# Конфликтне ситуације – Студент 2

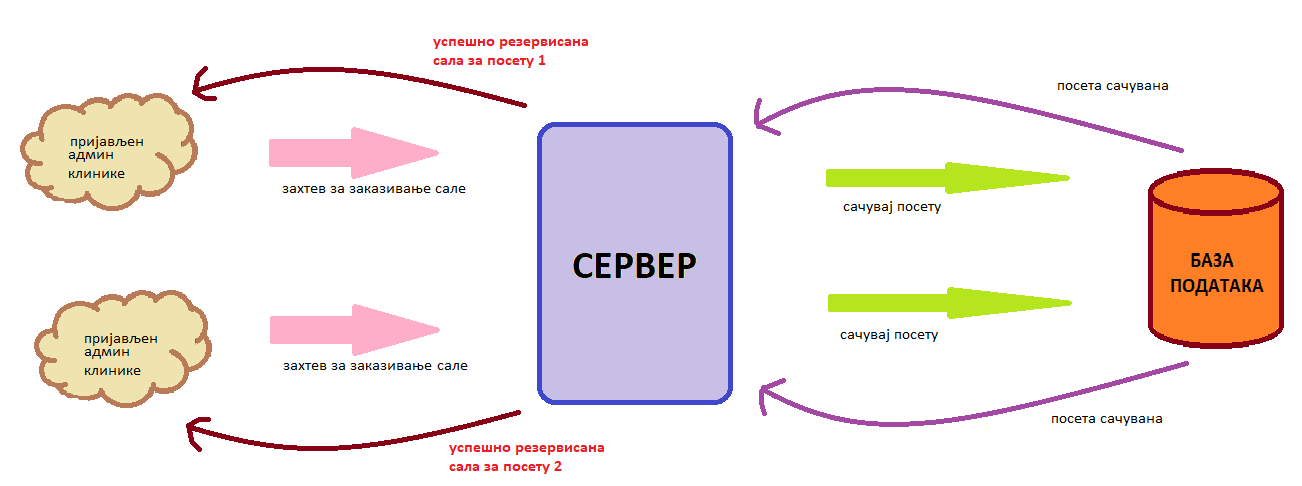
## Конкурентно заказивање сале

Сценарио заказивања сале:

Резервацију сале за преглед/операцију обавља админ клинике, који том приликом обрађује пристигли захтев за посету. Приликом заказивања посете админ бира слободну салу и покушава резервацију сале у жељеном термину. Приликом резервације сале захтев за посету постаје посета која је повезана са лекаром, пацијентом (преко картона) и салом. Посета је креирана и сачувана у бази само уколико постоје слободна сала и лекар/и за жељени термин. Уколико админ успешно резервише посету, сала, лекар/и и пацијент у листи посета виде новозаказану посету.

Проблеми и решење конкуретног заказивања сале за жељену посету

Проблем настаје када два админа (исте) клинике конкуретно резервишу салу за различите захтеве али у истом термину(Слика 1). До овакве ситуације долази када админ приликом резервације претражује салу по другом датуму. Односно креирање захтева у истом термину програмски је онемогућено, али претрагом сале за други датум два админа за различите захтеве могу добити исту салу као резултат претраге. Ово нарушава конзистентност базе у смислу да би сала била резервисана у невалидном термину. Како је решење резервације имплементирано тако да се приликом креирања посете проверава заузетост сале, решење ове конфликтне ситуације своди се на оптимистичко закључавање класa *Sala*. Kласy треба проширити пољем *long version* које је анотирано са *@Version*. На овај начин *Entity Manager* ће за нас упоредити верзије базе података и тренутно жељене трансакције, тако да резервација сале у истом термину није могућа и вратиће нам *OptimisticLockException* изузетак. Међутим, како обрада захтева за резервацију сале подразумева успешно креирање и чување објекта класе *Poseta* који има референцу ка сали*,* том приликом потребно је утицати на атрибут *version* класе *Sala*. Ово је могуће уколико класу *Sala* проширимо *Date poslednjaIzmena* атрибутом и *setPoslednjaIzmena(Date date)* методом.



