System Design Project: Employee Management System

Dustin Martin

Table of Contents

- Employee Management System, Dustin Martin

- Problem Statement

  Managing employees can be a hassle and a bit overwhelming. The Employee Management System, also known as the EMS is designed to help organize and manage employees in an intuitive, easy to use system.

- Objectives of the System

  The EMS will provide organization to employee data. Fields such as Name, Address, Department, and Pay are all examples of the type of data that will be stored and organized for the user. The user will also have the ability to edit and change the Job Title, among other fields, of the employee if they want, all in a convenient easy to use system.

- System Requirements

  Windows PC running Windows 11 or a Mac running at least MacOS 14 Sonoma.
  A webserver and a server capable of running MariaDB or PostgreSQL.

- Typical Customers

  Managers
  Supervisors
  Human Resources

Directors

- Project Planning and Development Approach

  1. Software (Front-end: HTML, Back-end: Java, Database: MariaDB or PostgreSQL)
  2. Hardware (PC or Mac, servers for HTML and the database)
  3. Network (Gigabit WiFi or Ethernet above is recommended)
  4. Approach: (Customers will enter employee information in HTML, will be processed in Java, then created or fetched in MariaDB)

- Development Plan (I like your development plan so this will be closely modeled after the example)

Week 1-2: Framework and Structure established. Connect the system and verify it works correctly. Front-end to back-end to database.

Week 3-4: Build login and layout of web page.

Week 5-7: Implement basic add and remove employee features.

Week 8: Test basic feature for mid-term.

Week 9-11: Implement edit and lookup employee features.

Week 12-14: Write test case and test implemented features.

Week 15: Record demo for Final.

Customer Problem Statement

- Problem Statement: From an HR perspective, managing employees' data can be time consuming and frustrating if you get behind in your work. We are currently using spreadsheets in Excel to store employees' information. This can lead to errors and be a hassle and bit overwhelming tracking employee's performance goals. With Employee Management System software, we can easily track, add, edit, and define certain parameters for employees.
- Glossary of Terms: All non-technical terms are self-explanatory in the Employee Management System.

System Requirements

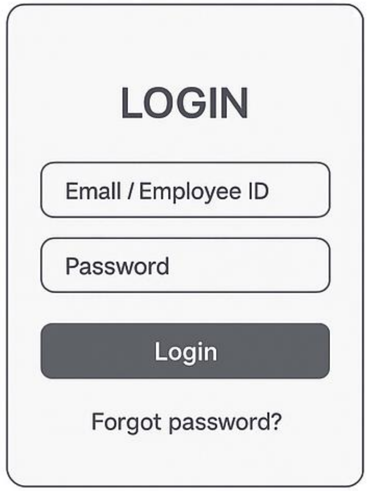
- Functional Requirements

No.	Priority Weight	Description
REQ-1 – Employee Profiles	High	Should store employee details such as name, address, job title etc. Records should have the ability to be updated.
REQ-2 – Authentication & Authorization	High	Employees should be able to login to view and edit their personal information. HR/Managers should be able to edit employee data.

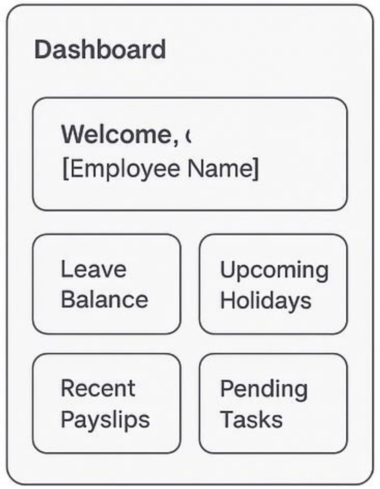
REQ-3 – Payroll Management	High	System should have the ability to track, calculate and issue paychecks.
REQ-4 - Attendance	High	Employees should be able to track attendance, request time-off days, such as sick days and vacation. Managers should be able to view requests and approve or deny.
REQ-5 – Reports	Medium	System should be able to generate reports by, employee ID, dept, etc.
REQ-6 – Performance Goal Tracking	Medium	Managers and employees should be able to set goals. Managers should be able to rate performance.

- Nonfunctional Requirements
    - Functionality – system does what it is supposed to do
    - Usability – system is easy to use
    - Reliability – system is accurate and available
    - Performance – system is fast and efficient
    - Supportability – system is flexible

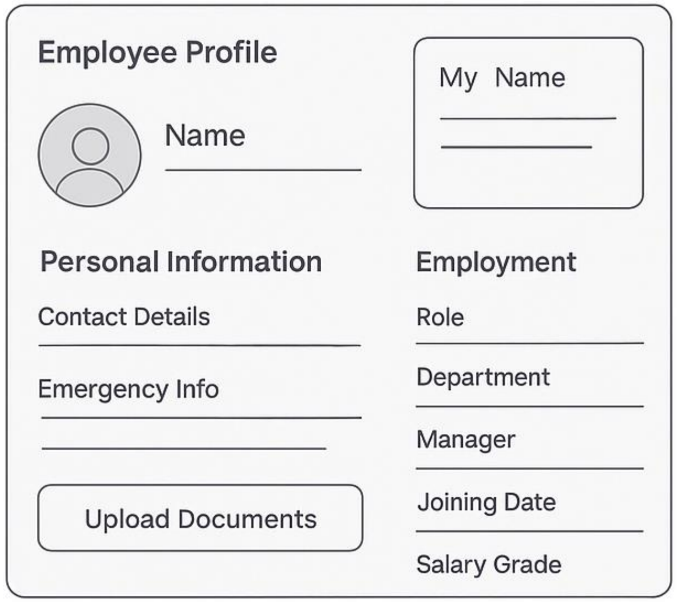
- User Interface Requirements
    - Login
        - Login screen
        - Forgot password button



