SOFTWARE PERSONNEL MANAGEMENT SYSTEM.

1.Problem Statement:

The CEO must must enter the name and password to login the form and select the particular employee to view the details about that employee and maintaining the employee details personally.This process of software personnel management system are described sequentially through following steps,The CEO login to the Software Personnel Management System.He/She search for the list of employees.Then select the particular employee.Then view the details of that employee.After displaying the employee details then logout.

2.Software Requirement Specification Document

2.1 Functional Requirements

A functional requirement describes an interaction between its environments.
Functional requirements describe the system function in detail, its input and output, exception etc.

The functional requirement are as follows

Analysis

In this phase, the project requirement is analyzed and the availability of the requirement is seen

Design

The design of a project is made by the project manager.

Implementation

The construction of the project is done and coding is developed.

Testing

Testing activities are made and several types of testing is carried on.

Maintenance

In this the software maintenance and the ways to avoid the drawback of the software is made.

Deployment

IT is the process of installing and kick starting the program.

2.2 Tools and Technology Requirements

Software Requirements

The following are the list of software requirements we are using to implement Software Personnel Management System.

- Operating System: Windows xp.
- Front end tool: Rational Rose Enterprise suit.
- Back end tool: Oracle 10.

Hardware Requirements:

The following are the hardware requirements with minimum configuration to get better performance of Software Personnel Management System.

- Processor : Pentium IV
- Harddisk : 40GB
- Ram : 512MB
- DVD drive : 1

Deployment Requirements:

- Front end: Java 1.8
- Technologies: JSP and JDBC
- Database: MySQL Server
- Web Server: Apache Tomcat 8.5

2.3 Non-functional Requirements :

Performance:

The performance of a Software Personnel Management System is paramount for ensuring effective human resource management within an organization. A responsive and scalable system is essential to meet the demands of a growing workforce and dynamic business requirements. Efficient database design and optimization contribute to quick and accurate data retrieval, while streamlined user interfaces enhance the overall user experience.

Maintainability:

Maintaining a Software Personnel Management System involves key practices to ensure longevity and ease of management. Employ a modular and well-documented code structure, adhering to standardized coding practices for consistency. Implement version control, ensuring the ability to track changes and roll back if needed. Design a normalized database for efficient data management, and employ robust error logging and monitoring mechanisms. Keep dependencies up-to-date and adopt configuration management for centralized settings.

Reliability:

Reliability in a Software Personnel Management System is paramount for ensuring consistent and error-free performance. To achieve this, the system should incorporate robust error-handling mechanisms, implement regular automated testing to identify and rectify potential issues, and maintain a stable and standardized codebase. A well-designed and normalized database, coupled with effective version control and documentation, contributes to the reliability of data management.

3. Design Documents

Creating design documents for a Software Personnel Management System is crucial to guide the development process and ensure a clear understanding of the system's architecture, components, and functionalities.

i Use Case Diagram:

Represents the object-oriented design of the system by showcasing classes, their attributes, methods, and relationships. This is especially relevant if you are using an object-oriented programming language.

Common Properties:

In a Use Case Diagram for a Software Personnel Management System, you typically represent the interactions between users (actors) and the system through various use cases.

Contents:

1.Actors

- ➔ Employee.
- ➔ HR Manager.
- ➔ System Administrator.

2.Use cases:

- ➔ Login.
- ➔ Personal Information.
- ➔ Manage Employee Information.
- ➔ General Reports.
- ➔ System Configuration.

3.Associations:

Lines connecting actors to use cases to represent which actors are involved in specific interactions with the system

4.Generalizations:

Employee and HR Manager use cases might inherit from a more general "User" use case.

Common Uses:

Common uses include defining the scope of the system, identifying user roles and responsibilities, and facilitating communication between stakeholders. They play a crucial role in requirements elicitation, serving as a guide for functional testing, and aiding in project planning by estimating resource requirements for different functionalities.

ii Class Diagram:

In a Class Diagram for a Software Personnel Management System, key classes may include like:

- ➔ Employee.
- ➔ Department.
- ➔ Users.

Attributes for Software Personnel Management System include:

- ➔ Employee ID.
- ➔ Name.
- ➔ Roles.

