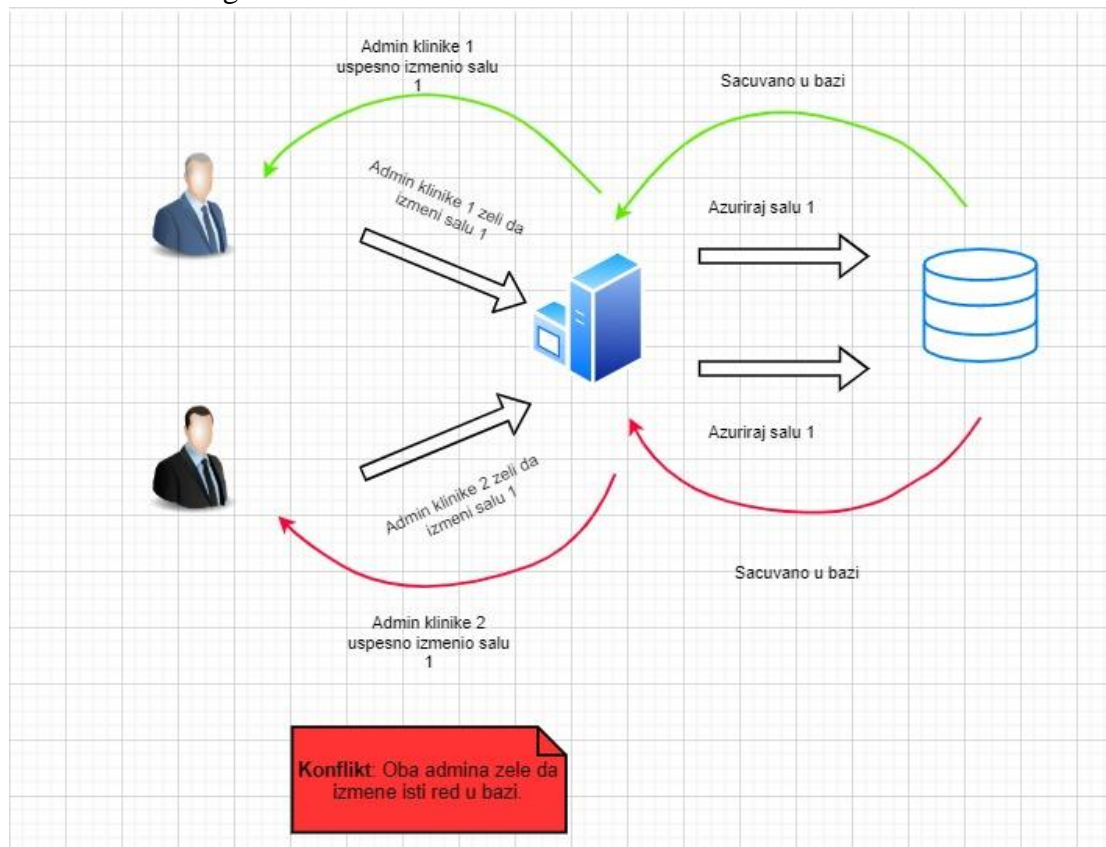


Konfliktne situacije Student 2

Konflikt 1: Ulogovani su admin klinike 1 i admin klinike 2. I jedan i drugi žele da u isto vreme izmene podatke o sali. Dolazi do *update* konflikta koji nastaje kada dva različita admina u približno isto vreme menjaju isti red u istoj tabeli baze podataka.

