**1. Inventory Management:**

package CCP;

import java.time.LocalDate;

import java.util.ArrayList;

import java.util.Scanner;

public class InventoryManagement {

private static final ArrayList<Medicine> inventory = new ArrayList<>();

private static final ArrayList<String> addedMedicines = new ArrayList<>();

private static final ArrayList<String> updatedMedicines = new ArrayList<>();

private static final ArrayList<String> deletedMedicines = new ArrayList<>();

public static void addMedicine(String name, int stock, LocalDate expiryDate)

{

inventory.add(new Medicine(name, stock, expiryDate));

addedMedicines.add(name);

System.out.println("Medicine added successfully.");

}

public static void updateMedicine(String name, int newStock, LocalDate newExpiryDate) {

for (Medicine medicine : inventory) {

//The equalsIgnoreCase() method in Java is used to compare two strings, ignoring their case sensitivity.

if (medicine.name.equalsIgnoreCase(name)) {

medicine.stock = newStock;

medicine.expiryDate = newExpiryDate;

updatedMedicines.add(name);

System.out.println("Medicine updated successfully.");

return;

}

}

System.out.println("Medicine not found.");

}

public static void deleteMedicine(String name) {

// (removeIf) Removes elements from the list that satisfy the given condition.

// In this case, it removes the medicine that the user wants to delete.

if (inventory.removeIf(medicine -> medicine.name.equalsIgnoreCase(name))) {

deletedMedicines.add(name);

System.out.println("Medicine deleted successfully.");

} else {

System.out.println("Medicine not found.");

}

}

public static void trackStockLevels() {

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

for (Medicine medicine : inventory) {

System.out.println("Name: " + medicine.name + ", Stock: " + medicine.stock + ", Expiry Date: " + medicine.expiryDate);

}

}

public static void generateAlerts() {

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

LocalDate today = LocalDate.now();

for (Medicine medicine : inventory) {

if (medicine.stock < 5) {

System.out.println("Low stock alert: " + medicine.name);

}

if (medicine.expiryDate.isBefore(today)) {

System.out.println("Expired medicine alert: " + medicine.name);

}

}

}

public static void displaySummary() {

System.out.println("\n--- Summary ---");

System.out.println("Added Medicines: " + addedMedicines);

System.out.println("Updated Medicines: " + updatedMedicines);

System.out.println("Deleted Medicines: " + deletedMedicines);

System.out.println("Out of Stock Medicines:");

for (Medicine medicine : inventory) {

if (medicine.stock == 0) {

System.out.println("- " + medicine.name);

}

}

System.out.println("Expired Medicines:");

LocalDate today = LocalDate.now();

for (Medicine medicine : inventory) {

if (medicine.expiryDate.isBefore(today)) {

System.out.println("- " + medicine.name);

}

}

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

while (true) {

System.out.println("\n--- Inventory Management ---");

System.out.println("1. Add Medicine");

System.out.println("2. Update Medicine");

System.out.println("3. Delete Medicine");

System.out.println("4. Track Stock Levels");

System.out.println("5. Generate Alerts");

System.out.println("6. Display Summary");

System.out.println("7. Exit");

System.out.print("Enter your choice: ");

int choice = scanner.nextInt();

scanner.nextLine();

switch (choice) {

case 1 -> {

System.out.print("Enter medicine name: ");

String name = scanner.nextLine();

System.out.print("Enter stock: ");

int stock = scanner.nextInt();

System.out.print("Enter expiry date (YYYY-MM-DD): ");

LocalDate expiryDate = LocalDate.parse(scanner.next()); // parse:Converts the string input into LocalDate

addMedicine(name, stock, expiryDate);

}

case 2 -> {

System.out.print("Enter medicine name to update: ");

String name = scanner.nextLine();

System.out.print("Enter new stock: ");

int stock = scanner.nextInt();

System.out.print("Enter new expiry date (YYYY-MM-DD): ");

LocalDate expiryDate = LocalDate.parse(scanner.next());

updateMedicine(name, stock, expiryDate);

}

case 3 -> {

System.out.print("Enter medicine name to delete: ");

String name = scanner.nextLine();

deleteMedicine(name);

}

case 4 -> trackStockLevels();

case 5 -> generateAlerts();

case 6 -> displaySummary();

case 7 -> {

System.out.println("Exiting program. Goodbye!");

scanner.close();

return;

}

default -> System.out.println("Invalid choice. Try again.");

