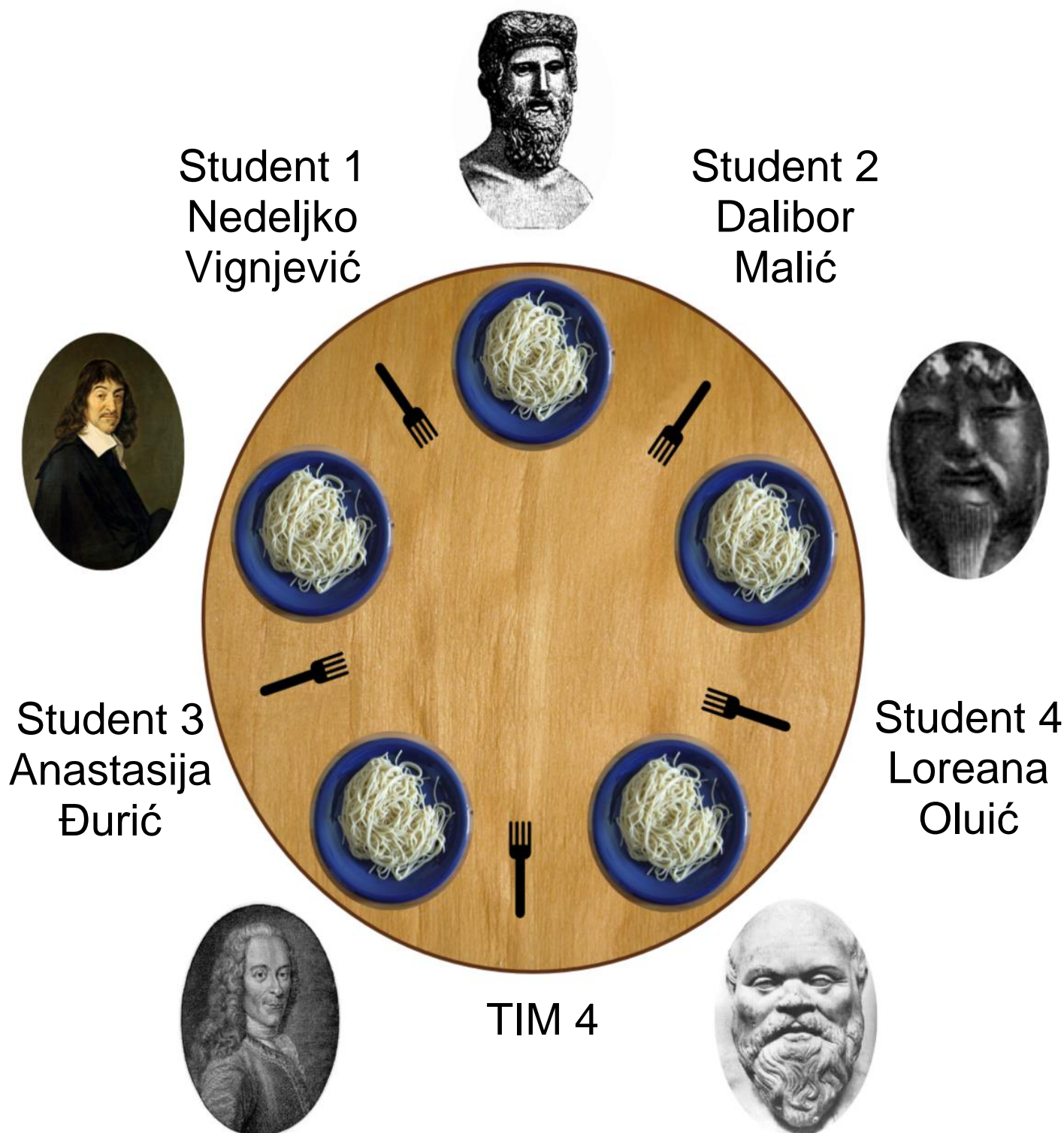


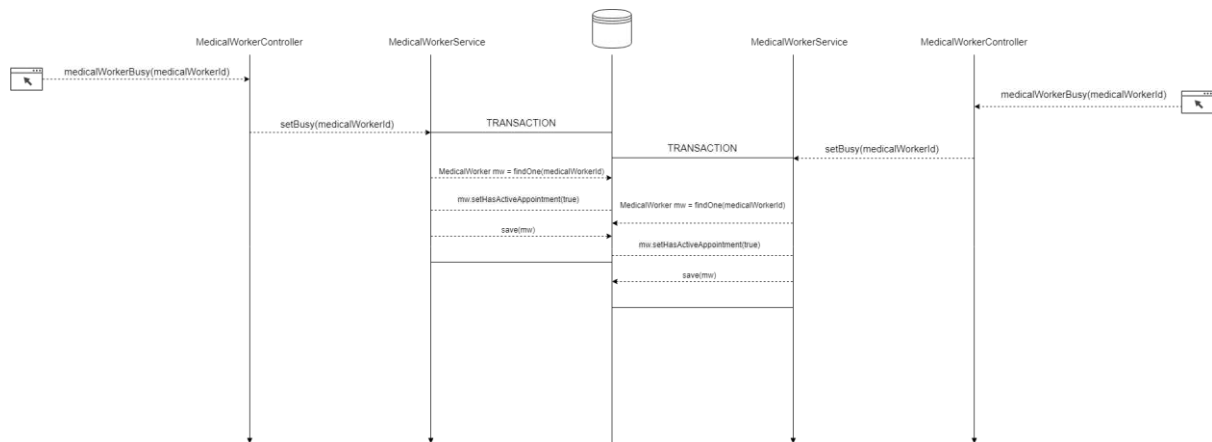
Transakcije i konkurentni pristup



1. Istovremeno prisustvo farmaceuta/dermatologa na pregledima (#TUS17)

Opis problema koji se rešava:

Farmaceut/dermatolog ulogovan je na svoj nalog sa dva različita računara/pretraživača. Medicinski radnik je započeo pregled sa jednog računara. Potrebno je onemogućiti da isti medicinski radnik započne isti ili neki drugi pregled sa drugog računara dok postoji pregled koji je u toku (Slika 1).



Slika 1. Istovremena izmena hasActiveAppointment svojstva

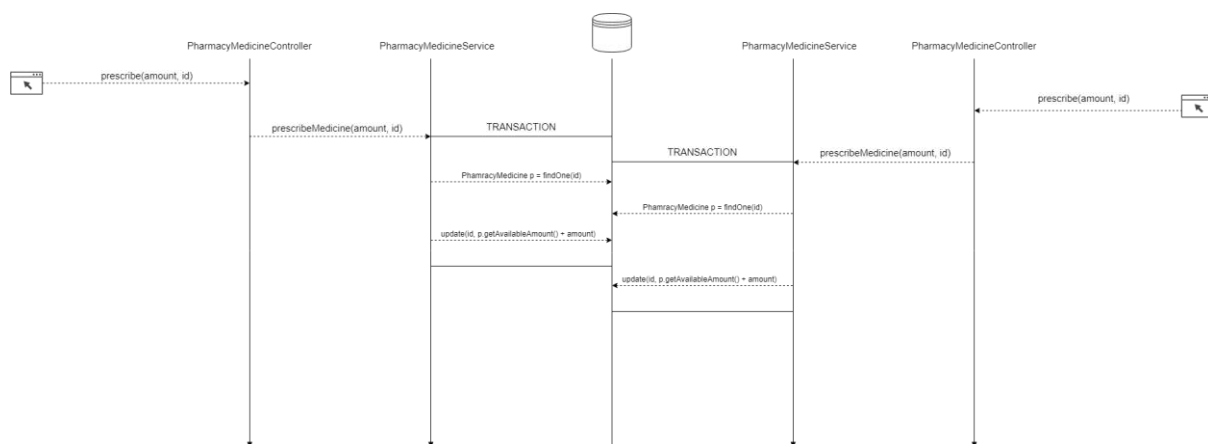
Opis rešenja:

Za rešavanje ovog problema korišćeno je optimističko zaključavanje s obzirom na to da se menja samo jedno svojstvo - *hasActiveAppointment* već postojećeg *MedicalWorker* entiteta. Na Slici 1 prikazan je slučaj u kom se poziva servisna metoda *setBusy* koja postavlja vrednost pomenutog svojstva na *true*. Na ovaj način obezbeđeno je da u određenom trenutku vrednost datog svojstva može biti promenjena sa *false* na *true* samo jednom, dok će ostali pokušaji izmene izazvati *OptimisticLockingFailure* izuzetak prilikom čije obrade će se korisniku poslati odgovarajuća poruka o neuspehoj akciji. Analogno važi za servisnu metodu *setFree* koja postavlja vrednost svojstva *hasActiveAppointment* na *false* i koja se poziva kada je pregled završen ili prekinut. U okviru klase *User* dodato je *version* svojstvo.

