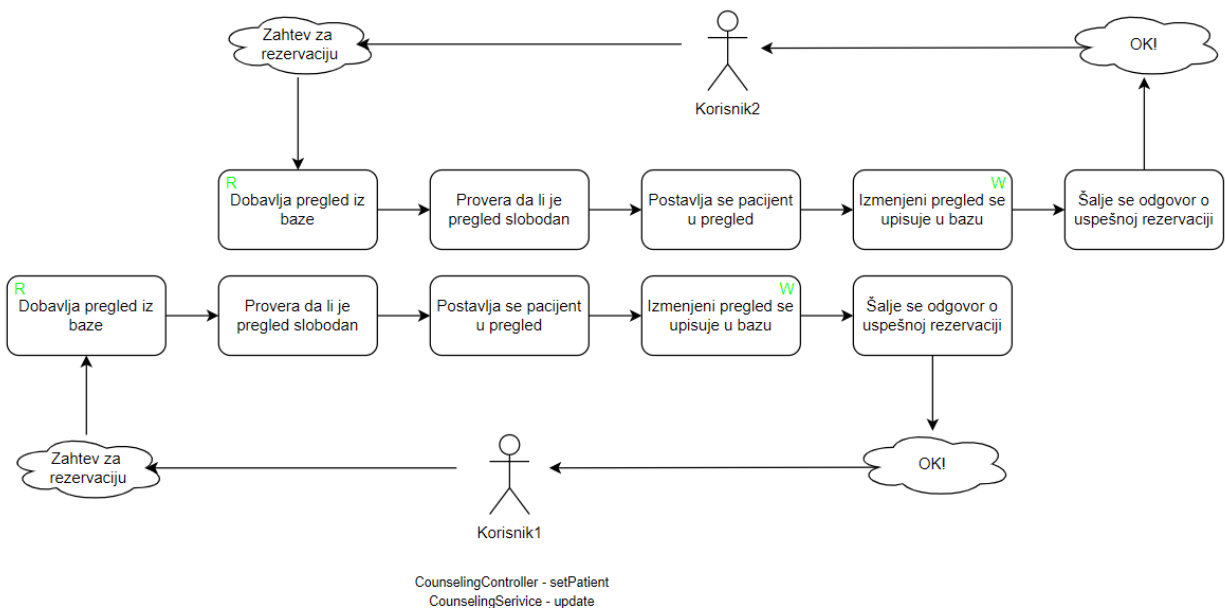


Rešenja konfliktnih situacija – student 3

SLUČAJ 1: više korisnika u isto vreme pokušava da zakaže isti, unapred definisani pregled kod dermatologa

Zakazivanje pregleda vrši se izborom jednog od ponuđenih slobodnih termina. Ovu funkcionalnost mogu da iniciraju pacijenti i/ili dermatolozi. Do konfliktne situacije dolazi kada dva ili više korisnika* pokušaju u isto ili slično vreme da zakažu isti pregled. Kao rezultat korisnici dobijaju informaciju da je zakazivanje termina uspešno. Ovakvo ponašanje sistema nije poželjno i potrebno ga je sprečiti.

Algoritam zakazivanja pregleda: Dobavlja slobodan pregleda iz baze, proverava se da li je pregled slobodan i ako jeste pacijent se upisuje u pregled. Izmenjeni pregled se čuva u bazi i korisnik dobija informaciju da je zakazivanje termina uspešno. Pretpostavimo da Korisnik1 traži zahtev za zakazivanje termina, neposredno nakon njega i Korisnik2 traži zahtev za isti termin. Korisnik1 uspešno završava zakazivanje. Za korisnika2 se i dalje obrađuje zahtev za pregled, koji više nije slobodan, jer je u njegovoj verziji taj pregled označen kao slobodan. Na ovaj način dobijamo situaciju da su dva korisnika zakazala jedan pregled.

