

Predmetni projekat XML i veb servisi

Školska godina 2022/2023.

Kroz predmetni projekat treba da implementirate platformu za ponudu i rezervisanje smeštaja, kao i aplikaciju kojom aviokompanija korisnicima omogućava kupovinu karata na njihovim letovima.

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)

3

spika

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)

10

gravara

1.3 Upravljanje nalogom (uloge: H, G)

spika

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

2

1.4 Brisanje naloga (uloge: H, G)

spika

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)

3



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)

7

Potrebno je omogućiti definisanje i izmenu termina kada je neki smeštaj dostupan. On se može rezervirati 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)

3

7

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)

3



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)

3

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)

4



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

Postavljanje front aplikacije
Postavljanje servisa na docker
Postavljanje bek aplikacija

2
2
2
gravara

spika
front

vukoman i kalina

kalina

spika front

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 Istaknuti host (uloge: H)



Istaknuti host je status koji host može steći ako ispunjava sledeće uslove:

- Ima ocenu veću od 4.7
- Ima stopu otkazivanja manju od 5%
- Imao je barem 5 rezervacija za smeštaje u prošlosti
- Ukupno trajanje svih rezervacija je veće od 50 dana

Na profilu host-a se prikazuje ovaj status ukoliko ga ima. Status nije trajan i uklanja se čim neki od navedenih uslova više nije ispunjen.

1.14 Filtriranje smeštaja (uloge: NK, H, G)



Rezultati pretrage smeštaja treba da se filtriraju prema sledećim kriterijumima:

- Opseg cene
- Pogodnosti koje smeštaj mora da ima
- Smeštaj mora da pripada istaknutom host-u

1.15 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
- Dobije ili izgubi status istaknutog host-a

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.



1.16 Preporuka i rezervacija letova (uloge: G)

Guest-u se za svaku rezervaciju prikazuje lista letova koje nudi aviokompanija. Letovi treba da se prikažu za dan početka rezervacije i za dan kraja rezervacije. Za prvi let korisnik mora da unese mesto polaska, a za drugi let mesto povratka. Ako korisnik ima nalog na sajtu aviokompanije, on može kupiti karte direktno sa platforme za rezervaciju smeštaja.

1.17 Preporuka smeštaja (uloge: G)

Korisniku treba prikazati preporuku smeštaja na osnovu njegovih prethodnih aktivnosti. Primer algoritma za preporuku:

- Pronađi slične korisnike. Sličan korisnik je neko ko je rezervisao isti smeštaj i davao slične ocene u prošlosti.
- Pronađi smeštaj koji su rezervisali slični korisnici i dali mu dobru ocenu
- Odbaci smeštaj koji ima više od 5 ocena manjih od 3 u prethodna tri meseca
- Rangiraj smeštaj prema ukupnoj oceni i preporuči prvih 10 rezultata

Parametre algoritma ili sam algoritam je moguće menjati, sve dok on ima sličnu ili veću kompleksnost. Za implementaciju preporuke neophodno je koristiti graf bazu.

Nefunkcionalni zahtevi

1.18 Tracing

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

1.19 Logging

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

1.20 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.
- cMetrice 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 neuspješnih 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.

Aplikacija za kupovinu avionskih karata

Arhitektura

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

Uloge u sistemu

- **Neautentifikovani korisnik (NK)** - Može da kreira novi nalog ili se prijavi na postojeći. Pored toga, može da pretražuje dostupne letove, ali ne može da kupuje karte.
- **Običan korisnik (OK)** - Kupuje karte za letove na kojima ima dostupnih mesta.
- **Administrator (A)** - Dodaje nove letove. Za svaki definiše maksimalan broj mesta, datum, vreme i mesto poletanja i sletanja.

Funkcionalni zahtevi

2.1 Registracija naloga (uloge: NK)

2.2 Prijava na sistem (uloge: NK)

2.3 Upravljanje letovima (uloge: A)

Svaki let treba da sadrži datum, vreme i mesto poletanja i sletanja, kao i cenu karte po putniku. Administrator može da kreira i briše letove. Pored toga, za svaki let može da vidi osnovne informacije i preostali broj karata.

