

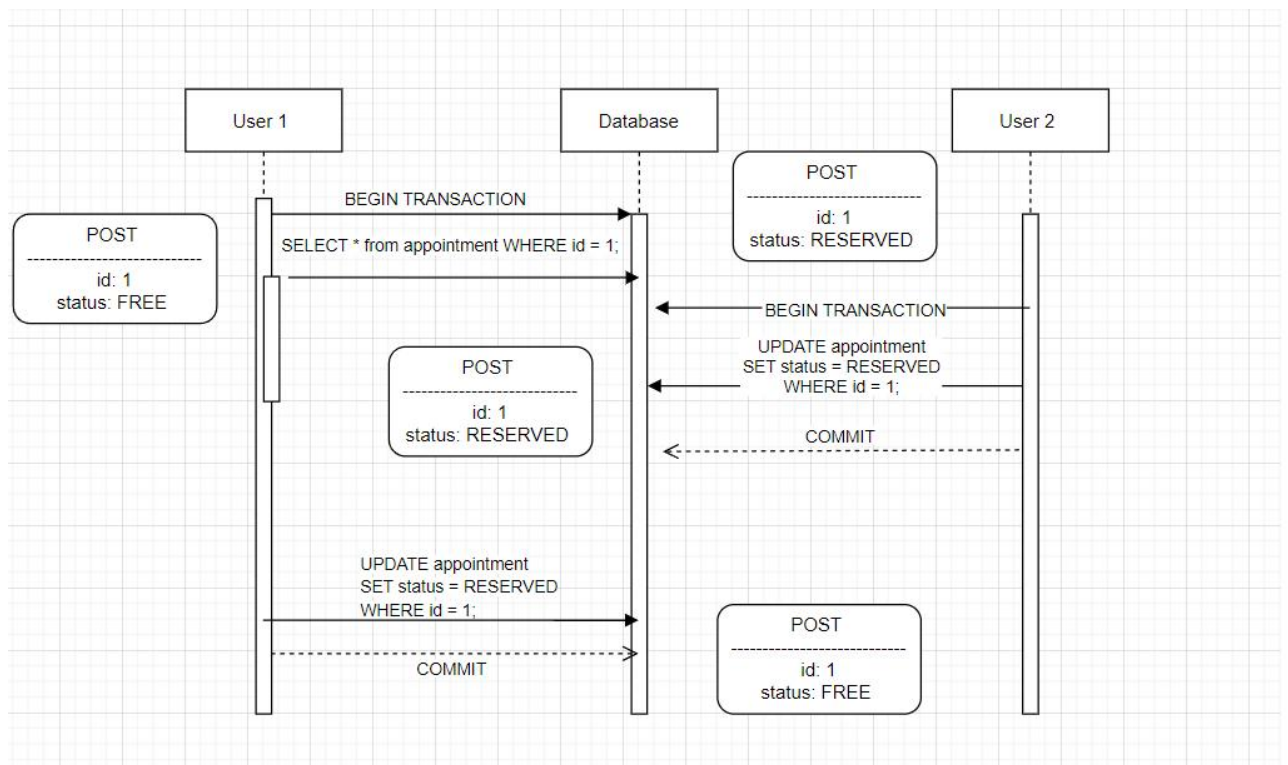
6.5. Konkurentni pristup resursima u bazi

- Konfliktne situacije:

1. više istovremenih korisnika aplikacije ne može da rezerviše termin preuzimanja koji je u međuvremenu postao nedostupan (u slučaju naše aplikacije, to se svodi na update-ovanje statusa appointmenta sa FREE na RESERVED)

2. termini koji su unapred definisani ne smeju biti rezervisani od strane više različitih korisnika - u našem projektu moguće je rešiti odjednom obe konfliktne situacije sledećim pristupom:

- Crtež toka aktivnosti koji dovodi do konflikta:



- Rešenje:

- Optimističko zaključavanje se koristi kako bi se rešila situacija kada dva administratora pokušavaju rezervisati isti termin (Appointment). Svaki termin ima polje status, koje administratori pokušavaju izmeniti. Optimističko zaključavanje koristi polje version, koje sam dodala u klasu Appointment.

- Implementacija u kodu:

Dva threada (Thread 1 i Thread 2) se pokreću paralelno, svaki pokušava rezervirati termin sa ID-om 1.

