EXPENSE MANAGEMENT SYSTEM

### PREPARED FOR

JEE2 Cloud Batch Pseudo Live Project Evaluation

### PREPARED BY

Anurag

Pritam

Pradeepta

Vivek

Zeeshan

# 2. Scope

Using the Online Expense Management System, a user can claim any type of expense made while working on any project. An employee can login and provide specific details for the expense made and apply for the claim which is then further processed to the finance team for their approval/ rejection. The admin has the right to add, modify, delete the project as well as creating various expenses based on there types and requirements.

## **Out of Scope:**

## The system involves basic operations concerning expenses and does not involve the setting of limits and assigning budgets to the project. It does not cover insurance, medical and other perks, and their business logic. It does not allow tracking or update system that enables the software to keep the concerned employee updated regarding his/her expenses.

# 

# 3.EXECUTIVE SUMMARY

## 

## **1. Abstract of the project**

This project is aimed at developing an online expense claim system. This is a web based application that can be accessed over the web.

## **2. Functional components of the project**

In this project, there are four independent functions and the description about those functional components are as:

#### **1. Employee Code Module**: Following is a list of functionalities of the module. The operations defined in this module are add, delete, update and view the employee details.

* **Add Employee Details** as listed below and inject the values into database table if data are valid else display appropriate error messages

· Employee ID has to be auto generated.

· Employee Name is a combination of uppercase, lowercase alphabets and whitespaces (cannot be empty).

· Employee PAN is a combination of uppercase letters and digits (cannot be empty).

· Employee Designation is a combination of uppercase, lowercase alphabets and whitespaces (cannot be empty).

· Employee Domain is a combination of uppercase, lowercase alphabets and whitespaces (cannot be empty).

· Employee DOJ needs to undergo regular date validation (DD/MM/YYYY) (cannot be empty).

· Employee DOB needs to undergo regular date validation (DD/MM/YYYY) (cannot be empty).

· Employee Salary needs to undergo regular number validation (cannot be empty).

· Employee Mail ID needs to undergo regular mail ID validation (cannot be empty).

· Employee Password can be of any combination of characters (cannot be empty).

1. **Modify Employee details:**

· Based on existing Employee ID, modify all the fields of Employee Entity and the changes will be reflected in the database.

1. **Display all Employee details:**

· Display Employee details based on the entered Employee ID.

· If ID is not found, an error message will be displayed.

1. **Search details of an employee based on his/her Employee ID**

· The results will be filtered and displayed based on the Employee ID

#### **2. Project Code Module**: Following is a list of functionalities of the system. There is an admin who can add, delete, update and view the project details.

1. Add project details as listed below and inject the values into database table if data are valid else display appropriate error messages

· Project Code has to be auto generated.

· Project Description can be a combination of alphabets, digits and underscores (cannot be empty).

· Start date and End date need to undergo regular date validation (DD/MM/YYYY) (cannot be empty).

· Project end date has to be greater than project start date (cannot be empty).

2. Modify Project details:

· Based on existing project ID, display the following fields, Project Description, Start date, End date and Business Unit (validation should be taken care of).

3. Display all Project details:

· Display Project details based on the entered Project ID.

· If ID is not found, an error message will be displayed.

4. Search Project details based on Project ID:

· The results will be filtered and displayed based on the Project ID.

5. Delete Project based on Project ID:

· The project will be searched and deleted based on the Project ID.

#### **3. Expense Code Module**: Following is a list of functionalities of the system. There is an admin who can add, delete, update and view the expense details.

1. Add Expense Code details:

• Insert the values into database table if data is valid, else display appropriate error messages.

• Expense Code has to be auto generated.

• Expense Type must start with capital character, should only contain alphabet and it should contain minimum 3 and maximum 15 characters (cannot be empty).

• Expense Description can be a combination of alphabets, digits and underscores and it should contain a minimum of 15 and a maximum of 100 characters (cannot be empty).

2. Modify Expense details:

• Based on existing expense code, update the following fields Expense Code, Expense Type and Expense Description (validation should be taken care of).

3. Display Expense details:

• Based on existing expense code, display the following fields Expense Code, Expense Type and Expense Description.

