

CSc3350 Software Development
Software Design Document
Group Team Project Fall 2025

TEAM 404

Team Members: Andrew Dana, Erick Maldonado, Hien Dao

Date: (12/08/2025)

Table of Contents

1.	<i>INTRODUCTION</i>	4
1.1	Software Purpose	4
1.2	Software Capabilities	4
2.	<i>DATABASE SCHEMA DIAGRAM</i>	4
3.	<i>JAVA CLASS DIAGRAMS</i>	7
UI LAYER:	Interact with user	7
SERVICE LAYER:	Business logic	7
DAO LAYER:	Query MySQL Server	8
MODEL LAYER:	Carrying data	10
SECURITY LAYER:	Authenticate, authorize user, give session context, hash password	12
UTILS LAYER		12
4.	<i>PROGRAMMING TASKS</i>	13
5.	<i>TEST CASES</i>	15
6.	<i>SEQUENCE DIAGRAMS</i>	17
7.	<i>APPENDIX</i>	18
7.1	Definitions and Acronyms	18
7.2	Other	18

1. INTRODUCTION

1.1 Software Purpose

The Employee Management System (EMS) for Company Z is designed to replace manual MySQL script management with a secure, and user-friendly application. The system is scalable, company Z can expand the number of employees, types of employees (Full time, Part Time, Contractor, etc). HR can add status of employees (Active, On Leave, etc) if needed. More types of employee contacts in the future is possible (LinkedIn, etc). Upgrade to include other user roles (HR Admin, Regular Employee, Division Manager, Region Manager, etc) can be added easily.

