**Employee Management System**

1. Add employee informations in a database table EMPLOYEE using a menu driven approach.
2. Add 2) Delete by Id 3) Display All 4) Update by id

5) Exit.

You must able to enter to perform add, delete, display all and update by id any number of time until you enter option 5

On Input 1 the program should ask for employee id, employee name, employee salary and employee designation, an employee instance would take all these values and stored in the database table.

After the successful store a main menu must appear which asks for same options as above, the main menu repeats for all the options exception for 5th options, on entering 5th option you must exit the main menu and display “Thank you for visiting”

On Input 2 on main menu ask for the id at runtime delete the employee based on id in the EMPLOYEE table and main menu repeats for the same 5 options

On Input 3 on main menu display all the employees using iterator and main menu repeats for the same 5 options

On Input 4 on main menu ask for an id and open a sub menu as

1. Name b) Salary c) Designation d) Exit

On option (a) update the name of a particular employee id by entering a new name and on option (b) update the salary of a particular employee id by entering a new salary on option (c) update the designation of a particular employee id and on option (d) this sub menu exits and show the main menu for the same 5 options

On Input 5 the main menu would exit and display “Thank you for visiting”.

Classes to create

Employee.java>> 4 variables, setters & getters

EmployeeDAO.java >> Performs CRUD operations based on option you select

EmployeeService.java >> calls to DAO methods

EmployeeMain.java >> main method

EmployeeRowMapper >> implements RowMapper and mapRow() returns Employee object

Note: Only main method must have Scanner instance and System.out.println(), you must not ask for input in either Service or DAO class

Same Output:

\*\*\*\*\*\*\*\* Employee Management System \*\*\*\*\*\*\*\*

1. Add 2) Delete by Id 3) Display All 4) Update by id 5) Exit

Input: 1

\*\*\*\* Performing Insert \*\*\*\*

Enter Id

111

Enter Name

Alexandar

Enter Salary

35000.0

Enter Designation

Manager

\*\*\*\* Employee information stored successfully \*\*\*\*

\*\*\*\*\*\*\*\* Employee Management System \*\*\*\*\*\*\*\*

1. Add 2) Delete by Id 3) Display All 4) Update by id 5) Exit

Input: 2

\*\*\* Performing Delete \*\*\*

Enter id

111

\*\*\* Employee deleted successfully \*\*\*

\*\*\*\*\*\*\*\* Employee Management System \*\*\*\*\*\*\*\*

1. Add 2) Delete by Id 3) Display All 4) Update by id 5) Exit

Input: 1

\*\*\*\* Performing Insert \*\*\*\*

Enter Id

222

Enter Name

Bobby

Enter Salary

45000.0

Enter Designation

Programmer

\*\*\*\* Employee information stored successfully \*\*\*\*

\*\*\*\*\*\*\*\* Employee Management System \*\*\*\*\*\*\*\*

1. Add 2) Delete by Id 3) Display All 4) Update by id 5) Exit

Input 3:

\*\*\*\* Displaying All Employees \*\*\*\*\*\*

Id = 111, Name = Alexandar, Salary = 35000.0, Designation = Manager

Id = 222, Name = Bobby, Salary = 45000.0, Designation = Tester

Id = 333, Name = Charles, Salary = 55000.0, Designation = Manager

....... And many more based on number of records in a table.....

\*\*\*\*\*\*\*\* Employee Management System \*\*\*\*\*\*\*\*

1. Add 2) Delete by Id 3) Display All 4) Update by id 5) Exit

Input 4:

Enter Id

111

\*\*\*\*\* A sub menu to update the following \*\*\*\*

1. Name (b) Salary (c) Designation (d) Exit

Input (a):

Enter new name

Raj

\*\*\*\*\* A sub menu to update the following \*\*\*\*

1. Name (b) Salary (c) Designation (d) Exit

Input (d):

\*\*\* Sub menu exit \*\*\*

\*\*\*\*\*\*\*\* Employee Management System \*\*\*\*\*\*\*\*

1. Add 2) Delete by Id 3) Display All 4) Update by id 5) Exit

Input 5:

\*\*\*Main menu exits\*\*\*

\*\*\*\*Thank you for visiting\*\*\*\*\*