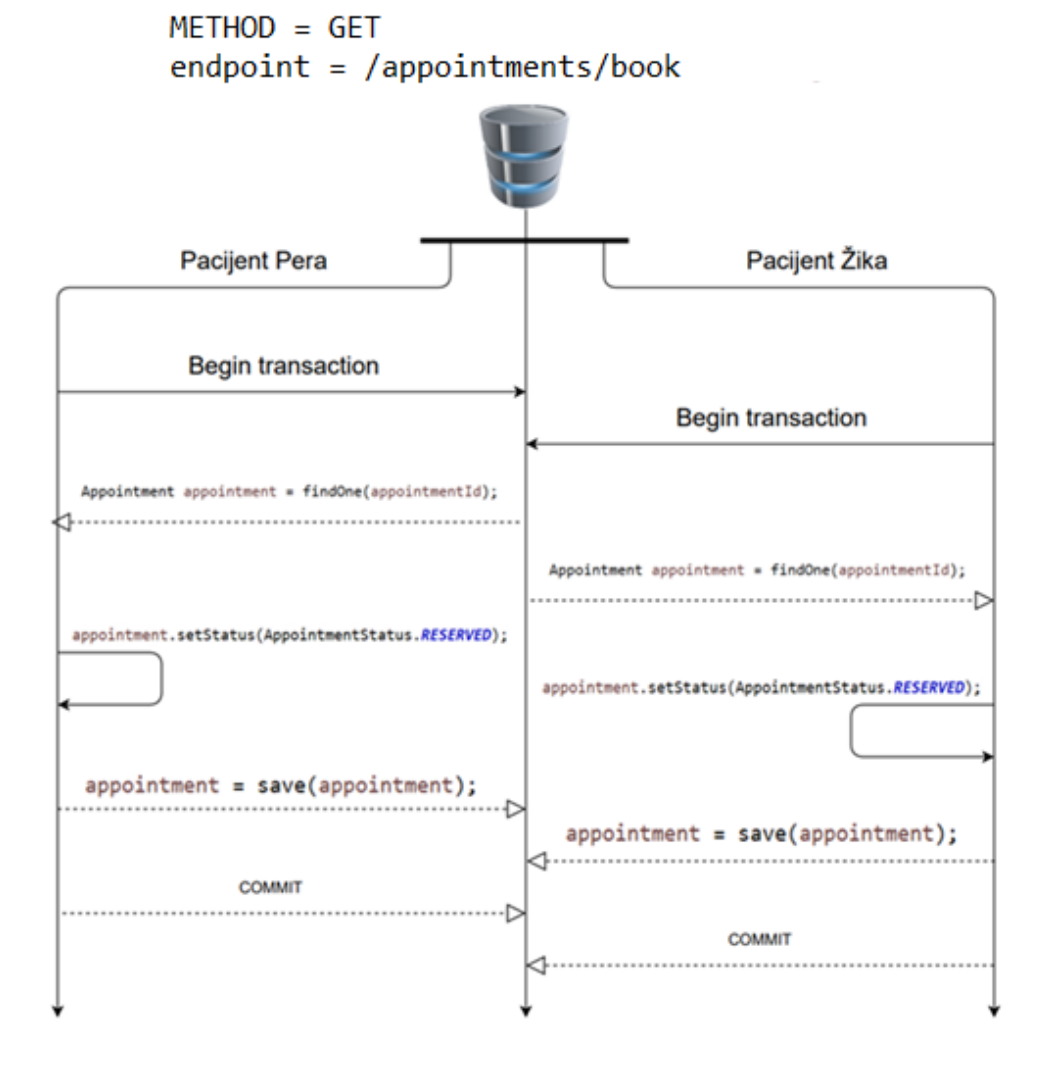


Konfliktna situacija 1:

Pregledi koji su unapred definisani ne smeju biti rezervisani od strane više različitih korisnika.

