

Predmetni projekat

DevOps

Školska godina 2022/2023.

Kroz predmetni projekat treba da implementirate platformu za ponudu i rezervisanje smeštaja.

Platforma za rezervisanje smeštaja

Arhitektura

Aplikacija treba da se sastoji iz back-end dela, koji sadrži poslovnu logiku sistema i ima mikroservisnu arhitekturu, i front-end dela, koji pruža grafički korisnički interfejs za pristup funkcionalnostima.

Uloge u sistemu

- **Neautentifikovani korisnik (NK)** - Može da kreira novi host ili guest nalog ili se prijavi na postojeći. Pored toga, može da pretražuje smeštaj, ali ne može da ga rezerviše niti da postavlja nove ponude.
- **Host (H)** - Kreira novi smeštaj i upravlja njime. Za svaki smeštaj definiše opremljenost, periode dostupnosti i cenu. Vlasnik nekretnine bira da li se ona može automatski rezervisati ili samo uz njegovo odobrenje. Host na svom nalogu može da vidi i pretražuje sve nekretnine, ali ne može da ih rezerviše.
- **Guest (G)** - Rezerviše smeštaj. Svaku rezervaciju i zahtev za rezervaciju može otkazati pre njenog početka. Dostupna mu je funkcionalnost ocenjivanja smeštaja i host naloga.

Funkcionalni zahtevi

1.1 Registracija naloga (uloge: NK)

Svaki nalog mora imati jedinstven username. Minimalan skup ličnih podataka koje korisnik mora uneti pri registraciji je: ime, prezime, email adresa i mesto stanovanja.

1.2 Prijava na sistem (uloge: NK)

1.3 Upravljanje nalogom (uloge: H, G)

Korisnik može menjati sve lične podatke koje je uneo prilikom registracije, kao i svoje kredencijale (username i lozinku).

1.4 Brisanje naloga (uloge: H, G)

Guest može obrisati nalog ako nema aktivnih rezervacija. Host može obrisati nalog samo ako nema aktivnih rezervacija u budućnosti ni za jedan smeštaj

kojim upravlja. Kada host obriše nalog, uklanjaju se i svi smeštaji koje je kreirao.

1.5 Kreiranje smeštaja (uloge: H)

Za smeštaj je potrebno uneti minimalno naziv, lokaciju, pogodnosti (na primer wifi, kuhinja, klima, besplatan parking itd), fotografije i minimalan i maksimalan broj gostiju.

1.6 Definisanje dostupnosti i cene smeštaja (uloge: H)

Potrebno je omogućiti definisanje i izmenu termina kada je neki smeštaj dostupan. On se može rezervisati samo za vremenski interval kada je dostupan i nije već zauzet. Pored toga, cena smeštaja može biti promenljiva. Na primer ona može biti veća u letnjim mesecima, vikendom ili za vreme praznika. Omogućiti host-u da definiše ovakve razlike u ceni. Cena može da se zada po gostu ili po celoj smeštajnoj jedinici, a host bira jednu od te dve mogućnosti na nivou nekretnine. Dostupnost smeštaja i cena mogu se menjati za neki interval samo ako u njemu nema rezervacija.

1.7 Pretraga smeštaja (uloge: NK, H, G)

Smeštaj se pretražuje prema lokaciji, broju gostiju i datumu početka i kraja putovanja. Rezultati treba da obuhvate sve dostupne smeštaje u zadatom intervalu koji se nalaze na traženoj lokaciji i mogu primiti zadati broj gostiju. Pored osnovnih informacija, za svaku nekretninu treba da se prikaže ukupna cena smeštaja za sva noćenja, kao i jedinična cena (po osobi po noćenju ili po smeštaju po noćenju).

