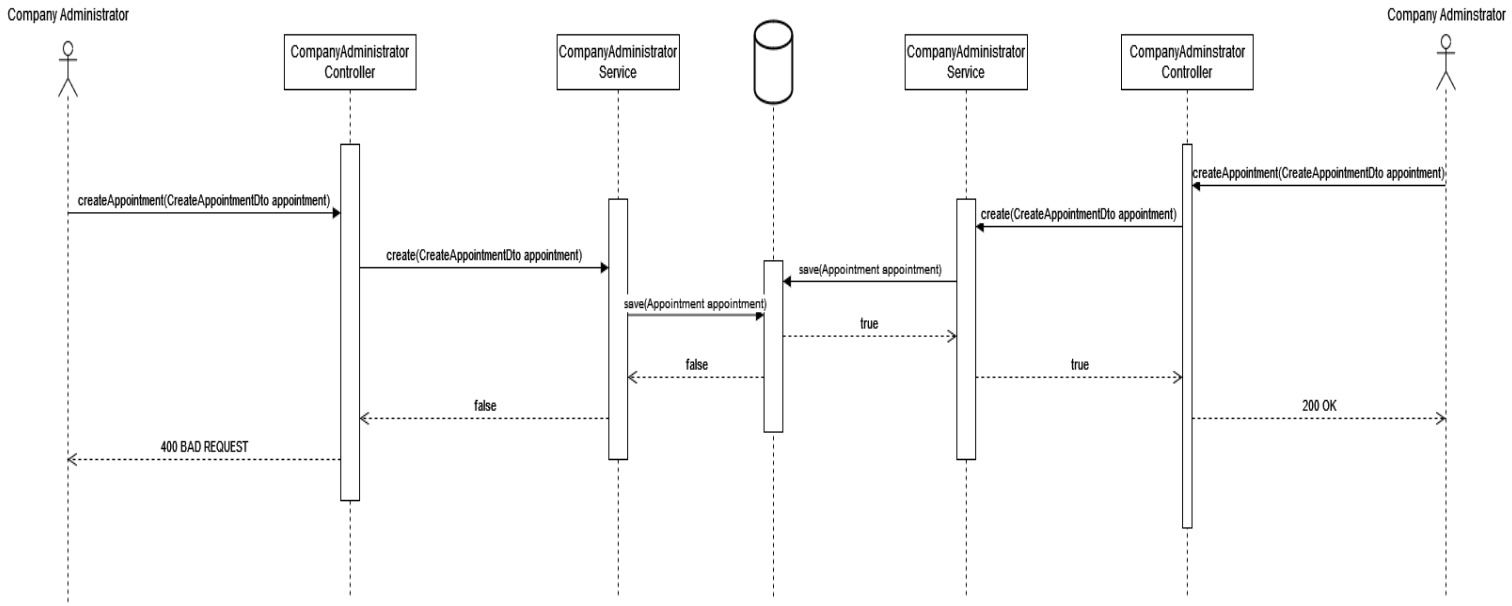
Konkurentni pristup bazi – Student 3

1. Više administratora kompanije ne mogu unapred definisati termine u isto ili preklapajuće vreme.

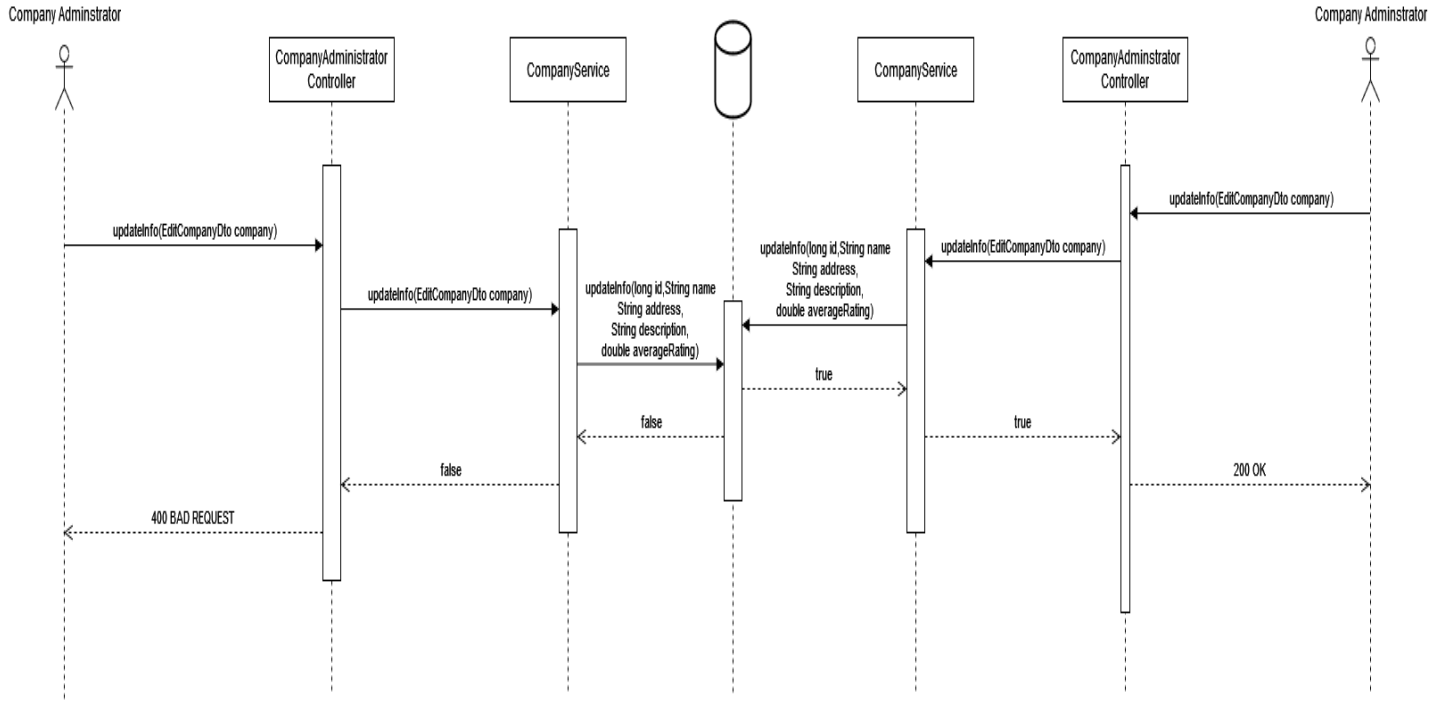
Jedna od vaznijih funkcionalnosti ovog sistema jeste da administratori mogu da kreiraju termine za rezervisanje neke opreme koju kompanija nudi. Potrebno je obezbediti da administratori ne kreiraju termine u isto ili preklapajuce vreme kako bi se obezbedilo adekvatno pracenje kreiranih termina i kako ne bi doslo do konflikata u samim terminima koji su kreirani. Da bi se ovaj problem resio i da bi izbegli da na primer dva razlicita administratora iste kompanije kreiraju termine u isto ili preklapajuce vreme, potrebno je resiti ovu konfliktnu situaciju. Na slici je prikazana situacija kad dva razlicita administratora pokusavaju da kreiraju u preklapajuce vreme termin gde je korisnik desno pokrenuo transakciju nekoliko sekundi ranije (na dijagramu je izbegnut prikaz svih metoda servisa koje ucestvuju u kompletnom ostvarenju zeljenog cilja kreiranja termina sa namerom da prikazem samu sustinu transakcije).