- Ova konfliktna situacija se može javiti u slučaju da dva ili više korisnika (pacijenata) istovremeno pokušavaju da rezervišu unapred definisani termin (termin koji je prethodno administrator apoteke kreirao) kod dermatologa.



Slika 1. Dva pacijenta u bliskim vremenskim trenutcima zakazuju pregled kod dermatologa

- Dva pacijenta (*Pera* i *Žika*) u približno istom vremenskom trenutku pokušavaju da rezervišu unaprijed definisani termin za pregled kod određenog dermatologa. Do konflikta dolazi ukoliko obojica zakažu predefinisani termin pregleda čija vremena se preklapaju. U ovom slučaju bi dva različita pacijenta bila prisutna na istom terminu za pregled kod dermatologa.
- Kako su oba korisnika dobila isti rezultat prilikom pravljenja upita `findOne(appointmentId)`, pacijent *Žika* nije svjestan da je klijent *Pera* prije njega potencijalno zakazao pregled u postojećem, slobodnom terminu koji se preklapa sa njegovim. U tom slučaju, nakon obavljanja *commit*-a obe akcije zakazivanja, u bazi podataka će se nalaziti podatak o pacijentu koji je posljednji izvršio *commit*, što predstavlja konflikt.
- Rešenje koje je iskorišćeno za rešavanje ovog problema je optimistično zaključavanje. Ovo rešenje je iskorišćeno jer se ova konfliktna situacija javlja u relativno retkim situacijama. Takođe, upravo zbog pomenutog rjeđeg javljanja ove situacije, nema potrebe zaključavati u potpunosti *appointment* pomoću pesimističnog zaključavanja, pogotovo uzevši u obzir da će time biti onemogućen pristup unaprijed definisanom terminu pregleda kod dermatologa, iako je taj termin slobodan.
- Korišćenjem ovog pristupa, oba pacijenta sa slike (*Pera* i *Žika*) će moći da dobave *appointment* iz baze podataka preko ID-a. Međutim, usljed optimističkog pristupa i verzionisanja, nakon prvog *commit*-a doći će do izmene verzije *appointment*-a, te će prilikom *commit*-ovanja druge transakcije doći do nepoklapanja verzija, što će dovesti do greške koja će spriječiti i onemogućiti pojavu ovog konflikta.