1.8 Zahtevi za rezervaciju smeštaja (uloge: G)

Svaki zahtev mora da sadrži smeštaj, početni i krajnji datum rezervacije i broj gostiju. Kada se zahtev potvrdi od strane vlasnika smeštaja, kreira se rezervacija i niko drugi ne može da šalje zahtev za isti smeštaj u istom vremenskom intervalu. Za neki smeštaj može postojati više zahteva u preklapajućim vremenskim intervalima koji čekaju odgovor. Guest koji je kreirao zahtev za rezervaciju može ga obrisati pre nego što dobije odgovor.

1.9 Otkazivanje rezervacije (uloge: G)

Guest može otkazati bilo koju rezervaciju koja mu je prihvaćena, do dana pre njenog početka. U tom slučaju smeštaj postaje ponovo slobodan za te datume.

1.10 Potvrda rezervacije (uloge: H)

Host za neki smeštaj može odabrati automatsku ili ručnu potvrdu rezervacije. Ako je aktivirana automatska potvrda, svaki zahtev se prihvata čim pristigne.

Ako je aktivna ručna potvrda, host može da prihvati ili odbije zahtev. Uz svaki zahtev mu se za guest-a prikazuje koliko puta u prošlosti je otkazao neku rezervaciju na platformi. Ako se zahtev potvrdi, svi ostali zahtevi čiji datumi se preklapaju sa prihvaćenim zahtevom treba da budu automatski odbijeni.

1.11 Ocenjivanje host-ova (uloge: G)

Guest može oceniti host-a ocenom od 1 do 5 samo ako je u prošlosti imao bar jednu rezervaciju u nekom njegovom smeštaju koju nije otkazao. Ocena se može promeniti ili ukloniti. Moguće je videti svaku pojedinačnu ocenu (ko ju je dao i kada), kao i srednju ocenu.

1.12 Ocenjivanje smeštaja (uloge: G)

Guest može oceniti smeštaj ocenom od 1 do 5 samo ako je tu odseo barem jednom u prošlosti. Ocena se može promeniti ili ukloniti. Moguće je videti svaku pojedinačnu ocenu (ko ju je dao i kada), kao i srednju ocenu.

1.13 Notifikacije (uloge: H, G)

Host treba da dobije notifikacije kada:

- Neko kreira zahtev za rezervaciju
- Neko otkaže rezervaciju
- Neko ga oceni
- Neko oceni njegov smeštaj

Guest treba da dobije notifikacije kada:

- Host odgovori na zahtev za rezervaciju

Svaki korisnik može uključiti ili isključiti notifikacije na svom profilu, pojedinačno za svaki tip notifikacije.

Nefunkcionalni zahtevi

1.14 Tracing

Neophodno je implementirati tracing nad svim servisima mikroservisne aplikacije i prikazati u nekom alatu za vizualizaciju.

1.15 Logging

Neophodno je implementirati agregaciju logova nad svim servisima mikroservisne aplikacije i prikazati u nekom alatu za vizualizaciju.

1.16 Metrike

- Metrike operativnog sistema host mašine na kojoj će mikroservisna aplikacija biti podignuta. Minimum treba obezbediti informacije o iskorišćenju procesora, RAM memorije, file sistema i protok mrežnog saobraćaja.

- Metrike kontejnera koji se koriste u mikroservisnoj aplikaciji. Minimum treba obezbediti informacije o iskorišćenju procesora, RAM memorije, file sistema i protok mrežnog saobraćaja.
- Metrike Web saobraćaja u mikroservisnoj aplikaciji:
 - Ukupan broj HTTP zahteva u prethodnih 24 sata.
 - Broj uspešnih HTTP zahteva u prethodnih 24 sata (2xx, 3xx).
 - Broj neuspešnih zahteva u prethodnih 24 sata (4xx, 5xx).
 - Broj jedinstvenih posetilaca (ista IP adresa, timestamp i web browser).
 - Broj neuspešnih zahteva sa statusom 404 sa njihovim endpoint-ovima u prethodnih 24 sata.
 - Ukupan protok saobraćaja izražen u GB.
- Metrike je neophodno prikazati u nekom alatu za vizualizaciju.