Слика 1 Конфликтна ситуација кокуретног заказивања сале за исту посете

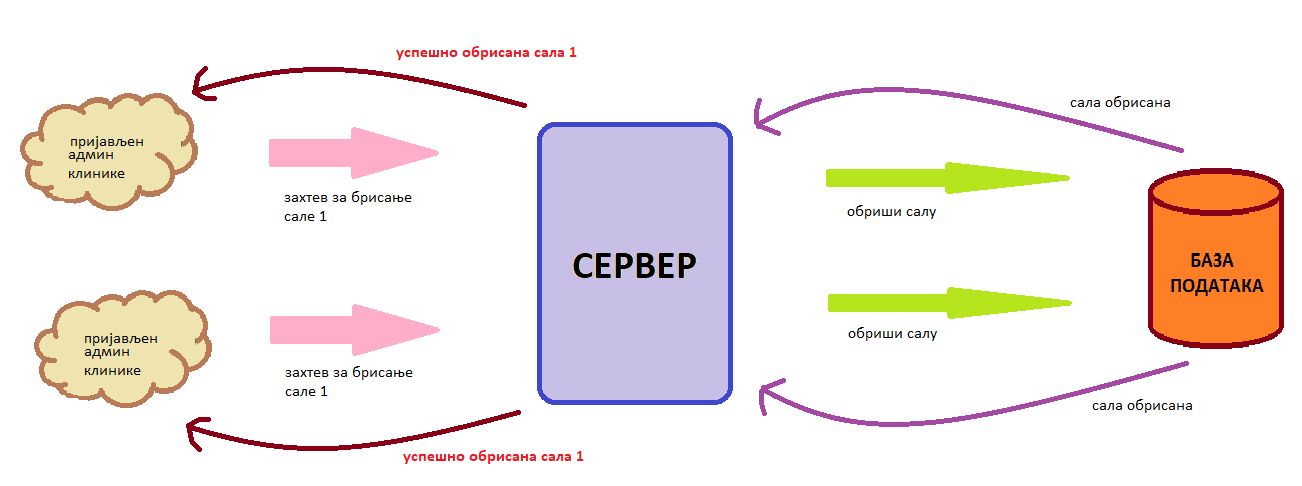
## Конкуретно брисање(измена) исте сале

Сценарио брисања/измене сале:

Администратор приликом претраге сала поред сваке сале има опције за брисање и изменз. Уколико сала нема заказаних прегледа/операција кликом на дугме *Obrisi* админ логички брише салу из базе (постављање атрибута *aktivan* на *false*), у супротном корисник је обавештен о разлогу немогућности операције. Слично је и када кликне на дугме *Izmeni*, уколико сала није резервисана админ је у могућности да мења назив сале, уколико је сала већ резервисана админ добија обавештење о немогућности овакве операције.

Проблем и решење конкуретног брисања/измене

Уколико минимум два пријављена администратора клинике у исто време покушају да обришу/измене салу коју је могуће обрисати/изменити настаје конфликтна ситуација (Слика 2) која би базу довела у неконзистентно стање и стога је овако нешто недопустиво. Како је логичко брисање заправо измена сале по атрибуту *aktivan*, решење овакве конфликтне ситуације добија се увођењем оптимистичког закључавања класе *Sala*. Kласy треба проширити пољем *long version* које је анотирано са *@Version*. На овај начин биће омогућена само трансакција која се првобитно изврши, док ће други корисник бити обавештен о измени прве трансакције и неће му бити дозвољена иста трансакција. На исти начин решен је и конфликт приликом измене назива сале.



