**Food Ordering System - Low-Level Design**

#### Overview:

The food delivery app is a mobile application that allows users to order food from participating restaurants. The app provides a user-friendly interface for browsing and ordering food. It also offers features like order tracking, payment integration, and delivery status updates.

**Components:**

1. User Interface (UI)
2. Authentication Module
3. Menu Management
4. Order Management
5. Payment Gateway Integration
6. Notification Service
7. Delivery Management

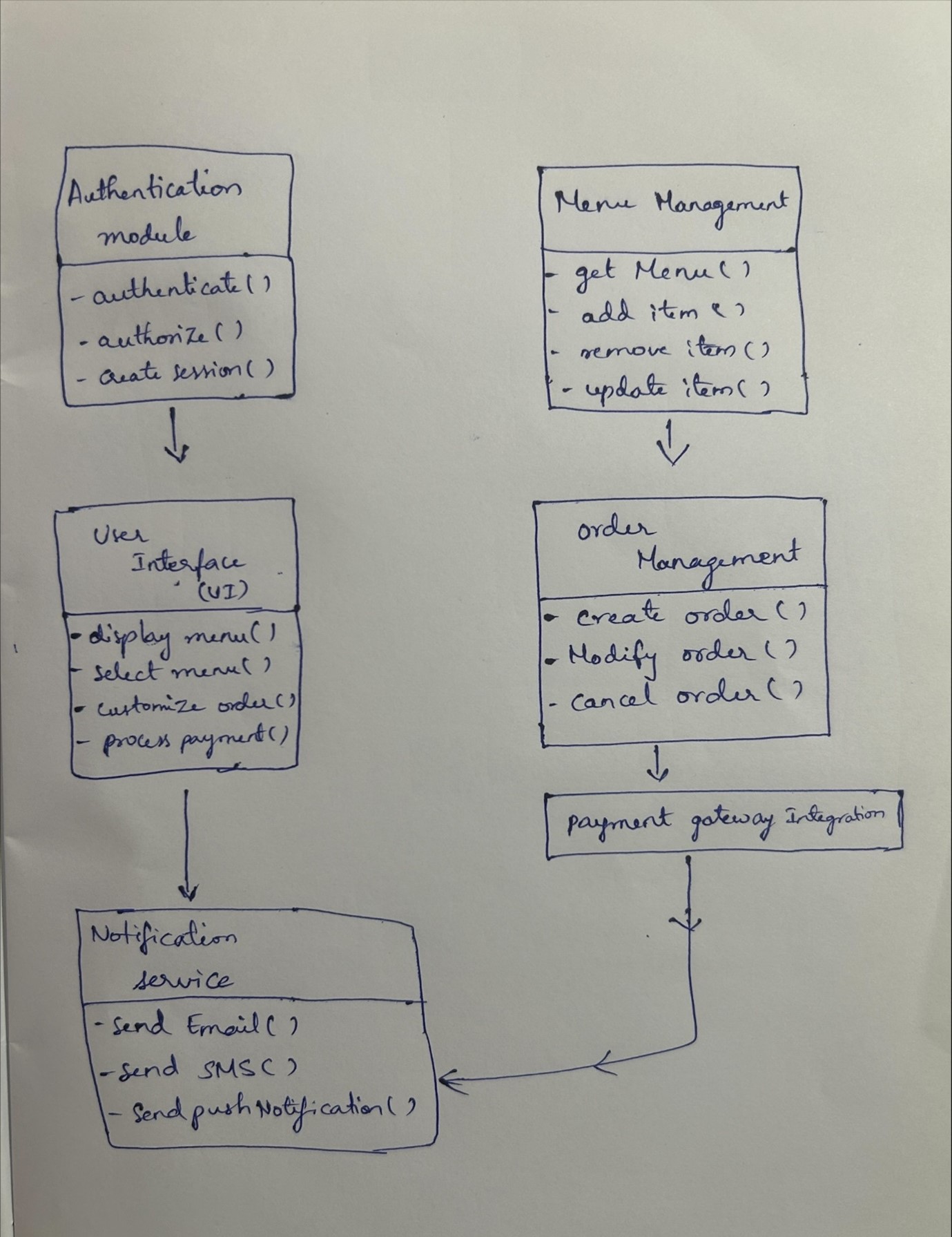
**Interaction Flow:**

1. **User Interaction:**
   * User interacts with the UI to browse menus, select items, customize orders, and proceed with payments.
2. **Authentication:**
   * UI calls the authentication module to authenticate users and create sessions.
3. **Menu Display:**
   * UI requests menu information from the Menu Management component.
   * Menu Management provides the menu details.
4. **Order Placement:**
   * User adds items to the cart, customizes orders if necessary, and proceeds to checkout.
   * UI creates an order request and sends it to the Order Management component.
5. **Payment Processing:**
   * Order Management requests payment processing from the Payment Gateway Integration component.
   * Payment Gateway Integration processes the payment and notifies Order Management of the transaction status.
6. **Order Confirmation and Notification:**
   * Order Management confirms the order and notifies the user through the Notification Service.
   * Notification Service sends confirmation via email, SMS, or push notification.
7. **Delivery Management (Optional):**
   * If applicable, Order Management coordinates with the Delivery Management component to assign a delivery person and track the delivery status.

**Security:**

To ensure the security of the food delivery app, the following security measures can be implemented:

1. User authentication: The app should require users to log in or register before accessing certain features.
2. Data encryption: The app should encrypt sensitive data, such as user passwords and payment information, before storing it in the DB.
3. Secure payment integration: The app should integrate with secure payment gateways to ensure secure payment processing.
4. Regular security updates: The app should receive regular security updates to address any potential vulnerabilities.



**1.Authentication Module**

Class: AuthenticationModule

