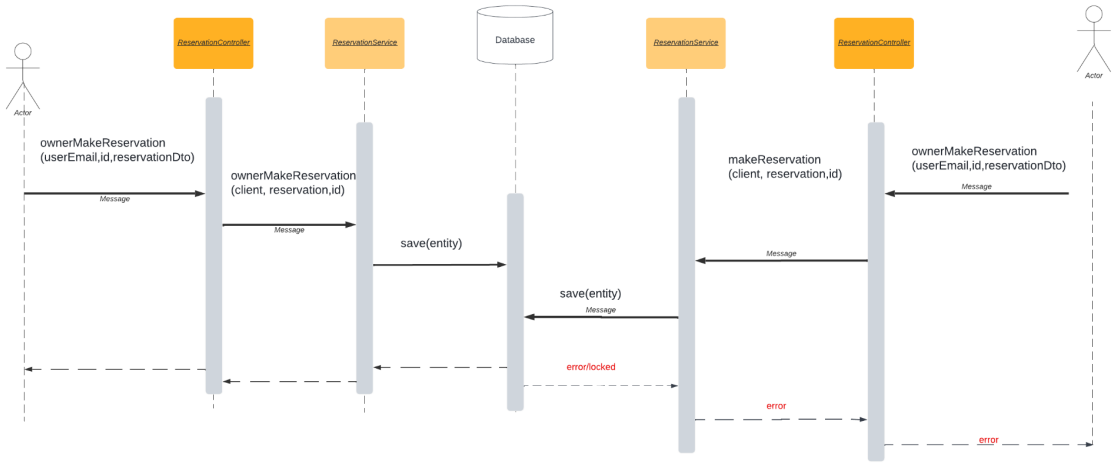
4.4 Konkurentni pristup resursima u bazi

1.Konflikta situacija: Vlasnik vikendice/broda ili instruktor kreira rezervaciju u isto vrijeme kad i drugi klijent

Pored klijenta, vikendicu, brod ili avanturu može da rezerviše i oglašivač za klijenta koji u tom trenutku ima aktivnu rezervaciju.Ovdje može doći do konflikta. Ukoliko i oglašivač i neki korisnik pristupe resursima u isto vrijeme može se desiti da uspješno rezervišu isti entitet za preklapajuće vrijeme.Zbog ovog problema neophodno je obezbjediti da oglašivač i klijent.




```
ownerRentHome(newReservation: IReservation, homeId: number, email: string): Observable<any> {
  return this._http.post('http://localhost:8090/vacation/homes/owner-rent/' + homeId + "/" + email,
    newReservation);
}
```

Slanje requesta

Rešenje konflikte situacije se nalazi se u klasama: Boat/Home Controller, Boat/Home Service, kao i interfejsima Boat/Home Repository.

```java
@PostMapping("/owner-rent/{homeId}/{userEmail}")
public ResponseEntity<?> ownerMakeReservation(@PathVariable String userEmail, @PathVariable Long homeId, @R
    Client client = clientService.getClientByEmail(userEmail);
    if (client.getNoOfPenalties() >= 3) {
        return new ResponseEntity<>(HttpStatus.FORBIDDEN);
    }
    try{
        Reservation reservation = reservationService.ownerMakeReservation(client, reservationDto, homeId);
        return new ResponseEntity<>(ReservationMapper.map(reservation), HttpStatus.OK);
    }
    catch (PessimisticLockingFailureException e){
        return new ResponseEntity<>( body: "Lock:" + e.getMessage(),HttpStatus.CONFLICT);
    }
}
```

Endpoint kontrolera

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

Metoda repozitorijuma

```java
Boat boat = boatRepository.findLockedById(boatId);
if ( boat == null ){
    throw new PessimisticLockingFailureException("Someone is already trying to reserve same boat at th
}
```

