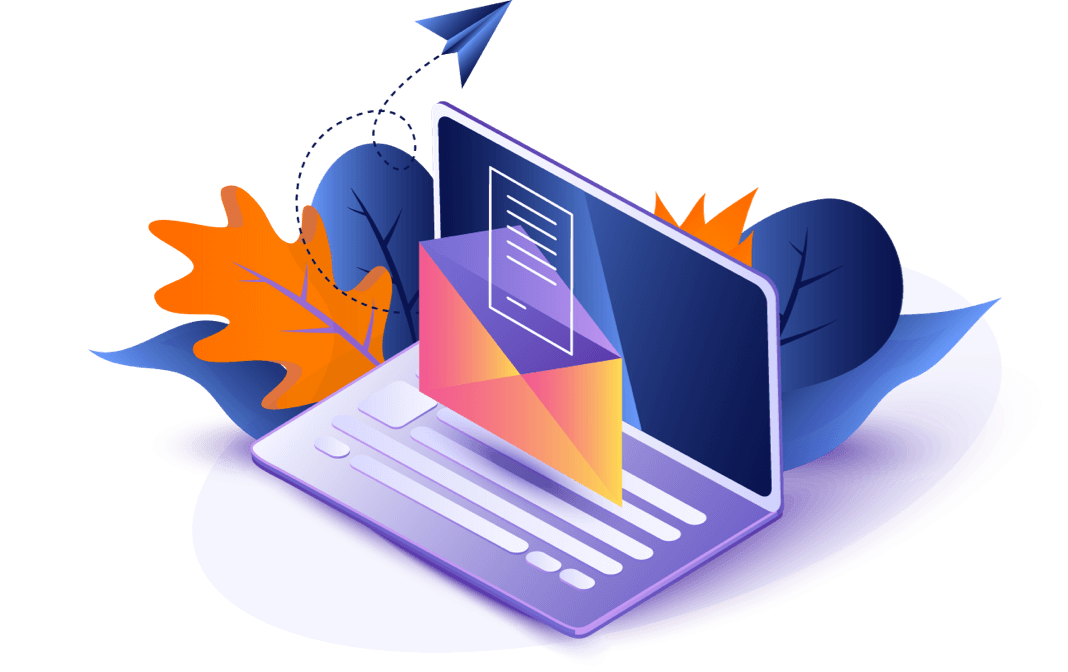
**ROLLOFF**

**API  
 Low Level Design (LLD)**



Date: 16/02/2023

DOCUMENT APPROVAL

**Approvers of this document**

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

**Document Creation History-**

|  |  |  |
| --- | --- | --- |
| **Member** | **Date** | **Description** |
| Donald Sahu | 16/02/2023 | Over All Flow and Guidance |
| Aniket Dhanait | 16/02/2023 | Over All Flow and Guidance |
| Shruti Agrawal | 16/02/2023 | Architecture Diagram and classes/functions Diagram |
| Apurva Gondkar | 16/02/2023 | Integrating All Together and Designing |
| Anita Nevase | 16/02/2023 | Flow Diagram and E-R Diagram |
| Gayatri Dhote | 16/02/2023 | Class Diagram and Solution Diagram |

Contents

[1.0 Document Purpose 5](#_Toc127521213)

[2.0 Project Objective(s) 5](#_Toc127521214)

[2.1 Project Objective 5](#_Toc127521215)

[3.0 Design Pattern 5](#_Toc127521216)

[4.0 Solution Diagram 6](#_Toc127521217)

[6](#_Toc127521218)

[5.0 Architecture Diagram 7](#_Toc127521219)

[6.0 Flow Diagram 8](#_Toc127521220)

[7.0 Class Diagram 8](#_Toc127521221)

[8.0 E-R Diagram 12](#_Toc127521222)

[9.0 User Requirements 13](#_Toc127521223)

[9.1 Hardware 13](#_Toc127521224)

[• Processor: Minimum 1.8 GHz. Recommended 2GHz or more. 13](#_Toc127521225)

[• Ethernet connection (LAN) OR a wireless adapter (Wi-Fi) 13](#_Toc127521226)

