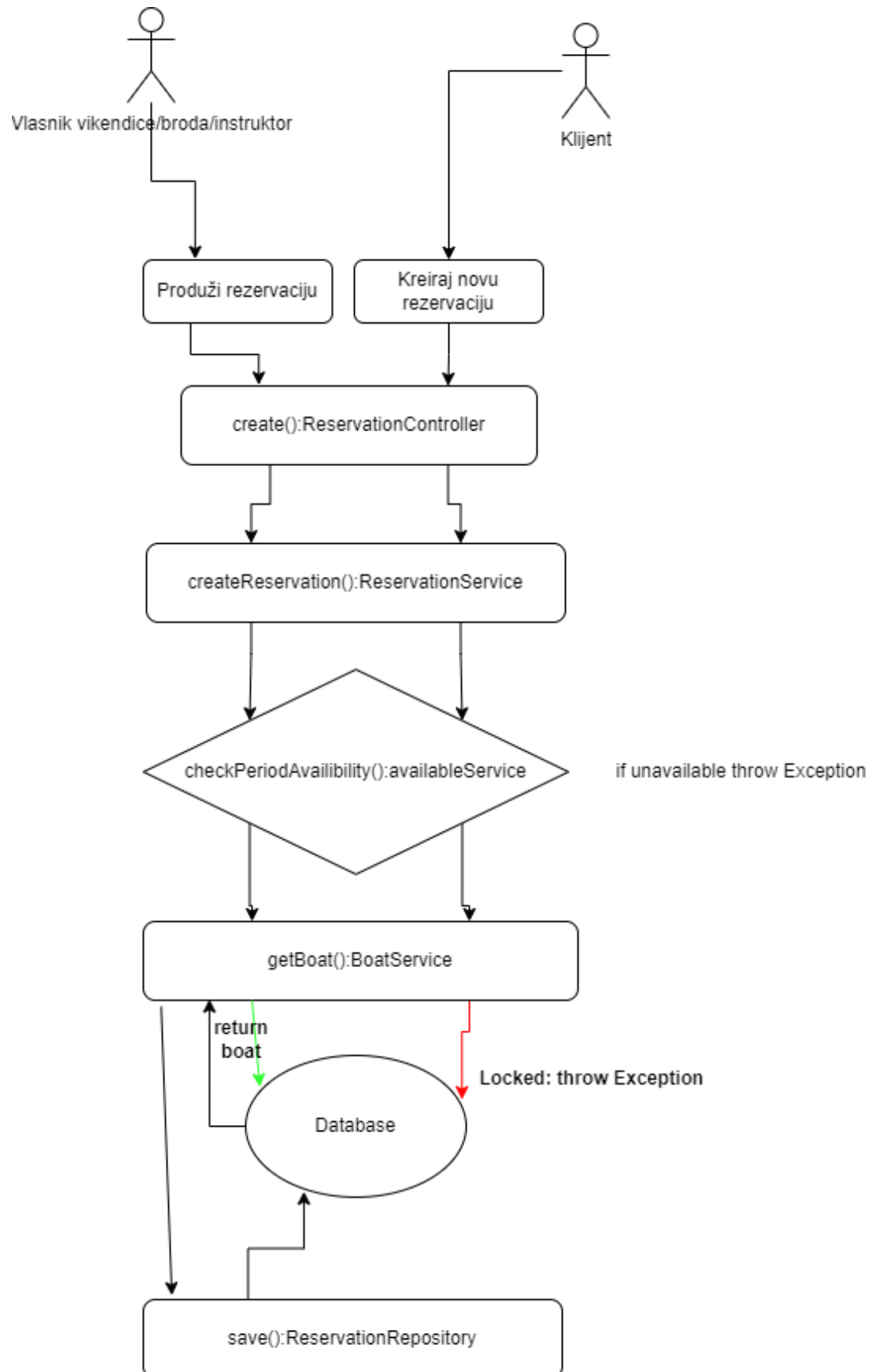
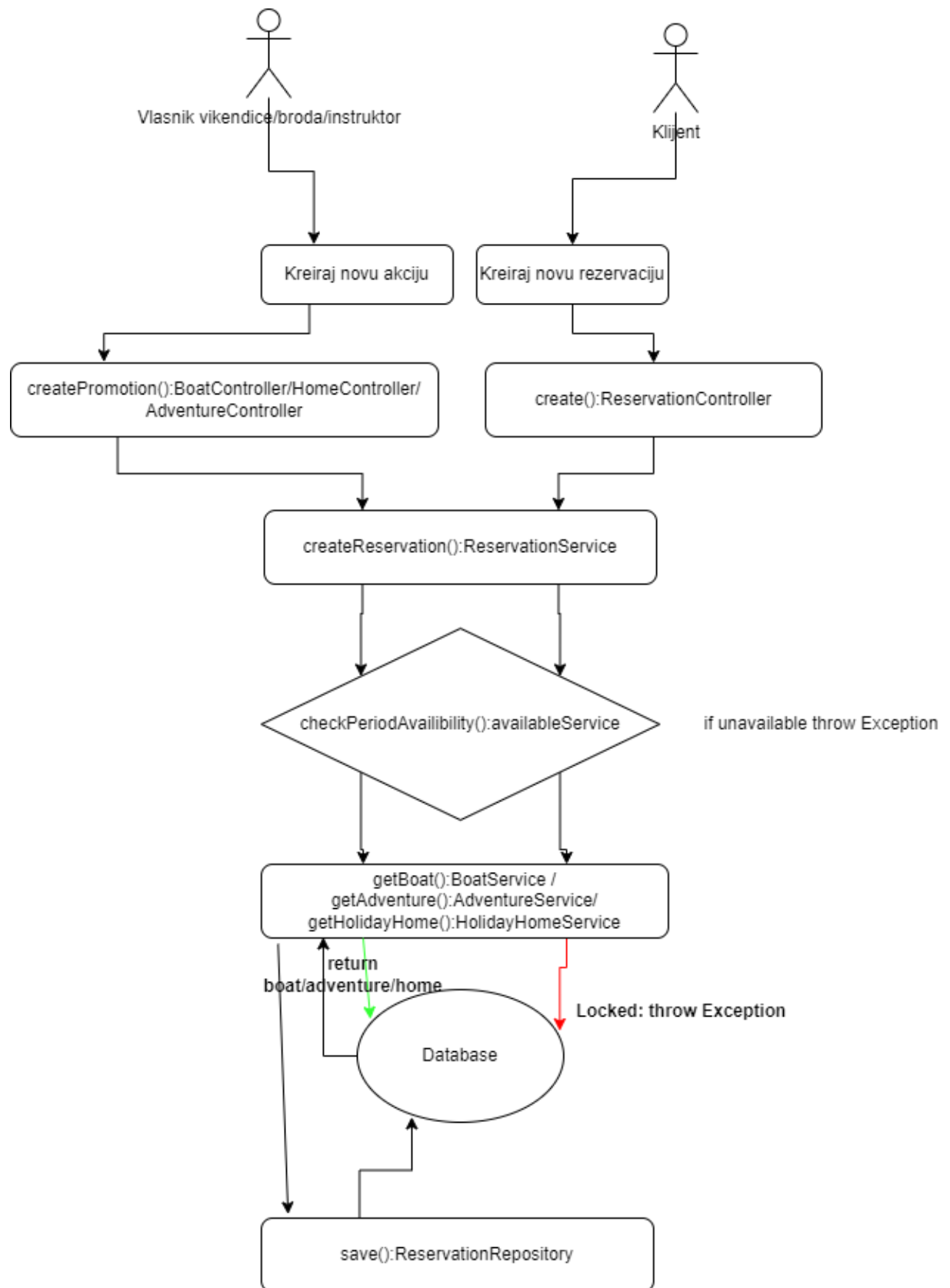


