

PROJECT SYNOPSIS

ON EMPLOYEE MANAGEMENT SYSTEM

Submitted for fulfillment of
Advanced Java Training Program
By
Anudip Foundation



in course duration of B.Tech. (CSE)
At



Greater Noida Institute of Technology, Greater Noida

Submitted by:

Anshul
Avneesh Kaushik

Under the Guidance of :

Mr. Anuj Kumar

March, 2024

Table of Contents

Description	Page No.
1. Abstract	1
2. Introduction	2
3. Objective	3
4. System Requirements	4
i) Functional & Non-Functional Requirements	4
ii) Software & Hardware Requirements	6
5. Database Design	8
6. Methodologies	12

ABSTRACT

This project aims to develop an Employee Management System (EMS) using Hibernate JPA, Java, and MySQL. The system is designed as a console-based application intended for administrators to efficiently manage employee data within an organization. By utilizing Hibernate JPA, the system benefits from an object-relational mapping framework, ensuring seamless interaction between Java objects and the underlying MySQL database. The project focuses on providing functionalities such as adding, updating, and deleting employee records, as well as generating reports and performing various administrative tasks. Through this system, administrators can streamline their workflow, enhance data organization, and improve overall efficiency in managing employee information.

INTRODUCTION

In today's dynamic business environment, efficient management of employee data is crucial for organizations to streamline operations and enhance productivity. To address this need, we propose the development of an Employee Management System (EMS) using Hibernate JPA, Java, and MySQL. This system targets administrators within organizations who are tasked with overseeing employee-related tasks, such as managing salaries, projects assigned to them managerial roles, etc.

The primary objective of this project is to create a user-friendly console-based application that empowers administrators to efficiently manage employee data. By leveraging the Hibernate JPA framework, the system facilitates seamless communication between Java objects and the MySQL database, thereby ensuring data integrity and reliability. This approach also allows for flexible and scalable development, enabling future enhancements and modifications as per organizational requirements.

The EMS will offer a range of functionalities tailored to meet the diverse needs of administrators. These functionalities include adding new employee records, updating existing information, deleting obsolete records, and generating comprehensive reports. Additionally, the system will incorporate features such as authentication and authorization mechanisms to ensure data security and access control.

Through the implementation of this EMS, administrators can expect to experience several benefits. Firstly, the system will streamline the process of managing employee data, reducing the time and effort required for manual record-keeping tasks. Secondly, by centralizing employee information in a structured database, the system will enhance data organization and accessibility, facilitating quick retrieval and analysis when needed. Finally, the system will contribute to improved decision-making processes by providing administrators with accurate and up-to-date insights into employee-related metrics. Overall, the development of this Employee Management System represents a significant step towards modernizing administrative processes within organizations.

OBJECTIVE

The objective of this project is to develop an Employee Management System (EMS) using Hibernate JPA, Java, and MySQL. The primary goal is to create a robust system tailored for administrative use within organizations, facilitating efficient management of employee data. This includes functionalities such as adding, updating, and deleting employee records, along with generating comprehensive reports for analysis. The system aims to streamline administrative workflows, enhance data organization, and improve overall efficiency in managing employee information.

In pursuit of this objective, the project seeks to achieve several specific goals. Firstly, it aims to implement a user-friendly interface for administrators to interact with the system seamlessly. Secondly, it focuses on integrating Hibernate JPA to ensure smooth communication between Java objects and the MySQL database, enhancing data handling capabilities. Thirdly, the project aims to incorporate advanced authentication and authorization mechanisms to safeguard sensitive employee data and ensure secure access control. Additionally, the system strives to provide administrators with actionable insights derived from comprehensive reports, enabling informed decision-making and strategic planning initiatives.

Overall, the objective of this project is to develop a sophisticated Employee Management System that sets a new standard for administrative efficiency and data management practices within organizations. By leveraging modern technologies and best practices, the system aims to empower administrators with the tools they need to effectively manage their workforce, optimize operations, and drive organizational success.