[• Hard Drive: Minimum 100 GB; Recommended 500GB or more. 13](#_Toc127521227)

[• Memory (RAM): Minimum 4 GB; Recommended 8 GB or above. 13](#_Toc127521228)

[• OS: Windows. 13](#_Toc127521229)

[9.2 Software 13](#_Toc127521230)

[• Any Latest Browsers. 13](#_Toc127521231)

[10.0 Developer Requirements 13](#_Toc127521232)

[10.1 Hardware 13](#_Toc127521233)

[• Processor: Minimum 1.8 GHz. Recommended 2GHz or more. 13](#_Toc127521234)

[• Ethernet connection (LAN) OR a wireless adapter (Wi-Fi) 13](#_Toc127521235)

[• Hard Drive: Minimum 100 GB; Recommended 500GB or more. 13](#_Toc127521236)

[• Memory (RAM): Minimum 4 GB; Recommended 8 GB or above. 13](#_Toc127521237)

[• OS: Windows. 13](#_Toc127521238)

[10.2 Software 13](#_Toc127521239)

[• Visual studio 2019. 13](#_Toc127521240)

[•SQL server management studio (SSMS). 13](#_Toc127521241)

[• Node, Angular. 13](#_Toc127521242)

[10.3 Technology 14](#_Toc127521243)

[• ASP.NET Core for backend. 14](#_Toc127521244)

[•SQL for the database operations. 14](#_Toc127521245)

[• Angular for frontend. 14](#_Toc127521246)

[11.0 Solution Steps 14](#_Toc127521247)

[12.0 Classes/function 16](#_Toc127521248)

# Document Purpose

This document describes the solution architecture for Roll OFF

# Project Objective(s)

## Project Objective

* In today’s scenario we are seeing that if a manager needs to request a Roll Off, he or she needs to communicate through emails manually.
* Because of this manual process, it has become very complex process and has become very difficult to manage the Roll Offs.
* In case Manager wants to Roll Off any Employee from a particular project then Manager needs to communicate with Admin and in rare cases even with the Super Admin through Emails.
* In some situation Admin or Super Admin may not be available, this can lead to delay in Roll Off or can hold on to the process.
* To overcome this situation this application will be helpful.

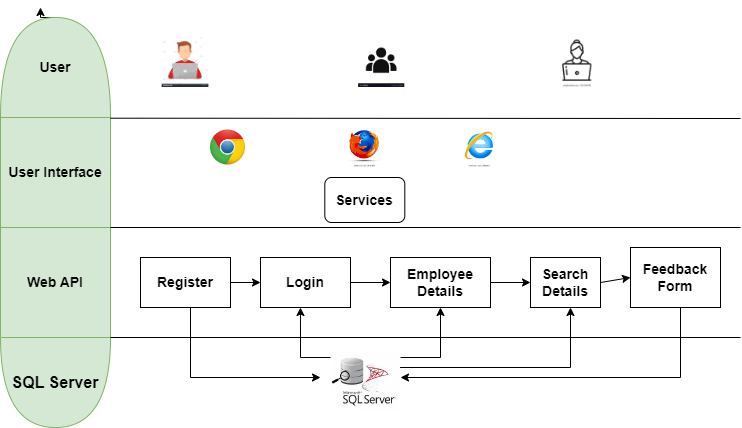
# Design Pattern

|  |  |  |
| --- | --- | --- |
| # | Name | Description |
| 1 | ANGULAR | Angular is used for designing the frontend of Application |
| 2 | Dot net Core | Dot net Core Web API is used to process all the http Requests. |
| 3 | SQL SERVER | SQL server is used to store data to the database |

