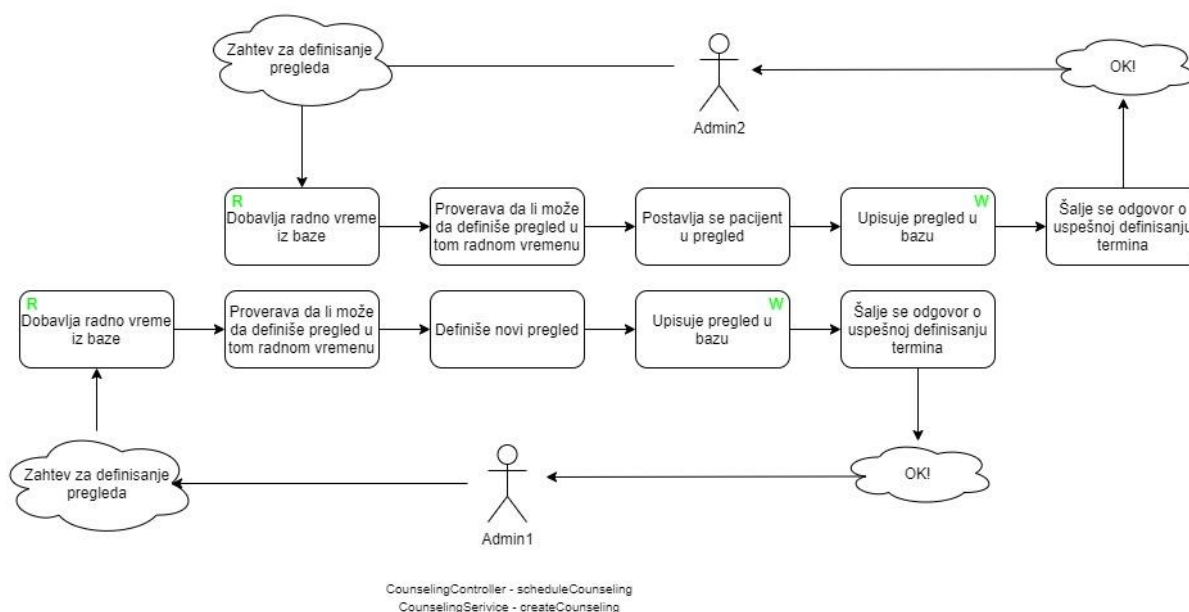


Rešenja konfliktnih situacija – student 2

SLUČAJ 1: jedan dermatolog je istovremeno prisutan na više različitih pregleda

Administrator apoteke definiše preglede kod dermatologa u terminu kada je dermatolog u apoteci. Za svaki pregled odabere datum i vreme početka pregleda, trajanje i cenu. Nakon toga u radnom kalendaru dermatologa se pojavljuju unapred definisani pregledi. Ovu funkcionalnost iniciraju administratori apoteke, te do konfliktne situacije dolazi ako dva ili više administratora iste apoteke pokušaju da u isto ili slično vreme zakažu termine koji se vremenski preklapaju, te će tada jedan administrator videti jedan predefinisani termin za dermatologa, a drugi administrator drugi predefinisani termin, a dermatolog će biti prisutan na dva pregleda istovremeno, ako se oba termina popune.

Algoritam definisanja pregleda kod dermatologa: Dobavlja radno vreme dermatologa za taj dan, upiše novi pregled za taj radni dan, sačuva novi pregled u bazu i šalje administratoru informaciju da je pregled uspešno definisan. Pretpostavimo da Admin1 traži zahtev za definisanje pregleda kod dermatologa, neposredno nakon njega i Admin2 traži zahtev za definisanje pregleda u istom danu i sa vremenom početka i trajanja koji se kose sa definisanim pregledom koji je zadao Admin1. Admin1 uspešno završava definisanje pregleda. Za Admin2 se i dalje obrađuje zahtev za definisanje pregleda, koji više ne bi smeo da može da se definiše jer je u tom vremenskom trenutku već definisan pregled od strane Admin1. Međutim Admin2, definisani pregled od strane Admin1 se ne vidi, te on smatra da može da definiše svoj pregled. Na ovaj način dobijamo da jedan dermatolog treba da bude na dva različita pregleda koji se vremenski preklapaju.



