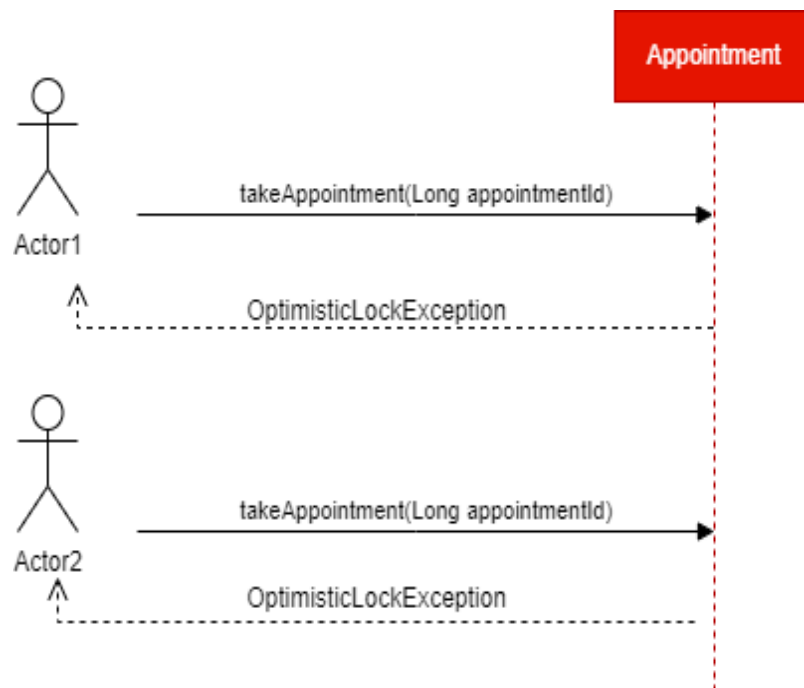


Konfliktne situacije

Više korisnika ne može da rezerviše termin koji je u međuvremenu postao nedostupan. Pošto objekat klase *Appointment* ima polja *status* i *version*, prilikom pokušaja rezervacije termina provjerava se njegov statut. Pošto više korisnika može u isto vrijeme može da pokuša da rezerviše bilo koji termin, provjerava se i verzija kako bi objekat bio u konzistentnom stanju. Ovdje je rađeno optimističko zaključavanje, pošto se pretpostavlja da bi pesimistično zaključavanje dovelo do pada performansi. U slučaju da dva korisnika pokušaju da rezervišu termin dobijaju *exception*. Korištena je anotacija *@Transactional*. Transakcija je testirana testom koji simulira situaciju kada dva korisnika pokušaju da rezervišu termin, te u tom slučaju baca *OptimisticLockException*.



Više korisnika ne može da rezerviše termin koji se preklapa. Provjera je implementirana na nivou metode za čuvanja objekta klase *Appointment*, tako da korisnici ne mogu da rezervišu termine koji se preklapaju ni prilikom mjenja termina ni prilikom kreiranja novog. Prilikom provjere, sabira se trajanje termina sa početkom termina i tako se gleda preklapanje sa već postojećim terminima. Provjera je testirana sa dva testa, jedan provjerava neuspješan, a jedan uspješan scenario. Pošto postoji verzija objekta klase *Appointment* i na taj način se osigurava konzistentnost objekta.



Više korisnika / transakcija ne može u isto vrijeme da mijenja broj jedinica krvi koje jedna banka krvi posjeduje. Pošto je količina krvi koje banka u datom momentu posjeduje, predstavlja djeljeni resource kome pristupa više učesnika uvedeno je polje *version* na osnovu koga se prati i očuvava konzistentnost jedinica krvi koje banka krvi posjeduje. U ovom slučaju rađeno je optimističko zaključavanje, zarad očuvanja performansi. Kako je osnovni resurs banke krvi, krv potrebno je da polje koje prati status tpg resursa uvijek dostupno i uvijek konzistentno. Ovaj slučaj je testiran sa dva testa, jedan koji provjerava da li je stanje ostalo nepromijenjeno nakon pojave izuzetka *OptimisticLockException*. Postoje dva korisnika, jedan radi čitanje, drugi radi izmjenu. U tom slučaju stanje objekta, koji se mijenja je nekonzistentno i za čitanje, jer je u fazi izmjena. Drugi test simulira slučaj kada dva korisnika pokušaju da mijenjaju stanje jedinica krvi. U tom slučaju ponovo se dešava *OptimisticLockException*.