Problem je resen upotrebom optimistickog zakljucavanja. U klasi Appointment dodato je polje version sa anotacijom @Version koje nam omogucava da izvrsimo verzionisanje torke u bazi. Za kreiranje termina koriscena je metoda create servisa CompanyAdministratorService. Metoda je oznacena kao transakciona anotacijom @Transactional, nivo izolacije je podesen na SERIALIZABLE, atribut readonly postavljen na false posto vrsimo izmenu u bazi, a atribut propagation na REQUIRED. Ovim je resen glavni problem konkurentnog pristupa bazi. Obezbedjeno je da metoda uvek pokrece novu transakciju, a ako postoji tekuca transakcija ona se suspenduje. Tako da na nasem primeru onaj koji je prvi pokrenuo transakciju uspece uspesno da kreira termin, dok ce pokusaj kreiranja od strane drugog administratora biti neuspesan posto ce njegova transakcija biti suspendovana zbog postojanja transakcije od strane administratora koji pre njega pokusava kreirati termin.

2. Informacije o kompaniji ne mogu biti menjane istovremeno od strane razlicitih administratora iste kompanije.

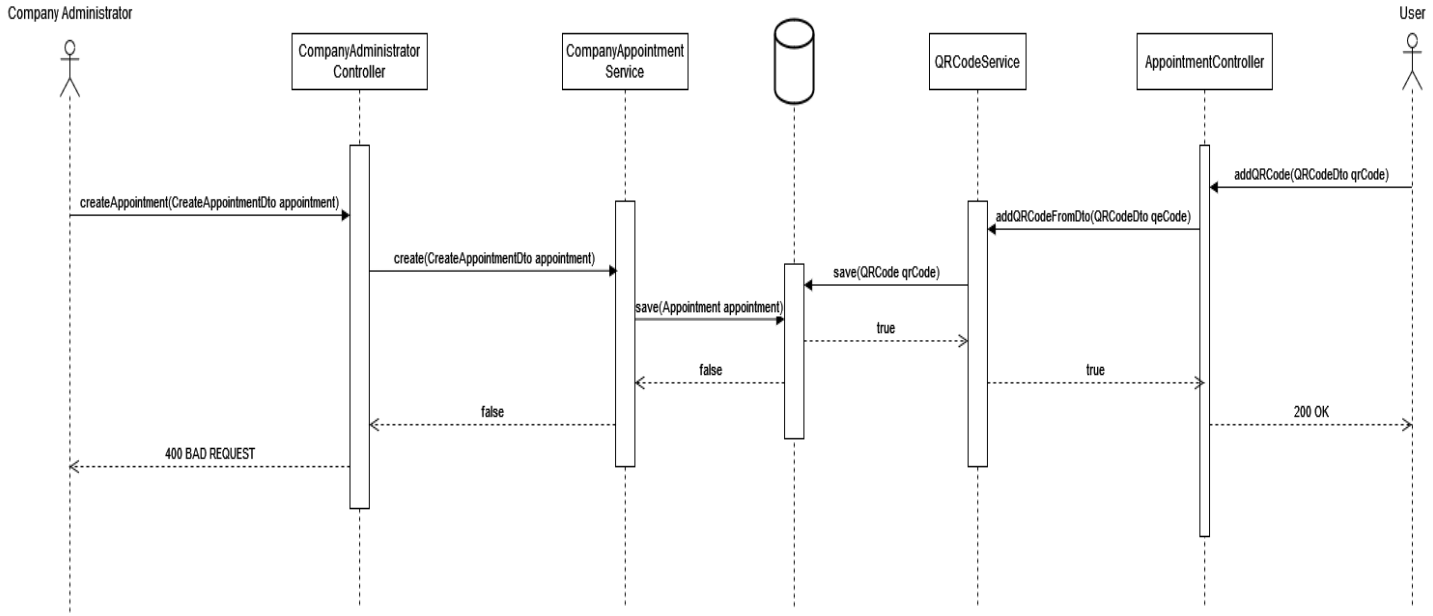
Administrator kompanije moze da menja informacije o kompaniji u kojoj radi. Posto kompanija moze da ima vise administratora, moze doci do problema gde vise razlicitih administratora iste kompanije pokusavaju da menjaju informacije kompaniji. Kako bi ovaj problem izmene informacija o kompaniji resili, potrebno je u stvari resiti konfliktnu situaciju koja nam se namece. Na slici je prikazana situacija kad dva razlicita administratora pokusavaju da izmene informacije o istoj kompaniji gde je administrator desno pokrenuo transakciju nekoliko sekundi ranije (na dijagramu je izbegnut prikaz svih metoda servisa koje ucestvuju u kompletnom ostvarenju zeljenog cilja izmene informacija sa namerom da prikazem samu sustinu transakcije).



