# 1.Implementing the Product Class:

The Product class should include attributes for product details and a method to update the stock quantity.

import java.util.UUID;

public class Product {

private String productId;

private String name;

private double price;

private int stockQuantity;

public Product(String name, double price, int stockQuantity) {

this.productId = UUID.randomUUID().toString();

this.name = name;

this.price = price;

this.stockQuantity = stockQuantity;

}

public String getProductId() {

return productId;

}

public String getName() {

return name;

}

public double getPrice() {

return price;

}

public int getStockQuantity() {

return stockQuantity;

}

public void updateStockQuantity(int quantity) {

this.stockQuantity += quantity;

}

@Override

public String toString() {

return "Product{" +

"productId='" + productId + '\'' +

", name='" + name + '\'' +

", price=" + price +

", stockQuantity=" + stockQuantity +

'}';

}

}

# 2.Implementing the Customer class:

The Customer class should include attributes for customer details and methods to manage the shopping cart.

import java.util.ArrayList;

import java.util.List;

public class Customer {

private String customerId;

private String name;

private String email;

private List<Product> cart;

public Customer(String name, String email) {

this.customerId = UUID.randomUUID().toString();

this.name = name;

this.email = email;

this.cart = new ArrayList<>();

}

public String getCustomerId() {

return customerId;

}

public String getName() {

return name;

}

public String getEmail() {

return email;

}

public void addToCart(Product product) {

cart.add(product);

}

public void removeFromCart(Product product) {

cart.remove(product);

}

public void viewCart() {

System.out.println("Cart contents:");

for (Product product : cart) {

System.out.println(product);

}

}

public void checkout() {

System.out.println("Proceeding to checkout...");

for (Product product : cart) {

product.updateStockQuantity(-1);

}

cart.clear();

}

}

# 3.Implementing the Order class:

The Order class should include attributes for order details and a method to calculate the total amount of the order.

import java.time.LocalDateTime;

import java.util.List;

public class Order {

private String orderId;

private Customer customer;

private List<Product> products;

private double totalAmount;

private LocalDateTime orderDate;

public Order(Customer customer, List<Product> products) {

this.orderId = UUID.randomUUID().toString();

this.customer = customer;

this.products = products;

this.orderDate = LocalDateTime.now();

this.totalAmount = calculateTotalAmount();

}

public String getOrderId() {

return orderId;

}

public Customer getCustomer() {

return customer;

}

public List<Product> getProducts() {

return products;

}

public double getTotalAmount() {

return totalAmount;

}

public LocalDateTime getOrderDate() {

return orderDate;

}

private double calculateTotalAmount() {

double total = 0;

for (Product product : products) {

total += product.getPrice();

}

return total;

}

}

# 4. Implementing the Inventory class:

The Inventory class should include methods to manage the product inventory.

java

Copy code

import java.util.ArrayList;

import java.util.List;

public class Inventory {

private List<Product> products;

public Inventory() {

this.products = new ArrayList<>();

}

public void addProduct(Product product) {

products.add(product);

}

public Product getProductById(String productId) {

for (Product product : products) {

if (product.getProductId().equals(productId)) {

return product;

}

}

return null;

}

public void updateProductStock(String productId, int quantity) {

Product product = getProductById(productId);

if (product != null) {

product.updateStockQuantity(quantity);

}

}

public void viewInventory() {

System.out.println("Inventory contents:");

for (Product product : products) {

System.out.println(product);

}

}

}

# 5. Developing a main class to the test system:

public class Main {

public static void main(String[] args) {

Product product1 = new Product("Laptop", 1000.0, 10);

Product product2 = new Product("Smartphone", 500.0, 20);

Inventory inventory = new Inventory();

inventory.addProduct(product1);

inventory.addProduct(product2);

inventory.viewInventory();

Customer customer = new Customer("John Doe", "john.doe@example.com");

customer.addToCart(product1);

customer.addToCart(product2);

customer.viewCart();

customer.checkout();

inventory.viewInventory();

}

}

