**Employee RollOff**

**Low Level Design (LLD)**

Date:16/02/2023

Current Document Version: [*1.0*]

DOCUMENT APPROVAL

**Approvers of this document**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Department** | **Role** | **Signature** | **Date** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Document Change History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Document Version #** | **Author** | **Date** | **Description** |
| 1.0 | Dnyaneshwari Suryavanshi | 16-02-2023 | Document Creation |
| 1.1 | Dnyaneshwari Suryavanshi | 16-02-2023 | User and Developer Requirements Added |
|  |  |  |  |
|  |  |  |  |

Contents

[1.0 Document Purpose 5](#_Toc109286148)

[2.0 Intended Audience 5](#_Toc109286149)

[3.0 Project Background, Objective(s) 5](#_Toc109286150)

[3.1 Project Background 5](#_Toc109286151)

[3.2 Project Objective 5](#_Toc109286152)

[4.0 Design Pattern 5](#_Toc109286153)

[5.0 Solution Diagram 6](#_Toc109286154)

[6.0 Architecture Diagram 6](#_Toc109286156)

[7.0 Flow Diagram 7](#_Toc109286157)

[8.0 Use case Diagram 8](#_Toc109286158)

[9.0 Class Diagram 9](#_Toc109286159)

[10.0 E-R Diagram 10](#_Toc109286160)

[11.0 User Requirements 11](#_Toc109286161)

[11.1 Hardware 11](#_Toc109286162)

[11.2 Software 11](#_Toc109286168)

[12.0 Developer Requirements 11](#_Toc109286170)

[12.1 Hardware 11](#_Toc109286171)

[12.3 Technology 12](#_Toc109286181)

[13.0 Solution Steps 12](#_Toc109286185)

[14.0 Classes/function 14](#_Toc109286186)

[15.0 Data Model 15](#_Toc109286187)

[16.0 API Canvas 16](#_Toc109286188)

[17.0 HTTP Status Code 16](#_Toc109286189)

# Document Purpose

This document describes the solution architecture for Employee RollOff.

# Project Background, Objective(s)

## Project Background

Employee rolloff service is implemented to make the rolloff process easy by managing employee details inside one application and performing rolloff to avoid manual work.

## Project Objective

The main objective of Employee rolloff is to act as bridge between

PSP->Account->Superadmin by registering these users it will list the employee details and psp will be able to send rolloff request by checking the parameters of whether the employee is eligible for rolloff or not , if yes it will send this to superadmin to take the final call. i.e. for approval and rejection.

