

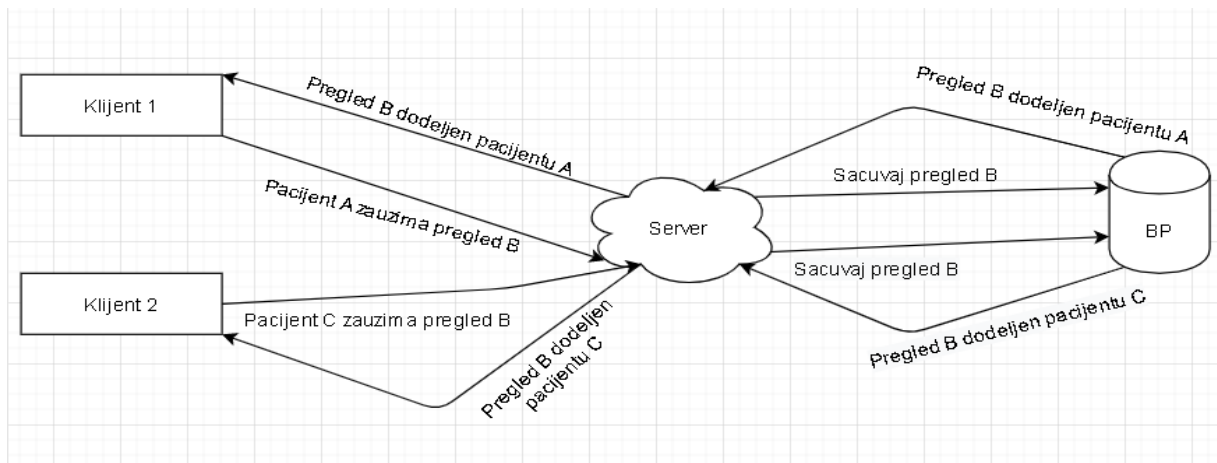
Ime i prezime: Jelena Cupać

Broj indeksa: SW13-2017

Zadatak: Rešavanje konfliktnih situacija za studenta 1

## 1. Konkurentno zakazivanje predefinisanih pregleda od strane više različitih pacijenata

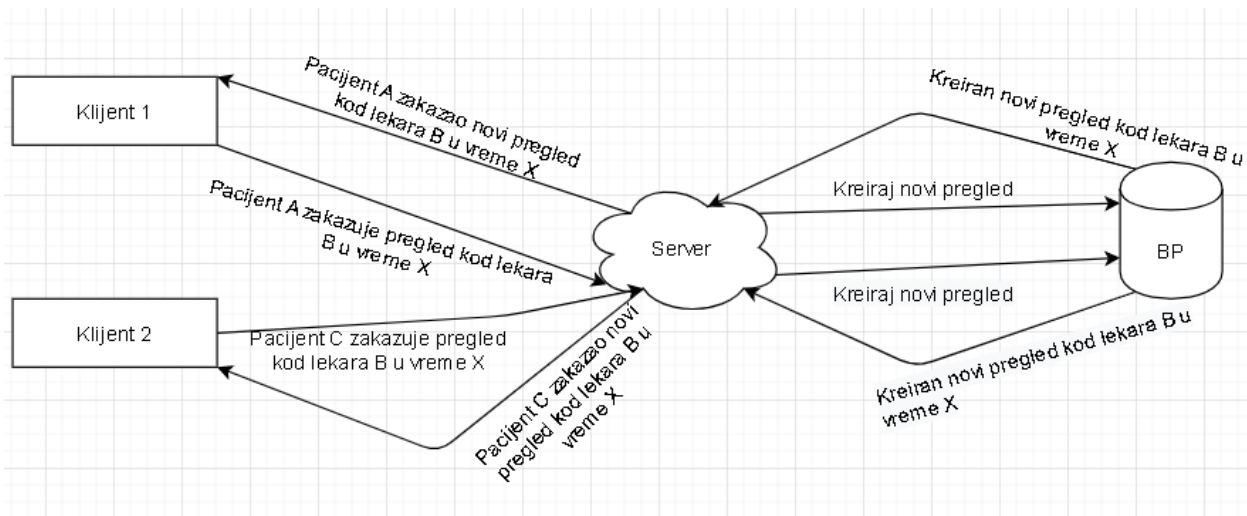
**Problem:** Admini klinike mogu unapred da kreiraju termine za pregled i to su predefinisani termini. Predefinisani termini imaju lekara i tip pregleda za koji se pregled izvršava, te salu i satnicu. Pacijent može da zatraži novi termin ili da rezerviše jedan od već postojećih predefinisanih pregleda. Problem nastaje kada više pacijenata istovremeno pokuša da zauzme isti predefinisani pregled. U tom slučaju će prvi pacijent da zauzme pregled, a zatim drugi pacijent da „pregazi“ to i on će zauzeti pregled.