Common Uses:

Common uses of class diagrams include modeling the entities and their associations, such as Employee, Department, and Position classes, along with their respective attributes and methods. These diagrams help in visualizing the data model and database schema, aiding developers in understanding the system's architecture.

iii Sequence Diagram:

In a Sequence Diagram for a Software Personnel Management System, various actors such as administrators, employees, and HR managers interact with the system in a chronological order. The diagram visually depicts the sequence of messages exchanged between these actors and the system components, showcasing the dynamic flow of actions. For instance, an HR manager might initiate a request for employee information, triggering a sequence of interactions involving database queries, data processing, and user authentication. Sequence diagrams provide a concise overview of

how different system elements collaborate to fulfill specific functionalities, aiding developers and stakeholders in understanding the dynamic behavior of the software.

Contents:

Contents for software personnel management systems include actors like HR managers, employees, and the system itself. The diagram illustrates the flow of messages and actions, detailing the sequence of steps involved in processes such as employee onboarding, data retrieval, or authorization.

Common Uses:

They commonly depict how various actors, such as HR managers or employees, communicate with the system over time. Sequence diagrams help visualize the flow of messages, method calls, and responses, aiding developers in understanding the dynamic behavior of the system.

iv Collaboration Diagrams:

In a Collaboration Diagram for a Software Personnel Management System, actors, such as HR managers and employees, interact with system components to perform specific tasks. Lines and arrows illustrate the flow of messages between these actors and components, showcasing how information is exchanged and processed. For instance, an HR manager may initiate a request for employee information, triggering messages between the user interface, business logic, and database components. Collaboration diagrams help visualize the dynamic interactions within the system, emphasizing the communication pathways during various scenarios, ultimately aiding in the understanding and design of the system's collaborative behavior.

4.Design Phase Tool

Star UML:

StarUML is an open source software modeling tool that supports the UML (Unified Modeling Language) framework for system and software modeling. It is based on UML version 1.4,provides different types of diagram and it acceptsUML.2.0 notation.It actively supports the MDA(Model DrivenArchitecture)approach by supporting the UML profile concept and allowing to generate code for multiple languages.

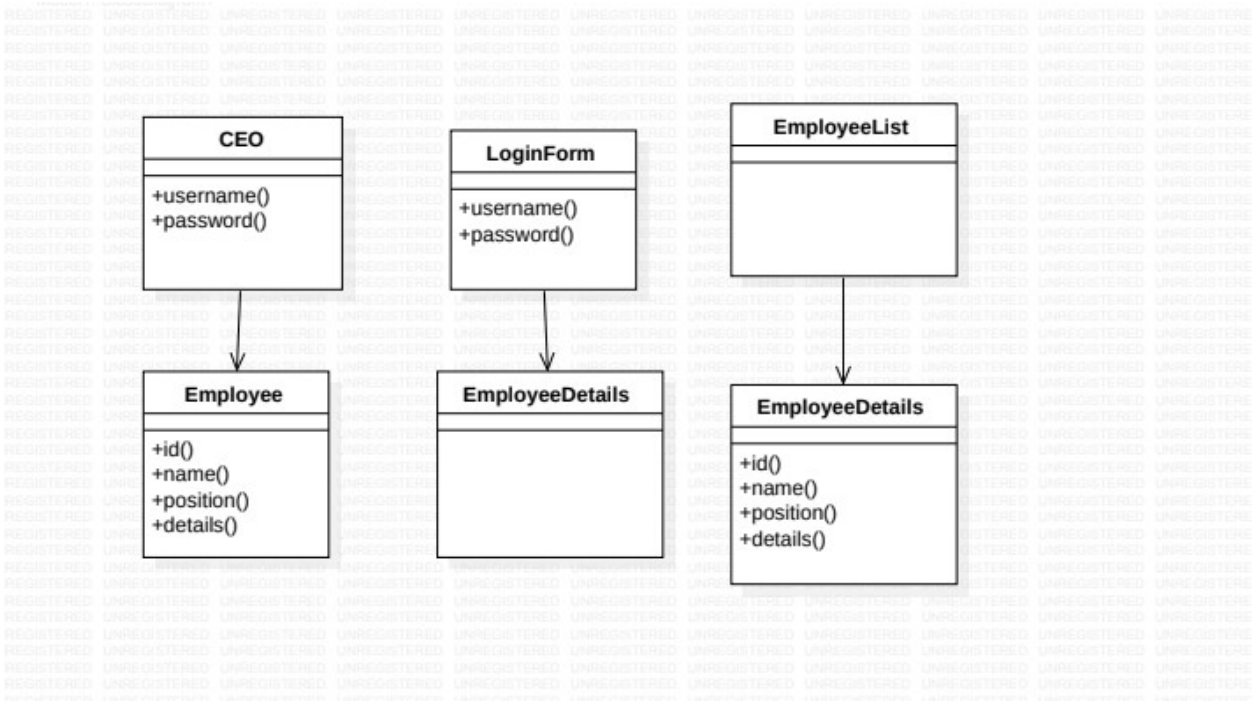
StarUML supports the following diagram types

- Use Case Diagram.
- Class Diagram.
- Sequence Diagram.
- Collaboration Diagram.
- State chart Diagram.
- Activity Diagram.
- Component Diagram.
- Deployment Diagram.
- Composite Structure Diagram.

