# Design Document for Payroll Management System


```
Name : Snigdha SV (PES1UG2CS600)
Name : Soumya LS (PES1UG22CS603)
```


---

# 1. Introduction

## 1.1 Purpose

The purpose of the architecture document is to outline the technical structure of the **Payroll Management System (PMS)** to ensure effective, scalable, and secure payroll processing.

## 1.2 Scope

The architecture covers the design of the entire PMS application, focusing on functional requirements such as payroll calculation, tax management, and report generation.

## 1.3 Definitions, Acronyms, and Abbreviations

- **PMS**: Payroll Management System
- SMEs: Small and Medium-sized Enterprises
- **HR**: Human Resources
- **UI**: User Interface
- **API**: Application Programming Interface

## 1.4 References

- IEEE standards for software architecture documentation.

# 2. Architectural Representation

The system will be a **client-server architecture**. The frontend will be developed using React, while the backend will use Node.js with a REST API to connect to the database.

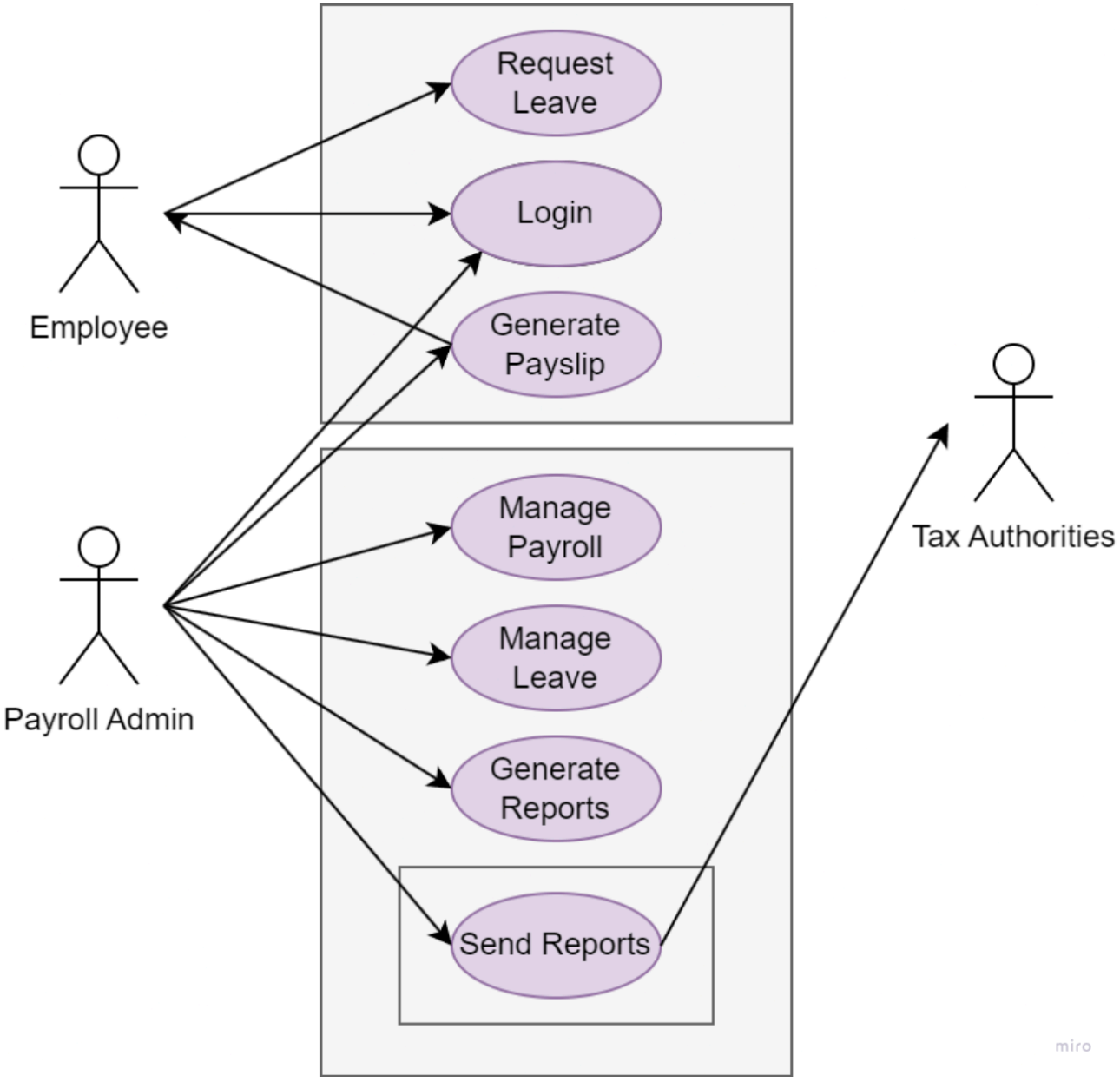
# 3. Architectural Goals and Constraints