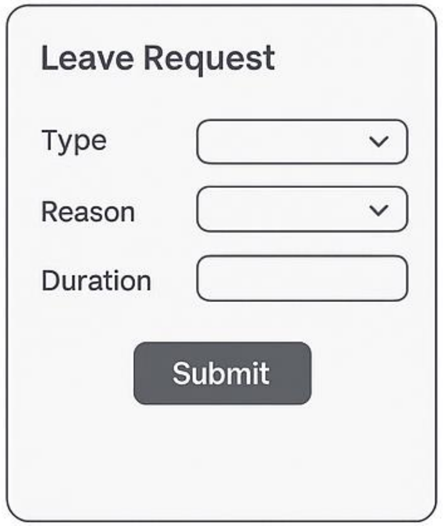
    - Employee Dashboard
        - Could view leave balance
        - Could view payslips and pending tasks



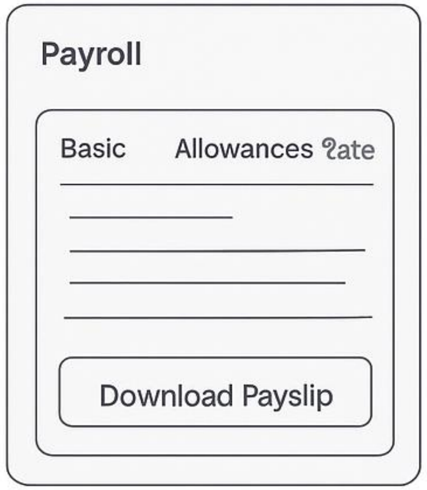
- Profile Management
  - View and update personal info
  - View employment details such as job title, dept, etc.



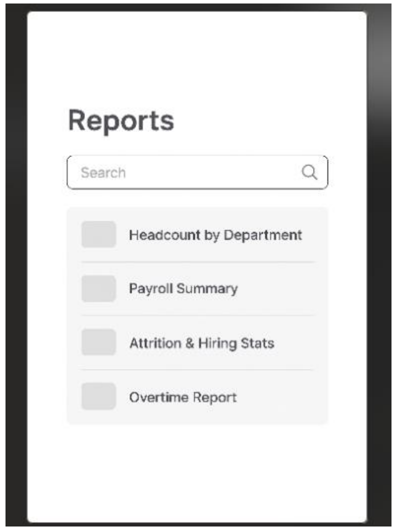
- Attendance
  - Can track timecard
  - Can request time-off such as, sick and vacation



- Payroll Management
  - View past payroll slips
  - View current salary

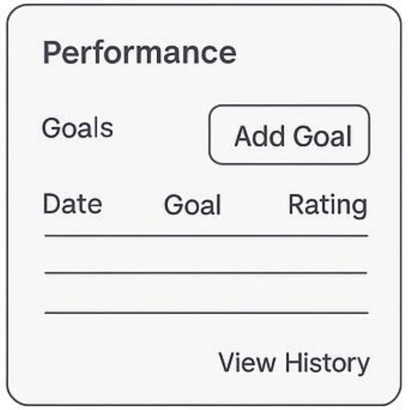


- Reports
  - View reports



- Performance Goal Tracking

- Can set goals
- View goal ratings



Plan of Work

My progress is going very slowly. I have created my MariaDB and currently in the process of linking Java to the database. I still have to link the front-end to Java. I have never done any of this before, so I have to look up every step as I go. I have not taken Web Design and Development yet. That is next semester for me. I have taken HTML a long time ago, so I think I will try to use that.

It may not be perfect, but I am trying.

General Use Cases

- Stakeholders
    - Human Resources
    - Managers
    - Employees
- Actors and Goals
    - Human Resources: They will be able to add, deactivate, modify employee's records, run reports., and modify employee's payroll.
    - Managers: They will be able to modify employee's payroll, add and edit Performance goals, and run reports.
    - Employee: They will be able to modify their own records and payroll.
    - System: Will keep track of all employees' data and assign each employee an employee ID.

- Use Cases

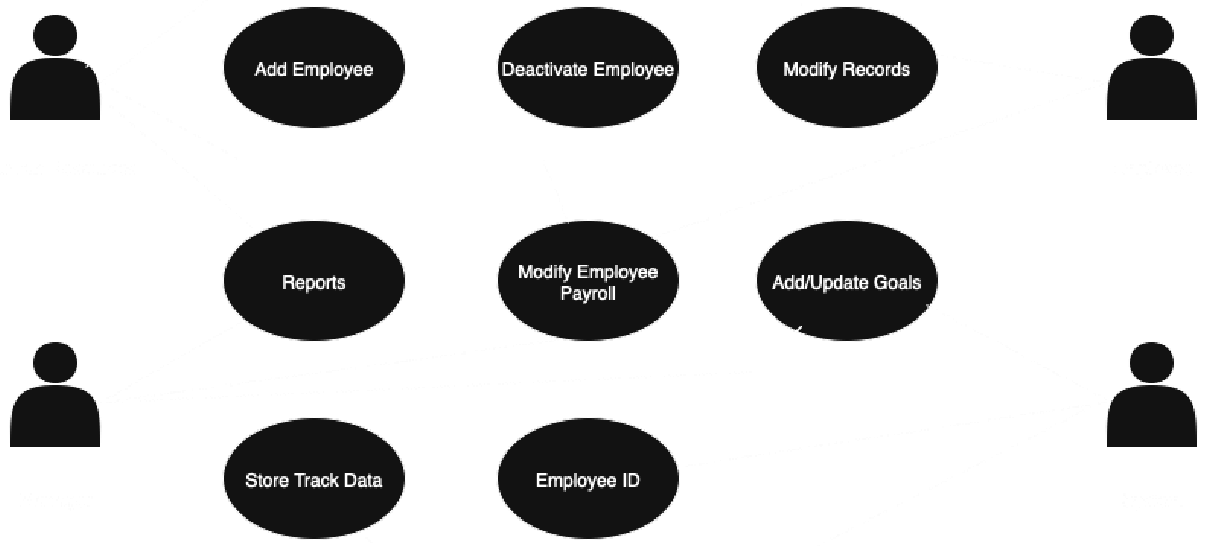
    - Human Resources (total: 10)
        - Add employee: User can add employee (2)
        - Deactivate employee: User can deactivate employee (2)
        - Modify Records: User can modify employee's records (2)
        - Modify Employees Payroll: User can chang payrate (2)
        - Reports: User can run reports on employees (2)