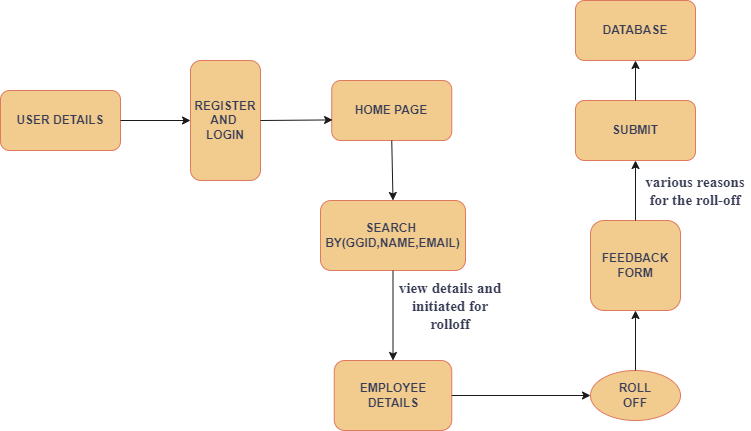
# Solution Diagram

# Graphical user interface Description automatically generated

# 5.0 Architecture Diagram



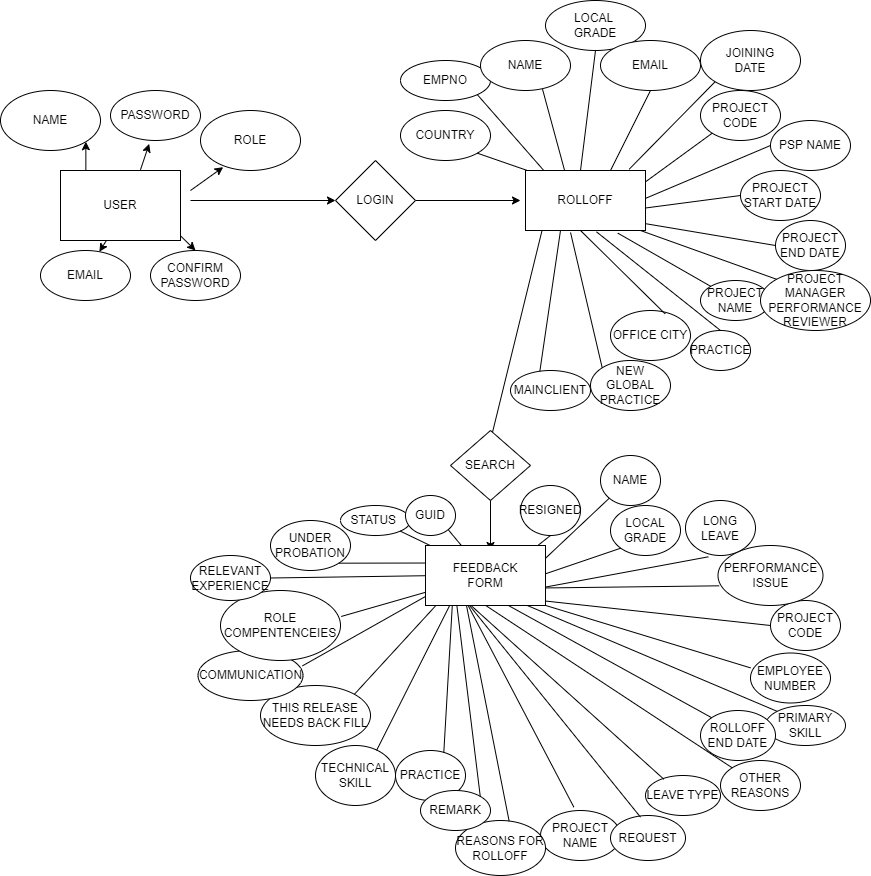
# 6.0 Flow Diagram



# 7.0 Class Diagram

# 

# 8.0 E-R Diagram



# 9.0 User Requirements

# 9.1 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.

# 9.2 Software

# • Any Latest Browsers.

# 10.0 Developer Requirements

# 10.1 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.

# 10.2 Software

# • Visual studio 2019.

# •SQL server management studio (SSMS).

# • Node, Angular.

# 10.3 Technology

# • ASP.NET Core for backend.

# •SQL for the database operations.

# • Angular for frontend.

# 11.0 Solution Steps

**11.1 User Registration**

1. The user will select their role as a Admin or Super admin.
2. The user will enter the Name, Role, Email and Password details during registration.
3. The input validation will be done
4. If validation fails, then it will return the error code and error description. with status code
5. If validation is successful, then the User details is stored in the database and success code is sent.

