**1)Customer class**

1. public int customerID(PK)

2. public string Name

3. public String address

4. public string phone\_number

Define a parameterized constructor in the Customer to initialize values to the above fields.

**Constructor:** Customer(int customerId,String name,String address,String phone\_number)

**Exception class:**

CustomerNotFoundException()

InvalidCustomerInputException()

**CustomerUtility class** with methods:

**public void createCustomer(Customer customer) throws InvalidCustomerInputExceptionpublic void AddCustomer(Customer obj)**

**public** Customer readCustomer(**int** customerId) **throws** CustomerNotFoundException

**public** **void** updateCustomer(Customer customer) **throws** CustomerNotFoundException, InvalidCustomerInputException

**public** **void** deleteCustomer(**int** customerId) **throws** CustomerNotFoundException

**2)MenuClass:**

Long restaurantId;(FK)

String name;

String description;

Double price;

**Parametrized constructor** : Menu(String name,String description,Double price)

**Exception Class:**

public class MenuItemNotFoundException extends OnlineFoodDeliveryException

public MenuItemUnavailableException(String itemName)

**Menu Utility class:**

We use hashmap to store store id as key along with menu object.

public Menu addMenuItem(Menu menu)

public Menu getMenuItem(Long restaurantId)

public Menu updateMenuItem(Long restaurantId, Menu updatedMenu)

public void deleteMenuItem(Long restaurantId)

public void checkMenuItemAvailability(Long restaurantId)

public Map<Long, Menu> getAllMenuItems()

**3) Restaurant class:**

private int restaturantId;

            private String name;

            private String location;

            private String cusine;

            private String contact;

            private float rating;

**Constructor :** Restaurant(int id,String name,String location,String cusine,String contact,float rating)

**Exception Class**

public class RestaurantClosedException extends OnlineFoodDeliveryException

public class RestaurantNotFoundException extends OnlineFoodDeliveryException

**RestaurantUtility Class**

The RestaurantUtility class will provide helper methods for managing the list of restaurants.

public static void addRestaurant(Restaurant restaurant)

public static void displayAllRestaurants()

public static Restaurant findRestaurantById(int id) throws RestaurantNotFoundException

public static void displayRestaurantDetails(Restaurant restaurant)

**4)Order class:**

Private int orderId;

Private int customerId;(FK)

Private int restaurantId;(FK)

Private String deliveryAddress:

Private String double totalPrice;

Private String String status;

**Constructor:** order(int orderId,String deliveryAddress,double totalPrice,String status)

* Both customerId,restaurantId are fields are inherited from customer,restaurant classes

**Exception class:**

This exception used to invalid order data.

public class OrderNotFoundException extends OnlineFoodDeliveryException

public class InvalidOrderStateException extends OnlineFoodDeliveryException

**OrderUtility class:**

public static Order findOrderById(int orderId, List<Order> orders) throws OrderNotFoundException

public static void validateOrder(Order order) throws InvalidOrderStateException

**5)Payment class:**

Private int paymentId;

Private int orderId;(FK)

Private double amount;

Private String payment\_method;

**Constructor:**Payment(int paymentId,double amount,String payment\_method)

* orderId are inherited from order class.

**Exception class:**

This throws exception when invalid payment will happen.

public class PaymentFailedException extends OnlineFoodDeliveryException

public class InsufficientFundsException extends OnlineFoodDeliveryException

**PaymentUtility class:**

public static void processPayment(Payment payment, double availableFunds) throws PaymentFailedException, InsufficientFundsException

**6)Rating class:**

Private int ratingId;

Private int orderId;(FK)

Private int stars;

Private String comment;

**Constructor:**Rating(int ratingId,int stars,String comment)

* orderId are inherited from order class.

**Exception Class**

custom exception class for handling issues related to the rating, such as invalid stars or missing comments.

public class InvalidRatingException extends OnlineFoodDeliveryException

**RatingUtility Class**

**public static void validateComment(String comment) throws RatingException**

Method to validate that the comment is not empty

**public static void processRating(Rating rating)**

Method to simulate processing the rating

**public static void validateStars(int stars) throws RatingException**

Method to validate the number of stars (it must be between 1 and 5)

**7)Driver class:**

Private int driverId;

Private String name;

Private String phoneNumber;

Private String vehicleType;

**Constructor:**Driver(int driverId,String name,String phoneNumber,String vehicleType)

* restaurantId are inherited from restaurant class;

**Exception Class:**

public class DriverNotFoundException extends OnlineFoodDeliveryException

public class DriverUnavailableException extends OnlineFoodDeliveryException

**DriverUtility Class:**

public void createDriver(Driver driver) throws DriverUnavailableException

public Driver readDriver(int driverId) throws DriverNotFoundException

public void updateDriver(Driver driver) throws DriverNotFoundException, DriverUnavailableException

public void deleteDriver(int driverId) throws DriverNotFoundException

**8)Promotion class:**

Private int promotionId;

Private int restaurantId;(FK)

Private String promotionDetails;

Private String startDate;

Private String endDate;

**Constructor:** public Promotion(int restaurantId, String restaurantName, String restaurantAddress, int promotionId, String promotionDetails, String startDate, String endDate)

**PromotionException Class:**

A custom exception class for handling issues related to the promotion, such as invalid dates or missing promotion details.