Problem je resen upotrebom optimistickog zakljucavanja. U klasi Company dodato je polje version sa anotacijom @Version koje nam omogucava da izvrsimo verzionisanje torke u bazi. Za izmenu informacija o kompaniji koriscena je metoda updateInfo servisa CompanyService. Metoda je oznacena kao transakciona anotacijom @Transactional, nivo izolacije je podesen na SERIALIZABLE, atribut readonly postavljen na false posto vrsimo izmenu u bazi, a atribut propagation na REQUIRED. Ovim je resen glavni problem konkurentnog pristupa bazi. Obezbedjeno je da metoda uvek pokrece novu transakciju, a ako postoji tekuca transakcija ona se suspenduje. Tako da na nasem primeru onaj koji je prvi pokrenuo transakciju uspece uspesno da izmeni informacije o kompaniji, dok ce pokusaj menjanja od strane drugog administratora biti neuspesan posto ce njegova transakcija biti suspendovana zbog postojanja transakcije od strane administratora koji pre njega pokusava menjati informacije o kompaniji.

3. Administrator kompanije ne može unapred definisati termin u isto ili preklapajuće vreme za koje i korisnik kreira rezervaciju termina

Jedna od vaznijih funkcionalnosti ovog sistema jeste da administratori mogu da kreiraju termine za rezervisanje neke opreme koju kompanija nudi. Potrebno je obezbediti da administrator ne kreira termin u isto ili preklapajuce vreme kada i korisnik kreira rezervaciju termina. Kako bi resili potencijalni problem preklapanja prilikom kreiranja termina od strane administratora i kreiranja rezervacije termina od strane korisnika, potrebno je resiti konfliktnu situaciju koja nam se namece. Na slici je prikazana situacija kad administrator pokusava da kreira termin ali nekoliko trenutaka nakon sto je korisnik pokusao da rezervise termin (korisnik je pokrenuo svoju transakciju nekoliko trenutaka ranije u odnosu na administratora) (na dijagramu je izbegnut prikaz svih metoda servisa koje ucestvuju u kompletnom ostvarenju zeljenih ciljeva sa namerom da prikazem samu sustinu transakcije).



Problem je resen upotrebom optimistickog zakljucavanja. U klasi Appointment dodato je polje version sa anotacijom @Version koje nam omogucava da izvrsimo verzionisanje torke u bazi. Za kreiranje termina koriscena je metoda create servisa CompanyAdministratorService, dok je za rezervisanje odnosno kreiranje QR koda (koji u sebi sadrzi appointment) koriscena je metoda addQRCodeFromDto servisa QRCodeService. Metode su oznacene kao transakcione anotacijom @Transactional, nivo izolacije je podesen na SERIALIZABLE, atribut readonly postavljen na false posto vrsimo izmenu u bazi, a atribut propagation na REQUIRED. Ovim je resen glavni problem konkurentnog pristupa bazi. Obezbedjeno je da metoda uvek pokrece novu transakciju, a ako postoji tekuca transakcija ona se suspenduje. Tako da na nasem primeru ukoliko je korisnik prvi pokrenuo transakciju koja se odnosi na rezervisanje termina uspece uspesno da rezervise termin, dok ce transakcija za kreiranje termina koju je pokrenuo administrator nekoliko trenutaka kasnije biti suspendovana i administrator u tom trenutku nece uspeti da kreira termin.