    - Managers (total: 8)
        - Modify Employees Payroll: User can edit timecard (2)
        - Add/Update Goals: User can update Performance Goals (4)
        - Reports: User can run reports on employees (2)

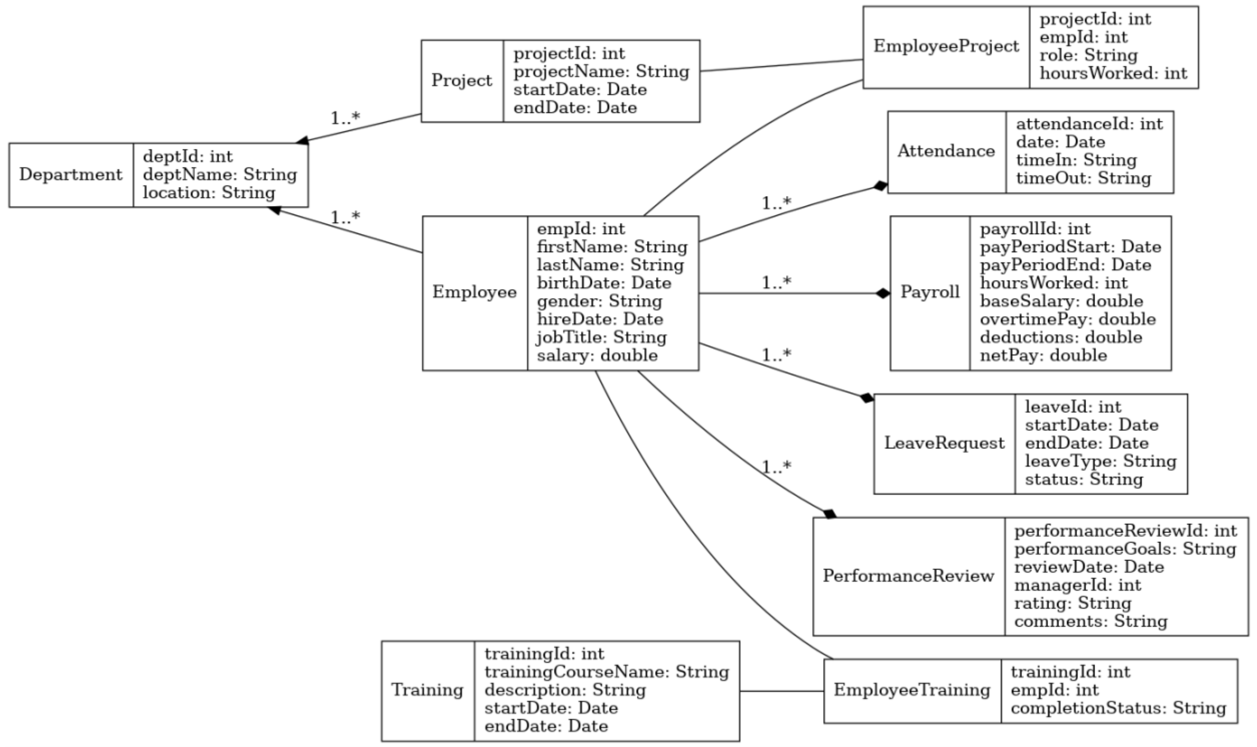
- Employee (total: 4)
    - Modify Records: User can modify own record (2)
    - Modify Employees Payroll: User can update timecard (2)

- System (total: 12)
    - Store/Track Data: System will store and track all employee's data (6)
    - Employee ID: System will generate each employee an employee ID (4)
    - Add/Update Goals: System will keep track of Performance Goals (2)

- Use Case Diagram



- Class Diagram



- Sequence Diagrams

Add new employee

Actor: Human Resources
Objects: User Interface, System, Database

1. Human Resources adds new employee information
2. User Interface verifies data, and all required fields are filled and processes request
3. System saves employee data in Database
4. Database processes save and sends confirmation to System
5. System sends success message to UI which appears on the screen for Human Resources

