High Level Document of ERP System

Contents

1 Introduction

The **Konecta ERP System** is a full-stack enterprise web application designed to integrate **HR** and **Finance** functions within a unified and secure platform. It automates workflows, enhances collaboration, and provides real-time analytics to management. The system is built using a microservices architecture, ensuring modularity, scalability, and independent service deployment.

1.1 System Objectives

- Automate HR and Finance workflows for efficiency and transparency.
- Enable employees to self-manage attendance, leaves, and payroll.
- Provide real-time dashboards and reports to HR and Finance managers.
- Ensure data integrity, security, and scalability using cloud deployment.

1.2 Core Modules

1. **Auth Service:** Handles JWT-based authentication and secure access.
2. **HR Service:** Manages recruitment, attendance, leaves, training, performance, and offboarding.
3. **Finance Service:** Manages payroll, expenses, invoices, and financial reports.
4. **Reporting Service:** Built using ASP.NET; generates analytics dashboards, exportable PDF/Excel reports, and cross-module summaries.
5. **Admin Portal:** Provides centralized control, analytics, and user management.
6. **Employee Portal:** Enables employees to view and manage their own data.

2 System Architecture Overview

The ERP system follows a **microservices-based architecture** with independent backend services connected through REST APIs.

2.1 Architecture Components

- **Frontend:** Angular web application (HR, Finance, Admin, and Employee dashboards)
- **Backend:** Spring Boot microservices (Auth, HR, Finance)
- **Reporting Service:** ASP.NET Core microservice integrated via REST APIs for generating and exporting reports.
- **Database:** PostgreSQL (separate schema per service)
- **API Gateway:** Spring Cloud Gateway for routing and centralized access
- **Service Discovery:** Eureka Server for dynamic service registration
- **Configuration Management:** Spring Cloud Config Server
- **Security:** JWT authentication with role validation
- **Deployment:** Docker containers and CI/CD pipeline integration