Слика 2 Конфликтна ситуација конкуретног брисања сале

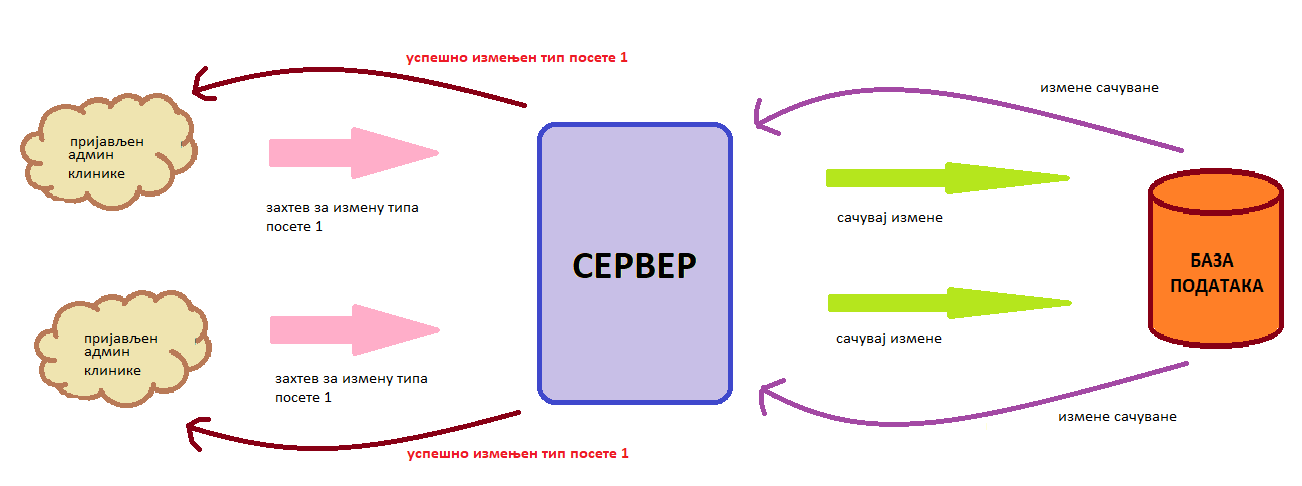
## Конкуретно брисање(измена) типа посете

Сценарио брисања/измене типа посета

Администратор клинике има листу типова посета, и за сваку посету има опцију обриши и измени. Кликом на дугме *Obrisi*, уколико не постоји заказана посета (преглед/операција) тог типа , тип посете се логички брише из базе, у супротном админ је обавештен о немогућности операције. Иста ситуација је и код измене типа посете, уколико не постоји заказана посета која је тог типа измена је могућа у суптотном не.

Проблем и решење конкуретног брисања/измене

Уколико минимум два пријављена администратора клинике у исто време покушају да обришу/измене тип посете коју је могуће обрисати/изменити настаје конфликтна ситуација(Слика 3) која би базу довела у неконзистентно стање и стога је овако нешто недопустиво. Како је логичко брисање заправо измена типа посете по атрибуту *aktivan*, решење овакве конфликтне ситуације добија се увођењем оптимистичког закључавања класе *TipPosete*. Kласy треба проширити пољем *long version* које је анотирано са *@Version*. На овај начин биће омогућена само трансакција која се првобитно изврши, док ће други корисник бити обавештен о измени прве трансакције и неће му бити дозвољена иста трансакција.

