

## Jovana Samardzic RA99/2017 – Student 4

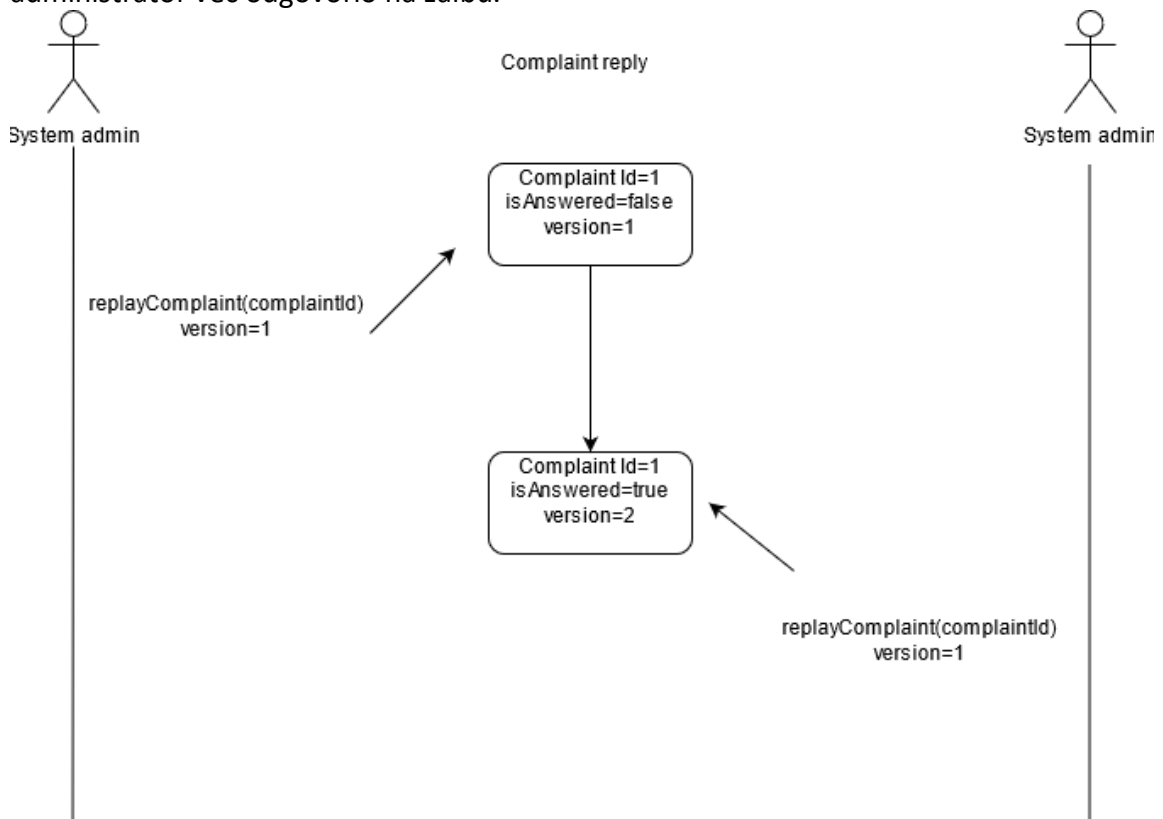
1. Na jednu žalbu može da odgovori samo jedan administrator sistema.

- Opis konfliktne situacije

Administrator sistema ima uvid u sve žalbe koje pacijenti salju i ima mogućnost da odgovori na te žalbe. Prilikom klika na dugme Reply na backend-u se pronadje žalba na koju administrator zeli da posalje odgovor i ukoliko je boolean polje `isAnswered` kod žalbe `false` odgovor se uvazava i salje pacijentu na mail. Problem nastaje ukoliko dva sistem administratora zele da odgovore na istu žalbu i kliknu da dugme Reply u približno isto vrijeme. U tom slucaju ce biti proslijedjen odgovor na žalbu onog administratora koji je prvi poslao odgovor, a drugom administratoru treba da se onemoguci slanje odgovora.

- Rjesenje situacije

Rjesenje situacije je postignuto pomocu optimistic locking-a, koristenjem anotacije `@Version`. U klasi `Complaint` je dodato polje `Long version` sa anotacijom `@Version`. Pri cuvanju odgovora i mijenjanja boolean polja `isAnswered` na `true` provjerava se verzija podatka koji se cuva. Ukoliko je verzija podatka ista kao pri učitavanju iz baze podaci ce se sacuvati i polje `version` ce se povecati, i to ce se desiti kada podatke na backend posalje prvi administrator, dok ce se pri slanju podataka na backend u slucaju drugog administratora provjeriti verzija i neće odgovarati jer je prvi administrator vec odgovorio na žalbu.