2.2 System Architecture Diagram

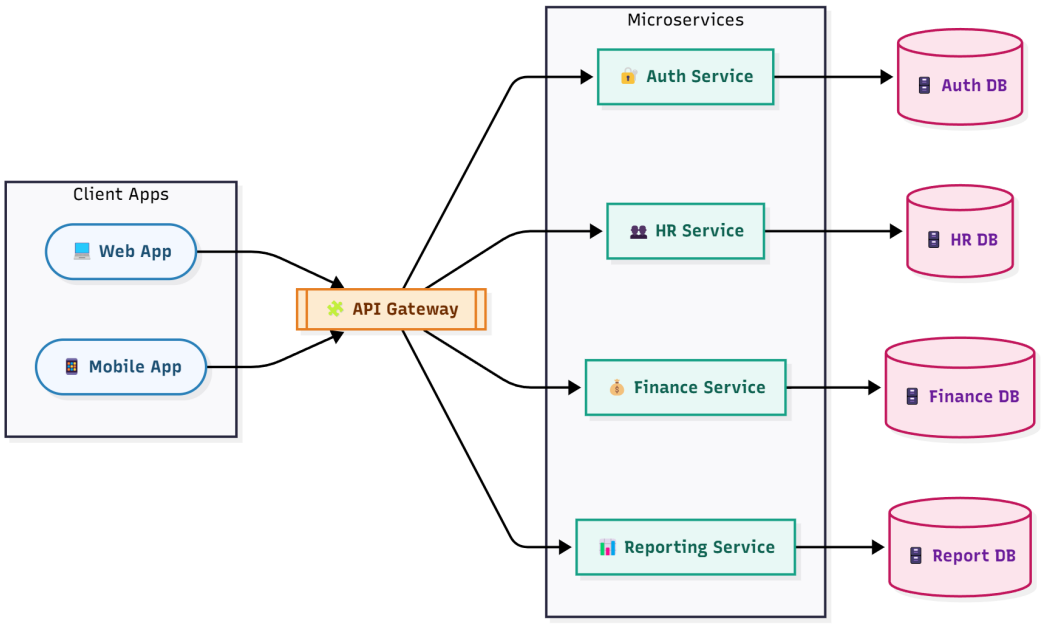

Figure 1: System Diagram

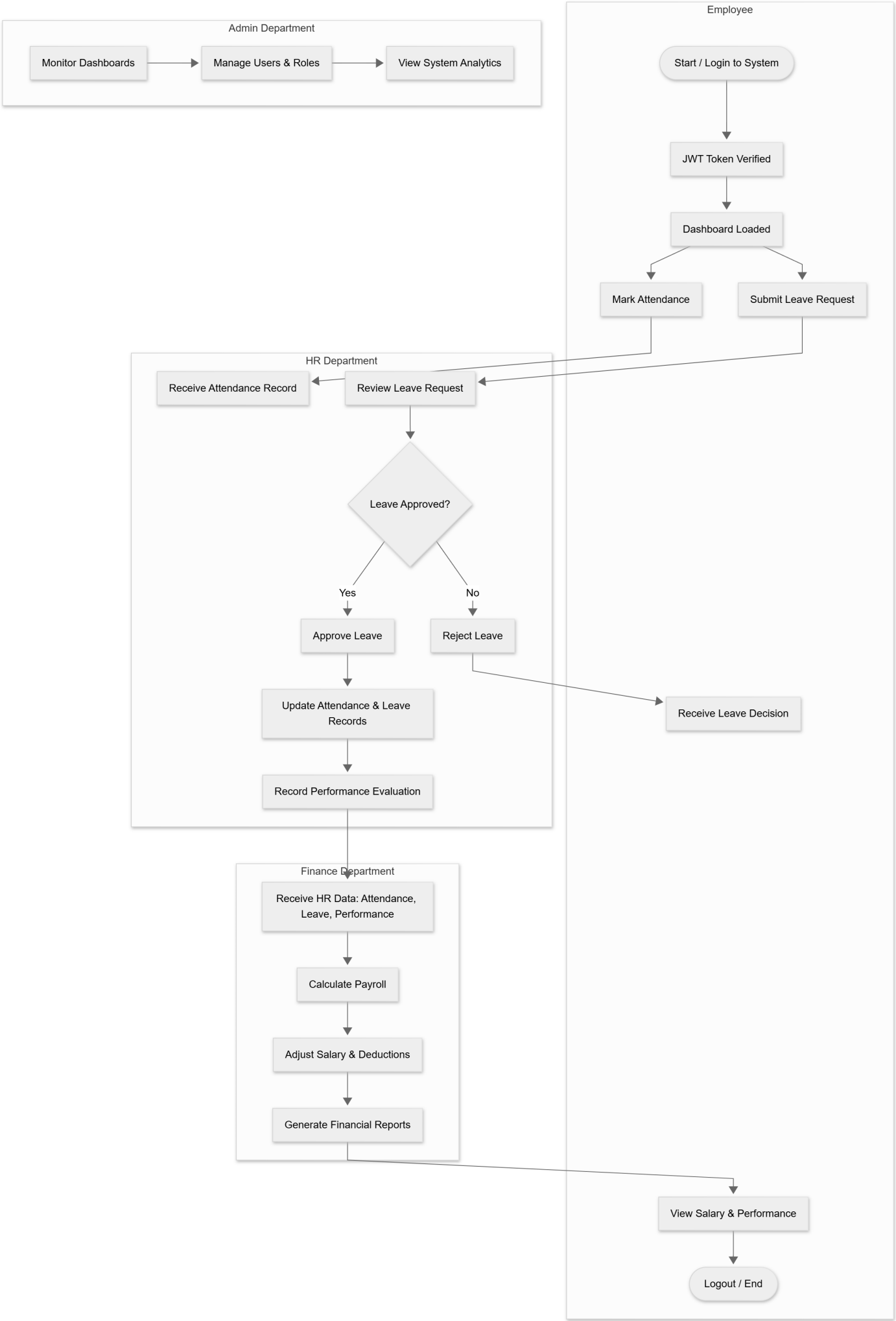

Figure 2: Flowchart

2.3 Database Architecture Overview

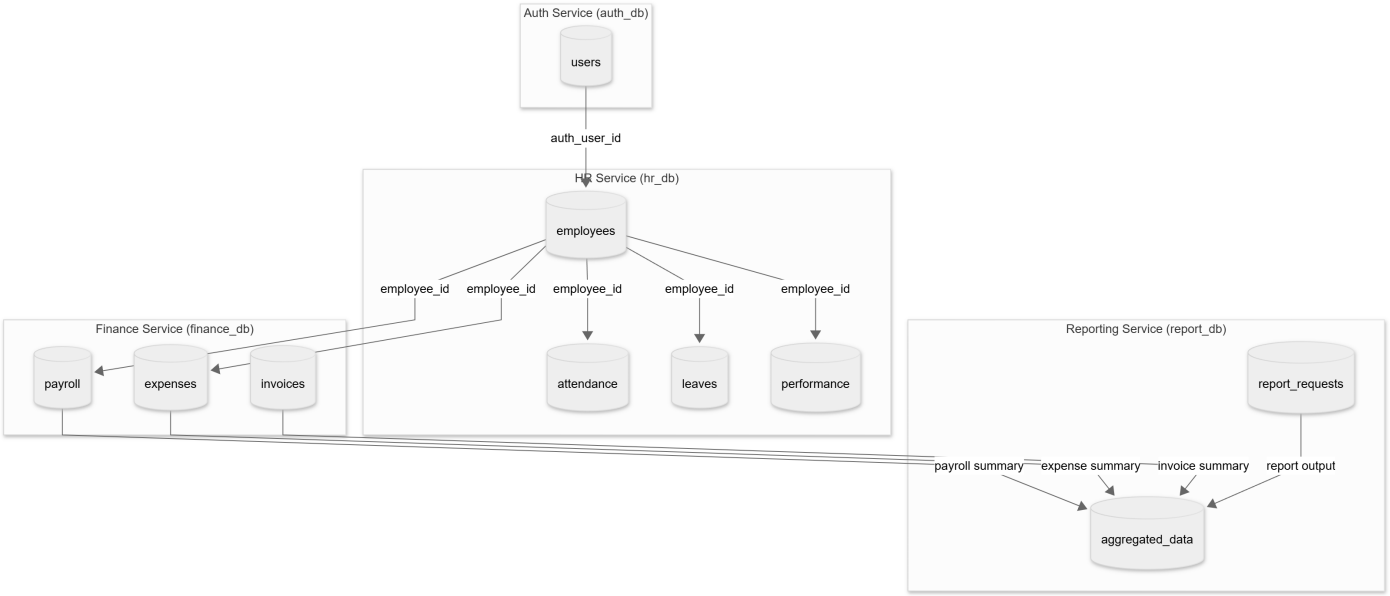

Figure 3: High-level Database Architecture

2.4 Detailed Database Schema (ERD)

