GENESIS - Learning Outcome & Shadow Project Summary Report

LTTS
GLOBAL
ENGINEERING
ACADEMY

L&T Technology Services

Details

PS No.	Release Date	Prepared. By	Reviewed By	To be Approved	Remarks/Revision Details
99004924	13-10-2021	Supriya Yadav	Sandeep Kumar Merala		
99004930	13-10-2021	Palaash Atri	Sandeep Kumar Merala		
99004934	13-10-2021	Mayank Gupta	Sandeep Kumar Merala		
99004936	13-10-2021	Shubham Phansekar	Sandeep Kumar Merala		

Contents

Shadow project -1 -EMS: Employee Management System[Team]

Introduction

Employee Management System is an application-based system, having two applications developed, one for the admin to manage employee details and another for employees to mark their attendance on a daily basis. This web application is built on Spring Framework, it discusses a much easier and less-time consuming approach for employees to mark their attendance, it also allows employees to update their data using some restrictions.

The project consists of three modules:

- Admin
    1. Admin can update any information of any employee in the organization.
    2. Admin set an employee inactive rather than deleting.
    3. Admin can accept or reject an employee's request for regularization of attendance.
- Employee
    1. Employees can update any information related to it with some restrictions.
    2. Employees can send a request for regularization of attendance.
- Attendance
    1. An intermediate module.
    2. Allows employees to send a request for regularization of attendance.
    3. Admin can accept or reject an employee's request for regularization of attendance.

Module/s

Modules linked to the Shadow project-

- ❖ Java Programming
- ❖ Spring Framework
- ❖ Unit Testing using Junit5

Topic and Subtopics

- ❖ Implementing it in the Java programming language helped us learn best practices of doing things and apply the
    - ➢ Packages and Classes(DAO,DAO Service,DAO Service Implementation) Java programming
    - ➢ SDLC:
        - ▪ Requirements Gathering
        - ▪ Design
            - ● High level
            - ● Low level
        - ▪ Implementation
        - ▪ Unit Testing
        - ▪ Deployment & Maintenance
    - ➢ Testing & debugging tools:
        - ▪ Junit

- ➢ IDE:
  - VS code
  - Spring Tool Suite

- ➢ Source code management system:
  - GitHub

Objectives & Requirements

High-Level Requirements

ID	Description	Category	Status(Implemented/Future)
HR01	Login Functionality	Technical	Implemented
HR02	Employee View	Technical	Implemented
HR03	Admin View.	Technical	Implemented

Low-level Requirements

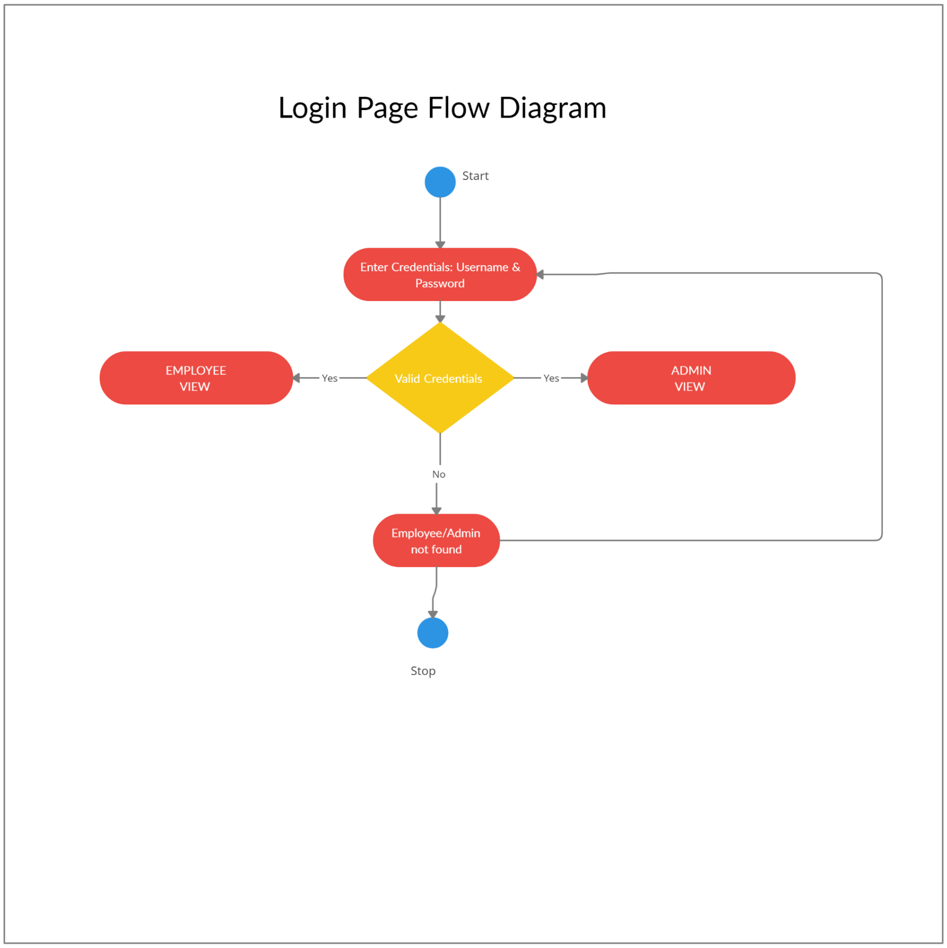
ID	Description	HLR_ID	Status (Implemented/Future)
LR01	Login Page	HR01	Implemented
LR02	User Authentication /Authorisation	HR01	Implemented
LR03	Show Employee details	HR02	Implemented
LR04	Update Employee details	HR02	Implemented
LR05	Update attendance	HR02	Implemented
LR06	Update Profile Picture	HR02	Implemented
LR07	New Employee Registration Page	HR03	Implemented
LR08	View All employees	HR03	Implemented
LR09	Update Employee	HR03	Implemented

