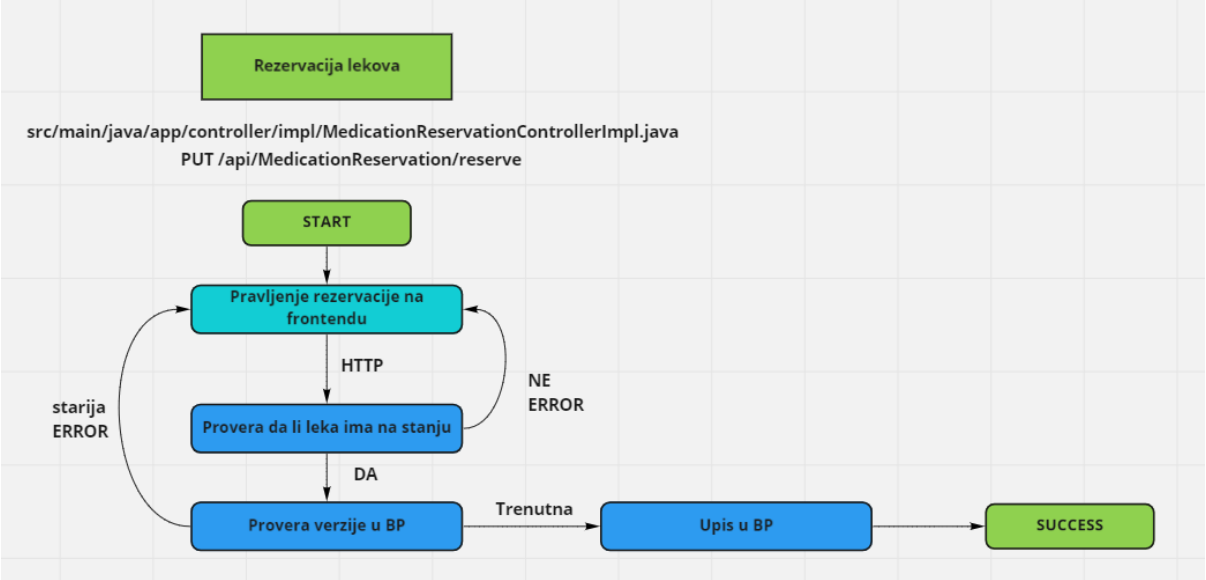
IZVEŠTAJ O TRANSAKCIJAMA

Student: Ilija Brdar RA101/2017 (Student 1)

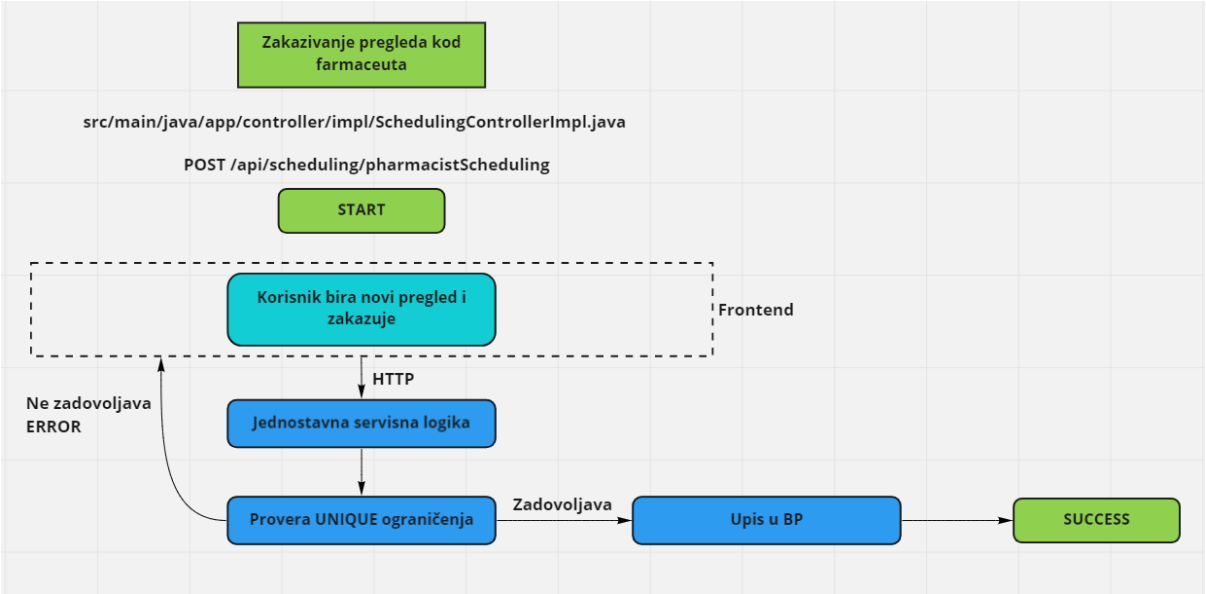
Funkcionalnosti pacijenta jesu da zakazje i oktazuje preglede, rezerviše lekove, otkazuje rezervacije, menja sopstveni profil i ocenjuje. Neke konfliktne situacije pri tim aktivnostima su:

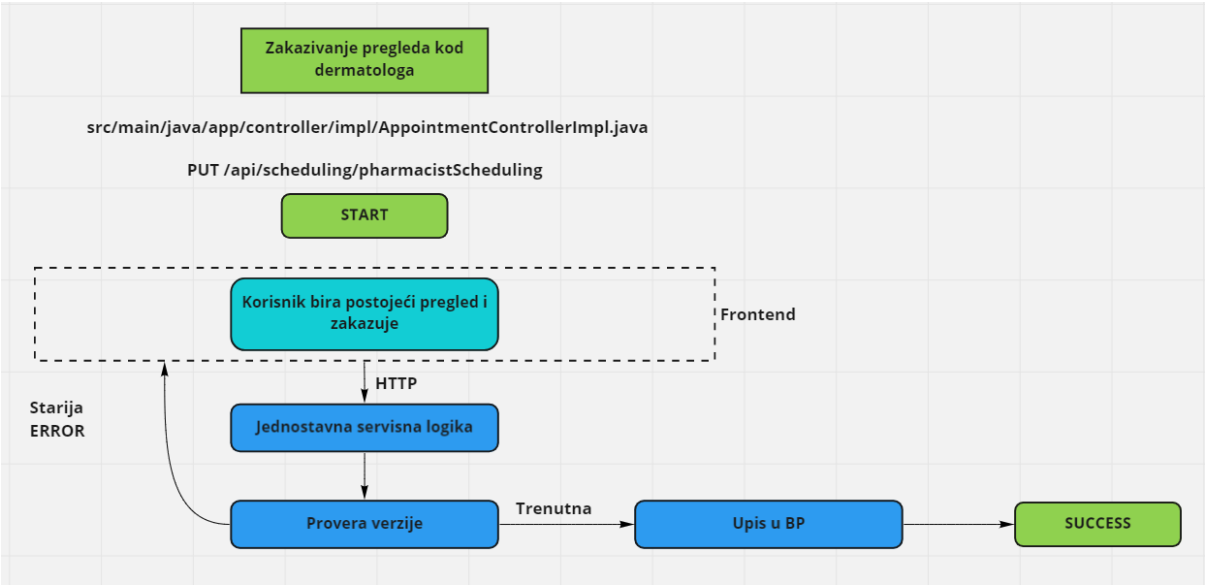
- više istovremenih korisnika aplikacije ne može da rezerviše lek koji je u međuvremenu postao nedostupan
  Opis situacije: Više korisnika aplikacije istovremeno definišu količinu leka koji žele da rezervišu, pod uslovom da se radi o istom leku u istoj apoteci, i istovremeno naprave rezervaciju. Tada se može desiti da prvi zahtev dođe do resursa i smanji količinu leka u apoteci, ali drugi zahtev će videti staru količinu leka, pa će i on izvršiti smanjivanje. Ovo je konfliktna situacija jer oba zahteva izvrše smanjivanje leka u apoteci za neki broj, iako ukupno nije bilo toliko leka na stanju.
  Rešenje: Za rešavanje odabrano je optimistično zaključavanje baze. Pre svakom pristupu proverava se verzija količine leka u apoteci. Ukoliko verzija nije najnovija, to znači da je neko prethodno promenio količinu leka (već je izvršena rezervacija ili otkazivanje) i korisnik nije u mogućnosti da ponovo smanji količinu. Konkretno u kodu, iznad metode za rezervaciju u klasi *MedicationReservationServiceImpl* je dodata anotacija *@Transactional* i uvedeno je verzionisanje klase *MedicationQuantity*. Napisan je i test koji proverava ovu situaciju.
  Grafik:



- količina leka na stanju se mora ispravno ažurirati nakon rezervacije leka od strane pacijenta, otkazivanja rezervacije leka
  Opis situacije: Slično kao prethodna situacija. Kada dva pacijenta rezervišu, ili otkažu lek, drugom po redu pacijentu ne treba dozvoliti da upiše novo stanje, jer on neće videti šta je pacijent pre njega uradio.
  Rešenje: Optimistično zaključavanje. Rešenje je isto kao i u prethodnoj situaciji.
  Grafik: isti kao u prethodnom primeru.

- Konfliktna situacija sa zakazivanjem pregleda – dva pacijenta ne mogu istovremeno da zakažu jedan termin
  Opis situacije: Do ove situacije može doći kada dva ili više pacijenata izaberu isti termin koji žele da zakažu i istovremeno izvrše zakazivanje. Klasa Appointment je u ovom slučaju deljeni resurs I izmene koje napravi drugi po redu pacijent nose opasnost da izbršu izmene (zakazivanje) prvog pacijenta. Ukoliko se radi o zakazivanju kod farmaceuta, to je problem upisa novog entiteta u bazu, a ukoliko se radi o zakazivanju kod dermatologa to je izmena postojećeg entiteta.
  Rešenje: Problem upisa novog entiteta je rešen uvođenjem unique constraint-a na nivou lekara, tipa pregleda (pregled ili savetovanje) i vremena pregleda (u klasi Appointment). To sprečava istovremeno kreiranje dva ista entiteta. Problem izmene termina (zakazivanje kod dermatologa) rešen je primenom optimističnog zaključavanja sa istom logikom kao i u prethodnim situacijama. Da bi to radilo, uvedeno je verzionisanje klase Appointment i stavljene su @Trasactional anotacije iznad metoda za zakazivanje savetovanja I pregleda.





Posebna uočena konfliktna situacija:

- konflikt pri davanju I brisanju korisničkih penala
  Opis situacije: Može da se desi, da farmaceut/dermatolog označe da se pacijent nije pojavio na pregledu i time mu daju jedan penal u isto vreme kad se pokrene metoda za brisanje penala (svakog prvog u mesecu). Tada može doći dao situacije da obe trasakcije vide isti početni broj penala, jedna ga poveća za jedan, a druga ga spusti na nulu. U zavisnoti od toga koja se desila prva, drugačiji će biti I krajnji rezultat. Takođe, budući da je klasa Patient deljena klasa, do problema može doći i pri izmeni profila pacijenta.
  Rešenje: Optimistično zaključavanje baze. Pre upisivanja u bazu, mora da se proveri verzija pacijenta. Ukoliko neka transakcija nema najnoviju verziju, ona ne može da piše u bazu (jer je neko pre nje promenio stanje). Konkretno u kodu, stavljene su anotacije @Transactional iznad metoda za brisanje penala u klasi PatientServiceImpl i metode za davanje penala ukoliko se pacijent nije pojavio na pregeldu u klasi AppointmentServiceImpl. Dodatno, stavljena je i @Transactional anotacija iznad save metode u klasi PatientServiceImpl radi resavanja problema izmene profila. U klasi Patient je uvedeno verzionisanje, jer je to klasa koja trpi promene.
  Grafik: