

Školska godina *2023 /2024*

PREDMET*:***IT355 - Veb sistemi 2**

Projektni zadatak

Ime i prezime: **Mladen Stolic**

      Broj indeksa:       **5319**

                                    Datum izrade:     **09.05.2024.**

Tehničke specifikacije

# Pregled projekta

Ovaj sistem je dizajniran da pruži sveobuhvatnu online platformu za upravljanje i prezentaciju informacija vezanih za putovanja, uključujući države, gradove i različite smeštaje. Cilj je da se korisnicima, kako administratorima tako i posetiocima, omogući jednostavan interfejs za prikazivanje, dodavanje, uređivanje i brisanje podataka o putovanjima.

# Komponente sistema

1. Frontend**:**
   * Framework: Angular
   * Funkcionalnosti:
     + Autentifikacija korisnika (prijava/odjava) jwt
     + Navigacioni meni
     + Administratorski deo dodat na deo korisnika
     + Javne stranice za prikaz destinacija za putovanja i smeštaja
2. Backend:
   * Implemntirano: Spring Boot, Spring Security za autentifikaciju i autorizaciju, modelski nivo urađen u Javi, nivo kontrolera urađen u Javi, odvojen sloj biznisa, modela i pregleda
   * Baza podataka: MySQL
   * Funkcionalnosti:
     + RESTful API endpointi za upravljanje podacima (CRUD operacije)
     + Autentifikacija i autorizacija za administrativne funkcionalnosti
     + JWT (JSON Web Tokens) za siguran prenos informacija
     + Mogućnosti upravljanja administratorima
3. Baza podataka:
   * Entiteti:
     + Accommodation: id, City, name, description, featured, img\_urk, price\_from, number\_of\_nights, available, priceListImageUrl, not\_included
     + Accommodationphoto: Accommodationphoto, imageUrl
     + Admin: id, username, password
     + City: id, Country, name, opisGrada, slikaGradaURL
     + Country: id, name, imageUrl
     + Travelday: Accommodation, dayNumber, tittle, description
4. Sigurnost:
   * Sigurna autentifikacija i autorizacija koristeći JWT
   * HTTPS za sigurnu komunikaciju između klijenta i servera(u toku)
   * Heširanje lozinki u bazi podataka za sigurnost

# Tehničke specifikacije

1. Frontend:
   * Angular CLI za kreiranje projekta
   * Komponente za svaku glavnu funkcionalnost (prijava, admin panel, lista država, detalji smeštaja…)
   * Servisi za upravljanje HTTP zahtevima ka backendu
   * Zaštita ruta za upravljanje pristupom na osnovu statusa autentifikacije
   * Reaktivne forme za upravljanje korisničkim unosom
   * Modularni dizajn koristeći Angular module za logičku organizaciju koda
2. Backend:
   * Spring Boot kao okvir za aplikaciju
   * Spring Data JPA za interakciju s bazom podataka
   * Spring Security za upravljanje autentifikacijom i autorizacijom
   * Maven kao alat za izgradnju
   * Upravljanje izuzecima za upravljanje i vraćanje odgovarajućih odgovora o greškama
   * Dokumentacija API-ja
3. Baza podataka:
   * MySQL kao relaciona baza podataka
   * Korišćenje JPA entiteta za predstavljanje tabela
   * Sema baze podataka koja podržava relacije kao što su OneToMany za Države prema Gradovima
4. Razmeštanje:
   * Docker kontejneri za frontend i backend za lako razmeštanje(…)
   * Stalne integracije/stalno razmeštanje (CI/CD) (opciono zavisno od vašeg setupa)(…)
5. Mere sigurnosti:
   * JWT za upravljanje sesijama i rukovanje stanjem autentifikacije korisnika između zahteva
   * Spring Security konfiguracije za ograničavanje pristupa osetljivim endpointima
   * Korišćenje HTTPS-a za šifrovanje podataka u tranzitu(u toku)

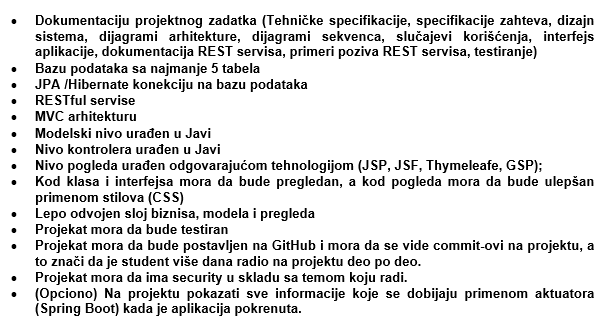
# Razvojni alati:

* IDE: Visual Studio Code za frontend, IntelliJ IDEA za backend
* Kontrola verzija: Git
* Testiranje: JUnit za testiranje backend-a

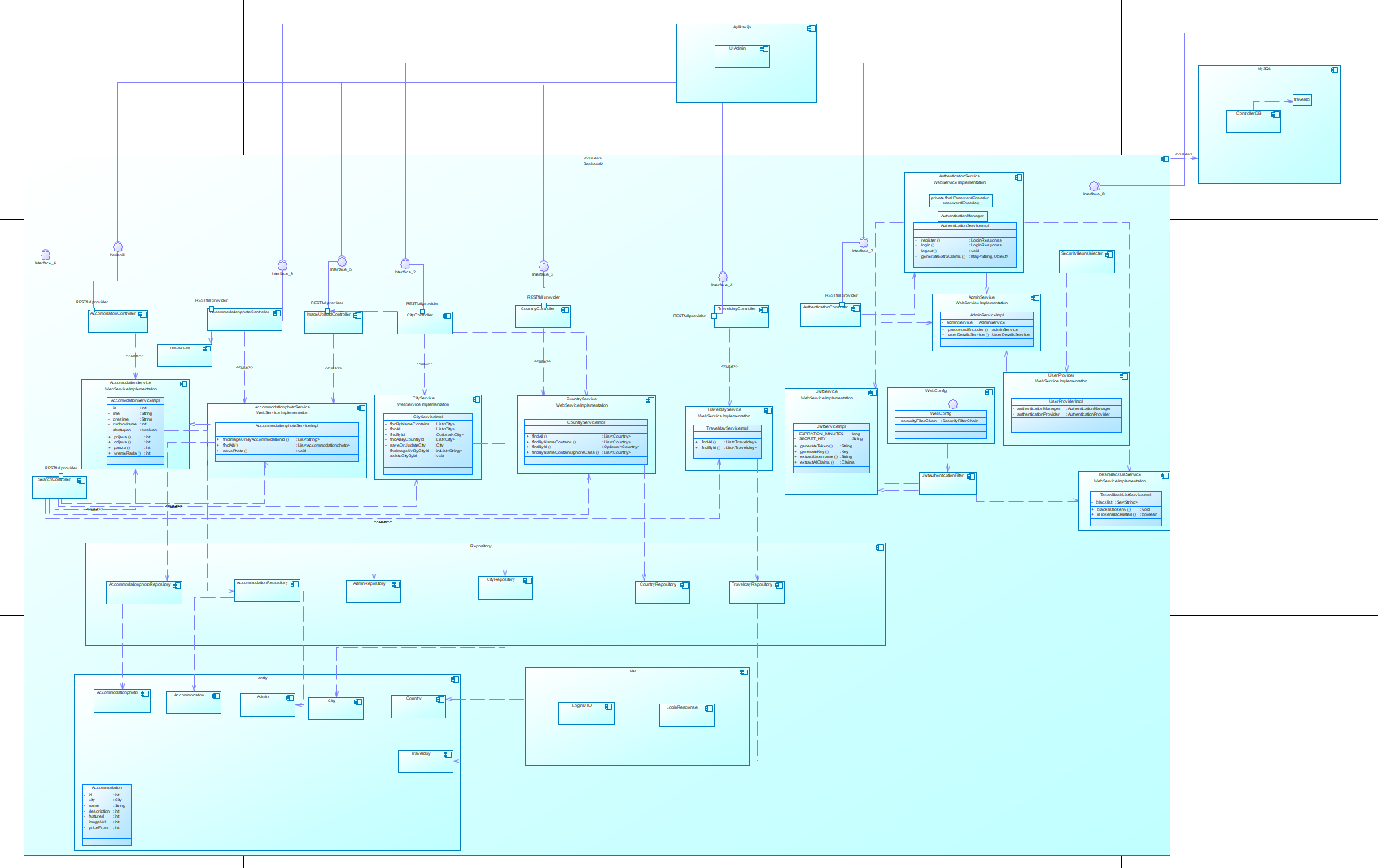
# Usklađenost i standardi

* Obezbediti usklađenost sa GDPR-om, ako je primenjivo
* Pratiti standarde dizajna REST API-a

Specifikacije zahteva

****

Dijagrami arhitekture

****

Dijagrami sekvenca

**1. Dijagram sekvence za korisnički login**

Ovaj dijagram prikazuje tok zahteva za login korisnika kroz sistem:

* **Korisnik**: Inicira proces logovanja.
* **Login komponenta (Frontend)**: Prima unos od korisnika.
* **Kontroler autentifikacije (Backend)**: Obrađuje zahteve za autentifikaciju.
* **Servis za autentifikaciju (Backend)**: Rukuje logikom za autentifikaciju korisnika.
* **Repozitorijum admina (Backend)**: Radi s bazom podataka radi dohvatanja podataka o korisnicima.
* **Baza podataka (MySQL)**: Čuva korisničke podatke i uloge.

Tok:

1. Korisnik unosi svoje akreditive na stranici za login i šalje ih.
2. Login komponenta šalje te akreditive Kontroleru autentifikacije.
3. Kontroler autentifikacije poziva Servis za autentifikaciju da verifikuje akreditive.
4. Servis za autentifikaciju koristi Repozitorijum admina da izvuče detalje korisnika iz Baze podataka.
5. Baza podataka vraća detalje korisnika Repozitorijumu admina.
6. Servis za autentifikaciju validira akreditive. Ako je uspešno, generiše JWT token.
7. Kontroler autentifikacije vraća JWT token Login komponenti.
8. Korisniku se prikazuje greška ili se preusmerava na osnovu odgovora.

**2. Dijagram sekvence za dohvatanje detalja o smeštaju**

Ovaj dijagram ilustruje proces dobijanja detaljnih informacija o smeštaju. Aktei i komponente:

* **Korisnik**: Zahteva da vidi detalje smeštaja.
* **Komponenta za smeštaj (Frontend)**: Prikazuje detalje o smeštaju.
* **Kontroler za smeštaj (Backend)**: Rukuje zahtevima za podacima o smeštaju.
* **Servis za smeštaj (Backend)**: Procesuira poslovnu logiku za dohvatanje podataka o smeštaju.
* **Repozitorijum za smeštaj (Backend)**: Dohvata podatke o smeštaju iz baze podataka.
* **Baza podataka (MySQL)**: Sadrži podatke o smeštaju.

Tok:

1. Korisnik izabira smeštaj za koji želi da vidi detalje na stranici sa listom smeštaja.
2. Komponenta za smeštaj šalje zahtev Kontroleru za smeštaj s ID-jem smeštaja.
3. Kontroler za smeštaj prosleđuje ID Servisu za smeštaj.
4. Servis za smeštaj upituje Repozitorijum za smeštaj da izvuče podatke.
5. Repozitorijum za smeštaj dohvata podatke iz Baze podataka i vraća ih Servisu za smeštaj.
6. Servis za smeštaj obrađuje podatke (po potrebi) i vraća ih Kontroleru za smeštaj.
7. Kontroler za smeštaj šalje detaljne podatke nazad Komponenti za smeštaj, koja ih prikazuje Korisniku.

Slučajevi korišćenja

**1. Autentifikacija Korisnika**

**Primarni Akter: Korisnik Cilj: Prijavljivanje u sistem kako bi se pristupilo zaštićenim resursima. Preduslovi: Korisnik mora biti registrovan u sistemu. Glavni Tok Dogadjaja:**

1. **Korisnik unosi korisničko ime i lozinku.**
2. **Sistem validira unete podatke.**
3. **Ako su podaci validni, sistem generiše i vraća JWT token korisniku.**
4. **Korisnik dobija pristup zaštićenim resursima. Alternativni Tok Dogadjaja:**

* **Ako podaci nisu validni, korisniku se prikazuje poruka o grešci.**

**2. Pregled Država**

**Primarni Akter: Posetilac Cilj: Pregledanje liste dostupnih država na platformi. Preduslovi: Nema. Glavni Tok Dogadjaja:**

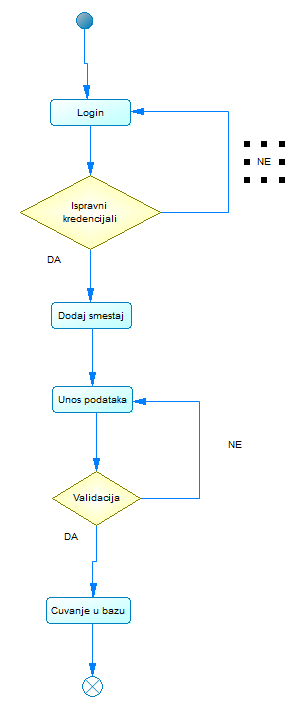
1. **Posetilac bira opciju za pregled država.**
2. **Sistem dohvata listu država iz baze podataka.**
3. **Sistem prikazuje listu država posetiocu. Alternativni Tok Dogadjaja: Nema.**

**3. Dodavanje Novog Smeštaja**

**Primarni Akter: Administrator Cilj: Dodavanje novog smeštaja u sistem. Preduslovi: Korisnik mora biti ulogovan kao administrator. Glavni Tok Dogadjaja:**

1. **Administrator unosi detalje o smeštaju uključujući ime, opis, cenu, broj noći i slike.**
2. **Sistem validira unete podatke.**
3. **Sistem sačuva smeštaj u bazu podataka.**
4. **Administrator dobija potvrdu o uspešno dodatom smeštaju. Alternativni Tok Dogadjaja:**

* **Ako podaci nisu validni ili nedostaje neki od podataka, administratoru se prikazuje poruka o grešci.**

****

**4. Upravljanje Profilom**

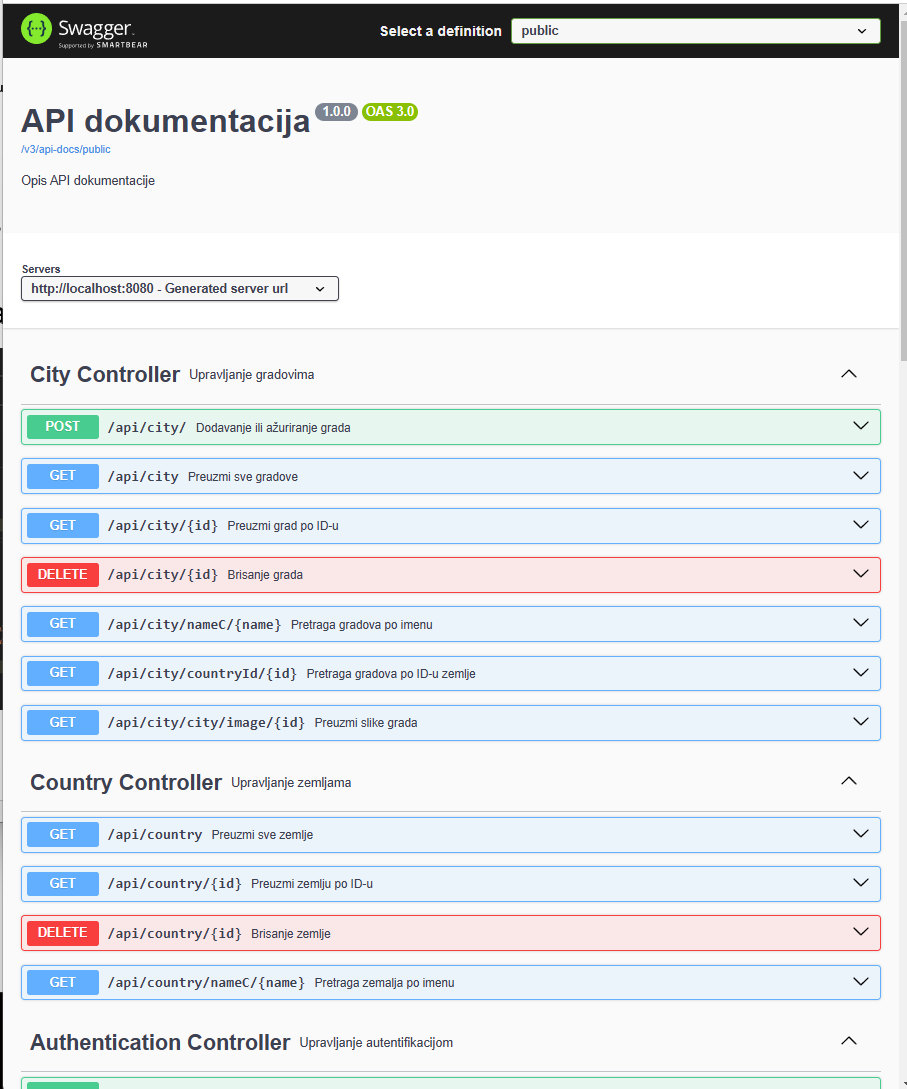
**Primarni Akter: Admin Cilj: Upravljanje sopstvenim profilnim informacijama. Preduslovi: Admin mora biti ulogovan. Glavni Tok Dogadjaja:**

1. **Admin pristupa svom profilu.**
2. **Admin menja željene informacije.**
3. **Sistem validira i ažurira informacije.**
4. **Admin dobija potvrdu o ažuriranim informacijama. Alternativni Tok Dogadjaja:**

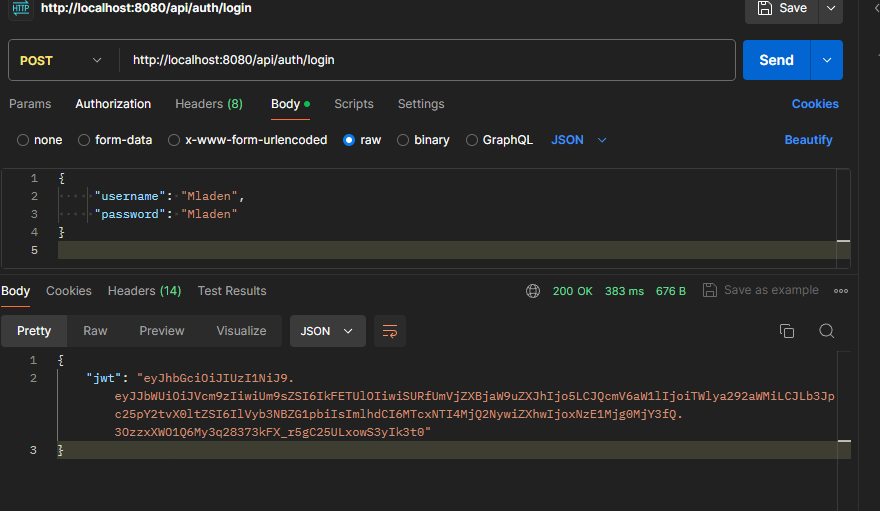
* **Ako su informacije nevalidne, korisniku se prikazuje poruka o grešci**

Dokumentacija REST servisa

Na admin panelu postoji link koji vodi do swagger dokumentacjie.

****

Primeri poziva REST servisa

****

Testiranje

Koristili smo Mockito za testiranje servisa.