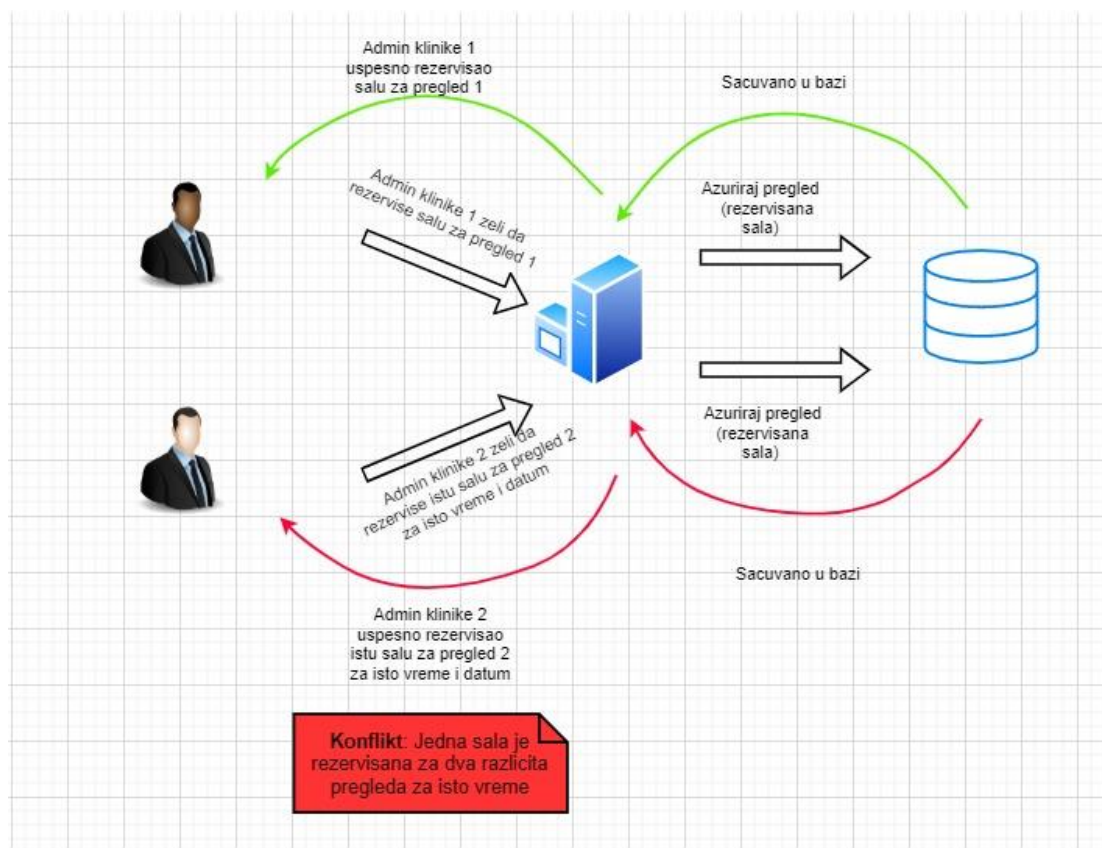
Tok zahteva i odgovora:



Rešenje: Koristim optimističko zaključavanje, tako što klasi *Sala* dodam atribut *version*, koji je anotiran sa *@Version* i omogućava *Hibernate*-u da pri svakoj izmeni inkrementira njegovu vrednost (verziju), ukoliko je ona ista kao i na početku transakcije. U slučaju da verzija nije ista kao na početku transakcije baca se izuzetak. Optimističko zaključavanje nije isključivo, što znači da neće uticati na performanse sistema. U *SalaService* klasi dodala sam *@Transactional* iznad klase i odgovarajuće metode (*save*), kako bi se ta metoda mogla ponašati kao transakcija. Ovim smo omogućili da imamo kontrolisano ažuriranje sale.

Konflikt 2: Ulogovani admin klinike 1 i admin klinike 2 žele istovremeno da rezervišu salu 1 za dva različita pregleda. Nakon što jedan od njih rezerviše salu, drugom se ne ažurira stranica nego i taj drugi admin može da rezerviše salu 1, za isti termin.

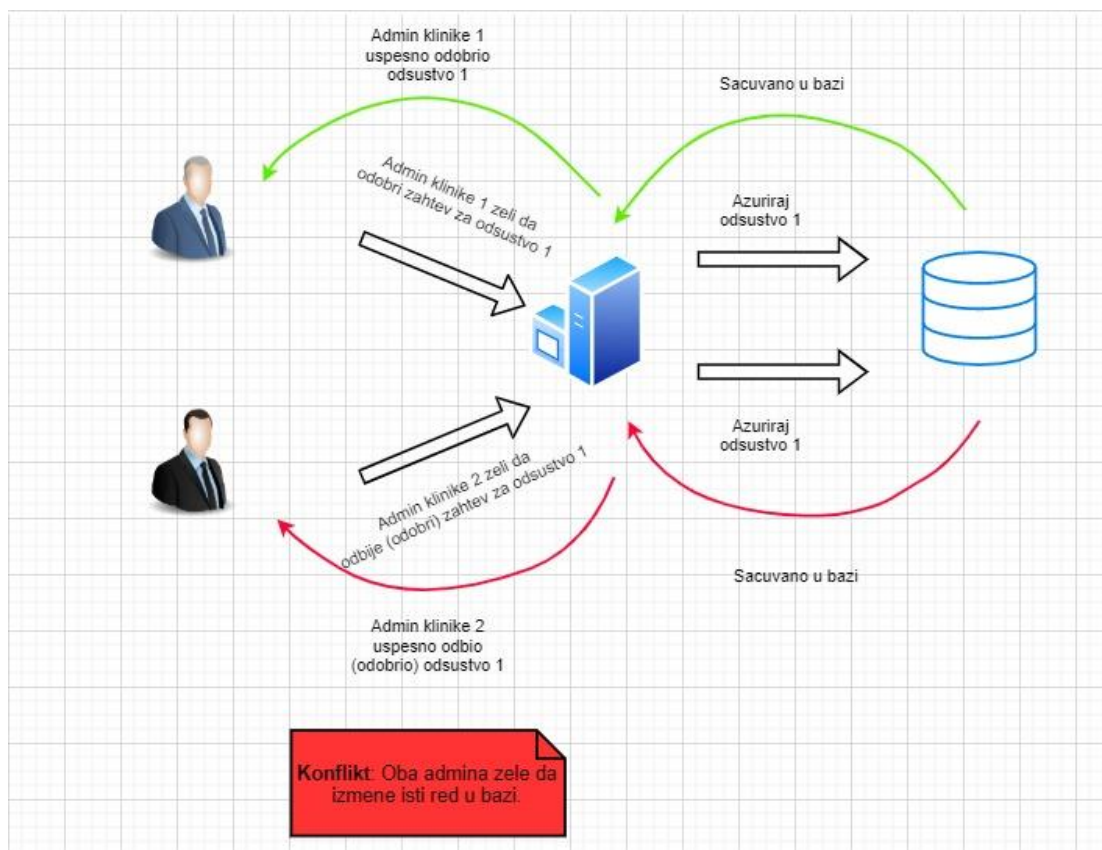
Tok zahteva i odgovora:



Rešenje: Koristim optimističko zaključavanje, tako što klasi *Sala* dodam atribut *version*, koji je anotiran sa *@Version* i omogućava *Hibernate*-u da pri svakoj izmeni inkrementira njegovu vrednost (verziju), ukoliko je ona ista kao i na početku transakcije. U slučaju da verzija nije ista kao na početku transakcije baca se izuzetak. U *SalaService* klasi dodala sam *@Transactional* iznad klase i odgovarajuće metode (*save*). Kako u tabeli *pregled* imamo kolonu *sala_id*, potrebno je prilikom rezervisanja sale za neki pregled napraviti promenu i u samoj sali, kako bi se trigerovalo *Hibernate*-ovo verzionisanje. U klasi *Sala* iz ovog razloga dodajemo atribut *int izmena*, čiju vrednost inkrementiramo kada se sala rezerviše. Ovim smo onemogućili da se ista sala rezerviše za dva različita pregleda u istom terminu.

Konflikt 3: Ulogovani admin klinike 1 i admin klinike 2 istovremeno odobravaju/odbijaju zahtev za odmor/odsustvo. Nakon što jedan od njih odobri/odbije zahtev, drugom se ne ažurira stranica nego i on može da odobri/odbije isti zahtev.

Tok zahteva i odgovora:



Rešenje: Koristim optimističko zaključavanje, tako što klasi *Odsustvo* dodajemo atribut *version*, koji je anotiran sa *@Version* i omogućava *Hibernate*-u da pri svakoj izmeni inkrementira njegovu vrednost (verziju), ukoliko je ona ista kao i na početku transakcije. U slučaju da verzija nije ista kao na početku transakcije baca se izuzetak. U *OdsustvoService* klasi dodala sam *@Transactional* iznad klase i odgovarajuće metode (*save*), kako bi se ta metoda mogla ponašati kao transakcija. Kada se zahtev za odsustvom odobri ili odbije, odgovarajućem objektu klase *Odsustvo* se setuje atribut *status* i zatim se upiše u bazu. Pri ovoj promeni se automatski poveća i *version* atribut. Ovim smo omogućili da imamo kontrolisano odobravanje ili odbijanje zahteva za odmorom/odsustvom i onemogućili da isti zahtev bude u isto vreme ažuriran od strane dva različita admina klinike.