**NAME : MANGESH A. GHADWAJE**

**ROLL NO:24**

**BATCH : B2**

**COURSE: OOPS PRACTICAL**

**Assginment No. 13**

**Problem Statment :**

**Industry Suggested Assignment**

**Design and implement an Online Food Ordering System using Object-Oriented Programming principles. The system should allow users to browse restaurants, view menus, place orders, and track delivery status. The system should cater to both customers and restaurant owners. Functional Requirements: 1. User Authentication: - Allow users to register as customers or restaurant owners and log in securely. - Implement authentication mechanisms to ensure secure access to user accounts. 2. Restaurant Management: - Enable restaurant owners to add their restaurant to the platform and manage their menu. - Allow restaurant owners to update menu items, prices, and availability. - Provide functionalities for restaurant owners to view and manage orders received from customers. 3. Menu Browsing and Ordering: - Display a list of restaurants available for ordering. - Allow customers to browse restaurant menus, view item details, and add items to their cart. - Implement a shopping cart system for customers to review and modify their orders before checkout. 4. Order Placement and Payment: - Enable customers to place orders, specifying delivery address and preferred payment method. - Implement secure payment processing to complete orders. - Send confirmation notifications to customers and restaurant owners upon successful order placement. 5. Order Tracking and Delivery: -Provide real-time order tracking functionality for customers to monitor the status of their orders. - Update order status (e.g., confirmed, preparing, on the way, delivered) and estimated delivery time. - Allow delivery personnel to update order status upon delivery completion. 6. User Ratings and Reviews: - Allow customers to rate and review restaurants and individual menu items. - Display average ratings and reviews to help users make informed decisions. - Enable restaurant owners to respond to customer feedback and address any concerns**

**CODE:**

**import java.util.ArrayList;**

**import java.util.List;**

**class User {**

**private String username;**

**private String password;**

**public User(String username, String password) {**

**this.username = username;**

**this.password = password;**

**}**

**// Getters and setters**

**}**

**class Customer extends User {**

**private String email;**

**private String address;**

**private List<Order> orders;**

**public Customer(String username, String password, String email, String address) {**

**super(username, password);**

**this.email = email;**

**this.address = address;**

**this.orders = new ArrayList<>();**

**}**

**// Getters and setters**

**}**

**class RestaurantOwner extends User {**

**private String restaurantName;**

**private List<MenuItem> menu;**

**private List<Order> orders;**

**public RestaurantOwner(String username, String password, String restaurantName, List<MenuItem> menu) {**

**super(username, password);**

**this.restaurantName = restaurantName;**

**this.menu = menu;**

**this.orders = new ArrayList<>();**

**}**

**// Getters and setters**

**}**

**class MenuItem {**

**private String name;**

**private double price;**

**private boolean availability;**

**public MenuItem(String name, double price, boolean availability) {**

**this.name = name;**

**this.price = price;**

**this.availability = availability;**

**}**

**// Getters and setters**

**}**

**class Order {**

**private Customer customer;**

**private List<MenuItem> items;**

**private String deliveryAddress;**

**private String paymentMethod;**

**private String status;**

**private long orderTime;**

**public Order(Customer customer, List<MenuItem> items, String deliveryAddress, String paymentMethod) {**

**this.customer = customer;**

**this.items = items;**

**this.deliveryAddress = deliveryAddress;**

**this.paymentMethod = paymentMethod;**

**this.status = "Confirmed";**

**this.orderTime = System.currentTimeMillis();**

**}**

**// Getters and setters**

**}**

**class DeliveryPerson {**

**private String name;**

**public DeliveryPerson(String name) {**

**this.name = name;**

**}**

**public void updateOrderStatus(Order order, String newStatus) {**

**order.setStatus(newStatus);**

**}**

**}**

**class Restaurant {**

**private RestaurantOwner owner;**

**private String name;**

**private List<MenuItem> menu;**

**private List<Order> orders;**

**public Restaurant(RestaurantOwner owner, String name, List<MenuItem> menu) {**

**this.owner = owner;**

**this.name = name;**

**this.menu = menu;**

**this.orders = new ArrayList<>();**

**}**

**public void addMenuItem(MenuItem item) {**

**menu.add(item);**

**}**

**public void updateMenuItem(String itemName, double price, boolean availability) {**

**for (MenuItem item : menu) {**

**if (item.getName().equals(itemName)) {**

**item.setPrice(price);**

**item.setAvailability(availability);**

**break;**

**}**

**}**

**}**

**public void processOrder(Order order) {**

**orders.add(order);**

**}**

**// Getters and setters**

**}**

**class OnlineFoodOrderingSystem {**

**private List<Customer> customers;**

**private List<RestaurantOwner> restaurantOwners;**

**private List<Restaurant> restaurants;**

**private List<Order> orders;**

**private List<DeliveryPerson> deliveryPersons;**

**public OnlineFoodOrderingSystem() {**

**customers = new ArrayList<>();**

**restaurantOwners = new ArrayList<>();**

**restaurants = new ArrayList<>();**

**orders = new ArrayList<>();**

**deliveryPersons = new ArrayList<>();**

**}**

**public Customer registerCustomer(String username, String password, String email, String address) {**

**Customer customer = new Customer(username, password, email, address);**

**customers.add(customer);**

**return customer;**

**}**

**public RestaurantOwner registerRestaurantOwner(String username, String password, String restaurantName, List<MenuItem> menu) {**