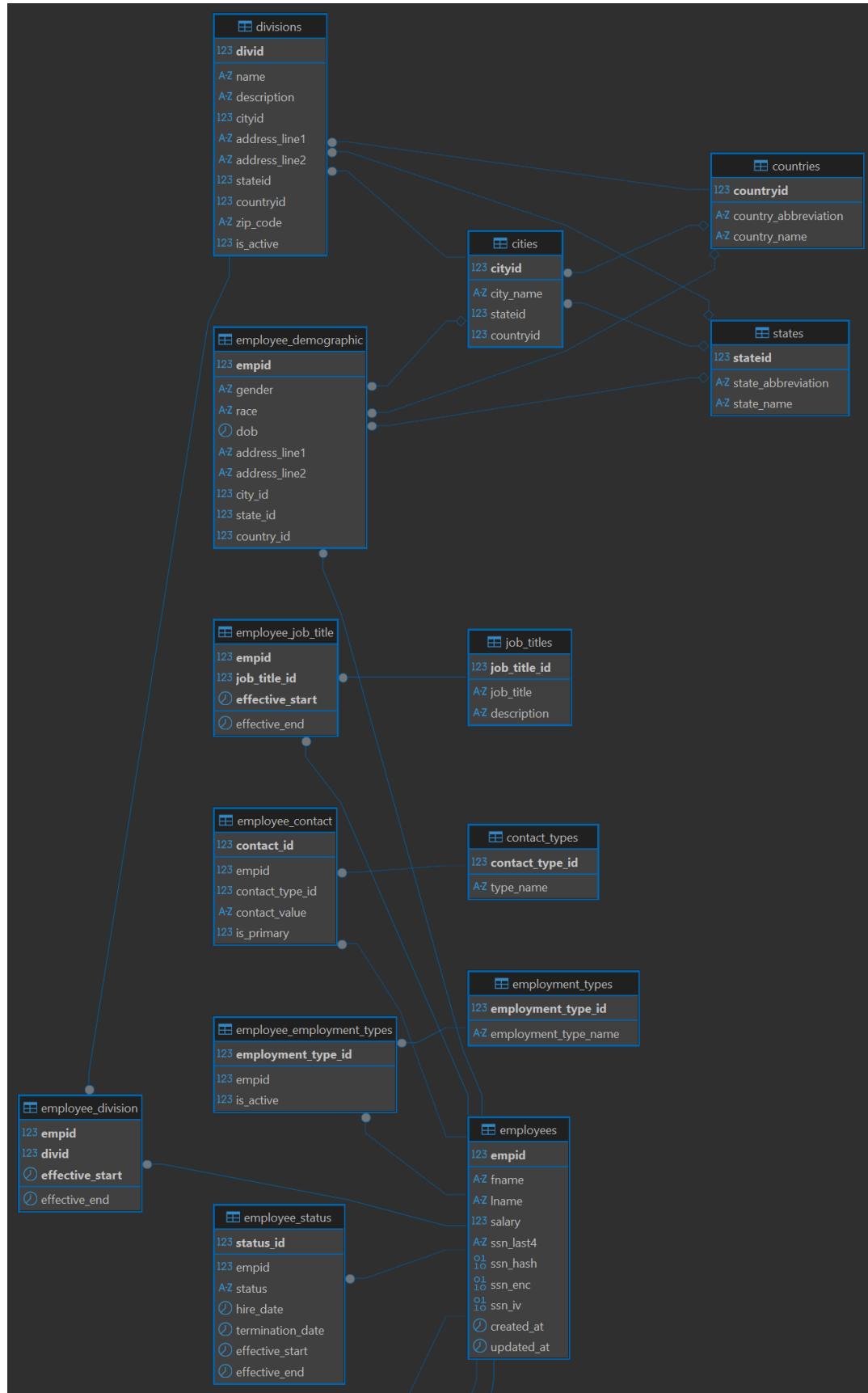
1.2 Software Capabilities

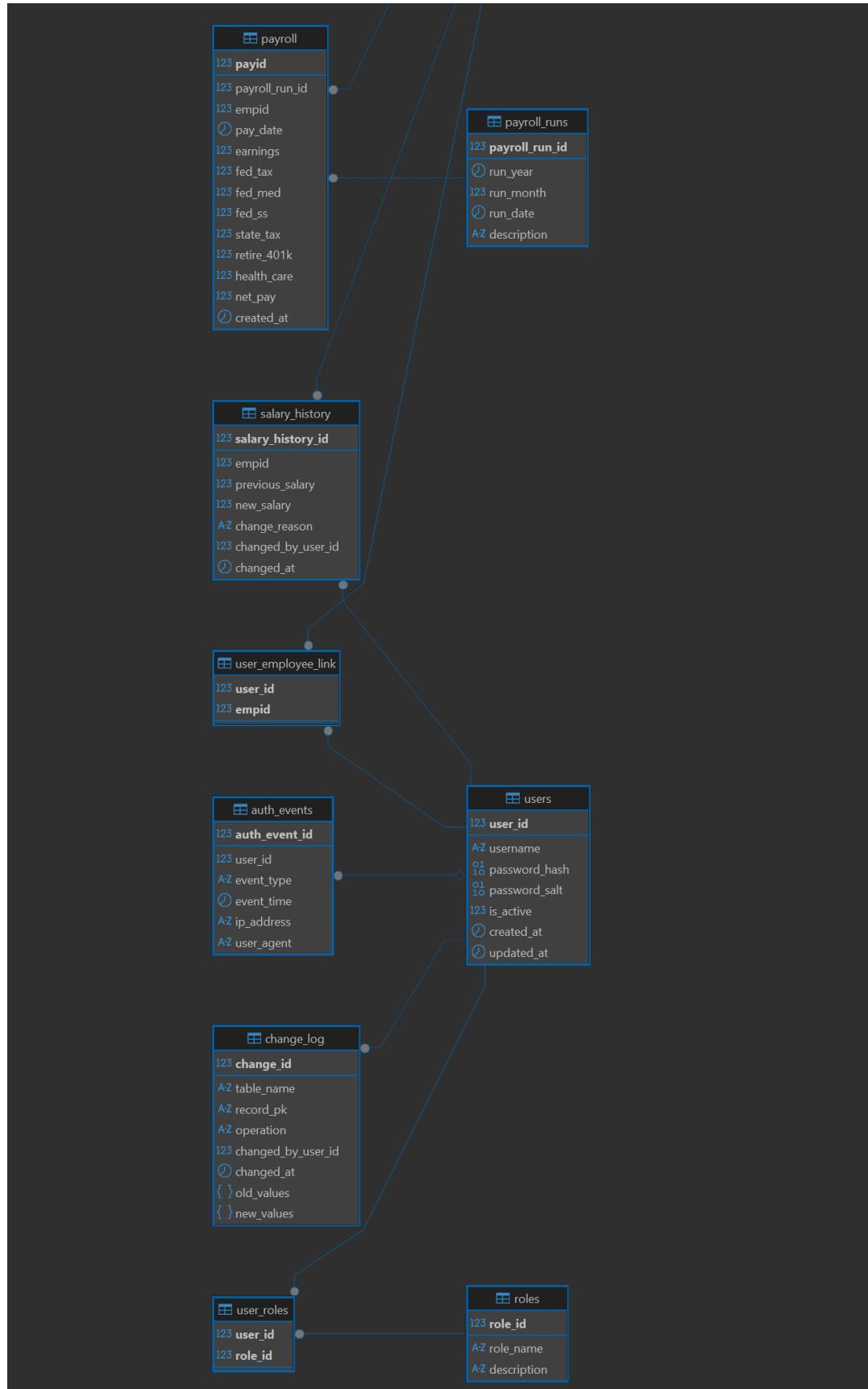
The system provides:

- **HR Administrators:** Full CRUD capabilities over employee data and payroll, salary adjustments, and reporting.
- **Employees:** Secure access to view their personal information and pay history.

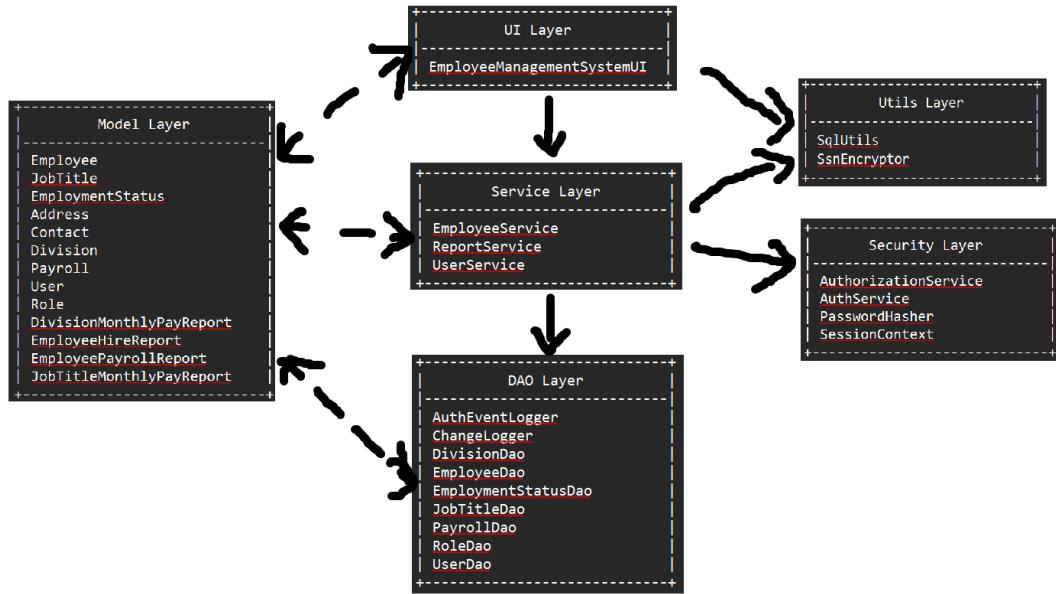
The intended audience includes HR administrators and employees.