SYSTEM REQUIREMENTS

I(a). Functional Requirements

1 Employee Registration:

- a. Users should be able to register new employees into the system.
- b. Mandatory fields such as name, email, contact details, and department should be captured.
- c. Optional fields may include employee ID, job title, and joining date.
- d. Validation to ensure unique email addresses and employee IDs.

2. Attendance Tracking:

- a. Ability to record attendance, including clock-in and clock-out times.
- b. Support for manual entry of attendance records for instances such as leaves, vacations, or work from home.
- c. Calculation of total work hours per day/week/month.
- d. Generation of attendance reports for individual employees and departments.

3. Salary Management:

- 1. System should facilitate the calculation and management of employee salaries.
- 2. Ability to define salary structures based on factors such as experience, position, and performance.
- 3. Automatic calculation of salaries based on attendance records and predefined salary structures.
- 4. Generation of salary slips detailing earnings and deductions.

4. Project Management:

- 1. Capability to assign employees to different projects.
- 2. Tracking of project details including project name, start date, end date, and associated team members.
- 3. Monitoring of project progress and milestones.
- 4. Generation of project reports and analytics.

5. Department Management:

- 1. System should support the creation and management of departments within the

organization.

2. Ability to assign employees to specific departments.
3. Departmental hierarchy management, including the designation of department heads.
4. Visualization of departmental structure and reporting relationships.

6. Managerial Roles:

1. Identification of managerial roles within the organization.
2. Assignment of employees to managerial positions.
3. Authorization for managers to access relevant employee data and perform managerial tasks.
4. Workflow for managerial approvals and decision-making processes.

I(b). Non-Functional Requirements

1. Security:
 - Implementation of role-based access control to restrict unauthorized access to sensitive employee data.
 - Encryption of confidential information stored in the database.
 - Protection against common security threats like SQL injection
2. Performance:
 - System should be responsive even with a large number of concurrent users.
 - Efficient database indexing and query optimization to minimize response times.
 - Load testing to ensure the system can handle peak loads during payroll processing periods.
3. Scalability:
 - Design should support easy scalability to accommodate future growth in the number of employees and system users.
 - Horizontal scalability by adding more servers or instances to distribute the load.
 - Vertical scalability by upgrading hardware resources such as CPU, memory, and storage.
4. Reliability:
 - Minimal downtime for system maintenance and upgrades.
 - Regular backups of employee data to prevent data loss in case of system failure.
 - Error handling and logging mechanisms to facilitate troubleshooting and debugging.

II(a). Software Requirements

1. Java Development Kit (JDK):

- Version: JDK 8 or higher
- Description: JDK is necessary for developing and running Java applications. Ensure compatibility with Eclipse IDE.

2. Eclipse IDE:

- Version: Eclipse IDE for Java Developers (or higher)
- Description: Eclipse IDE provides an integrated development environment for Java projects. It offers features such as code editing, debugging, and project management.

3. Hibernate Framework:

- Version: Hibernate 5.x
- Description: Hibernate is an ORM (Object-Relational Mapping) framework used to map Java objects to relational database tables. It simplifies database operations and provides a higher level of abstraction for database interactions.

4. Hibernate JPA (Java Persistence API) Implementation:

- Description: JPA is a standard specification for accessing, persisting, and managing data between Java objects and relational databases. Hibernate provides an implementation of JPA, offering additional features and functionalities.

5. MySQL Database Server:

- Version: MySQL 5.x (or higher)
- Description: MySQL is an open-source relational database management system (RDBMS) widely used for storing and managing data. It is compatible with Hibernate and provides robust features for data storage and retrieval.

6. MySQL Connector/J:

- Version: MySQL Connector/J 8.x
- Description: MySQL Connector/J is a JDBC (Java Database Connectivity) driver for MySQL databases. It enables Java applications to connect and interact with MySQL databases, facilitating data access and manipulation.

7. Apache Maven (Optional):

- a. Version: Apache Maven 3.x

- b. Description: Maven is a build automation tool used for managing project dependencies, building, and packaging Java applications. While optional, Maven can simplify project configuration and dependency management.