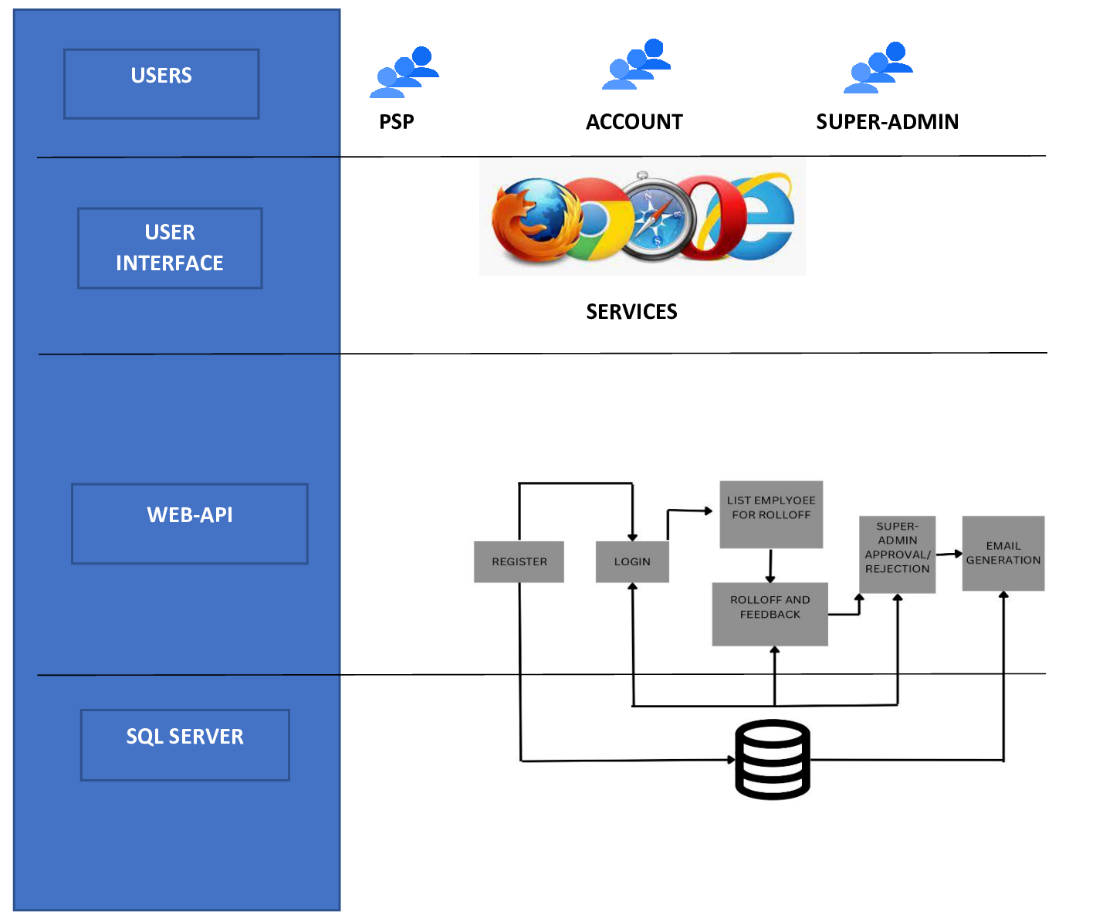
# Design Pattern

|  |  |  |
| --- | --- | --- |
| # | Name | Description |
| 1 | ANGULAR | Angular is used for the frontend and designing of the application |
| 2 | API | Api is used to process all the http  Requests. |
| 3 | SQL SERVER | SQL server is used to store data to the database |