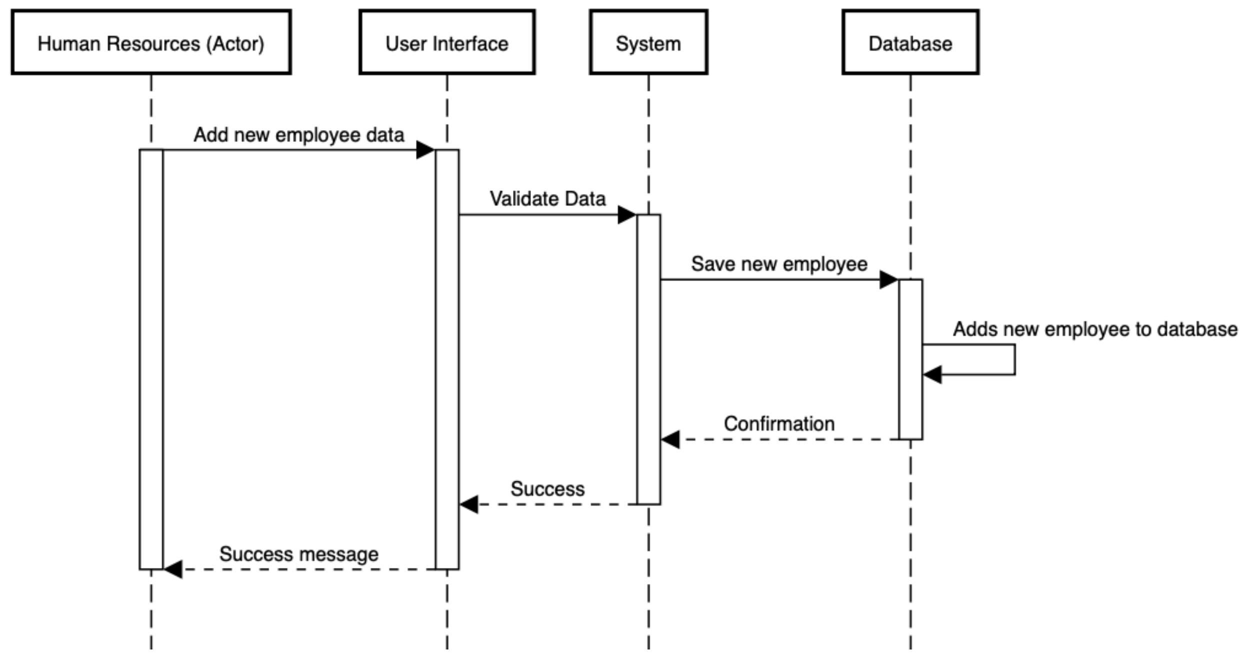
Deactivate Employee

Actor: Human Resources

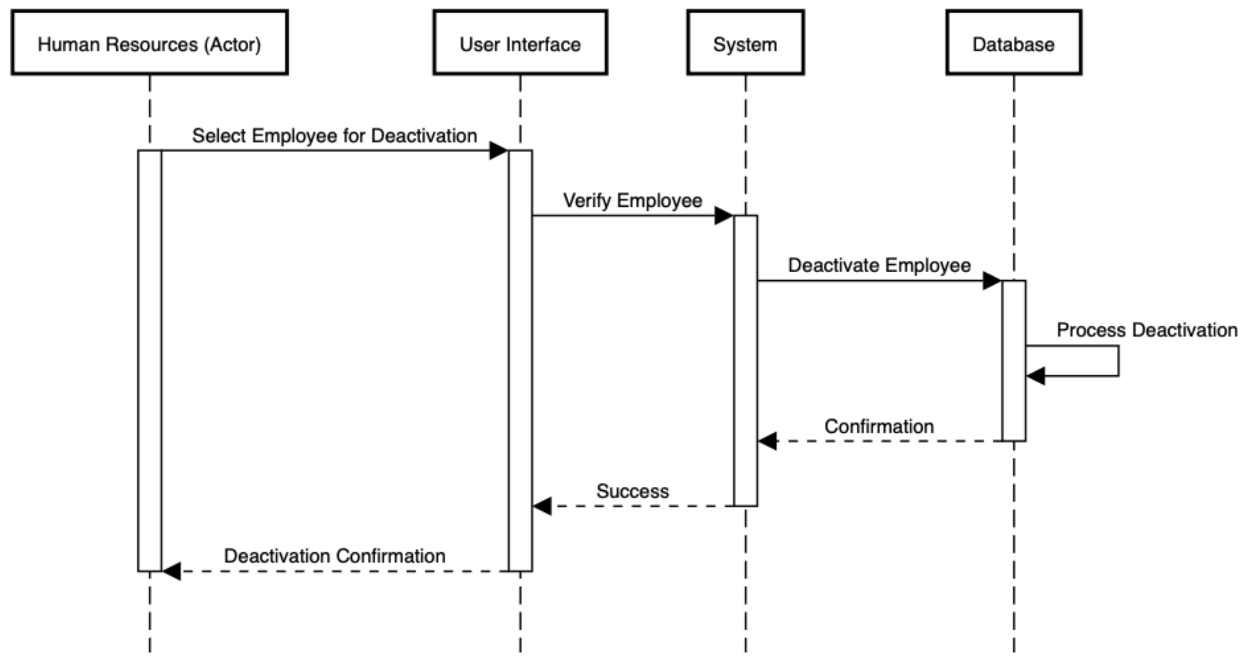
Objects: User Interface, System, Database

1. Human Resources selects employee to deactivate
2. User Interface verifies employee
3. System deactivates employee in Database
4. Database processes deactivation and sends confirmation to System
5. System sends confirmation message to UI which appears on the screen for Human Resources

Add new employee



Deactivate Employee



- Activity Diagrams

Add new Employee
States

- Initial State: Human Resources starts the process by adding new employee data
- Final State: 1. User Interface displays confirmation success message. 2. Employee data is invalid, and error message is shown to Human Resources.

Actions

Human Resources adds new employee data in the UI. The UI confirms details and submits information to the System. The System process the request and checks if the data is valid or not. The Database saves the new employee data. System sends success message to UI which shows on the screen for Human Resources.