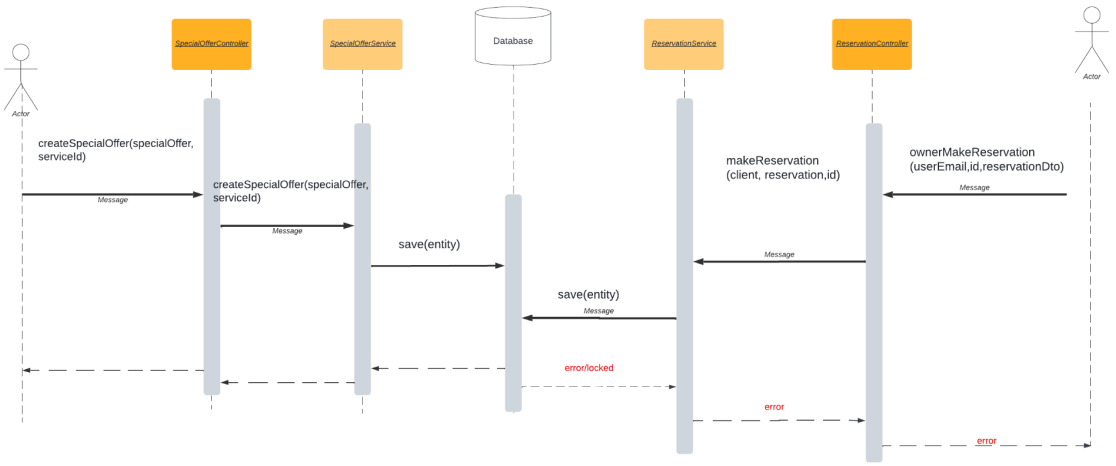
U slučaju da je resurs zaključan bacamo exception

U interfejsima je odrađeno pesimističko zaključavanje findLockerById(Long id) metode na nivou jednog reda u tabeli koji predstavlja entitet. Korišten je PESSIMISTIC_WRITE kao tip zaključavanja.U klasi kontrolleru je dodato rukovanje izuzetkom, te oglašivač ili klijent dobija odgovarajuću poruku.

Pesimističkim zaključavanjem jedne torke smo obezbedili nemogućnost rezervacije entiteta u isto vreme od strane oglašivača i klijenta. Ovakvo zaključavanje onemogućuju rezervaciju čak i slučaju da se termini rezervacija ne poklapaju.

2.Konflikta situacija: Vlasnik vikendice/broda ili instruktor kreira akcij u isto vrijeme kad i drugi klijent rezervaciju

Oglašivačima je omogućeno da kreiraju akcije,predefinisane rezervacije sa popustom koje korisnik kasnije može da rezerviše jednim klikom.Prilikom kreiranja akcije može se desiti da klijent kreira običnu rezervaciju istog entiteta,ovdje može doći do konflikta ako akcija i rezervacija imaju iste termine.




```
createSpecialOffer(specialOffer: SpecialOffer, serviceId: number) {
  return this._http.post(`${this.baseURL}/special-offers/${serviceId}`, specialOffer);
}
```

Slanje requesta

Rešenje konflikte situacije se nalazi se u klasama: SpecialOfferController, SpecialOfferServiceImpl, kao i interfejsima Boat/Home Repository.

```
@PostMapping("/{serviceId}")
@PreAuthorize("hasAnyRole('INSTRUCTOR','HOME_OWNER','BOAT_OWNER')")
public ResponseEntity<?> createSpecialOffer(@RequestBody NewSpecialOfferDto specialOffer,
                                            @PathVariable Long serviceId) throws MessagingE
    SpecialOffer offer = SpecialOfferMapper.toNewEntity(specialOffer);
    try {
        SpecialOffer created = specialOfferService.createSpecialOffer(offer, serviceId);
        return ResponseEntity.ok(SpecialOfferMapper.toDto(created));
    }catch (PessimisticLockingFailureException e){
        return new ResponseEntity<>( body: "Lock:" + e.getMessage(),HttpStatus.CONFLICT);
    }
}
```

Endpoint kontrolera

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

Metoda repozitorijuma

```
Boat boat = boatService.findLockedById(serviceId);
if ( boat == null ){
    throw new PessimisticLockingFailureException("Someone is already trying to reserve same boat at
}
```

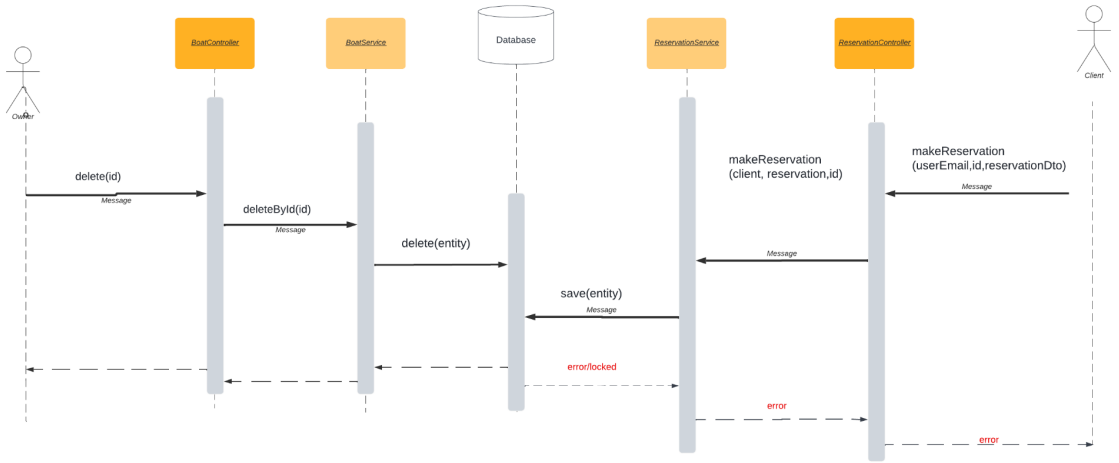
U slučaju da je resurs zaključan bacamo exception

U interfejsima je odrađeno pesimističko zaključavanje findLockerById(Long id) metode na nivou jednog reda u tabeli koji predstavlja entitet. Korišten je PESSIMISTIC_WRITE kao tip zaključavanja.U klasi kontrolleru je dodato rukovanje izuzetkom, te oglašivač ili klijent dobija odgovarajuću poruku.

Pesimističkim zaključavanjem jedne torke smo obezbedili nemogućnost rezervacije entiteta i kreiranja akcije u isto vreme od strane oglašivača i klijenta. Ovakvo zaključavanje onemogućuju kreiranje akcije i rezervacije čak i slučaju da se termini ne poklapaju.

3.Konflikta situacija: Vlasnik vikendice/broda ili instruktor briše avanturu u istom vreme kad korisnik kreira rezervaciju za istu avanturu

Oglašivači mogu da obrišu svoje entitete samo ukoliko za njih ne postoji ni jedna rezervacija u budućnosti. Ako već postoje rezervacije request će biti odbijen, ali ako ne postoje može se desiti da klijent kreira rezervaciju u isto vreme kad korisnik kreira rezervaciju, može doći do toga da klijent ima rezervaciju nad logički obrisanim objektom. Isti konflikt može da se desi i ako klijent krene da zauzme brzu rezervaciju.



```java
@DeleteMapping("/{id}")
@PreAuthorize("hasAnyRole('ADMIN', 'BOAT_OWNER')")
public ResponseEntity<?> deleteABoat(@PathVariable Long id) {
    try{
        boatService.deleteById(id);
        return new ResponseEntity<>( body: "Deleted", HttpStatus.OK);
    }
    catch (PessimisticLockingFailureException e){
        return new ResponseEntity<>( body: "Lock:" + e.getMessage(),HttpStatus.CONFLICT);
    }
}
```

Enpoint kontrolera

```java
@Override
@Transactional
public void deleteById(Long id) {
    Boat found = boatRepository.findLockedById(id);
    if ( found == null ){
        throw new PessimisticLockingFailureException("Someone is already trying to reserve same boat at t
    }
    var noOfFutureRes :int = reservationService.getNoOfIncomingReservationsForBoat(id);
    if (!(noOfFutureRes > 0)) {
        found.setDeleted(true);
        boatRepository.save(found);
    }

}
```

Servisna metoda(baca exception)

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

Metoda repozitorijuma

Rešenje konfliktne situacije: analogno sa prethodnim slučajevima.Ovog puta nema loše strane jer je bitno da se resurs zaključa.

Jelena Stajić RA109/2018