

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)

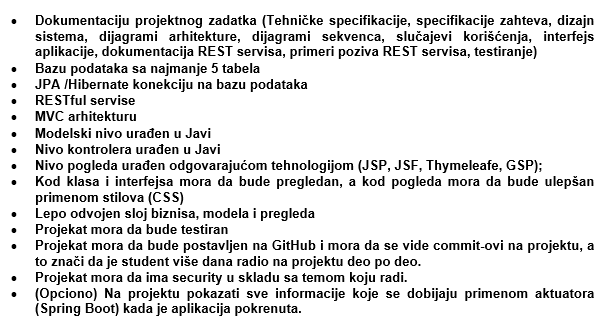
# Razvojni alati:

* 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

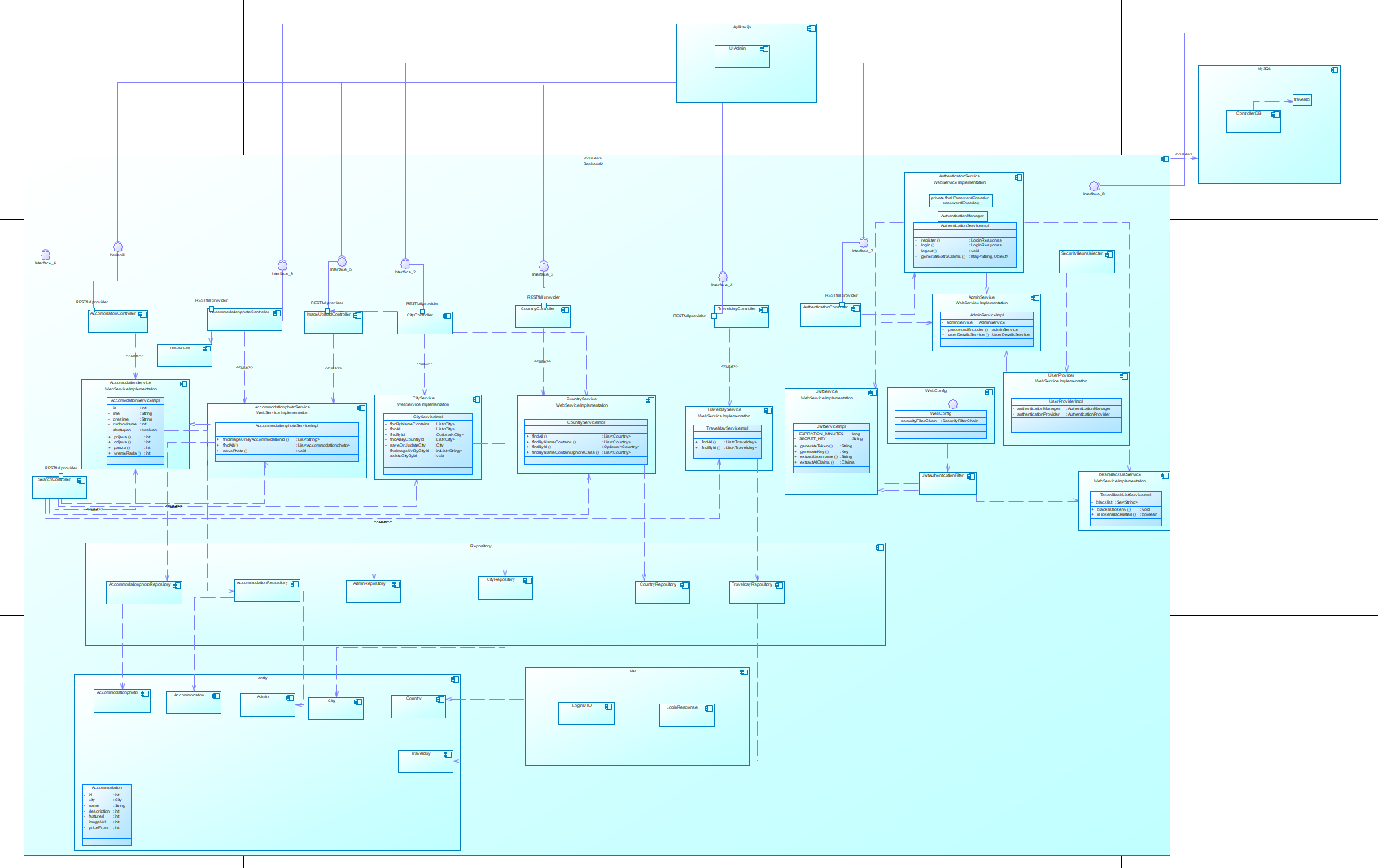
Specifikacije zahteva

****

Dizajn sistema

/

Dijagrami arhitekture

****

Dijagrami sekvenca

**1. User Login Sequence Diagram**

**This diagram will show how the user's login request flows through your system.**

**Actors and Components:**

* **User: Initiates the login process.**
* **Login Component (Frontend): Accepts input from the user.**
* **Authentication Controller (Backend): Processes authentication requests.**
* **Authentication Service (Backend): Handles the logic to authenticate users.**
* **Admin Repository (Backend): Interacts with the database to retrieve user data.**
* **Database (MySQL): Stores user credentials and roles.**

**Flow:**

1. **User enters their credentials on the Login Page and submits them.**
2. **Login Component sends these credentials to the Authentication Controller.**
3. **Authentication Controller calls the Authentication Service to verify the credentials.**
4. **Authentication Service uses the Admin Repository to fetch user details from the Database.**
5. **Database returns user details to the Admin Repository.**
6. **Authentication Service validates credentials. If successful, it generates a JWT token.**
7. **Authentication Controller sends the JWT token back to the Login Component.**
8. **User is either shown an error or redirected based on the response.**

**2. Fetching Accommodation Details Sequence Diagram**

**This diagram illustrates the process of retrieving detailed information about an accommodation.**

**Actors and Components:**

* **User: Requests to view accommodation details.**