LR10	Delete Employee(Set Active/Inactive state)	HR03	Implemented
LR11	Send Request for Attendance	HR02	Implemented
LR12	Manage Employee Attendance	HR03	Implemented

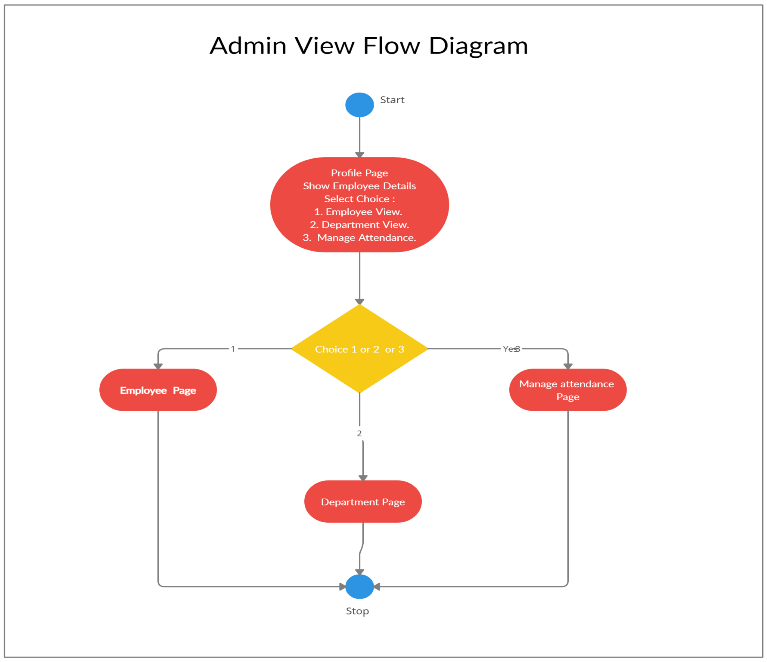
Design

A. High Level Design