}

} }

}

**2. Prescription Management:**

**package** CCP;

**import** java.util.\*;

**class** Prescription {

**private** String patientName;

**private** String prescription;

**private** List<String> medicineList;

**public** Prescription(String patientName, String prescription) {

**this**.patientName = patientName;

**this**.prescription = prescription;

**this**.medicineList = **new** ArrayList<>();

}

**public** String getPatientName() {

**return** patientName;

}

**public** String getPrescription() {

**return** prescription;

}

**public** **void** addMedicine(String medicine) {

**this**.medicineList.add(medicine);

}

**public** List<String> getMedicineList() {

**return** medicineList;

}

}

**package CCP;**

**import java.util.Arrays;**

**import java.util.HashMap;**

**import java.util.List;**

**import java.util.Map;**

**import java.util.Scanner;**

**public class PrescriptionManagement {**

**private static Map<String, Prescription> prescriptions = new HashMap<>();**

**// Method to add prescription**

**public static void addPrescription(String patientName, String prescription, List<String> medicines) {**

**Prescription newPrescription = new Prescription(patientName, prescription);**

**for (String medicine : medicines) {**

**newPrescription.addMedicine(medicine);**

**}**

**prescriptions.put(patientName, newPrescription);**

**}**

**// Method to verify prescription and medicine**

**public static void verifyPrescription(String patientName, List<String> medicineToVerify) {**

**Prescription prescription = prescriptions.get(patientName);**

**if (prescription != null) {**

**System.out.println("Verifying Prescription for " + patientName);**

**List<String> validMedicines = prescription.getMedicineList();**

**boolean isValid = true;**

**for (String medicine : medicineToVerify) {**

**if (!validMedicines.contains(medicine)) {**

**System.out.println("Invalid medicine: " + medicine);**

**isValid = false;**

**}**

**}**

**if (isValid) {**

**System.out.println("Prescription and medicines are valid for " + patientName);**

**}**

**} else {**

**System.out.println("No prescription found for " + patientName);**

**}**

**}**

**public static void main(String[] args) {**

**Scanner sc = new Scanner(System.in);**

**// Input to add a prescription**

**System.out.println("Enter patient name for new prescription:");**

**String patientName = sc.nextLine();**

**System.out.println("Enter the prescription details:");**

**String prescriptionDetails = sc.nextLine();**

**System.out.println("Enter medicines for the prescription:");**

**String medicinesInput = sc.nextLine();**

**List<String> medicines = Arrays.asList(medicinesInput.split(","));**

**// Add the prescription**

**addPrescription(patientName, prescriptionDetails, medicines);**

**// Input to verify prescription and medicine**

**System.out.println("Enter patient name to verify prescription:");**

**String verifyPatientName = sc.nextLine();**

**System.out.println("Enter medicines to verify:");**

**String medicinesToVerifyInput = sc.nextLine();**

**List<String> medicinesToVerify = Arrays.asList(medicinesToVerifyInput.split(","));**

**verifyPrescription(verifyPatientName, medicinesToVerify);**

**sc.close();**

**}**

**}**

**3. Sales and Billing:**

**package** CCP;

**import** java.util.Scanner;

**public** **class** SalesAndBilling {

**public** **static** **void** main(String[] args) {

Scanner scanner = **new** Scanner(System.***in***);

**while** (**true**) {

System.***out***.print("Enter product name: ");

String productName = scanner.nextLine();

System.***out***.print("Enter product price: ");

**double** productPrice = scanner.nextDouble();

System.***out***.print("Enter quantity: ");

**int** quantity = scanner.nextInt();

**double** totalAmount = productPrice \* quantity;

**int** paymentChoice = 0;

**while** (paymentChoice != 1 && paymentChoice != 2) {

System.***out***.println("\nSelect payment method:");

System.***out***.println("1. Cash");

System.***out***.println("2. Card");

System.***out***.print("Enter choice (1 or 2): ");

paymentChoice = scanner.nextInt();

**if** (paymentChoice != 1 && paymentChoice != 2) {

System.***out***.println("Invalid payment method. Please choose 1 for Cash or 2 for Card.");

}

}

**double** paymentAmount = 0.0;

**switch** (paymentChoice) {

**case** 1:

System.***out***.print("Enter cash amount: ");

paymentAmount = scanner.nextDouble();

**if** (paymentAmount < totalAmount) {

System.***out***.println("Insufficient cash! Please provide correct amount.");

} **else** {

System.***out***.println("Payment successful.");

**double** change = paymentAmount - totalAmount;

System.***out***.println("Change: " + change);

}

**break**;

**case** 2:

System.***out***.print("Enter card number: ");

String cardNumber = scanner.next();

System.***out***.println("Processing card payment...");

System.***out***.println("Payment successful.");

**break**;

**default**:

System.***out***.println("Invalid payment method.");

**break**;

}

*generateReceipt*(productName, productPrice, quantity, totalAmount, paymentAmount, paymentChoice);

