import java.util.ArrayList;

import java.util.Scanner;

// Employee class

class Employee {

    private int id;

    private String name;

    private String role;

    private double salary;

    private String department;

    public Employee(int id, String name, String role, double salary, String department) {

        this.id = id;

        this.name = name;

        this.role = role;

        this.salary = salary;

        this.department = department;

    }

    // Getters and setters

    public int getId() {

        return id;

    }

    public void setId(int id) {

        this.id = id;

    }

    public String getName() {

        return name;

    }

    public void setName(String name) {

        this.name = name;

    }

    public String getRole() {

        return role;

    }

    public void setRole(String role) {

        this.role = role;

    }

    public double getSalary() {

        return salary;

    }

    public void setSalary(double salary) {

        this.salary = salary;

    }

    public String getDepartment() {

        return department;

    }

    public void setDepartment(String department) {

        this.department = department;

    }

    // toString method for displaying employee details

    @Override

    public String toString() {

        return "ID: " + id + ", Name: " + name + ", Role: " + role + ", Salary: " + salary + ", Department: " + department;

    }

}

// Manager class inherits from Employee

class Manager extends Employee {

    private double bonus;

    public Manager(int id, String name, String role, double salary, String department, double bonus) {

        super(id, name, role, salary, department);

        this.bonus = bonus;

    }

    public double getBonus() {

        return bonus;

    }

    public void setBonus(double bonus) {

        this.bonus = bonus;

    }

    @Override

    public String toString() {

        return super.toString() + ", Bonus: " + bonus;

    }

}

// EmployeeDatabase class to manage employees

class EmployeeDatabase {

    private ArrayList<Employee> employees;

    public EmployeeDatabase() {

        employees = new ArrayList<>();

    }

    // Add a new employee

    public void addEmployee(Employee employee) {

        employees.add(employee);

    }

    // List all employees

    public void listAllEmployees() {

        if (employees.isEmpty()) {

            System.out.println("No employees found.");

        } else {

            for (Employee emp : employees) {

                System.out.println(emp);

            }

        }

    }

    // Get employee by ID

    public Employee getEmployeeById(int id) {

        for (Employee emp : employees) {

            if (emp.getId() == id) {

                return emp;

            }

        }

        return null;

    }

    // Remove employee by ID

    public boolean removeEmployee(int id) {

        Employee emp = getEmployeeById(id);

        if (emp != null) {

            employees.remove(emp);

            return true;

        }

        return false;

    }

    // Update employee details

    public boolean updateEmployee(int id, String name, String role, double salary, String department) {

        Employee emp = getEmployeeById(id);

        if (emp != null) {

            emp.setName(name);

            emp.setRole(role);

            emp.setSalary(salary);

            emp.setDepartment(department);

            return true;

        }

        return false;

    }

    // Search employee by name

    public void searchEmployeeByName(String name) {

        boolean found = false;

        for (Employee emp : employees) {

            if (emp.getName().toLowerCase().contains(name.toLowerCase())) {

                System.out.println(emp);

                found = true;

            }

        }

        if (!found) {

            System.out.println("No employees found with that name.");

        }

    }

}

// Main class for the Employee Management System

public class EmployeeManagementSystem {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        EmployeeDatabase database = new EmployeeDatabase();

        // Add some initial employees

        database.addEmployee(new Employee(1, "John Doe", "Developer", 60000, "IT"));

        database.addEmployee(new Manager(2, "Jane Smith", "Manager", 80000, "HR", 5000));

        boolean running = true;

        while (running) {

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

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

            System.out.println("2. View All Employees");

            System.out.println("3. Update Employee");

            System.out.println("4. Remove Employee");

            System.out.println("5. Search Employee by Name");

            System.out.println("6. Exit");

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

            int choice = scanner.nextInt();

            scanner.nextLine(); // Consume newline character after integer input

            switch (choice) {

                case 1:

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

                    int id = scanner.nextInt();

                    scanner.nextLine(); // Consume newline

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

                    String name = scanner.nextLine();

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

                    String role = scanner.nextLine();

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

                    double salary = scanner.nextDouble();

                    scanner.nextLine(); // Consume newline

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

                    String department = scanner.nextLine();

                    System.out.print("Is this employee a Manager (yes/no): ");

                    String isManager = scanner.nextLine();

                    if (isManager.equalsIgnoreCase("yes")) {

                        System.out.print("Enter Manager Bonus: ");

                        double bonus = scanner.nextDouble();

                        database.addEmployee(new Manager(id, name, role, salary, department, bonus));

                    } else {

                        database.addEmployee(new Employee(id, name, role, salary, department));

                    }

                    System.out.println("Employee added successfully!");

                    break;

                case 2:

                    database.listAllEmployees();

                    break;

                case 3:

                    System.out.print("Enter Employee ID to update: ");

                    int updateId = scanner.nextInt();

                    scanner.nextLine(); // Consume newline

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

                    String newName = scanner.nextLine();

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

                    String newRole = scanner.nextLine();

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

                    double newSalary = scanner.nextDouble();

                    scanner.nextLine(); // Consume newline

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

                    String newDepartment = scanner.nextLine();

                    if (database.updateEmployee(updateId, newName, newRole, newSalary, newDepartment)) {

                        System.out.println("Employee updated successfully!");

                    } else {

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

                    }

                    break;

                case 4:

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

                    int removeId = scanner.nextInt();

                    if (database.removeEmployee(removeId)) {

                        System.out.println("Employee removed successfully!");

                    } else {

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

                    }

                    break;

                case 5:

                    System.out.print("Enter name to search for: ");

                    String searchName = scanner.nextLine();

                    database.searchEmployeeByName(searchName);

                    break;

                case 6:

                    running = false;

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

                    break;

                default:

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

            }

        }

        scanner.close();

    }

}