

## Projekat iz predmeta Internet softverske arhitekture i Metodologije razvoja softvera

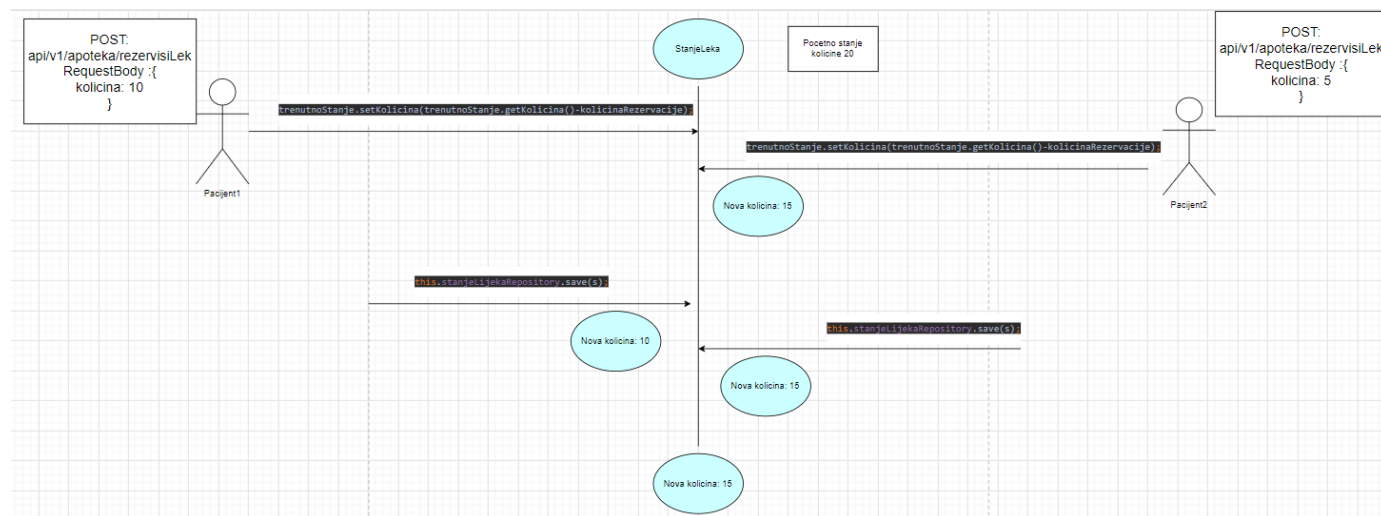
### Konkurentan pristup resursima u bazi

Rajtarov Nataša sw-2-2018

Kako bi baza podataka imala uvek ispravne podatke u sebi, potrebno je da pristup bazi bude konkurentan. Potrebno je da se očuva ispravnost podataka, što znači da je potrebno upisati podatke u nju po redosledu nastanka. U projektu postoji više situacija kada može doći do konflikata, a neki od njih su: rezervisanje laka, otkazivanje leka, rezervisanje pregleda kod dermatologa, rezervisanje savetovanja kod farmaceuta, i ocenjivanje dermatologa, farmaceuta, leka i apoteke. U nastavku rada biće objašnjeni načini rešavanja navedenih problema.

