Prototype 1

<https://www.programiz.com/online-compiler/2BWvklPH4Fv2S>

Prototype 2

<https://www.programiz.com/online-compiler/3i78IjeYZPCKJ>

Prototype 3

<https://www.programiz.com/online-compiler/3aYuWULVzeIfV>

Prototype 4

<https://www.programiz.com/online-compiler/60m6OuxIIUF9S>

import java.util.\*;

class Product {

String name;

double price;

int stock;

Product(String name, double price, int stock) {

this.name = name;

this.price = price;

this.stock = stock;

}

}

class Purchase {

Product product;

int quantity;

Purchase(Product product, int quantity) {

this.product = product;

this.quantity = quantity;

}

double getTotal() {

return product.price \* quantity;

}

}

public class Main {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

// Hardcoded database of products (15 items)

List<Product> products = new ArrayList<>();

products.add(new Product("Apple", 10.0, 123));

products.add(new Product("Bread", 25.0, 65));

products.add(new Product("Milk", 30.0, 67));

products.add(new Product("Eggs", 7.5, 202));

products.add(new Product("Rice (1kg)", 50.0, 15));

products.add(new Product("Chicken", 120.0, 34));

products.add(new Product("Beef", 250.0, 43));

products.add(new Product("Pork", 180.0, 48));

products.add(new Product("Fish", 90.0, 52));

products.add(new Product("Cheese", 60.0, 122));

products.add(new Product("Butter", 45.0, 64));

products.add(new Product("Juice", 35.0, 60));

products.add(new Product("Coffee", 80.0, 40));

products.add(new Product("Sugar (1kg)", 40.0, 15));

products.add(new Product("Salt (1kg)", 20.0, 20));

// Sort products alphabetically by name

Collections.sort(products, Comparator.comparing(p -> p.name));

// Cart for purchases

List<Purchase> cart = new ArrayList<>();

char moreItems;

System.out.println("=== Simple Cashier System ===");

do {

// Show product menu with formatted table

System.out.println("\n===== PRODUCT LIST =====");

System.out.printf("%-3s %-15s %8s %8s%n", "No", "Item", "Price", "Stock");

System.out.println("---------------------------------------------");

for (int i = 0; i < products.size(); i++) {

Product p = products.get(i);

System.out.printf("%-3d %-15s %8.2f %8d%n", i + 1, p.name, p.price, p.stock);

}

// Choose product

int choice;

Product selected = null;

while (true) {

System.out.print("Choose a product (number): ");

choice = sc.nextInt();

if (choice >= 1 && choice <= products.size()) {

selected = products.get(choice - 1);

if (selected.stock > 0) {

break;

} else {

System.out.println("ERROR: That product is out of stock. Please pick another.");

}

} else {

System.out.println("ERROR: Invalid product choice.");

}

}

// Choose quantity

int qty;

while (true) {

System.out.printf("Enter quantity of %s: ", selected.name);

qty = sc.nextInt();

if (qty <= 0) {

System.out.println("ERROR: Quantity must be at least 1.");

} else if (qty > selected.stock) {

System.out.printf("ERROR: Only %d left in stock.%n", selected.stock);

} else {

break;

}

}

// Update stock and add to cart

selected.stock -= qty;

cart.add(new Purchase(selected, qty));

System.out.printf("%d x %s added to cart.%n", qty, selected.name);

// Validation loop for Y/N input

while (true) {

System.out.print("Add more items? (Y/N): ");

moreItems = sc.next().charAt(0);

if (moreItems == 'Y' || moreItems == 'y' ||

moreItems == 'N' || moreItems == 'n') {

break;

} else {

System.out.println("ERROR: Please enter only 'Y' or 'N'.");

}

}

} while (moreItems == 'Y' || moreItems == 'y');

// Generate receipt

System.out.println("\n===== RECEIPT =====");

System.out.printf("%-15s %5s %8s %10s%n", "Item", "Qty", "Price", "Total");

System.out.println("------------------------------------------------");

double grandTotal = 0;

for (Purchase p : cart) {

double total = p.getTotal();

System.out.printf("%-15s %5d %8.2f %10.2f%n",

p.product.name, p.quantity, p.product.price, total);

grandTotal += total;

}

System.out.println("------------------------------------------------");

System.out.printf("%-30s %10.2f%n", "GRAND TOTAL:", grandTotal);

// Payment handling

double cash = 0;

while (cash < grandTotal) {

System.out.print("Enter cash received: ");

cash += sc.nextDouble();

if (cash < grandTotal) {

System.out.printf("Insufficient payment. Remaining balance: %.2f%n", grandTotal - cash);

}

}

double change = cash - grandTotal;

System.out.printf("Change: %.2f%n", change);

System.out.println("Thank you for your purchase!");

sc.close();

}

}