- **Security**: Protecting sensitive employee data.
- **Scalability**: Capable of handling an increasing number of employees.
- **Integration**: Seamless interaction with HR and finance systems.

# 4. Use-Case View

## 4.1 Architecturally-Significant Use Cases

- **Payroll Calculation**: Automating salary and tax calculations.
- **Payslip Generation**: Generating and distributing payslips to employees.
- **Tax Management**: Accurate deduction of taxes and report generation.




# 5. Logical View

## 5.1 Architecture Overview – Package and Subsystem Layering

- **UI Layer**: Frontend interface for HR and employees.
- **Business Logic Layer**: Salary calculation, tax management, leave processing.
- **Database Layer**: Storing employee, payroll, and tax data.

## 5.2 Process View

### 5.2.1 Processes

Processes include data input from HR systems, payroll calculation algorithms, and report generation routines.

### 5.2.2 Process to Design Elements

- **Payroll Calculation Module**: Handles salary computation based on employee records.
- **Payslip Module**: Generates payslips in PDF format.

#### 5.2.3 Process Model to Design

Interaction between UI, API, and database through RESTful services.

### 5.2.4 Model Dependencies

Dependencies among modules will be clearly defined to avoid integration issues.

### 5.2.5 Processes to the Implementation

Implementation strategies will align with defined processes for seamless execution.

# 6. Deployment View

## 6.1 External Desktop PC

User interface accessible via web browsers on standard desktop PCs.

## 6.2 Desktop PC

Internal access for HR administrators through a secure network connection.

## 6.3 Registration Server

Handles authentication of HR users and employees.

## 6.4 Payroll Database

Stores all payroll and employee data securely.

