT sc FROM ScheduleCalendar sc WHERE " +
        "sc.scheduleDate = :date")
@Lock(LockModeType.PESSIMISTIC_WRITE)
@QueryHints({ @QueryHint(name = "javax.persistence.lock.timeout",
value = "0") })
public List<ScheduleCalendar> searchByScheduleDate(
        @Param("date") LocalDate date);

@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 primer se nalazi u *ScheduleCalendarSearchDeclineTest.java* fajlu.