Deactivate Employee

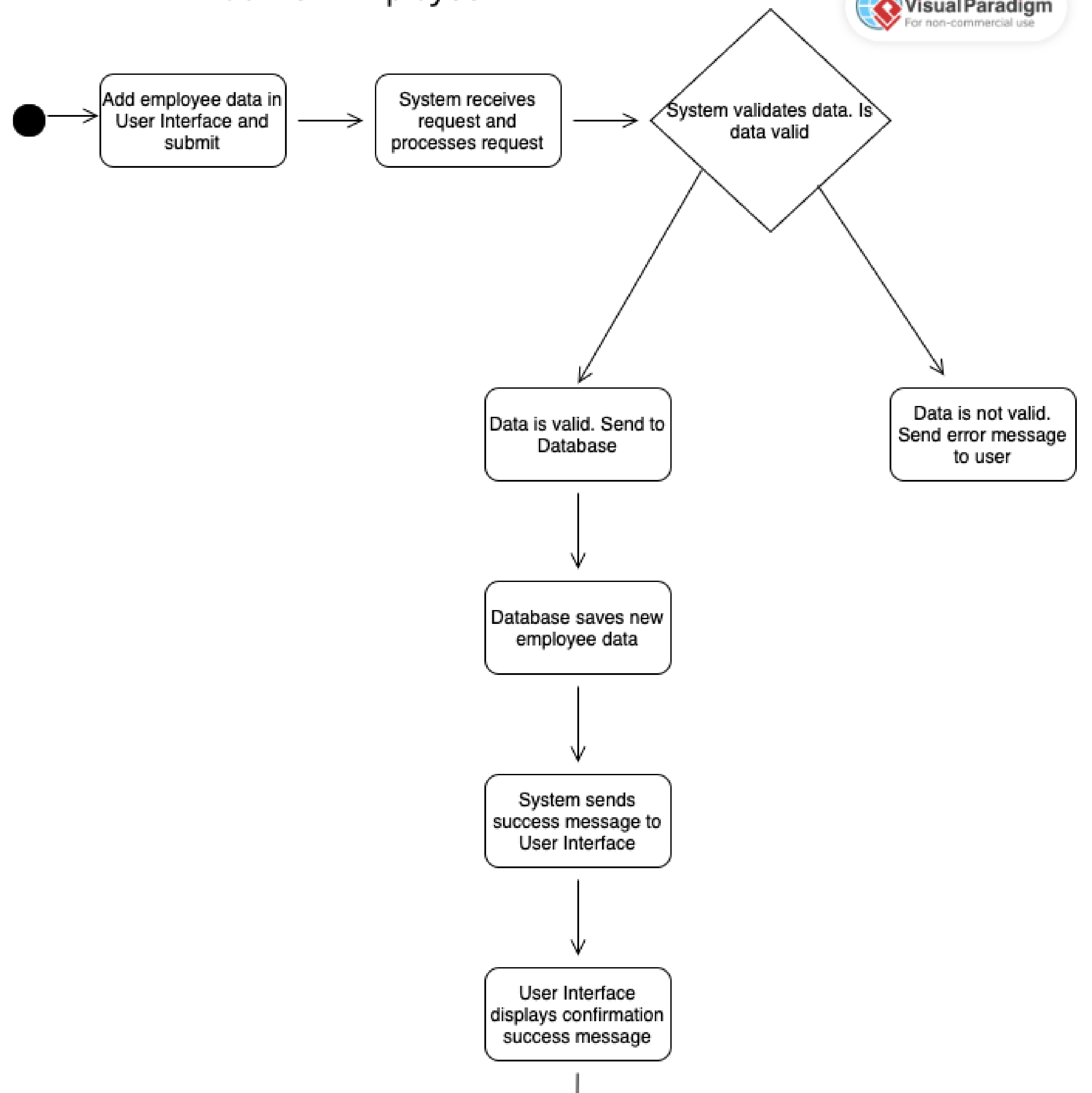
States

- Initial State: Human Resources searches and selects employee to deactivate
- Final State: 1. User Interface displays confirmation success message. 2. Employee is not found, or already inactive and error message is sent to User Interface

