

KONKURENTNOST

ISA_2021-tim6

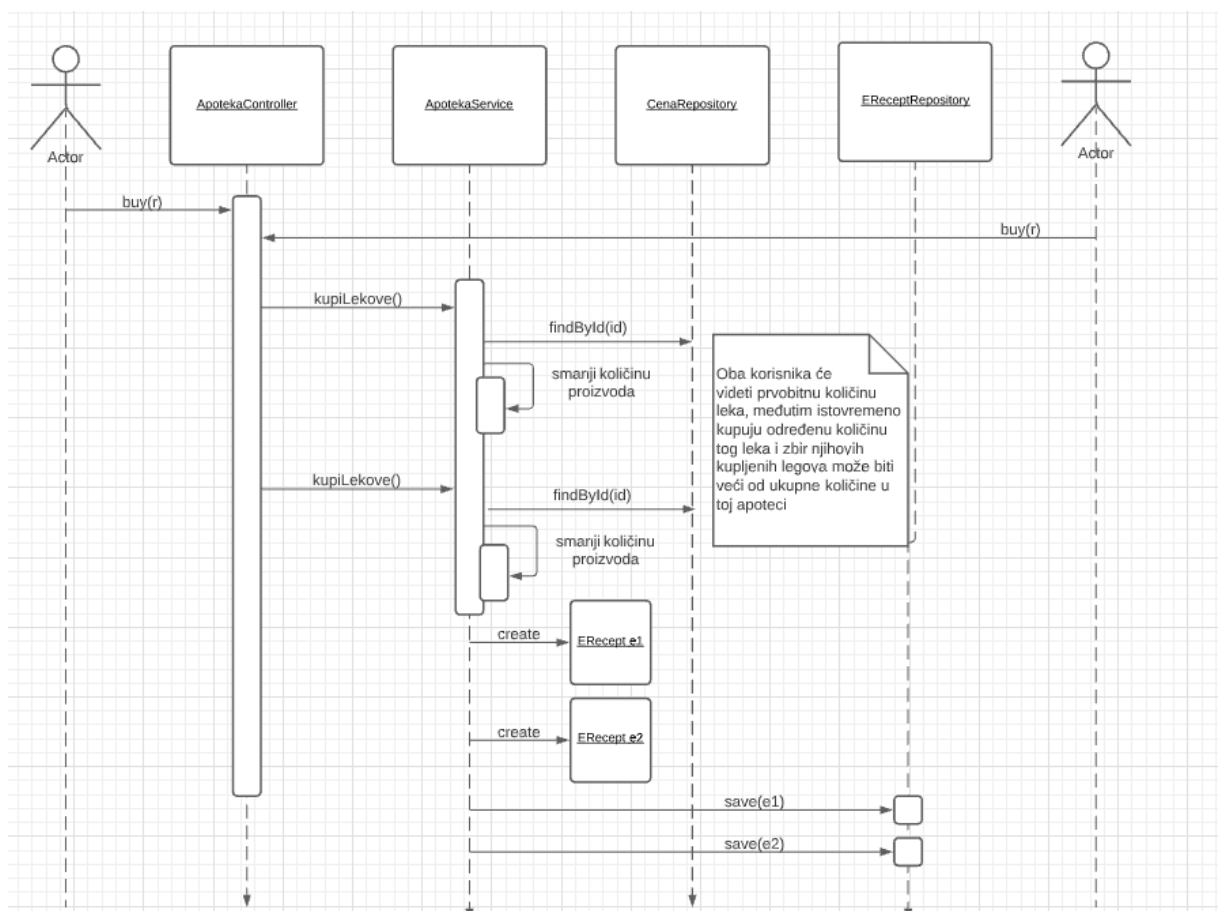
Petar Marković

SW-73-2018

***Napomena:** Sve servisne klase su anotirane sa `@Transactional` anotacijom, tako da se pomenute *commit* i *rollback* operacije obavljaju nad čitavom metodom a ne samo nad pojedinačnim upitom (što ne bi imalo smisla). Takođe, ovim putem se doseg pesimističkog zaključavanja prenosi na čitavo telo metode.

Scenario 1:

- Prilikom izdavanja eRecepta se izdaju ili svi ili ni jedan lek i stanje leka u apoteci se ažurira.