**11.2 User Login**

1. The User will login using his Email and password.
2. If the Email and Password matches the data stored in registration database, then and then the User is Allowed to Login
3. If the Email and Password does not match the data stored in registration database, then and then the User is not Allowed to Login.

**11.3 Roll OFF details**

1. The User logins with their Email and password.
2. The User sees all details of the Employees on the dashboard.
3. The User will search the specific Employee by Email, GGID and Name.
4. The User will select the employee and his/her details will be displayed briefly.

**11.4 Roll OFF Page**

1. After viewing the employee details the User will click the Roll off Button which will direct him to Roll Off page.

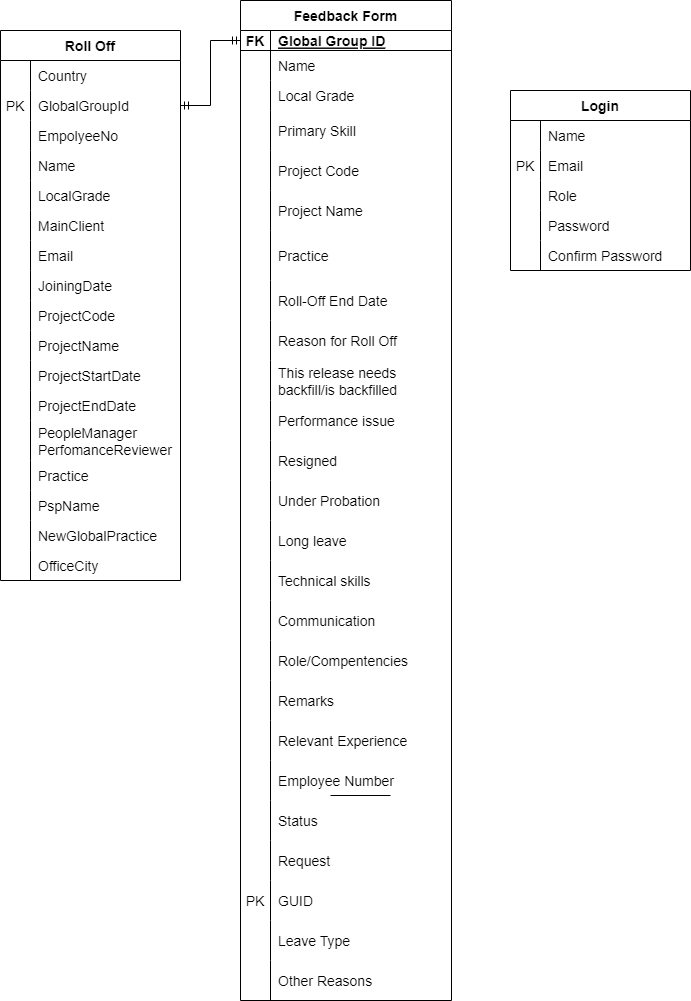
2. On Roll Off page user will give roll of details and feedback.

3. Feedback will be stored in database after submitting.

# 12.0 Classes/function

|  |  |  |
| --- | --- | --- |
| **#** | **Class** | **Description** |
| 1 | Login.cs | Model holds the user schema details |
| 2 | LoginController.cs | It contains the core business logic for the registration of User which calls the Login Repository class to create the User Details in database. |
| 3 | LoginRepository.cs | This class deals with the data accessibility for User registration |
| 4 | RollOff.cs | Model holds the user Roll Off schema details |
| 5 | RollOffController.cs | It contains the core business logic for the accessibility of Employee which calls the Roll Off Repository class to get the Employee Details in database. |
| 6 | RollOffRepository.cs | This class deals with the data accessibility for Employee Details. |
| 7 | FeedbackForm.cs | Model holds the user Feedback Form schema details |
| 8 | FormController.cs | It contains the core business logic for the accessibility of Employee which calls the Feedback Repository class to create the Feedback Form Details in database. |
| 9 | FromRepository.cs | This class deals with the data accessibility of Form Details. |

**Data Model:**

****

13.0 HTTP Status Code

* 201 – Roll Off registered
* 200 - Request succeeded
* 400 – Inputs are invalid
* 404 – Data not found
* 502 – Bad gateway