# 7. Performance

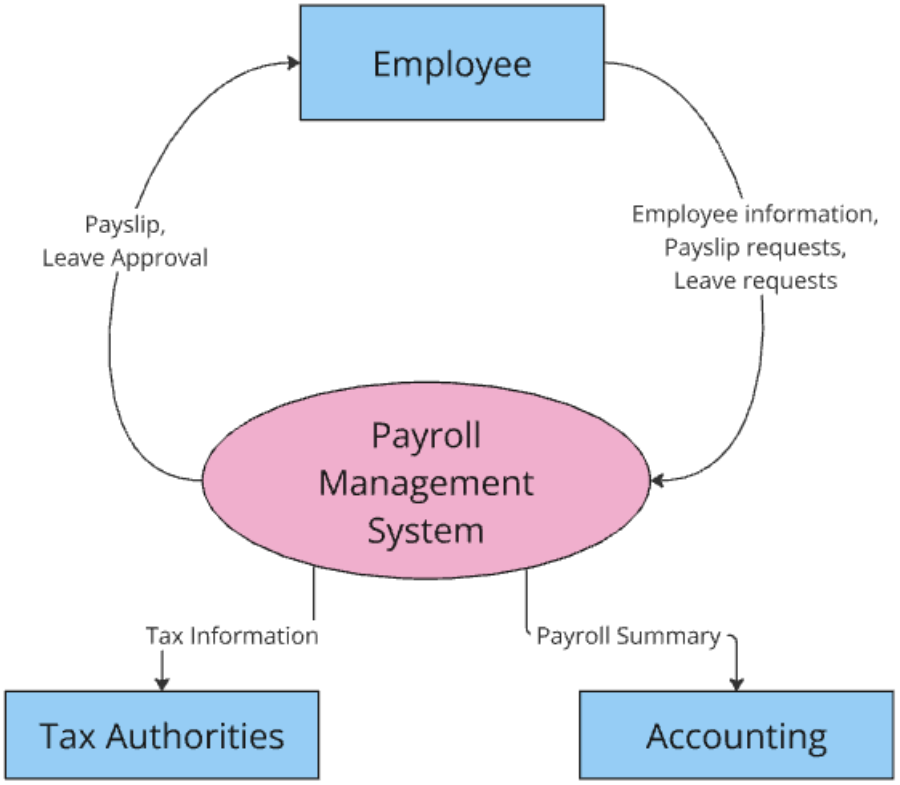
- The system shall generate a payslip for 1,000 employees within 2 minutes.
- The payroll report for tax purposes will be generated in under 5 seconds.

# 8. Quality

Quality assurance measures will ensure compliance with functional requirements while maintaining high usability standards for end-users.

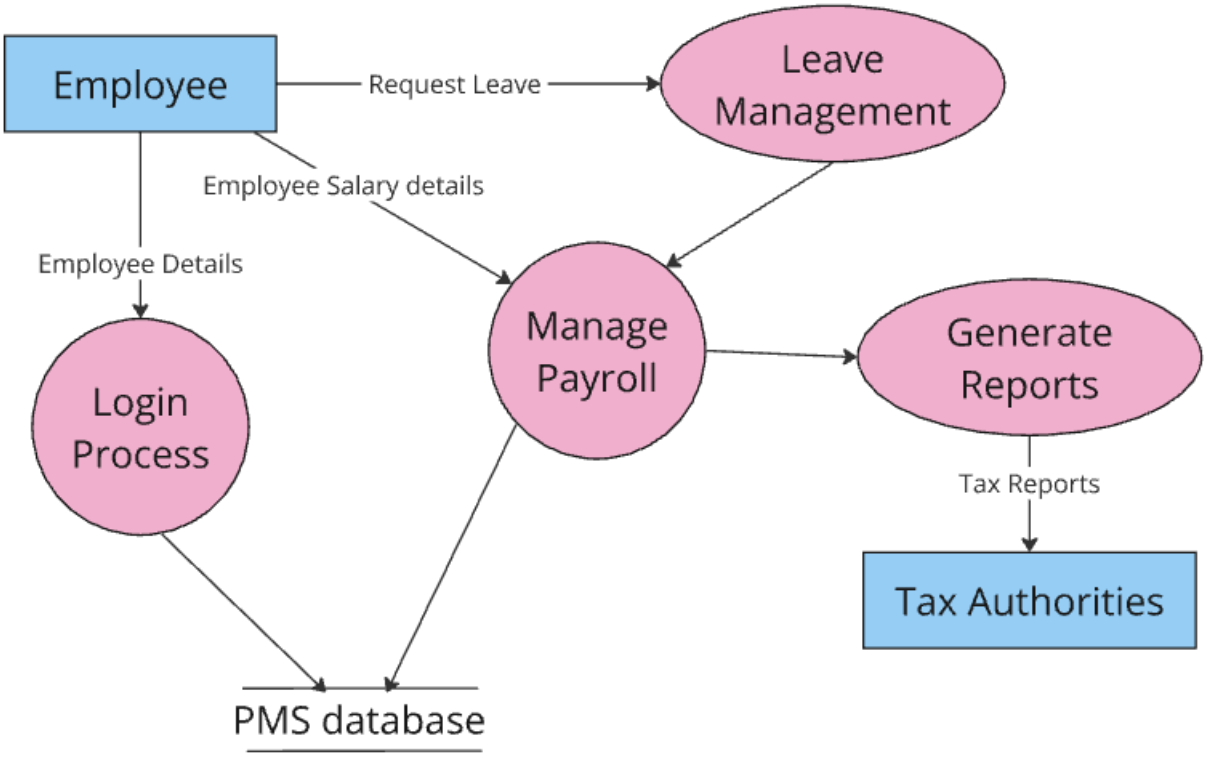
# 9. Data Flow Diagram (DFD)