Dva korisnika istovremeno pokušavaju da rezervišu lek kod kojeg je trenutna količina u apoteci = N. Prvi korisnik pokušava da rezerviše L količina tog leka, a drugi M količina. Prilikom čitanja iz baze, stanje je N za oba korisnika i pritom je $N > M$ i $N > L$ te provera dostupnosti prolazi, oba stanja se

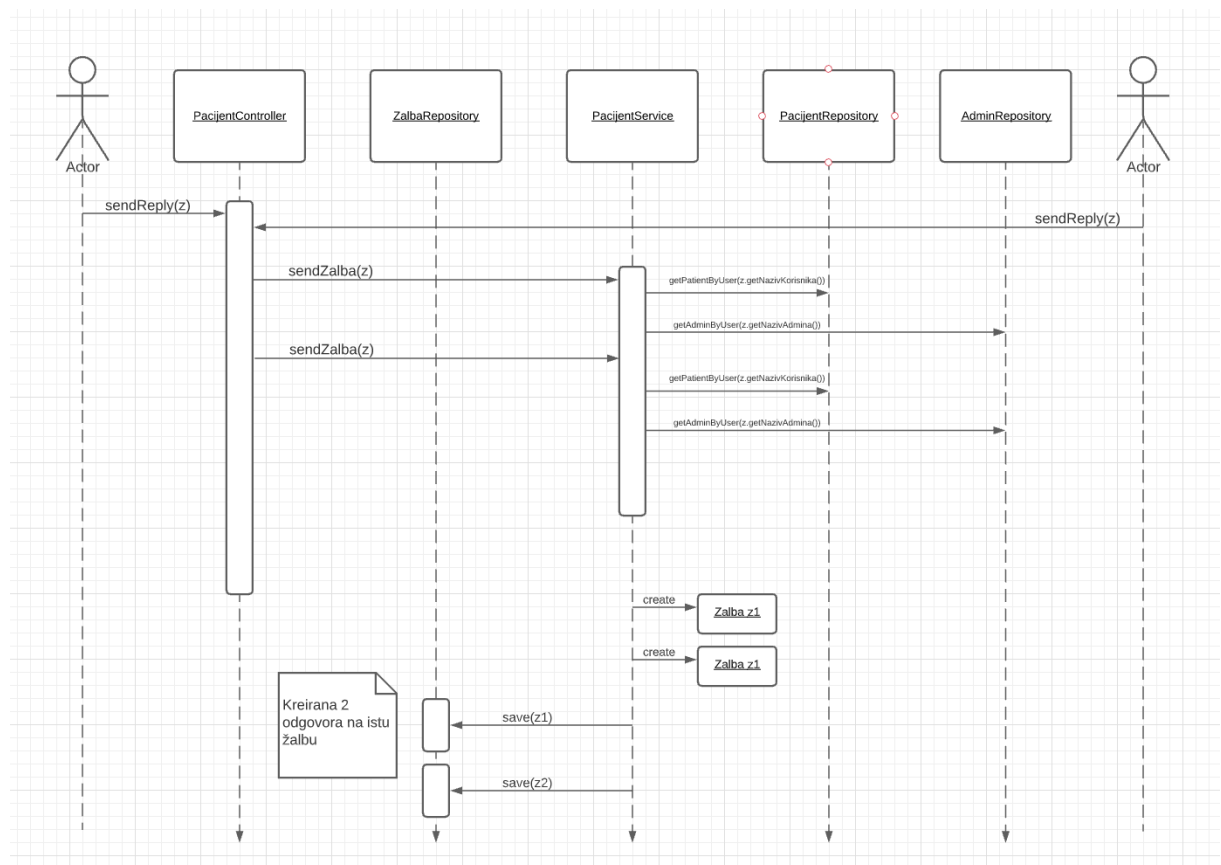
lokalno ažuriraju na N - M i N - L, i promene se commit-uju na bazu. Ovim su oba korisnika uspešno rezervisali lek a finalno stanje u bazi neće biti N - M - L.

Rešenje:

Uveden je mehanizam pesimističkog zaključavanja. Prilikom dobavljanja cenovnika apoteke metodom *findById(int id)* u klasi *CenaRepository* vrši se pesimističko zaključavanje tabele cenovnika (pessimistic write).

Scenario 2

- Na jednu žalbu može da odgovori samo jedan administrator sistema.



