**PAYROLL PROCESSING APPLICATION**

**Objective**

To streamline salary processing, improve accuracy, and ensure timely payroll execution by developing an automated Payroll Management System. The system will eliminate manual errors, provide structured job execution, real-time monitoring, and compliance reporting.

**Problem Statement**

Current Challenges:

* Manual Processing: Payroll is calculated and disbursed manually, leading to inefficiencies and errors.
* No Automation: Lack of scheduled job execution, monitoring, or alerting mechanisms.
* Delayed Issue Detection: Failures or delays are identified only after impacting employees.
* Compliance Risks: No structured audit trail for tax, benefits, and regulatory reporting.

Impact:

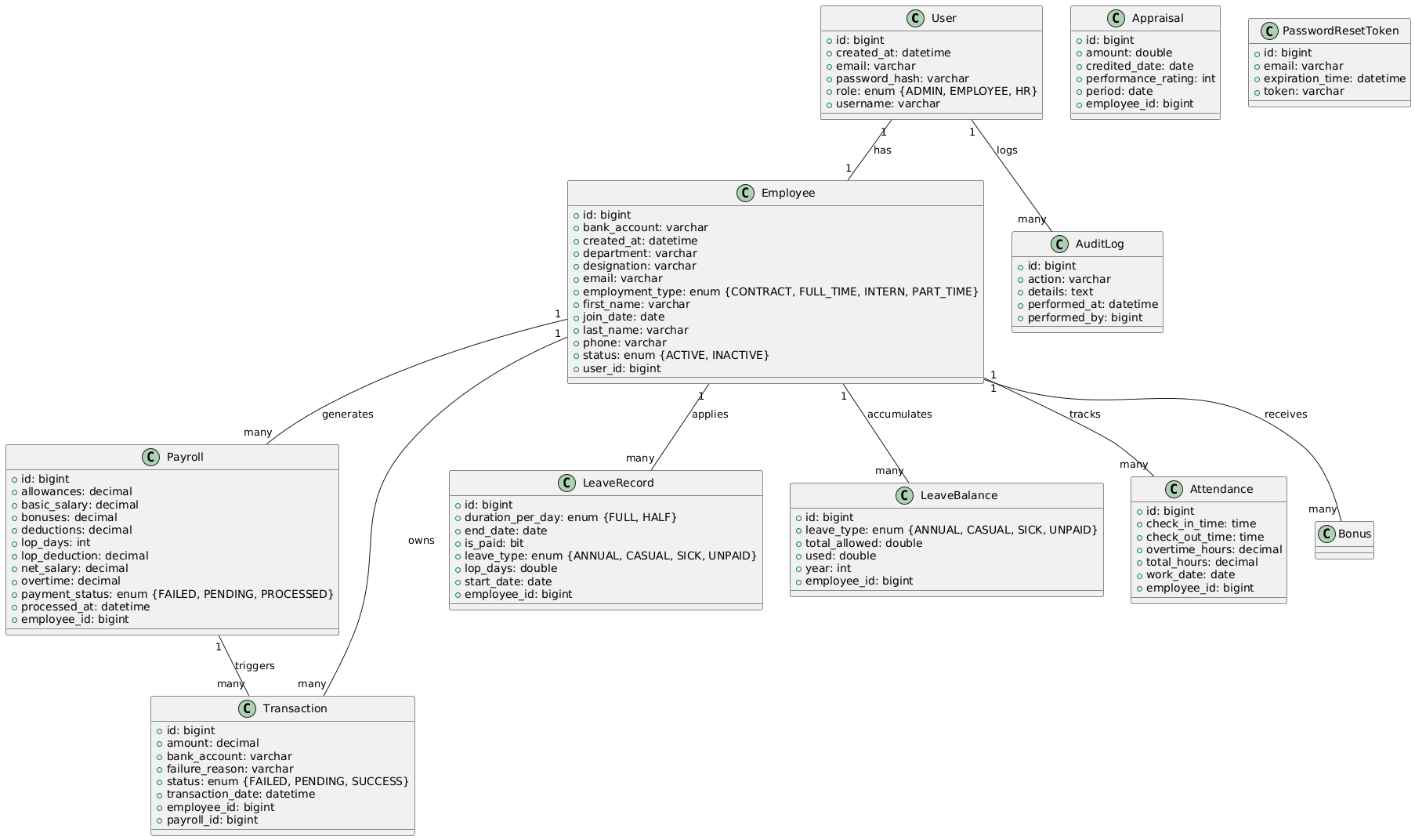
* Increased operational costs due to manual efforts.
* Employee dissatisfaction due to incorrect or delayed payments.
* Compliance risks due to inaccurate tax and benefit calculations.

