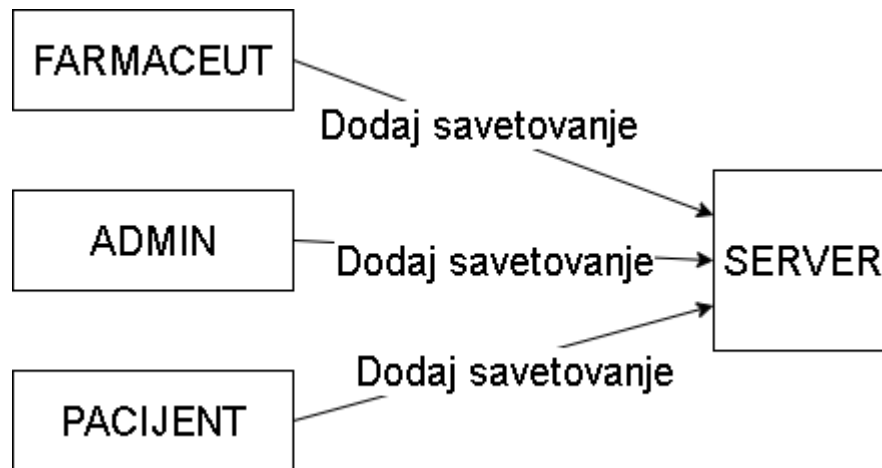


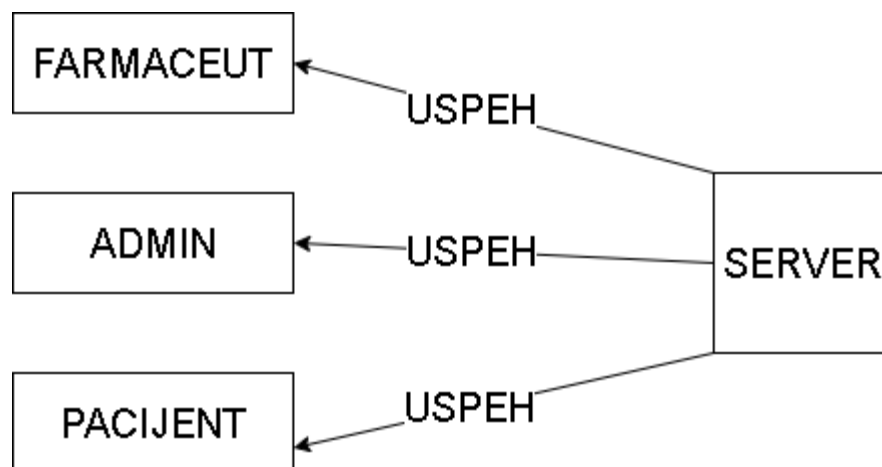
Student 3: Konflikti

Slučaj 1

Više korisnika (Farmaceut, Admin Apoteke i Pacijent) pokušavaju istovremeno da zakažu savetovanje u isto vreme kod istog Farmaceuta.



Server svima javlja da je dodavanje uspešno i pacijent se obaveštava više puta da je savetovanje dodato. Takođe savetovanja se preklapaju i javlja se višak podataka u bazi.