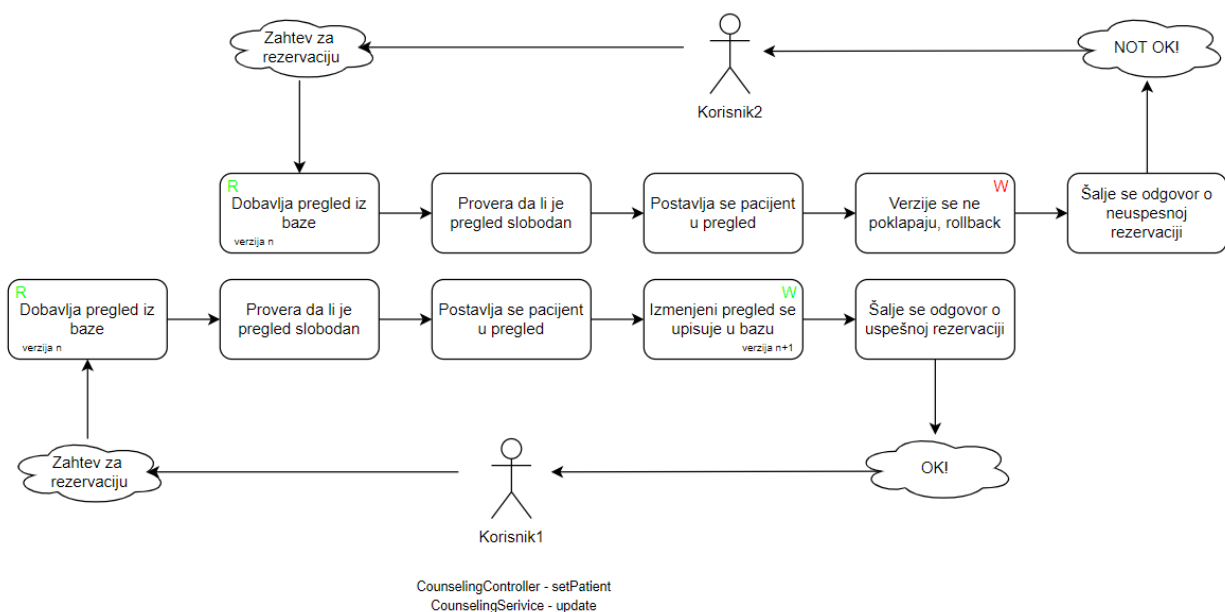


*U ovom slučaju pod korisnicima se podrazumevaju pacijenti i dermatolozi

R Uspešno čitanje iz baze W Uspešno pisanje u bazu W Neuspešno pisanje u bazu

Ovaj konflikt rešen je uz pomoć transakcija i optimističkog zaključavanja. Transakcije omogućavaju da se izmene podataka u BP unutar jedne transakcije ili sačuvaju (commit) ili odbace (rollback). Ovakvo ponašanje nam odgovara za rešavanje ovog problema. Optimističko zaključavanje je izabrano zbog boljih performansi u odnosu na pesimističko zaključavanje, ali i zbog toga što nije neophodno zaključavanje podataka. U tabelu *Counseling* dodato je polje *version*.

