**EMPLOYEE PAYROLL SYSTEM**

**A MINI PROJECT REPORT**

**CS23333- OBJECT ORIENTED PROGRAMMING USING JAVA**

Submitted by

**ASWANTH V R - 231001021**

**GOWTHAM S – 231001050**

Of

BACHELOR OF TECHNOLOGY

In

**INFORMATION TECHNOLOGY**

**DEPARTMENT OF INFORMATION TECHNOLOGY**

**RAJALAKSHMI ENGINEERING COLLEGE**

**(An Autonomous Institution)**



i

**BONAFIDE CERTIFICATE**

Certified that this project titled “**Employee Payroll Management System**” is the bonafide work of “**GOWTHAM S(231001050), ASWANTH V R (231001021)**” who carried out the project work under my supervision. Certified further that to the best of my knowledge the work reported herein does not from part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

**SIGNATURE SIGNATURE**

**Dr.P.Valarmathie Mrs. S. Usha**

**HEAD OF DEPARTMENT COURSE INCHARGE,**

**Assistant Professor(S.G)**

Department of Information Technology, Department of Information Technology,

Rajalakshmi Engineering College, Rajalakshmi Engineering College,

This mini project is submitted for CS23333 Object oriented programming using java viva voce examination held on \_\_\_\_\_\_\_\_\_ .

## INTERNAL EXAMINAR EXTERNAL EXAMINER

ii

# ABSTRACT

# The primary objective of the JDBC-powered Employee Payroll Management System in Java is to offer a streamlined and efficient platform for managing employee salary details, deductions, bonuses, and other payroll-related tasks. This system utilizes JDBC for seamless connectivity with a relational database, ensuring real-time updates on employee payment records, tax calculations, and compensation adjustments. By leveraging the power of JDBC transactions, the system maintains data integrity and consistency, preventing errors during payroll processing. With its user-friendly interface and scalability, the system is designed to meet the needs of various organizations, enhancing payroll management processes across different business environments.The Employee Payroll Management System abstracts the complexities of payroll operations into a streamlined and modular framework, enabling efficient management of organizational processes.

iii

**TABLE OF CONTENTS**

**CHAPTER TITLE PAGE NO**

**1 INTRODUCTION** 5

* 1. Purpose 6
  2. Scope of Project 7
  3. Software Requirement 7

**2 SYSTEM FLOW DIAGRAM** 8

2.1 Use-Case Diagrams 9

2.2 Entity-Relationship Diagram 10

2.3 Data Flow Diagram 11

**3 MODULE DESCRIPTION** 12

**4 DESIGN AND IMPLEMENTATION** 13

4.1 Design and Output 14

Screenshots

4.2 Database 22

4.3 Implementation- Code 23

**5 CONCLUSION** 28

28

**6 REFERENCES**

iv

# LIST OF FIGURES

**FIGURE TITLE PAGE NO**

# 2.1 Use Case Diagram 15

# 2.2 Entity Relationship Diagram 16

# 2.3 Data Flow Diagram 17

# 4.1.1 Login Page 19

# 4.1.2 Homepage 19

# 4.1.3 Employee list 20

# 4.1.4 Employee modification 20

# 4.1.5 Employee Accessing member list 21

# 4.1.6 Fields and Profession 21

v

# CHAPTER 1

# Introduction:

The Employee Payroll Management System is an advanced software solution designed to automate and modernize the management of employee compensation and payroll-related tasks. This system eliminates the need for manual processing, reducing both time and operational costs. Access is restricted through secure authentication, ensuring that only authorized personnel can manage payroll data, including salary calculations, deductions, bonuses, and other critical functions. By integrating advanced technology into payroll management, this system delivers a more efficient, accurate, and user-friendly experience, significantly improving the payroll process compared to traditional methods.

The **Employee Payroll Management System** is a sophisticated, technology-driven solution designed to streamline the critical process of managing employee payrolls within an organization. Payroll management, often regarded as one of the most essential and intricate administrative tasks, involves a multitude of operations such as salary calculations, tax deductions, benefits processing, and compliance with regulatory standards. This system addresses these complexities by automating payroll workflows and consolidating employee data, allowing organizations to focus on strategic objectives rather than operational challenges.

Developed using Java, JDBC, and MySQL database managed via XAMPP, the system is built with a modular and user-centric approach. It comprises five primary modules: *Employee*, *Payroll*, *Users*, *Tax*, and *Designation/Department*. Each module is tailored to handle specific functionalities, such as maintaining employee records, generating payrolls, enforcing security through user access controls, and managing organizational hierarchies. The integration of these modules ensures seamless data flow and efficient management of resources, reducing redundancy and minimizing errors.

The system employs advanced database normalization techniques to ensure data consistency, integrity, and scalability. Relationships between key tables are meticulously designed to enable rapid data retrieval and transaction handling. The use of role-based access control mechanisms adds a layer of security, protecting sensitive data from unauthorized access while allowing designated personnel to perform specific operations such as adding or updating records.

6

# Purpose:

The purpose of this project is to develop an efficient and user-friendly employee payroll management system that benefits both employees and administrators.

The system aims to:

* Simplify the process of managing employee salaries, bonuses, deductions, and tax calculations.
* Provide employees with easy access to their payroll details, including payslips and compensation breakdowns.
* Enable administrators to manage payroll information, employee records, and payment schedules effectively.
* Store and analyze payroll data for generating insights into company expenditures, trends, and decision-making.

# Scope of the Project:

The envisioned Employee Payroll Management System aims to seamlessly interact with administrators and effectively fulfill all proposed functionalities. Built on Java (JDBC) with a MySQL database, the system ensures efficient management of employee payroll details, reducing response time for queries and payroll processing. This project addresses the complexities associated with manual payroll processes, storing comprehensive information about employee salaries, bonuses, deductions, and tax calculations. Emphasizing features like verification, validation, security, and user-friendliness, the system strives to optimize payroll management, offering a scope that encompasses the entire spectrum of employee **compensation and financial record-keeping.**

7

# Software Requirement Specification:

Introduction:

The Employee Payroll Management System is designed to manage all aspects of employee payroll processing, including salary details, deductions, bonuses, tax calculations, and payment schedules. It serves as an automated alternative to traditional manual payroll systems, enhancing accuracy and efficiency in payroll management.