2. DATABASE SCHEMA DIAGRAM





3. JAVA CLASS DIAGRAMS



UI LAYER: Interact with user

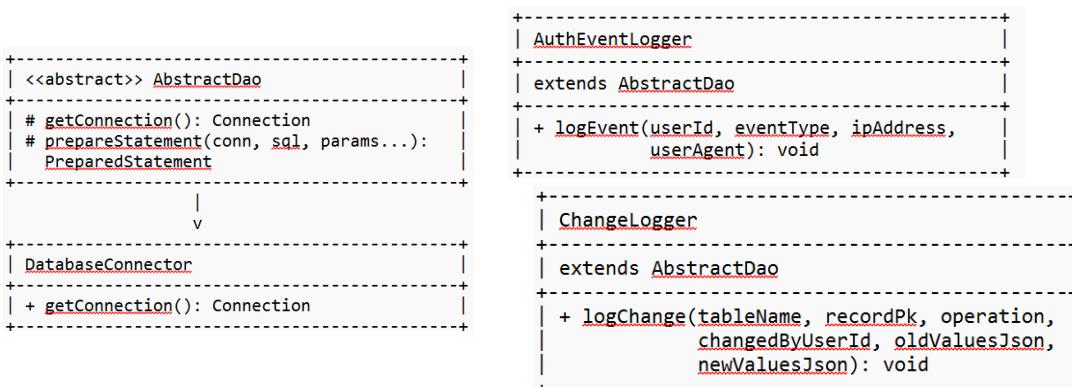
```

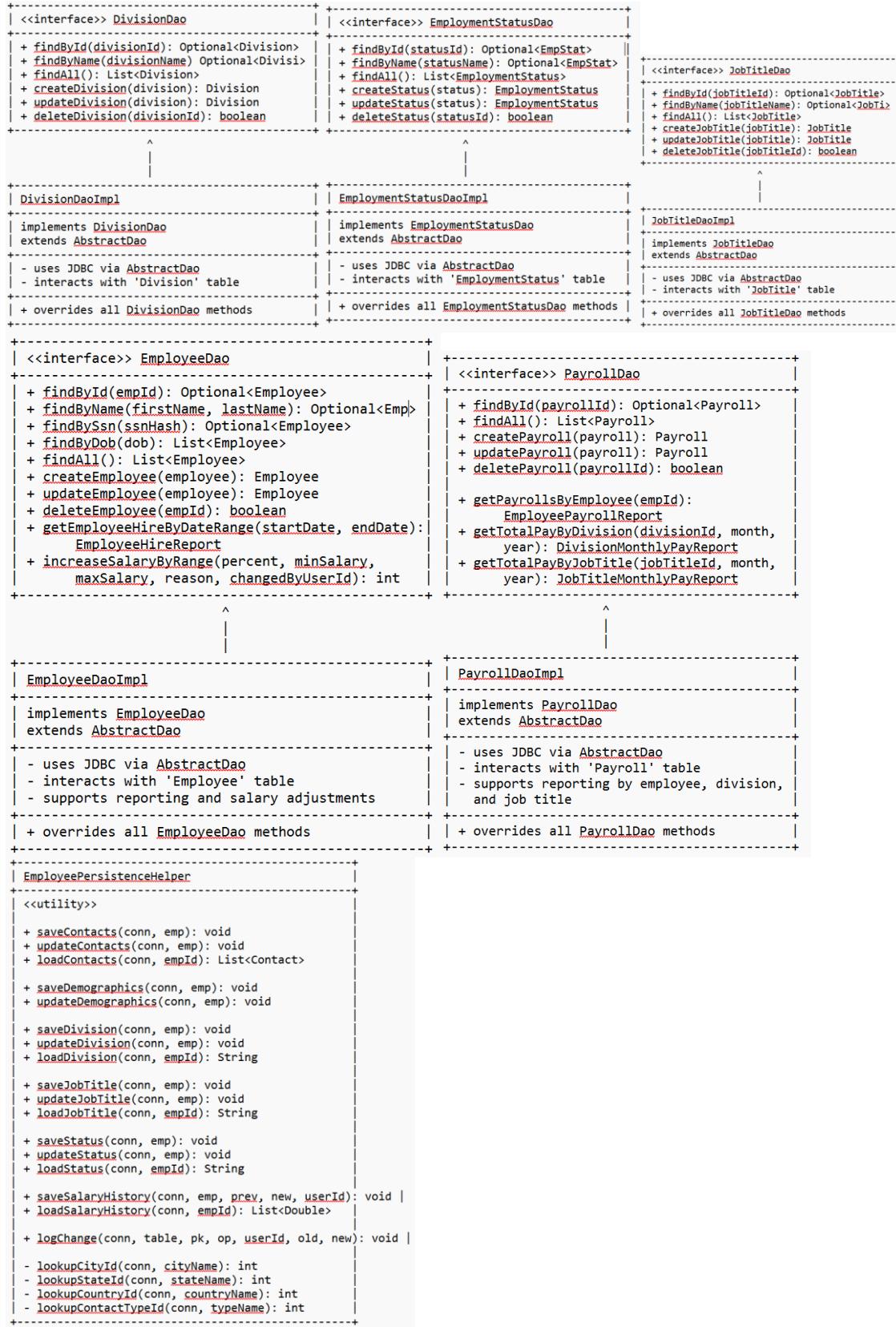
+-----+
| EmployeeManagementSystemUI
+-----+
| + start(): void
| + showLoginScreen(): void
| + showDashboard(): void
| + createMenuBar(): void
| + createDashboardTab(): void
| + createEmployeesTab(): void
| + createPayrollTab(): void
| + createReportsTab(): void
| + createUsersTab(): void
| + showAbout(): void
| + showInfo(message: String): void
| + showError(message: String): void
+-----+
  