**Proposed Solution:**

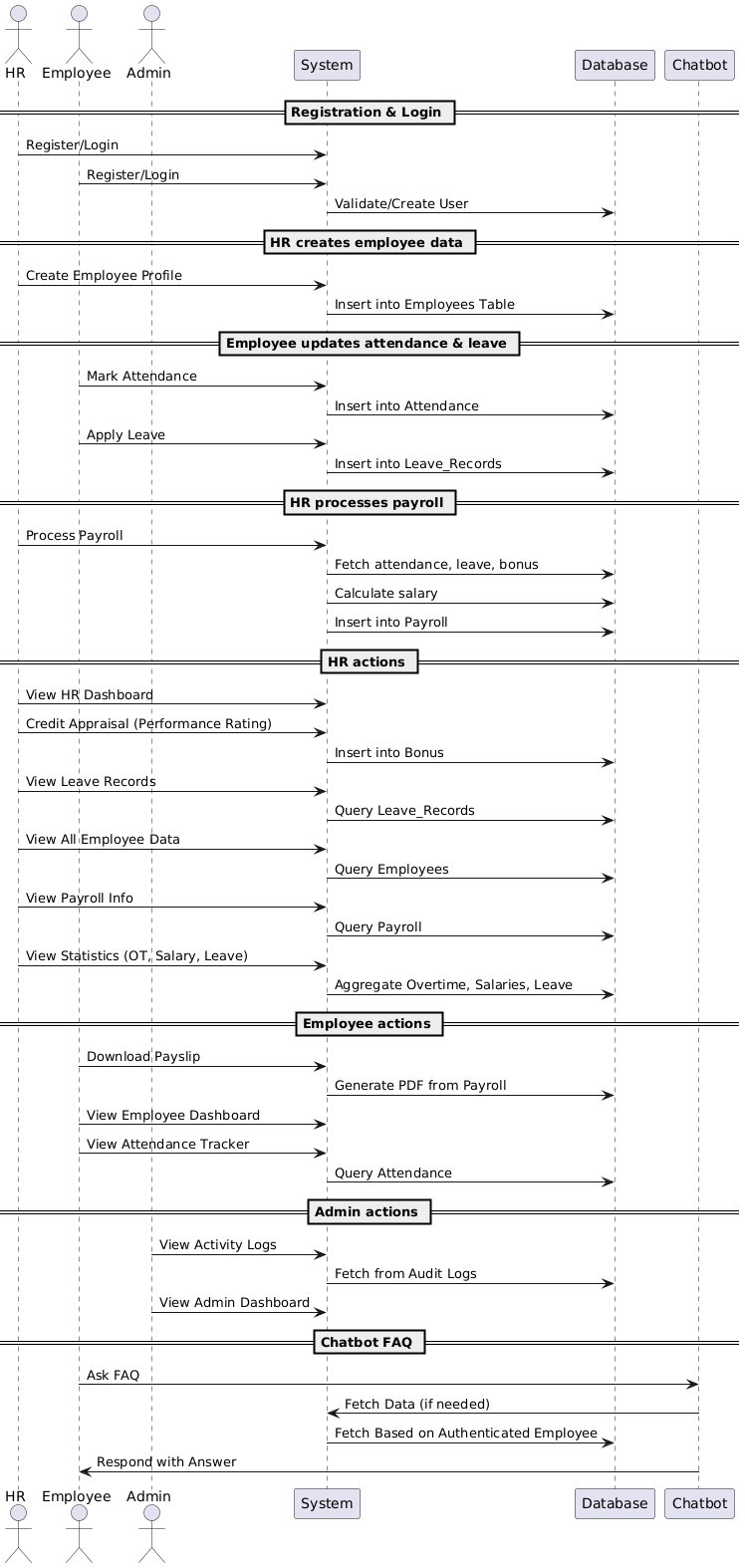
A Payroll Management System that automates:

* ✅ **Role-Based Access**: Separate portals for Admin, HR, and Employee with secure login and authentication.
* 👤 **Employee Management**: HR can create, view, and manage employee profiles and employment details.
* 🕒 **Attendance & Leave**: Employees mark attendance, apply for leave; HR reviews and tracks leave records.
* 💰 **Payroll Processing**: HR processes payroll based on attendance, leave, and bonuses; system calculates net salary.
* 📈 **HR Dashboard**: Displays employee stats, salary reports, overtime analytics, and leave distribution.
* 📄 **Payslip Access**: Employees can securely download their payslips and view salary details.
* ⭐ **Performance-Based Bonus**: HR credits bonuses based on performance rating.
* 🧠 **Chatbot Assistant**: Provides instant answers to FAQs and fetches employee-specific data after login.
* 🔍 **Admin Monitoring**: Admin views system logs, activity history, and error reports.
* 🌐 **Technology Stack**: Built using ReactJS (frontend), Spring Boot (backend), and MySQL (database).

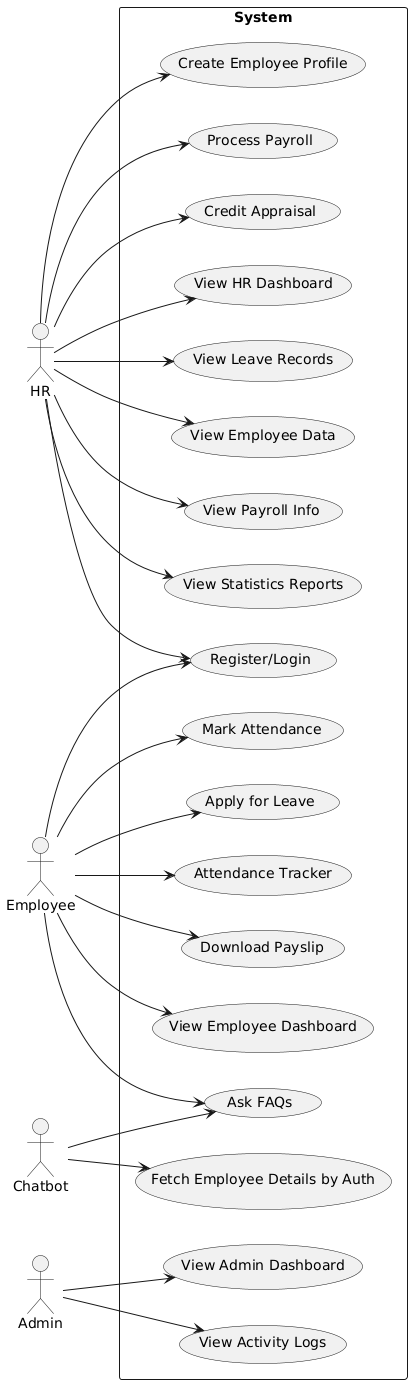
**Class Diagram:**



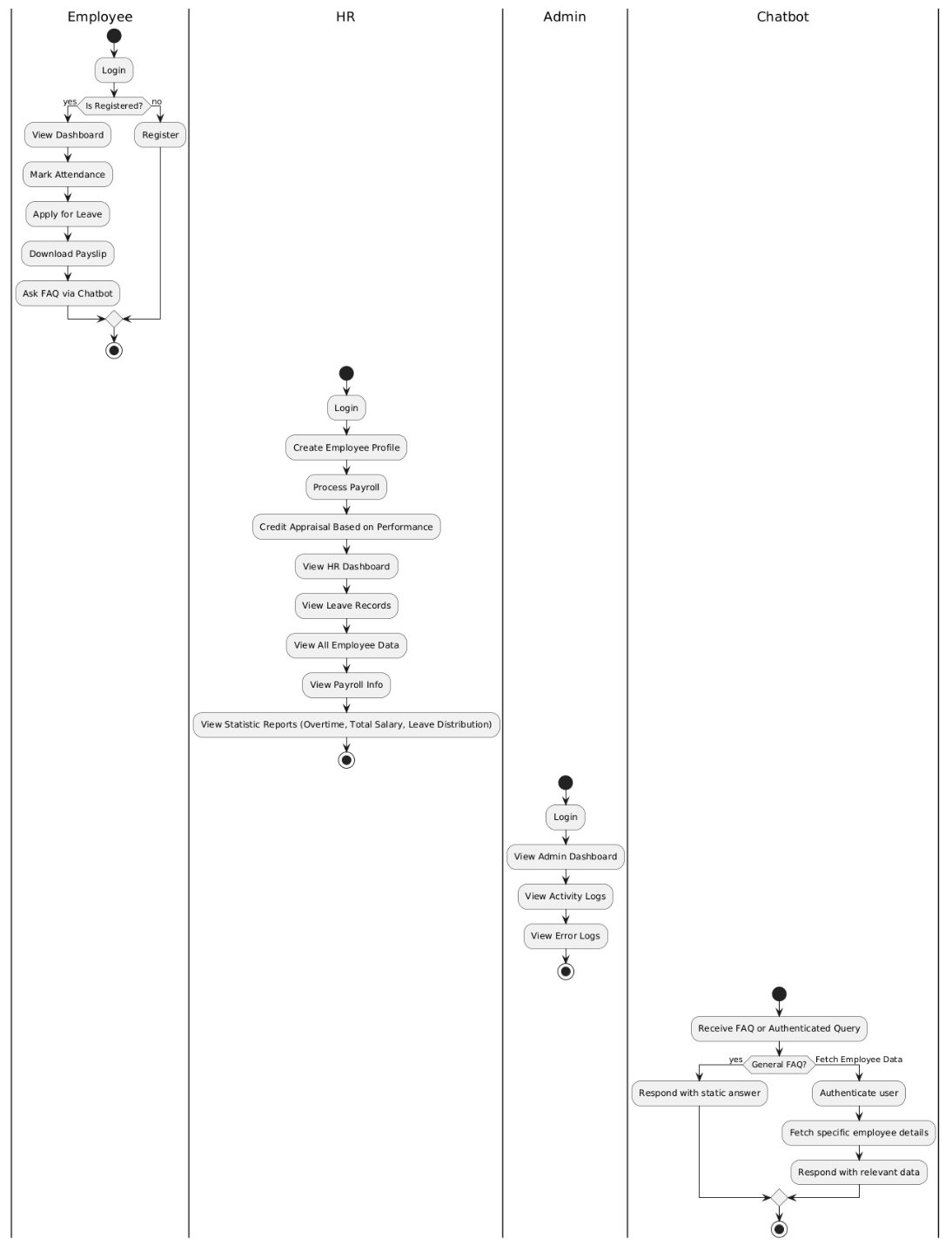
**LLD – Sequence Diagram**



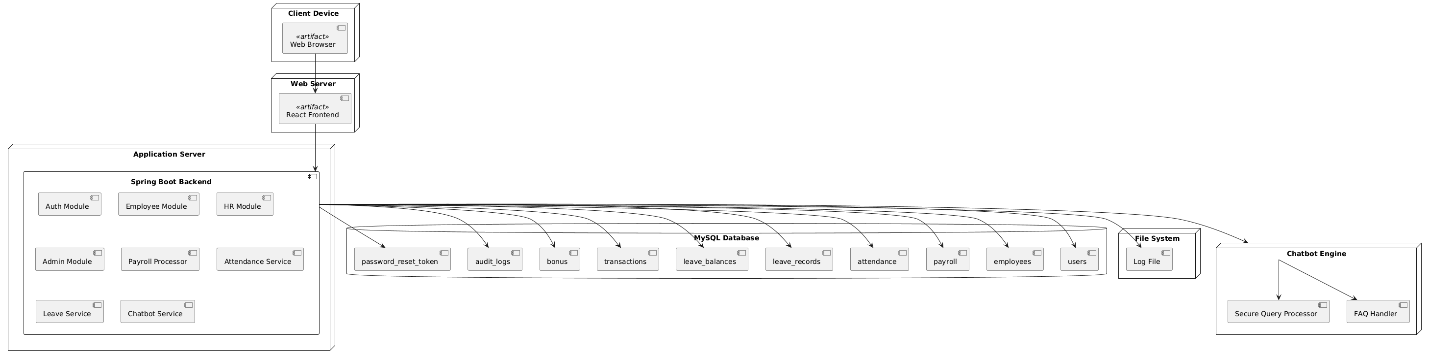
**UseCase Diagram:**



**Activity Diagram:**



**Deployment Diagram:**



**Component diagram:**