II(b). Hardware Requirements

- 1. Processor:
 - a. Minimum: Intel Core i3 or equivalent
 - b. Recommended: Intel Core i5 or higher
 - c. Purpose: Sufficient processing power to compile and run Java applications smoothly.
- 2. RAM:
 - a. Minimum: 4 GB
 - b. Recommended: 8 GB or higher
 - c. Purpose: Adequate memory for running Eclipse IDE and MySQL database server simultaneously.
- 3. Storage:
 - a. Minimum: 100 GB HDD/SSD
 - b. Recommended: 250 GB HDD/SSD or higher
 - c. Purpose: Sufficient storage space for Eclipse IDE installation, project files, and MySQL database storage.
- 4. Display:
 - a. Resolution: 1280x800 pixels or higher
 - b. Purpose: Clear and readable display for coding and viewing application output.

DATABASE DESIGN

The entities of the Employee management system are:-

1. Employee
2. Department
3. Manager
4. Project
5. Salary
6. Attendance

1. Employee

- Attributes:

EmployeeID (Primary Key)

FullName

Email

Phone

Gender

Address

- Relationships:

One Employee can work in one Department (Many-to-One)

One employee can work on many Projects (many-to-many relationship).

One employee can have one Salary (one-to-one relationship).

One employee can have one Attendance record (one-to-one relationship).

Each employee managed by a Manager (many-to-one relationship).

2. Department

- Attributes:

DepartmentID (Primary Key)

DepartmentName

ManagerID (Foreign Key)

Location

- Relationships:

One department can have many employees (one-to-many relationship).

Many department can have one manager (Many-to-one relationship).

3. Manager

- Attributes:

ManagerID (Primary Key)

ManagerName

Title

Email

- Relationships:

One manager can be associated with many departments (one-to-many from manager to department).

One manager can also manage one projects (one-to-one relationship).

One manager manages many Employee(one-to-many relationship).

4. Project

- Attributes:

ProjectID (Primary Key)

ManagerID(Foreign key)

ProjectName

StartDate

EndDate

Status

- Relationships:

One project can have many employees working on it (one-to-many relationship).

5. Salary

- Attributes:

SalaryID (Primary Key)

EmployeeID (Foreign key)

Salary

Bonus

Overtime

- Relationships:

Each salary record corresponds to exactly one employee (one-to-one relationship).

6. Attendance

- Attributes:

AttendanceID (Primary Key)

EmployeeID (Foreign key)

PresentDays

AbsentDays

- Relationships:

Each attendance record corresponds to exactly one employee. (One-to-one)

Table Structure

1. Employee

```
mysql> desc employee;
```

Field	Type	Null	Key	Default	Extra
employee_id	varchar(255)	NO	PRI	NULL	
address	varchar(255)	YES		NULL	
email	varchar(255)	NO		NULL	
full_name	varchar(255)	NO		NULL	
gender	varchar(255)	YES		NULL	
phone_no	bigint	NO		NULL	

6 rows in set (0.00 sec)

2. Department

```
mysql> desc department;
```

Field	Type	Null	Key	Default	Extra
department_id	varchar(255)	NO	PRI	NULL	
department_name	varchar(255)	YES		NULL	
location	varchar(255)	YES		NULL	
manager_id	varchar(255)	NO	MUL	NULL	

4 rows in set (0.00 sec)

3. Manager

```
mysql> desc manager;
```

Field	Type	Null	Key	Default	Extra
manager_id	varchar(255)	NO	PRI	NULL	
email	varchar(255)	YES		NULL	
manager_name	varchar(255)	YES		NULL	
title	varchar(255)	YES		NULL	

4 rows in set (0.00 sec)

4. Project

```
mysql> desc project;
```

Field	Type	Null	Key	Default	Extra
project_id	varchar(255)	NO	PRI	NULL	
end_date	datetime(6)	YES		NULL	
project_name	varchar(255)	YES		NULL	
start_date	datetime(6)	YES		NULL	
status	varchar(255)	YES		NULL	
manager_id	varchar(255)	YES	MUL	NULL	

6 rows in set (0.00 sec)

5. Salary

```
mysql> desc salary;
```

Field	Type	Null	Key	Default	Extra
salary_id	varchar(255)	NO	PRI	NULL	
bonus	double	YES		NULL	
overtime	double	YES		NULL	
salary	double	YES		NULL	
employee_id	varchar(255)	YES	MUL	NULL	

5 rows in set (0.00 sec)

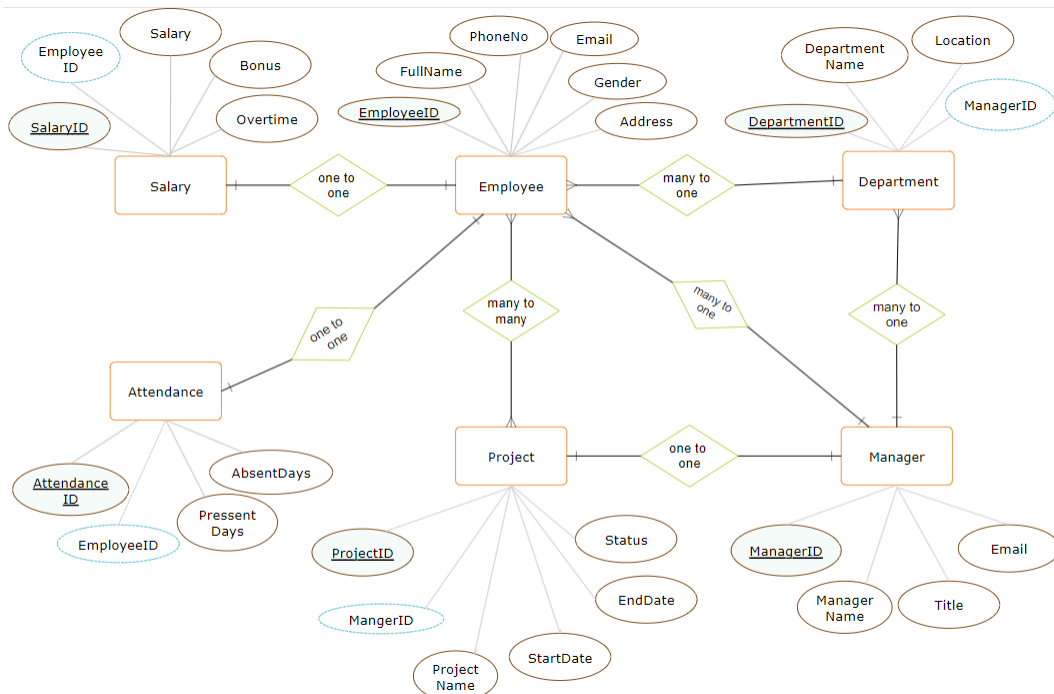
6. Attendance

```
mysql> desc attendance;
```

Field	Type	Null	Key	Default	Extra
attendance_id	int	NO	PRI	NULL	auto_increment
absent_days	int	YES		NULL	
present_days	int	YES		NULL	
employee_id	varchar(255)	YES	MUL	NULL	

4 rows in set (0.00 sec)

ER Diagram



METHODOLOGY

Object-Relational Mapping (ORM) with Hibernate:

Description: Hibernate is a popular ORM framework for Java that automates the mapping between Java objects and relational database tables. It simplifies database access and manipulation by allowing developers to work with objects instead of SQL queries directly.

Purpose: Utilizing Hibernate for ORM streamlines the development of your application by eliminating the need to write boilerplate JDBC code for database interactions. Hibernate handles the mapping between your domain entities (Employee, Project, Salary, etc.) and the corresponding database tables, reducing development time and minimizing the risk of errors.

Technologies Used:-

1. Java
2. Eclipse IDE
3. MySQL Database
4. Hibernate Framework
5. Hibernate JPA
6. JDBC (Java Database Connectivity)