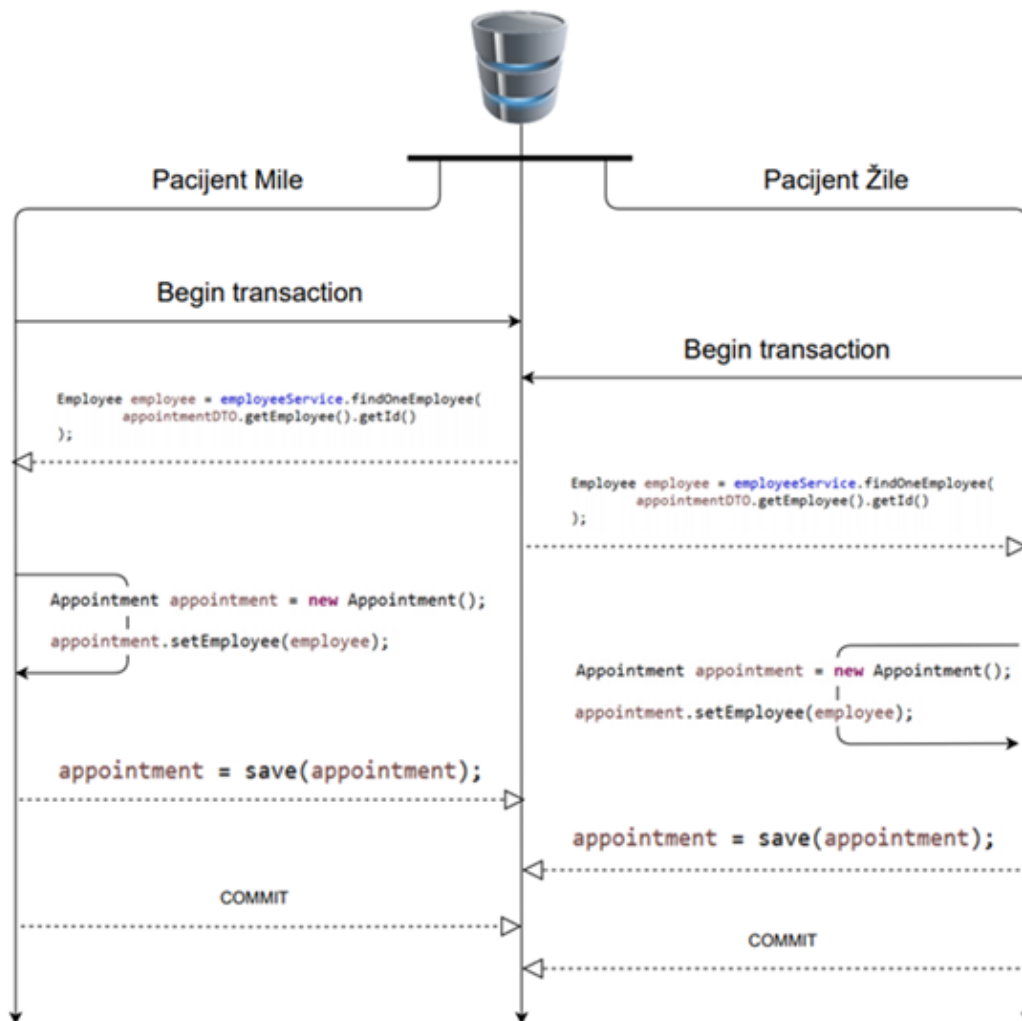
Konfliktna situacija 2:

Više istovremenih korisnika aplikacije ne može da zakaže savjetovanje u istom terminu kod istog farmaceuta (termini se ne smeju ni preklapati).

- Ova konfliktna situacija se može javiti u slučaju da dva ili više korisnika (pacijenata) istovremeno pokušavaju da rezervišu termin za savjetovanje kod farmaceuta.

