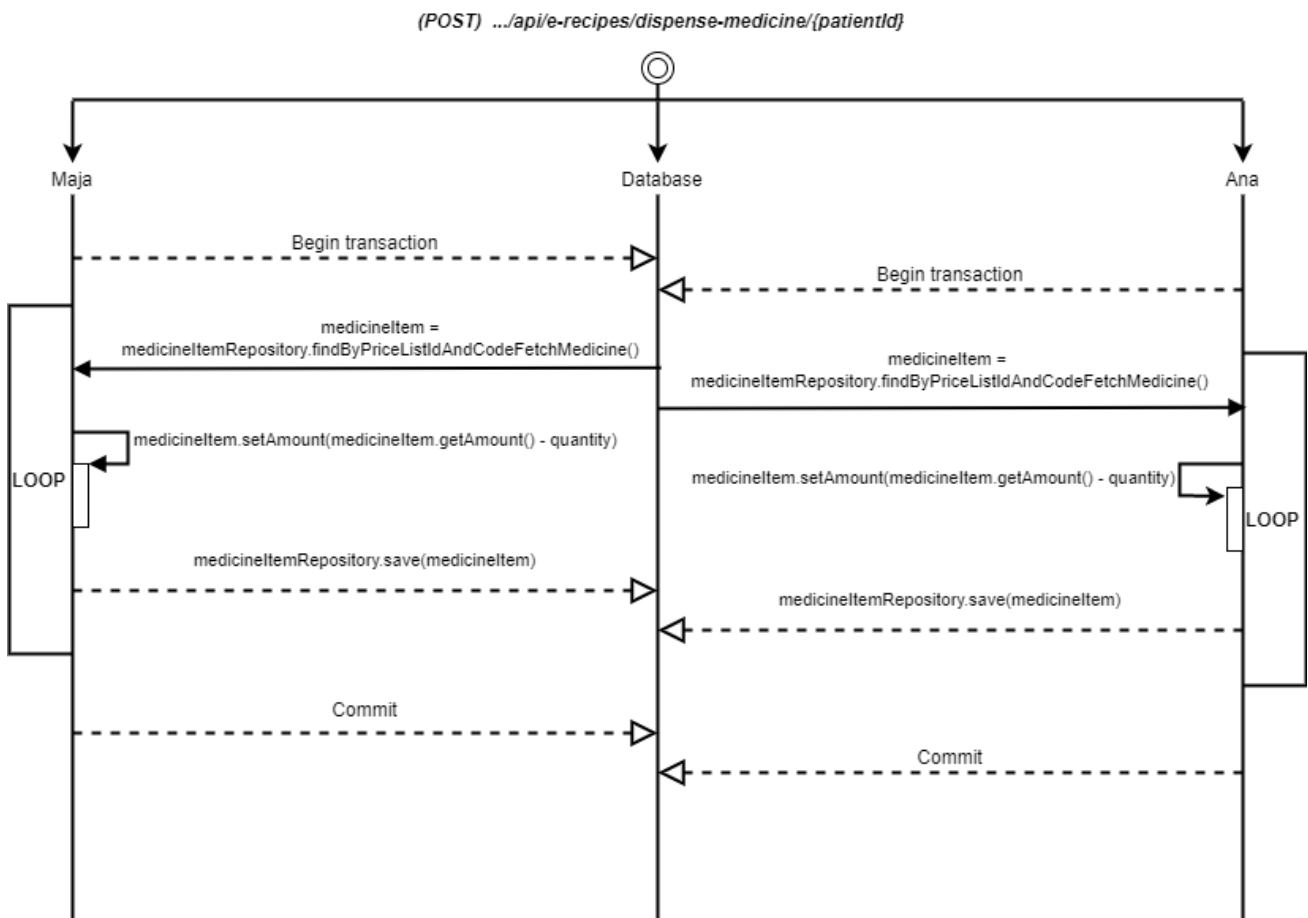


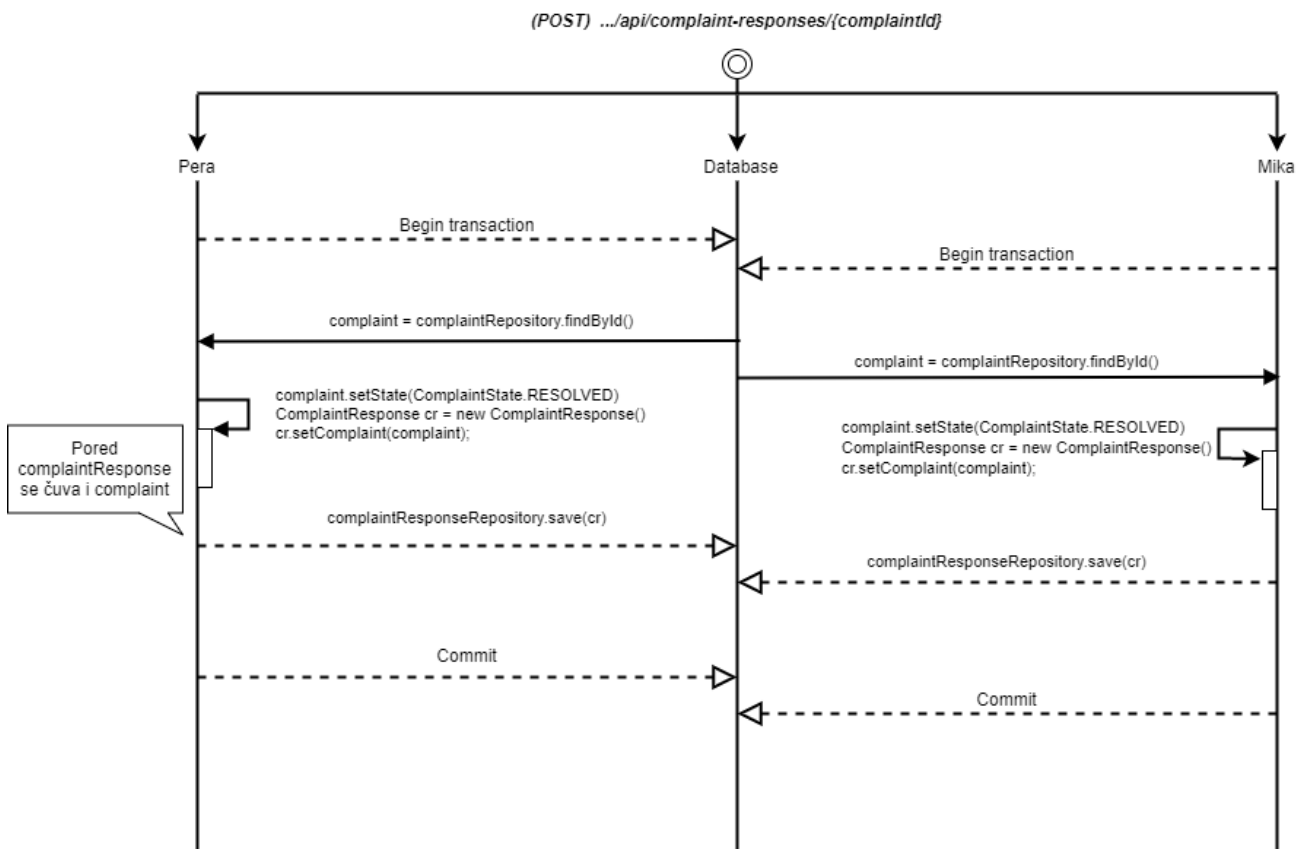
Konfliktna situacija 1: Izdavanje eRecepta tako da se izdaju ili svi ili nijedan lek

