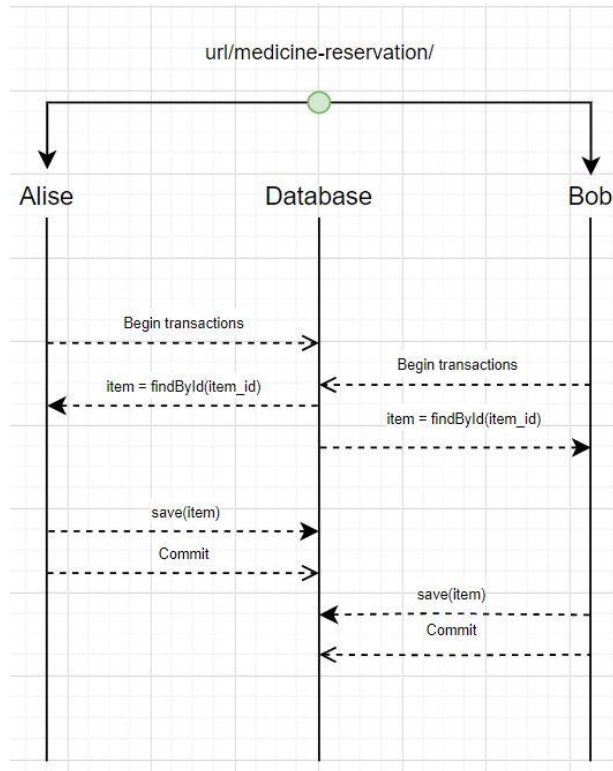


Transakcije – Jovan Simić, sw26-2018

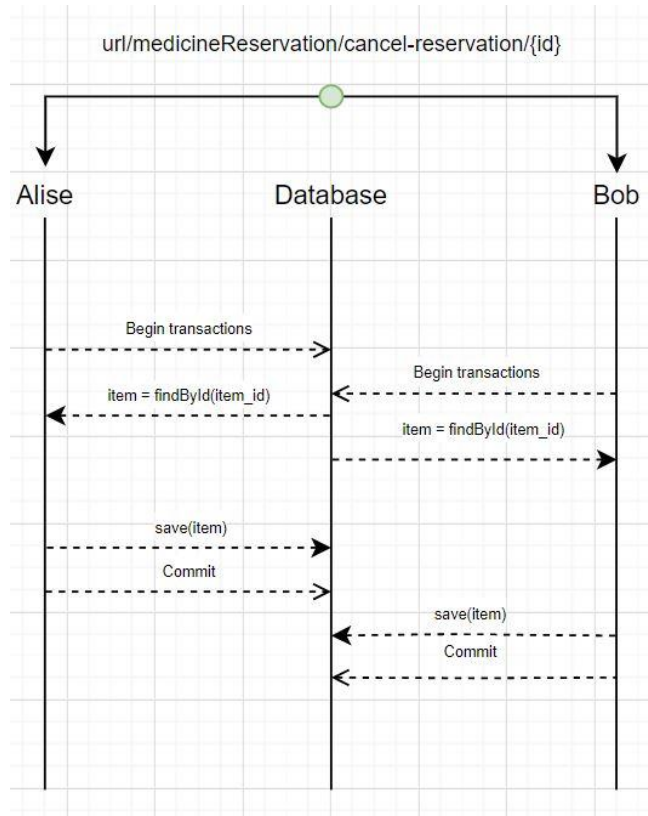
- Početni url za sve endpointe je: <http://localhost:3000/api/>

