Employee Management System

DESIGN DOCUMENT

By Raymond Ynoa for CEN-4802C Final Project

## Introduction

### Project Goal

Build a Java project that functions as a basic employee management system that includes practices in code development, distributed version control, build automation, unit testing, release automation (CI/CD), application development, operations management, and systems monitoring.

### User Experience

A User should be able to Structured Query Language within the IDE to retrieve and insert information onto an employee table.

### Platform

The project is built with the Windows platform in mind using the IntelliJ IDEA integrated development environment.

### Development Software

* Spring Boot framework for initialization
* Maven for build automation
* MySQL Workbench for database
* Hibernate framework to interact with database
* GitHub for version control
* Jenkins for pipeline

## Planned Project Structure

employee-management-system/

│

├── src/

│ ├── main/

│ │ ├── java/

│ │ │ └── com/

│ │ │ └── company/

│ │ │ └── employeemanagementsystem/

│ │ │ ├── controller/

│ │ │ │ └── EmployeeController.java

│ │ │ ├── model/

│ │ │ │ └── Employee.java

│ │ │ ├── repository/

│ │ │ │ └── EmployeeRepository.java

│ │ │ ├── service/

│ │ │ │ └── EmployeeService.java

│ │ │ └── EmployeemanagementsystemApplication.java

│ │ │

│ │ ├── resources/

│ │ │ ├── static/

│ │ │ ├── templates/

│ │ │ ├── application.properties

│ │ │ └── application.yml

│ │ │

│ │ └── test/

│ │ └── java/

│ │ └── com/

│ │ └── company/

│ │ └── employeemanagementsystem/

│ │ ├── controller/

│ │ │ └── EmployeeControllerTest.java

│ │ └── service/

│ │ └── EmployeeServiceTest.java

│ │

│ ├── test/

│ │ └── java/

│ │ └── com/

│ │ └── company/

│ │ └── employeemanagementsystem/

│ │ └── EmployeemanagementsystemApplicationTests.java

│ │

│ ├── target/

│ └── ...

│

├── .gitignore

├── Dockerfile

├── kubernetes/

│ ├── deployment.yaml

│ └── service.yaml

├── mvnw

├── mvnw.cmd

├── pom.xml

└── README.md

## Development Timeline

| **#** | **Assignment** | **Type** | **Status** | **Finish By** | **Notes** |
| --- | --- | --- | --- | --- | --- |
| 1 | Design Document | Other | Finished | Dec 7, 2023 | Documentation of project |
| 2 | Dependencies & Plug-ins | Coding | Finished | Dec 7, 2023 | Persistence, frameworks, and other tools. |
| 3 | Java Classes | Coding | Finished | Dec 7, 2023 | The code of the project |
| 4 | GitHub | Other | Finished | Dec 7, 2023 | Commit and push app onto repository |
| 5 | Database | Other | Finished | Dec 7, 2023 | Organized collection of structured information |
| 6 | JUnit | Coding | Finished | Dec 7, 2023 | Java unit testing framework |
| 7 | Jenkins | Coding | Finished | Dec 7, 2023 | Open source automation used for pipeline. |
| 8 | Docker | Coding | In progress | Dec 7, 2023 | Containerize the application |
| 9 | Kubernetes | Coding | Not started | Dec 7, 2023 | Manage the containerized workload |
| 10 | DataDog | Other | Not started | Dec 7, 2023 | Monitor performance of database and tools |