- Može se desiti da se stanje lekova u apoteci u toku provere dostupnosti promeni i time baza dovede u nekonzistentno stanje. Na primer, ukoliko više korisnika pokušava da podigne lekove korišćenjem eRecepta, a u apoteci postoje lekovi samo za jedan od njih, mora sa obezbediti da bude izdat jedan eRecept, a ne više njih.
- Kako je pristupanje stavkama apoteke (i njihovo menjanje) relativno česta stvar, ukoliko bismo koristili optimističko zaključavanje, dolazilo bi do čestih promena verzija i grešaka (samim tim i *rollback*-ova, što može biti skupo), pa je za ovo odabrano pesimističko zaključavanje. Korišćeno je `@Lock(LockModeType.PESSIMISTIC_WRITE)` zaključavanje nad metodom repozitorijuma koja dobavlja stavku iz kataloga za datu apoteku. Ovim se onemogućava pristup tom podatku za vreme trajanja transakcije. Ukoliko se desi da u međuvremenu nestane neki od lekova koji još uvek nije zaključan (a treba da bude u nekoj od narednih iteracija), baciće se *CustomException* sa odgovarajućom porukom i transakcija će biti *rollback*-ovana. Primer konfliktne situacije dat je u nastavku.



Konfliktna situacija 2: Odgovaranje na žalbu

- Može se desiti da više admina u isto vreme pokušaju da odgovore na istu žalbu. Ukoliko nemamo zaključavanje resursa (žalbe), desiće se da je na jednu žalbu odgovoreno nekoliko puta (onoliko koliko je admina poslalo istovremeni zahtev). Ovo dovodi bazu u nekonzistentno stanje.
- Primer konflikta:* Pera je poslao zahtev koji sadrži podatke odgovora (id žalbe, tekst odgovora...). Istovremeno, Mika šalje isti zahtev, samo sa drugačijim odgovorom. Pera pročitao da žalba postoji i da na nju nije odgovoreno. Isti odgovor dobije i Mika. Pera odgovora na žalbu, dobija poruku da je uspešno odgovorio, Mika odgovara na žalbu i dobija poruku da je uspešno odgovorio, baza je u nekonzistentnom stanju.
- Primer rešenja:* ... Pera pročitao da žalba postoji i da na nju nije odgovoreno. Verzija žalbe je X. Mika dobije isti odgovor, takođe sa verzijom X. Pera pokušava da sačuva odgovor, verzija žalbe koju je pročitao je ista kao trenutna i on uspeva da ga sačuva, menjajući verziju na Y. Mika pokušava da sačuva odgovor, ali verzija žalbe koju je pročitao i koja je sad u bazi nisu iste, Mika dobija grešku!
- Opisani način predstavlja optimistički pristup. Dodato je polje *version* u klasi *Complaint* koje označava trenutnu verziju podatka u bazi. Smatram da će se situacije gde admini sistema istovremeno odgovaraju na jednu žalbu relativno retko dešavati i da *rollback* nije previše „skup“, pa nema potrebe za striktnijim zaključavanjem. Primer konflikta je dat u nastavku.



Konfliktna situacija 3: Pravljenje više istovremenih ponuda jednog dobavljača

- Može se desiti da jedan dobavljač pokuša da istovremeno (na primer iz dva pretraživača) da ponudu za dve različite narudžbine. Postoji mogućnost da on na stanju ima lekove potrebne samo za jednu od te dve narudžbine. Ukoliko bi se desilo da oba zahteva proveravaju stanje istog leka (ili istih lekova) istovremeno, možemo doći u nekonzistentu situaciju tako što će ponude za obe narudžbine proći, iako dobavljač nema dovoljno lekova na stanju za obe.
- Ovaj konflikt je rešen korišćenjem pesimističkog zaključavanja. Metoda repozitorijuma koja dobavlja stavku dobavljača (klasa *SupplierItem*) za dati lek je anotirana sa *@Lock(LockModeType.PESSIMISTIC_WRITE)*, što sprečava i čitanje i menjanje ovog podatka. Ovim postizemo to da, ukoliko dobavljač pokuša da istovremeno pošalje nekoliko zahteva i „prevari“ sistem, dobije grešku. Kako je ovo izuzetno redak slučaj (dobavljač svesno i namerno mora da izazove događaj), smatram pesimističko zaključavanje dobrim izborom. Takođe, zbog karakteristika uslova u kojima može doći do konflikta, *timeout* metode za čitanje je postavljen na 0. Ukoliko se desi da u međuvremenu nestane neki od lekova koji još uvek nije zaključan (a treba da bude u nekoj od narednih iteracija), baciće se *CustomException* sa odgovarajućom porukom i transakcija će biti *rollback*-ovana.