# Solution Diagram

# 

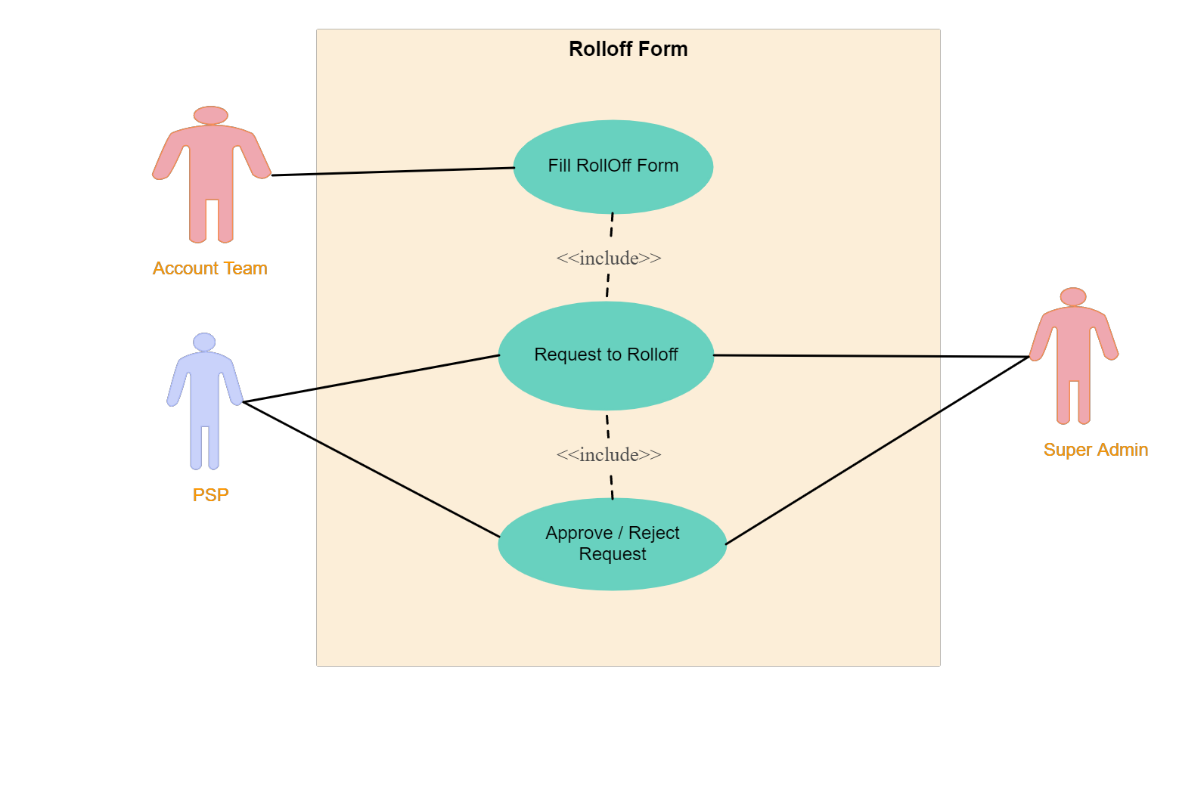
# Architecture Diagram



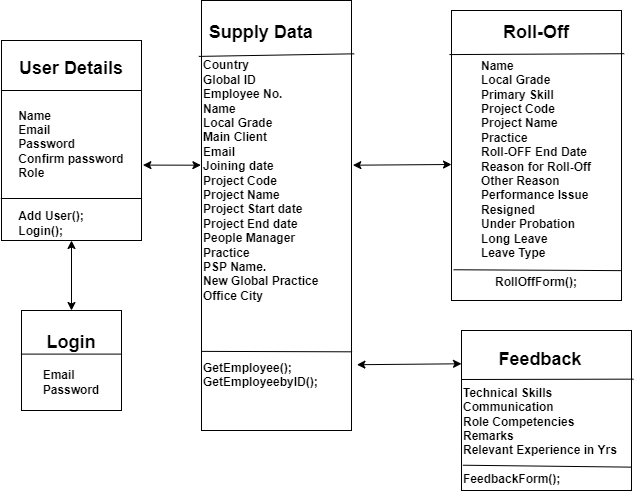
# Flow Diagram

# 

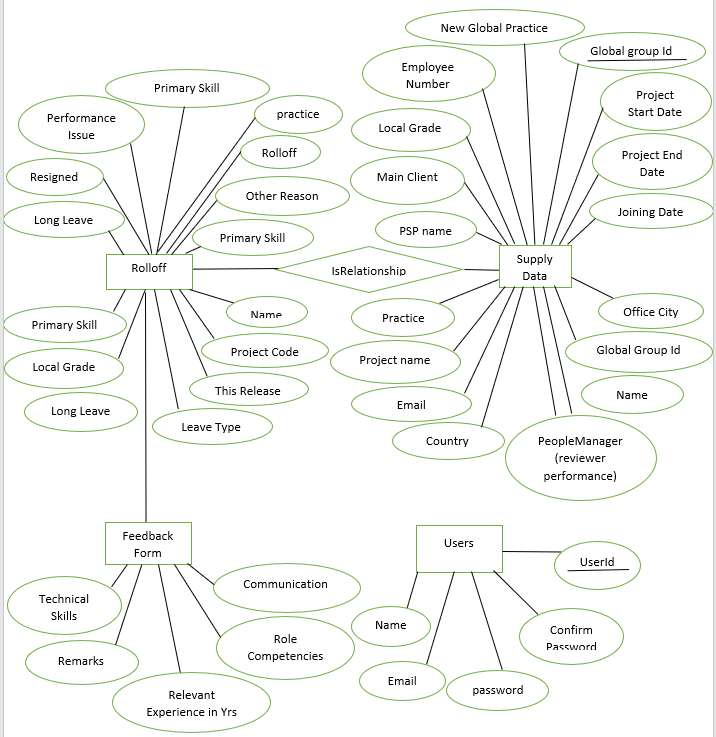
# 8.0 Use case Diagram



# 9.0 Class Diagram



# 10.0 E-R Diagram



# 11.0 User Requirements

# 11.1 Hardware

# • Processor: Minimum 1.8 GHz. Recommended 2GHz or more.

# • Ethernet connection (LAN) OR a wireless adapter (Wi-Fi)

# • Hard Drive: Minimum 100 GB; Recommended 500GB or more.

# • Memory (RAM): Minimum 4 GB; Recommended 8 GB or above.

# • OS: Windows.

# 11.2 Software

# • Any Latest Browsers.

# 12.0 Developer Requirements

# 12.1 Hardware

# • Processor: Minimum 1.8 GHz. Recommended 2GHz or more.

# • Ethernet connection (LAN) OR a wireless adapter (Wi-Fi)

# • Hard Drive: Minimum 100 GB; Recommended 500GB or more.

# • Memory (RAM): Minimum 4 GB; Recommended 8 GB or above.

# • OS: Windows.

# 12.2 Software

# Visual studio 2019.

* + - Visual Studio Code

# Sql server management studio (ssms).

# Node, Angular.

# 12.3 Technology

# ASP.NET Core WEB API for backend

# SQL for the database operations

# Angular for frontend.

# 13.0 Solution Steps

**13.1 User Registration**

1. The user will select their role as a PSP,Account,Admin.
2. The user will enter the name, Email, Password and select the role during registration.
3. The input validation will be done
4. If validation fails, then it will return the error code and error description.
5. If validation is successful, then the User details is stored in the database and user will logged in successfully.