**Rešenje:** Želeli bismo na neki način da „zaključamo“ pregled tako da samo jedan pacijent može da ga menja, tj. da ga zauzme. Način na koji sam rešila ovaj problem jeste upotrebom optimističkog zaključavanja. Optimističko zaključavanje zahteva uvođenje novog atributa sa anotacijom *@Version* u tabeli *Pregled*. Na ovaj način, kada jedna transakcija izmeni predefinisani pregled, odnosno, kada tom pregledu dodeli pacijenta, vrednost atributa anotiranog sa *@Version* se uvećava za jedan. Da bi transakcija bila uspešna vrednosti atributa anotiranog sa *@Version* u transakciji i u bazi moraju biti jednake. Nakon što se prva transakcija završi i pregled bude zauzet, sve ostale transakcije kada pokušaju da učine isto, dobiće izuzetak, jer se vrednosti atributa anotiranog sa *@Version* u transakciji i u bazi ne podudaraju. Na ovaj način je rešen problem konkurentnog zakazivanja predefinisanih pregleda od strane više različitih pacijenata.

## 2. Konkurentno kreiranje pregleda za isti termin kod istog lekara od strane pacijenata

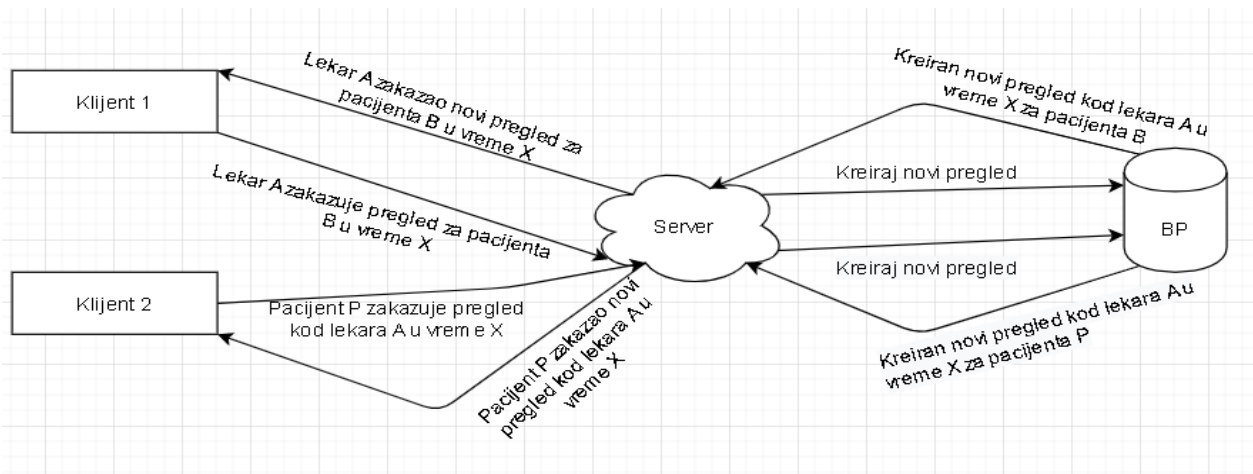
**Problem:** Pacijenti imaju mogućnost da zakažu pregled u određenom terminu kod određenog lekara. Pacijent prvo pretragom izabere kliniku, a zatim, takođe, pretragom, izabere lekara kod kojeg želi da obavi pregled. Nakon odabira lekara, pacijentu se prikazuje lista slobodnih termina. Pacijent bira željeni termin, a onda mu se na novoj stranici prikazuju sve informacije o odabranom pregledu. Na toj stranici pacijent može da pošalje zahtev za pregledom u odabranom terminu kod odabranog lekara ili da poništi slanje tog zahteva. Problem nastaje u situaciji kada više pacijenata pošalje zahtev za pregledom u istom terminu kod istog lekara. Na taj način će se kreirati dva ili više zahteva za pregledom, za koje dalje admin klinike dodeljuje salu. Na ovaj način sistem dolazi u nekonzistentno stanje, jer će jedan isti lekar biti prisutan istovremeno na dva ili više različitih pregleda.



**Rešenje:** Da bismo rešili ovaj problem, na neki način je potrebno zaključati odabranog lekara. Međutim, ako pokušamo sa optimističkim zaključavanjem kao u prethodno objašnjenom problemu, samo sa dodavanjem atributa anotiranog sa *@Version* nećemo uspeti da rešimo ovaj problem. Problem je što kada kreiramo objekat klase *Pegled* i tom objektu dodelimo lekara, a zatim taj objekat sačuvamo u bazu, objekat lekar ostaje nepromenjen te se vrednost atributa anotiranog sa *@Version* neće uvećati te će biti i dalje kreirano više zahteva za pregledom u istom terminu kod istog lekara. Da bismo rešili ovaj problem treba na neki način da izvršimo izmenu u objektu klase *Lekar*. Za potrebe ovog problema, u tabeli *Lekar* uveden je novi atribut *lastChange*. Sad kada dodelimo lekara pregledu izmeni ćemo atribut *lastChange* i sačuvati promene lekara. Na ovaj način kada prva transakcija uradi *commit*, obzirom da je objekat klase *Lekar* izmenjen, ažuriraće se atribut anotiran sa *@Version*. Svaka sledeća transakcija kada pokuša da uradi *commit* dobiće izuzetak, jer se vrednosti atributa anotiranog sa *@Version* kod lekara u bazi i u tim transakcijama neće podudarati. Na ovaj način, rešen je problem kreiranja više zahteva za pregledom u istom terminu kod istog lekara.

