Univerzitet Union
Računarski fakultet
Predmet: Softverske komponente

Specifikacija drugog projekta

Osmisliti i implementirati sistem za **Notifikacije** putem maila u mikroservisnoj arhitekturi.

Kratak opis:

Korisnik se prijavljuje na sistem i dobija korisnički token sa kojim se dalje autorizuje prilikom svakog obraćanja sistemu. Može da pregleda liste obaveštenja koja sistem nudi i da se pretplaćuje na njih.

Obaveštenja mogu biti: informacije o vremenskoj prognozi, letovima sa određenih aerodroma, kursnoj listi i sl. Kada se korisnik pretplati na određenu listu, sistem će mu periodično (zavisno od tipa pretplate), putem maila slati informacije od interesa.

Funkcionalnosti:

Sistem minimalno treba da se sastoji od tri mikroservisa:

- **Servis 1** - funkcionalnosti:
  - Registracija i prijava korisnika
  - Pregled i čuvanje informacija o korisnicima u bazi podataka (po izboru)
  - Pregled informacija o tipovima obaveštenja koje sistem nudi kao i pretplatama korisnika na tipove obeveštenja
  - Mogućnost korisnika da se prijavi/odjavi na određenu listu obeveštenja (**napomena**: korisnik bira na koju se listu prijavljuje i koji vremenski tip obaveštenja želi, ukoliko je podržano više mogućih tipova (časovni, dnevni, nedeljni…))
  - Povlačenje podataka sa **servisa 2** i prosleđivanje na **servis za slanje maila** preko message brokera
- **Servis 2** - funkcionalnosti:
  - Integracija na neki open api (minimalno jedan) radi povlačenja informacija od interesa
  - Periodično povlačenje podataka sa udaljenog servisa i skladištenje u lokalnoj bazi podataka
  - Otvorene API-je za pregled ponudjenih podataka na koje se integriše **Servis 1**
- **Servis 3** - funkcionalnosti:

- Autorizacija na mail servis i slanje maila iz aplikacije
- Pretplaćivanje na neki message broker (RabbitMQ, Kafka, ActiveMQ…) i osluškivanje poruka za slanje. Nije potrebno implementirati templejte i pripremu stilizovanog sadržaja maila koji se šalje. Dovoljno je samo sa brokera pokupiti informaciju o adresi na koju se šalje mail i text koji ide u samu poruku.

Servisi 1 i 2 moraju komunicirati preko http-a a poželjno je (ne i obavezno) da se prilikom komunikacije implementira i neki **circuit breaker** mehanizam (npr. Hystrix) radi bolje rezilijentnosti aplikacije. **Servis 1** će povlačiti podatke od **Servisa 2** za određene liste na koje korisnici mogu biti pretplaćeni. **Servis 1** će potom, prikupljene podatke izbaciti na neki queue message brokera, a **Servis 3** (mail servis) to osluškuje, čita poruke sa brokera i šalje ih mailom.

Korisnici ne smeju direktno kontaktirati servise, već između njih mora postojti **api gateway** koji će se baviti rutiranjem. (Na vežbama ćemo obrađivati Netflix biblioteke za api gateway (**Zuul**) i service discovery (**Eureku**) i dozvoljeno je iskoristiti ih u projektu).

Takođe, potrebno je obezbediti minimalističku klijentsku aplikaciju (desktop ili web) u tehnologiji po izboru, koja će olakšati korisniku interakciju sa sistemom. Aplikacija treba da ima par tabela u kojima će biti prikazane liste na koje se korisnik može prijaviti kao i korisnike koji su prijavljeni na njih. Korisnik se, interakcijom sa frontendom može prijaviti/odjaviti na neku od lista.

Bodovi:

- Klijentska aplikacija - 5
- Integrisan Api-Gateway - 5
- Servis 1 - 7
- Servis 2 - 7
- Servis 3 - 6

Rok za predaju:

Početak druge kolokvijumske nedelje: **10.01.2020. 23:59**

Projekat predajete asistentu kod koga slušate vežbe (prema grupi u koju ste raspoređen)i slanjem linka do git repozitorijuma ili google drive-a.