Prva nit (Thread 1) učitava termin sa ID-jem 1, menja status na RESERVED i čeka 3 sekunde kako bi stvorila vremenski prozor za drugu nit.

Druga nit (Thread 2) takođe učitava termin sa ID-jem 1 i menja status na RESERVED.

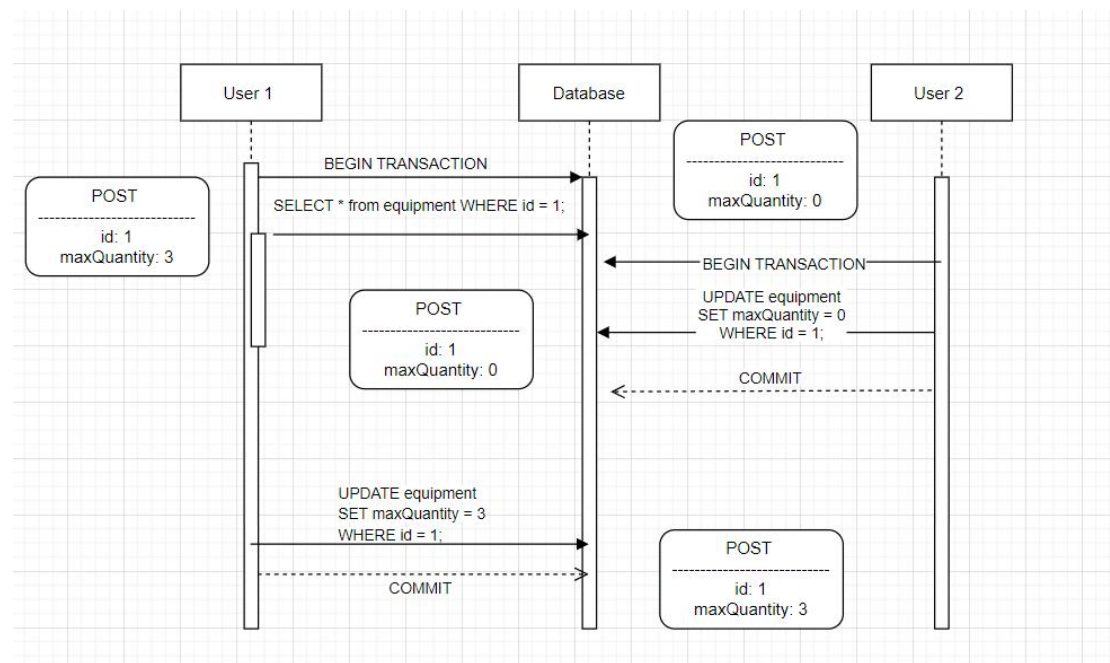
Obe niti pokušavaju sačuvati izmene. Druga nit će uspešno sačuvati izmene, dok će prva nit doživeti izuzetak

ObjectOptimisticLockingFailureException zbog sukoba u verziji.

- Konfliktna situacija:

1. više istovremenih korisnika aplikacije ne može da rezerviše opremu koja je u međuvremenu postala nedostupna (svodi se na update-ovanje polja maxQuantity neke torke u tabeli equipment u bazi podataka)

- Crtež toka koji dovodi do konflikte situacije:



- Implementacija u kodu:

Dva threada (Thread 1 i Thread 2) se pokreću paralelno, svaki pokušava rezervirati opremu sa ID-jem 1 (umanjiti maxQuantity ako je količina opreme dovoljna za korisnika).

Prva nit (Thread 1) učitava opremu sa ID-jem 1, menja vrednost polja maxQuantity na 3 i zatim čeka 3 sekunde kako bi stvorila vremenski prozor za drugu nit.