**RestaurantOwner owner = new RestaurantOwner(username, password, restaurantName, menu);**

**Restaurant restaurant = new Restaurant(owner, restaurantName, menu);**

**restaurantOwners.add(owner);**

**restaurants.add(restaurant);**

**return owner;**

**}**

**public User authenticateUser(String username, String password) {**

**for (User user : customers) {**

**if (user.getUsername().equals(username) && user.getPassword().equals(password)) {**

**return user;**

**}**

**}**

**for (User user : restaurantOwners) {**

**if (user.getUsername().equals(username) && user.getPassword().equals(password)) {**

**return user;**

**}**

**}**

**return null;**

**}**

**public List<String> browseRestaurants() {**

**List<String> restaurantNames = new ArrayList<>();**

**for (Restaurant restaurant : restaurants) {**

**restaurantNames.add(restaurant.getName());**

**}**

**return restaurantNames;**

**}**

**public List<MenuItem> browseMenu(String restaurantName) {**

**for (Restaurant restaurant : restaurants) {**

**if (restaurant.getName().equals(restaurantName)) {**

**return restaurant.getMenu();**

**}**

**}**

**return null;**

**}**

**public String placeOrder(Customer customer, String restaurantName, List<MenuItem> items, String deliveryAddress, String paymentMethod) {**

**Restaurant restaurant = null;**

**for (Restaurant r : restaurants) {**

**if (r.getName().equals(restaurantName)) {**

**restaurant = r;**

**break;**

**}**

**}**

**if (restaurant == null) {**

**return "Restaurant not found.";**

**}**

**Order order = new Order(customer, items, deliveryAddress, paymentMethod);**

**restaurant.processOrder(order);**

**customer.getOrders().add(order);**

**orders.add(order);**

**return "Order placed successfully.";**

**}**

**public String trackOrder(Order order) {**

**return order.getStatus();**

**}**

**public String updateOrderStatus(Order order, String newStatus) {**

**order.setStatus(newStatus);**

**return "Order status updated successfully.";**

**}**

**public DeliveryPerson addDeliveryPerson(String name) {**

**DeliveryPerson deliveryPerson = new DeliveryPerson(name);**

**deliveryPersons.add(deliveryPerson);**

**return deliveryPerson;**

**}**

**}**

**public class Main {**

**public static void main(String[] args) {**

**// Initialize the system**

**OnlineFoodOrderingSystem system = new OnlineFoodOrderingSystem();**

**// Register a new customer**

**Customer customer = system.registerCustomer("john\_doe", "password123", "john@example.com", "123 Main St");**

**System.out.println("Customer registered: " + customer.getUsername());**

**// Register a new restaurant owner**

**List<MenuItem> menu = new ArrayList<>();**

**menu.add(new MenuItem("Burger", 5.99, true));**

**menu.add(new MenuItem("Pizza", 7.99, true));**

**RestaurantOwner owner = system.registerRestaurantOwner("owner1", "password456", "Pizza Place", menu);**

**System.out.println("Restaurant owner registered: " + owner.getUsername());**

**// Authenticate user**

**User authenticatedUser = system.authenticateUser("john\_doe", "password123");**

**if (authenticatedUser != null) {**

**System.out.println("User authenticated: " + authenticatedUser.getUsername());**

**} else {**

**System.out.println("Authentication failed");**

**}**

**// Browse restaurants**

**List<String> restaurants = system.browseRestaurants();**

**System.out.println("Available restaurants:");**

**for (String restaurantName : restaurants) {**

**System.out.println(restaurantName);**

**}**

**// Browse menu**

**List<MenuItem> menuItems = system.browseMenu("Pizza Place");**

**System.out.println("Menu for Pizza Place:");**

**for (MenuItem item : menuItems) {**

**System.out.println(item.getName() + " - $" + item.getPrice());**

**}**

**// Place an order**

**List<MenuItem> orderItems = new ArrayList<>();**

**orderItems.add(menuItems.get(0)); // Adding Burger to the order**

**String result = system.placeOrder(customer, "Pizza Place", orderItems, "123 Main St", "Credit Card");**

**System.out.println(result);**

**// Track order**

**Order order = customer.getOrders().get(0);**

**System.out.println("Order status: " + system.trackOrder(order));**

**// Update order status**

**DeliveryPerson deliveryPerson = system.addDeliveryPerson("Delivery Guy");**

**deliveryPerson.updateOrderStatus(order, "On the way");**

**System.out.println("Order status after update: " + system.trackOrder(order));**

**// Final status update**

**deliveryPerson.updateOrderStatus(order, "Delivered");**

**System.out.println("Order status after final update: " + system.trackOrder(order));**

**}**

**}**

**OUTPUT :**

**Customer registered: john\_doe**

**Restaurant owner registered: owner1**

**User authenticated: john\_doe**

**Available restaurants:**

**Pizza Place**

**Menu for Pizza Place:**

**Burger - $5.99**

**Pizza - $7.99**

**Order placed successfully.**

**Order status: Confirmed**

**Order status after update: On the way**

**Order status after final update: Delivered**