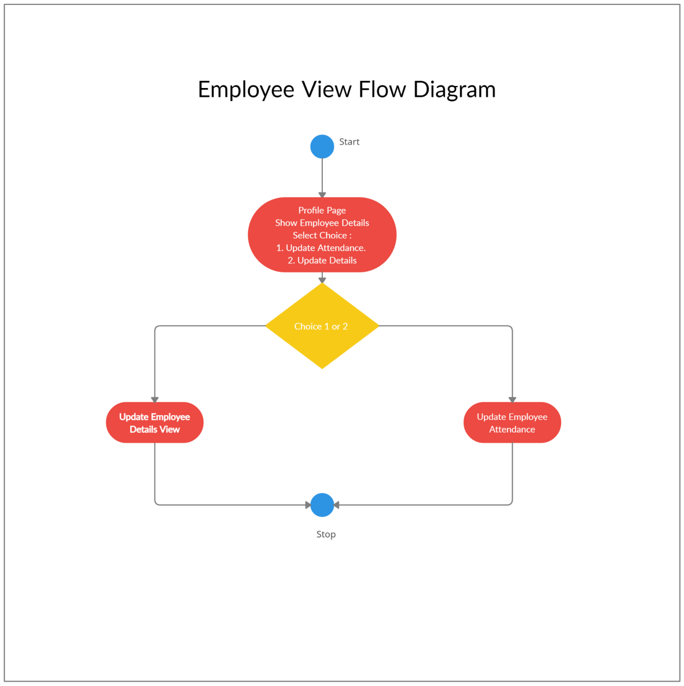
1. System Diagram



2. Admin View Flow Diagram

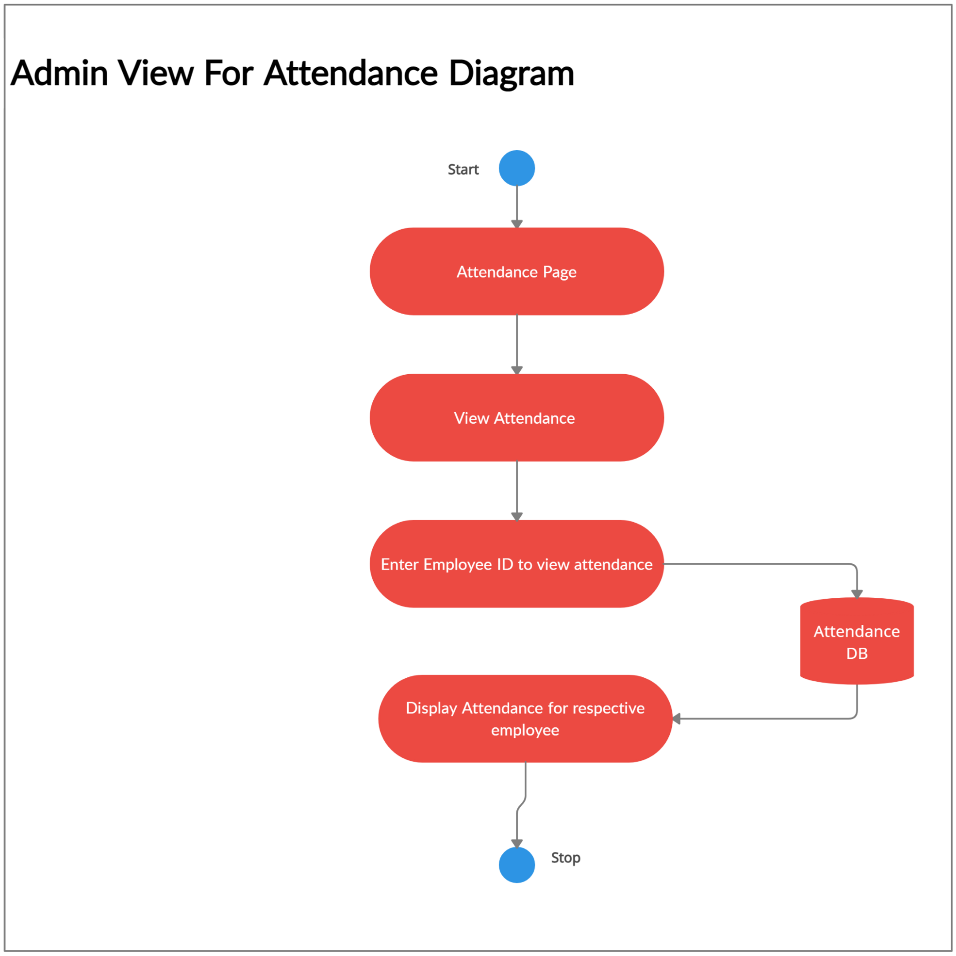


3. Employee View Flow Diagram

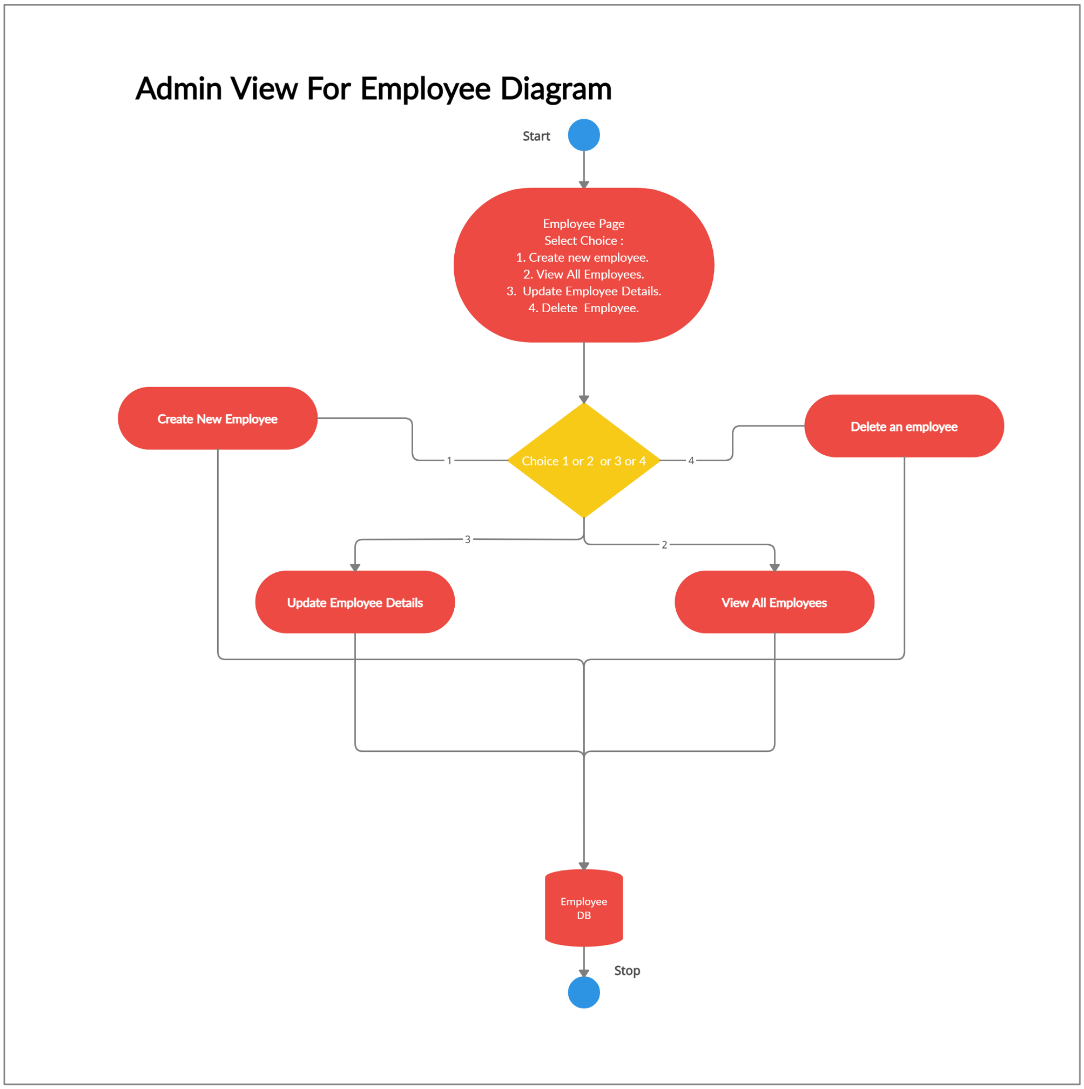


B. Low-Level Design