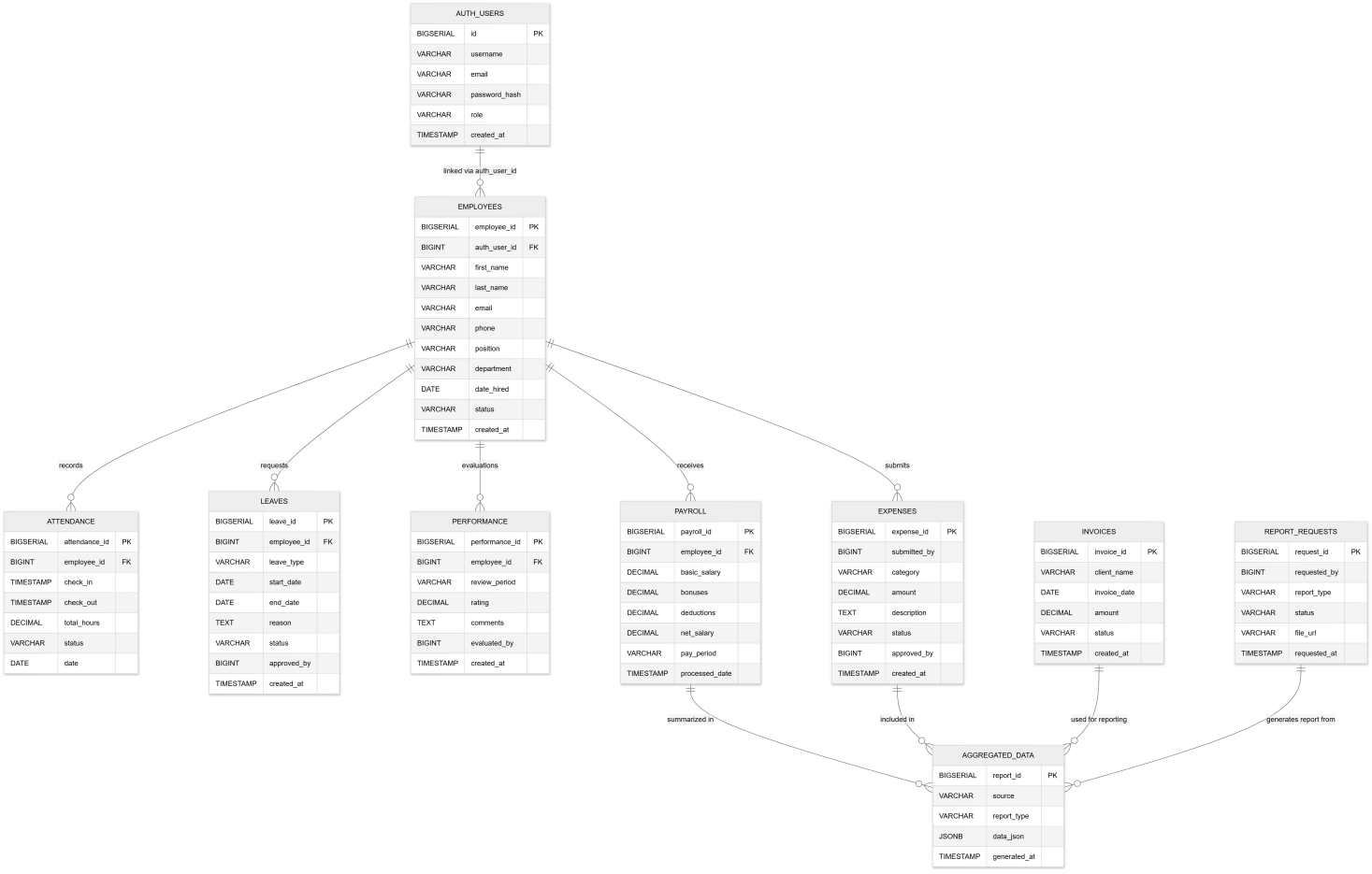

Figure 4: Low-level Database Schema ERD

3 HR Module Architecture

The **HR Service** automates the complete employee lifecycle, from recruitment to offboarding. It integrates attendance, training, and performance management for a seamless workflow.

