UML Diagram for Inventory Management System

Here's a UML Class Diagram for an inventory management system:

Classes:

- **Product:**
  - Attributes: ID, name, description, category (reference), unit price, stock level, minimum reorder level.
  - Operations:
    - Getters and setters for attributes.
    - `checkStock(quantity)`: Checks if enough stock is available.
    - `updateStock(quantity)`: Updates stock level after sale or purchase.
- **Category:**
  - Attributes: ID, name, description.
  - Operations:
    - Getters and setters for attributes.
    - `getProducts()`: Returns all products in the category.
- **Order:**
  - Attributes: ID, customer (reference, optional), order date, status (e.g., placed, shipped, delivered), order items (list of product and quantity pairs), total price.
  - Operations:
    - Getters and setters for attributes.i
    - `calculateTotal()`: Calculates the total price of the order.
    - `processOrder()`: Updates product stock levels and order status.
- **InventoryManager:**
  - Attributes: list of products, dictionary of products by category, order history.
  - Operations:
    - `addProduct(product)`: Adds a product to the inventory.
    - `removeProduct(productID)`: Removes a product from the inventory.
    - `getProduct(productID)`: Retrieves details of a specific product.
    - `getProductsByCategory(categoryID)`: Returns all products in a category.
    - `placeOrder(order)`: Processes an order and updates inventory.
    - `generateReport()`: Generates reports on inventory levels, sales, etc.
- **User:** (Optional)
  - Attributes: ID, name, role (e.g., administrator, salesperson).
  - Operations:
    - Login/logout.
    - View/manage products.
    - Place/manage orders.
    - View reports.

Relationships:

- **Product** is associated with **Category** (one-to-many).
- **Order** contains a list of **Product** (many-to-many).
- **InventoryManager** manages **Product**, **Category**, and **Order**.
- **User** (optional) interacts with **InventoryManager**.

Additional Notes:

- This is a simplified diagram, and more classes can be added for specific functionalities.
- You can use different notation details depending on your chosen UML tool.
- Consider adding additional relationships like inheritance or aggregation for more complex scenarios.

This UML diagram provides a visual overview of the key classes and their relationships in an inventory management system. You can use it as a starting point for further design and implementation.