Experiment 4.1

Student Name: Muskan Pandey **UID: 22BCS12593**
Branch: CSE **Section/Group: 22BCS_IOT-618/B**
Semester: 6th **Date: 21-02-25**
Subject: Java **Subject Code: 22CSH-359**

1. **Aim:** Create an ArrayList that stores employee details (ID, Name, and Salary). Allow users to add, update, remove, and search employees.

2. **Objective:** The objective of this program is to store and manage employee details (ID, Name, and Salary) using ArrayList in Java. The program allows users to:

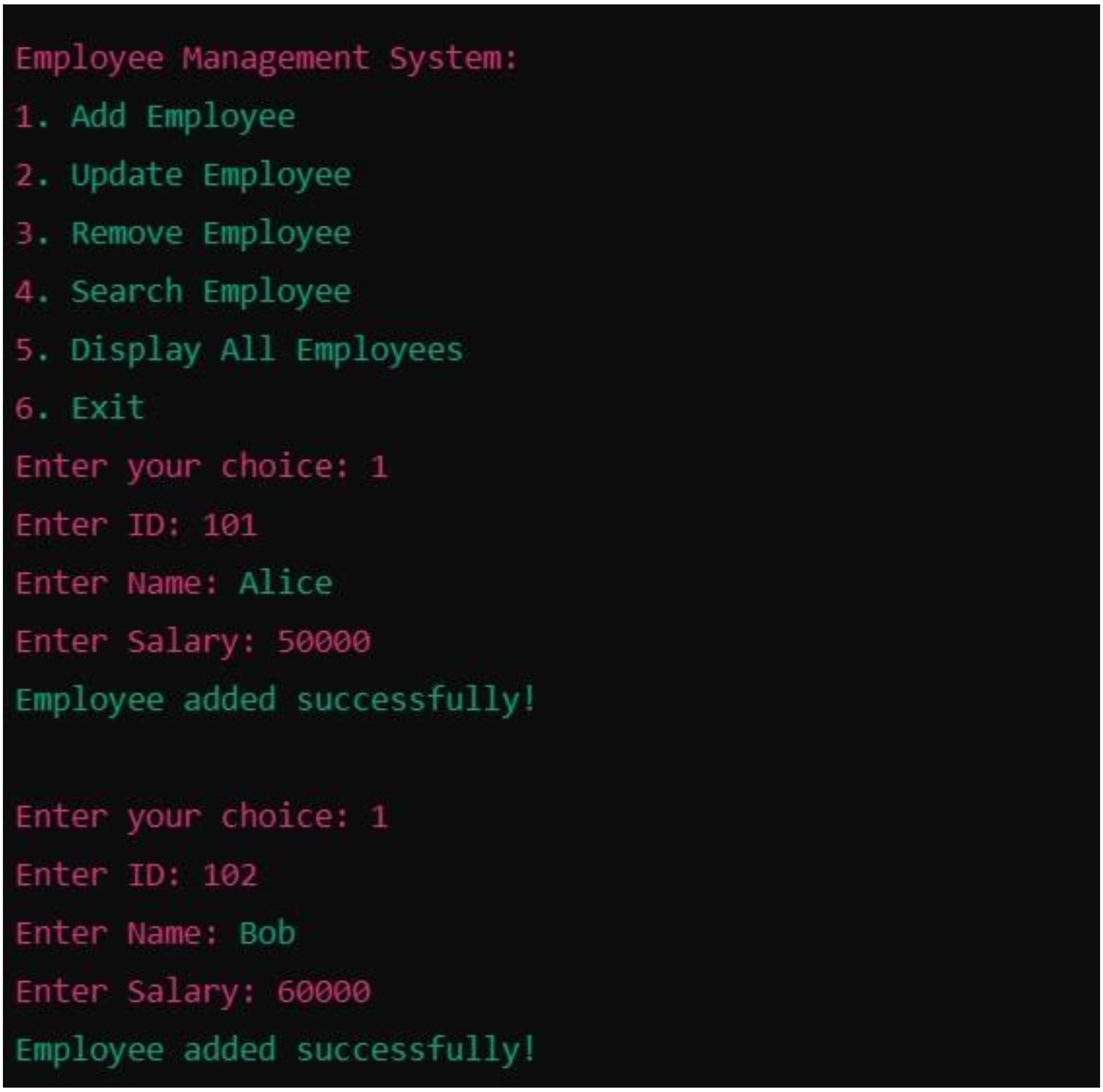
Add a new employee.
Update an existing employee's details.
Remove an employee by ID.
Search for an employee by ID. Display all employees.

3. **Code:**

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

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

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

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

    public static void main(String[] args) {
        while (true) {
            System.out.println("\n1. Add Employee\n2. Update Employee\n3. Remove
Employee\n4. Search Employee\n5. Display All\n6. Exit");
System.out.print("Choose an option: ");
            int choice = scanner.nextInt();             switch
(choice) {
                case 1: addEmployee(); break;
case 2: updateEmployee(); break;                  case 3:
removeEmployee(); break;                 case 4:
searchEmployee(); break;                case 5:
displayEmployees(); break;
                case 6: System.exit(0);
                default: System.out.println("Invalid choice! Try again.");
            }
        }
}

    static void addEmployee() {
System.out.print("Enter ID: ");         int id =
scanner.nextInt();         scanner.nextLine();
        System.out.print("Enter Name: ");
        String       name       =       scanner.nextLine();
System.out.print("Enter  Salary:  ");            double
salary = scanner.nextDouble();
        employees.add(new Employee(id, name, salary));
        System.out.println("Employee added successfully!");
    }

    static void updateEmployee() {
        System.out.print("Enter ID to update: ");
        int id = scanner.nextInt();
        for (Employee e : employees) {
if (e.id == id) {
scanner.nextLine();
                System.out.print("Enter New Name: ");
e.name = scanner.nextLine();
                System.out.print("Enter New Salary: ");
e.salary = scanner.nextDouble();
                System.out.println("Employee updated successfully!");
return;
            }
        }

System.out.println("Employee not found!");
    }
```

```
  static void removeEmployee() {
    System.out.print("Enter ID to remove: ");
    int id = scanner.nextInt();      employees.removeIf(e ->
e.id == id);
    System.out.println("Employee removed successfully!");
  }

  static void searchEmployee() {
    System.out.print("Enter ID to search: ");
    int id = scanner.nextInt();
    for (Employee e : employees) {
if (e.id == id) {
System.out.println(e);
        return;
      }
    }
    System.out.println("Employee not found!");
  }

  static void displayEmployees() {      if
(employees.isEmpty()) {
      System.out.println("No employees found!");
    } else {
      for (Employee e : employees) {
        System.out.println(e);
      }
    }
  }
}
```

4. **Output:**



```
Employee Management System:
1. Add Employee
2. Update Employee
3. Remove Employee
4. Search Employee
5. Display All Employees
6. Exit
Enter your choice: 1
Enter ID: 101
Enter Name: Alice
Enter Salary: 50000
Employee added successfully!

Enter your choice: 1
Enter ID: 102
Enter Name: Bob
Enter Salary: 60000
Employee added successfully!
```