Nefunkcionalan zahtev 4.4 – Konkurentan pristup bazi podataka

[Nikola Kajtes RA116/2018]

1) Konfliktna situacija: Više istovremenih klijenata ne mogu da naprave rezervaciju istog entiteta u isto (ili preklapajuće) vreme.

Problem se javlja kad više korisnika u “isto” vreme pristupi endpoint-u: "api/boatowners/getAllFreeTermsForBoat/{id}" (analogno nazivu ovog endpoint-a za brodove može da pristupi i za entitete WeekendHouse i FishingLesson).

U tom slučaju svim korisnicima će se izlistati oni entiteti koji su slobodni za selektovan datumski opseg koji su izabrali shodno situaciji baze u trenutku izlistavanja slobodnih termina. Ako više korisnika pokuša da rezerviše isti entitet sa preklapajućim terminom nekog drugog klijenta, softver treba da omogući rezervaciju samo onom klijentu koji je prvi kliknuo na “Reserve”, a ostalima da prikaze neku odgovarajuću poruku.

Pre rešavanja ovog problema, svi klijenti koji bi stisnuli “Reserve” bi bili upisani u bazu, i na taj način bi bilo moguće da 100 ljudi rezerviše isti entitet u isto vreme, što naravno nije realna situacija.

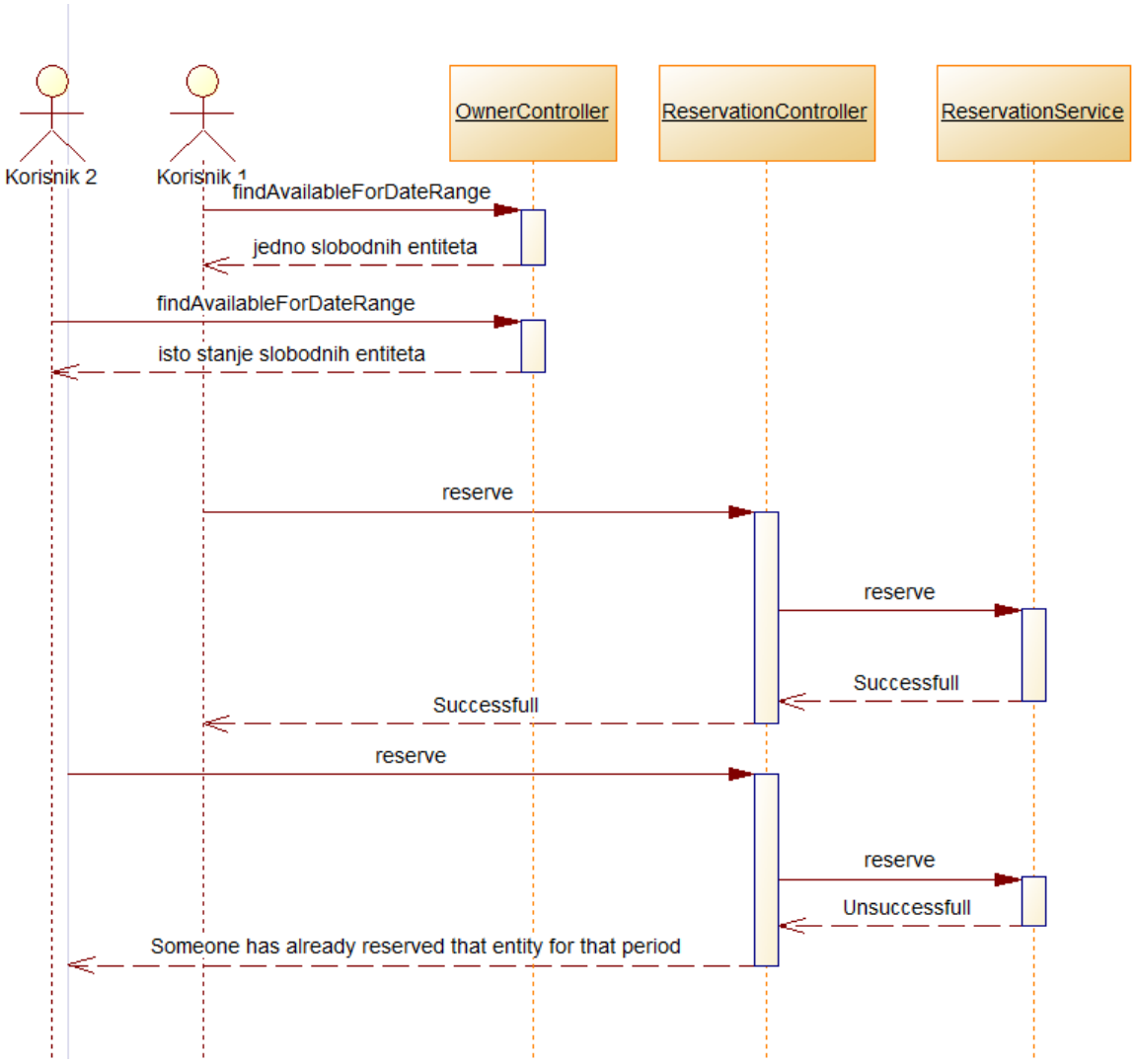
Ovaj problem se sprečava time što se doda još jedna provera u service-u koja će da uporedi da li se nov datumski opseg rezervacije nalazi već u bazi u vidu preklapanja sa nekim drugim periodom zauzetosti tog određenog entiteta (na taj način će biti omogućena uspešna rezervacija samo onog user-a koji je prvi kliknuo na „Reserve“).