METHOD = POST

endpoint = /appointments/savePharmacistAppointment



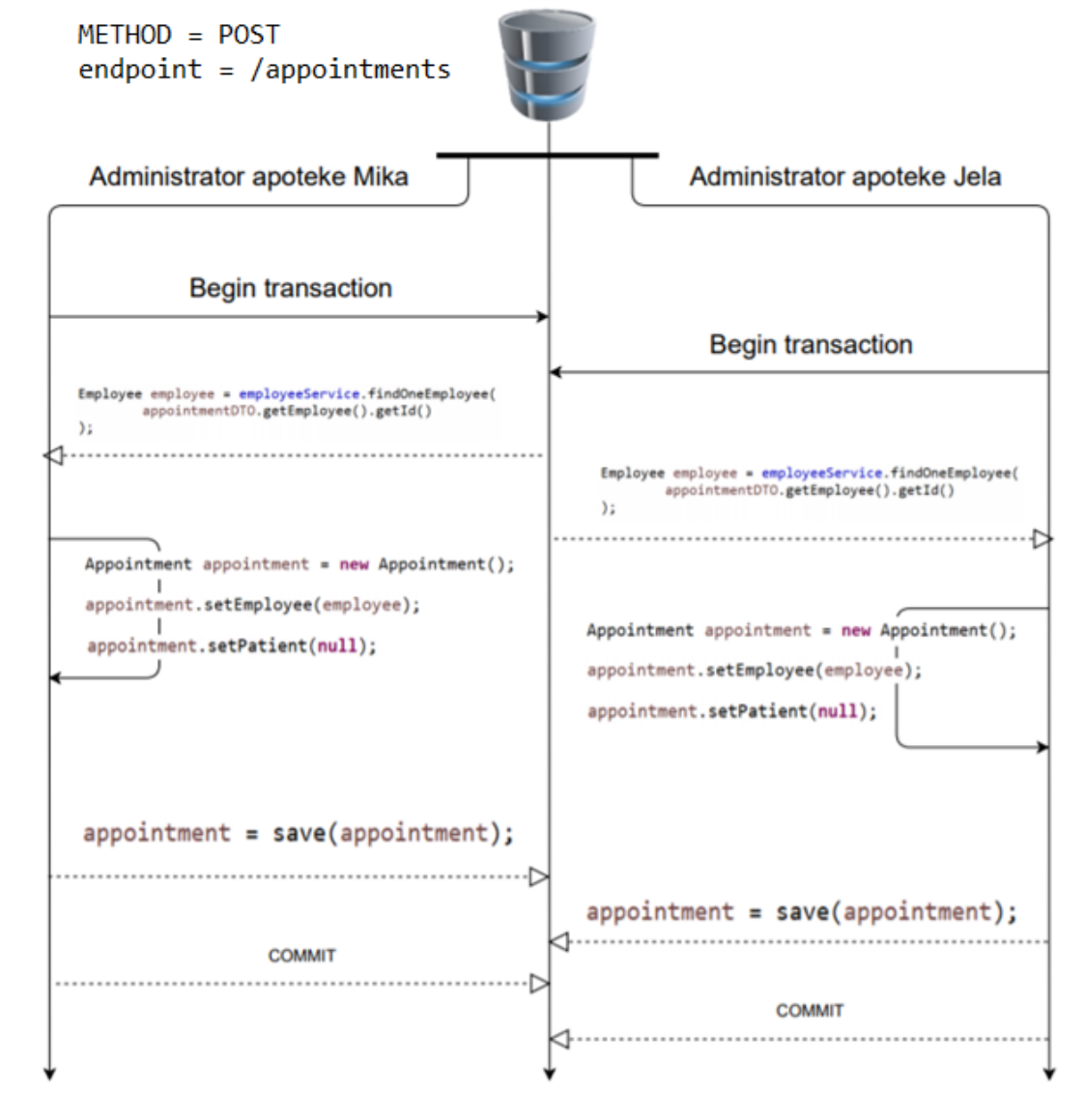
Slika 2. Dva pacijenta u približno bliskim vremenskim intervalima pokušavaju da zakažu savjetovanje kod farmaceuta.

- Dva pacijenta (*Mile* i *Žile*) u približno bliskom vremenskom trenutku pokušavaju da rezervišu termin za savjetovanje kod određenog farmaceuta. Do konflikta dolazi ukoliko obojica zakažu termin savjetovanja čija vremena se preklapaju. U ovom slučaju bi dva različita pacijenta bila prisutna na istom terminu za savjetovanje kod farmaceuta.
- Kako su oba korisnika dobila isti rezultat prilikom pravljenja upita `findOneEmployee(employeeId)`, pacijent *Žika* nije svjestan da je klijent *Pera* prije njega potencijalno zakazao savjetovanje u slobodnom terminu koji se preklapa sa njegovim. U tom slučaju, nakon obavljanja `commit`-a obe akcije zakazivanja, u bazi podataka će se nalaziti podaci o nova dva termina savjetovanja čija će se vremena preklapati, što predstavlja konflikt.
- Kao rešenje ovog konflikta korišćen je pristup pesimističnog zaključavanja. Kako se radi o konfliktu koji nastaje prilikom dodavanja novih objekata u bazu, optimistični pristup nije od pomoći - u ovoj situaciji nije bitna verzija pojedinačnih termina sačuvanih u bazi. Stoga, korišćen je pesimistički pristup prilikom čega dolazi do zaključavanja i čitanja (`PESSIMISTIC_READ`, funkcija `EmployeeRepository::findOneEmployee(Long employeeId)`). Na taj način, vrši se zaključavanje pristupa farmaceutu koji treba da održi savjetovanje koje se zakazuje, te će zbog toga jedna od transakcija dva korisnika sa slike (u ovom slučaju korisnika *Žile*) morati biti zaustavljena dok prva ne oslobodi resure, čime će drugi pacijent dobiti informaciju o tome da taj termin nije dostupan već da je u međuvremenu zakazan, te pacijent *Žile* neće uspjeti zakazati savjetovanje u tom terminu.
- Korišćenjem ovog pristupa, oba pacijenta sa slike (*Mile* i *Žile*) će moći da dobave zaposlenog farmaceuta iz baze podataka preko ID-a. Međutim, usljed pesimističkog čitanja, prvi pacijent *Mile* će uspjeti da zakaže termin za savjetovanje dok će drugi po redu *Žile* biti zablokiran opercijom što će dovesti do toga da će se onemogućiti pojava ovog konflikta, odnosno situacije da dva ili više pacijenata zakažu savjetovanje kod istog farmaceuta u isto vrijeme.