### 3. Konkurentno zakazivanje pregleda od strane lekara i pacijenta

**Problem:** Lekar ima mogućnost da za pacijenta, za kojeg trenutno vrši pregled zakaže novi termin. Istovremeno, neki drugi pacijent ima mogućnost da kreira zahtev za pregledom u istom terminu kod tog lekara. Na ovaj način, sistem ponovo dolazi u nekonzistentno stanje jer će jedan isti lekar imati dva pregleda u istom terminu, a to nije nešto što bismo želeli da se desi.

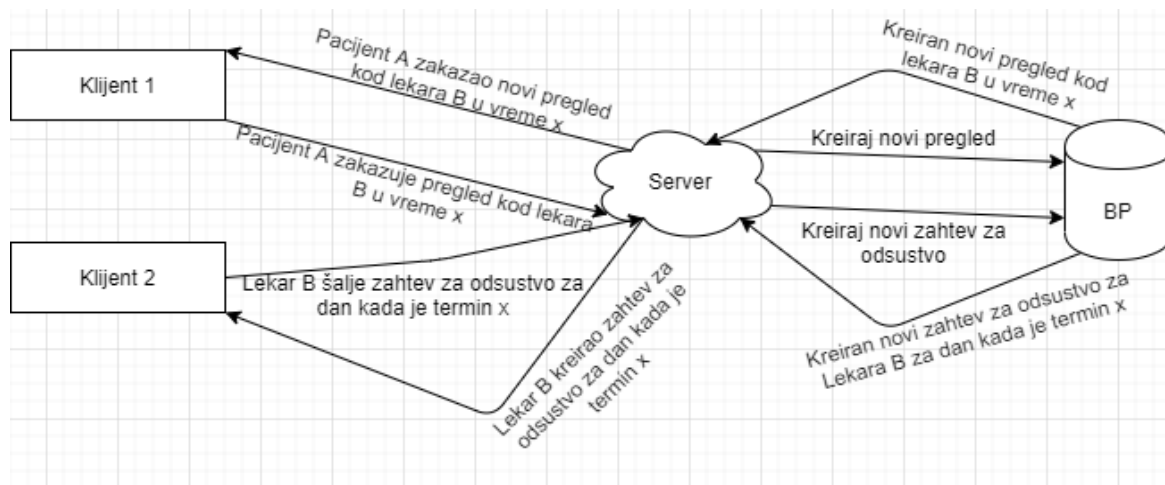


**Rešenje:** Ovaj problem je već rešen pri rešavanju problema konkurentnog kreiranja pregleda za isti termin kod istog lekara od strane više pacijenata. Ovde je takođe potrebno da se na neki način zaključa lekar kod kojeg se vrši pregled, a to se ponovo postiže ažuriranjem atributa *lastChange* u tabeli *Lekar*. Na ovaj način, ako se transakcija kreiranja novog pregleda od strane lekara izvrši prva, pacijent koji je pokušao da zakaže kod istog tog lekara u istom terminu pregled dobiće izuzetak jer se ponovo vrednosti atributa anotiranog sa *@Version* u toj transakciji i u bazi neće podudarati.

Pored ovog problema, rešavanjem problema konkurentnog kreiranja pregleda za isti termin kod istog lekara od strane više pacijenata, rešen je i problem konkurentnog kreiranja predefinisanih pregleda od strane admina i kreiranja zahteva za pregledom u istom terminu kod istog lekara od strane pacijenta (ili lekara), kao i kreiranje predefinisanih pregleda od strane dva admina za istog lekara i isti termin. Iz tog razloga taj problem neće biti objašnjavaan.

#### 4. Konkurentno zakazivanje pregleda od strane pacijenta i slanje zahteva za odsustvo od strane lekara

**Problem:** Lekar ima mogućnost da pošalje zahtev za odsustvo adminu klinike, dok pacijent može da zakazuje slobodne termine lekara za pregled. Problem nastaje kada pacijent šalje zahtev za pregled, a istovremeno lekar kreira zahtev za odsustvo za isti dan kada je i taj pregled.



**Rešenje:** Ovaj problem je rešen optimističkim zaključavanjem i ažuriranjem atributa *lastChange* u tabeli Lekar. Rešenje je analogno rešenju problema 3. Na ovaj način ako se transakcija kreiranja novog pregleda od strane pacijenta izvrši prva, lekar koji je pokušao da pošalje zahtev za odsustvo će dobiti izuzetak jer se vrednosti atributa anotiranog sa *@Version* u toj transakciji i u bazi neće podudarati i biće obavešten da je u tom periodu zauzet.