4. Display All Expense details:

• Display the following fields Expense Code, Expense Type and Expense Description for all expense codes in the database.

5. Delete Expense details:

• Based on existing expense code, delete the following fields, Expense Code, Expense Type and Expense Description.

#### 

#### **4. Finance Team Module**: Following is a list of functionalities of the system. There is a user who can show,approve, and reject the claims details.

1. Show Claim details:

• Display the following fields claimID, employeeId, projectId, expenseId, amount, startDate, endDate status.

2. Approve/ Reject Claim:

• Based on existing expenses and authority the admin will have the power to approve or disapprove the claim based on the client's budget.

#### 

#### **5. Expense Claim Details Module**: The functionality of this module is to integrate the micro services, i.e. Employee Code Module, Project Code Module, Expense Code Module and Finance Team Module. The portal is created through which a user can claim, view, update and delete the expense. Following is a list of detailed functionalities under this module.

1. Claim an Expense:

· Enter the Employee ID. If valid, the corresponding details will be fetched from Employee Code Module and displayed.

· Else, the error message will be displayed and the user will not be able to proceed further.

· Select the Project Code from drop down list. The corresponding details will be fetched from Project Code Module and displayed.

· Select Expense Code from drop down list. The details will be fetched from Project Code Module and displayed.

· Enter some additional details, i.e. Start Date, Expense Amount, etc. These details, along with some required details of other micro services (Employee ID, Project Code and Expense Code) will be stored into the database.

· The Expense Code ID will be auto generated and displayed on the screen.

2. View Expense Claim Details by ID:

· Enter the Expense Code ID. If valid, the corresponding details will be displayed. Else if the ID is wrong, the error message will be displayed.

3. Update an Expense Claim:

· Enter the Expense Code ID. If valid, the details will be displayed and the user will be able to modify the details wherever required. Else if the ID is wrong, the error message will be displayed.

4. Delete an Expense Claim:

· Enter the Expense Code ID. If valid, all the corresponding details will be deleted from the database. Else if the Id is wrong, the error message will be displayed.

# 4. Tools Used

* Java 8(base programming language)
* Spring framework(v5.x for creating Web Services)
* MySQL (v5.5.17 database storage)
* Spring Data(v2.1.x)
* Spring Boot(v2.1.9)
* Maven (v3.6.1 DevOps Tool- Build and manage project)
* Angular (v6.0.0)
* NodeJS (v10.16.3 JavaScript Runtime Environment)
* Git 2.23.0 (DevOps VCS Tool)
* Amazon EC2 v4.9.3865
* Amazon RDS v2014-10-31
* Eclipse 2019-06 (4.12.0)
* VSCode: v1.38.1
* Postman: v7.9.0

# 

## 

# 5. Milestones

### 

|  |  |
| --- | --- |
| **Milestone** | **Tasks** |
| **1 - Analysis** | |
| 1.1 | Analysis and design stage, gather data and create system mockup |
| 1.2 | Architecture design |
| 1.3 | Design work plan (distribution of tasks to the team) |
| **2 - Development** | |
| 2.1 | Create database |
| 2.2 | Create GUI |
| 2.3 | Test cases, Validation, logging |
| **3 - Testing** | |
| 3.1 | Testing application |
| 3.2 | Finalise documentation |
| **4 - Deployment** | |
| 4.1 | Deployment to cloud |