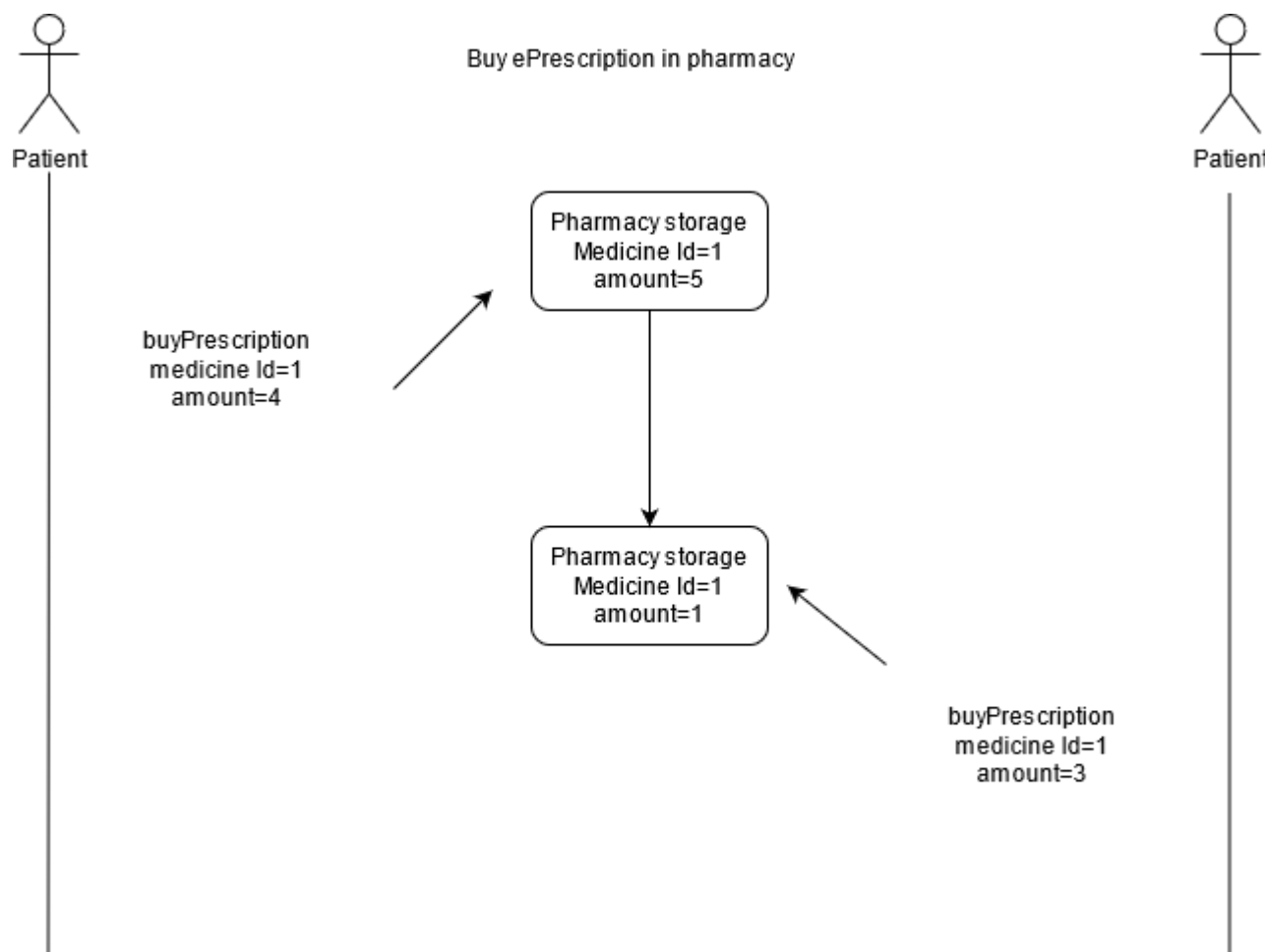
2. Prilikom izdavanja eRecepta se izdaju ili svi ili ni jedan lijek i stanje lijeka u apoteci se azurira.

- Opis konfliktne situacije

Pacijent ima mogućnost upload-ovanja QR koda na kome su učitani podaci o propisanim lijekovima i na osnovu učitanoog QR koda mu se nudi lista apoteka koje imaju sve lijekove u traženim količinama sa eRecepta. Apoteka mora imati sve lijekove na stanju, u suprotnom ne može izdati ni jedan lijek sa eRecepta. Pacijent može izabrati apoteku koja će mu izdati lijekove. Slanjem zahtjeva za izdavanje lijekova sa eRecepta na server potrebno je da se azuriraju količine svih traženih lijekova. Pri slanju zahtjeva, u međuvremenu može se promijeniti neka količina lijeka na stanju u toj apoteci i tada apoteka ne može izdati sve lijekove. Pacijentu se isčitavaju podaci iz baze a u isto vrijeme se mijenjaju u bazi i tada dolazi do neslaganja podataka.

- Rjesenje situacije

Rjesenje situacije je postignuto korištenjem anotacije @Transactional. Metoda buyPrescription() u EPrescriptionServisu je anotirana sa @Transactional(readOnly = false, propagation = Propagation.REQUIRES\_NEW), pri čemu je sa readOnly omogućena izmjena podataka.



3. U sistemu moze biti definisan samo jedan loyalty program u datom trenutku od strane jednog administratora sistema.

- Opis konfliktne situacije

U jednom vremenskom trenutku u sistemu apoteka moze biti definisan samo jedan loyalty program. Ukoliko administrator sistema zeli da vec definisani loyalty program zamijeni nekim novim mijenjace postojeći loyalty program u bazi. Ukoliko 2 administratora sistema ze le u isto vrijeme da definisu novi loyalty program doci ce do konflikta.

- Rjesenje situacije

Rjesenje situacije je postignuto pomocu optimistic locking-a, koristenjem anotacije @Version. U klasi LoyaltyProgram je dodato polje Long version sa anotacijom @Version. Prilikom cuvanja izmjena u bazi kod prvog administratora ukoliko je verzija podatka ista kao pri ucitavanju iz baze podaci ce se sacuvati i polje version ce se povecati, dok ce se pri slanju podataka na backend u slucaju drugog administratora provjeriti verzija i nece odgovarati jer je prvi administrator u tom trenutku izmijenio program.

