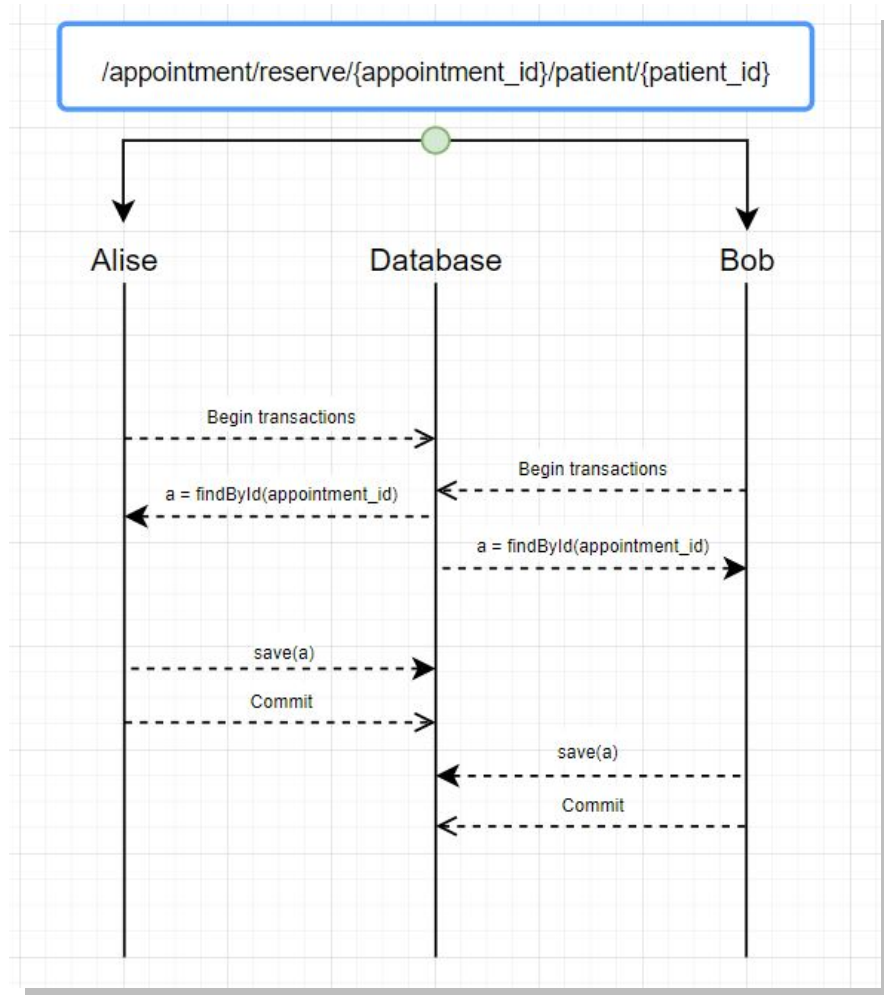


Svi endpoint-i počinju s <http://localhost:8000/api>.