System.***out***.println("\nDo you want to process another sale? (yes/no)");

scanner.nextLine(); // Consume the leftover newline character

String choice = scanner.nextLine();

**if** (!choice.equalsIgnoreCase("yes")) {

System.***out***.println("Thank you for using our system!");

**break**; // Exit the loop if the user does not want another transaction

}

}

}

**public** **static** **void** generateReceipt(String productName, **double** productPrice, **int** quantity, **double** totalAmount, **double** paymentAmount, **int** paymentChoice) {

System.***out***.println("\n--- Receipt ---");

System.***out***.println("Product: " + productName);

System.***out***.println("Price: " + productPrice);

System.***out***.println("Quantity: " + quantity);

System.***out***.println("Total: " + totalAmount);

**if** (paymentChoice == 1) {

System.***out***.println("Payment Method: Cash");

System.***out***.println("Cash Given: " + paymentAmount);

System.***out***.println("Change: " + (paymentAmount - totalAmount));

} **else** **if** (paymentChoice == 2) {

System.***out***.println("Payment Method: Card");

}

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

}

}

**4. Customer Record Management:**

**package** CCP;

**import** java.util.\*;

**class** Customer {

**private** String name;

**private** String contactDetails;

**private** List<String> prescriptionHistory;

**public** Customer(String name, String contactDetails) {

**this**.name = name;

**this**.contactDetails = contactDetails;

**this**.prescriptionHistory = **new** ArrayList<>();

}

**public** String getName() {

**return** name;

}

**public** String getContactDetails() {

**return** contactDetails;

}

**public** List<String> getPrescriptionHistory() {

**return** prescriptionHistory;

}

**public** **void** addPrescription(String prescription) {

**this**.prescriptionHistory.add(prescription);

}

**public** **void** updateContactDetails(String newContact) {

**this**.contactDetails = newContact;

}

}

**package** CCP;

**import** java.util.HashMap;

**import** java.util.Map;

**import** java.util.Scanner;

**public** **class** CustomerRecordManagement {

**private** **static** Map<String, Customer> *customerRecords* = **new** HashMap<>();

// Method to add a new customer

**public** **static** **void** addCustomer(String name, String contactDetails) {

**if** (name.isEmpty() || contactDetails.isEmpty()) {

System.***out***.println("No data entered. Returning to main menu...");

**return**;

}

Customer customer = **new** Customer(name, contactDetails);

*customerRecords*.put(name, customer);

System.***out***.println("Customer added successfully.");

}

// Method to update customer's contact details

**public** **static** **void** updateCustomerContact(String name, String newContact) {

**if** (newContact.isEmpty()) {

System.***out***.println("No data entered. Returning to main menu...");

**return**;

}

Customer customer = *customerRecords*.get(name);

**if** (customer != **null**) {

customer.updateContactDetails(newContact);

System.***out***.println("Contact details updated successfully.");

} **else** {

System.***out***.println("Customer not found.");

}

}

// Method to add a prescription to a customer's history

**public** **static** **void** addPrescriptionToCustomer(String customerName, String prescription) {

**if** (prescription.isEmpty()) {

System.***out***.println("No prescription entered. Returning to main menu...");

**return**;

}

Customer customer = *customerRecords*.get(customerName);

**if** (customer != **null**) {

customer.addPrescription(prescription);

System.***out***.println("Prescription added successfully.");

} **else** {

System.***out***.println("Customer not found.");

}

}

// Method to generate a simple report for a customer

**public** **static** **void** generateCustomerReport(String customerName) {

**if** (customerName.isEmpty()) {

System.***out***.println("No customer name entered. Returning to main menu...");

**return**;

}

Customer customer = *customerRecords*.get(customerName);

**if** (customer != **null**) {

System.***out***.println("Customer Report for: " + customerName);

System.***out***.println("Contact Details: " + customer.getContactDetails());

System.***out***.println("Prescription History: " + customer.getPrescriptionHistory());

} **else** {

System.***out***.println("Customer not found.");

}

}

**public** **static** **void** main(String[] args) {

Scanner sc = **new** Scanner(System.***in***);

**boolean** continueRunning = **true**;

**while** (continueRunning) {

System.***out***.println("\nCustomer Record Management Menu:");

System.***out***.println("1. Add Customer");

System.***out***.println("2. Update Customer Contact Details");

System.***out***.println("3. Add Prescription to Customer");

System.***out***.println("4. Generate Customer Report");

System.***out***.println("5. Exit");

System.***out***.print("Enter your choice: ");

**int** choice = sc.nextInt();

sc.nextLine(); // Consume the newline character

**switch** (choice) {

**case** 1:

// Add new customer

System.***out***.print("Enter customer name (or press Enter to go back): ");