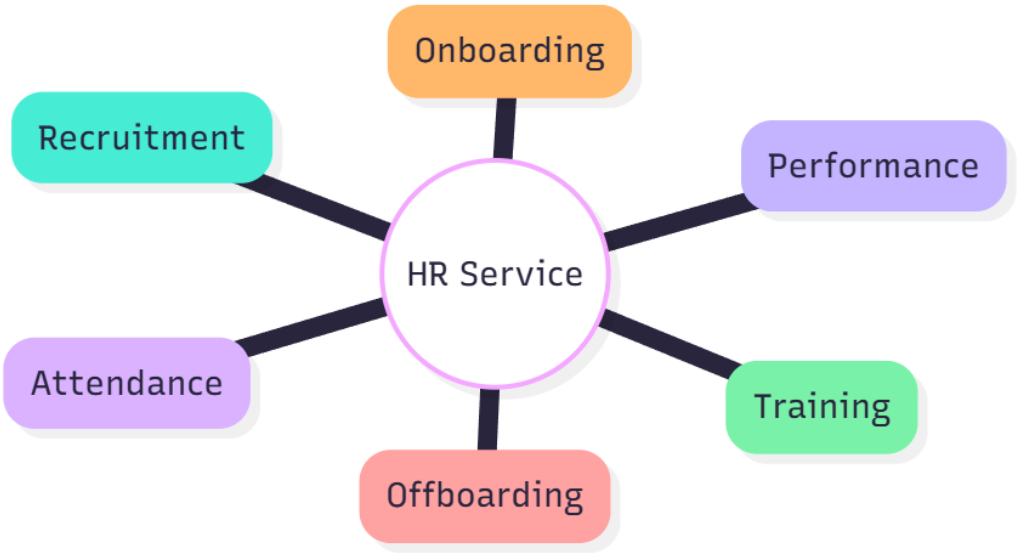

Figure 5: HR-mindmap

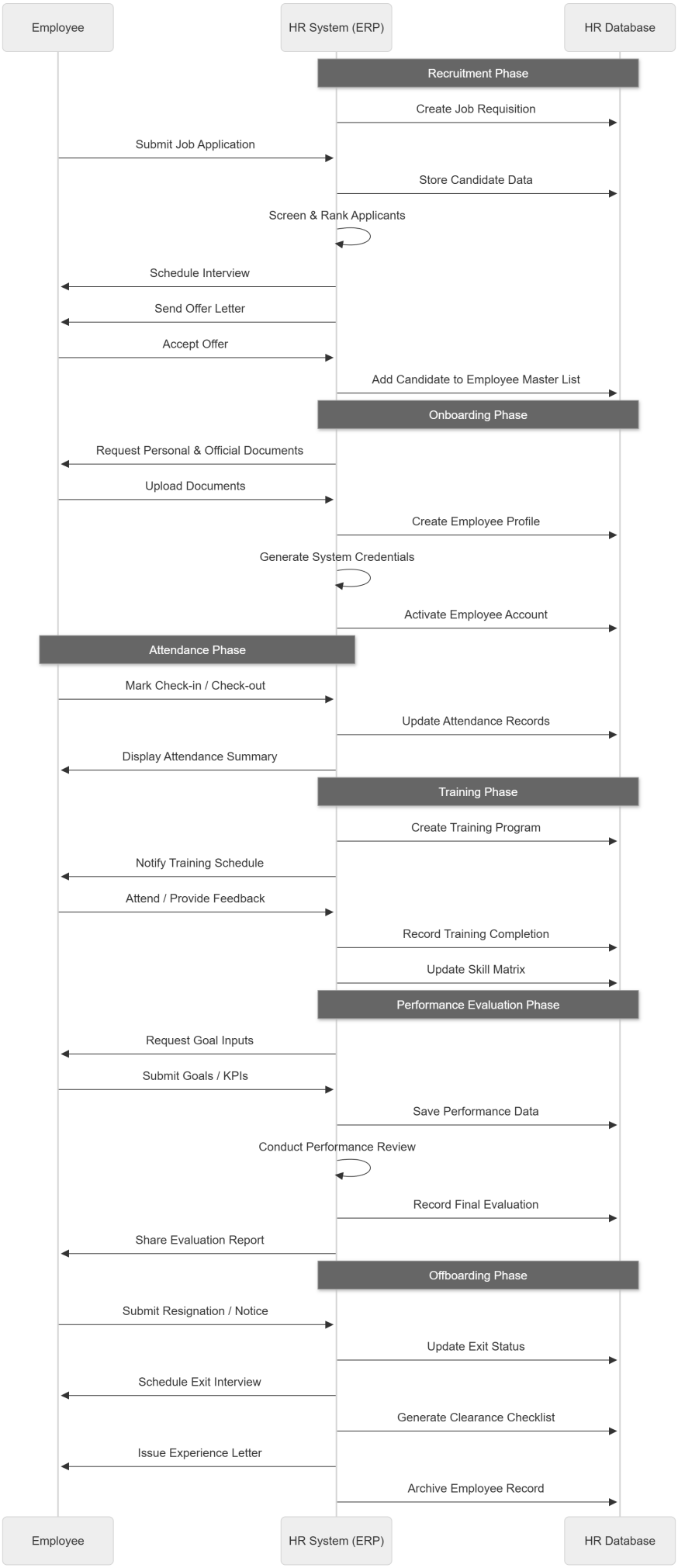

Figure 6: HR Squence Diagram

Table 1: HR Sub-Modules Overview

Sub-Module	**Description**	**ERP Output**
Recruitment	Job requisition, posting, screening, interviews, and offers	Hired candidates added to employee master list
Onboarding	Employee profile creation, credential setup, document upload	Active employee account with full access
Attendance	Check-in/out tracking, absence records, overtime calculation	Real-time attendance logs synced to payroll
Training	Program scheduling, nominations, feedback, certification	Updated skill matrix and learning record
Performance Evaluation	Goal setting, reviews, and scoring	Performance data linked to payroll module
Offboarding	Exit interviews, clearance, document generation	Securely archived employee record

4 Finance Module Architecture

The **Finance Service** manages all financial operations including payroll, expenses, and financial reporting.

Table 2: Finance Sub-Modules Overview

Sub-Module	**Description**
Payroll Management	Automatically calculates salaries using attendance and performance data.
Expense Management	Records and tracks organizational and departmental expenses.
Invoicing	Handles client and vendor billing with automated report generation.
Financial Reports	Provides dashboards and summaries of budgets, expenditures, and profitability.

