import java.util.ArrayList;

import java.util.Scanner;

class StaffMember {

private int staffId;

private String staffName;

private String department;

private double salary;

public StaffMember(int staffId, String staffName, String department, double salary) {

this.staffId = staffId;

this.staffName = staffName;

this.department = department;

this.salary = salary;

}

public int getStaffId() {

return staffId;

}

public String getStaffName() {

return staffName;

}

public void setStaffName(String staffName) {

this.staffName = staffName;

}

public String getDepartment() {

return department;

}

public void setDepartment(String department) {

this.department = department;

}

public double getSalary() {

return salary;

}

public void setSalary(double salary) {

this.salary = salary;

}

@Override

public String toString() {

return "StaffMember [ID=" + staffId + ", Name=" + staffName + ", Department=" + department + ", Salary=" + salary + "]";

}

}

public class StaffManagementApp {

private static ArrayList<StaffMember> staffList = new ArrayList<>();

private static Scanner input = new Scanner(System.in);

public static void main(String[] args) {

while (true) {

System.out.println("\nStaff Management System");

System.out.println("1. Add Staff Member");

System.out.println("2. Display All Staff");

System.out.println("3. Modify Staff Member");

System.out.println("4. Remove Staff Member");

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

System.out.print("Choose an option: ");

int option = input.nextInt();

switch (option) {

case 1:

addStaffMember();

break;

case 2:

displayAllStaff();

break;

case 3:

modifyStaffMember();

break;

case 4:

removeStaffMember();

break;

case 5:

System.out.println("Exiting the system.");

System.exit(0);

default:

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

}

}

}

private static void addStaffMember() {

System.out.print("Enter Staff ID: ");

int id = input.nextInt();

input.nextLine(); // Consume newline

System.out.print("Enter Staff Name: ");

String name = input.nextLine();

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

String dept = input.nextLine();

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

double salary = input.nextDouble();

StaffMember staff = new StaffMember(id, name, dept, salary);

staffList.add(staff);

System.out.println("Staff member added successfully.");

}

private static void displayAllStaff() {

if (staffList.isEmpty()) {

System.out.println("No staff members found.");

} else {

for (StaffMember staff : staffList) {

System.out.println(staff);

}

}

}

private static void modifyStaffMember() {

System.out.print("Enter Staff ID to modify: ");

int id = input.nextInt();

input.nextLine(); // Consume newline

StaffMember staff = findStaffById(id);

if (staff != null) {

System.out.print("Enter new name (current: " + staff.getStaffName() + "): ");

String name = input.nextLine();

System.out.print("Enter new department (current: " + staff.getDepartment() + "): ");

String dept = input.nextLine();

System.out.print("Enter new salary (current: " + staff.getSalary() + "): ");

double salary = input.nextDouble();

staff.setStaffName(name);

staff.setDepartment(dept);

staff.setSalary(salary);

System.out.println("Staff member details updated.");

} else {

System.out.println("Staff member not found.");

}

}

private static void removeStaffMember() {

System.out.print("Enter Staff ID to remove: ");

int id = input.nextInt();

StaffMember staff = findStaffById(id);

if (staff != null) {

staffList.remove(staff);

System.out.println("Staff member removed successfully.");

} else {

System.out.println("Staff member not found.");

}

}

private static StaffMember findStaffById(int id) {

for (StaffMember staff : staffList) {

if (staff.getStaffId() == id) {

return staff;

}

}

return null;

}

}