

Transakcije

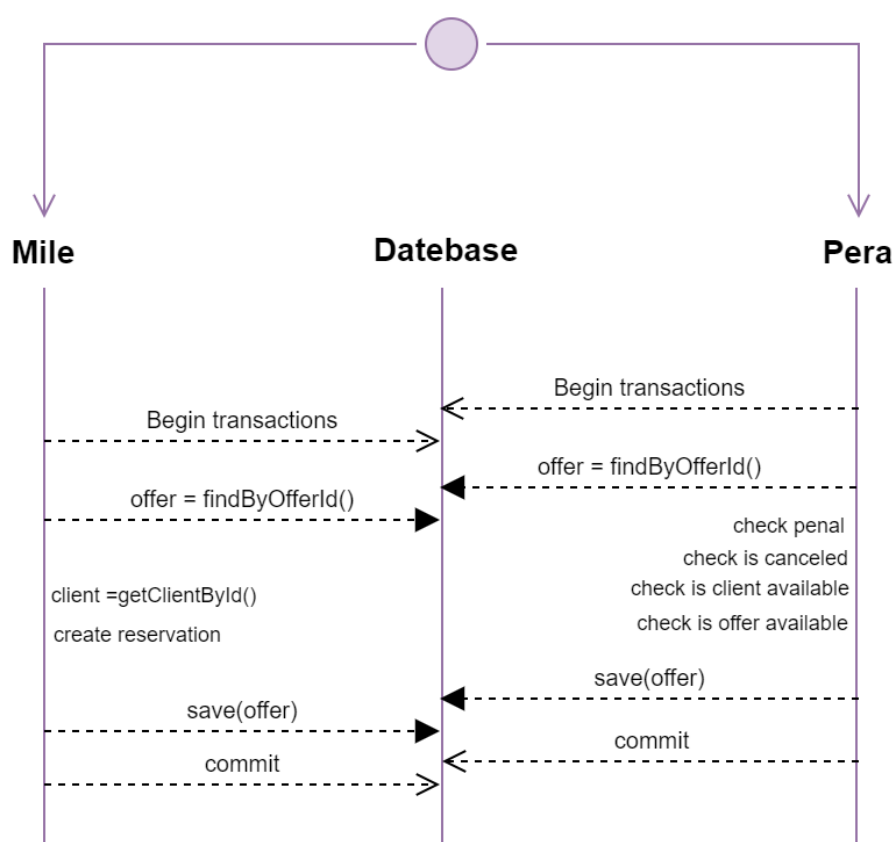
Student 2

Svi endpoint-i počinju sa: <http://localhost:8082>

1. Vlasnik vikendice/broda ili instruktor ne može da napravi rezervaciju u isto vreme kad i drugi klijent

Putanja - vlasnik: [/reservation/make-by-owner \(POST\)](#)

Putanja - klijent: [/reservation/make \(POST\)](#)

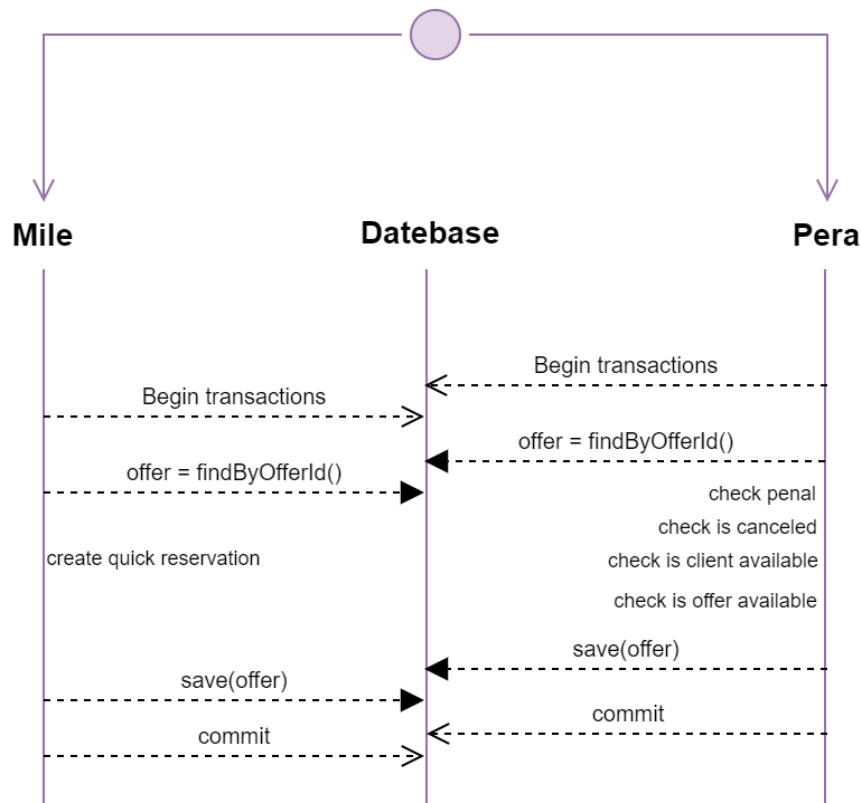


- Do konflikta dolazi kada vlasnik vikendice/broda ili instruktor pokuša da kreira novu rezervaciju umesto klijenta dok u istom ili u trenutku sa malim razmakom drugi klijent pokušava da rezerviše isti entitet. Ovo predstavlja problem jer postoji mogućnost da se rezervisani termini poklope i samim tim će se kreirati dve rezervacije za isti ili preklapajući termin što nije dozvoljeno. S ovim problemom dolazi do kreiranja dva objekta i njihovog čuvanja u bazi što uzrokuje nekonzistentnost naše baze.
- Za rešavanje ovog konflikta je korišćeno optimističko zaključavanje. Ponude koje postoje imaju listu rezervacija gde će se dodati nova rezervacija nakon svih potrebnih proverama. Entitet offer ima atribut **numberOfReservations** koji će se porediti pri njegovom učitavanju sa onim pri kraju transakcije, na ovaj način je implementirano optimističko zaključavanje.