2.4 Pretraga letova (uloge: NK, OK)

Letovi se pretražuju prema datumu, mestu sletanja i poletanja i broju putnika. Rezultati treba da sadrže samo letove koji su zakazani za traženi datum i imaju odgovarajuća mesta sletanja i poletanja, kao i dovoljan broj slobodnih mesta. Za svaki let treba prikazati cenu karte po putniku, kao i ukupnu cenu za sve putnike.

2.5 Kupovina avionskih karata (uloge: OK)

Za let se može kupiti jedna ili više karata, ako je taj broj karata dostupan. Korisnik može da pogleda informacije o kartama koje je kupio.

2.6 Upravljanje API ključevima (uloge: OK)

Zaštićeni endpoint-i (na primer endpoint za kupovinu karata, pregled kupljenih karata itd) moraju da autorizuju svaki pristigli zahtev, što znači da pre toga nekako moraju autentifikovati korisnika koji ga šalje. API ključ je token preko kog se korisnici mogu autentifikovati. Svako ko poseduje validan API ključ nekog korisnika može slati zahteve u njegovo ime. Potrebno je omogućiti korisnicima da kreiraju API ključeve za svoje naloge. API ključ može imati vek trajanja, a može biti i trajan, u zavisnosti od toga koju opciju korisnik odabere. API ključ treba da omogući kupovinu karata u ime nekog korisnika, ali sistem ne sme da dozvoli da se bilo koja druga akcija izvrši pomoću API ključa, na primer pregled svih kupljenih karata.

Pravila polaganja

- Projekat se radi u timovima od 4 člana. Članovi tima ne moraju slušati vežbe u istim grupama.
- Projekat možete implementirati u bilo kom programskom jeziku i radnom okviru. Ako se odlučite za tehnologiju koja nije pokrivena na vežbama, pomoć sa naše strane u tom slučaju je ograničena.
- Projekat nosi ukupno 70 bodova i polaže se kroz dve kontrolne tačke u toku semestra i finalnu odbranu u ispitnom roku. Projekat se smatra položenim ako ste osvojili 30 ili više bodova. **Za svaku kontrolnu tačku na koju ne izađete gubite po 5 bodova koje ne možete nadoknaditi.**
- Pored termina u julu, odbarana projekta će se održati u septembru i u januaru ili februaru. **U septembru se bodovi neće skalirati, dok se u januaru/februaru ocena smanjuje za jednu.**
- Pitanja vezana za specifikaciju projekta i način polaganja predmeta možete postaviti u ovoj [formi](#). Odgovore možete naći na sledećem [linku](#).
- Za sve slučajeve koji nisu pokriveni u specifikaciji, studentima se daje mogućnost da ih reše na način koji je njima najprikladniji.

Bodovanje projekta

Kontrolna tačka 1 - 15 bodova

- Potrebno je implementirati zahteve: 2.1, 2.2, 2.3, 2.4 i 2.5
- Podatke je potrebno čuvati u dokument bazi
- Svi servisi treba da se pokrenu u docker kontejnerima pomoću docker compose alata

Kontrolna tačka 2 - 25 bodova

- Treba da nacrtate dijagram arhitekture mikroservisne aplikacije, definišete modele podataka svakog servisa i odlučite u kojim bazama podataka servisi skladište podatke

- Potrebno je implementirati zahteve: 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 1.10
- Servisi treba da koriste gRPC za sinhronu komunikaciju, dok klijentima aplikacije treba obezbediti REST API

Finalna odbrana - 30 bodova

- **Prvi skup zahteva - 10 bodova**
 - Potrebno je implementirati zahteve: 1.11, 1.12, 1.13, 1.14
 - Svi servisi treba da se pokrenu u docker kontejnerima pomoću docker compose alata
 - Implementirati SAGA obrazac sa barem tri koraka u slučaju uspešno izvršene transakcije i pokrivene sve neuspešne tokove
- **Drugi skup zahteva - 10 bodova**
 - Potrebno je implementirati zahteve: 1.15, 1.16, 1.17, 2.6
- **Treći skup zahteva - 10 bodova**
 - Potrebno je implementirati zahteve: 1.18, 1.19, 1.20
- **Napomena:** Kako bi neki skup funkcionalnosti bio pregledan, potrebno je da odradite sve zahteve iz prethodnog skupa