**13.2 Employee Details**

1. The User will login using his username and password.
2. The User will search the Employee based on Email ,Global Group Id or Name.
3. After searching for employee user will be able to see employee details and option to Rolloff.

**13.3 RollOff Form**

1. After Clicking RollOff user will redirect to rolloff form where he/she can Fill the details such as Reason for rolloff, Start and End date, etc.
2. After filling the details user will submit the form and it will redirect to feedback form .
3. The request of that particular rolloff will be send to the account and superadmin for further procedure.

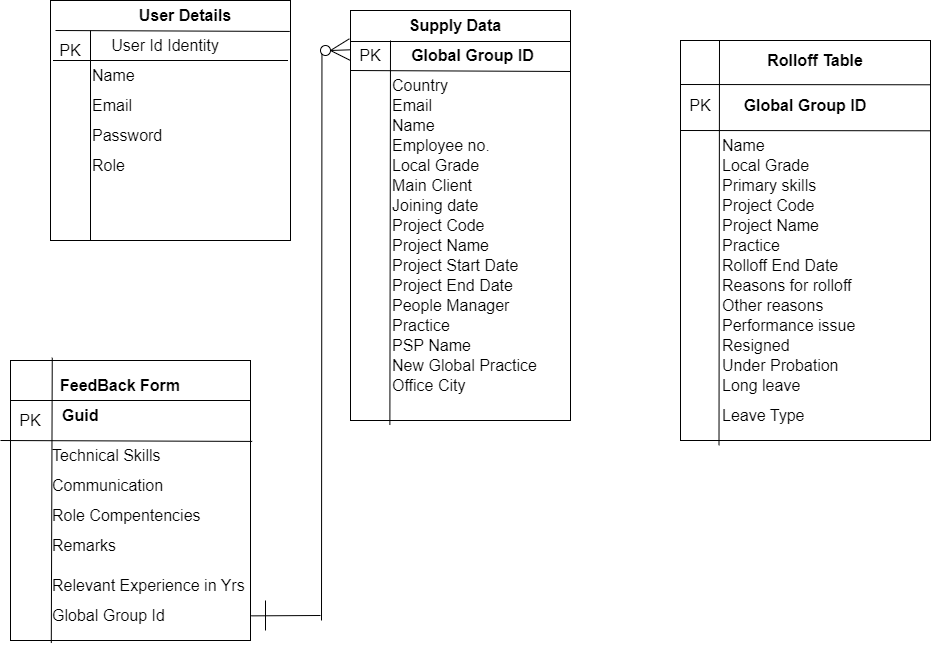
**13.4 SuperAdmin**

1. Then Superadmin will get the notification regarding the rolloff of resource and he/she can accept ,reject the request after verifying the reasons.
2. If the superadmin approve the request it will redirect to psp and account to to confirm the rolloff and the automatic mail will be send to that resource saying that you have been rolledoff.
3. If superadmin reject the request it will redirect to psp and account team for further process.

# 14.0 Classes/function

|  |  |  |
| --- | --- | --- |
| **#** | **Class** | **Description** |
| 1 | UserDetails.cs | Model holds the user schema details |
| 2 | RegisterController.cs | It contains the core business logic for the registration of Users. Which calls the UserDetails class to register the user in database |
| 3 | Login.cs | This class deals with the data accessibility for User registration. |
| 4 | Employee.cs | This class deals with data accessibility for employee list |
| 5 | EmployeeController.cs | It contains business logic for getting employee details |
| 6 | RolloffTable.cs | It contains the data accessibility for roll off details . |
| 7 | RolloffController.cs | It contains the business logic for inserting rolloff details of employee. |
| 8 | FormTable.cs | It contains feedback form. |
| 9 | FormController.cs | It contains the business logic for inserting feedback details of employee. |

# 15.0 Data Model



# 16.0 API Canvas

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| POC | Path | Verb | API Description | Role | Auth |
| Crop-Deal | /app-register | POST | User registration | PSP,Account,  Superadmin | True |
| Crop-Deal | /app-login | POST | User login | PSP,Account,  Superadmin | True |
| Crop-Deal | /app-employees | GET | To get the list of all employees. | Account | True |
| Crop-Deal | /app-employees/id | GET | To search employee details. | Account | True |
| Crop-Deal | /feedback-form | POST | To insert rolloff details and feedback form. | PSP | True |

# 17.0 HTTP Status Code

201 –User registered.

200 – Ok Success

[203 – Non-Authoritative Information](https://developer.mozilla.org/en-US/docs/Web/HTTP/Status/203)

400 – Inputs are invalid

403– Forbidden

404 – Data not found

502 – Bad gateway