Document Purpose:

This SRS document outlines the software requirements for the Employee Payroll Management System, detailing design decisions, architectural design, and the necessary components for successful implementation. It provides an in-depth understanding of the system's structure and serves as a reference for ongoing software maintenance and support.

Product Scope:

The Employee Payroll Management System is developed for broad organizational use, aiming to replace outdated, manual payroll processes. It simplifies the payroll workflow, offering a comprehensive solution for managing employee compensation. The system is flexible, providing mechanisms for managing salary adjustments, deductions, and bonuses while ensuring compliance with tax regulations and other financial considerations.

Definitions, Acronyms, and Abbreviations:

EPS - Employee Payroll System

SRS - Software Requirements Specification

8

References and Acknowledgement:

1. [**https://www.javatpoint.com/java-awt**](https://www.javatpoint.com/java-awt)
2. [**https://www.javatpoint.com/java-swing**](https://www.javatpoint.com/java-swing)

**Overall Description:**

The Employee Payroll Management System provides authorized users with efficient access to employee payroll records, streamlining payroll processing for organizations. The system simplifies payroll operations across various businesses, ensuring accurate and timely salary calculations, tax deductions, and compensation management.

**Product Perspective:**

Utilizing a client/server architecture, the system is designed to be compatible with the Microsoft Windows Operating System. The front end is developed using Java AWT and SWING, while the backend uses MySQL for efficient data management, ensuring secure storage and retrieval of employee payroll data.

**Product Functionality:**

a) **Admin Register:** Allows the registration of new administrators who can manage payroll data.  
b) **Admin Login:** Enables existing administrators to log in securely and access the system.  
c) **Add Employee:** Facilitates the addition of new employee details, including salary and tax information.  
d) **View Employee:** Allows users to view and update existing employee payroll records.  
e) **Delete Employee:** Permits the removal of employee records from the system.  
f) **Generate Payslip:** Shift of employee allows to check the employee wages.  
g) **Update Payroll:** Allows administrators to modify salary details, bonuses, and deductions for existing employees.  
h) **Remove Admin:** Provides the option to delete specific administrator accounts.

User and Characteristics:

9

i.)**Qualification:** Users should have at least basic educational qualifications, such as a high school diploma, and should be comfortable with English.  
j.)**Experience:** Familiarity with payroll processes or HR systems is advantageous.  
k.)**Technical Experience:** Users are expected to have elementary knowledge of computers and the ability to interact with the system for optimal payroll management.

**Operating Environment:**

**Hardware Requirements**:

* Processor: Any Processor over i3
* Operating System: Windows 8, 10, 11
* Processor Speed: 2.0 GHz
* RAM: 4GB
* Hard Disk: 500GB

**Software Requirements:**

* Database: MySQL
* Frontend: Java (SWING, AWT)
* Technology: Java (JDBC)

**Constraints:**

* System access limited to administrators.
* Delete operation restricted to administrators without additional checks for simplicity.
* Administrators must exercise caution during deletion to maintain data consistency.

**Assumptions and Dependencies:**

* System administrators create and confidentially communicate login IDs and passwords to users.

10

Specific Requirements:

**User Interface:**

The Employee Payroll Management System provides user-friendly, menu-driven interfaces for:

a) **Admin Register:** Registering new administrators to manage payroll data.  
b) **Admin Login:** Logging in existing administrators to access payroll functionalities securely.  
c) **Add Employee:** Storing new employee payroll details, including salary, bonuses, and deductions.  
d) **View Employee:** Viewing and updating existing employee payroll information.  
e) **Delete Employee:** Deleting employee records from the system as needed.  
f) **Generate Payslip:** Creating payslips for employees based on salary and deductions.  
g) **Update Payroll:** Viewing and modifying existing payroll details, such as salary adjustments or bonus calculations.  
h) **Remove Admin:** Deleting specific administrator accounts for system access management.

**Hardware Interface:**

* Screen resolution of at least 640 x 480 or above.
* Compatible with any version of Windows 8, 10, 11.

11

**Software Interface:**

1. MS-Windows Operating System
2. Java AWT and SWING for designing the front end
3. MySQL for the backend
4. Platform: Java Language.
5. Integrated Development Environment (IDE): XAmpp

Functional Requirements:

1. Log in Module (LM):
   * Users (admins) access the Login Module.
   * LM supports user login with a username and password.
   * Passwords are masked for security.
   * Successful login verification by the database administrator is required for access.
2. Registered Users Module (RUM):
   * After successful login, users (admins) can navigate through the application.
   * Users can view detailed information about movies, showtimes, and reservations.
   * Users can update and maintain movie details, including modifying showtimes and seat allocations.
3. Administrator Module (AM)
   * Upon successful login, the system displays administrative functions.
   * Functions include adding and updating movie details.
   * The "Add" function allows administrators to input new movie details and remove unused entries.

12

* + The "Update" function enables administrators to modify existing movie details in the database.
  + All add, update, or delete requests trigger the AM module to communicate with the Server Module (SM) for necessary database changes.

1. Server Module (SM):
   * SM acts as an intermediary between various modules and the database (DB).
   * Receives requests from different modules and formats pages for display.
   * Validates and executes requests received from other modules.
   * Handles communication with the database, ensuring data consistency and integrity, especially regarding movie details, showtimes, and reservations.

