**Employee Maintenance System**



**Team members:**

1. Sumit Chavan

2.Rannvijay Kumar

3. Manish Jawage

4. Kaustubh Raigaonkar

Table of Contents

[1 Introduction 4](#_Toc454875701)

[1.1 Setup Checklist for Mini Project 4](#_Toc454875702)

[2 Problem Statement 5](#_Toc454875704)

[2.1 Objective 5](#_Toc454875705)

[2.2 Abstract of the project 5](#_Toc454875706)

[2.3 Functional components of the project 5](#_Toc454875707)

[2.4 Technology used 6](#_Toc454875708)

[3 Implementation in JEE LOT 7](#_Toc454875709)

[3.1 Summary of the functionality to be built 8](#_Toc454875710)

[3.2 Guidelines on the functionality to be built 8](#_Toc454875711)

[3.3 Evaluation and assessment parameters 1](#_Toc454875712)0

**1 Introduction**

This document outlines a case study for sprint 2 project. The project is to develop an online banking system as integration of all independent microservices. This document contains the work flow of the system and gives guidelines on how to build the functionality gradually in each of the course modules.

## Setup Checklist for Sprint 2 Project

Minimum System Requirements

* Intel Pentium 90 or higher (P166 recommended)
* Microsoft Windows 95, 98, or NT 4.0, 2k, XP, Windows 7 or higher
* Memory: 4GB of RAM (8GB or more recommended)
* Internet Explorer 6.0 or higher or Chrome 55 or above

**Software System Requirements**

* JDK 1.8
* STS 3.9
* MAVEN
* Apache Tomcat
* Postman Master
* MySQL or H2 Database
* Visual Studio

**2 Problem Statement**

## 2.1 Objective

Development of Employee Maintenance System (EMS)

## 2.2 Abstract of the project

This project is aimed at developing an online Employee Maintenance System (EMS)for employees and Company administrator. EMS can be used to search for Employees based on search condition, add individual employee, modify an existing employee details and display all employee details across locations within an organization. Employees can apply for leave and the leave record will be updated and can be approved/rejected by manager.

The model followed was an agile model. Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product. Agile Methods break the product into small incremental builds. These builds are provided in iterations.

## 2.3 Functional components of the project

Following is a list of functionalities of the system.

There are two types of users who would access the system viz. Company Administrator and Employees. Each one of them would have some exclusive privileges as follows.

1. Company Administrator

A set of administrators are assigned for managing the system. An admin has been assigned a set of privileges to manage the system. An admin can perform the following functionalities:

* + Login to the system using his/her credentials.
  + Add individual employee details by accepting all the field values from end user as listed below and inject the values into database table if data are valid else display an appropriate error messages.
  + Modify Employee details.
  + Display all employee details.

1. Employees
   * Login to the system using his/her credentials.
   * Search an employee details based on any of the fields - ID, First Name, etc.
   * Apply for leave

**3 Implementation**

## 3.1 Summary of the functionality to be built

The participants need to develop an online Employee Maintenance System by integrating different micro-services in the backend and Angular in frontend.

## 3.2 Guidelines on the functionality to be built

The functionality and components to be built are provided below:

1. Databases to be created:
   1. Create the following database tables:
      1. Employee: This will contain details of the Employee.
      2. Leave: This will contain details of all the leaves applied by each employee (applied, approved, or rejected).
      3. Login: This will contain all the Login details.
   2. The structure of the above listed tables is as follows
      1. **Employee :**  empId NUMBER(10) password VARCHAR2(255), address VARCHAR2(255), department VARCHAR2(255), designation VARCHAR2(255), doj Date, dob Date, fname VARCHAR2(255), lname VARCHAR2(255), grade VARCHAR2(25), gender VARCHAR2(255), maritalStatus VARCHAR2(255), salary NUMBER(15), managerId NUMBER(10).
      2. **Leave:** leaveId NUMBER ,Tran\_description VARCHAR2(255), DateofTransaction DATE , TransactionType VARCHAR2(255) TranAmount NUMBER(15) ,Account\_No NUMBER(10)
      3. **Login**: username VARCHAR2(25), password NUMBER (10), role VARCHAR2(25)
2. Course: OOP & UML **:**
   1. Develop relevant Use case and Class diagrams for the OBS application.
3. Angular 6, Web Basics

**1.Login / Sign In**: Where the employee/admin will enter his/her credentials, if valid, will be redirected to the Homepage, else appropriate error message will be displayed in the same page.

**2.Home Page**:

On successful user authentication the homepage must be displayed according to the type of user. For the admin, the Homepage would display the employee table with the following links

* + - * Home
      * Search for employees
      * Apply for leave
      * Show applied Leaves

**3**.**Apply for leave page:**

This page is used by Employees to apply for leaves and to check the status of already applied leaves. The Employee with the role of manager can approve / reject the leave.

**4.Check Leave Status**

* + - * Displays the currently applied leaves by an employee.

**5. Approve / Reject Leave**

* + - * Allows an employee to approve / reject the leaves applied the other employees to which he/she is the manager.

**For the Bank Admin following pages must be provided**

**1.Add new Employee link:**

* This page must accept employee ID, password, first name, last name, address, gender, designation, grade, date of joining, date of birth, manager ID.
* Delete details of employee/s:

1. This link deletes the employee details of a particular or all employees, as applicable.

**4. JPA WITH HIBERNATE + SPRING BOOT + MICROSERVICES**

**Employee CRUD Module:** This micro-service is used to perform all the CRUD operations related to the Employees which will be managed by the Admin. The respective operations will store the corresponding data in the Employee database.

* Following is the list of all the functionalities in this module:

1. Add Employee
2. Delete Employee
3. Update Employee
4. Search Employee
5. View all Employee details

**Leave Module:** This micro-service is used by Employees to apply for leaves and to check the status of already applied leaves. The Employee with the role of manager can approve / reject the leave.

* Following is the list of all the functionalities in this module:

1. Apply for Leave

2. Check leave status

3. Approve / Reject leaves

**Login Module:** This micro-service is used to allow either the Employee or the Admin to log into the system using appropriate credentials.

* Following is the list of all the functionalities in this module:

1. Login