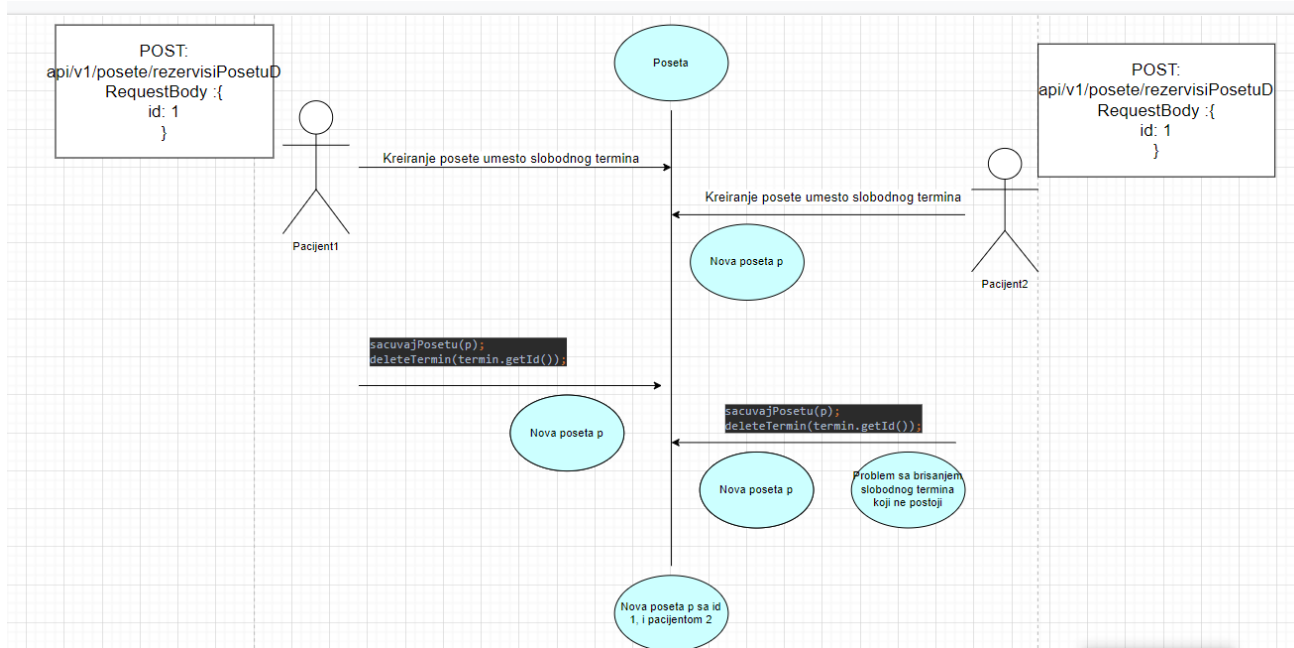
#### Rezervisanje leka

Prilikom rezervisanja leka potrebno je obezbediti da količina leka na stanju uvek bude ispravna, i da ne može da se desi da dva čoveka u isto vreme rezervišu isti lek, i da ne može da se pojavi negativna količina leka na stanju. Za rešavanje ovog problema odabrano je optimističko zaključavanje. Dodata je anotacija `@Version` iznad atributa `version` u klasi `stanjeLek-a`. Takođe je iznad metode `rezervisiLek` dodata anotacija `@Transactional`. U kombinaciji ove dve anotacije proveravaće se da li je verzija objekta koja je preuzeta na početku transakcije ista u toku cele transakcije i ako se uvidi da je došlo do promena verzije izaziva se `ObjectOptimisticLockingFailureException` nakon koje se baza vraća u prethodno stanje.



Prilikom otkazivanja leka potrebno je omogućiti da se količina dostupnog leka adekvatno ažurira, i da ne može da se desi da jedan isti pacijent u isto vreme otkaže istu rezervaciju. Za rešavanje ovog problema odabrano je optimističko zaključavanje. Dodata je anotacija `@Version` iznad atributa `version` u klasi `stanjeLek-a`. Takođe je iznad metode `izbaciRezervaciju` dodata anotacija `@Transactional`. U kombinaciji





Prilikom ocenjivanja potrebno je omogućiti da se ne oceni isti predmet od strane istog pacijenta u istom trenutku 2 puta. Za rešavanje ovog problema odabrano je optimističko zaključavanje. Dodata je anotacija `@Version` iznad atributa `version` u klasi `Ocena` i u klasi `Poseta`. Takođe je iznad metode `posaljiOcenu` dodata anotacija `@Transactional`. U kombinaciji ove dve anotacije proveravaće se da li je verzija objekta koja je preuzeta na početku transakcije ista u toku cele transakcije i ako se uvidi da je došlo do promena verzije izaziva se `ObjectOptimisticLockingFailureException` nakon koje se baza vraća u prethodno stanje.