Transakcije

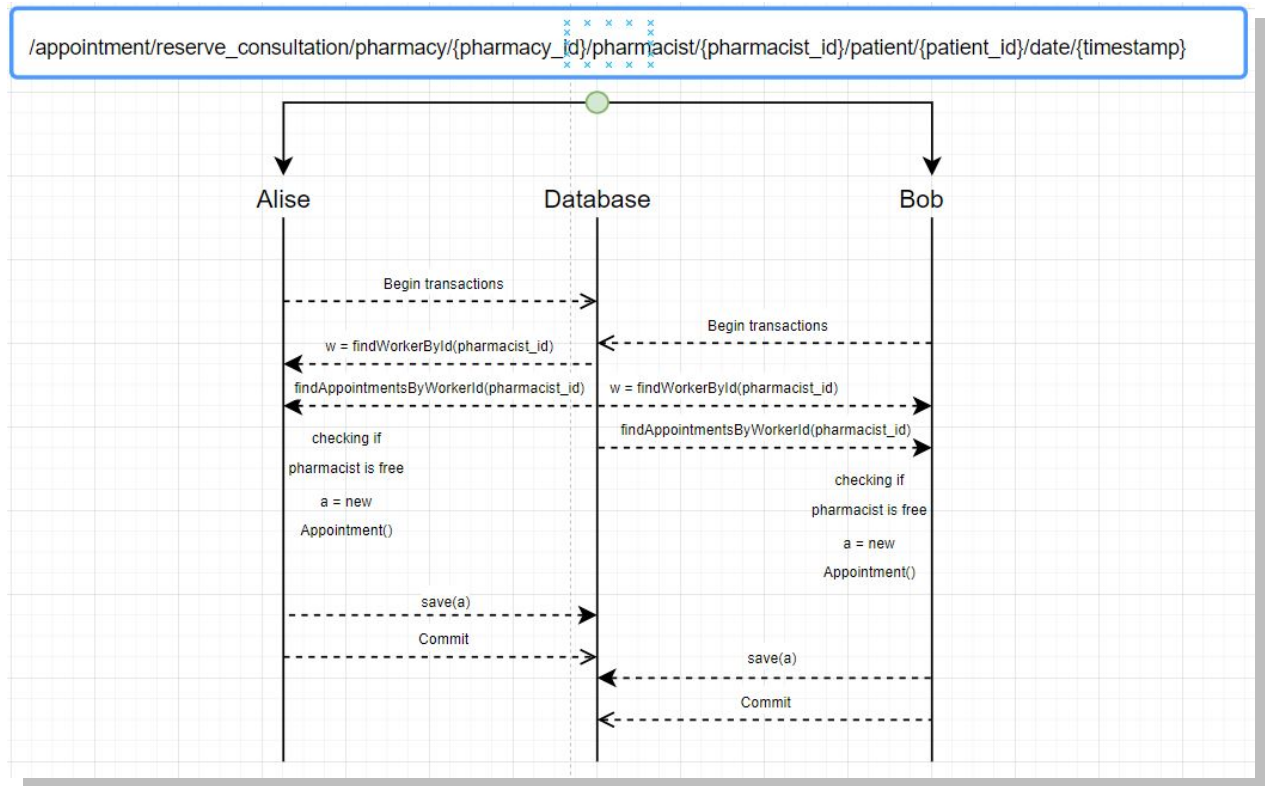
Konflikt kod zakazivanja pregleda kod dermatologa



- Ukoliko dva pacijenta pokušaju da rezervišu termin kod dermatologa, a da je pritom veoma mala vremenska razlika između te dve akcije, desiće se ovaj konflikt. Glavni problem kod ove situacije je taj što će Bobova transakcija “pregaziti” entitet koji je prethodno transakcija od Elis izmenila. Akcija zakazivanja termina kod dermatologa je veoma česta, i veoma je bitno da se što brže izvršava.
- Za rešavanje ovog konflikta sam izabrao optimističan pristup, iz razloga što će on rešiti ovaj problem u većini slučajeva. I dalje je moguća pojava Lost update-a, ali verovatnoća njegovog pojavljivanja je veoma mala i nije vredna korišćenja nekih striktnijih tehnika. Optimističan pristup sam implementirao pomoću verzija, koje porede verziju termina pri njegovom učitavanju sa onom pri kraju transakcije.

Svi endpoint-i počinju s <http://localhost:8000/api>.