* **Accommodation Component (Frontend): Displays accommodation details.**
* **Accommodation Controller (Backend): Handles accommodation data requests.**
* **Accommodation Service (Backend): Processes business logic for accommodation data retrieval.**
* **Accommodation Repository (Backend): Retrieves accommodation data from the database.**
* **Database (MySQL): Holds accommodation data.**

**Flow:**

1. **User selects an accommodation to view details from the Accommodation List Page.**
2. **Accommodation Component sends a request to the Accommodation Controller with the accommodation ID.**
3. **Accommodation Controller forwards the ID to the Accommodation Service.**
4. **Accommodation Service queries the Accommodation Repository to fetch the data.**
5. **Accommodation Repository retrieves data from the Database and returns it to the Accommodation Service.**
6. **Accommodation Service processes the data (if necessary) and returns it to the Accommodation Controller.**
7. **Accommodation Controller sends the detailed data back to the Accommodation Component, which displays it to the User.**

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. Rezervacija Smeštaja**

**Primarni Akter: Korisnik Cilj: Rezervisanje odabranog smeštaja. Preduslovi: Korisnik mora biti ulogovan. Glavni Tok Dogadjaja:**

1. **Korisnik bira smeštaj koji želi da rezerviše.**
2. **Korisnik unosi potrebne informacije za rezervaciju (broj osoba, datum).**
3. **Sistem obrađuje zahtev i rezerviše smeštaj.**
4. **Korisnik dobija potvrdu o rezervaciji. Alternativni Tok Dogadjaja:**

* **Ako nema dostupnih mesta, korisniku se prikazuje poruka da smeštaj nije dostupan.**