Metoda servisa findOneById se pesimistički zaključava u write režimu do kraja transakcije, tako da prvi koji dođe do nje, može da je koristi, a svi ostali su prinuđeni da sačekaciju da se transakcija završi (u slučaju sa postgress bazom to čekanje nije moguće pa će im se odmah izbaciti obaveštenje da pokušaju ponovo).

2) Konfliktna situacija: Više istovremenih klijenata ne mogu da naprave rezervaciju istog entiteta na akciji u isto vreme.

U našem projektu, brza i obična rezervacija se čuvaju u istoj tabeli, pa je taj problem rešen ujedno i sa konfliktnom situacijom 1.

Reprezentacija obične rezervacije u bazi je rezervacija koja ima popunjenu kolonu “customer_id”, a kolona “end_special_offer” je postavljena na null, dok se akcijom (brzom rezervacijom), koja jos uvek nije rezervisana, smatra ona rezervacija koja nema nikakvog customer-a, a ima vrednost u koloni “end_special_offer”. Kada se akcija rezerviše, kolona “customer_id” se samo popuni sa odgovarajućim klijentom koji ju je rezervisao.


Korisnik 2
Korisnik 1
OwnerController
ReservationController
ReservationService
findAvailableForDateRange
jedno slobodnih entiteta
findAvailableForDateRange
isto stanje slobodnih entiteta
reserve
reserve
Successfull
Successfull
reserve
reserve
Unsuccessfull
Someone has already reserved that entity for that period

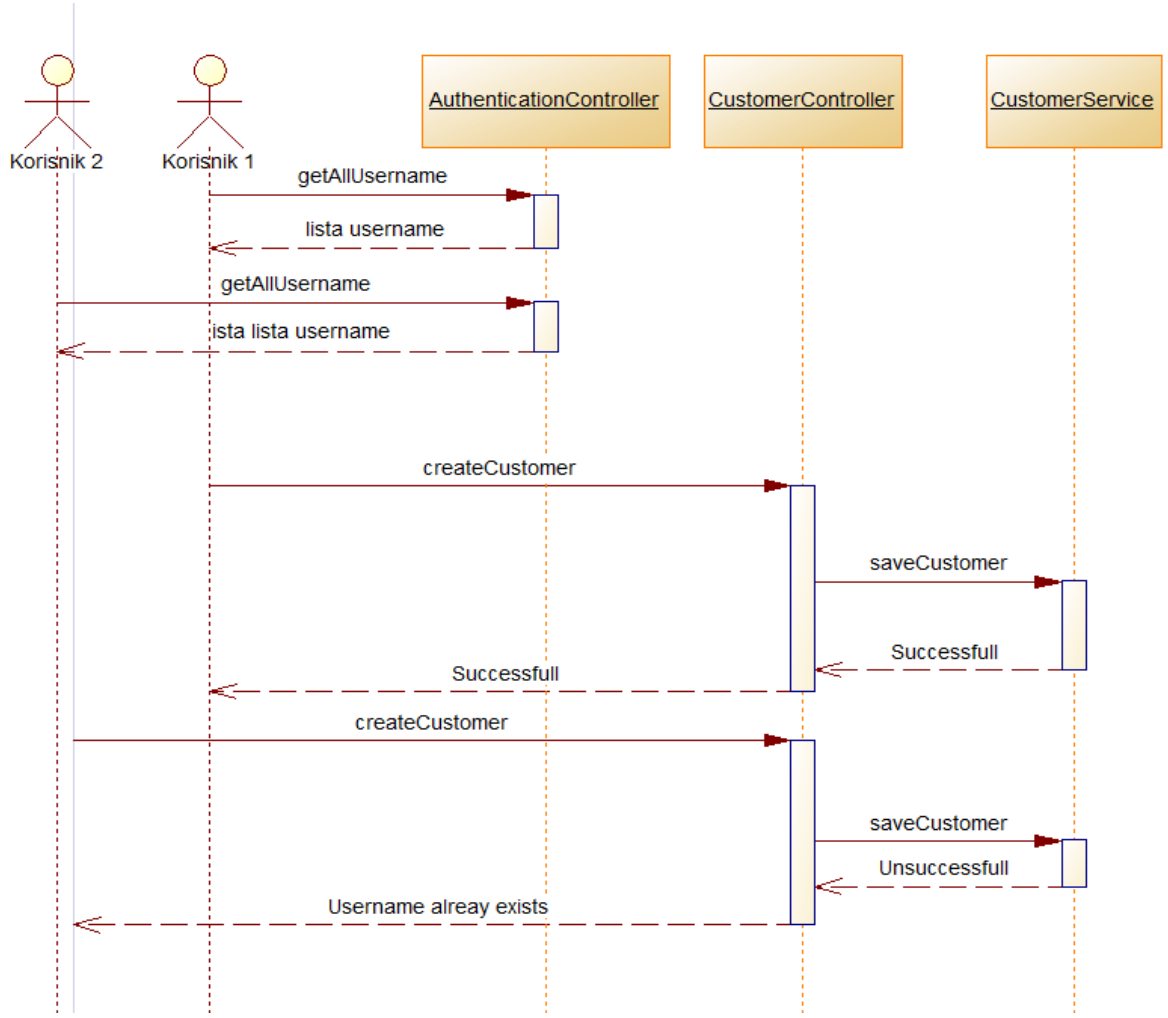
3) Konfliktna situacija: Kreiranje naloga za više klijenata koji imaju isti username u isto vreme.