Ovaj problem je rešen pomoću 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.

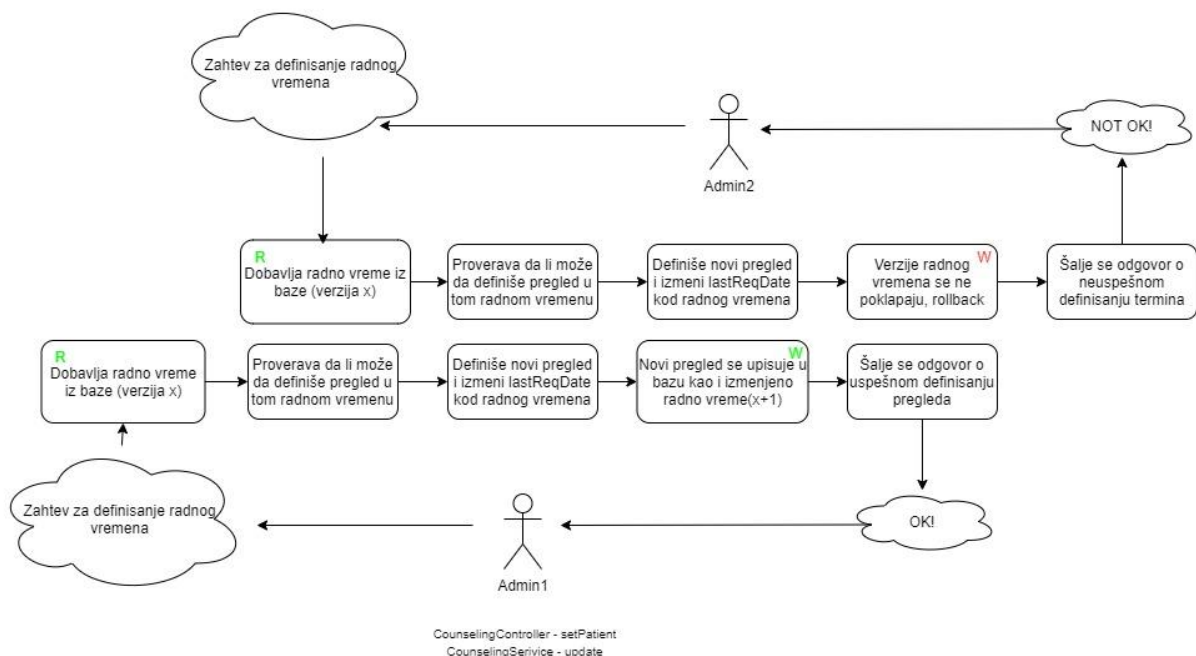
R Uspešno čitanje iz baze

W Uspešno pisanje u bazu

W Neuspešno pisanje u bazu

Optimističko zaključavanje je izabrano zbog boljih performansi u odnosu na pesimističko zaključavanje, ali i zbog toga što nije neophodno zaključavanje podataka. Zbog ovoga je u tabelu Dermatologist_working_period , dodato polje version.