```

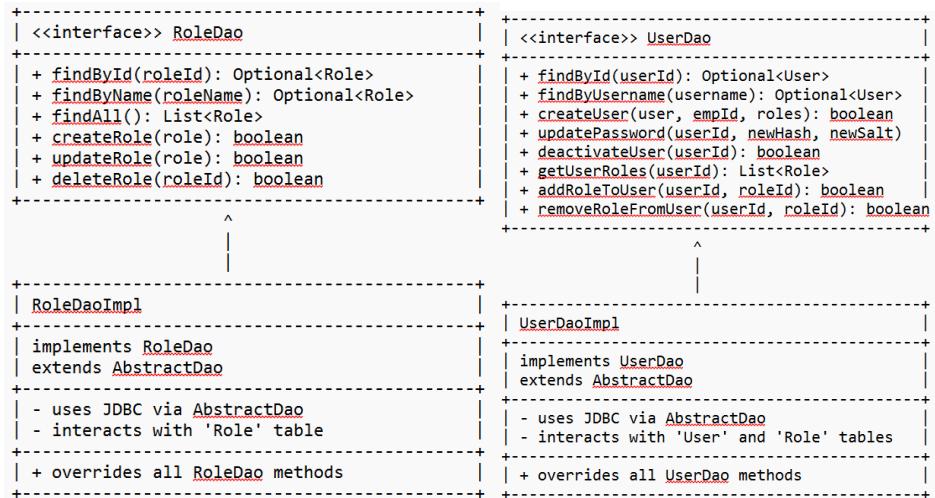
SERVICE LAYER: Business logic



DAO LAYER: Query MySQL Server







MODEL LAYER: Carrying data

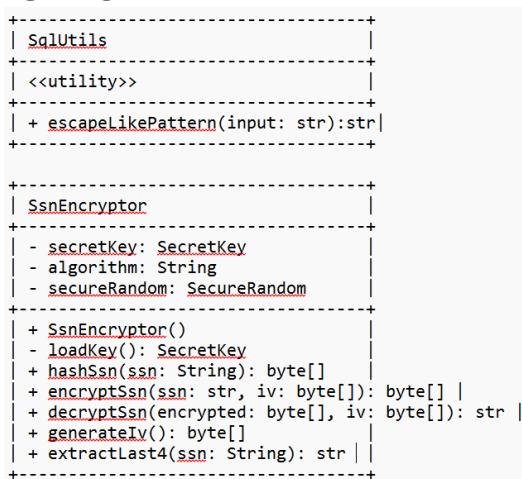




SECURITY LAYER: Authenticate, authorize user, give session context, hash password



UTILS LAYER



4. PROGRAMMING TASKS

1. Database Setup and Schema

- a. Create SQL tables: users, roles, user_roles, employees, divisions, job_titles, payroll, etc
- b. Define primary keys, foreign keys, and indexes for efficient search.
- c. Secure SSN storage with encryption (AES) and hashing (SHA-256).