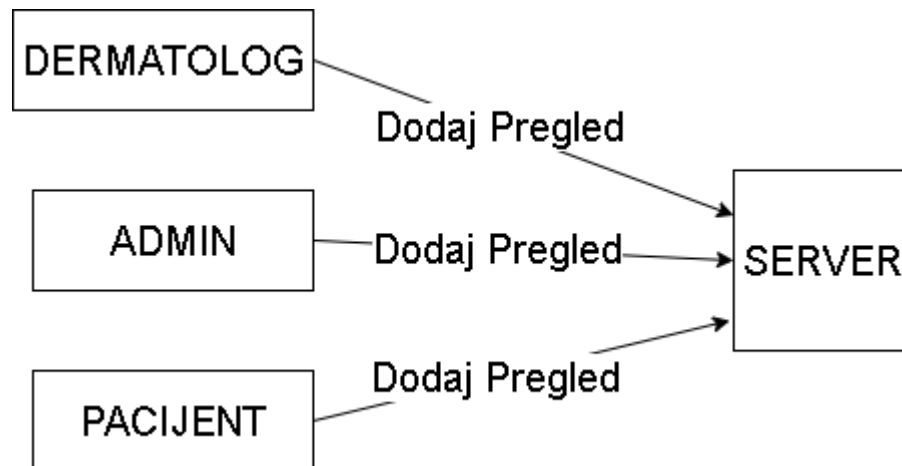
Pošto sva 3 korisnika gađaju isti endpoint("/savetovanje/dodaj/{id}") potrebno je uvesti metodu servisa koja će izvršiti transakciju u kojoj će se izvršiti neophodno zaključavanje, a zatim provere i dodavanje željenog termina.

Metoda koja se koristi ima naziv `dodajSavetovanje` i nalazi se u klasi `SavetovanjeService`. U ovoj metodi pesimističkim zaključavanjem pokušavamo da rešimo problem koji se javlja. Pre svega zaključavaju se aktivna Savetovanja i Odmori koje Farmaceut ima u unetom vremenskom rasponu. Međutim, često se javlja da ne postoji ni jedno ovakvo Savetovanje ili Odsustvo zbog čega je potrebno uvesti dodatna zaključavanja. Problem rešavamo tako što zaključavamo samog Farmaceutu kom se

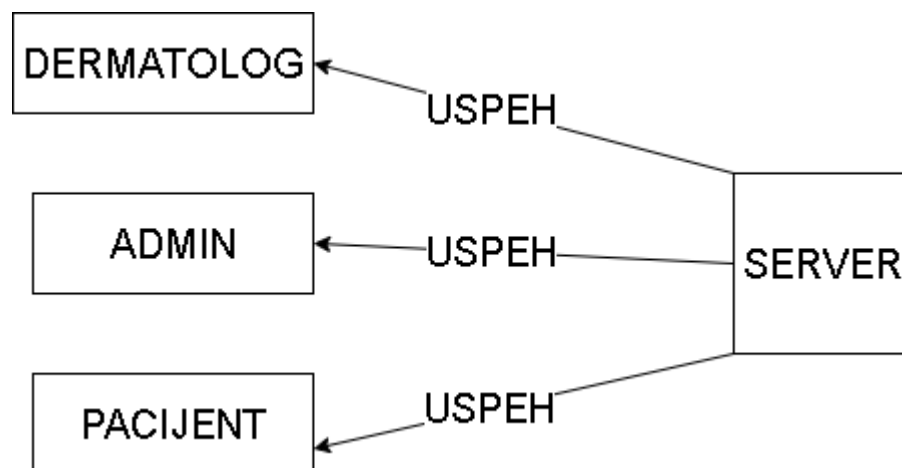
dodaje Savetovanje. Na ovaj način onemogućavamo da se u određenom vremenskom periodu doda više Savetovanja.

Slučaj 2

Više korisnika (Dermatolog, Admin Apoteke i Pacijent) pokušavaju istovremeno da zakažu pregled u isto vreme kod istog Dermatologa.