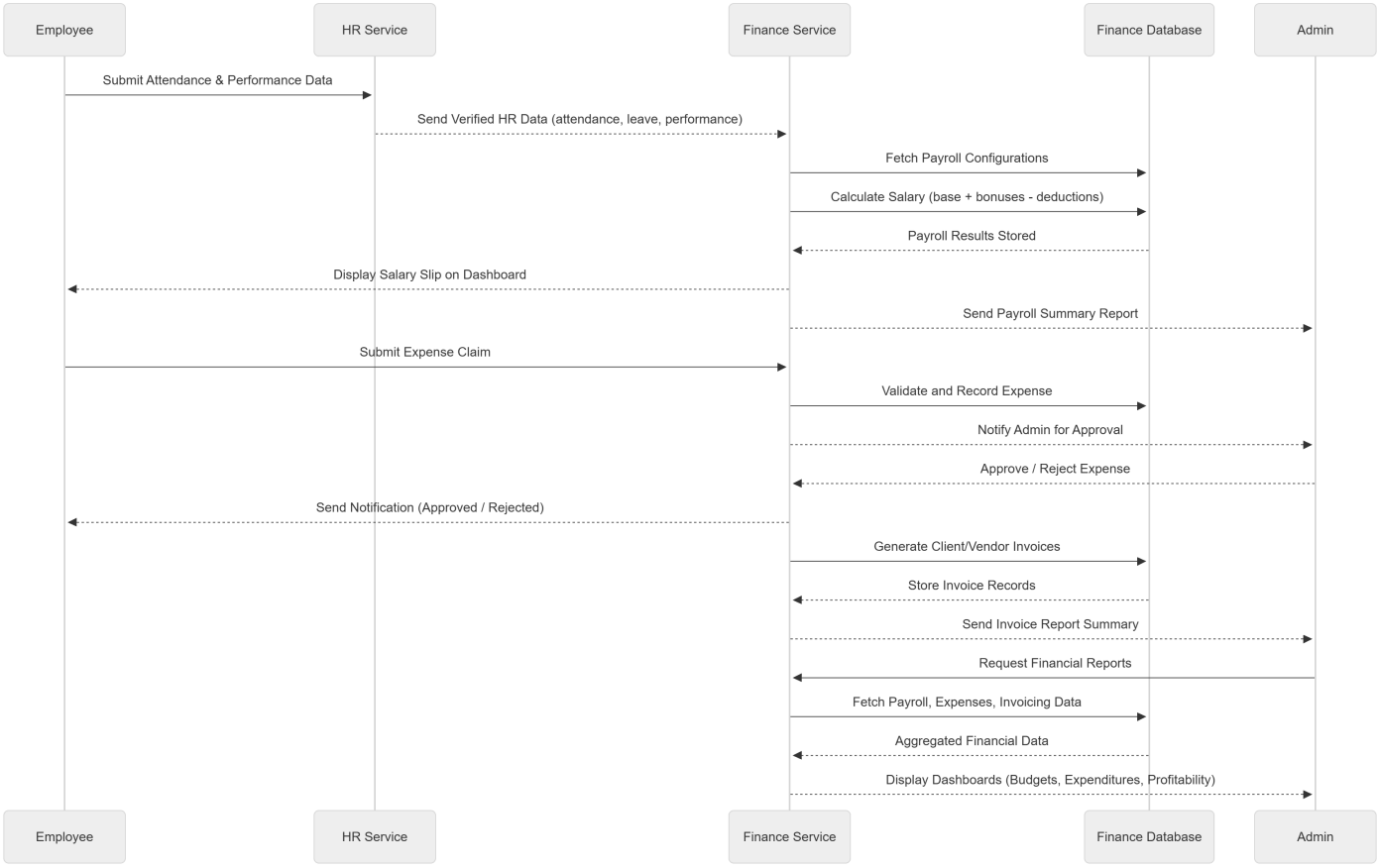

Figure 7: Finance Sequence Diagram

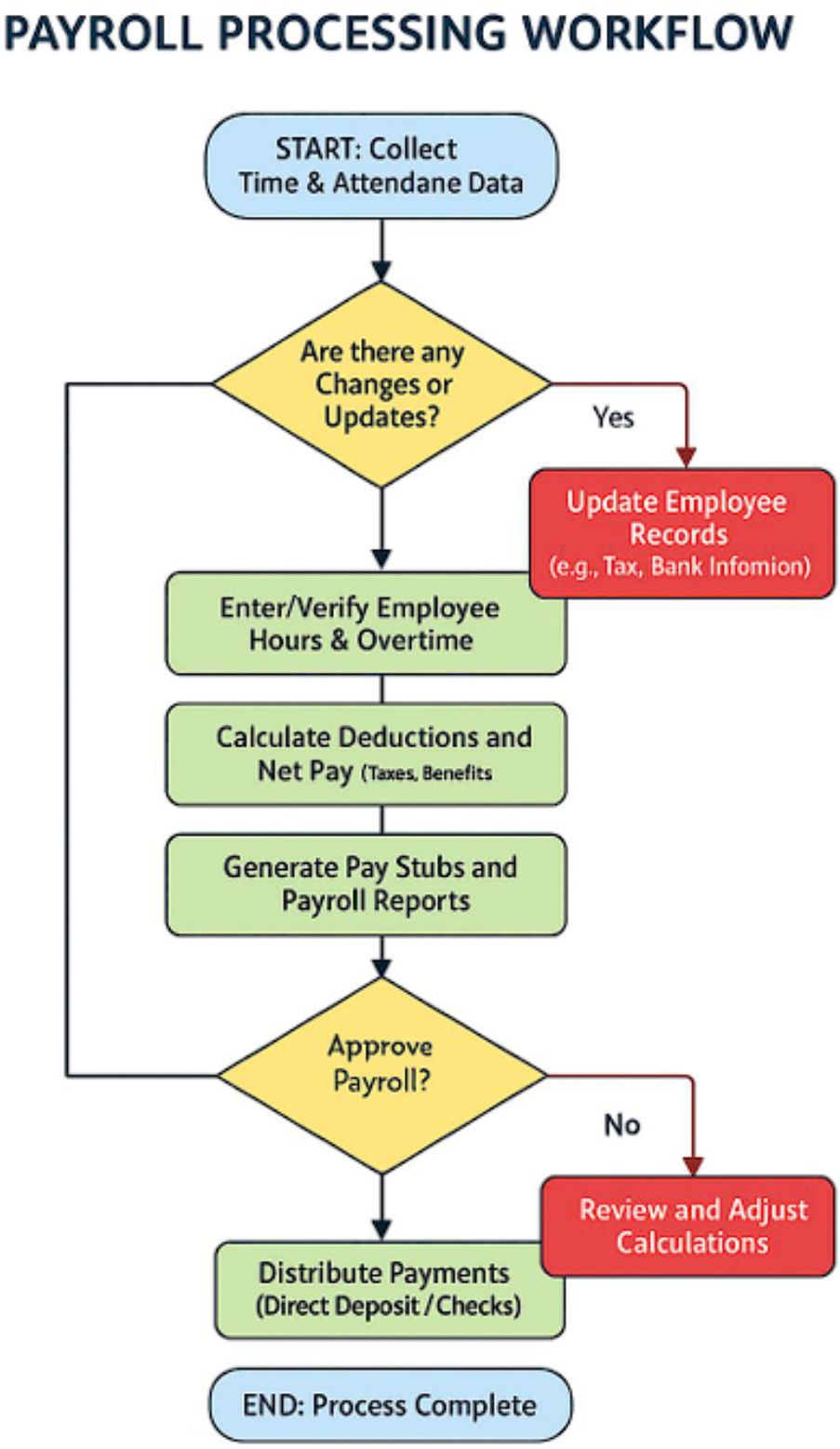

Figure 8: Payroll Process Workflow

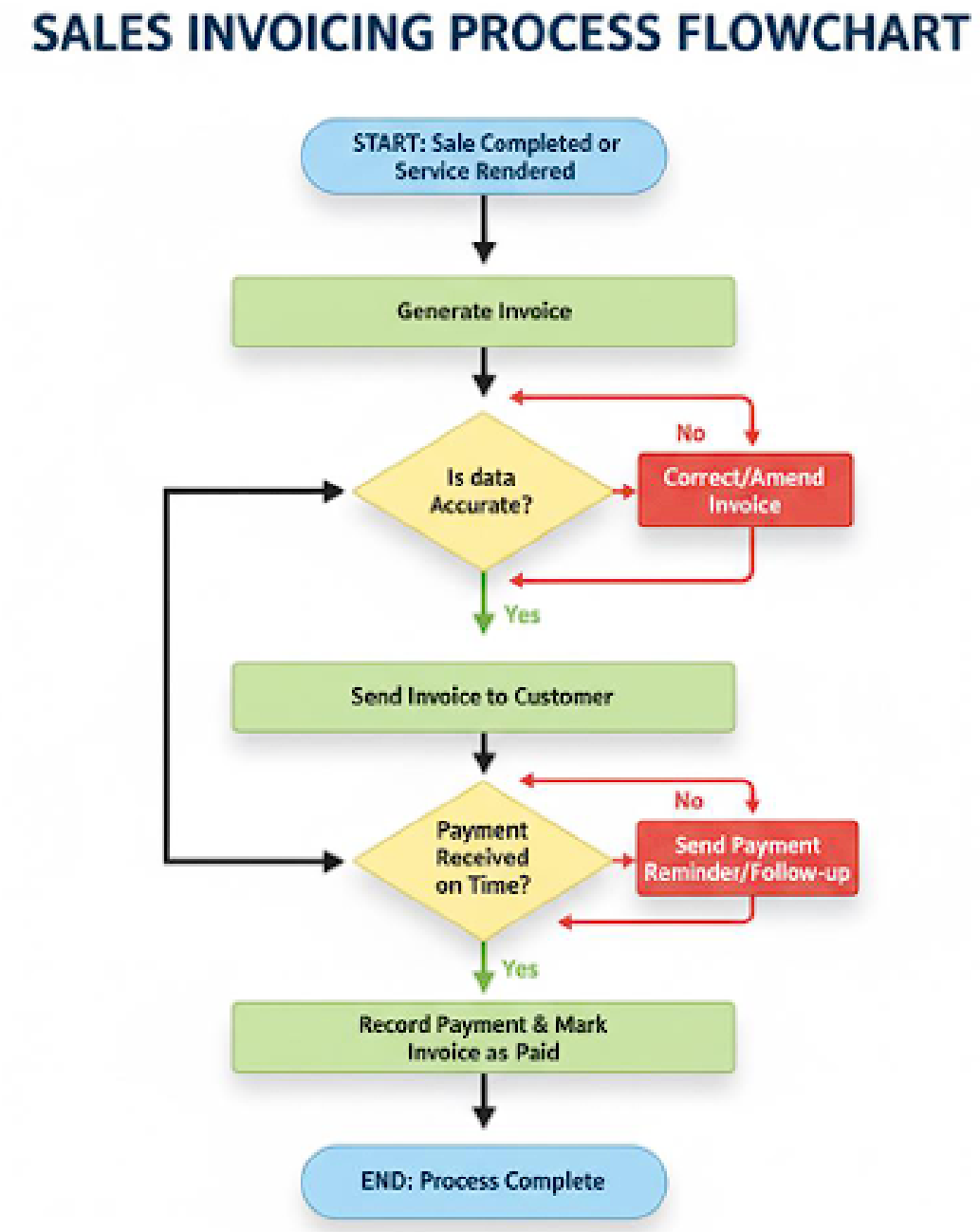

Figure 9: Sales Invoicing Process Workflow

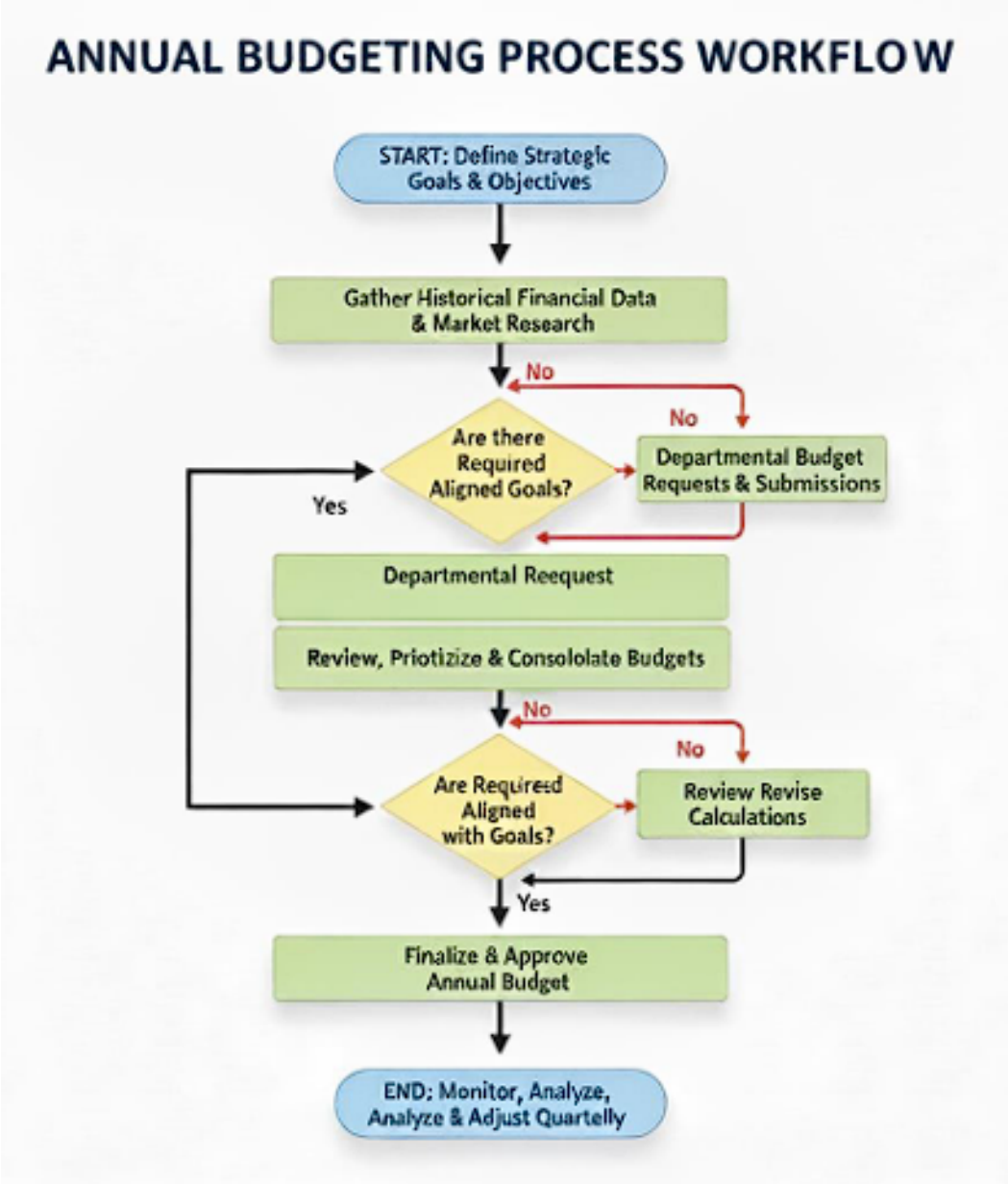

Figure 10: Annual Budgeting process workflow

5 Reporting Service Architecture

The **Reporting Service** is an independent ASP.NET Core microservice responsible for analytics visualization, PDF/Excel report generation, and consolidated ERP data summaries. It integrates with both the HR and Finance microservices through REST APIs.

- **Technology Stack:** ASP.NET Core 8.0, C#, Entity Framework, SQL Server (or PostgreSQL), and RESTful APIs.
- **Integration:** Consumes endpoints from HR and Finance services to compile data for dashboards and exports.
- **Output Formats:** Interactive dashboards, downloadable PDF reports, and Excel summaries.
- **Access Control:** Only Admins and Managers can generate organization-wide reports; HR and Finance users can generate module-specific reports.

