Experiment 4

Name: Piyush Sharma
UID: 22BCS13018
Branch: BE-CSE
Section: EPAM-801-B
Semester:6th
Date : 17/02/2025
Subject: PBLJ
Subject Code: 22CSH-359

1. Aim:

Write a Program to perform the basic operations like insert, delete, display and search in list. List contains String object items where these operations are to be performed.

2. Objective:

The objective of this program is to perform basic operations (insert, search, delete, display) on a list of strings using Java. It demonstrates the use of ArrayList, user input handling, and control structures for efficient data management

3. Implementation/Code:

```
import java.util.ArrayList;
import java.util.Scanner;

class Employee {
    private int id;
    private String name;
    private double salary;

    public Employee(int id, String name, double salary) {
        this.id = id;
        this.name = name;
        this.salary = salary;
    }
    public int getId() { return id; }
    public String getName() { return name; }
    public double getSalary() { return salary; }
    public void setName(String name) { this.name = name; }
    public void setSalary(double salary) { this.salary = salary; }
    @Override
    public String toString() {
        return "Employee { ID: " + id + ", Name: " + name + ", Salary: " + salary + " }";
    }
}

public class EmployeeManager {
    private static ArrayList<Employee> employees = new ArrayList<>();
    private static Scanner scanner = new Scanner(System.in);

    public static void main(String[] args) {
        while (true) {
            System.out.println("\nEmployee Management System");
            System.out.println("1. Add Employee");
            System.out.println("2. Update Employee");
            System.out.println("3. Remove Employee");
            System.out.println("4. Search Employee");
            System.out.println("5. Display All Employees");
            System.out.println("6. Exit");
            System.out.print("Choose an option: ");

            int choice = scanner.nextInt();
            scanner.nextLine();

            switch (choice) {
                case 1 -> addEmployee();
                case 2 -> updateEmployee();
                case 3 -> removeEmployee();
                case 4 -> searchEmployee();
                case 5 -> displayEmployees();
                case 6 -> {
                    System.out.println("Exiting program...");
                    return;
                }
                default -> System.out.println("Invalid choice. Try again!");
            }
        }
    }
```

```
// 1 Add Employee
private static void addEmployee() {
    System.out.print("Enter Employee ID: ");
    int id = scanner.nextInt();
    scanner.nextLine();
    System.out.print("Enter Employee Name: ");
    String name = scanner.nextLine();
    System.out.print("Enter Employee Salary: ");
    double salary = scanner.nextDouble();

    employees.add(new Employee(id, name, salary));
    System.out.println("Employee added successfully!");
}

// 2 Update Employee
private static void updateEmployee() {
    System.out.print("Enter Employee ID to update: ");
    int id = scanner.nextInt();
    scanner.nextLine();

    for (Employee emp : employees) {
        if (emp.getId() == id) {
            System.out.print("Enter new Name: ");
            String newName = scanner.nextLine();
            System.out.print("Enter new Salary: ");
            double newSalary = scanner.nextDouble();

            emp.setName(newName);
            emp.setSalary(newSalary);
            System.out.println("Employee updated successfully!");
            return;
        }
    }
    System.out.println("Employee not found!");
}

// 3 Remove Employee
private static void removeEmployee() {
    System.out.print("Enter Employee ID to remove: ");
    int id = scanner.nextInt();
```