public class PromotionExpiredException extends OnlineFoodDeliveryException

public class PromotionNotFoundException extends OnlineFoodDeliveryException

**PromotionUtility Class:**

public static void validateDates(String startDate, String endDate) throws PromotionExpiredException

//Method to validate that the start date is before the end date

public static void validatePromotionDetails(String promotionDetails) throws PromotionNotFoundException

//Method to validate that promotion details are not empty

public static void applyPromotion(Promotion promotion)

//Method to simulate applying a promotion to a restaurant

**9)Delivery class:**

Private int deliveryId;

Private int orderId;(FK)

Private int driverId;

Private String deliveryStatus;

**Constructor:**Delivery(deliveryId,driverId,deliveryStatus)

**Exception Class:**

A custom exception class to handle errors related to deliveries, such as invalid status or missing driver details.

public class DeliveryDelayedException extends OnlineFoodDeliveryException

public class DeliveryUnavailableException extends OnlineFoodDeliveryException

**DeliveryUtility Class:**

**public static void validateDeliveryStatus(String deliveryStatus) throws DeliveryException**

Method to validate delivery status (supports "Pending", "In Transit", "Delivered")

**public static void assignDriverToDelivery(Delivery delivery) throws DeliveryException**

Method to assign a driver to a delivery

**public static void trackDelivery(Delivery delivery)**

Method to simulate delivery tracking.

**Inheritance:**

**User Class (Base Class)**

* **Shared by Customer and Driver.**
* **Contains general attributes like userId, name, and contactInfo.**

**Customer Class (Inherits from User)**

* **Contains additional attributes like address.**

**Driver Class (Inherits from User)**

* **Contains additional attributes like vehicle.**

**BusinessEntity Class (For entities that represent businesses like Restaurant and Promotion)**

* **General attributes like entityId, name, and contactInfo.**

**Restaurant Class (Inherits from BusinessEntity)**

* **Specific to restaurants, containing attributes like location and menu.**
* **Restaurant has a relationship with Menu.**

**Menu Class (Belongs to a Restaurant)**

* **Contains menuItems and is associated with a Restaurant.**

**Order Class (Manages order information)**

* **Central entity connecting Customer, Restaurant, and Driver.**
* **Has relationships with Delivery, Payment, and Rating.**

**Delivery Class (Handles delivery information)**

* **Associated with Order and Driver.**

**Payment Class (Handles payment information)**

* **Associated with Order.**

**Rating Class (Handles ratings for orders, restaurants, or drivers)**

* **Associated with Order, Restaurant, or Driver.**

**Promotion Class (Belongs to BusinessEntity)**

* **Can apply to either Customer or Restaurant**

// Base User class

class User {

protected int userId;

protected String name;

protected String phoneNumber;

public User(int userId, String name, String phoneNumber) {

this.userId = userId;

this.name = name;

this.phoneNumber= phoneNumber;

}

}

// Customer class inheriting from User

class Customer extends User {

private String address;

public Customer(int userId, String name, String phoneNumber, String address) {

super(userId, name, phoneNumber);

this.address = address;

}

}

// Driver class inheriting from User

class Driver extends User {

private String vehicleType;

public Driver(int userId, String name, String phoneNumber, String vehicleType) {

super(userId, name, contactInfo);

this.vehicleType = vehicleType;

}

}

// Base BusinessEntity class

class BusinessEntity {

protected int entityId;

protected String name;

protected String contactInfo;

public BusinessEntity(int entityId, String name, String contactInfo) {

this.entityId = entityId;

this.name = name;

this.contactInfo = contactInfo;

}

}

// Restaurant class inheriting from BusinessEntity

class Restaurant extends BusinessEntity {

private String location;

private Menu menu;

public Restaurant(int entityId, String name, String contactInfo, String location) {

super(entityId, name, contactInfo);

this.location = location;

this.menu = new Menu();

}

public Menu getMenu() {

return menu;

}

}

// Menu class associated with a Restaurant

class Menu {

private List<String> menuItems;

public Menu() {

this.menuItems = new ArrayList<>();

}

public void addMenuItem(String item) {

menuItems.add(item);

}

public List<String> getMenuItems() {

return menuItems;

}

}

// Order class that links Customer, Restaurant, and Driver

class Order {

private int orderId;

private Customer customer;

private Restaurant restaurant;

private Driver driver;

public Order(int orderId, Customer customer, Restaurant restaurant, Driver driver) {

this.orderId = orderId;

this.customer = customer;

this.restaurant = restaurant;

this.driver = driver;

}

}

// Delivery class

class Delivery {

private int deliveryId;

private Driver driver;

private String deliveryTime;

public Delivery(int deliveryId, Driver driver, String deliveryTime) {

this.deliveryId = deliveryId;

this.driver = driver;

this.deliveryTime = deliveryTime;

}

}

// Payment class

class Payment {

private int paymentId;

private Order order;

private String paymentType;

public Payment(int paymentId, Order order, String paymentType) {

this.paymentId = paymentId;

this.order = order;

this.paymentType = paymentType;

}

}

// Rating class

class Rating {

private int ratingId;

private int ratingValue;

private String comments;

public Rating(int ratingId, int ratingValue, String comments) {

this.ratingId = ratingId;

this.ratingValue = ratingValue;

this.comments = comments;

}

}

// Promotion class inheriting from BusinessEntity

class Promotion extends BusinessEntity {

private double discount;

public Promotion(int entityId, String name, String contactInfo, double discount) {

super(entityId, name, contactInfo);

this.discount = discount;

}

}