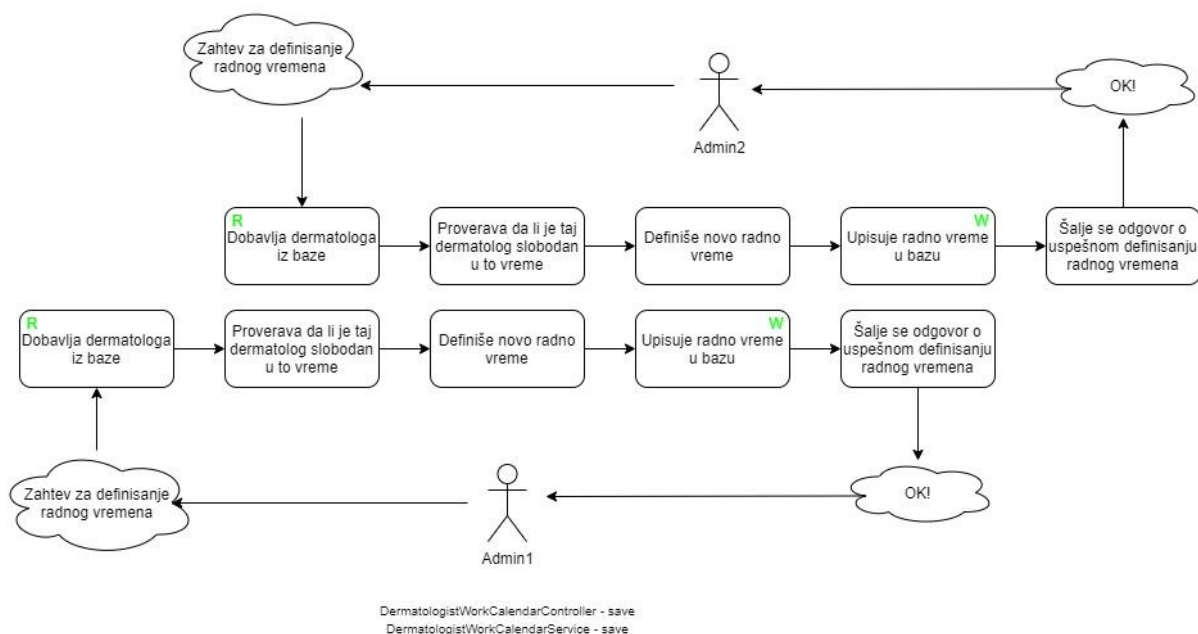
Kada Admin1 zatraži zahtev za definisanje pregleda, on dobavlja radno vreme lekara za taj dan iz baze. Admin2 neposredno nakon toga zatraži zahtev za definisanje pregleda u istom danu, ali sa drugim vremenom početka. Oba admina dobijaju nazad radno vreme dermatologa sa istim verzijama. Admin1 definiše pregled kod dermatologa i čuva pregled u bazu. Takođe izmeni polje lastReqDate u radnom vremenu dermatologa. To polje govori o tome kada je poslednji put potraživano radno vreme dermaologa radi definisanja novih pregleda. Radno vreme dermatologa se čuva u bazi. Admin2 definiše pregled kod dermatologa i pokušava da sačuva taj pregled u bazu. Takođe izmeni polje lastReqDate i pokuša da sačuva radno vreme dermatologa u bazu, ali to ne uspeva, jer se verzija radnog vremena ne poklapa sa trenutnom verzijom radnog vremena iz baze.(izmenila se zbog definisanog pregleda Admin1). Ova transakcija se rollbackuje i na ovaj način rešavamo konflikt.



SLUČAJ 2: jedan dermatolog radi u dve apoteke u isto vreme

Dermatolog može da bude zaposlen u više apoteke. Administrator apoteke definiše radno vreme dermatologa za svaki dan pojedinačno. Kako bi definisao radno vreme, on mora da odabere dan za koji će definisati, zatim satnicu od-do. Nakon toga u radnom kalendaru dermatologa se pojavljuje upisano radno vreme za taj dan u određenoj apoteci. Ovu funkcionalnost inicira administrator apoteke, te do konfliktne situacije dolazi ako dva ili više administratora iz različitih apoteka pokušaju da u isto ili slično vreme zakažu radna vremena dermatolozima. Tada može doći do situacije da jedan dermatolog ima definisana radna vremena u različitim apotekama, koja se preklapaju. Te iz toga proizilazi da jedan dermatolog može da radi u isto vreme u dve različite apoteke.