**Non-functional Requirements:**

**Performance:**

* The system must efficiently handle real-time payroll processing requests, ensuring a response time of less than 2 seconds for salary calculations and payslip generation.
* Critical failures, such as miscalculation of salaries or deductions, must be addressed immediately to ensure smooth payroll management.

**Reliability:**

* The system is business-critical; in the event of abnormal operation or downtime, immediate action must be taken to resolve issues and restore normal functionality.

**Availability:**

* Under normal operating conditions, user requests for payroll processing, such as generating payslips or updating payroll details, should be processed within 2 seconds to maintain operational efficiency.
* Immediate feedback on payroll updates or calculations should be communicated to administrators to ensure transparency and accuracy.

13

**Security:**

* A robust security mechanism must be implemented on the server side to prevent unauthorized access, safeguard employee salary data, and ensure the integrity of the payroll management system.
* Employee privacy, including personal and financial details, must be securely stored and managed to uphold confidentiality.

**Maintainability:**

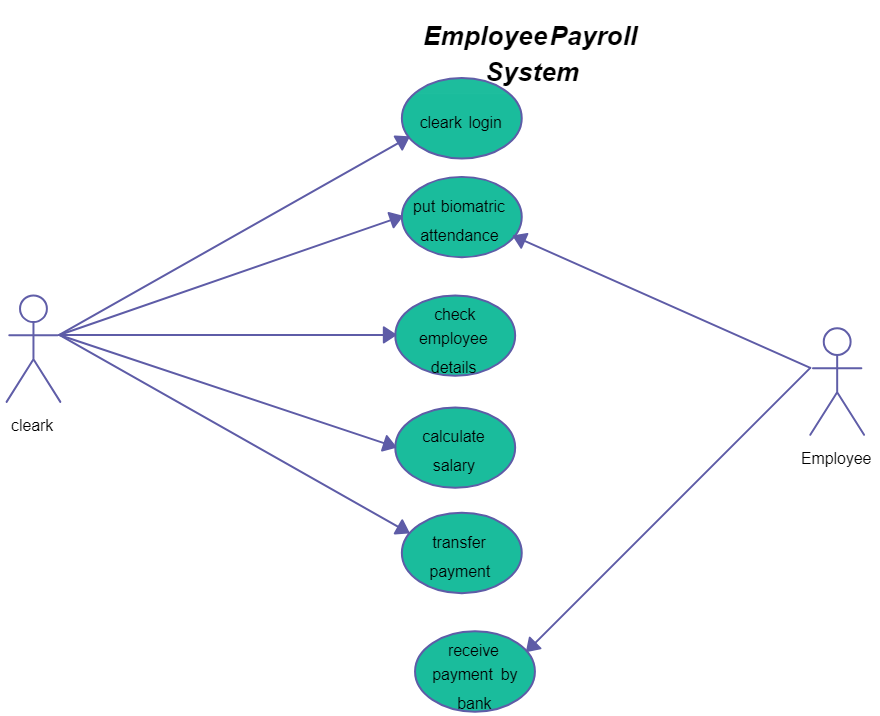
* Design documents detailing software and database maintenance procedures must be available to facilitate regular updates and modifications to the payroll management system.
* Administrative access should be provided for proper maintenance of both the front-end and back-end components, ensuring the system's long-term reliability and adaptability.