Konfliktna situacija 1: Rezervacija leka koji je u međuvremenu postao nedostupan



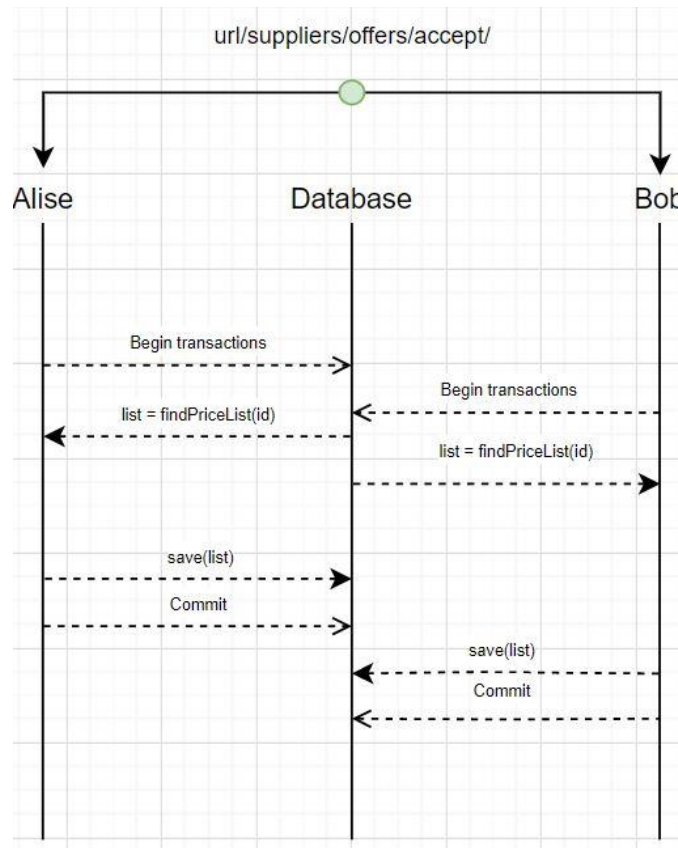
- Na stanju je ostao jedan lek, koji vide Alice i Bob. Oboje pokušavaju da ga zakažu u približno isto vreme (u pitanju su milisekunde). Šalje se request ka serveru i pogađa endpoint prikazan na slici. Otvaraju se dve paralelne transakcije i započinju svoje izvršavanje.
- Ukoliko nemamo zaključavanje zajedničkog resursa (u ovom slučaju leka, gde se umanjuje količina rezervisanog leka) napraviće se 2 rezervacije istog leka, iako je na stanju bio samo jedan lek.
- Za rešavanje ovog konflikta sam koristio pesimistički pristup uvođenjem odgovarajućih anotacija u repozitorijumu rezervacije leka. Korišćenjem exclusive lock-a, onemogućeno je drugima da pristupaju resursu, i time se neće naći u nekonzistentnom stanju. Za ovaj pristup sam se odlučio, jer je ova funkcionalnost primarna i korisnici će je često koristiti. Pošto očekujem da će ovde biti često konflikata, optimističko “zaključavanje” nije možda prikladno, jer se može javiti takozvani **time-of-check to time-of-use** (TOCTOU). To je softverski bag koji se da javiti kod ovakvog pristupa. Prilikom poređenja stare verzije i verzije dobavljene iz baze, iako verzije možda jesu iste, u samom trenutku poređenja te dve verzije je neka druga transakcija mogla izvršiti UPDATE. Nakon toga i ova transakcija izvrši isto i time pregazi pređašnju transakciju. Iz tog razloga sam ovde koristio pesimistički pristup, kako ne bih doveo bazu u nekonzistentno stanje.

Konfliktna situacija 2 - Otkazivanje rezervacije leka



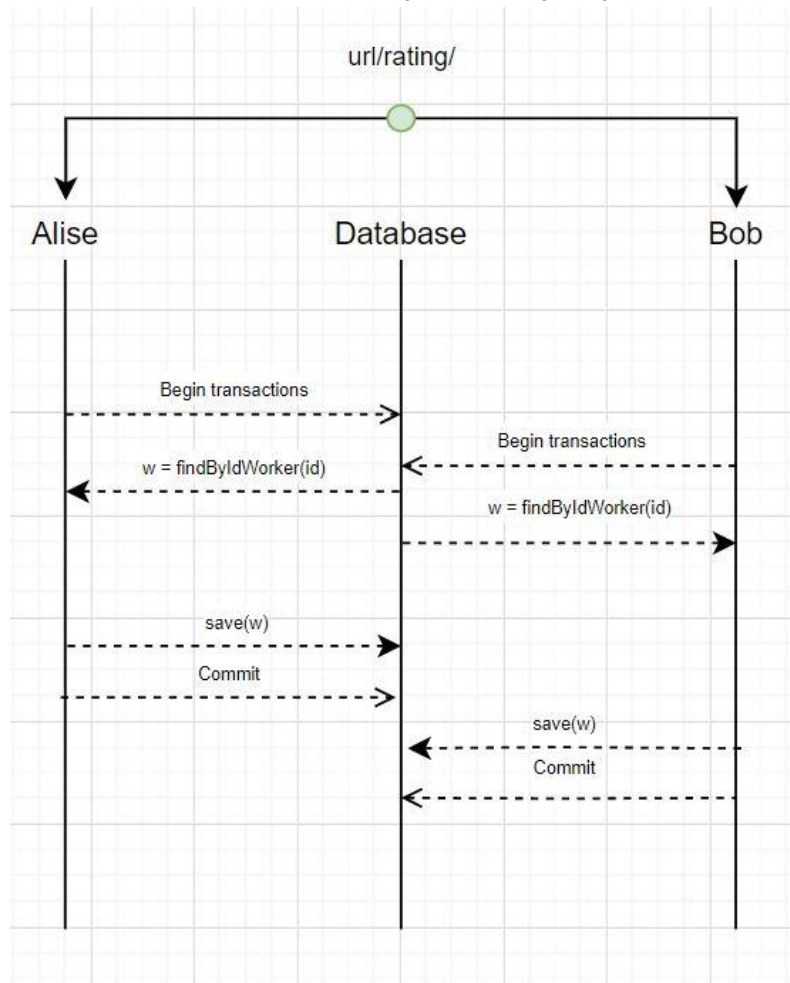
- Alice i Bob imaju svoje rezervacije istog leka iz iste apoteke. Oboje pokušavaju da ih otkažu u približno isto vreme (u pitanju su milisekunde). Šalje se request ka serveru i pogađa endpoint prikazan na slici. Otvaraju se dve paralelne transakcije i započinju svoje izvršavanje.
- Do konflikta može doći tako što Alice pri otkazivanju svoje rezervacije ažurira stanje leka, tako što ga uveća i sačuva svoje izmene. Malo potom Bob uradi istu stvar, ali kako on nema ažurirano stanje apoteke, koje je Alice načinila, čuvanjem svojih izmena će narušiti Alicine izmene.
- Za rešavanje ovog konflikta sam koristio optimistički pristup dodavanjem novog polja version u klasi `MedicineItem`. Kako je ovo akcija koja će se najverovatnije retko dešavati (nije primarna akcija koju korisnici koriste svakodnevno), ne očekujem da će se konflikti često dešavati i zbog toga smatram da nema potrebe za striktnim zaključavanjem korišćenjem pesimističkog pristupa, već će i ovaj pristup poslužiti ukoliko do konflikta dođe.