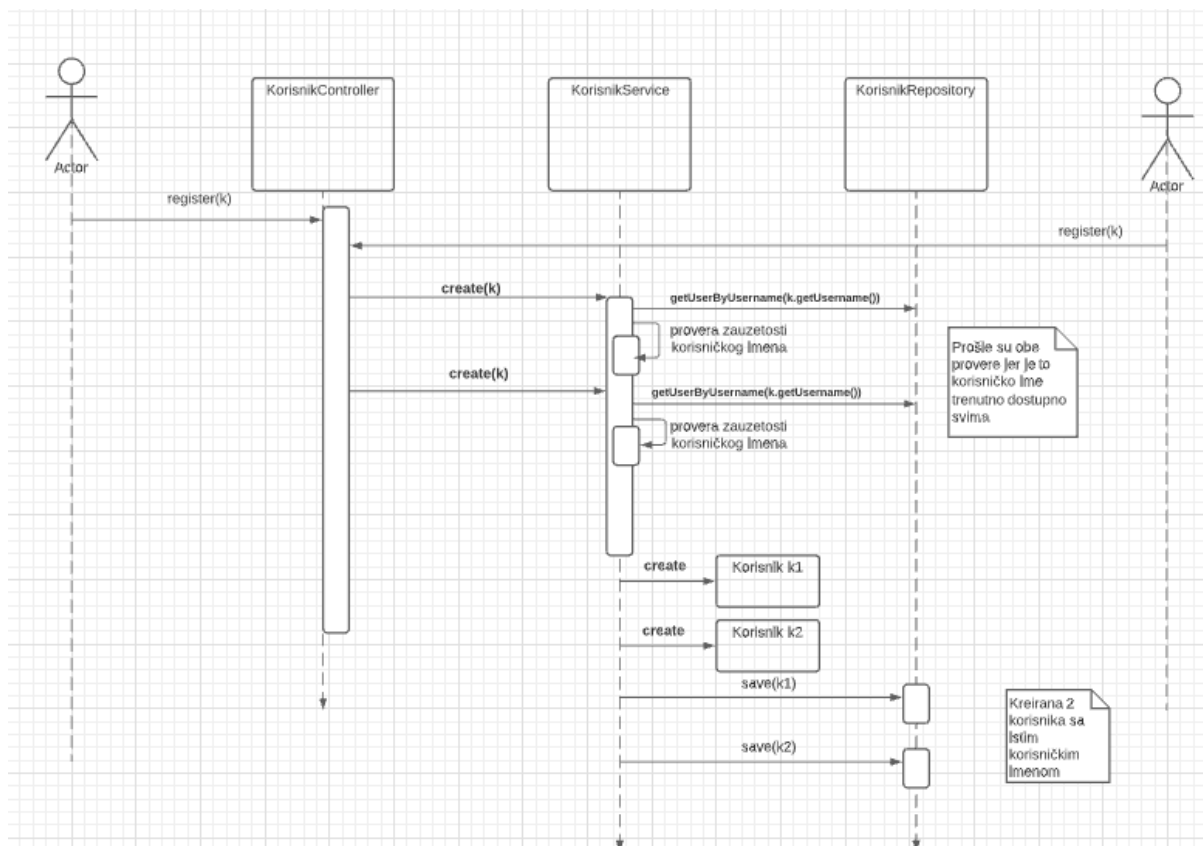
Dva administratora sistema u isto vreme pokušavaju da odgovore na istu žalbu jednog korisnika. Prilikom čitanja iz baze, korisnikova žalba je obojici prikazana kao neodgovorena, te validacije prolaze. Lokalno se kreira nov odgovor na žalbu i *commit*-uje se na bazu. Oba administratora sistema su uspela da odgovore na žalbu u isto vreme.

Rešenje:

Uveden je mehanizam pesimističkog zaključavanja. Prilikom dobavljanja admina sistema metodom *findAdminByUser(String user)* u klasi *AdminRepository* vrši se pesimističko zaključavanje tabele admina (pessimistic write).

Scenario 3

- Svaki korisnik mora imati jedinstveno korisničko ime.



Dva korisnika u isto vreme pokušavaju da se registruju sa istim korisničkim imenom. Prilikom čitanja iz baze, uneto korisničko ime je obojici prikazano kao slobodno, te validacije prolaze. Lokalno se kreira nov korisnik i *commit*-uje se na bazu. Oba korisnika su uspela da se registruju sa istim korisničkim imenom.

Rešenje:

Uveden je mehanizam pesimističkog zaključavanja. Prilikom dobavljanja korisnika sa korisničkim imenom upotrebom metode `getUserByUsername(String user)` u klasi `UserRepository` vrši se pesimističko zaključavanje tabele korisnika (pessimistic write).