Table 3: Reporting Sub-Modules Overview

Sub-Module	**Description**
Data Aggregation	Fetches data from HR and Finance microservices for consolidated analysis.
Report Generation	Creates PDF/Excel reports for payroll, attendance, and financial summaries.
Dashboard Analytics	Displays performance indicators, expense trends, and productivity charts.
Export and Sharing	Allows exporting reports and sharing with Admin/HR/Finance departments.

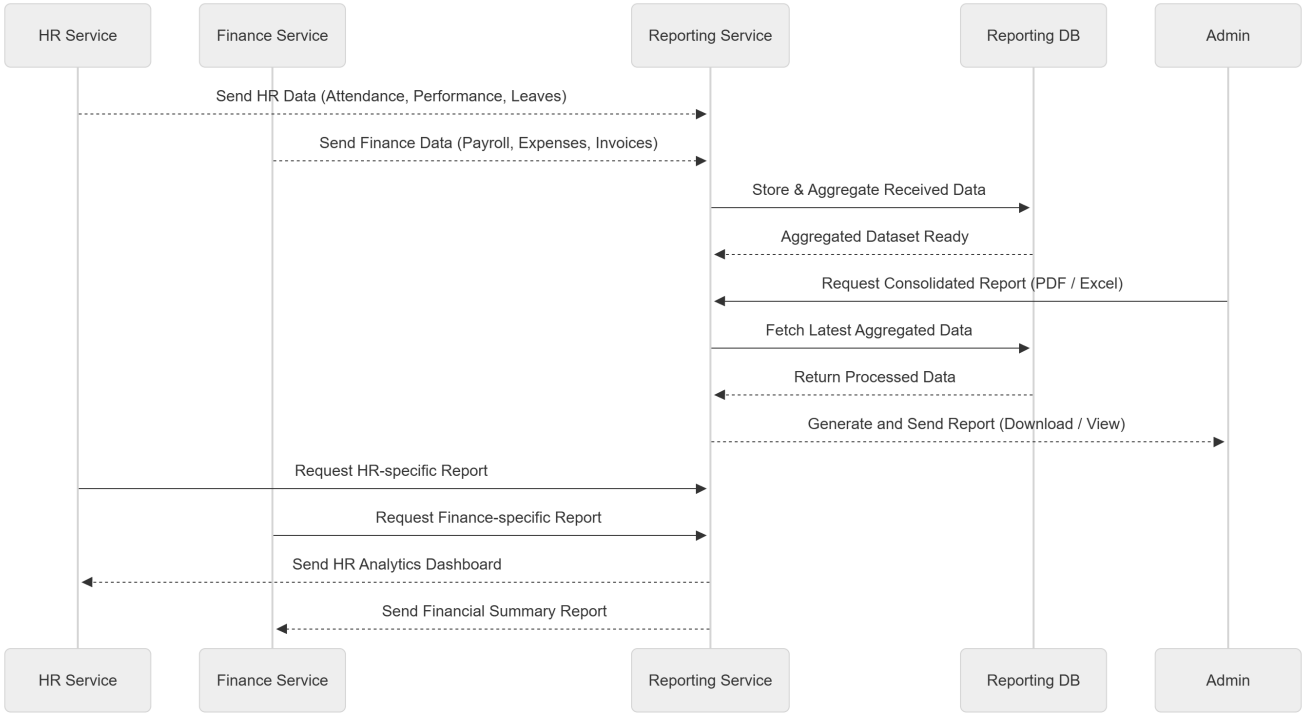

Figure 11: Report Sequence Diagram

6 Functional Requirements

This section defines the functional capabilities that the Konecta ERP System must provide. Each requirement describes the expected behavior of a specific module or user interaction within the system.

6.1 Overview

The ERP system supports multiple roles — **Admin**, **HR**, **Finance**, and **Employee**. Each role has a distinct set of features aligned with its operational responsibilities.

6.2 Functional Requirements Table

ID	Requirement Description	Priority	Module
FR-1	The system shall allow users to log in using username/email and password.	High	Auth
FR-2	The system shall verify credentials and issue a JWT token upon successful authentication.	High	Auth
FR-3	The system shall allow the Admin to manage users (add, update, delete, and assign roles).	High	Admin
FR-4	The HR module shall allow HR staff to register new employees and maintain employee records.	High	HR
FR-5	The HR module shall allow employees to mark daily attendance and track working hours.	High	HR
FR-6	The HR module shall allow employees to submit leave requests and HR to approve or reject them.	High	HR
FR-7	The HR module shall allow HR to evaluate employee performance and record ratings and feedback.	Medium	HR
FR-8	The Finance module shall automatically calculate payroll based on attendance, performance, and deductions.	High	Finance
FR-9	The Finance module shall allow Finance staff to process expenses, including submission, review, and approval.	High	Finance
FR-10	The Finance module shall handle client and vendor invoicing and track payment statuses.	Medium	Finance
FR-11	The Reporting module shall aggregate data from HR and Finance to generate analytical dashboards.	High	Reporting
FR-12	The Reporting module shall generate exportable reports in PDF and Excel formats.	Medium	Reporting
FR-13	The system shall restrict access to data and APIs based on JWT roles (Admin, HR, Finance, Employee).	High	Security
FR-14	The system shall allow the Admin to view overall analytics and monitor module activity.	Medium	Admin
FR-15	The Employee portal shall display personal information, attendance, salary, and performance data.	High	Employee
FR-16	The system shall allow the Finance and HR modules to communicate securely with the Reporting service via REST APIs.	High	Integration

FR-17	The system shall log all critical operations (logins, approvals, and updates) for audit purposes.	Medium	Security
FR-18	The system shall provide error messages and validation feedback for all input fields.	Medium	General
FR-19	The system shall support data export for payroll summaries and employee attendance.	Medium	Reporting
FR-20	The system shall provide dashboards summarizing HR, Finance, and overall business KPIs.	High	Reporting

6.3 Functional Requirement Categories

- **Authentication and Access Control:** Secure login, token generation, and role validation.
- **Human Resources Management:** Employee data management, attendance, leave, and performance tracking.
- **Finance Management:** Payroll processing, expense tracking, and invoicing.
- **Reporting and Analytics:** Real-time dashboards, performance insights, and exportable reports.
- **Administration:** User management, monitoring, and global configuration.

7 Test Cases for Functional Requirements

This section describes the test cases that validate each functional requirement of the Konecta ERP System. Each test case specifies the input, expected output, and success criteria to ensure the requirement functions as intended.