DevOps zahtevi

1.17 Konfiguracija Git repozitorijuma

- Koristiti [Feature Branch Workflow](#).
- PR (Pull Request) mora da se odobri od barem jednog člana tima.
- Podesiti pipeline koji će se okidati prilikom kreiranja svakog PR-a i njegovih izmena. Pipeline treba da build-uje aplikaciju, pokrene unit i integracione testove. Podesiti opciju da u slučaju da pipeline ne prolazi, PR se ne može integrisati.
- Uključiti linearnu istoriju (linear history) master i develop grana.
- Koristiti [Conventional Commits](#) za sadržaj commit poruka.
- Neophodno je da svaki mikroservis ima svoj repozitorijum.

1.18 Konfiguracija CI/CD pipeline-ova

Za svaki repozitorijum neophodno je napraviti CI pipeline koji će se okidati prilikom novih promena na develop i master grani. CI pipeline treba minimalno da obuhvati:

- Build aplikacije, pokretanje unit i integracionih testova. Za kreiranje neophodne infrastrukture (baza, message brokera itd.) za izvršavanje integracionih testova koristiti [Testcontainers](#) ili docker compose.
- Analiza koda pomoću SonarCloud-a ili nekog drugog alata (dovoljno na jednom mikroservisu).
- Kreiranje slike kontejnera i publish-ovanje na DockerHub ili neki drugi container registry. Koristiti [Semantic Versioning](#) prilikom verzionisanja kontejnera.
- Ubrzati pipeline keširanjem dependency-a ili korišćenjem build slike kontejnera u kojoj su zapakovani neophodni alati i dependency.

Pravila polaganja

- Projekat se radi u timovima do 4 člana.
- Projekat možete implementirati u bilo kom programskom jeziku i radnom okviru. Ako se odlučite za tehnologiju koja nije pokrivena na vežbama, pomoć u tom slučaju je ograničena.
- Projekat nosi ukupno 50 bodova. Projekat se smatra položenim ako ste osvojili 15 ili više bodova.
- Pored termina u julu, odbarana projekta će se održati u septembru i u januaru ili februaru.
- Za sve slučajeve koji nisu pokriveni u specifikaciji, studentima se daje mogućnost da ih reše na način koji je njima najprikladniji.

Bodovanje projekta

Odbrana - 50 bodova

- **Prvi skup zahteva - 15 bodova (za ocenu 6)**
 - Potrebno je implementirati zahteve: 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 1.10, 1.17, 1.18.
 - Neophodno je koristiti NoSQL bazu na nekom od servisa.
 - Svi servisi treba da se pokrenu u docker kontejnerima pomoću docker compose alata.
- **Drugi skup zahteva (A) - 25 bodova (9 maksimalna ocena)**
 - Potrebno je implementirati zahteve: 1.11, 1.12, 1.13, 1.14, 1.15, 1.16.
 - Podici infrastrukturu pomocu Docker Swarm platforme.
 - Korisiti [Docker Secrets](#) i [Docker Configs](#).
- **Drugi skup zahteva (B) - 35 bodova (10 maksimalna ocena)**
 - Potrebno je implementirati zahteve: 1.11, 1.12, 1.13, 1.14, 1.15, 1.16.
 - Podici infrastrukturu na neki Kubernetes kluster (npr. [minikube](#)).
 - Koristiti [kaniko](#) za kreiranje slika kontejnera.
 - Koristi [helm](#) za deployment infrastrukture.
- **Napomena:** Kako bi neki skup funkcionalnosti bio pregledan, potrebno je da odradite sve zahteve iz prethodnog skupa
- Primer mikroservisne arhitekture: [System Design Architecutre for Hotel Booking Apps](#). Niste u obavezi da implementirate ovakvu arhitekturu, već služi kao pomoć prilikom dizajna.