2. Vlasnik vikendice/broda ili instruktor ne može da napravi akciju u isto vreme kad i drugi klijent vrši rezervaciju postojećeg entiteta

Putanja - vlasnik: [/quick-reservation/add \(POST\)](#)

Putanja - klijent: [/reservation/make \(POST\)](#)

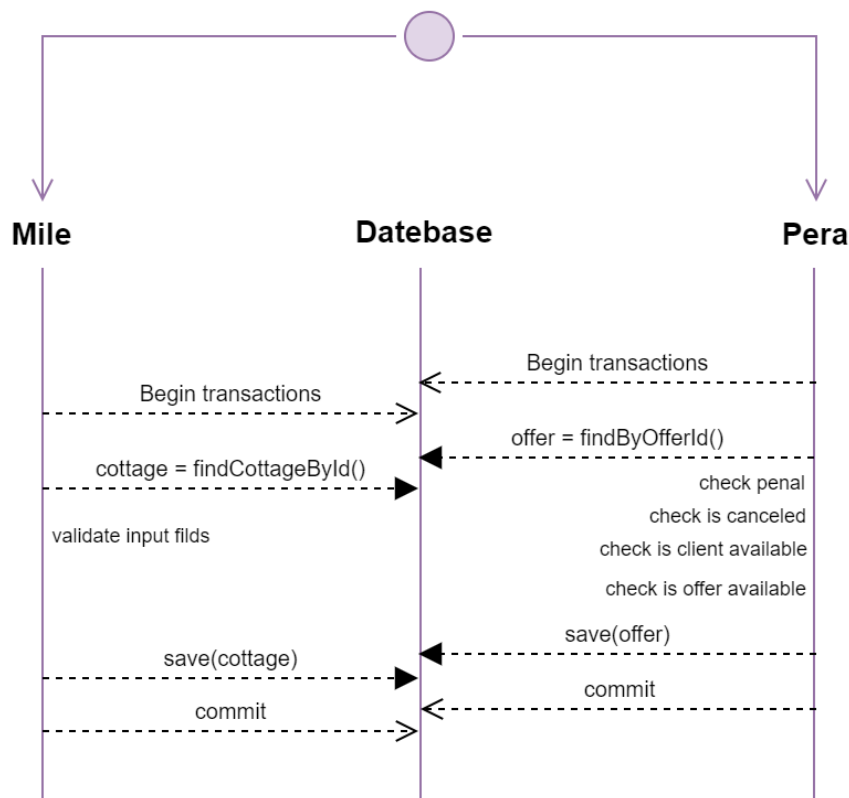


- Do konflikta dolazi kada vlasnik vikendice/broda ili instruktor pokuša da kreira novu akciju dok u istom ili u trenutku sa malim razmakom drugi klijent pokušava da rezerviše isti entitet. Ovo predstavlja problem jer postoji mogućnost da vlasnik kreira akciju za period koji je klijent izabrao za rezervaciju, samim tim cena i ostali atributi rezervacije neće biti konzistentni i postoji mogućnost da neki drugi klijent rezerviše entitet u istom periodu uz pomoć akcije. Ukoliko bi se iskoristila akcija došlo bi do dve rezervacije sa istim ili preklapajućim vremenskim intervalom, kao i do kreiranja dva objekta i njihovog čuvanja u bazi što uzrokuje nekonzistentnost naše baze.
- Za rešavanje ovog konflikta je korišćeno optimističko zaključavanje. Ponude koje postoje imaju listu akcija gde će se dodati nova akcija nakon svih potrebnih provera. Entitet offer ima atribut **numberOfQuickReservations** koji će se porediti pri njegovom učitavanju sa onim pri kraju transakcije, na ovaj način je implementirano optimističko zaključavanje.

3. Vlasnik vikendice/broda ne može da menja podatke svoje vikendice/broda kada klijent vrši rezervaciju tog entiteta (ukoliko vikendica/brod nema budućih rezervacija i izmena podataka je dozvoljena)

Putanja - vlasnik: **/cottage/update (POST)**

Putanja - klijent: **/reservation/make (POST)**



- Do konflikta dolazi kada vlasnik vikendice/broda ili instruktor pokuša da izmeni podatke postojećeg entiteta (entitet nema budućih rezervacija i izmena je moguća) dok u istom ili u trenutku sa malim razmakom drugi klijent pokušava da rezerviše isti entitet. Ovo predstavlja problem jer postoji mogućnost da vlasnik izmeni podatke i samim tim cena i ostali atributi rezervacije neće biti konzistentni sa napravljenom rezervacijom od strane klijenta. Takođe ne bi trebalo menjati podatke ukoliko je entitet rezervisan par trenutaka pre čuvanja izmena. Podaci vezani za entitet u našoj bazi postaju nekonzistentni.
- Za rešavanje ovog konflikta je korišćeno optimističko zaključavanje. Entitet offer ima atribut **numberOfModify** koji će se porediti pri njegovom učitavanju sa onim pri kraju transakcije, na ovaj način je implementirano optimističko zaključavanje.