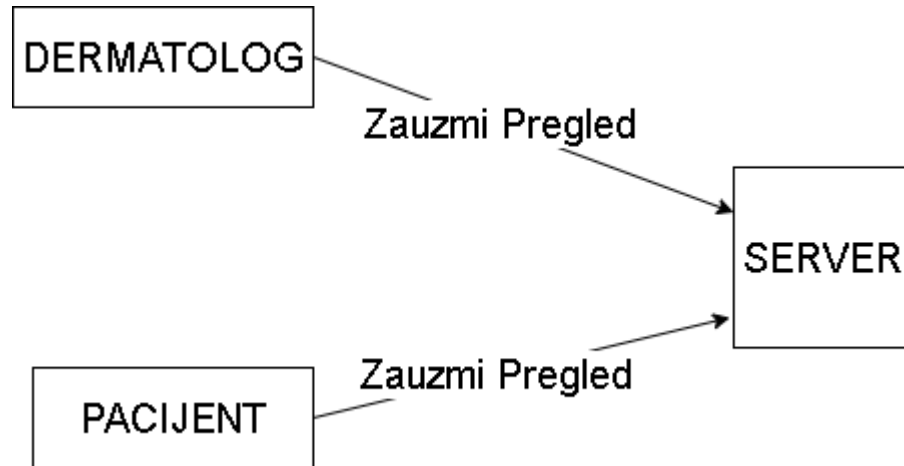
Server svima javlja da je dodavanje uspešno i pacijent se obaveštava više puta da je pregled dodat. Takođe pregledi se preklapaju i javlja se višak podataka u bazi.



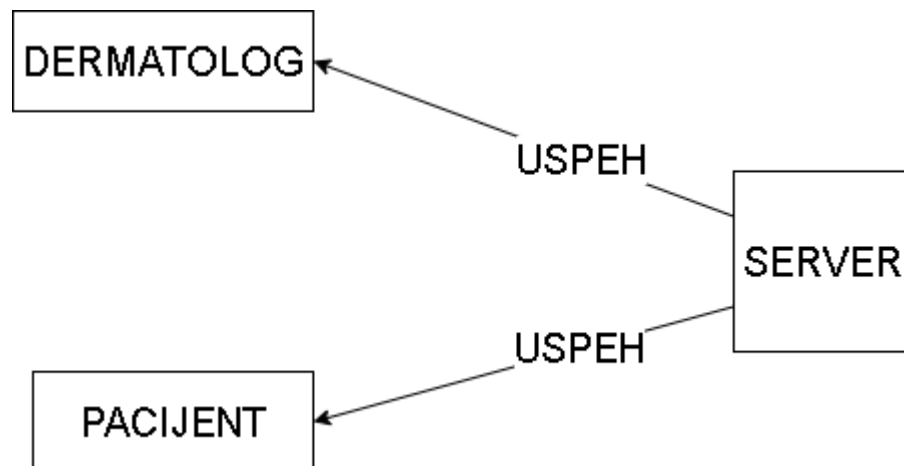
Rešenje ovog problema je slično rešenju slučaja 1 osim što sada sve metode pogađaju endpoint `"/pregled/dodaj/{id}"` i `"/pregled/dodajSlobodan/{id}"` i vrši se zaključavanje Dermatologa. Metoda koja vrši transakciju se naziva `dodajPregled` i nalazi se u `PregledService` klasi.

Slučaj 3

Pacijent i Dermatolog na kraju pregleda pokušavaju istovremeno da zauzmu unapred definisan slobodan pregled.



Server u oba slučaja javlja da je zauzimanje uspešno i pacijent se obaveštava više puta da je pregled zauzet.

