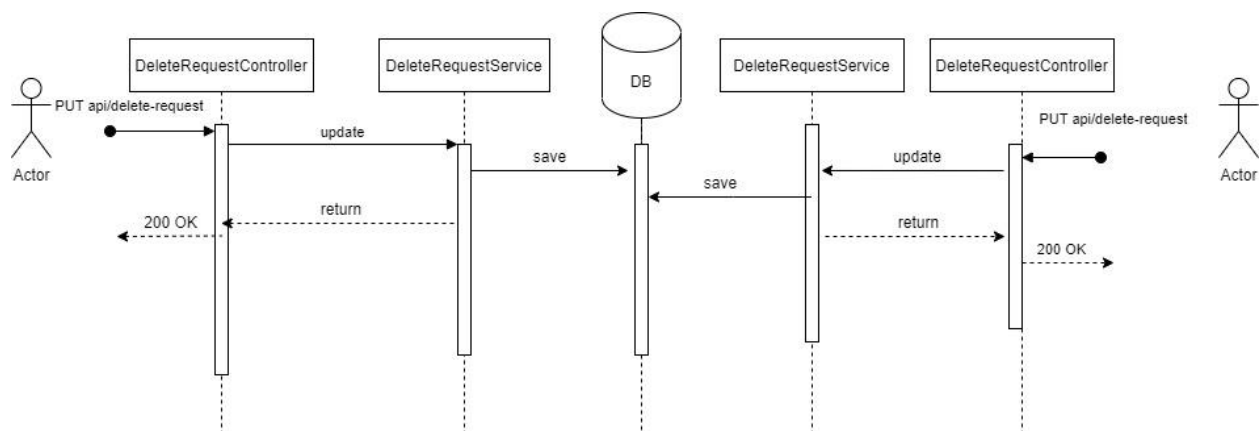


Конкурентни приступ – Александар Бркљач РА225/2018

1. На један захтијев за брисање налога може да одговори само један администратор система

Конфликтна ситуација подразумева да истовремено два администратора желе да одговоре на исти захтијев за брисање корисничког налога. На један захтијев за брисање налога потребан је само један одговор и потребно је онемогућити додатне измијене.



Слика 1: Графички приказ прве конфликтне ситуације

Са секвенционалног дијаграма се види неки главни ток одвијања конфликтне ситуације с тим да су изостављени поједини детаљи који се тичу логике ради једноставнијег приказа.

Ситуација је ријешена помоћу оптимистичког закључавања. У класу *DeleteRequest* додата је *version* промијенљива која служи за верзионисање торке у бази. Метода у сервисном нивоу апликације помоћу које се мијења стање захтијева означена је као трансакциона.

```
1
2 @Entity
3 public class DeleteRequest {
4
5     @Id
6     @Column(name = "delete_request_id")
7     @GeneratedValue(strategy = GenerationType.IDENTITY)
8     private int id;
9     private String reason;
10    private Status requestStatus;
11
12    @OneToOne
13    private User user;
14
15    @OneToOne
16    @JoinColumn(nullable = true)
17    private Admin adminResponded;
18
19    @Version
20    private Long version;
21 }
```

Слика 2. Доменска класа захтијева за брисање налога

```

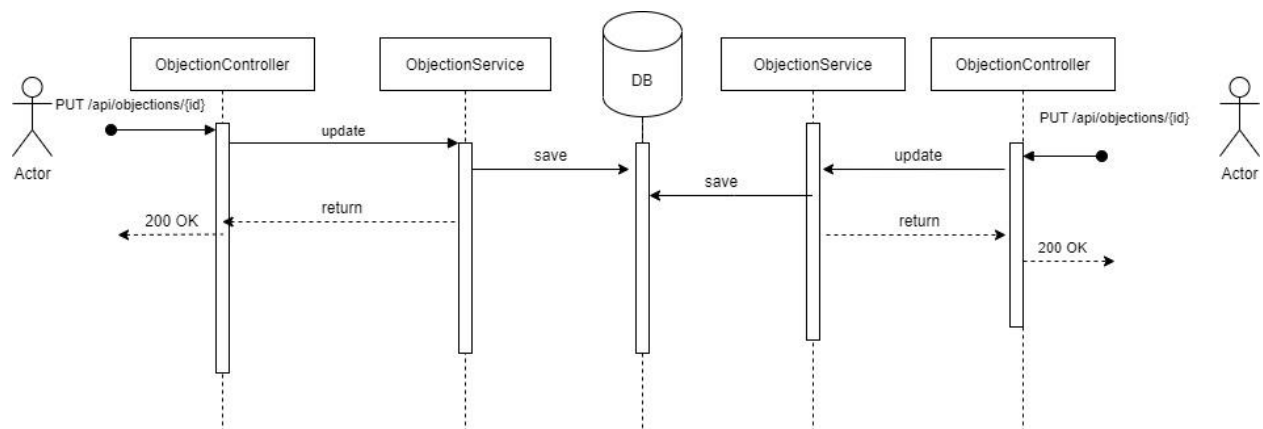
@Override
@Transactional
public void update(DeleteRequest item, String adminResponse, Admin admin) {
    DeleteRequest dbRequest = getById(item.getId());
    if(!dbRequest.getRequestStatus().equals(Status.PENDING)) throw new DeleteRequestResponded();
    if (item.getRequestStatus().equals(Status.ADMIN_CONFIRMED))
        this.updateUserProfileStatus(item.getUser().getId());
    dbRequest.setRequestStatus(item.getRequestStatus());
    dbRequest.setAdminResponded(admin);
    this.update(dbRequest);
    emailService.sendDeleteRequestResponseNotification(userService.getById(item.getId()), item.getRequestStatus(),
        adminResponse);
}

```

Слика 3. Метода за ажурирање статуса захтијева

2. На једну жалбу може да одговори само један администратор система

Конфликтна ситуација је слична као и претходна. Потребно је спријечити преписивање једног одговора од стране другог администратора приликом конкурентних захтијева.



Слика 4: Графички приказ друге конфликтне ситуације

Ситуација је ријешена као и у претходном примјеру.

```

1
2 @Entity
3 public class Objection {
4     @Id
5     @GeneratedValue(strategy = GenerationType.IDENTITY)
6     private Integer id;
7     @ManyToOne
8     @JoinColumn(name = "appointment_id")
9     private Appointment appointment;
10    private String objection;
11    @OneToOne
12    private User creator;
13    @OneToOne
14    private Admin adminResponded;
15    private String response;
16    @Version
17    private Long version;
18 }

```

Слика 5. Доменска класа жалбе са промијењивом за верзионисање

```

@Override
@Transactional
public void addAdminResponse(Objection o,String adminResponse,Admin admin) {
    Objection objection=this.getById(o.getId());
    if(objection.getAdminResponded()!=null) throw new RequestHasResponseException();
    objection.setAdminResponded(admin);
    objection.setResponse(adminResponse);
    emailService.sendObjectionResponseNotification(objection.getUserEmail(), adminResponse);
    emailService.sendObjectionResponseNotification(objection.getAppointment().getOwner().getEmail(), adminResponse);
    this.update(objection);
}

```

Слика 6. Метода за додавање одговора на захтијев