2. Authenticate and Authorization Portal

- a. Implement login using users and roles tables.
- b. Hash passwords with BCrypt + salt; store securely.
- c. Compare entered password hash with stored hash.
- d. Assign privileges based on role (HR_ADMIN vs REGULAR_EMPLOYEE).
- e. Sanitize input with prepared statements to prevent SQL injection.

3. Session Management

- a. Create SessionContext class to track userId, role, employeeId, login time, last activity.
- b. Enforce session timeout (e.g., 15 minutes inactivity).
- c. Provide logout functionality to invalidate sessions.

4. Search Employee Data Module

- a. For employees: return only their own record using emp_id, mask SSN except last 4 digits, return Employee object.
- b. For HR admin: search by emp_id, name, ssn_hash, or dob; return array of Employee objects; provide links to update modules.

5. Update Employee Data Module

- a. Restrict access to HR admin only.
- b. Sanitize input with prepared statements.
- c. Update employee fields (division, job title, contact info, salary).
- d. Log changes in an audit table.

6. Update Employee Salary Module

- a. Restrict access to HR admin only.
- b. Accept salary range boundaries and percentage increase.
- c. Execute SQL update with transactional safety.
- d. Return count of affected rows.

7. Payroll Management

- a. Insert payroll entries for each employee per pay period.
- b. Calculate net pay = earnings – deductions.
- c. Ensure payroll entries are linked to employees by emp_id.
- d. Provide DAO methods for CRUD operations on payroll.

8. Reporting Module

- a. EmployeePayrollReport: show employee's pay history sorted by pay date.
- b. DivisionMonthlyPayReport: total pay by division for month/year.
- c. JobTitleMonthlyPayReport: total pay by job title for month/year.
- d. EmployeeHireReport: employees hired within a date range.

9. User Interface (JavaFX)

- a. Build login screen with username/password fields.

- b. HR Admin dashboard: search employees, edit data, salary increase tool, reports.
- c. Employee dashboard: view profile and payroll history.
- d. Ensure role-based navigation.

10. Security Utilities

- a. Implement PasswordHasher (BCrypt + salt).
- b. Implement SsnEncryptor (AES encryption, IV handling, SHA-256 hash).
- c. Implement SqlUtils for safe LIKE queries.
- d. Implement AuthorizationService for role/self checks.

5. TEST CASES

Update Employee Data

Task Description:

Create a function that allows editing any employee information (e.g., name, department, or salary) using their employee ID. It should validate that the employee exists and the field being updated is valid.

1. Function searches for employee by ID.
2. If found, updates the specified field with the new value.
3. Saves or reflects changes in data storage.

Success Cases:

- Employee 101's department updated from "Sales" to "Marketing."
- Salary for employee 102 successfully changed to 75,000.

Fail Cases:

- Attempt to update non-existent employee ID returns "Employee not found."
- Invalid field (e.g., "height") rejects with an error message.

b) Search for Employee (Admin User)

Task Description:

Implement a function that allows admin users to search for employees by ID, name, or department.

1. Admin enters a search query.
2. System filters employee data and returns matching results.
3. If no match found, display "Employee not found."

Success Cases:

- Searching "103" returns employee record with ID 103.
- Searching "Finance" lists all Finance department employees.

Fail Cases:

- Query for a non-existent name returns "Employee not found."
- Non-admin user attempts search returns "Access denied."

c) Update Salary for Employees Below a Certain Amount

Task Description:

Create a function that increases the salary of all employees whose current pay is below a given threshold.

1. Function iterates through all employees.
2. Compares current salary to threshold.
3. Updates only those with salaries below that amount.