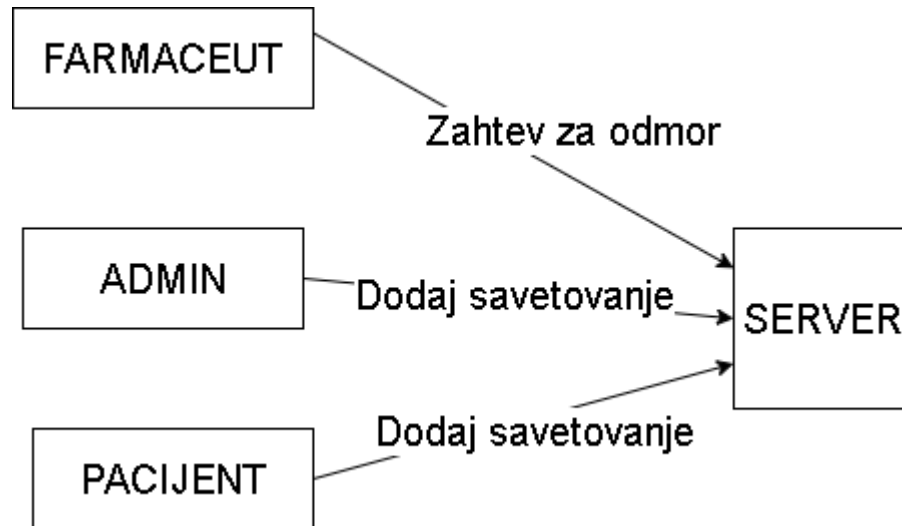


Za rešenje ovog problema se koristi optimističko zaključavanje. Endpoint koji se pogađa je `"/pregled/zauzmi/{id}/{pacijentId}"`.

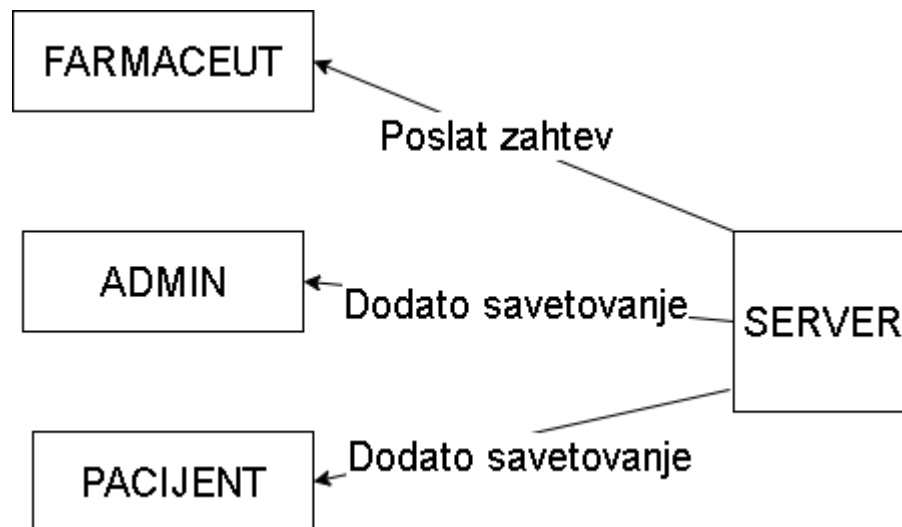
Razlog zbog kog se koristi optimističko zaključavanje je to što pregledi već postoje u bazi i potrebno je samo izvršiti update. Optimističko zaključavanje se vrši tako što se u roditeljsku klasu Pregleda, Termin dodaje atribut `version`. Kada se pokuša izmena pregleda ukoliko se verzije ne poklapaju dolazi do greške, a ako se poklapaju izmena je uspešna.

Slučaj 4

Farmaceut kreira zahtev za odmor za koji je zahtev da nema zakazanih savetovanja u tom periodu. Istovremeno Pacijent ili Admin zakazuju savetovanje u periodu odmora.



Rezultat je da se kreira zahtev za odmor i ako postoje savetovanja.