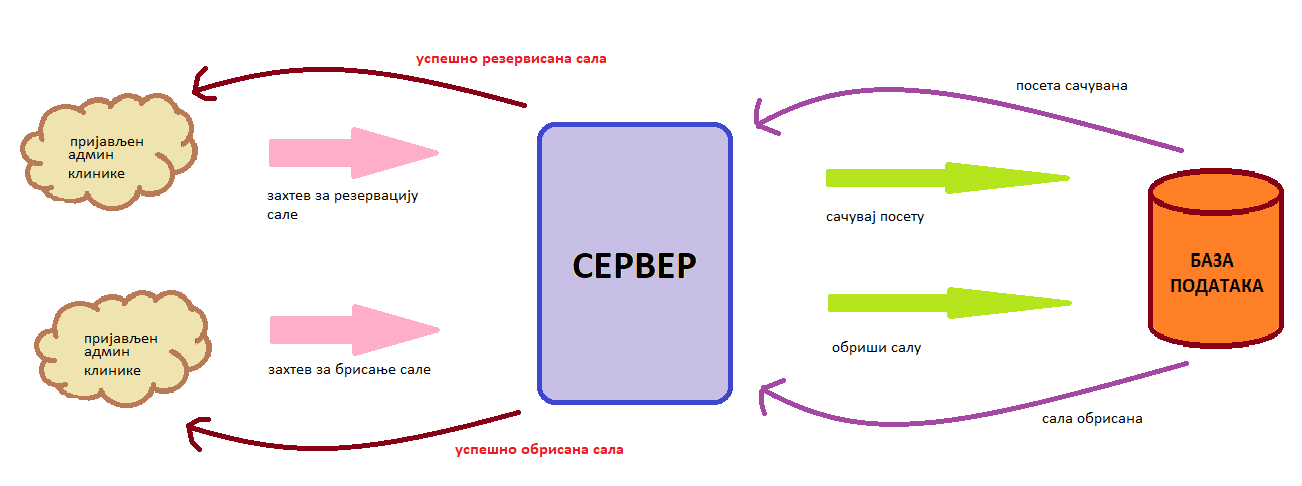


Слика 3 Конфликтна ситуација конкуретне измене типа посете

## Конкурентно брисање и резервација сале

Сценарии брисања и резервације сале обрађени су у претходним конфликтним случајевима.

Конкуретно брисање и резервација сале(Слика *4*), смер брисање па резервација(прво је завршила трансакција брисање) решено је проширењем класе *Sala* атрибутом *long version,* смер резервација па брисање треба приликом креирања посете позвати методу сале *setPoslednjaIzmena(Date date)* којом се утиче на атрибут *long version* класе *Sala.*

**

Слика 4 Конфликтна ситуација конкуретна резервација и брисање сале

## Конкурентно брисање типа посете и креирање захтева за посету

Сценарио брисања типа посете

Админ клинике има увид у све типове посете своје клинике. Поред сваког типа посете има опције измени и обриши. Ове опције је могуће извршити само уколико не постоји заказана посета типа који админ жели да мења, у супротном админ добија грешку.

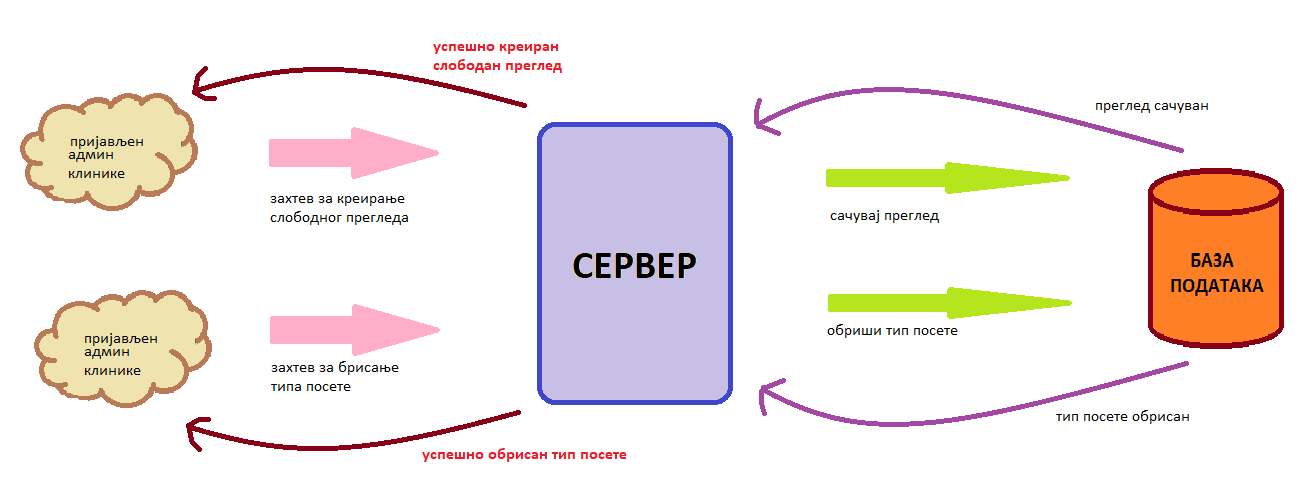
Сценарио креирања захтева за посету (слободан преглед)

Админ клинике има опцију да креира слободан преглед тако што за преглед дефинише датум и време, тип посете, салу и лекара.