- Vlasnik vikendice/broda ili instruktor ne može da napravi rezervaciju u isto vreme kad i drugi klijent
 - Vlasnik vikendice/broda ili instruktor mogu praviti rezervacije za klijente čija rezervacija je u toku, ali kako se je moguće, potrebno je rešiti konflikt koji nastaje kada vlasnik vikendice/broda ili instruktor pokuša da napravi rezervaciju u isto vreme kad i klijent.
 - Problem se rešava tako što se pesimistički zaključava resurs (avantura, brod ili vikendica) pri dobavljanju jer ako ne može da se dobavi ne može ni da se zauzme. Ovo je implementirano preko anotacije `@Lock(LockModeType.PESSIMISTIC_READ)`. Takođe je potrebno preko anotacije `@Transactional` na nivou servisa obezbediti transakcioni tok podataka da bi se održala konzistencija. U pitanju je endpoint `@PostMapping("/create/{promotion_id}")` public `ResponseEntity<HttpStatus>`
`createWithPromotion(@Valid @RequestBody CreateReservationDTO`

createReservationDTO,@PathVariable Long promotion_id).



- Vlasnik vikendice/broda ili instruktor ne može da napravi akciju u isto vreme kad i drugi klijent rezervaciju
 - Vlasnik vikendice/broda ili instruktor mogu praviti akcije, ali kako se akcijama zauzima isti vremenski prostor kao i rezervacijama potrebno je rešiti konflikt koji nastaje kada i vlasnik vikendice/broda ili instruktor pokuša da napravi akciju u isto vreme kad i klijent rezervaciju.
 - Kao i u prethodnom primeru, problem se rešava tako što se pesimistički zaključava resurs (avantura, brod ili vikendica) pri dobavljanju jer ako ne može da se dobavi ne može ni da se zauzme. Ovo je implementirano preko anotacije `@Lock(LockModeType.PESSIMISTIC_READ)`. Takođe je potrebno preko anotacije `@Transactional` na nivou servisa obezbediti transakcioni tok podataka da bi se održala konzistencija. U pitanju je endpoint `@PostMapping("/create")` `public ResponseEntity<HttpStatus> create(@Valid @RequestBody CreateReservationDTO createReservationDTO)`.



- Vlasnik vikendice/broda ili instruktor ne može da obriše slobodan period za vikendicu/brod/avanturu u isto vreme kad i drugi klijent pravi rezervaciju
 - Vlasnik vikendice/broda ili instruktor mogu brisati slobodne periode, ali kako se rezervacijama oni zauzimaju potrebno je rešiti konflikt koji nastaje kada vlasnik vikendice/broda ili instruktor pokuša da obriše period u isto vreme kad i klijent napravi rezervaciju.
 - Kao i u prethodnom primeru, problem se rešava tako što se pesimistički zaključava resurs (slobodni periodi) pri dobavljanju jer ako ne može da se dobavi ne može ni da se zauzme. Ovo je implementirano preko anotacije `@Lock(LockModeType.PESSIMISTIC_READ)`. Takođe je potrebno preko anotacije `@Transactional` na nivou servisa obezbediti transakcioni tok podataka da bi se održala konzistencija. U pitanju je endpoint


```
@DeleteMapping(value = "{id}") public ResponseEntity<HttpStatus>
deletePeriod(@PathVariable Long id)@PostMapping("/create") public
ResponseEntity<HttpStatus> create(@Valid @RequestBody CreateReservationDTO
createReservationDTO).
```