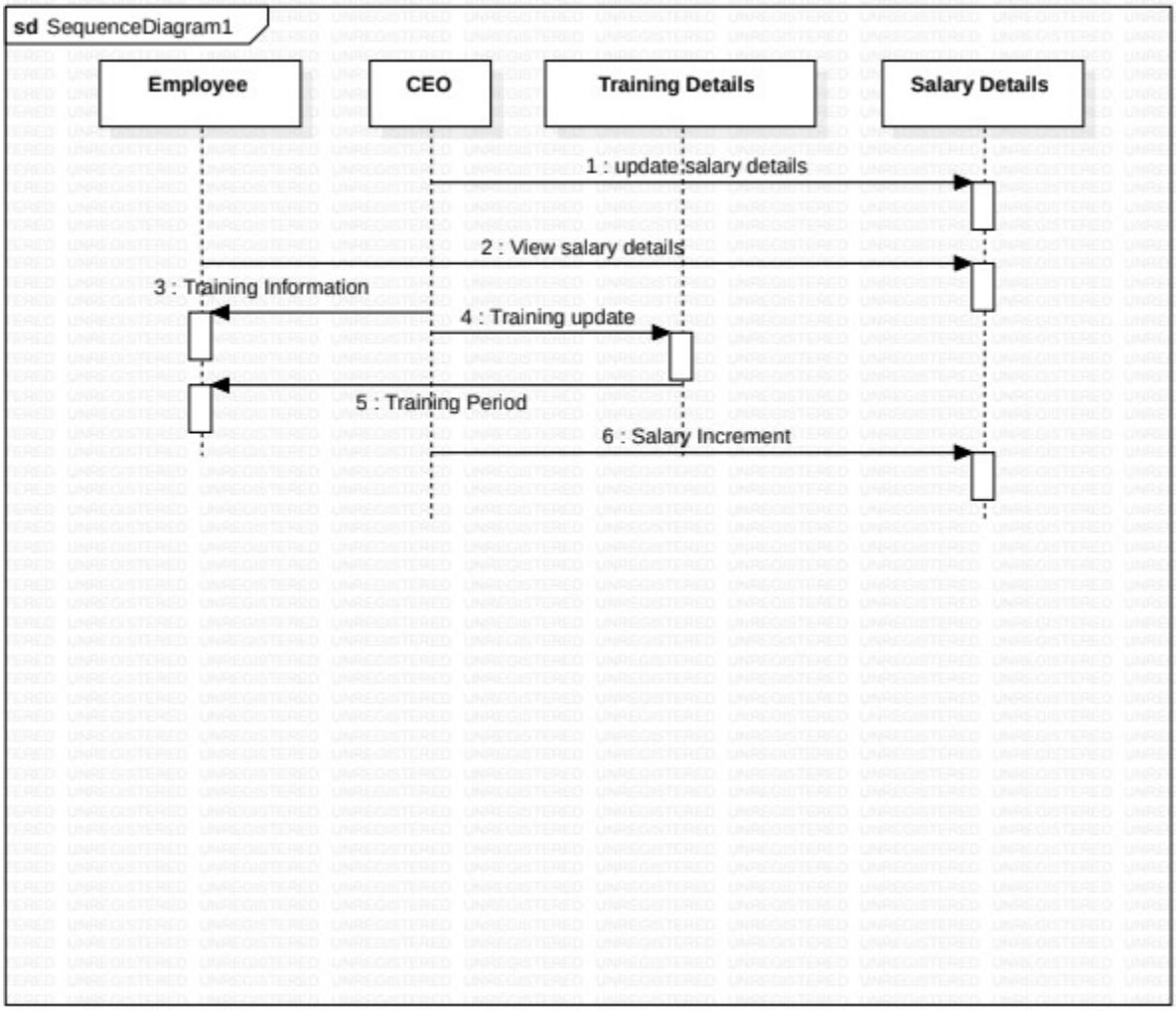
5. Design Models:

USECASE DIAGRAMS

CLASS DIAGRAM:



SEQUENCE DIAGRAM:



COLLABORATION DIAGRAM: