Advanced Intelligent Tourist Guide

Nowadays people use mobile phones and other mobile devices. Most of us have a small computing device that is always with us. People use it example for calling, as calendar and organizer. Mobile devices with GPS receiver are also used to find paths in navigation.

The main idea of this thesis was to design a system that will run on most of phones and palms and will be helpful when visiting some new places and cities. This system should be able to find a route using user criteria.

Those criteria should be simple and natural, like for example: a list of museums, the most famous historical objects, restaurants to visit, constraints to travel by bus and by walking. The system should find a path that fulfils those criteria, show it on screen, show names of objects, some short descriptions and photos of them and possible entrance costs.

It should also be able to estimate time needed to travel from one object to the next and if it is possible, advise which bus line or other public means of transport may be used. It should be helpful for people that want to visit a city without having much information about it. Paths that are output of this system are only a proposition for trip.

Modules:

The system comprises of 2 major modules with their sub-modules as follows:

1. **Admin**
    1. **Login:** Admin can login his personal account.
    2. **Add Places:** Admin can add places.
        1. Name: Name of a place.
        2. Images: Admin can add images of the particular places.
        3. Address: Admin can add address of that place.

        4. Area: Proper area of a place.
        5. Distance: Add distance from current location.
        6. Description: Add description of that place.
    3. **View/Edit Places:** Admin Can view tags and description of that place.
    4. **View User:** Admin can see the user details.
    5. **View Feedback:** Admin can see users Feedback.
    6. **Logout:** Admin can logout from his account.
2. **User**
    1. **Register:** User can register his personal detail.
    2. **Login:** User can login his/her account.
    3. **Tour Plan:** User will be able to view the places according to preferences. He/she need to select two between dates and the plan will be displayed.
    4. **Feedback:** User can send his/her feedback.
    5. **Logout:** User can logout from his account.

Reference Link:

https://youtu.be/1ZjHGhKAjxk?si=QAR1LfA08x67y1V_

Software Requirements:

- Windows 7 or higher.
- Java 17
- Spring Boot Version 3.2.0
- Swagger 2 or above
- MySQL8.5
- Visual studio
- Eclipse
- Lombok

Required implementation:

- Spring Security using with JWT

- Implement Loggers (slf4j) in the service class, controller class and utility classes and store the loggers in the .log file
- Implementation of Frontend using Angular 17 version.
- Application as Spring Restful Service
- Implement microservice architecture with proper intercommunication.
- Implement Custom Exception and Global Exception Handler
- Implement spring security with JWT
- Implement a Response Entity to wrap up the data with HTTP status code.
- Implement Swagger 2 and Above
- Include Junit 5 and Mockito test cases.
- JUnit Test Code coverage through test cases
- Spring and JPA Integration- Application development using JPA with Hibernate
- PMD - code quality level check

Advantages:

- Registered user gets the recommendation of the places of their preferences.
- They can find the places using this system.
- User can easily view the place on map with its description, image and address.
- The system also provides one food place in the results.
- System GUI is easy to use.
- The system in robust and secure in nature.

Disadvantages:

- User cannot book a tour package using this system.
- It may provide inaccurate results if data entered incorrectly.

Application:

This system can be used in any travel agency system. In addition, it can be implemented in government tourism to view the places.

Dependencies

- Spring Web
- Spring Data Jpa
- OpenFeign
- Eureka Discovery Client
- Eureka server (service registery)
- API GateWay
- MySQL
- Lombok
- Spring Security
- JWT
- Validations
- Actuators

MicroService Architecture

Admin Micro Service

- adminId
- adminUserName
- password
- email
- mobileNumber
- roles

User Micro Service

- userId
- userName
- password
- email
- mobileNumber
- address
- roles

Places Micro Service

- placeId
- placeName
- image
- address (exact location)
- area (city)
- distance
- description
- tags (examples: beach, museum, hill station)

FeedBack Micro Service

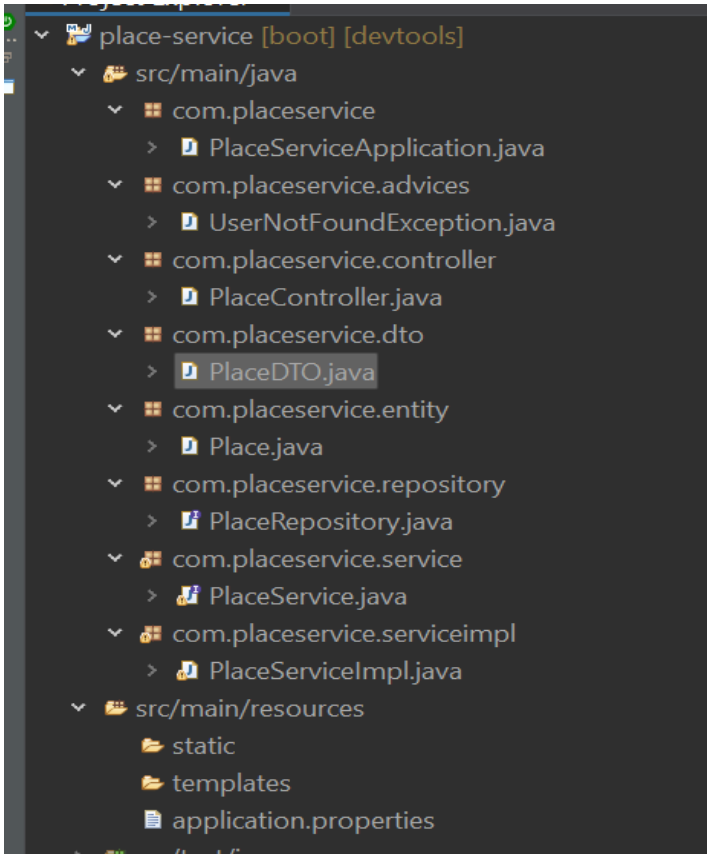
- feedbackId
- feedbackComment
- place
- user

TourPlan Micro Service

- tourId
- user
- List< > Places
- startDate
- endDate

Note: Provide Mappings. Make changes on required.

DTO’s should use. Not need of DAO instead you can use Repository which Implements JpaRepository



Should follow proper naming conventions.