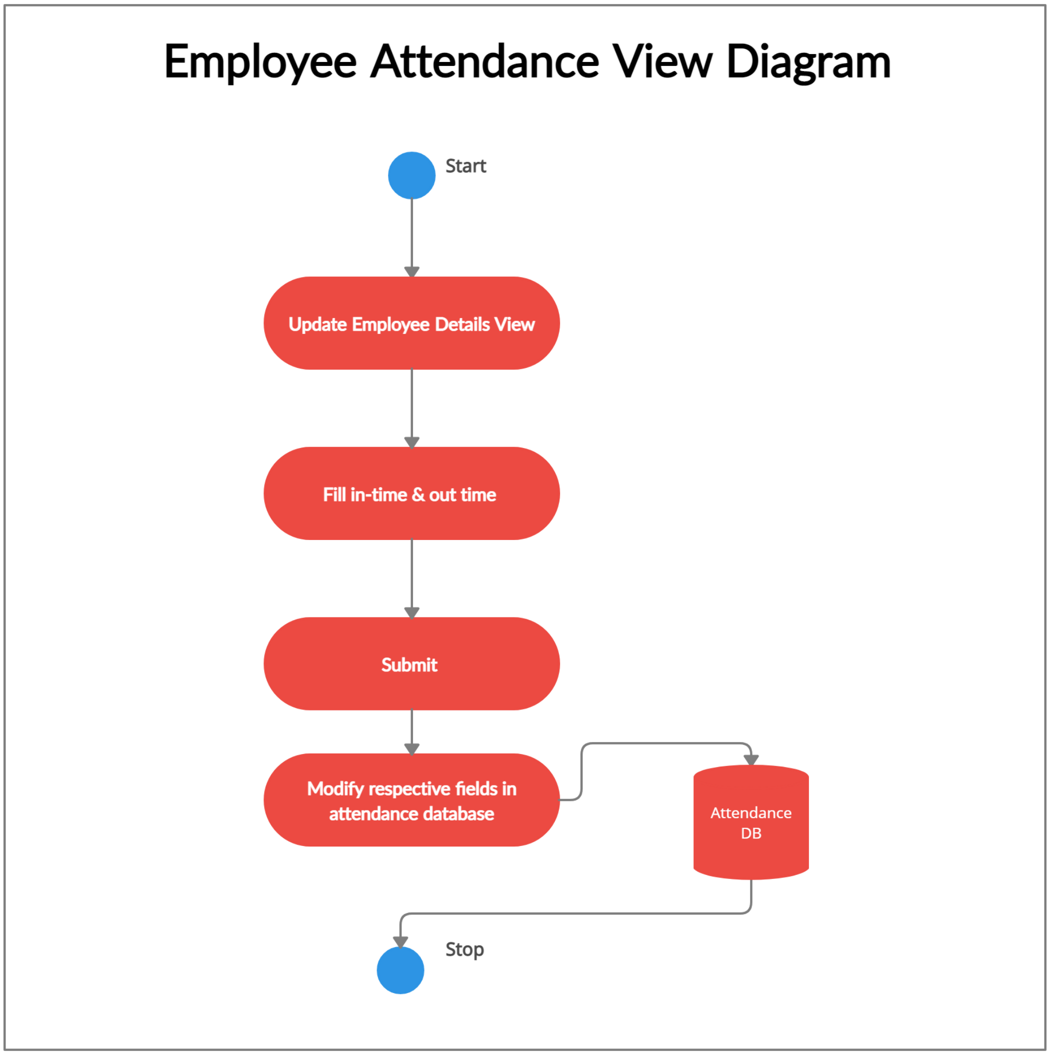
1. Admin view for Attendance Diagram


Admin View For Attendance Diagram
Start
Attendance Page
View Attendance
Enter Employee ID to view attendance
Attendance DB
Display Attendance for respective employee
Stop

