**Assignment 5:**

K.R.VISHNU CHAITHANYA

192372057

**Case Study: Online Shopping Cart System**

**Product Class:**

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;

}

}

**Customer Class:**

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 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.getName() + " - $" + product.getPrice());

}

}

public void checkout() {

if (cart.isEmpty()) {

System.out.println("Your cart is empty.");

return;

}

double totalAmount = 0;

for (Product product : cart) {

totalAmount += product.getPrice();

}

Order order = new Order(this, cart, totalAmount);

order.calculateTotalAmount();

System.out.println("Order placed successfully. Total amount: $" + order.getTotalAmount());

cart.clear();

}

}

**Order Class:**

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, double totalAmount) {

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

this.customer = customer;

this.products = products;

this.totalAmount = totalAmount;

this.orderDate = LocalDateTime.now();

}

public double calculateTotalAmount() {

totalAmount = 0;

for (Product product : products) {

totalAmount += product.getPrice();

}

return totalAmount;

}

public double getTotalAmount() {

return totalAmount;

}

}

**Inventory Class:**

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);

}

}

}

**Main Class to Test the System:**

public class Main {

public static void main(String[] args) {

// Create instances of Inventory and Customer

Inventory inventory = new Inventory();

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

// Add products to the inventory

Product product1 = new Product("Laptop", 999.99, 10);

Product product2 = new Product("Smartphone", 499.99, 20);

inventory.addProduct(product1);

inventory.addProduct(product2);

// Simulate adding products to the customer's cart

customer.addToCart(product1);

customer.addToCart(product2);

// View the cart contents

customer.viewCart();

// Checkout

customer.checkout();

}

}