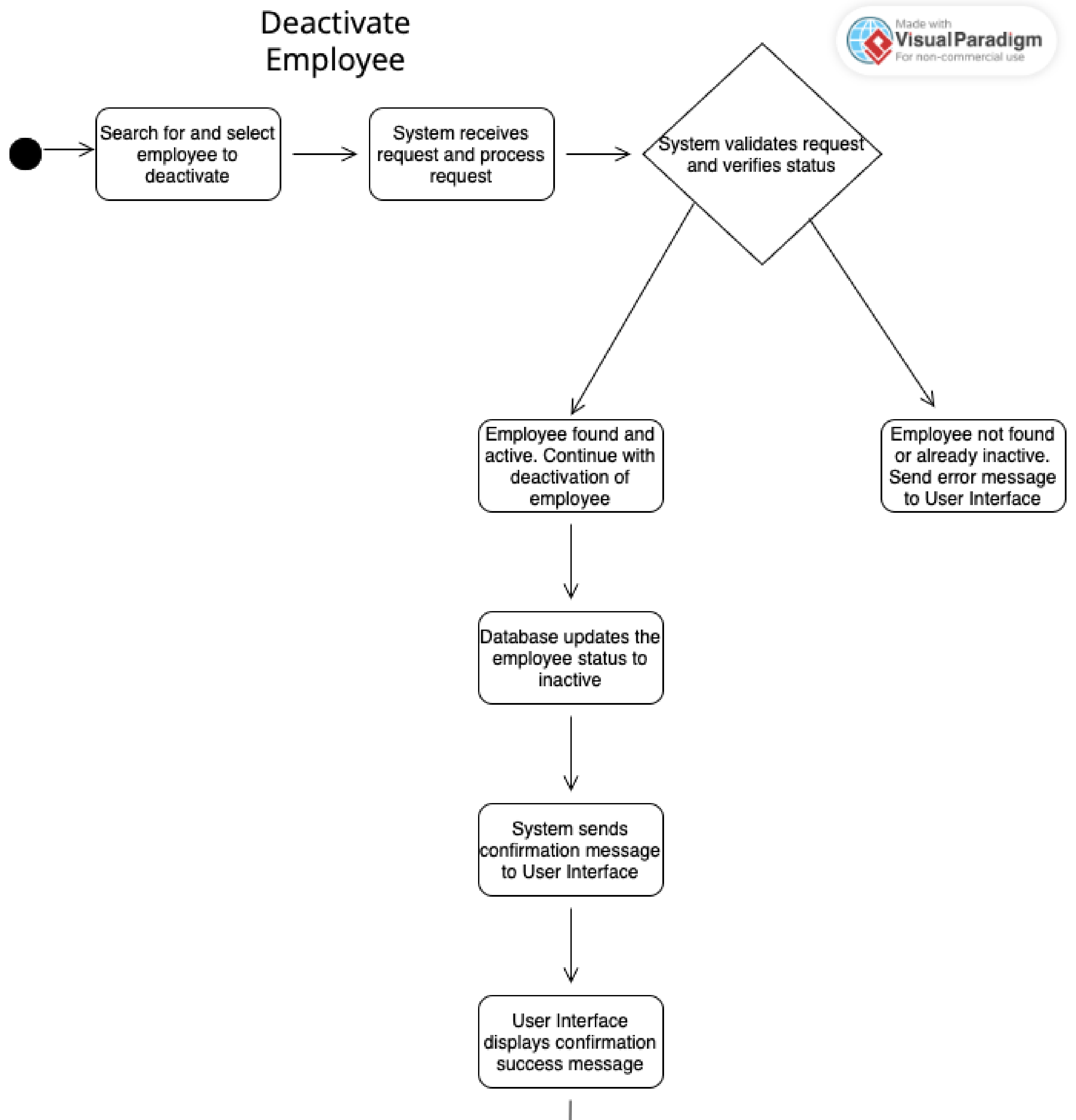
Actions

Human Resources searches for and selects employee to deactivate in the UI. The System receives request and processes it. The System validates the request and checks if the employee is found/not found and active/inactive. The Database updates the employee status to inactive. System sends confirmation message to UI which shows on the screen for Human Resources.

Add New Employee



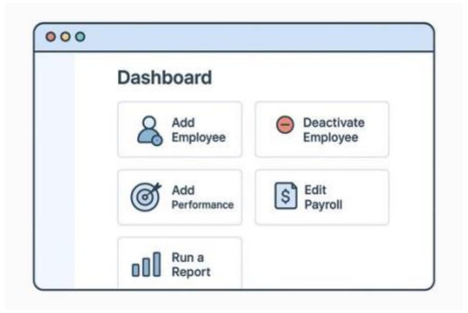
Made with Visual Paradigm For non-commercial use
Add employee data in User Interface and submit
System receives request and processes request
System validates data. Is data valid
Data is valid. Send to Database
Data is not valid. Send error message to user
Database saves new employee data
System sends success message to User Interface
User Interface displays confirmation success message


Deactivate Employee
Made with Visual Paradigm For non-commercial use
Search for and select employee to deactivate
System receives request and process request
System validates request and verifies status
Employee found and active. Continue with deactivation of employee
Employee not found or already inactive. Send error message to User Interface
Database updates the employee status to inactive
System sends confirmation message to User Interface
User Interface displays confirmation success message

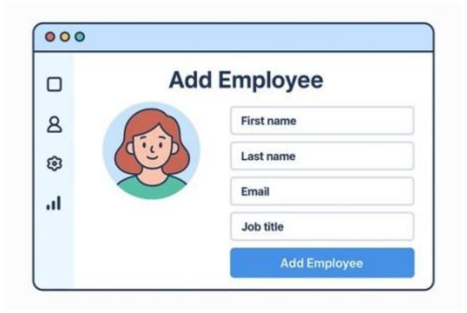
User Interface Specifications

Preliminary Design

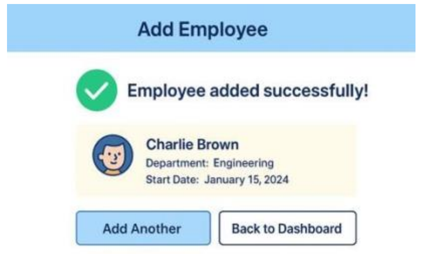
Use Case: Add Employee



Click on Add Employee



Add new employee

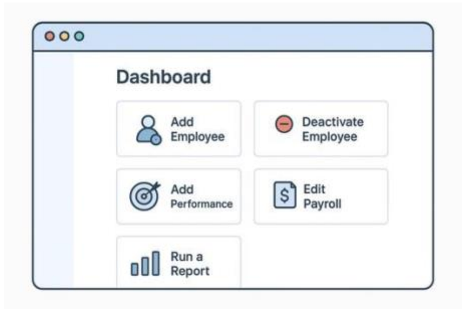

Add Employee

Employee added successfully!

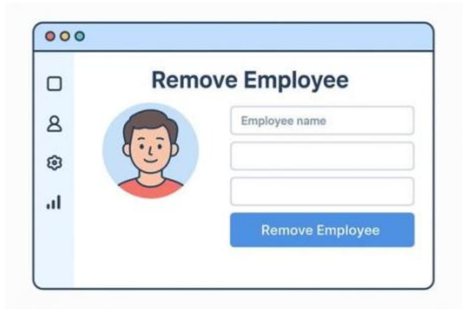
Charlie Brown
Department: Engineering
Start Date: January 15, 2024

Add employee confirmation screen

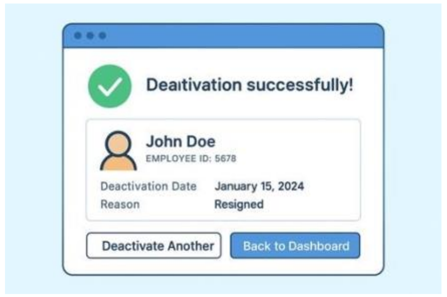
Use Case: Deactivate Employee



Click on Deactivate Employee



Select employee and click Deactivate

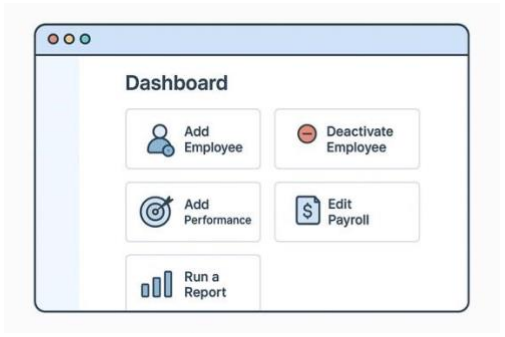

Deaitivation successfully!