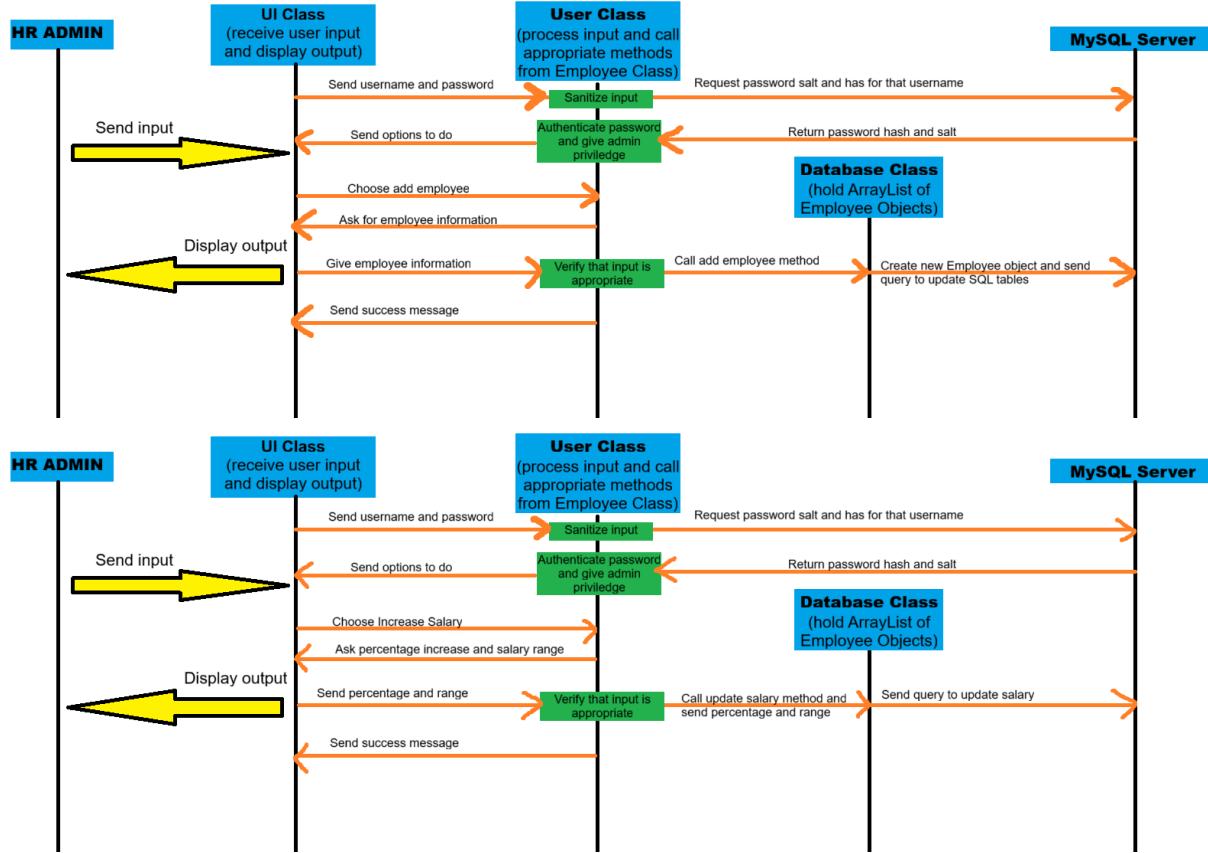
Success Cases:

- All employees earning < \$50,000 now updated to \$55,000.
- Function returns the count of employees updated.

Fail Cases:

- No employees under threshold returns 0 updates made.
- Invalid input (e.g., non-numeric salary) returns error message.

6. SEQUENCE DIAGRAMS



7. APPENDIX

7.1 Definitions and Acronyms

- CRUD: Create, Read, Update, Delete operations.
- EMS: Employee Management System.
- HR Admin: Human Resources Administrator with elevated privileges.
- SSN: Social Security Number.
- DAO: Data Access Object, a programming pattern for database interaction.
- DTO: Data Transfer Object, used to carry data between layers.
- BCrypt: A password hashing function.
- IV: Initialization Vector, used in encryption.
- UX: User Experience.
- MySQL: Relational database management system.

7.2 Other

- Configuration details: application.properties keys for configurations.
- Seed data: Check src\main\resources\db