Part B

Hands-on Problem

TcsCabs is an application for booking cabs. Its following functionalities need to be exposed as a REST API:

1. Book a cab
2. Get booking details
3. Cancel booking

1. Implement the CabBookingAPI class based on the instructions given below:

CabBookingAPI -> (com.tcs.api)

CabBookingAPI
- bookingService: BookingService
- environment: Environment

1. Annotate this class with proper annotation to the controller class.
2. Annotate this class with proper annotation so that all its methods are mapped with /bookings as base URI.
3. Inject bookingService and environment using appropriate annotation.

Method description:

bookCab(CabBookingDTO cabBookingDTO)

- This is a REST controller method to book a cab.
- Implement it using proper annotations according to the description given below:
    - Resource endpoint: /
    - HTTP method: POST

    - Input: Booking details as part of HTTP request body.
- It should invoke the bookCab() method of BookingServiceImpl class, which returns a booking id.
- Retrieve the success message associated with the property API.BOOKING_SUCCESSFUL from application.properties files using environment and append it to the booking id in the following format:
    - <success message>booking id
- It returns an object of ResponseEntity created using the above message and HTTP status code as CREATED.

getBookingDetails(Long mobileNo)

- This is a REST controller method to get cab booking details based on the mobile number of the user.
- Implement it using proper annotations according to the description given below:
    - Resource endpoint: /{mobileNo}
    - HTTP method: GET
    - Input: mobileNo as path variable.
- It should invoke the getBookingDetails() method of BookingServiceImpl class, which returns a List<CabBookingDTO>.
- It returns an object of ResponseEntity created using List<CabBookingDTO> obtained inthe previous step and HTTP status code as OK.

cancelBooking(Integer bookingId)

- This is a REST controller method to cancel a cab booking based on bookingId.
- Implement it using proper annotations according to the description given below:
    - Resource endpoint: /{bookingId}
    - HTTP method: PUT
    - Input: bookingId as path variable.
- It should invoke cancelBooking() method of BookingServiceImpl class.

- Retrieve the message associated with the property "API.BOOKING_CANCELLED" from the properties file.
- It returns an object of ResponseEntity created using the above message and HTTP status code as OK.

DTO classes (com.tcs.dto)

CabBookingDTO
- bookingId: Integer
- source: String
- destination: String
- fare: Float
- travelDate: LocalDate
- userMobile: Long
- status: Character

Entity Classes (com.tcs.entity)

CabBooking
- bookingId: Integer
- source: String
- destination: String
- fare: Float
- travelDate: LocalDate
- userMobile: Long
- status: Character

FareEstimation
- fareId: Integer
- source: String
- destination: String
- fare: Float

Exception class (com.tcs.exception)

TcsCabException
- serialVersionUID: long (final, static)

BookingServiceImpl (com.tcs.service)

This is the service class of the application. Implement this class according to the class diagram and description given below:

UserServiceImpl
- bookingRepository: BookingRepository
- fareRepository: FareRepository

Instance variable description

- Inject bookingRepository and fareRepository using appropriate annotation.

Method description

bookCab(CabBookingDTO bookingDTO)

- Call the method findFareBySourceAndDestination() present in FareRepository and get the fare from the Fare table.
- If the Fare is coming as null throw TcsCabException
- Set the details in CabBooking entity and save it in db.

getBookingDetails(Long mobileNo)

- Get the List<CabBookingDTO> based on the phone no using which it has been booked and return it.

Repository (com.tcs.repository)

Based on the requirement, analyze and figure out and implement the required details.

LoggingAspect (com.tcs.utility)

This is an AOP aspect class. It contains advices which log exceptions thrown from service and repository classes.

LoggingAspect
- LOGGER: Log

Method:

logServiceException(Exception exception)

- Annotate this method with appropriate advice that gets executed when exceptions are thrown in ServiceImpl class.
- It should also log the thrown exception at ERROR level.

2. Use the Postman tool to test the API.

3. Develop a REST client using RestTemplate and consume the API.