John Doe
EMPLOYEE ID: 5678

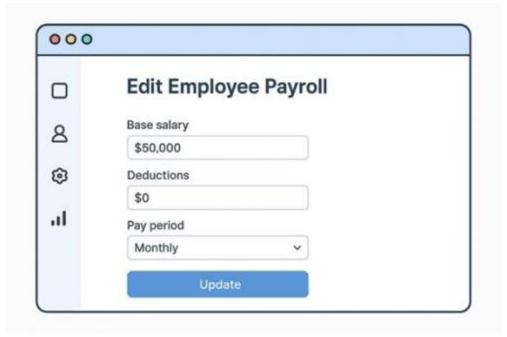
Deactivation Date January 15, 2024
Reason Resigned

Deactivation confirmation screen

Use Case: Edit Employee Payroll

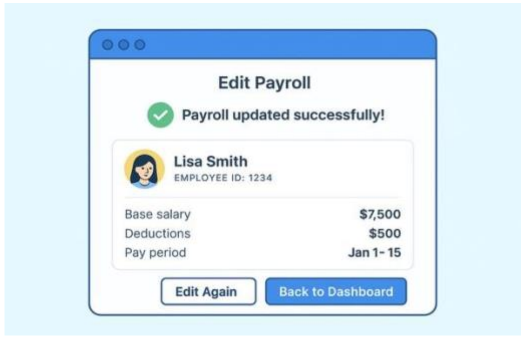


Click on Edit Payroll



Base salary $50,000
Deductions $0
Pay period Monthly

Enter information and click Update



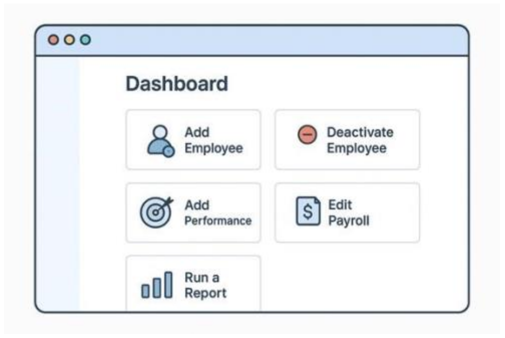
Payroll updated successfully!

Lisa Smith
EMPLOYEE ID: 1234

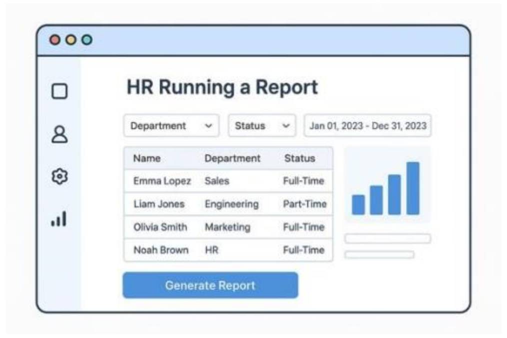
Base salary	$7,500
Deductions	$500
Pay period	Jan 1- 15

Edit Payroll Confirmation screen

Use Case: Running A Report



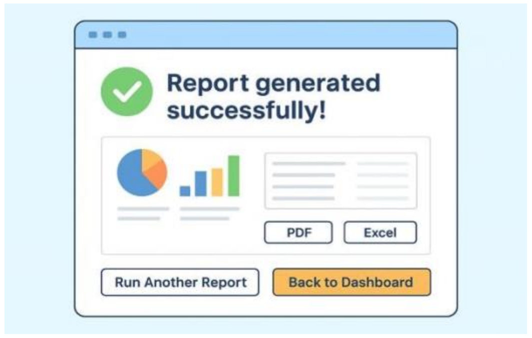
Click on Run a Report


HR Running a Report

Department | Status | Jan 01, 2023 - Dec 31, 2023

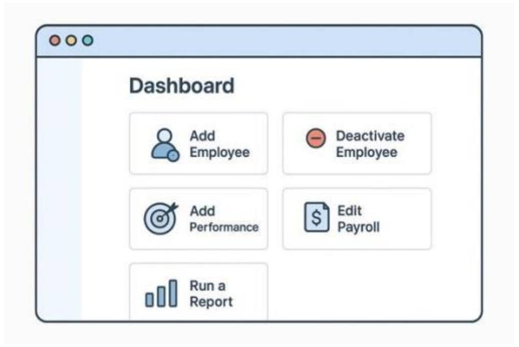
Name	Department	Status
Emma Lopez	Sales	Full-Time
Liam Jones	Engineering	Part-Time
Olivia Smith	Marketing	Full-Time
Noah Brown	HR	Full-Time

Enter criteria for report and click Generate Report

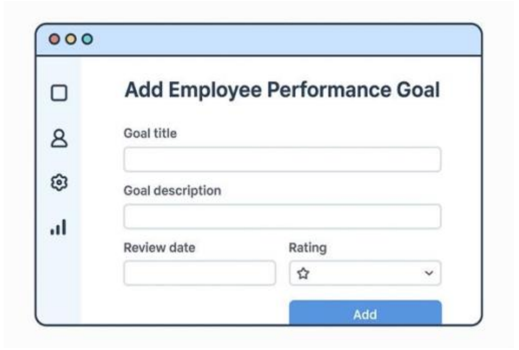

Report generated successfully!