FR ID	Test Case Description	Test Input	Expected Output
FR-1	Verify user login with valid credentials.	Username and password.	User is successfully logged in and redirected to dashboard.
FR-2	Validate JWT token is generated upon successful login.	Valid credentials.	JWT token issued and stored in frontend securely.
FR-3	Test Admin user management functions (add, edit, delete, assign role).	Admin account and user details.	New user created/updated/deleted successfully and reflected in Auth DB.
FR-4	Verify HR can register new employees.	HR inputs employee data.	New employee record appears in employee list and HR DB.
FR-5	Verify attendance marking and logging.	Employee clicks “Mark Attendance”.	Attendance record created with timestamp and status “Present”.
FR-6	Test leave request submission and HR approval/rejection.	Employee submits leave form.	Leave status changes to “Pending” → “Approved/Rejected” after HR action.

FR-7	Verify HR can record employee performance evaluation.	HR enters rating and feedback.	Performance record saved and visible in employee profile.
FR-8	Test automatic payroll generation using attendance and performance data.	Payroll period input.	Net salary calculated and stored in payroll table.
FR-9	Validate expense management workflow.	Employee submits expense claim.	Expense recorded, pending for approval, status updates correctly.
FR-10	Verify invoicing and payment status tracking.	Finance creates new invoice.	Invoice generated, saved in database, and status updated (Paid/Unpaid).
FR-11	Test report aggregation from HR and Finance services.	Request analytics report.	Data successfully fetched from both services and displayed in dashboard.
FR-12	Validate PDF/Excel report export function.	Admin clicks "Export Report".	File successfully generated and downloaded.
FR-13	Verify access control for each role (Admin, HR, Finance, Employee).	Login using each role.	Restricted modules accessible only to authorized roles.
FR-14	Check Admin dashboard monitoring and analytics.	Admin login.	Dashboard displays live KPIs and user activity logs.
FR-15	Test employee self-service features.	Employee login.	Employee views personal attendance, salary, and performance data.
FR-16	Validate secure communication between HR, Finance, and Reporting services.	API call between services.	Encrypted REST response received with valid JWT verification.
FR-17	Test system logging for major operations.	Perform create/update/delete actions.	Log entry stored in system log file or DB with timestamp.
FR-18	Check form input validation and error handling.	Submit form with invalid fields.	Proper error message displayed (e.g., "Invalid Email").
FR-19	Test data export for payroll summaries and attendance logs.	Admin clicks export button.	CSV/XLS file generated containing requested data.
FR-20	Verify dashboard KPIs for HR, Finance, and overall performance.	Open analytics dashboard.	Metrics displayed correctly with up-to-date data.

7.1 Test Execution Notes

- All test cases should be executed on the staging environment before production deployment.
- JWT tokens must be validated in every API call to ensure secure communication.
- Testing tools such as **Postman** and **JUnit** are recommended for backend verification.
- **Manual UI testing** should be performed for dashboard features and form validation.

- Test results must be documented with "Pass" or "Fail" outcomes and attached screenshots when applicable.

8 Security Architecture

Table 6: Security Layers and Features

Security Component	**Description**
Authentication	JWT token-based authentication using Spring Security.
Authorization	Role validation for Admin, HR, Finance, and Employee.
Encryption	Data encrypted in transit (SSL/TLS) and at rest.
Password Protection	All user passwords hashed using BCrypt.
Incident Response	Logging, alerting, and monitoring of security events.
Compliance	GDPR-like data access and retention policies.

9 Role Policies and User Flows (JWT-based)

This ERP system implements authentication using **JSON Web Tokens (JWT)**. Each user logs in and receives a signed token that identifies their role (`ADMIN`, `HR`, `FINANCE`, or `EMPLOYEE`). Spring Security annotations such as `@PreAuthorize("hasRole('HR')")` enforce what each user can access at the API level.

9.1 Role Access Summary

Table 7: Role Access Policies

Role	**What They Can See**	**What They Can Do**	**Example APIs**
Admin	All dashboards (HR + Finance + Employee data)	Manage users, approve HR/Finance actions, and access all reports	`/api/auth/*`, `/api/hr/*`, `/api/finance/*`
HR	Employee records, attendance, leaves, and performance	Add/update employees, approve/reject leaves, record performance	`/api/hr/employees`, `/api/hr/attendance`, `/api/hr/leaves`, `/api/hr/performance`
Finance	Payroll, expenses, and invoices	Process payroll, manage expenses, generate financial reports	`/api/finance/payroll`, `/api/finance/expenses`, `/api/finance/invoices`
Employee	Personal dashboard (attendance, leave, salary, performance)	Mark attendance, request leave, view salary, and performance	`/api/hr/attendance`, `/api/hr/leaves`, `/api/finance/payroll/{id}`

9.2 Access Control Policy Table

Table 8: API Access Control Summary

API Prefix	Accessible By	Description
`/api/auth/*`	Admin	User and authentication management
`/api/hr/employees*`	HR, Admin	Employee CRUD operations
`/api/hr/attendance*`	HR, Employee	Attendance management
`/api/hr/leaves*`	HR, Employee	Leave requests and approvals
`/api/hr/performance*`	HR, Admin	Performance evaluations
`/api/finance/payroll*`	Finance, Admin	Payroll processing
`/api/finance/expenses*`	Finance	Expense management
`/api/finance/invoices*`	Finance	Invoice management
`/api/reports/*`	Admin, HR, Finance	Reporting endpoints

9.3 JWT Security Flow

1. User sends login credentials to `/api/auth/login`.
2. Authentication Service validates credentials and returns a JWT token.
3. The frontend stores the token securely (e.g., localStorage).
4. Each request includes the token in the `Authorization` header.
5. The API Gateway verifies the token before routing.
6. The backend checks user role claims using Spring Security.
7. Access is granted or denied based on token claims.

10 Technologies Used

Table 9: Technology Stack

Layer	Technology
Frontend	Angular 18, TypeScript
Backend	Spring Boot 3.5.6, Java 21
Reporting Service	ASP.NET Core 8.0, C#, Entity Framework
Database	PostgreSQL
Security	Spring Security, JWT
API Gateway	Spring Cloud Gateway
Service Discovery	Eureka Server
Configuration Management	Spring Cloud Config
Deployment	Docker, Jenkins (CI/CD)