**Methods:**

* **authenticate (username, password)**: Authenticates the user with the provided credentials.
* **authorize (user\_id, role)**: Authorizes the user based on their role.
* **createSession(user\_id)**: Creates a session for the authenticated user.

Data Structures:

* **Session**: Stores session information including user ID, session ID, and expiration time.

**2. Menu Management**

Class: Menu Manager

**Methods:**

* **getMenu(restaurant)**: Retrieves the menu for the given restaurant.
* **addItem (restaurant, item\_details)**: Adds a new item to the menu of the specified restaurant.
* **removeItem (restaurant\_id, item\_id)**: Removes an item from the menu of the specified restaurant.
* **updateItem (restaurant\_id, item\_id, new\_details)**: Updates the details of an existing item on the menu.

Data Structures:

* **MenuItem**: Represents a food item with attributes such as name, description, price, etc.
* **Menu**: Stores a list of MenuItem objects.

**3. Order Management**

Class: OrderManager

**Methods:**

* **createOrder(user\_id, cart\_items)**: Creates a new order for the user with the specified items in the cart.
* **modifyOrder(order\_id, updated\_items)**: Modifies an existing order by updating the items.
* **cancelOrder(order\_id)**: Cancels the order with the given ID.
* **processPayment(order\_id, payment\_details)**: Initiates payment processing for the specified order.

Data Structures:

* **Order**: Represents an order with attributes such as order ID, user ID, items, status, etc.

**4. Payment Gateway Integration**

Class: PaymentGateway

**Methods:**

* **processPayment(order\_id, payment\_details)**: Processes the payment for the specified order using the provided payment details.

**5. Notification Service**

Class: NotificationService

**Methods:**

* **sendEmail(to\_email, subject, message)**: Sends an email notification to the specified email address.
* **sendSMS(to\_phone, message)**: Sends an SMS notification to the specified phone number.
* **sendPushNotification(user\_id, message)**: Sends a push notification to the user's device.

**6. Delivery Management (Optional)**

Class: DeliveryManager

**Methods:**

* **assignDelivery(order\_id, delivery\_person)**: Assigns a delivery person to deliver the specified order.
* **trackDelivery(order\_id)**: Tracks the delivery status of the specified order.

Data Structures:

* **DeliveryPerson**: Represents a delivery person with attributes such as name, vehicle details, current location, etc.