Problem se javlja kad više korisnika u “isto” vreme pristupi endpoint-u: "/api/customers/create".

Prilikom registracije novog korisnika njemu će biti omogućena informacija da li korisničko ime, koje je uneo već postoji u bazi, i ukoliko ne postoji, biće mu omogućeno da pristupi odgovarajućem endpoint-u. Ukoliko više korisnika istovremeno otvori formu za unos novog korisnika, svima će pisati ista zauzeta korisnička imena i biće im omogućeno da se svi oni prijave sa istim koriscničkim imenom koje se ne nalazi u bazi do trenutka njihovog kliktanja na formu za registraciju.

To se sprečava time što se doda još jedna provera u service-u koja će da uporedi da li se novo korisničko ime nalazi već u bazi (na taj način će biti omogućena registracija samo onog user-a koji se prvi prijavio sa tim korisničkim imenom).

Metoda servisa findOneByUsername se pesimistički zaključava u write režimu do kraja transakcije, tako da prvi koji dođe do nje, može da je koristi, a svi ostali su prinuđeni da sačekaciju da se transakcija završi (u slučaju sa postgress bazom to čekanje nije moguće pa će im se odmah izbaciti obaveštenje da pokušaju ponovo).



4) Konfliktna situacija: Menjanje podataka svog korisničkog profila u isto vreme sa različitih uređaja.

Problem se javlja kad korisnik pokuša da u “isto” vreme menja podatke na svom profilu sa dva odvojena uređaja (npr. sa dva različita računara ili računara i telefona) i time pristupa endpoint-u: "/api/customers/ edit".

Prilikom ove situacije dešavaće se sledeće: korisnik pristupi svom profilu u “isto” vreme sa dva odvojena uređaja, pa zatim izvrši neku promenu na jednom uređaju i klikne edit (promena se sačuva u bazi), a zatim na drugom uređaju izvri opet neku promenu. U tom slučaju on će drugom izmenom “pregaziti” prvu.

Problem sam rešio optimističnim pristupom, tako što sam ubacio u model customer-a dodatno polje “version” (anotirano sa @Version) i napravio customer service transakcionim, tj. učinio njegove metode transakcionim. Tada će se desiti sledeće: oba uređaja dobiju model koji u sebi sadrži istu verziju. Kada prvi napravi izmenu u bazi, tada se njegova verzija (autoinkrement) poveća za 1. Zatim, kada drugim klijent sa drugog uređaja pristupi istoj metodi proveriće mu se verzija iz baze i ona koju on ima od pre i ukoliko se ne poklapaju, a ne poklapaju se, biće mu onemogućen pristup izmeni entiteta (baca ObjectOptimisticLockingFailureException).