# 5. UML

### Abstract Use Case Diagram

### 

### Sequence Diagram Admin

### 

### 

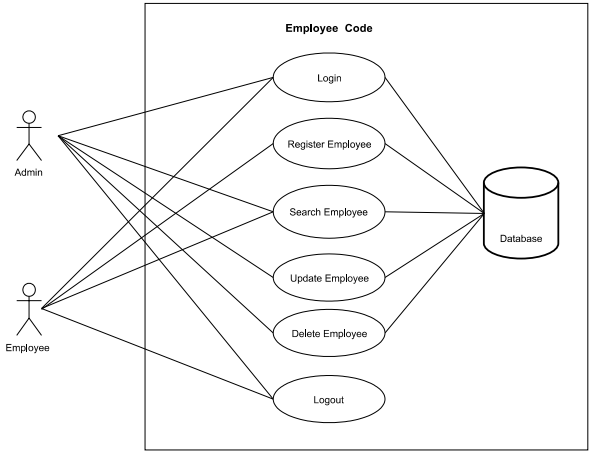
### 

### Sequence Diagram Employee

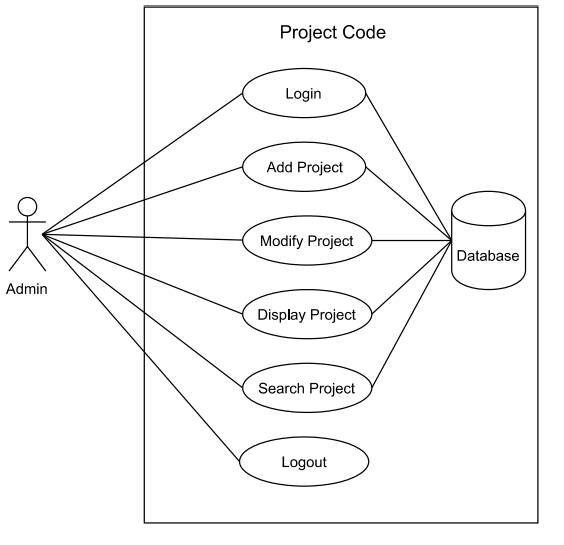
### Sequence Diagram Finance Team Module