Konflikta situacija 3 – Prihvatanje ponude za narudžbenicu



- Alice i Bob koji su administratori iste apoteke pokušavaju da prihvate narudžbenice u približno isto vreme (u pitanju su milisekunde). Šalje se request ka serveru i pogađa endpoint prikazan na slici. Otvaraju se dve paralelne transakcije i započinju svoje izvršavanje.
- Ovde dolazi do konflikta kada se ažurira stanje apoteke tj. njene količine lekova. Ukoliko Alice prihvati svoju porudžbinu i ažurira stanje količine lekova u apoteci, a malo potom i Bob uradi isto, pregazio bi izmene koje je Alice načinila, zato što on kod sebe ima stanje apoteke kakvo je bilo pre Alicinih izmena.
- Za rešavanje ovog konflikta sam koristio pesimistički pristup uvođenjem odgovarajućih anotacija u repozitorijumu PriceList-a. Kako je ovo akcija koja će se najverovatnije retko dešavati, a i administratora apoteke nikad neće biti količinski mnogo, ukoliko nekim slučajem i dodje do konflikta, biće obezbeđena konzistentnost uz potencijalno čekanje. Pošto sam dovaljao sve lekove u okviru apoteke i odgovarajuće menjao, nisam video način na koji bih iskoristio optimistično zaključavanje i time sam se odlučio za ovaj pristup.

Konfliktna situacija 4 - Ocenjivanje



- Alice i Bob pokušavaju da ocene istog dermatologa u približno isto vreme (u pitanju su milisekunde). Šalje se request ka serveru i pogađa endpoint prikazan na slici. Otvaraju se dve paralelne transakcije i započinju svoje izvršavanje.
- Neka imamo 7 ocena posmatranog dermatologa, na osnovu kojih je formirana srednja ocena. Do sukoba može doći tako što će Alice na osnovu svoje nove ocene koja se dodaje i preostalih 7 ocena formirati novu srednju ocenu. Sačuvaće se novo izračunata vrednost kod dermatologa. Malo zatim će Bob isto formirati novu srednju ocenu na osnovu svoje ocene i preostalih 7 ocena (neće uzeti u obzir ažurirano stanje svih ocena, kojih je sad 8). Samim tim će se pregaziti srednja ocena formirana od strane Alice i neće biti uračunata.
- Ovaj konflikt sam rešio optimističkim pristupom, tako što sam uveo jedno novo polje version u klase PharmacyWorker, Pharmacy i Medicine. Time sam sprečio da svako ko pokuša da promeni stanje srednje ocene posmatranog entiteta, naruši nečije prethodno ocenjivanje, tako što će se gledati version polje entita koji se ocenjuje. Ako se poklapa sa verzijom koju ima kod sebe, izmena će biti omogućena, inače će biti zahtevano od korisnika da ponovo pokuša. Izmjena ocene je rađena po istom principu. Za ovaj pristup sam se odlučio jer smatram da neće biti konflikata često, jer ovo nije primarna funkcionalnost.