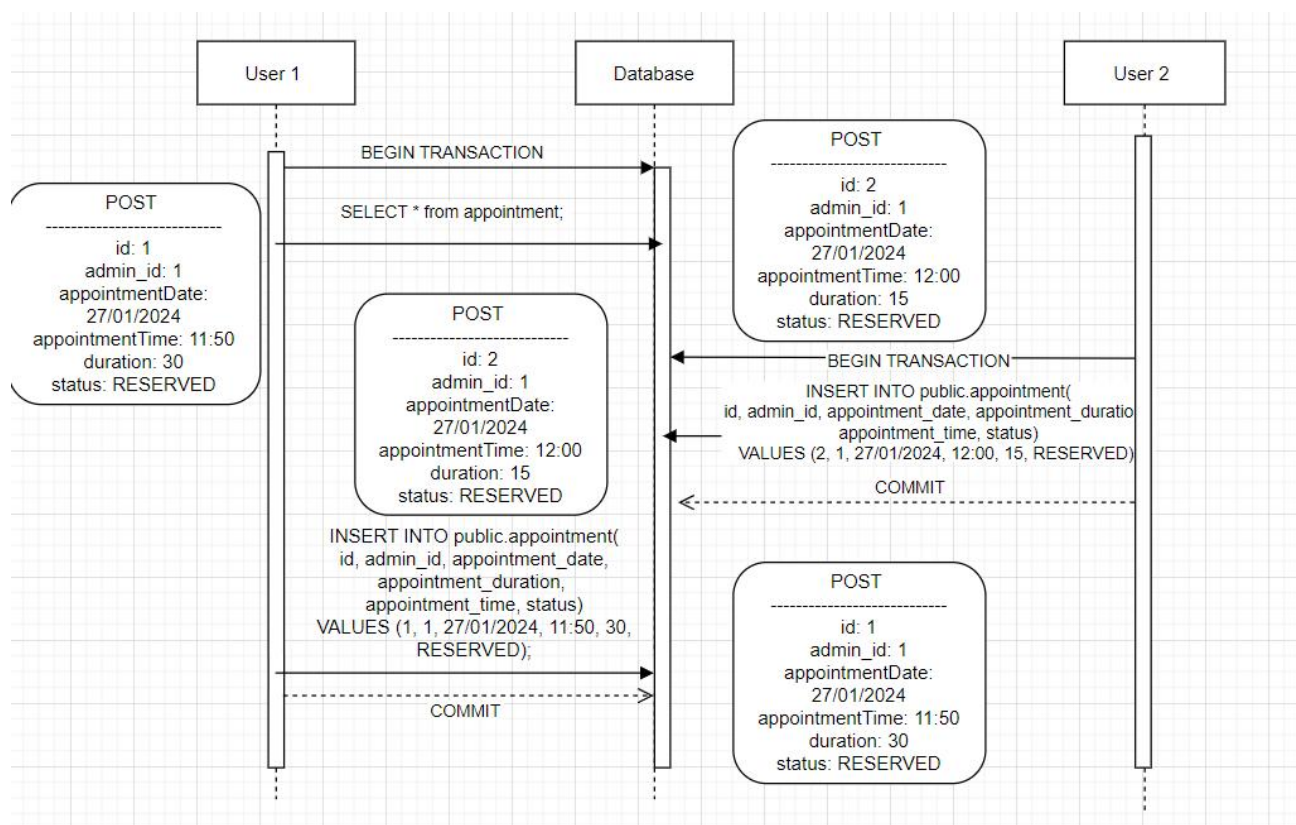
Druga nit (Thread 2) takođe učitava istu opremu sa ID-om 1 i menja vrednost polja maxQuantity na 0.

Obe niti zatim pokušavaju sačuvati izmene. Druga nit će uspešno sačuvati izmene, dok će prva nit doživeti izuzetak

ObjectOptimisticLockingFailureException zbog sukoba u verziji. Ovaj izuzetak se javlja jer prva nit pokušava sačuvati izmene na objektu koji je u međuvremenu izmenjen od strane druge niti, što stvara konflikt u verziji objekta.

Konfliktna situacija:

1. Više različitih korisnika ne smeju definisati isti ili preklapajući vandredni termin kom će prisustvovati isti administrator.



- Rešenje - implementacija u kodu:

- Sistem proverava da li postoji bilo koji admin koji je slobodan u prosleđenom terminu (proveravaju se i edge case-ovi - preklapajući termini). Ako ne postoji nijedan, korisnik se obaveštava o tome da ne

postoji nijedan slobodan administrator koji bi prihvatio njegov appointment i rezervaciju, te se moli da pokuša da zakaže drugi termin.

- Endpoint: `api/appointments/addAdminToAppointment/{companyId}`
- Metoda: `addAdminToAppointment` iz `AppointmentService`-a.