

Konfliktna situacija u ISA projektu

Ovaj pdf objasjava jednu od konfliktnih situacija sa kojima sam se susreo prilikom izrade projekta. Sadrži opis situacije i način na koji je situacija rešena sa crtežom tokova zahteva.

Zadatak po specifikaciji mi je bio da realizujem funkcionalnosti i razvijem drugacije tipove korisnika vezane za studenta 2. Ovo se većim delom svodi na implementaciju funkcija koje mogu da obavljaju doctor i administrator klinike (u tekstu **clinicAdmin**). Stoga su se konfliktna situacije koje sam rešio većim delom svodile na CRUD operacije sa podacima kojima rukovode.

Bitan deo aplikacije je i rad oko pregleda (u tekstu **medicalCheckup**). Nekoliko funkcionalnosti se odnosilo na rezervisanje i iscitavanje **medicalCheckup**-a, pa se i tu pojavilo nekoliko konfliktnih situacija.

U ovom dokumentu sam opisao konflikt sa kojim sam se susreo prilikom prikaza liste zahteva za odmor clinicAdmin-u.

O Aplikaciji

Organizacija

Ova aplikacija je Monolitnog tipa. Organizacija aplikacije je do neke mere slična Layered Architecture organizaciji. Sastoji se iz:

- Front end dela (html, css I js file-ovi)
- Controllera (reaguju na ajax call-ove od strane frontEnd-a I vraćaju rezultate na front)
- Response I Request klasa (ono što dobija/prosleđuje kontroler se pojavljuje na serveru u ovakvom obliku)
- Register klasa (koriste se za validaciju unetih podataka gde za to ima potrebe)
- Repo klasa (direktno prica sa bazom)
- DbModel klasa (podaci koji se čuvaju ili vade iz baze se pojavljuju na server u ovakvom obliku)
- Baze podataka (u ovom slučaju to je H2)

Izrada I Okruženje

Aplikacija je radjena u Spring Tool Suite-u. Za izradu aplikacije sam koristio Spring I Maven.

Na frontu sam koristio JQuery i Bootstrap. Kod client-side validacije sam koristio Thymeleaf.

Za komunikaciju između frontend-a I backend-a sam koristio Ajax.

Backend sam radio u Javi. Kao bazu podataka sam koristio H2.

Zahtev za godisnji odmor – clinicadmin I enable-ovan doctor su iz razlicite bolnice

Opis Problema

ClinicAdmin na Clinic Manager page-u ima pristup stranici za odobravanje ili odbijanje zahteva za godisnji odmor od strane doktora. Cim udje na stranicu, bice mu izlistani svi zahtevi koje on moze da odobri ili odbije.

ClinicAdmin ne sme da prihvata ili odbija zahteve za godisnje odmore od strane doktora iz drugih klinika, stoga ne smeju njihovi zahtevi da mu se pojave u tabeli.

Moze da se desi da su doctor I clinicAdmin iz iste bolnice (u tekstu clinic), a da je doctor izbacen Iz Sistema. Drugim recima, clinicAdmin ne treba da dobija stare odmore od doctor-a koji nije vise u sistemu.

Detalji Relevantni Problemu

Za proveru da li je doctor enable-ovan se koristi enabled field unutar doctorDbModela koji se nabavlja iz baze zahvaljujuci doctorRepo-u.

Svi zahtevi za odmor se cuvaju u tabelu u bazi. Odmor u bazi ima **id**, **startDate**, **endDate**, **approved** I **enabled** field-ove. Doctor slanjem zahteva ubacuje novi vacation entry u vacation table.