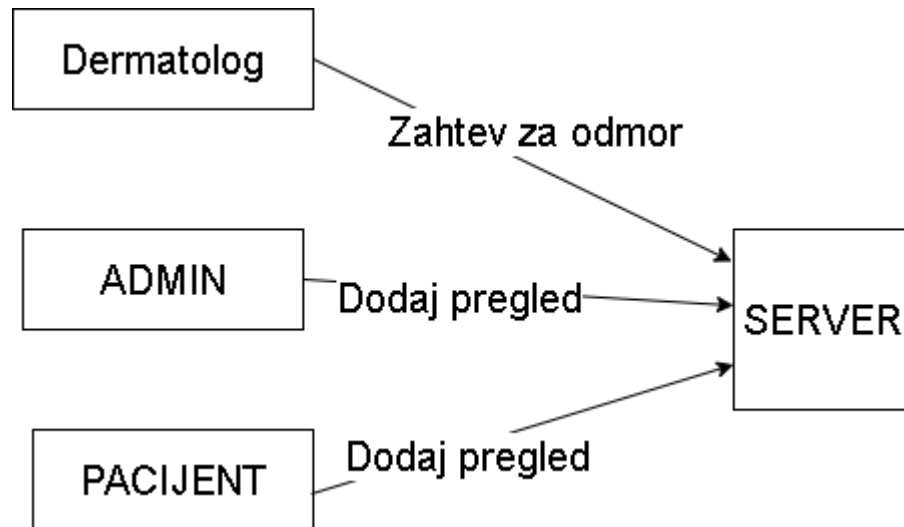


Pošto Admin i Pacijent gađaju isti endpoint ("/savetovanje/dodaj/{id}"), a Farmaceut drugi endpoint ("/odsustvo/farmaceut/dodaj/{id}") potrebno je zaključati neki zajednički resurs pomoću pesimističkog zaključavanja.

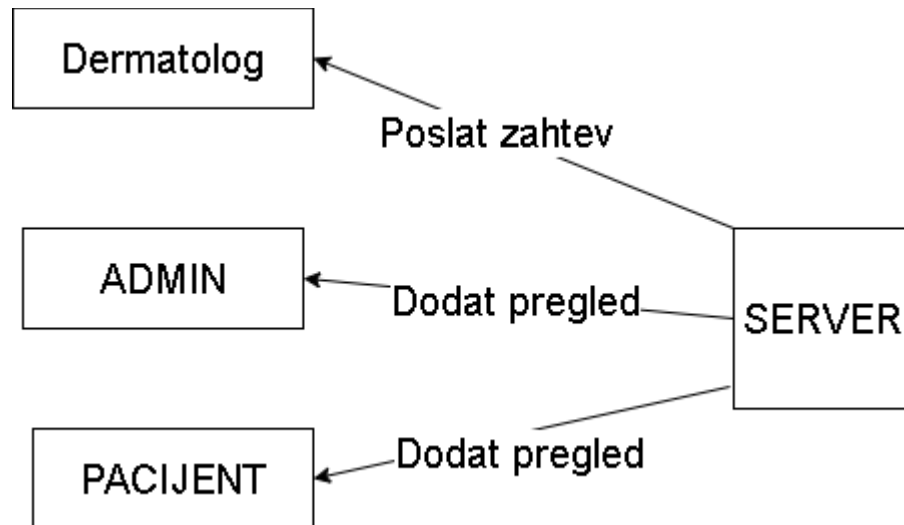
U metodi dodajOdsustvo klase OdsustvoFarmaceutService vrši se zaključavanje Farmaceuta, isto kao i u metodi dodajSavetovanje koja se nalazi u klasi SavetovanjeService. Na ovaj način samo korisnik čiji se zahtev prvi obradio će uspeti da obavi željenu akciju, što će osigurati da ograničenja aplikaciju budu zadovoljena.

Slučaj 5

Dermatolog kreira zahtev za odmor za koji je zahtev da nema zakazanih savetovanja u tom periodu. Istovremeno Pacijent ili Admin zakazuju pregled u periodu odmora.



Rezultat je da se kreira zahtev za odmor i ako postoje pregledi.



Pošto Admin i Pacijent gađaju isti endpoint (`"/pregled/dodaj/{id}"`), a Dermatolog drugi endpoint (`"/odsustvo/dermatolog/dodaj/{id}"`) potrebno je zaključati neki zajednički resurs pomoću pesimističkog zaključavanja.

U metodi `dodajOdsustvo` klase `OdsustvoDermatologService` vrši se zaključavanje Dermatologa, isto kao i u metodi `dodajPregled` koja se nalazi u klasi `PregledService`. Na ovaj način samo korisnik čiji se zahtev prvi obradio će uspeti da obavi željenu akciju, što će osigurati da ograničenja aplikaciju budu zadovoljena.