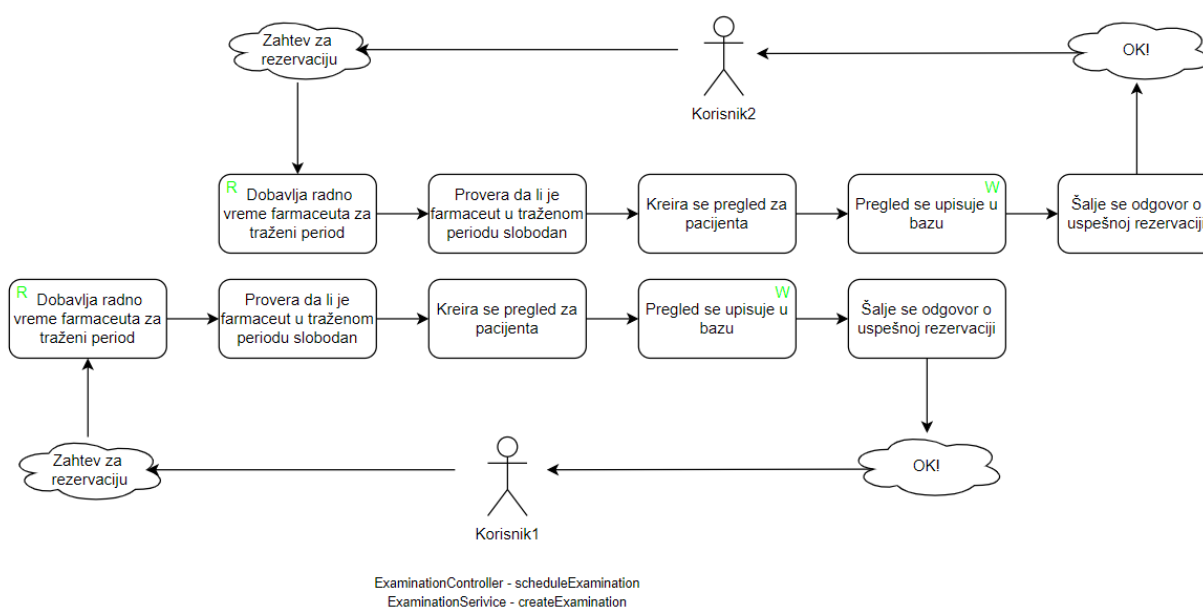
Kada Korisnik1 zatraži zahtev za zakazivanje pregleda, za njega se dobavlja pregled iz baze. Korisnik2 neposredno nakon Korisnika1 zatraži zahtev za zakazivanje za isti pregled. Za oba korisnika dobavljen je isti pregled sa istim verzijama. Korisnik1 upisuje pacijenta u pregled i čuva pregled u bazu. Za Korisnika2 upisuje se pacijent u pregled i pokušava se čuvanje pregled, ali to ne uspeva, jer se verzija dobavljenog pregleda ne poklapa sa trenutnom verzijom pregleda iz baze. Ova transakcija se rollbackuje i na ovaj način rešavamo konflikt.



SLUČAJ 2: više različitih korisnika u isto vreme pokušava da zakaže termine kod farmaceuta koji se međusobno preklapaju

Zakazivanje pregleda kod farmaceuta vrši se izborom dana, vremena početka i trajanja pregleda. Ovu funkcionalnost mogu da iniciraju pacijenti i/ili farmaceuti. Do konfliktne situacije dolazi kada dva ili više korisnika* pokušaju u isto ili slično vreme da zakažu preglede koji počinju u isto vreme ili se međusobno preklapaju. Kao rezultat korisnici dobijaju informaciju da je zakazivanje termina uspešno. Ovakvo ponašanje sistema nije poželjno i potrebno ga je sprečiti.

Algoritam zakazivanja pregleda: Dobavlja radno vreme (smenu) farmaceuta za kojeg se pregled definiše, proverava da li je farmaceut slobodan za traženi period. Ukoliko je farmaceut slobodan, upisuje se novi pregled za pacijenta u bazu za i šalje se informaciju da je zakazivanje termina uspešno. Pretpostavimo da Korisnik1 traži zahtev za zakazivanje termina, neposredno nakon njega i Korisnik2 traži zahtev za isti termin. Korisnik1 uspešno završava zakazivanje. Za korisnika2 se i dalje obrađuje zahtev za pregled, iako farmaceut u tom periodu više nije slobodan. Na ovaj način dobijamo situaciju da su dva korisnika zakazala preglede čiji se termini preklapaju.