### 

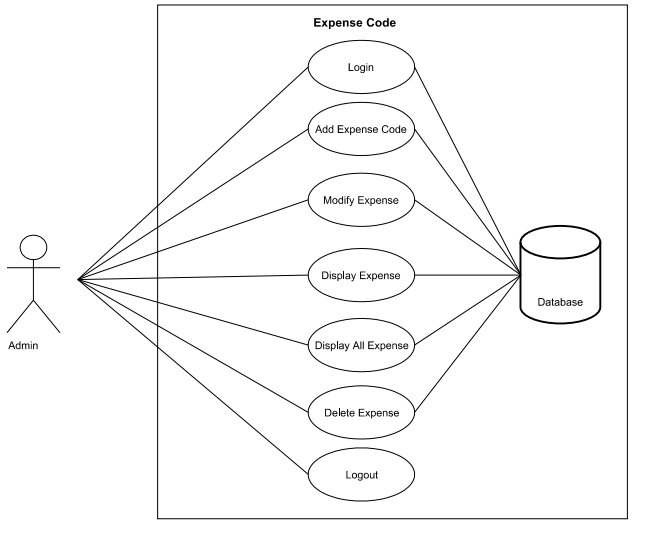
### Employee UCD



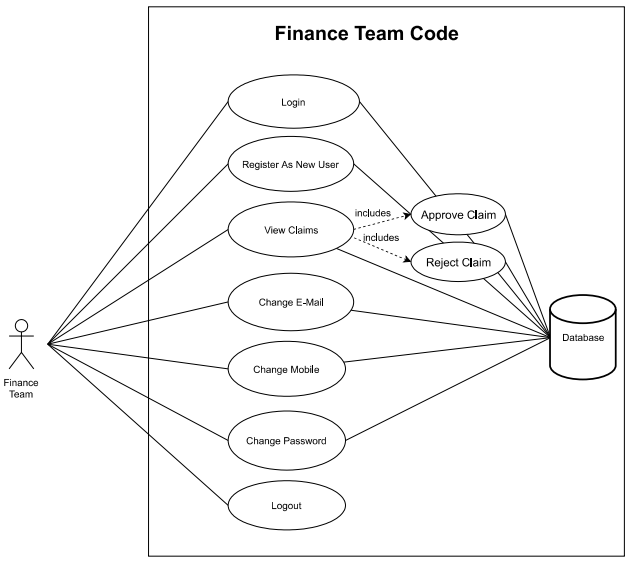
### Project UCD



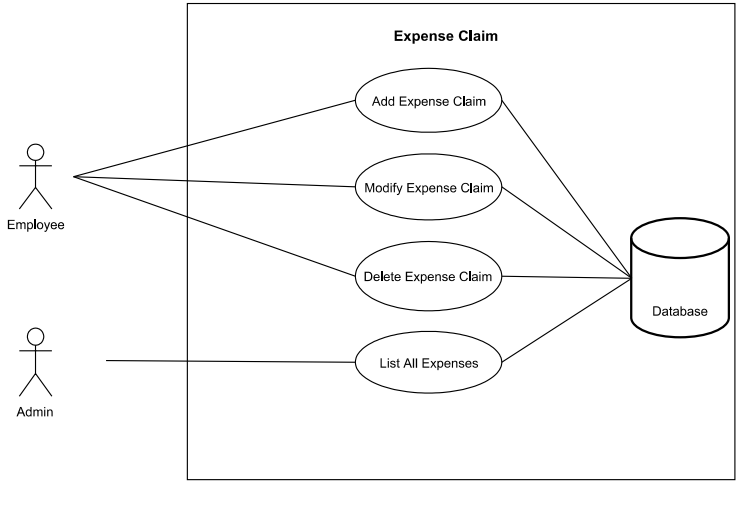
### Expense UCD



### Finance Team UCD



### Expense Claim UCD



## 

### Activity Diagram Admin

### 

### 

### Activity Diagram Employee

### 

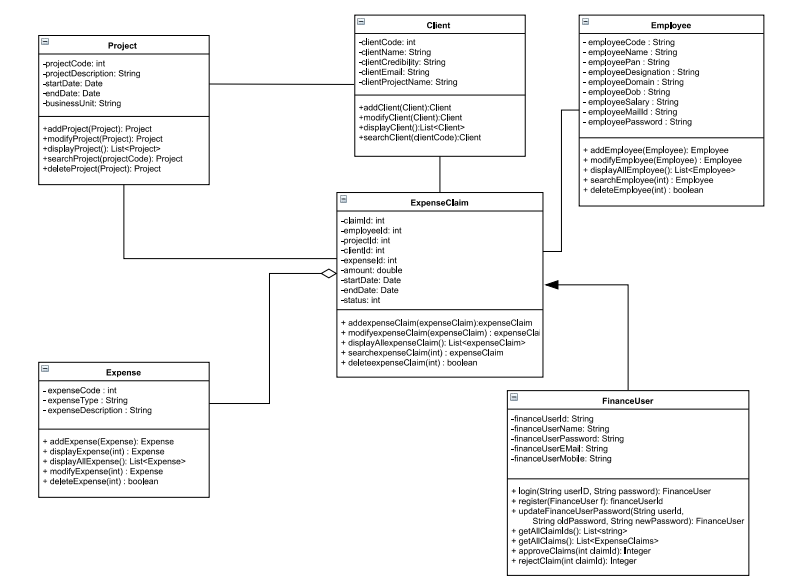
### 

### Activity Diagram Finance Team

### 

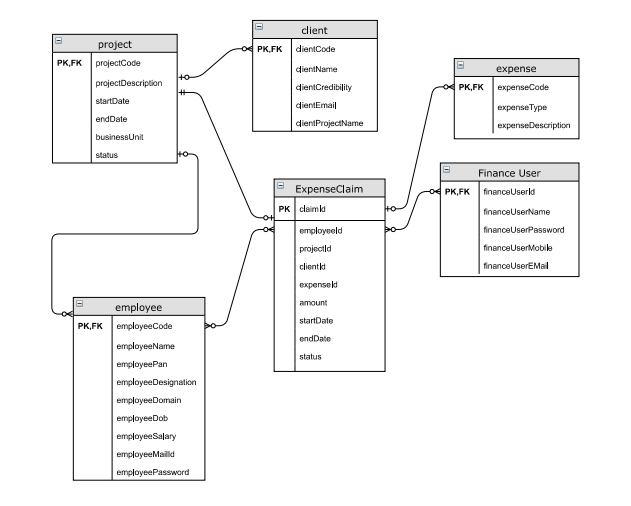
## 

## Class Diagram



## 

## Entity Relation Diagram



## Working