Konfliktna situacija 3:

Više administratora apoteke ne mogu istovremeno da kreiraju slobodni termin za pregled kod dermatologa u istom vremenskom periodu.

- Ova konfliktna situacija se može javiti u slučaju da dva ili više administratora apoteke istovremeno pokušavaju da kreiraju slobodni termin za pregled kod dermatologa.



Slika 3. Dva administratora apoteke u približno bliskim vremenskim intervalima pokušavaju kreirati slobodan termin za pregled kod dermatologa.

- Dva administratora apoteke (*Mika* i *Jela*) u približno istom vremenskom trenutku pokušavaju da naprave slobodan termin za pregled kod dermatologa, da bi se klijentu olakšalo zakazivanje pregleda kod dermatologa. Do konflikta dolazi ukoliko oboje pokušavaju da kreiraju slobodan termin u istom vremenskom intervalu. U ovom slučaju bi dva različita administratora apoteke napravila duple termine za pregled za jednog dermatologa.
- Kako su oba korisnika dobila isti rezultat prilikom pravljenja upita `findOneEmployee(employeeId)`, administrator *Jela* nije svjesna da je klijent *Mika* prije nje potencijalno kreirao slobodni terminu koji se preklapa sa njenim. U tom slučaju, nakon obavljanja *commit*-a obe akcije zakazivanja, u bazi podataka će se nalaziti podaci o nova dva termina za pregled kod dermatologa čija će se vremena preklapati, što predstavlja konflikt.
- Kao rešenje ovog konflikta korišćen je pristup pesimističnog zaključavanja. Kako se radi o konfliktu koji nastaje prilikom dodavanja novih objekata u bazu, optimistični pristup nije od pomoći - u ovoj situaciji nije bitna verzija pojedinačnih termina sačuvanih u bazi. Stoga, korišćen je pesimistički pristup prilikom čega dolazi do zaključavanja i čitanja (*PESSIMISTIC_READ*, funkcija `EmployeeRepository::findOneEmployee(Long employeeId)`). Na taj način, vrši se zaključavanje pristupa dermatologu koji treba da održi pregled koji se zakazuje, te će zbog toga jedna od transakcija dva korisnika sa slike (u ovom slučaju administrator *Jela*) morati biti zaustavljena dok prva ne oslobodi resure, čime će drugi pacijent dobiti informaciju o tome da taj termin nije dostupan već da je u međuvremenu zakazan, te administrator *Jela* neće uspjeti zakazati savjetovanje u tom terminu.
- Korišćenjem ovog pristupa, oba pacijenta sa slike (*Mika* i *Jela*) će moći da dobave zaposlenog farmaceuta iz baze podataka preko ID-a. Međutim, usljed pesimističkog čitanja, prvi pacijent *Mika* će uspjeti da kreira slobodni termin za pregled dok će druga po redu *Jela* biti zablokirana opercijom što će dovesti do toga da će se onemogućiti pojava ovog konflikta, odnosno situacije da dva ili više administratora kreiraju slobodni termin u isto vrijeme.