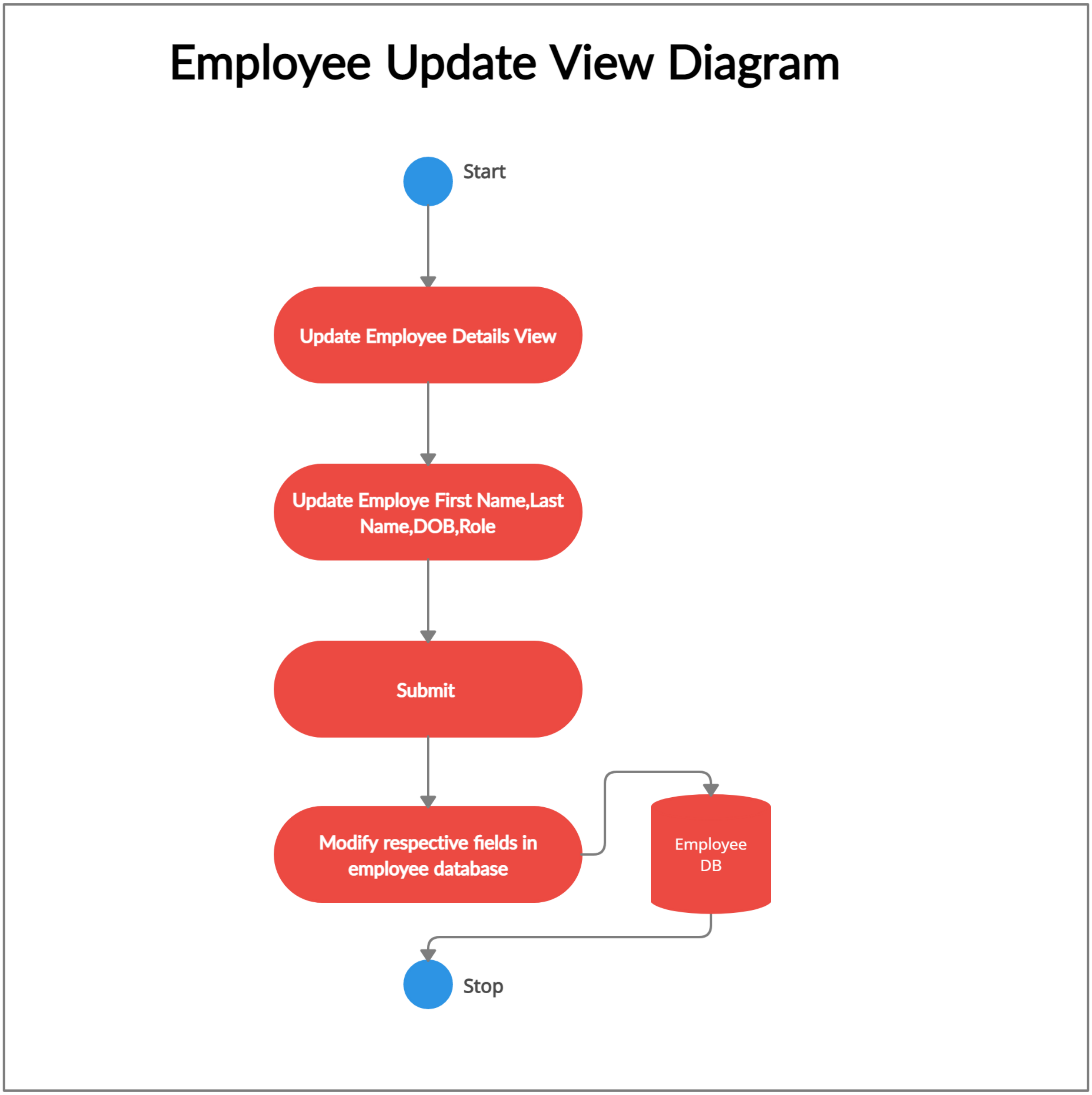
2. Admin view for Employee Diagram



3. Employee Attendance View Diagram


Employee Attendance View Diagram
Start
Update Employee Details View
Fill in-time & out time
Submit
Modify respective fields in attendance database
Attendance DB
Stop

4. Employee Update View Diagram


Employee Update View Diagram
Start
Update Employee Details View
Update Employe First Name,Last Name,DOB,Role
Submit
Modify respective fields in employee database
Employee DB
Stop

Database Design

1. Departments (department_id int primary key, department_name varchar(255), department_location varchar(255));

2. Employees (emp_id int primary key, first_name varchar(255), last_name varchar(255), username varchar(255) unique, password varchar(255), date_of_joining date, gender char(1), date_of_birth date, role varchar(255), department_id int, emp_photo varchar, foreign key(department_id) references Departments(department_id));

3. Attendance (att_id int primary key, emp_id int, in_time date, out_time date, reason varchar(255), status varchar(255), foreign key(emp_id) references Employees(emp_id));

Test Plan

Test plan table

Test ID	Description	Exp I/P	Exp O/P	Actual O/p	Type Of Test
01	Check Controller	controller object	true (Boolean value)	true (Boolean value)	Technical
02	Check Port	port number	true (Boolean value)	true (Boolean value)	Technical

Implementation Summary

❖ The project is based on Java Programming and its Spring Framework to build an Employee Management System application.

❖ Once the application starts, it will allow a user to log in:

1. When a user is an admin then
   a. he/she can create, update and set active/inactive an employee
   b. accept or reject the request for attendance regularisation.

2. When a user is an employee then
   a. he/she can update specific information with certain restrictions.
   b. send attendance requests for regularisation.

❖ Once the task is done the user(admin/employee) can log out of the system

Git Link

Link: https://github.com/netizener/genesis-ems.git

Unit Testing

Unit Testing was done with the help of the JUnit5. The detailed test plan is given in the Test Plan section. All the test cases are passing successfully.

Challenges faced and how were they overcome

No.	Challenge	Solution
1	Integration issues	Debugging the code
2	Image/file uploading.	Solved it by visiting various learning resources and using multipart file concept
3	Attendance Module From Both Employee and Admin View	Solved it by visiting various learning resources.