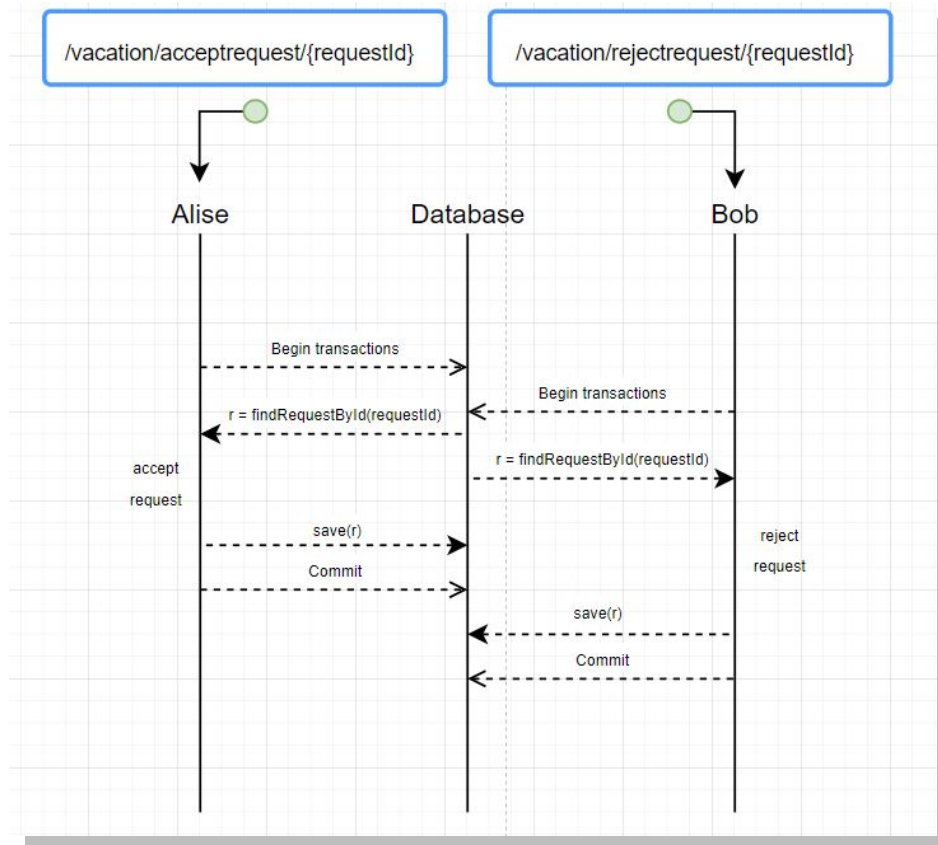
Konflikt kod zakazivanja savetovanja kod farmaceuta



- Ukoliko dva pacijenta pokušaju da rezervišu savetovanje kod farmaceuta, a da je pritom veoma mala vremenska razlika između te dve akcije, desiće se ovaj konflikt. Glavni problem kod ove situacije je taj što će oba pacijenta videti da je farmaceut u tom terminu slobodan. Iz tog razloga će obe transakcije kreirati svoj termin i dodati ga u bazu podataka. To znači da će biti kreirana dva objekta za isti termin, što nije ispravno i konzistentnost nase baze je narušena.
- Glavna karakteristika ovog problema je ta što se termin na kraju svih provera i validacija, kreira i dodaje u bazu. Iz tog razloga, ja nisam mogao da koristim optimistički pristup i da poredim verzije entiteta, jer taj entitet prethodno nije ni postojao. Iz tog razloga, prirodnije mi je bilo da zaključam termine koji predstavljaju deljeni resurs, i time omogućim da samo jedna transakcija može da se izvršava, dok će sve ostale čekati u redu. Pesimistički pristup je onaj koji sam odabrao. Zaključao sam metodu koja dobavlja termine farmaceuta, i samim tim, rešio problem konkurentne provere i validacije. Na ovaj način sam rešio problem višestrukog zakazivanja termina kod farmaceuta.

Svi endpoint-i počinju s <http://localhost:8000/api>.

Konflikt kod prihvatanja I odbijanja zahteva za odsustvom



- Ukoliko jedan admin apoteke pokuša da prihvati zahtev za odmorom, a drugi “istovremeno” (pod istovremeno smatram da je veoma mala vremenska razlika izmedju te dve akcije) pokuša da isti odbije, desiće se ovaj konflikt. Glavni problem kod ove situacije je taj što će se obe operacije izvršiti. Druga akcija će pregaziti prvu, iako se to ne bi smelo desiti.
- Uočio sam da korišćenjem optimističnog pristupa u ovoj situaciji, rešavamo problem u većini slučajeva. I dalje je moguća pojava Lost update-a, ali verovatnoća njegovog pojavljivanja je veoma mala i nije vredna korišćenja nekih striktnijih tehnika. Optimističan pristup sam implementirao pomoću verzija koje porede verziju zahteva pri njegovom učitavanju kao i pri kraju transakcije.