

Konfliktne situacije

1. Opis konfliktnih situacija koje su rjesavane

1.1 Prihvatanje i odbijanje zahtjeva za registraciju

Ukoliko dva administratora klinickog centra odaberu jedan zahtjev za registraciju da obojica odbiju ili da jedan prihvati, a drugi da odbije ili da obojica odbiju dolazi do konfliktne situacije u kojoj se moze otvoriti nalog pacijenta, a da dobije obavjestenje da je odbijen ili suprotno ili cak da se pokusa pravljenje dva ista naloga.

1.2 Ovjera recepata

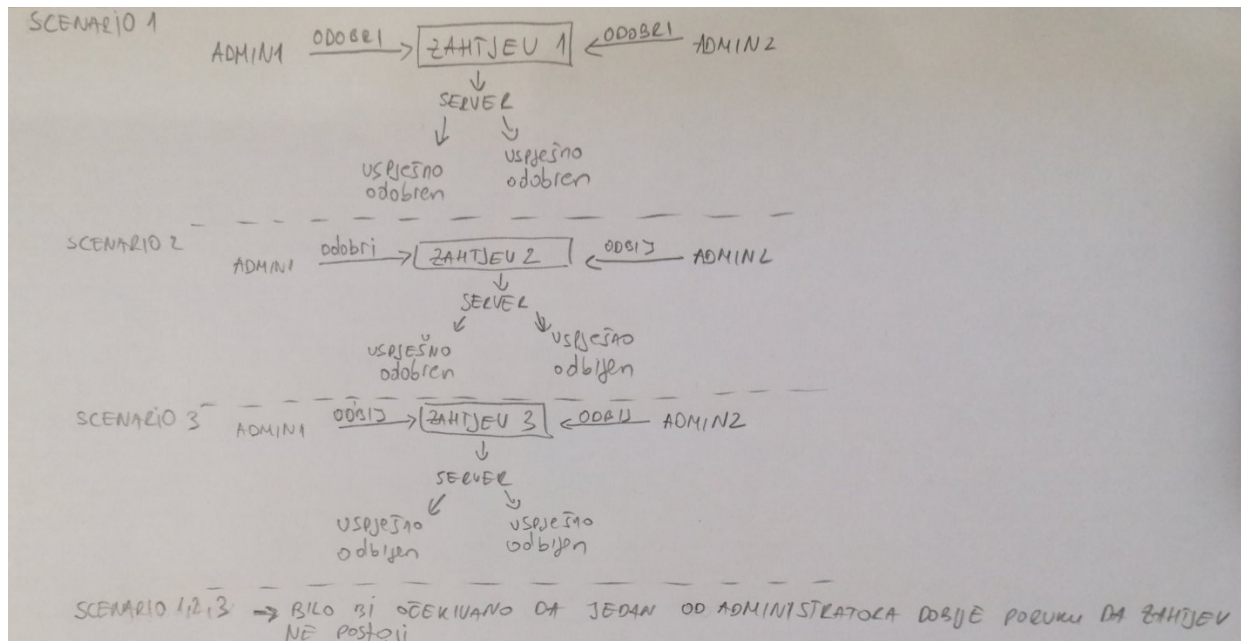
Ukoliko dvije ili vise sestara pokusaju da ovjere isti recept sistem ce evidentirati, da je ona sestra koja je poslednja poslala zahtjev, ovjerila recept sto narusava konzistentnost i imamo pogresnu informaciju koja je sestra ovjerila recept.