String name = sc.nextLine();

**if** (name.isEmpty()) {

System.***out***.println("Returning to main menu...");

**break**;

}

System.***out***.print("Enter customer contact details (or press Enter to go back): ");

String contactDetails = sc.nextLine();

**if** (contactDetails.isEmpty()) {

System.***out***.println("Returning to main menu...");

**break**;

}

*addCustomer*(name, contactDetails);

**break**;

**case** 2:

// Update customer contact details

System.***out***.print("Enter customer name to update contact (or press Enter to go back): ");

String customerNameForUpdate = sc.nextLine();

**if** (customerNameForUpdate.isEmpty()) {

System.***out***.println("Returning to main menu...");

**break**;

}

System.***out***.print("Enter new contact details (or press Enter to go back): ");

String newContact = sc.nextLine();

**if** (newContact.isEmpty()) {

System.***out***.println("Returning to main menu...");

**break**;

}

*updateCustomerContact*(customerNameForUpdate, newContact);

**break**;

**case** 3:

// Add prescription to customer

System.***out***.print("Enter customer name to add prescription (or press Enter to go back): ");

String customerNameForPrescription = sc.nextLine();

**if** (customerNameForPrescription.isEmpty()) {

System.***out***.println("Returning to main menu...");

**break**;

}

System.***out***.print("Enter prescription details (or press Enter to go back): ");

String prescription = sc.nextLine();

**if** (prescription.isEmpty()) {

System.***out***.println("Returning to main menu...");

**break**;

}

*addPrescriptionToCustomer*(customerNameForPrescription, prescription);

**break**;

**case** 4:

// Generate customer report

System.***out***.print("Enter customer name for report (or press Enter to go back): ");

String customerNameForReport = sc.nextLine();

**if** (customerNameForReport.isEmpty()) {

System.***out***.println("Returning to main menu...");

**break**;

}

*generateCustomerReport*(customerNameForReport);

**break**;

**case** 5:

// Exit the program

System.***out***.println("Exiting program...");

continueRunning = **false**;

**break**;

**default**:

System.***out***.println("Invalid choice. Please try again.");

}

}

sc.close();

}

}

**5. Reporting:**

**package** CCP;

**import** java.util.ArrayList;

**import** java.util.List;

**public** **class** Pharmacyreport {

// Simulate inventory data

**private** **static** List<String[]> *inventoryData* = **new** ArrayList<>();

**private** **static** List<String[]> *salesData* = **new** ArrayList<>();

**private** **static** List<String[]> *prescriptionData* = **new** ArrayList<>();

// Method to initialize data

**public** **static** **void** initializeData() {

// Sample inventory data

*inventoryData*.add(**new** String[]{"Paracetamol", "100", "2025-01-01"});

*inventoryData*.add(**new** String[]{"Ibuprofen", "50", "2024-12-31"});

// Sample sales data

*salesData*.add(**new** String[]{"1", "Paracetamol", "5", "2024-12-26", "50.00"});

*salesData*.add(**new** String[]{"2", "Ibuprofen", "3", "2024-12-25", "30.00"});

// Sample prescription data

*prescriptionData*.add(**new** String[]{"1", "John Doe", "Paracetamol", "2024-12-25", "1 tablet twice daily"});

*prescriptionData*.add(**new** String[]{"2", "Jane Smith", "Ibuprofen", "2024-12-24", "1 tablet after meals"});

}

**public** **static** **void** generateReport(String reportType) {

**switch** (reportType) {

**case** "inventory":

System.***out***.println("\nInventory Report:");

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

**for** (String[] item : *inventoryData*) {

System.***out***.printf("Medicine: %s, Quantity: %s, Expiry Date: %s%n", item[0], item[1], item[2]);

}

**break**;

**case** "sales":

System.***out***.println("\nSales Report:");

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

**for** (String[] sale : *salesData*) {

System.***out***.printf("Sale ID: %s, Medicine: %s, Quantity Sold: %s, Sale Date: %s, Total Price: %s%n",

sale[0], sale[1], sale[2], sale[3], sale[4]);

}

**break**;

**case** "prescriptions":

System.***out***.println("\nPrescription History Report:");

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

**for** (String[] prescription : *prescriptionData*) {

System.***out***.printf("Prescription ID: %s, Patient: %s, Medicine: %s, Prescribed Date: %s, Dosage: %s%n",

prescription[0], prescription[1], prescription[2], prescription[3], prescription[4]);

}

**break**;

**default**:

System.***out***.println("Invalid report type.");

}

}

**public** **static** **void** main(String[] args) {

// Initialize data

*initializeData*();

// Generate reports

*generateReport*("inventory");

*generateReport*("sales");

*generateReport*("prescriptions");

}

}