Из приложеног се јасно уочава проблем конкуретног брисања типа посете и креирања слободног термина прегледа, јер је за креирање слободног прегледа неопходно имати тип посете.

Решења конфликтне ситуације

Конкуретно брисање типа посете и креирање захтева за посету(Слика 5), смер брисање типа посете па креирање захтева за посету решено је проширењем класе *TipPosete* атрибутом *long version,* смер креирање захтева за посету па брисање типа посете: потребно је приликом креирања захтева за посету позвати методу типа посете *setPoslednjaIzmena(Date date)* којом се утиче на атрибут *long version* класе *TipPosete.*



Слика 5 Конфликтна ситуација креирања слободног термина и брисања типа посете

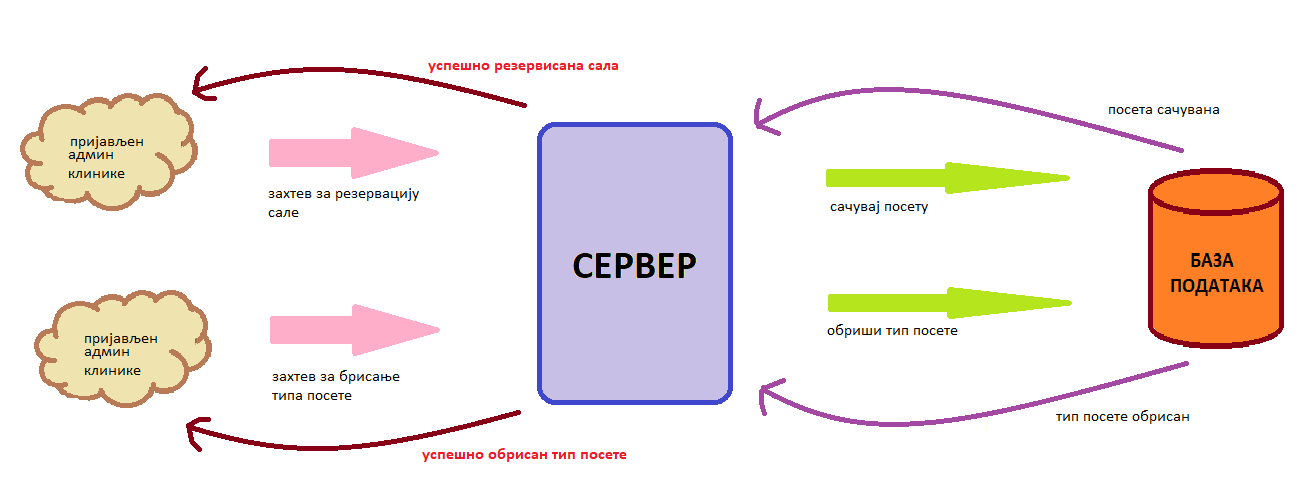
## Конкурентно брисање типа посете и резервација сале

Сценарии брисања типа посете и резервације сале обрађени су раније у документу.

Проблем настаје из разлога што посета за коју се резервише сала мора имати специфициран тип посете, а кокуретно брисање типа посете и резервација сале доводе базу у неконзистентно стање.

Решења конфликтне ситуације

Конкуретно брисање типа посете и резервација сале(Слика 6), смер брисање посете па резервација сале решено је проширењем класе *TipPosete* атрибутом *long version,* смер резервација сале па брисање типа посете: потребно је приликом креирања посете (код резервације сале) позвати методу типа посете *setPoslednjaIzmena(Date date)* којом се утиче на атрибут *long version* класе *TipPosete.*



Слика 6 Конфликтна ситуација конкурентне резервације сале и брисања типа посете