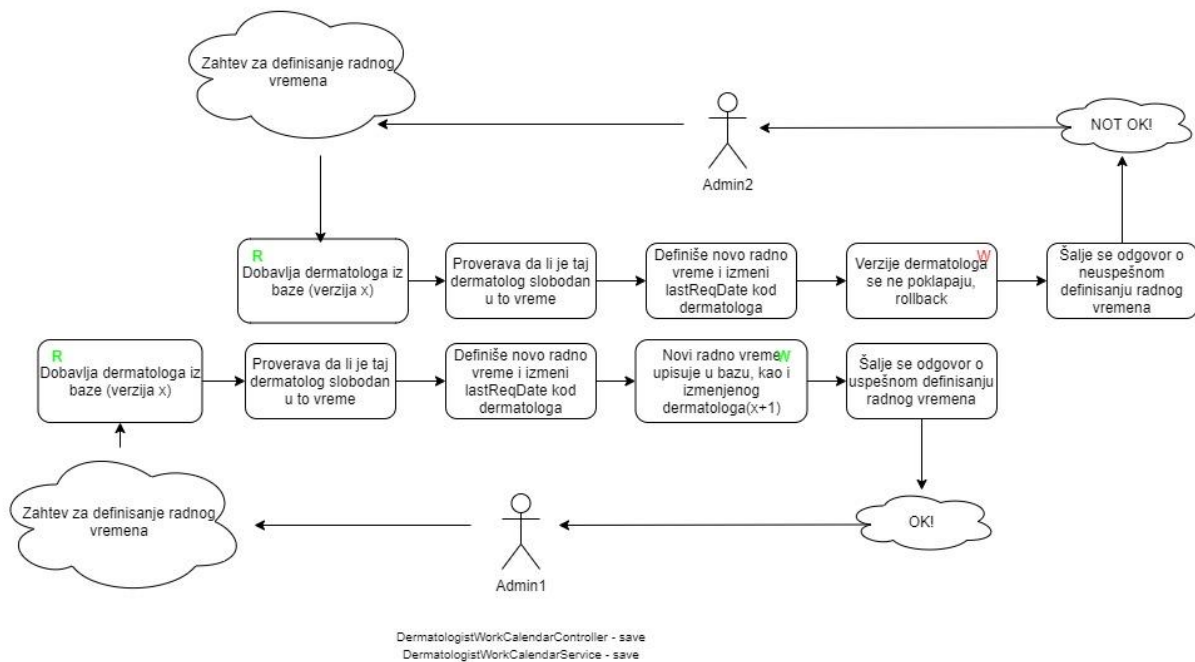
Algoritam definisanja radnog vremena za dermatologa: Dobavlja se dermatolog za koga se određuje radno vreme, zatim se proverava da li taj dermatolog već imati zadato radno vreme za taj dan i ako ima da li se satnice preklapaju. Ako se ništa ne preklapa, radno vreme može da se sačuva u bazu. Nakon toga šalje se informacija o uspešnom upisivanju novog radnog vremena. Pretpostavimo da Admin1 traži zahtev za definisanje radnog vremena, neposredno nakon njega i Admin2 traži zahtev za definisanje istog radnog vremena ali u nekoj drugoj apoteci. Admin1 uspešno završava definisanje radnog vremena. Za Admin2 se i dalje obrađuje zahtev za definisanje radnog vremena, koji više ne bi smeo da može da se definiše, jer je to radno vreme već zadato dermatologu, ali u drugoj apoteci. Međutim kod Admin2, definisano radno vreme od strane Admin1 se ne vidi, te on smatra da može da definiše i radno vreme od strane Admin2. Na ovaj način dobijamo da jedan dermatolog radi u isto vreme u dve različite apoteke.



Ovaj problem je rešen pomoću 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. Zbog ovoga je u tabelu Dermatologist(User), dodato polje version.

Kada Admin1 zatraži zahtev za definisanje radnog vremena, on dobavlja lekara iz baze i proverava da li je on slobodan taj dan i kada je slobodan. Admin2 neposredno nakon toga zatraži zahtev za definisanje radnog vremena u istom danu, ali naravno u drugoj apoteci. Oba admina dobijaju nazad dermatologa sa istim verzijama. Admin1 definiše novo radno vreme, upisuje ga u bazu, menja lastReqDate (polje koje govori o tome kada je poslednji put dermatolog potraživan radi izmene radnog vremena) kod dermatologa i sačuva i njega. Admin2 definiše radno vreme i lastReqDate, te pokušava da sačuva novo radno vreme i lekara u bazu, ali to ne uspeva, jer se verzija dermatologa ne poklapa sa trenutnom verzijom

dermatologa iz baze.(izmenila se zbog definisanog radnog vremena Admin1). Ova transakcija se rollbackuje i na ovaj način rešavamo konflikt.