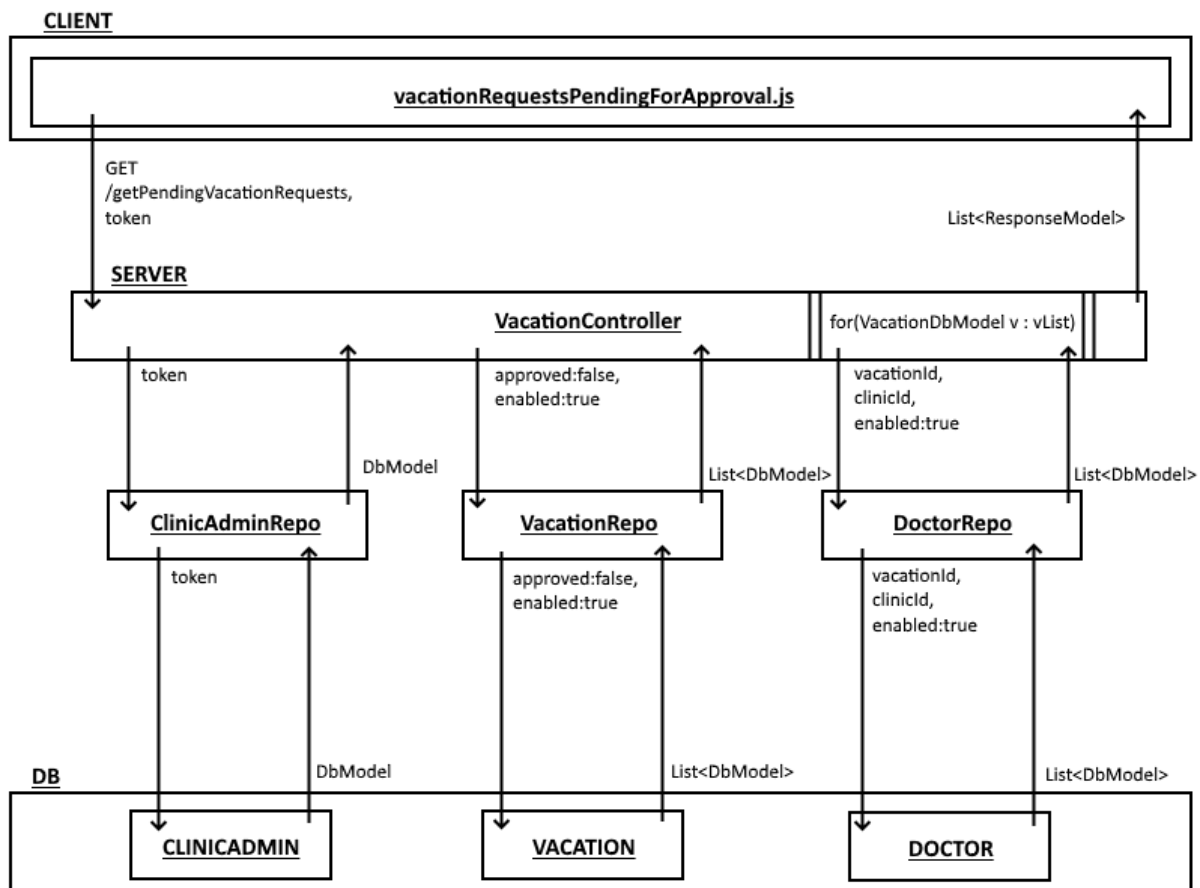
Za iscrtavanje tabele koristi se Tabulator (css I js file-ovi). On se puni tako sto se ajax GET pozivom nabavi lista svih relevantnih vacation-a I preko Tabulator-ove funkcije setData se load-uje data u tabelu.

ClinicAdmin dobija listu svih enable-ovanih vacation-a koji **nisu** approve-ovani. Approve-ovanjem vacation postaje **approved** I **enabled**, a Rejectovanjem ostaje ne approve-ovan I postaje **disabled**.

Vacation, ClinicAdmin, Doctor i Clinic svi imaju svoje odgovarajuće tabele u bazi, repo klase i DbModel klase.

Resenje

Crtez toka zahteva:



Postupak realizacije resenja:

- Počinje se od ajax GET poziva na **“/getPendingVacationRequests”**. Na taj poziv odreaguje `VacationController`.
- Unutar `VacationController`-a se pokreće prvi poziv bazi. Proverava se da li je taj koji pokreće akciju stvarno `clinicAdmin` (traži se u tabeli `CLINICADMIN` po `securityToken`-u). Ako nije,

akcija se završava I klijentu se šalje

HttpStatus.UNAUTHORIZED.

- Ako jeste, vrši se drugi poziv bazi kako bi se dobavila lista svih relevantnih vacation-a. Vadi se iz baze po kriterijumima **approved=false I enabled=true.**
- Nakon toga se kreira Prazna lista VacationResponseModel-a. Ova lista će se na kraju poslati client-u.
- Za svaki pronađeni vacation se vrši poseban poziv bazi. Traži se lista doctora čiji se **id-ovi poklapaju sa id-om vacation-a**, koji pripadaju **istoj klinici kao I clinicAdmin** I koji su **enable-ovani**. Ako je lista prazna, ovaj vacation se preskace I sledeći se ispituje.
- Ako lista ima barem jednog doctor-a, gradi se VacationResponseModel od **vacation id-a, imena I prezimena prvog doctor-a u listi, startDate-a I endDate-a vacation-a**. Novo napravljeni VacationResponseModel se stavlja u listu.
- Nakon što su pronađeni svi relevantni vacation-i I stavljeni u listu, Lista se šalje client-u kao **ResponseEntity** sa statusom **HttpStatus.OK.**