2. Izmena količine leka u apoteci prilikom dodavanja ili brisanja leka sa liste preporučenih lekova u toku pisanja izveštaja (#TUS17)

Opis problema koji se rešava:

Dva medicinska radnika istovremeno popunjavanju izveštaj o pregledu u istoj apoteci. Svaki od njih preporučuje jedan po jedan lek pacijentu prilikom čega se količina datog leka smanjuje u toj apoteci. Takođe, postoji i mogućnost uklanjanja leka sa liste preporučenih, prilikom čega se količina uklonjenog leka poveća na stanju u datoj apoteci. Primer problema: započete su dve transakcije za isti *PharmacyMedicine* objekat. U oba slučaja iz baze je preuzet objekat čija je vrednost količine 5. U prvoj transakciji količina se uvećava za 2 i vrednost 7 se upisuje u bazu. U drugoj transakciji vrednost se uvećava za 1 nakon čega se vrednost 6 upisuje u bazu što će pregaziti vrednost 7 (pogledati primer na Slici 2).



Slika 2. Ažuriranje vrednosti availableAmount

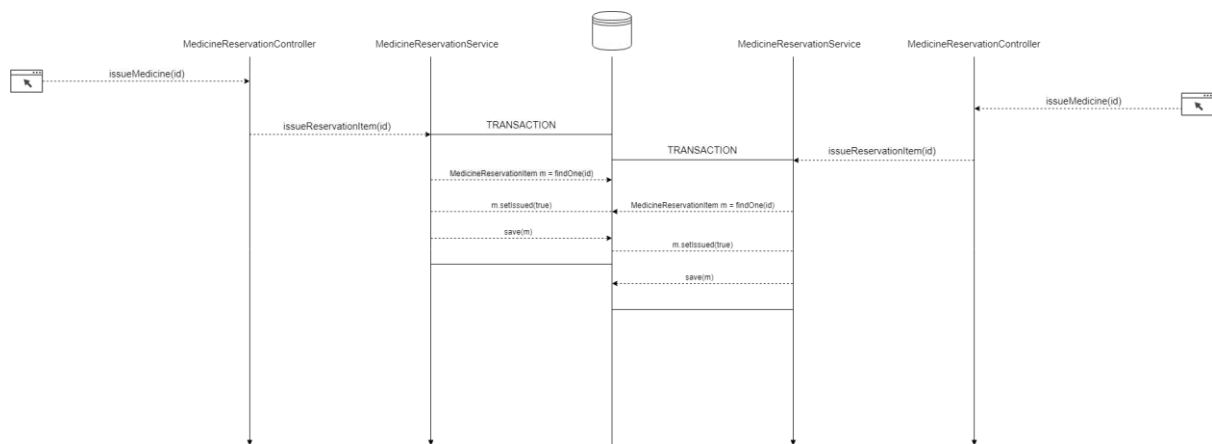
Opis rešenja:

Za rešavanje ovog problema korišćeno je optimističko zaključavanje s obzirom na to da se menja samo jedno svojstvo - *availableAmount* već postojećeg *PharmacyMedicine* entiteta. Na Slici 2 prikazan je slučaj u kom se poziva servisna metoda *prescribeMedicine* koja smanjuje količinu izabranog leka u datoj apoteci. U okviru date metode preuzima se odgovarajući *PharmacyMedicine* objekat. Ukoliko je verzija preuzetog objekta zastarela, korisniku će biti vraćena odgovarajuća poruka i akcija će biti obustavljena. Analogno važi za servisnu metodu *removeMedicine* unutar koje se smanjuje vrednost svojstva *availableAmount*. U okviru klase *PharmacyMedicine* dodato je *version* svojstvo.

3. Izdavanje rezervisanih lekova od strane farmaceuta (#TUS17)

Opis problema koji se rešava:

Dva farmaceuta zaposlena u istoj apoteci (ili jedan isti farmaceut ulogovan sa dva računara) unose jedinstveni broj iste rezervacije i biraju opciju *Confirm* za prvi izlistani lek. Potrebno je obezbediti da mejl o potvrdi preuzimanja leka pacijentu stigne smao jednom. Primer problema prikazan je na Slici 3.



Slika 3. Postavljanje svojstva Issued na true dva puta

Opis rešenja:

Za rešavanje ovog problema korišćeno je optimističko zaključavanje s obzirom na to da se menja samo jedno svojstvo - *issued* već postojećeg *MedicineReservationItem* entiteta. Na Slici 3 prikazan je slučaj u kom se poziva servisna metoda *issueReservationItem* koja postavlja vrednost svojstva *issued* na *true* za datu rezervaciju leka. Ukoliko je verzija preuzetog objekta rezervacije zastarela, korisniku će biti vraćena odgovarajuća poruka i akcija će biti obustavljena. U okviru klase *MedicineReservationItem* dodato je *version* svojstvo.