**Lab Experiment: Inventory Management System**

**Aim: To draw the structural view diagram: Class Diagram**

This UML Class Diagram for an Inventory Management System represents the static structure of the system, showing key classes, attributes, methods, and relationships between them. This the UML class diagram for the inventory management system.

A diagram of a product

AI-generated content may be incorrect.

**Class Diagram Explanation**

**1. Product Class**

* **Attributes:**
  + ProductID: Unique identifier for each product.
  + Name: Name of the product.
  + Category: Category the product belongs to.
  + Quantity: Current stock quantity of the product.
  + Price: Price per unit of the product.
* **Methods:**
  + AddProduct(): Allows addition of a new product to the inventory.
  + UpdateStock(): Updates the stock quantity for the product.
  + GetProductDetails(): Retrieves detailed information about the product.
* **Relationships:**
  + Association (1:M) with InventoryTransaction:
    - A product can be involved in multiple transactions, but each transaction is linked to a single product.

**2. InventoryTransaction Class**

* **Attributes:**
  + TransactionID: Unique ID for each inventory transaction.
  + ProductID (FK): Foreign Key referring to the product involved in the transaction.
  + Date: Date of the transaction.
  + Quantity: Number of units involved in the transaction.
  + TransactionType: Type of transaction (e.g., Add, Remove, Adjust).
* **Methods:**
  + RecordTransaction(): Records details of the inventory transaction.
* **Relationships:**
  + Aggregation (M:1) with InventoryManager:
    - Multiple transactions can be managed by a single inventory manager, but transactions can exist independently.

**3. InventoryManager Class**

* **Attributes:**
  + ManagerID: Unique identifier for each inventory manager.
  + Name: Name of the manager.
  + ContactInfo: Contact details of the manager.
* **Methods:**
  + ManageStock(): Manages the stock levels of products.
  + ApproveTransaction(): Approves inventory transactions.
* **Relationships:**
  + Association (M:M) with Supplier:
    - An inventory manager can interact with multiple suppliers, and a supplier can provide goods to multiple inventory managers.

**4. Supplier Class**

* **Attributes:**
  + SupplierID: Unique identifier for each supplier.
  + Name: Name of the supplier.
  + ContactInfo: Contact details of the supplier.
  + ProductCatalog: List of products supplied.
* **Methods:**
  + SupplyProduct(): Supplies products to the inventory.
  + UpdateCatalog(): Updates the product catalog.
* **Relationships:**
  + Association (M:M) with InventoryManager:
    - Multiple suppliers can interact with multiple inventory managers.

**5. Order Class**

* **Attributes:**
  + OrderID: Unique identifier for each order.
  + ProductID (FK): Foreign key linking the order to a product.
  + Date: Date of the order.
  + Quantity: Number of units ordered.
  + Status: Current status of the order (Pending, Approved, Rejected).
* **Methods:**
  + PlaceOrder(): Places an order for a product.
  + CancelOrder(): Cancels an existing order.
  + ApproveOrder(): Approves an order if the product is available.
* **Relationships:**
  + Dependency (→) with Notification:
    - When an order status changes, a notification is sent.

**6. Notification Class**

* **Attributes:**
  + NotificationID: Unique identifier for notifications.
  + Message: Content of the notification.
  + Timestamp: Time when the notification was generated.
* **Methods:**
  + SendNotification(): Sends a notification to the relevant party.
* **Relationships:**
  + Dependency (→) with Order:
    - Notifications are triggered based on order status changes.