**5. Upravljanje Profilom**

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

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

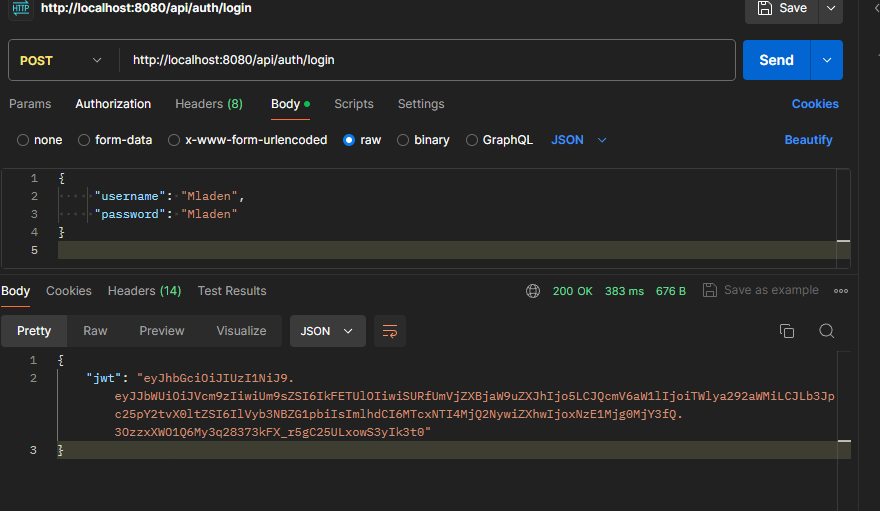
* **Ako su informacije nevalidne, korisniku se prikazuje poruka o grešc**

Interfejs aplikacije

Dokumentacija REST servisa

\*\* AccommodationphotoController \*\*  
GET  
 /api/accomodationphoto ->vrati sve  
 /images/{accommodationphotoId}" lista stringova slika po id smestaja  
  
  
\*\* AccomodationController \*\*  
GET  
 /api/accommodation sve  
 /api/accommodation/name/{name} Vraca listu po nazivu celom  
 /api/accommodation/nameC/{name} lista po nazivu koji zadrzi name u imenu  
 /api/accommodation/istaknutiOglasi lista istaknutih oglasa  
 /api/accommodation/cityId/{city\_id} lista svih smestaja u jednom gradu  
 /api/accommodation/{id} pretraga po id smestaja vraca listu ali sa jednim elementom Accommodation  
POST  
 /api/accommodation/ Dobija smestaj i ako je nov dodaje ako nije azurira postijeci  
DELETE  
 /api/accommodation/{id} dobije id i brise deleteAccommodationById  
PUT  
 /api/accommodation/{id}/image dobija url slike gde se cuva i tip 1 ili 2 da zna koju sliku da menja u kojoj koloni  
  
  
\*\* CityController \*\*  
GET  
 /api/city lista svega  
 /api/city/name/{name} lista po nazivu sadrzi  
 /api/city/{id} Vraca jedan grad po id  
 /api/city/countryId/{id} Vraca listu gradova po countryId  
 /api/city/image/{id} Vraca url slike po id grada  
 /api/city/images/{filename:.+ vraca direktno sliku iz folderaq po nazivu u bazi  
 /api/city  
 /api/city  
  
DELETE  
 /api/city/{id} Brise po id  
POST  
 /api/city/ dodaje grad i poziva metodu za cuvanje slike  
 @RequestParam("name") String name,  
 @RequestParam("countryId") int countryId,  
 @RequestParam("opisGrada") String opisGrada,  
 @RequestParam("image") MultipartFile image) {  
  
\*\* CountryController \*\*  
GET  
 /api/country lista sve  
 /api/country/nameC/{name} lista zemalja sadrzi u nazivu  
 /api/country/{id} lista po id gradova  
  
\*\* TraveldayController \*\*  
GET  
 /api/travelday sve lista  
 /api/travelday/{id} vraca listu po id  
  
\*\* SearchController \*\*  
GET  
 /api/search/city/{name} lista po nazivu sadrzi limit 4  
 /api/search/accommodation/{name} \*\*  
 /api/search/country/{name} \*\*  
  
\*\* AuthenticationController \*\*  
POST  
 /login Prima loginDTO poziva servis za autentifikaciju i vraca odgovor  
 /register Dobija admina i vraca odgovor servisa  
 /logout ocekuje header Authorization gde se nalazi Bearer token i dodaje ga na blacklistu  
  
 \*\* ImageUploadController \*\*  
 POST  
 /upload postavlja vise slika kroz petlju i upisuje u bazu naziv slike  
 /single postavlja jednu sliku i vraca naziv fajla

Primeri poziva REST servisa

****

Testiranje