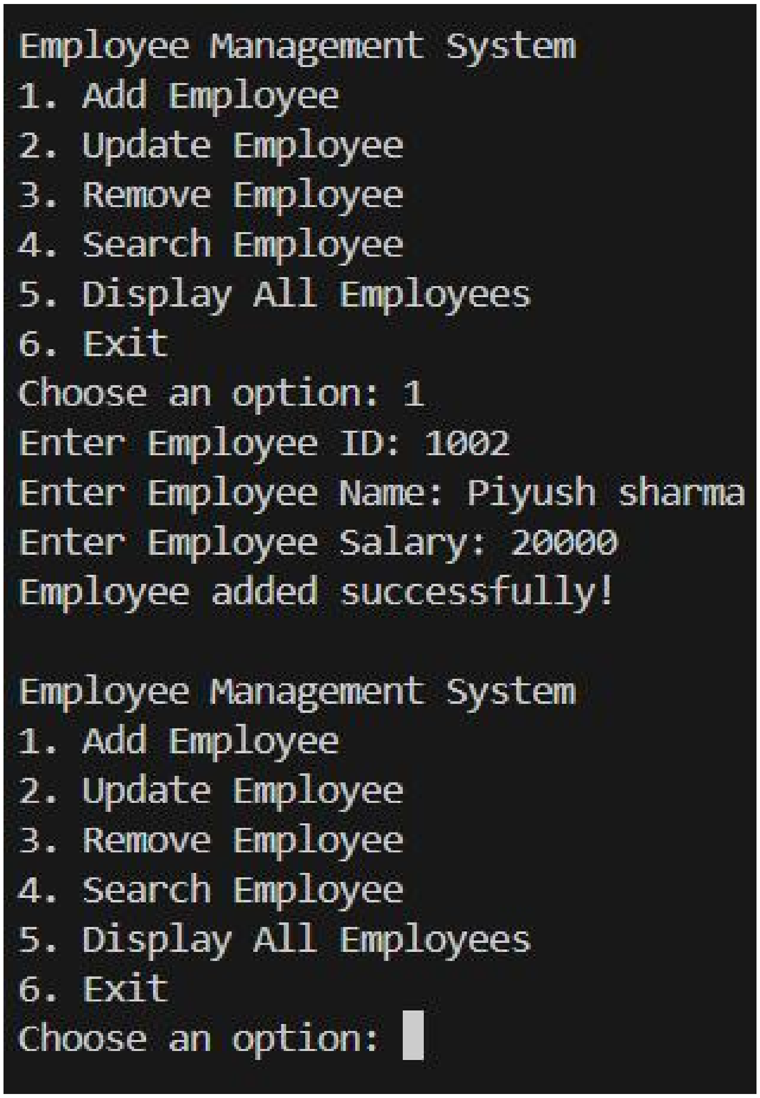
```
        employees.removeIf(emp -> emp.getId() == id);
        System.out.println("Employee removed successfully (if existed).");
    }

    // 4 Search Employee
    private static void searchEmployee() {
        System.out.print("Enter Employee ID to search: ");
        int id = scanner.nextInt();

        for (Employee emp : employees) {
            if (emp.getId() == id) {
                System.out.println("Employee Found: " + emp);
                return;
            }
        }
        System.out.println("Employee not found!");
    }

    // 5 Display All Employees
    private static void displayEmployees() {
        if (employees.isEmpty()) {
            System.out.println("No employees in the list.");
            return;
        }
        System.out.println("\nList of Employees:");
        for (Employee emp : employees) {
            System.out.println(emp);
        }
    }
}
```

4. Output:



```
Employee Management System
1. Add Employee
2. Update Employee
3. Remove Employee
4. Search Employee
5. Display All Employees
6. Exit
Choose an option: 1
Enter Employee ID: 1002
Enter Employee Name: Piyush sharma
Enter Employee Salary: 20000
Employee added successfully!

Employee Management System
1. Add Employee
2. Update Employee
3. Remove Employee
4. Search Employee
5. Display All Employees
6. Exit
Choose an option: 
```

5. Learning Outcomes:

- Understanding ArrayList Operations – Learn how to insert, search, delete, and display elements in an ArrayList.
- User Input Handling – Gain experience in handling user input using the Scanner class.
- Control Structures – Implement decision-making using switch-case and loops for menu-driven programs.
- Exception Handling Awareness – Learn to handle potential input errors, such as invalid choices.
- Practical Java Application – Develop a real-world application demonstrating list manipulation and dynamic data storage