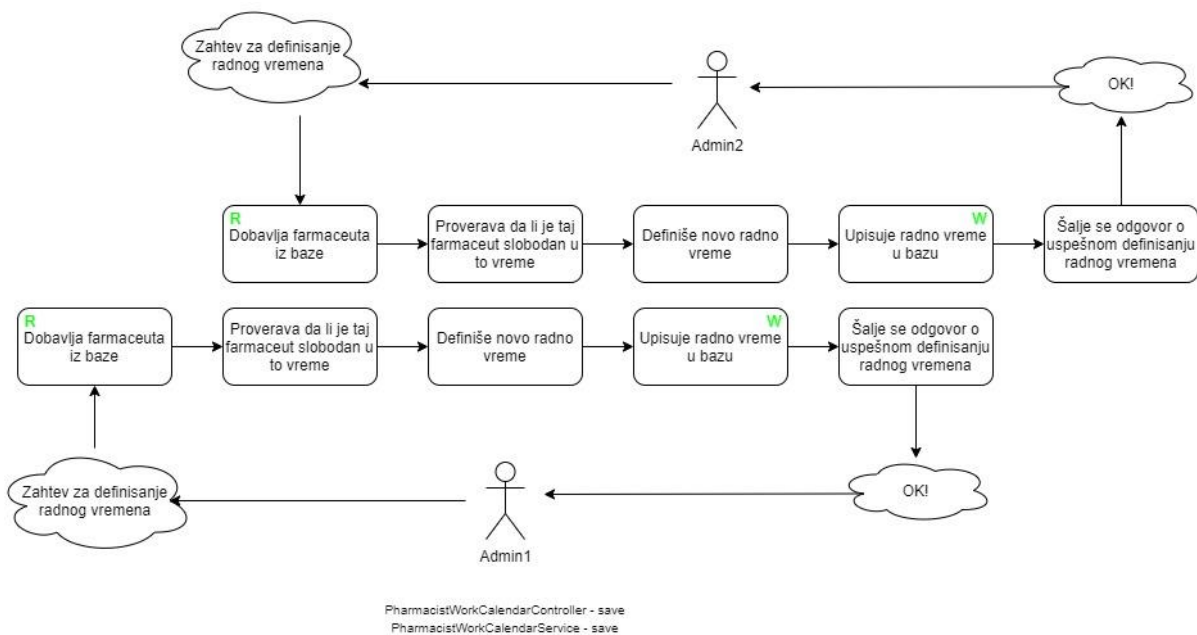
SLUČAJ 3: jedan farmaceut je istovremeno prisutan na više različitih savetovanja

S obzirom da je student 3 rešio konflikt vezan za istovremeno zakazivanje savetovanja kod farmaceuta i da je naša implementacija takva da farmaceut može da započne pregled samo dok on traje, jedini još način na koji jedan farmaceut može da bude prisutan na više različitih pregleda jeste da dva administratora u isto vreme zadaju različita radna vremena farmaceutu, te da se onda na osnovu tih radnih vremena zakazuju pregledi koji nisu usklađeni i dolazi do konflikta da je jedan farmaceut prisutan na više savetovanja.

Administrator apoteke definiše radno vreme farmaceuta za svaki dan pojedinačno. Kako bi definisao radno vreme, on mora da odabere dan za koji će definisati, zatim satnicu od-do. Nakon toga u radnom kalendaru farmaceuta se pojavljuje upisano radno vreme za taj dan. Ovu funkcionalnost inicira administrator apoteke, te do konfliktne situacije dolazi ako dva ili više administratora iz iste apoteke pokušaju da u isto ili slično vreme zakažu radno vreme farmaceutu. Tada može doći do situacije da jedan farmaceut ima definisana dva radna vremena za isti dan i na osnovu toga se mogu i zakazivati savetovanja, te jedan farmaceut može biti prisutan na više različitih savetovanja.