1.3 Odobravanje zahtjeva za operaciju

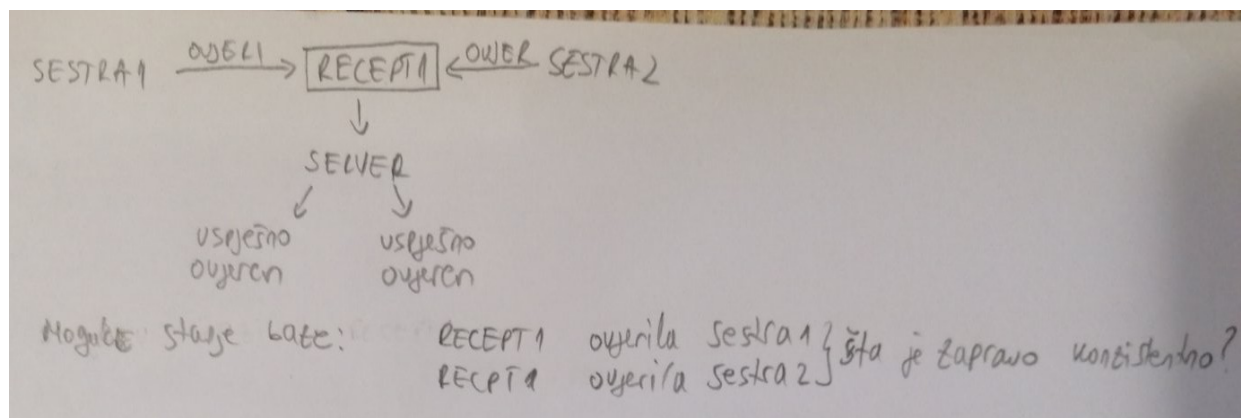
Ukoliko vise administratora klinike zeli da odobri isti zahtjev za registraciju moze doci do vise konflikata:

- od jednog istog zahtjeva ce se napraviti dva entiteta operacije
- mogu se dodijeliti isti doktori za razlicite operacije koje su istovremene
- moze se dodijeliti vise operacija jednoj sobi u isto vrijeme

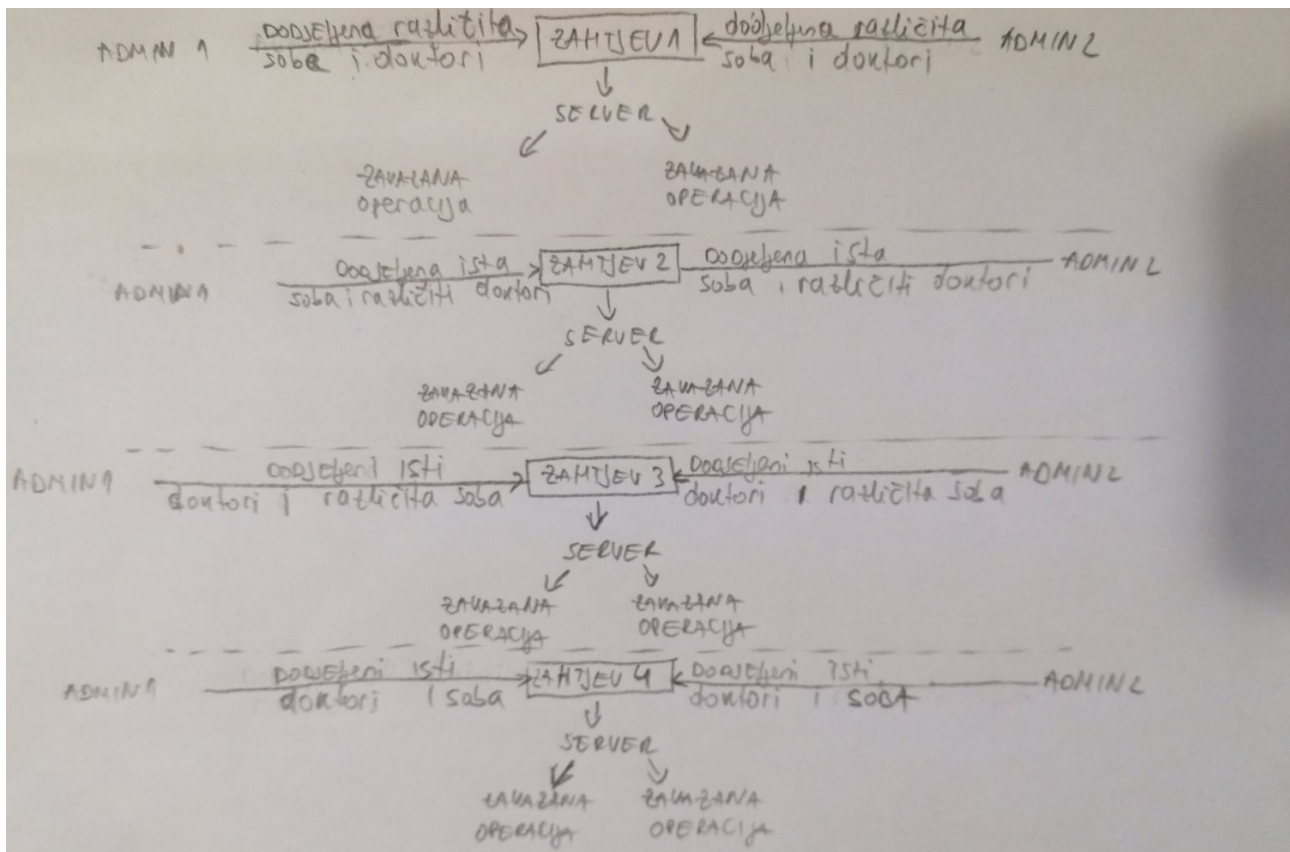
2. Opis konfliktnih situacija koje su rješavane



