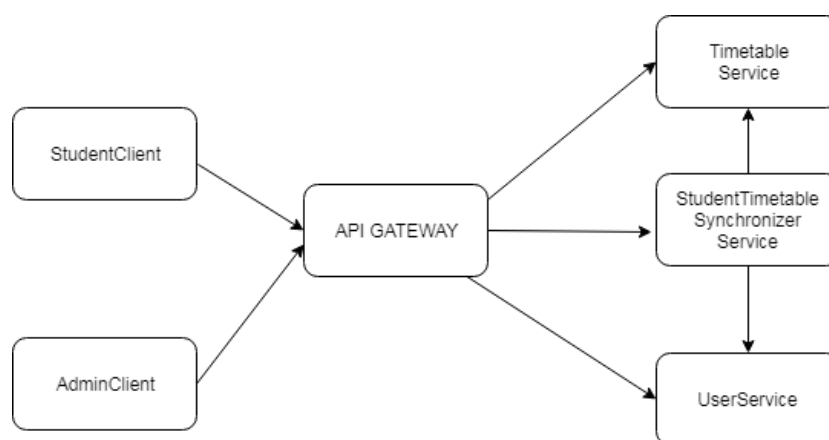


Specifikacija drugog projekta iz predmeta Softverske komponente - školska 2017/2018. godina

Implementacija modularnog softverskog sistema za pretragu i sinhronizaciju rasporeda studenta

Opis projekta:

Osmisliti i implementirati softverski sistem za pretragu i sinhronizaciju rasporeda na fakultetu. Sistem treba da bude baziran na mikroservisnoj softverskoj arhitekturi kako je prikazano na sledećem dijagramu:



Sistem se sastoji od 3 servisne komponente (mikroservisa), to su:

- *TimetableService* - u bazi podataka sadrži informacije o rasporedu na fakultetu i obezbeđuje servis za unos rasporeda i pretragu rasporeda po grupi, učionici i danu u nedelji. Potrebno je osmisliti operacije servisa koje omogućavaju unos rasporeda u bazu, pretragu po jednom ili više navedenih kriterijuma kombinovano (sa operatorom and). Operacije za pretragu treba da vrate listu predavanja i vežbi koji zadovoljavaju unete kriterijume pretrage sa svim podacima iz rasporeda.
- *UserService* - u bazi podataka čuvaju se nalozi korisnika sistema, što mogu biti studenti ili administratori. Informacije o studentima uključuju korisničko ime i lozinku, ime, prezime, broj indeksa (smer, broj i godinu) i grupu kojoj pripada student. Administratori imaju podatke za logovanje, ime, prezime i email.
- *StudentTimetableSynchronizerService* - komponenta koja raspored studenta sinhronizuje sa Google kalendarom. Komponenta treba da obezbedi servis koji prima id studenta (ili broj indeksa), datum početka i završetka važenja rasporeda i da na osnovu tih podataka izvrši sinhronizaciju rasporeda studenta sa Google kalendarom. Implementacija servisa treba da obuhvati komunikaciju sa drugim komponentama, prvo iz komponente *UserService* preuzima grupu kojoj pripada student, zatim iz komponente *TimetableService* preuzima raspored za tu grupu. Zatim se preuzeti raspored sinhronizuje sa Google kalendarom uzimajući u obzir

datume početka i završetka važenja rasporeda. Operacija servisa treba da vrati poruku o uspešnosti izvršene sinhronizacije, u slučaju uspešne sinhronizacije u poruci treba vratiti i broj kreiranih događaja u kalendaru.

Sistem treba da obezbedi i dve klijentske komponente koje pozivaju servise:

- *StudentClient* - aplikacija preko koje student može da se registruje na sistem. Posle registracije može da se loguje i da pretraži raspored po grupi, učionici i danu u nedelji. Preko ove aplikacije student može da izabere grupu iz ponuđene liste svih grupa koje postoje u rasporedu. Pošto je izabrana grupa sačuvana, student može da pozove sinhronizaciju sa Google kalendarom, pre toga unosi preko forme datume početka i završetka važenja rasporeda.
- *AdminClient* - aplikacija preko koje se loguju administratori (za potrebe testiranja uneti direktno u bazu jednog administratora) i unose raspored. Unos treba implementirati kao učitavanje csv fajla (skinutog sa RAF sajta) i pozivanje odgovarajućeg mikroservisa.

U primeru je predloženo da se koristi API Gateway patern u kom se koristi dodatna serverska komponenta kao single-point entry sistema koji služi da usmeri zahteve na odgovarajući servis. Korišćenje ovog paternu nije obavezno i druga rešenja su prihvatljiva.

Preporučeni jezik je Java sa Spring Boot tehnologijom i REST-om, a dozvoljeno je koristiti i druge programske jezike, tehnologije i standarde koja podržavaju servise. Od baza se može koristiti bilo koja relacionala ili NoSQL baza (može i in-memory varijanta).

Klijentske aplikacije se mogu napraviti kao veb ili desktop aplikacije po slobodnom izboru tehnologije. Dovoljno je implementirati jednostavne ekranske forme, bez komplikovanog dizajna.

Raspodela poena:

- implementacija unosa rasporeda preko admin naloga i poziva servisa - 5p,
- implementacija funkcionalnosti registracije i logovanja studenta (autentifikacije preko tokena), logovanje se može implementirati na osnovu naloga u bazi (UserService) ili preko Google-a - 5p,
- implementacija pretraživanja rasporeda - 4p,
- implementacija izbora grupe - 3p,
- implementacija sinhronizacije rasporeda studenta - 8p.

U slučaju da se neki od delova izostave, da bi ostali mogli da se testiraju mogu se "ručno" ubaciti podaci u bazu.

Projekti se mogu raditi individualno ili u paru. U slučaju rada u paru oba studenta moraju da znaju da objasne ceo projekat.

Rok za predaju je 7.6. 2018. u 20h, radove slati na email bdimicsurla@raf.rs, kao link do šerovanih fajlova ili link do git repozitorijuma. Projekat se brani u četvrtoj kolokvijumskoj nedelji u zakazanom terminu za predmet.