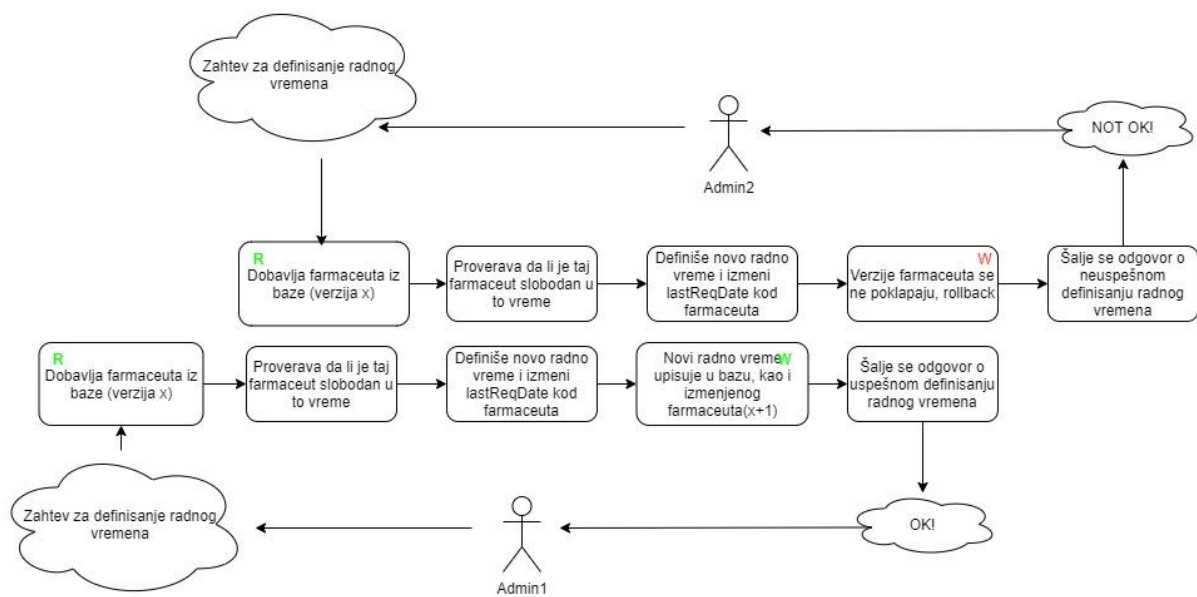
Algoritam definisanja radnog vremena za farmaceuta: Dobavlja se farmaceut za koga se određuje radno vreme, zatim se proverava da li taj farmaceut već ima zadato radno vreme za taj dan i ako ima da li se satnice preklapaju. Ako se ništa ne preklapa, radno vreme može da se sačuva u bazu. Nakon toga šalje se informacija o uspešnom upisivanju novog radnog

vremena. Pretpostavimo da Admin1 traži zahtev za definisanje radnog vremena, neposredno nakon njega i Admin2 traži zahtev za definisanje radnog vremena za tog farmaceuta. Admin1 uspešno završava definisanje radnog vremena. Za Admin2 se i dalje obrađuje zahtev za definisanje radnog vremena, koji više ne bi smeo da može da se definiše, jer je to radno vreme već zadato farmaceutu. Međutim definisano radno vreme od strane Admin1 se ne vidi, te se definiše i radno vreme od strane Admin2. Na ovaj način dobijamo da jedan farmaceut ima dva različita radna vremena.



Ovaj problem je rešen pomoću 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. Zbog ovoga je u tabelu Pharmacist(User), dodato polje version.

Kada Admin1 zatraži zahtev za definisanje radnog vremena, on dobavlja farmaceuta iz baze i proverava da li je on slobodan taj dan i kada je slobodan. Admin2 neposredno nakon toga zatraži zahtev za definisanje radnog vremena u istom danu(ista ili slična satnica). Oba admina dobijaju nazad farmaceuta sa istim verzijama. Admin1 definiše novo radno vreme, upisuje ga u bazu, menja lastReqDate (polje koje govori o tome kada je poslednji put farmaceut potraživan radi izmene radnog vremena) kod farmaceuta i sačuva i njega. Admin2 definiše radno vreme i lastReqDate, te pokušava da sačuva novo radno vreme i farmaceuta u bazu, ali to ne uspeva, jer se verzija farmaceuta ne poklapa sa trenutnom verzijom farmaceuta iz baze.(izmenila se zbog definisanog radnog vremena Admin1). Ova transakcija se rollbackuje i na ovaj način rešavamo konflikt.



PharmacistWorkCalendarController - save
PharmacistWorkCalendarService - save

R Uspešno čitanje iz baze

W Uspešno pisanje u bazu

W Neuspešno pisanje u bazu