14

**CHAPTER -2**

# System Flow Diagrams:

* 1. **Use Case Diagrams :**

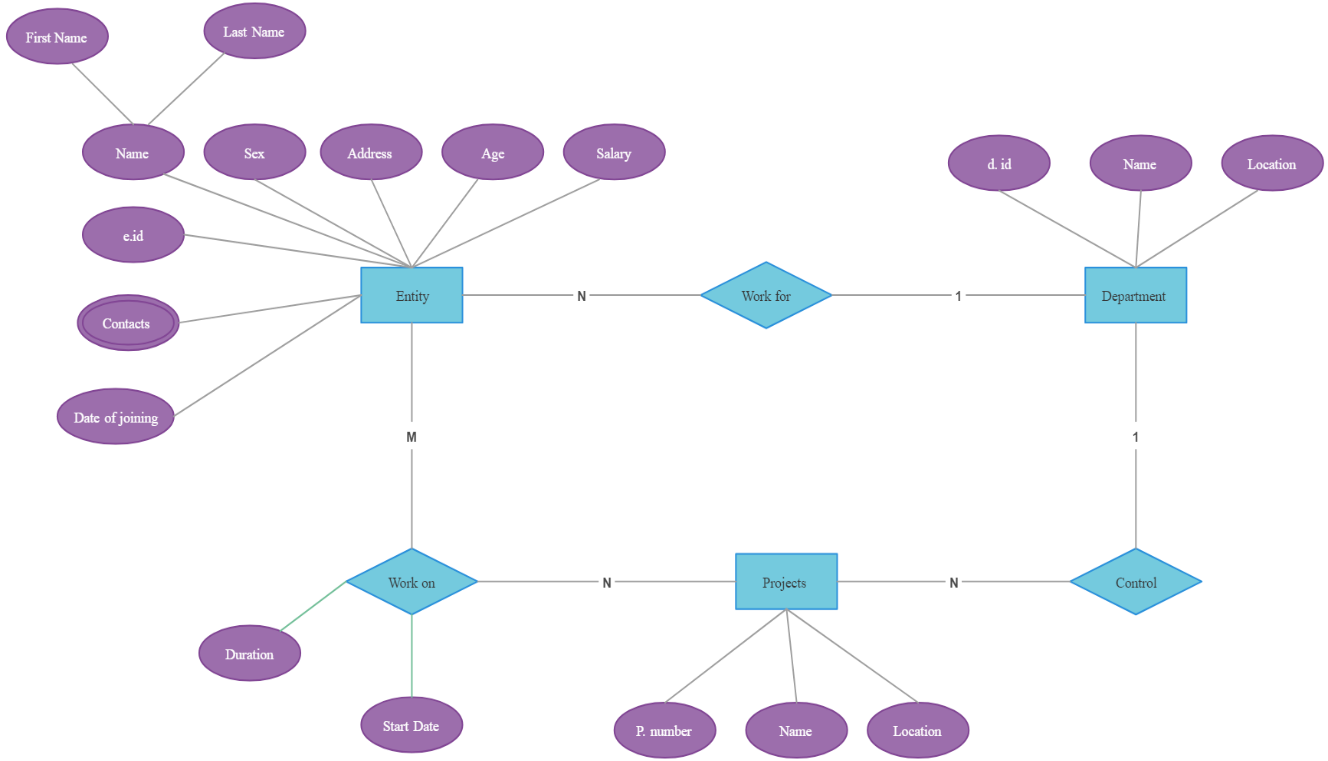
****

**Figure 2.1 Use case Diagram**

15

* 1. **Entity-relationship diagram:**

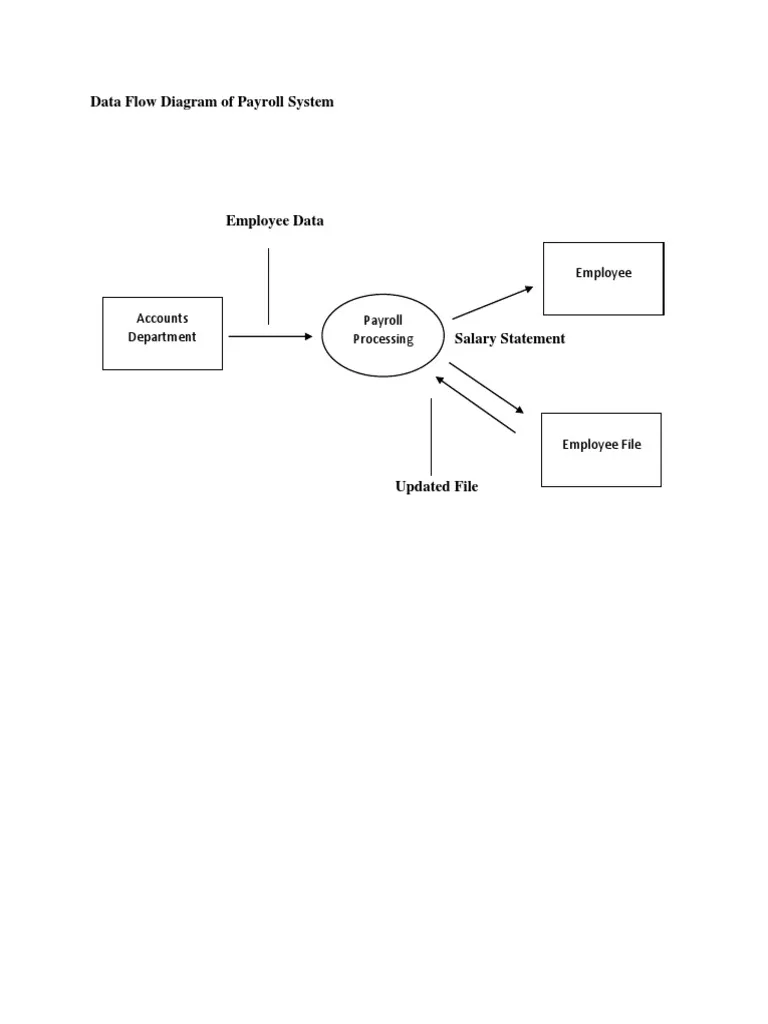
E-R (Entity-Relationship) Diagram is used to represents the relationship between entities in the table.

****

**Figure 2.2 Entity Relationship Diagram**

16

**2.3 Data-flow diagram:**

****

**Figure 2.3 Data Flow Diagram**

17

# CHAPTER 3

# Module description:

**Admin:**

Admin controls all the payment related things of the employees.

**Register:**  
Admin can register by providing a username and password for secure system access.

**Login:**  
Admin can log in using their username and password.

**After Login:**

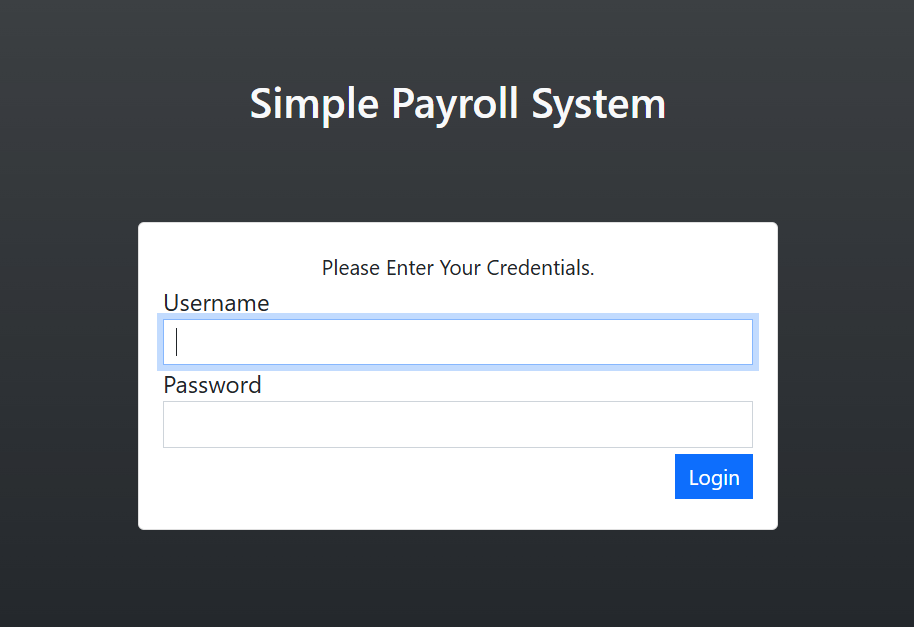
1. **Add Employee:**  
   Admin can add new employee details, including name, designation, salary, tax information, and other relevant data.
2. **View Employee:**  
   Admin can view and update employee details such as name, designation, and salary information.
3. **Delete Employee:**  
   Admin can delete employee records, removing them from the payroll system.
4. **Generate Payslip:**  
   Admin can generate a payslip for employees, including details about salary, deductions, bonuses, and net pay.
5. **Update Payroll:**  
   Admin can update payroll information, such as modifying salaries, bonuses, or deductions for employees.
6. **Remove Admin:**  
   The system allows the deletion of admin details if necessary, ensuring proper account management and security.

18

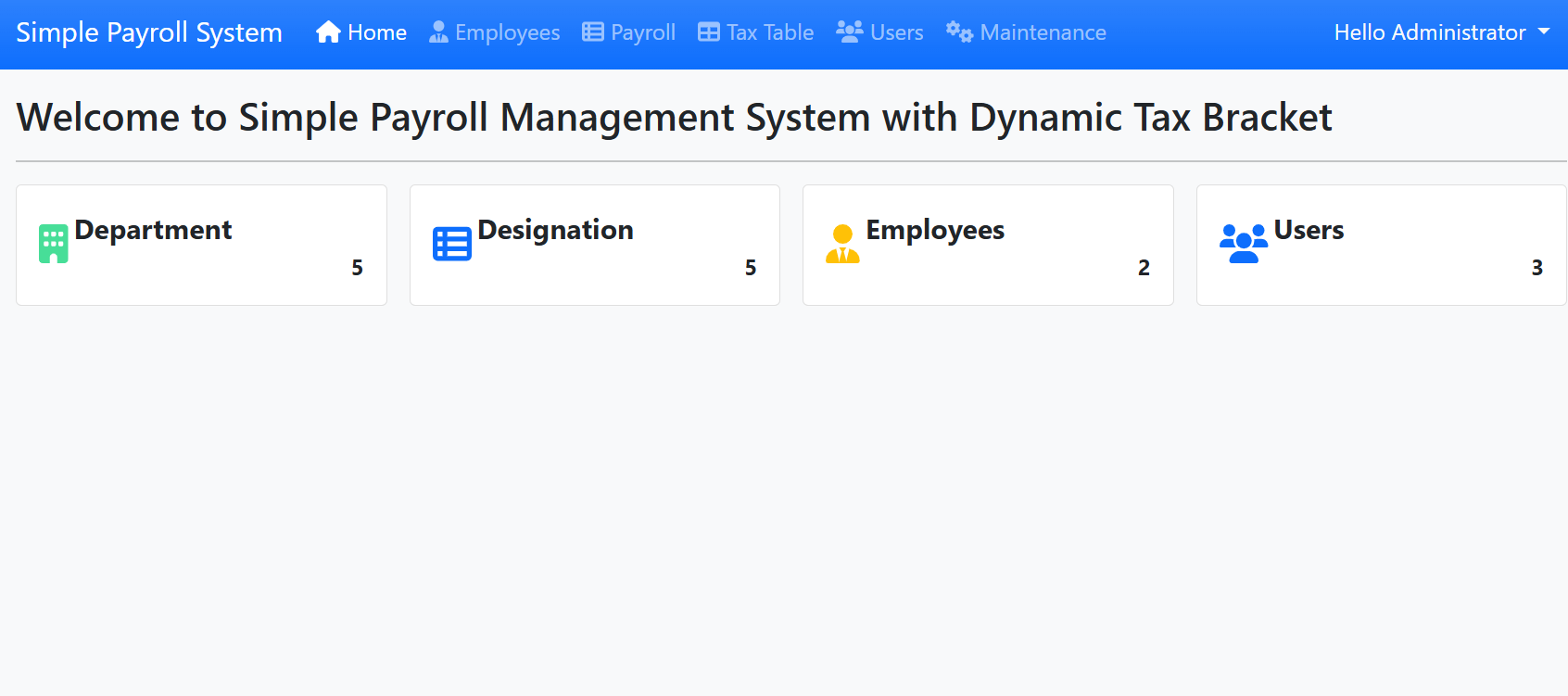
**CHAPTER 4**

**DESIGN AND IMPLEMENTATION**

* 1. **Design:**

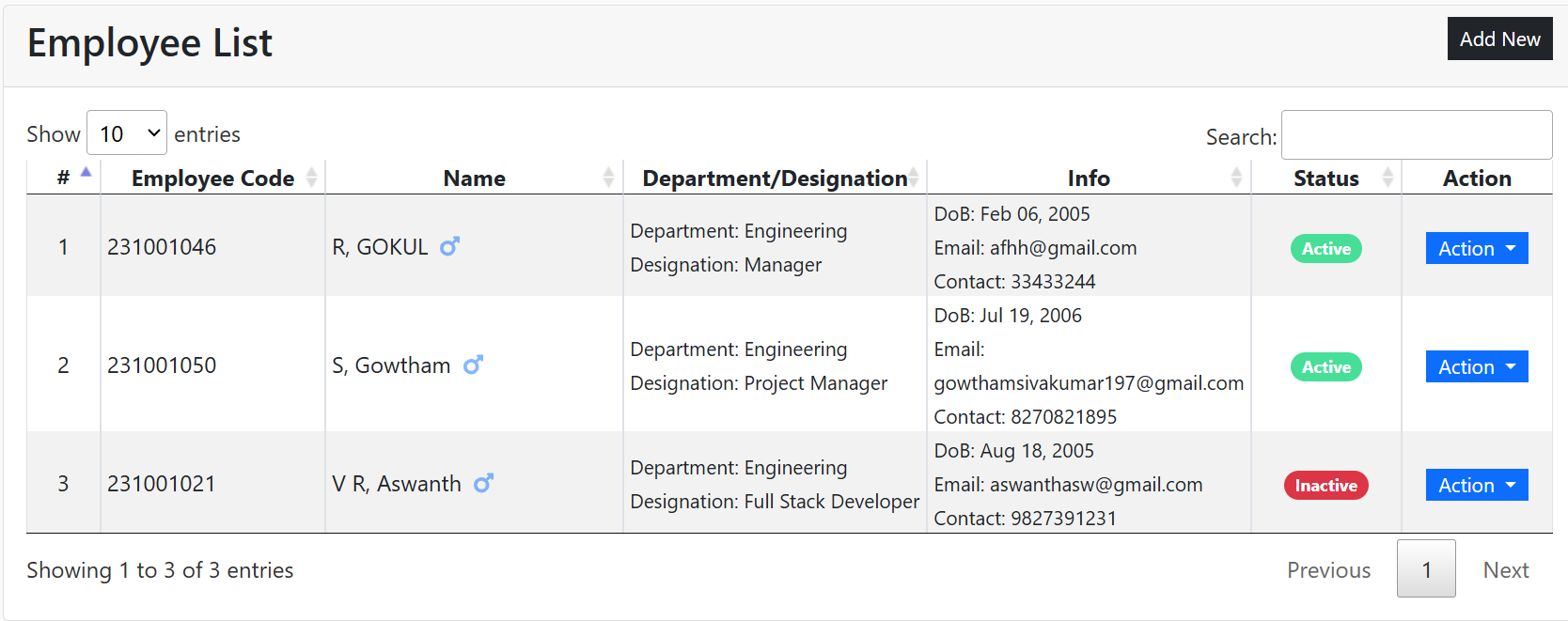


**Figure 4.1.1. Login Page**

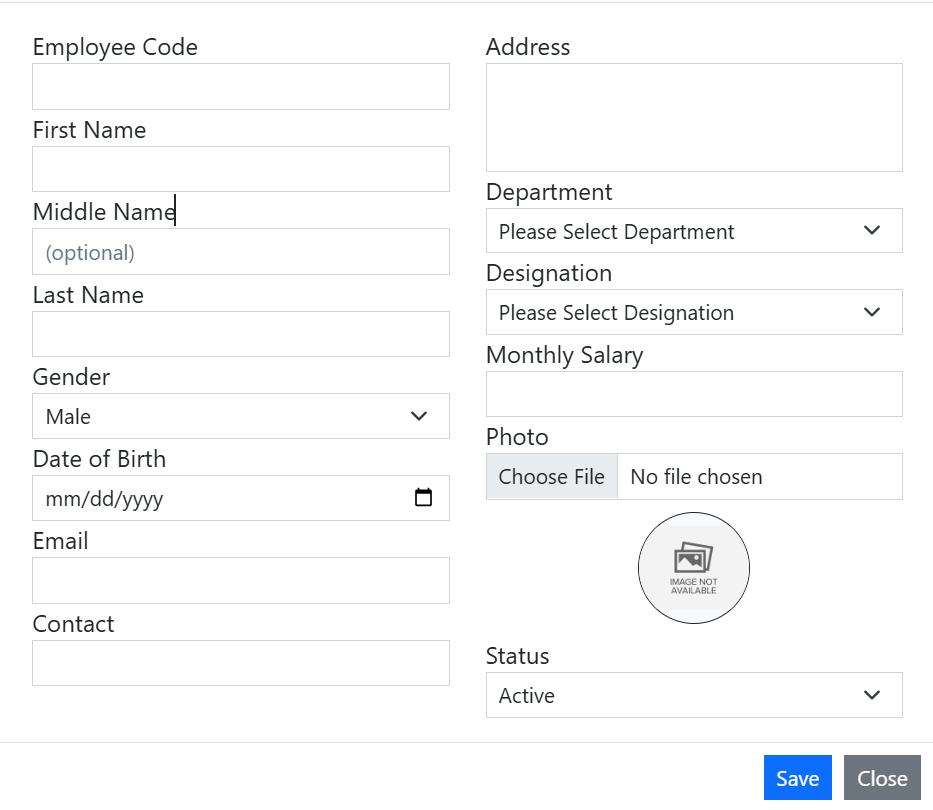


**Figure 4.1.2. Homepage**

19

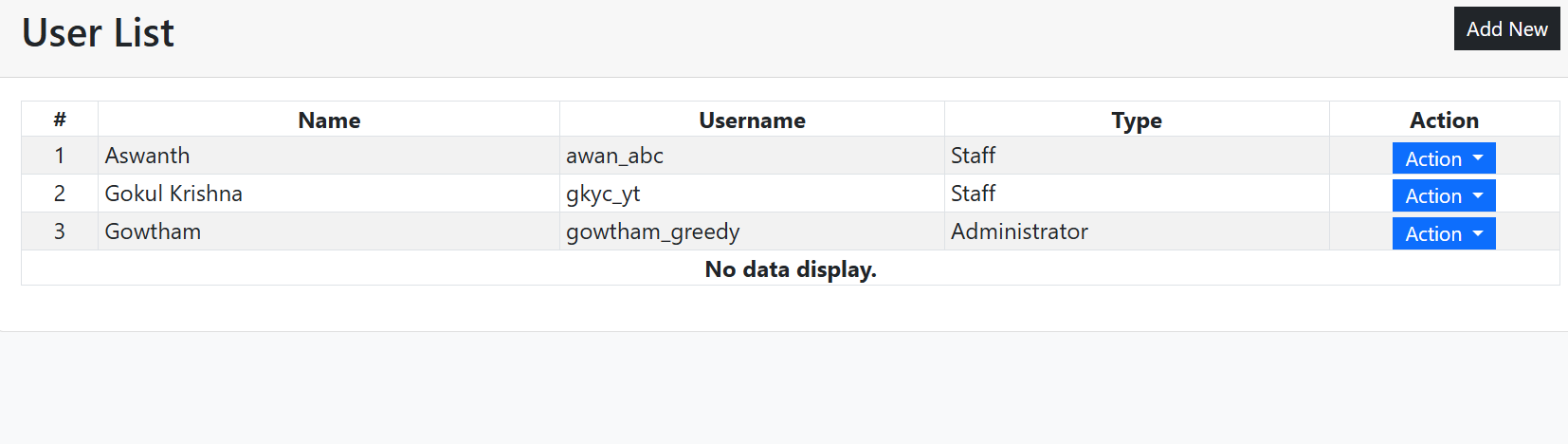


**Figure 4.1.3 Employee List**

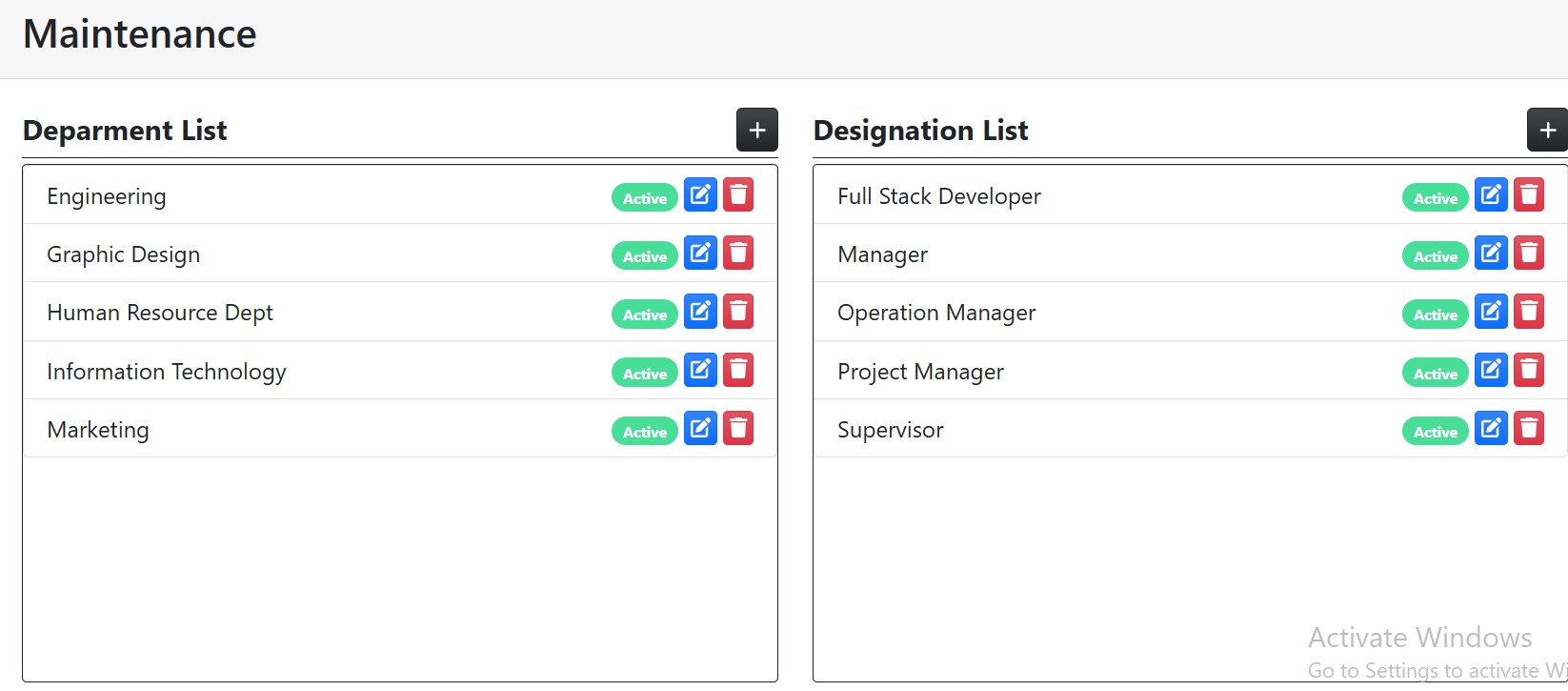


**Figure 4.1.4 Employee Modification**

20



**Figure 4.1.5 Accessing members List:**



**Figure 4.1.6.** **Fields and Professions:**

21

**4.2**  **Database Design:**

The Employee Payroll Management System utilizes MySQL, managed through the XAMPP Control Panel, for efficient data storage and retrieval. Database design is a crucial part of the system, ensuring payroll-related data is stored with minimal redundancy and optimized for quick and flexible access. Relationships are established between data items such as employee details, payroll records, and tax information, while unnecessary or redundant data is eliminated to enhance efficiency. Normalization is applied to maintain data consistency, minimize redundancy, and optimize updates, ensuring the database remains stable and reliable. By leveraging MySQL's robust capabilities within XAMPP, the system provides secure storage and seamless access to employee and payroll data, supporting the accurate and efficient management of payroll operations.

The **database design** for the Employee Payroll Management System is structured to ensure efficient data storage, retrieval, and integrity. Built using MySQL and managed via XAMPP, the database is normalized to reduce redundancy and maintain consistency across interconnected tables. Key tables include *Employee* (storing personal and job details), *Payroll* (tracking salary calculations and payment records), *Users* (managing system access and roles), *Tax* (handling deductions and compliance data), and *Designation/Department* (defining organizational hierarchy). Relationships between these tables are established through primary and foreign keys to maintain referential integrity. This design ensures scalability, quick query execution, and secure data handling, catering to the system's functional and operational requirements.

The database for the **Employee Payroll Management System** is designed to support seamless interaction between various modules while ensuring data security and integrity. Each table is meticulously crafted to represent a distinct functional area, such as *Employee* for storing demographic and employment data, *Payroll* for salary records and transaction details, *Tax* for deductions and compliance metrics, *Users* for access control, and *Designation/Department* for hierarchical management. The schema incorporates normalization techniques to optimize storage and prevent anomalies during data operations. With MySQL as the database and XAMPP for local server management, the design facilitates fast retrieval, robust transaction handling, and easy scalability. These features ensure that the system remains adaptable to organizational growth and evolving operational needs.

22

**4.3 IMPLEMENTATION (CODE):**

import java.sql.\*;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

public class DatabaseConnection {

private static final String URL = "jdbc:mysql://localhost:3306/EmployeeDB";

private static final String USER = "root"; // replace with your DB username

private static final String PASSWORD = ""; // replace with your DB password

public static Connection getConnection() throws SQLException {

return DriverManager.getConnection(URL, USER, PASSWORD);

}

}

public class EmployeePayrollSystem {

private Connection con;

private Statement stmt;

private JFrame frame;

private JPanel mainPanel;

public EmployeePayrollSystem() {

setupDatabaseConnection();

frame = new JFrame("Employee Payroll Management System");

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setSize(800, 600);

23

mainPanel = new JPanel(new CardLayout());

mainPanel.add(createEmployeePage(), "Employee");

mainPanel.add(createPayrollPage(), "Payroll");

mainPanel.add(createUsersPage(), "Users");

mainPanel.add(createTaxPage(), "Tax");

mainPanel.add(createDesignationPage(), "Designation");

JPanel navPanel = new JPanel();

String[] pages = {"Employee", "Payroll", "Users", "Tax", "Designation"};

JComboBox<String> pageSelector = new JComboBox<>(pages);

pageSelector.addActionListener(e -> {

CardLayout cl = (CardLayout) mainPanel.getLayout();

cl.show(mainPanel, (String) pageSelector.getSelectedItem());

});

navPanel.add(new JLabel("Select Page: "));

navPanel.add(pageSelector);

frame.add(navPanel, BorderLayout.NORTH);

frame.add(mainPanel, BorderLayout.CENTER);

frame.setVisible(true);

}

private void setupDatabaseConnection() {

try {

con = DriverManager.getConnection("jdbc:mysql://localhost:3306/payrollsystem", "root", "root");

stmt = con.createStatement();

} catch (SQLException e) {

e.printStackTrace();

}

}

private JPanel createEmployeePage() {

24

JPanel employeePanel = new JPanel(new BorderLayout());

employeePanel.add(new JLabel("Employee Page", SwingConstants.CENTER), BorderLayout.NORTH);

JTextField empNameField = new JTextField(15);

JTextField empDeptField = new JTextField(15);

JButton insertButton = new JButton("Insert");

JButton updateButton = new JButton("Update");

JButton deleteButton = new JButton("Delete");

insertButton.addActionListener(e -> {

try {

String query = "INSERT INTO employee (name, department) VALUES (?, ?)";

PreparedStatement pstmt = con.prepareStatement(query);

pstmt.setString(1, empNameField.getText());

pstmt.setString(2, empDeptField.getText());

pstmt.executeUpdate();

JOptionPane.showMessageDialog(frame, "Employee inserted successfully.");

} catch (SQLException ex) {

ex.printStackTrace();

}

});

updateButton.addActionListener(e -> {

try {

String query = "UPDATE employee SET department = ? WHERE name = ?";

PreparedStatement pstmt = con.prepareStatement(query);

pstmt.setString(1, empDeptField.getText());

pstmt.setString(2, empNameField.getText());

pstmt.executeUpdate();

JOptionPane.showMessageDialog(frame, "Employee updated successfully.");

} catch (SQLException ex) {

25

ex.printStackTrace();}})

deleteButton.addActionListener(e -> {

try {

String query = "DELETE FROM employee WHERE name = ?";

PreparedStatement pstmt = con.prepareStatement(query);

pstmt.setString(1, empNameField.getText());

pstmt.executeUpdate();

JOptionPane.showMessageDialog(frame, "Employee deleted successfully.");

} catch (SQLException ex) {

ex.printStackTrace();

}

});

JPanel inputPanel = new JPanel();

inputPanel.add(new JLabel("Name: "));

inputPanel.add(empNameField);

inputPanel.add(new JLabel("Department: "));

inputPanel.add(empDeptField);

inputPanel.add(insertButton);

inputPanel.add(updateButton);

inputPanel.add(deleteButton);

employeePanel.add(inputPanel, BorderLayout.CENTER);

return employeePanel;

}

private JPanel createPayrollPage() {

JPanel payrollPanel = new JPanel();

payrollPanel.add(new JLabel("Payroll Page"));

return payrollPanel;

}

private JPanel createUsersPage() {

JPanel usersPanel = new JPanel();

26

usersPanel.add(new JLabel("Users Page"));

return usersPanel;

}

private JPanel createTaxPage() {

JPanel taxPanel = new JPanel();

taxPanel.add(new JLabel("Tax Page"));

return taxPanel;

}

private JPanel createDesignationPage() {

JPanel designationPanel = new JPanel();

designationPanel.add(new JLabel("Designation and Department Page"));

return designationPanel;

}

public static void main(String[] args) {

SwingUtilities.invokeLater(EmployeePayrollSystem::new);

}

}

27

# 5. Conclusion:

The Employee Payroll Management System, developed with expert guidance, represents a thoughtfully designed and implemented solution for efficient payroll operations. By incorporating user-friendly functionalities such as employee management, payroll processing, tax calculations, and department handling, the system streamlines organizational workflows. Emphasizing robust security and data integrity, particularly in user management modules, it ensures reliable and protected operations. This project not only fulfills current organizational needs but also provides a scalable framework for future enhancements, making it a comprehensive and adaptable solution.

# 6.Reference links:

1. <https://www.javatpoint.com/java-awt>
2. <https://www.javatpoint.com/java-swing>

[3] https://www.javatpoint.com/jdbc-tutorial

28