**Level 0 DFD**



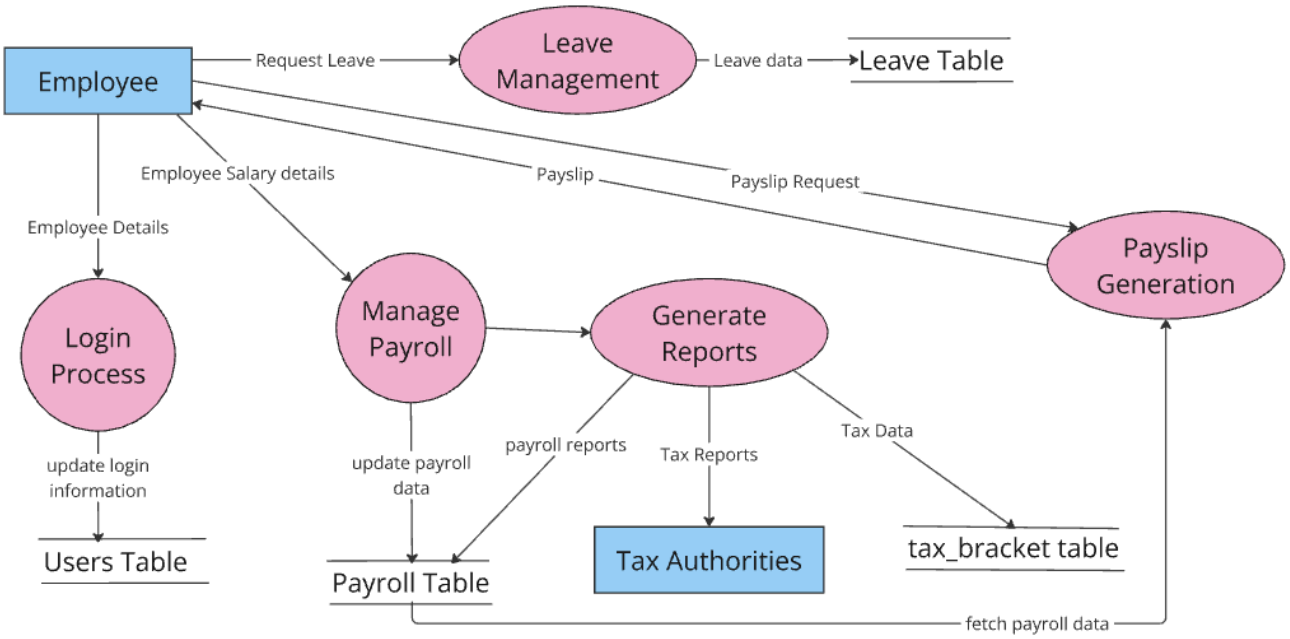

Level -0 DFD

**Level 1 DFD**




Level - 1DFD

**Level 2 DFD**


Employee
Request Leave
Leave Management
Leave data
Leave Table
Employee Salary details
Payslip
Payslip Request
Employee Details
Payslip Generation
Login Process
Manage Payroll
Generate Reports
update login information
update payroll data
payroll reports
Tax Reports
Tax Data
Users Table
Payroll Table
Tax Authorities
tax_bracket table
fetch payroll data

Level - 2 DFD