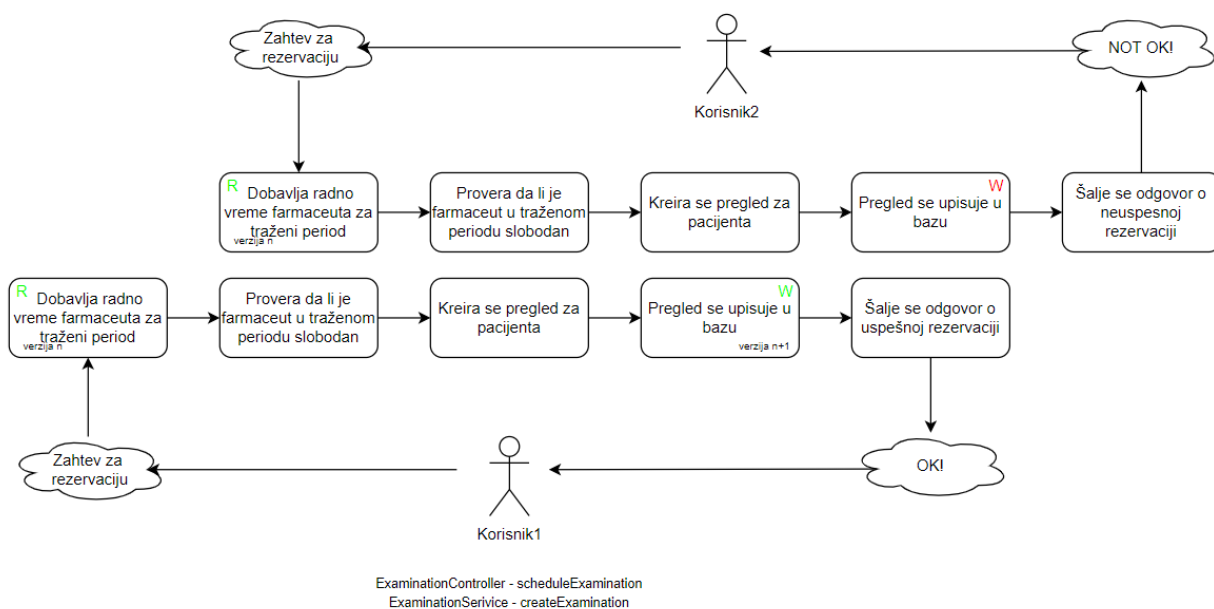


*U ovom slučaju pod korisnicima se podrazumevaju pacijenti i farmaceuti

R Uspešno čitanje iz baze W Uspešno pisanje u bazu W Neuspešno pisanje u bazu

Ovaj konflikt rešen je uz pomoć transakcija i optimističkog zaključavanja. Transakcije omogućavaju da se izmene podataka u BP unutar jedne transakcije ili sačuvaju (commit) ili odbace (rollback). Ovakvo ponašanje nam odgovara za rešavanje ovog problema. Optimističko zaključavanje je izabrano zbog boljih performansi u odnosu na pesimističko zaključavanje, ali i zbog toga što nije neophodno zaključavanje podataka. U tabelu *PharmacistWorkingPeriod* dodato je polje *version*, kao i polje *lastReqDate*, koje označava vreme kada je poslednji put izvršeno zakazivanje pregleda za smenu.

Kada Korisnik1 zatraži zahtev za zakazivanje pregleda kod nekog farmaceuta, on dobavlja radno vreme(smenu koja odgovara za taj pregled) iz baze. Korisnik2 neposredno nakon Korisnika1 zatraži zahtev za zakazivanje za isti pregled. Za oba korisnika dobavlja se smena sa istim verzijama. Za Korisnika1 se čuva pregled u bazu, nakon toga u prethodno dobavljenom radnom vremenu(smeni) postavlja se nova vrednost za *lastReqDate* i izmena se čuva u bazi. Za Korisnika2 upisuje se novi pregled u bazu, postavlja se nova vrednost za *lastReqDate* i pokušava čuvanje izmene u bazi, ali to ne uspeva, jer se verzija dobavljene smene ne poklapa sa trenutnom verzijom smene iz baze. Ova transakcija se rollbackuje(poništava se upis pregleda i smene) i na ovaj način rešavamo konflikt.

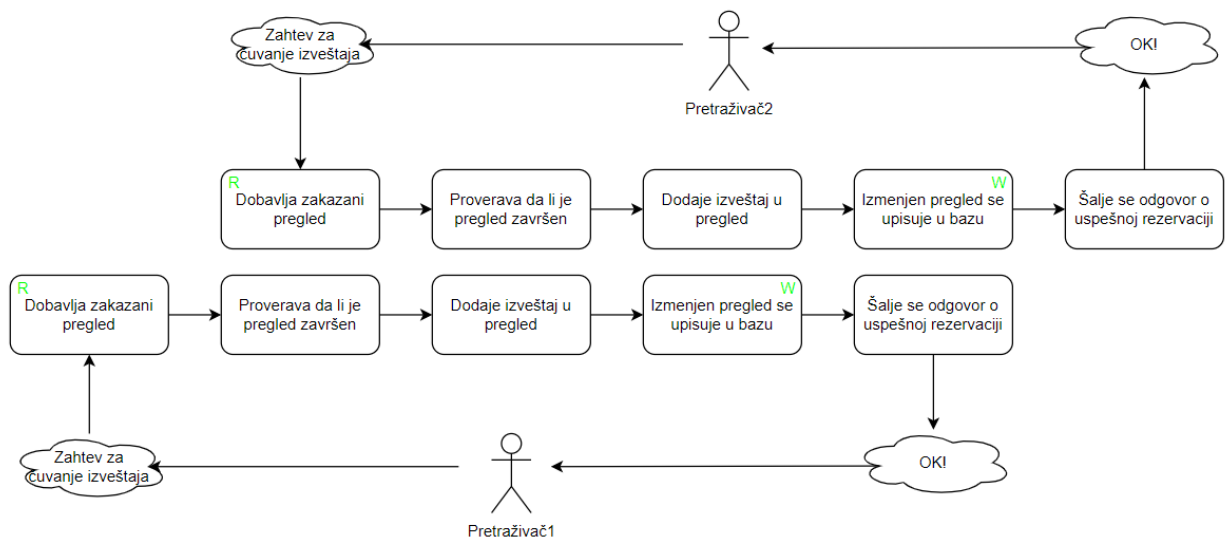


R Uspešno čitanje iz baze W Uspešno pisanje u bazu W Neuspešno pisanje u bazu

SLUČAJ 3: Dermatolog iz dva ili više pretraživača pokušava da sačuva izveštaj za jedan, isti pregled

Čuvanje izveštaja za pregled počinje tako što se započne zakazan pregled, zatim se popuni izveštaj i potvrdi unos. Ovu funkcionalnost mogu da iniciraju dermatolozi. Do konfliktne situacije dolazi kada dermatolog iz dva ili više pretraživača pokuša u isto ili slično vreme da sačuva izveštaj za jedan pregled. Kao rezultat korisnik preko pretraživača dobija informaciju da je čuvanje izveštaja uspešno. Ovakvo ponašanje sistema nije poželjno i potrebno ga je sprečiti.

Algoritam čuvanja izveštaja: Dobavlja zakazan pregleda iz baze, proverava se da li je pregled završen (popunjen izveštaj), ukoliko nije, izveštaj se upisuje u pregled. Pregled se čuva u bazi i korisnik dobija informaciju da je čuvanje izveštaja uspešno. Pretpostavimo da dermatolog sa Pretraživača1 zatraži zahtev za čuvanje izveštaja, neposredno nakon toga dermatolog sa Pretraživača2 traži zahtev za čuvanje izveštaja za isti pregled. Dermatolog sa Pretraživača1 uspešno završava čuvanje izveštaja. Za Pretraživa2 se i dalje obrađuje zahtev za čuvanje izveštaja pregleda, koji je već sačuvan. Na ovaj način dobijamo situaciju da je dermatolog iz dva pretraživača dva puta sačuvao izveštaj za jedan pregled.



CounselingController - setReport
CounselingService - updateReport

Ovaj konflikt rešen je uz pomoć transakcija i optimističkog zaključavanja. Transakcije omogućavaju da se izmene podataka u BP unutar jedne transakcije ili sačuvaju (commit) ili odbace (rollback). Ovakvo ponašanje nam odgovara za rešavanje ovog problema. Optimističko zaključavanje je izabrano zbog boljih performansi u odnosu na pesimističko zaključavanje, ali i zbog toga što nije neophodno zaključavanje podataka. U tabelu *Counseling* dodato je polje *version*.

Kada dermatolog putem Pretraživača1 zatraži zahtev za čuvanje izveštaja za pregled, za njega se dobavlja zakazani pregled iz baze. Dermatolog putem Pretraživača2, neposredno nakon što je putem Pretraživača1 zatražio zahtev, takođe traži zahtev za čuvanje izveštaja za isti pregled. Za oba pretraživača dobavljen je isti pregled sa istim verzijama. Za Pretraživač1 upisuje se izveštaj u pregled i tako izmenjen pregled se čuva u bazu. Za Pretraživač2 upisuje se izveštaj u pregled i pokušava se čuvanje pregled, ali to ne uspeva, jer se verzija dobavljenog pregleda ne poklapa sa trenutnom verzijom pregleda iz baze. Ova transakcija se rollbackuje i na ovaj način rešavamo konflikt.