Slika 2.1 – konflikti kod prihvatanja zahtjeva za registraciju



Slika 2.2 – konflikti kod ovjere recepta



Slika 2.3 – konflikti kod prihvatanja zahtjeva za operaciju

3. Opis nacina na koji su situacija rjesavane

3.1 Pihvatanje i odbijanje zahtjeva za registraciju

U klasi modela zahtjeva za registraciju dodato je polje version koje je anotorano za `@Version` cime je ono naznaceno za pracenje verzije. Iznad metoda koje prihvataju i odbijaju zahtjeve stavljena je anotacija `@Transactional(propagation = Propagation.REQUIRES_NEW)`

cime je naznaceno da sa pocetkom metode pocinje izvorsavanje nove transakcije. Ovim metodom je izvršen optimistic lock ovog resursa.

3.2 Ovjera receptata

U klasi modela za recepte dodato je polje version koje je anotorano za `@Version` cime je ono naznaceno za pracenje verzije. Iznad metoda koja ovjerava recepte stavljena je anotacija `@Transactional(propagation = Propagation.REQUIRES_NEW)`

cime je naznaceno da sa pocetkom metode pocinje izvorsavanje nove transakcije. Ovim metodom je izvršen optimistic lock ovog resursa.

3.3 Odobravanje zahtjeva za operaciju

U klasi modela za zahtjeve za operaciju, doktora i sobu dodato je polje `version` koje je anotirano sa `@Version` cime je ono naznaceno za pracenje verzije. Iznad metoda koja ovjerava recepte stavljena je anotacija `@Transactional(propagation = Propagation.REQUIRES_NEW)` cime je naznaceno da sa pocetkom metode pocinje izvorsavanje nove transakcije. Posto se jedino nad resursom zahtjev za operaciju izvorsava update ili delete po defaultu bice primjeceno nekonzistentno stanje ako vise admina ze da odobre isti zahtjev tako da je optimisic lock izvrsen nad ovim resursom. Nad resursima doktor i soba se ne vrsi nikakav update i delete pa nece biti primjeceno da se verzija promijenila, ali u metodi se za sobu I za doktore poziva metoda `findOneById` iz njihovih servisa radi dobavljanja potrebnih podataka, ta metoda je zapravo obmotac za istu metodu repozitorija. Ove metode u repozitorijumu su anotirane sa `@Lock(LockModeType.OPTIMISTIC_FORCE_INCREMENT)` sto ce zapravo prilikom poziva metode uvecati polje za pracenje verzije tako da kod pristizanja sledeceg zahtjeva dolazi do optimistic lock exceptiona.