Report generated successful screen

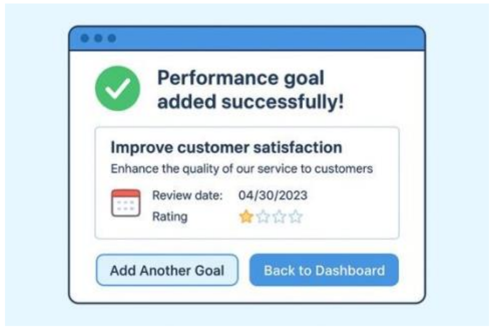
Use Case: Add Employee Goal



Click Add Performance


Add Employee Performance Goal

Enter Performance Goals and click Add


Performance goal added successfully!

Improve customer satisfaction

Enhance the quality of our service to customers

Review date: 04/30/2023

Rating ★☆☆☆☆

Performance Goals added screen

User Effort Estimation

Usage Scenario	Navigation	Clicks	Keystrokes
Add Employee	Dashboard, add employee, confirmation screen	2	<200
Deactivate Employee	Dashboard, deactivate employee, confirmation screen	2	<20
Edit Employee Payroll	Dashboard, edit payroll, change payroll info click update, confirmation screen	2	<100
Running a Report	Dashboard, run a report, enter criteria for report click generate, report generated	2	<20
Add Employee Goals	Dashboard, Add Performance Goals, enter info click add, confirmation screen	2	<200

Traceability Matrix

Req't	**PW**	UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8	UC9	UC10
REQ1 – Employee Profiles	5	X	X	X					X	X	
REQ2 – Authentication & Authorization	5			X					X		

REQ3 – Payroll Management	5				X						
REQ4 – Attendance	4							X			
REQ5 – Reports	2					X					
REQ6 – Performance Goal Tracking	3						X				X
Max PW		5	5	5	5	2	3	4	5	5	3
Total PW		5	5	10	5	2	3	4	10	5	3

System Architecture and System Design

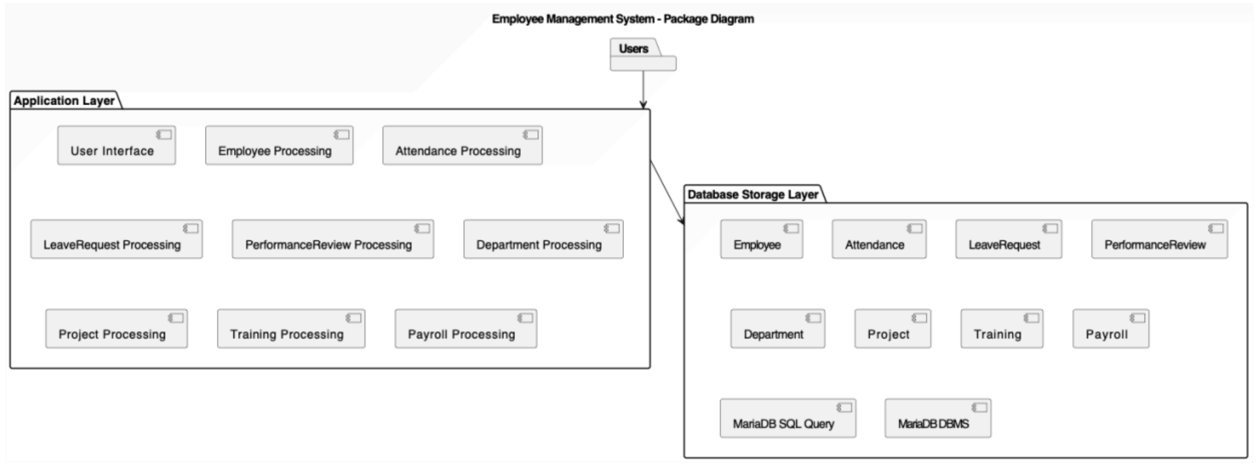
Architectural Styles

The Employee Management System is a client-server architectural style. The client logs into their account via a login screen. The technologies that I am using are HTML/CSS, Django, Python and MariaDB. I am making this on my local machine, and it will only run on one PC. All the webpages have been done using HTML/CSS. The local host will run on Django and the programming language used to power the Information System is Python. MariaDB is used to store and retrieve all the data in the database.

Identifying Subsystems

As stated earlier, the Employee Management System uses a client-server architecture and will implement a three-tier design choice. This means it will have a Database layer, Application layer and User Interface layer. The UI layer will run off HTML/CSS for the web pages and when data is inputed or a retrieve request is initiated by the user, the Application layer will send this request via SQL to the MariaDB Database where the information is processed and then returned with either a message to the user that the data has been processed successfully or the appropriate data is retrieved and shown on the screen.

UML Package Diagram:



Persistent Data Storage

The Employee Management System will save data outliving a single execution of the system. The objects stored would include Employee, Department, Project, Attendance, Training, Leave Requests, Payroll, and Performance Review data. This data will be stored in a relational database using MariaDB.

Global Control Flow

Execution Orders

The Employee Management System is an event-driven system. Whether the user wants to add or view Employee data or modify Payroll, it depends entirely on the user, and the user can initiate these actions in any order they wish.

Time Dependency

The Employee Management System is an event-response type system. The user can go to the Add Employee screen, for example, and add a new employee and once they submit it, this will update in the database and then show the new employee on the Employee list screen. As such, there are no time constraints.

Hardware Requirements

To run the Employee Management System, you will need a screen display and a PC/Mac computer. The full specs are listed below.

- Color Display with a minimum resolution of 1280 x 720 pixels
- A PC or Mac computer
- 8 GB of Ram
- Up to 1 GB of storage may be used