|  |
| --- |
|  |
|  |
| **HRMS- Assign manager to project Api analysis** |
|  |



**Process Revision History**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Ver. No** | **Date** | **Author** | **Approved by** | **Description** |
| 1 | 16-Sept-2020 | Sabiha Sultana |  | Writing technical documentation on Assign Manager to Project API analysis. |

# PURPOSE

The purpose of this documentation is to:

* List out the business rules of the HRMS existing application.
* List out the database and other dependencies of the existing application.
* Helps other developers in the team to understand the existing business rules.

# SCOPE

The scope of this document is to provide technical document of one of the project screens.

# ACRONYMS AND DEFINITIONS

|  |  |
| --- | --- |
| **Definition** | **Description** |
|  |  |

## Project

## Assign Manager To Project:

## Project/GetAllActiveProjects:

**Existing Implementation:**

* + - **Controller**: MasterDataController.cs
    - **Method:** GetProjectsList
    - This Method used for fetching active projects i.e., Created and Executed state projects.
    - Call usp\_GetProjects to get the required data.
    - usp\_GetProjects:
      * Purpose: This stored procedure is to fetch active project records.
      * Tables: Projects, ProjectManagers, Clients, PracticeArea, ProjectType, Employee, Status, Department and CategoryMaster
      * Decision: Move the stored procedure to LINQ.
      * Rules:
* System Admin can access Assign Manager To Project Menu to assign the program Manager, Reporting Manager and Lead of the project.
  + - * Return: active project related details.

**New Implementation:**

* + - Make sure all older Business Rules are applied.
    - convert the SP to LINQ
      * Purpose:
      * Database Tables:
        + **HRMSProjectDB:**

Projects

ProjectManagers

* + - * + **HRMSOrganization:**

Status

Department

CategoryMaster

Clients

PracticeArea

ProjectType

* + - * + **HRMSEmployee:**

Employee

* + - * Cross communication: Cross communication required for fetching data Status, User roles, Employee, and user API’s.
      * Return: The project related details.
    - API Details:
      * MicroService: Project
      * Controller: ProjectController.cs
      * Method: GetAllActiveProjects
      * Path: {{HostName}}/project/api/v1/Project/ GetAllActiveProjects

## Project/GetProjectsList:

**Existing Implementation:**

* + - **Controller**: ProjectController.cs
    - **Method:** GetEmployees
    - **Parameters**

String searchString

* + - This Method used for fetching Employees based on search string.
    - Call usp\_GetEmployees to get the required data based on search String.
    - : usp\_GetEmployees
      * Purpose: This stored procedure is to fetch Employee details based on search string.
      * Tables: Employee
      * Decision: Move the stored procedure to LINQ.
      * Return: Employee related details.

**New Implementation:**

* + - Make sure all older Business Rules are applied.
    - convert the SP to LINQ
      * Purpose:
      * Database Tables:
        + **HRMSEmployee:**

Employee

* + - * Cross communication: Cross communication not required.
      * Return: The employee related details.
    - API Details:
      * MicroService: Employee
      * Controller: EmployeeController.cs
      * Method: GetEmployees
      * Path: {{HostName}}/project/api/v1/Employee/GetEmployees

## Project/ GetReportingDetailsByProjectId:

**Existing Implementation:**

* + - **Controller**: ProjectController.cs
    - **Method:** GetReportingDetailsByProjectId
    - This Method used for fetching Program Manager, Reporting Manager and Lead details.
    - It is already a LINQ query.:
      * Purpose: This stored procedure is to fetch PM, RM and lead details.
      * Tables: Projects and ProjectManagers
      * Rules:
* System Admin can access Assign Manager To Project Menu to assign the program Manager, Reporting Manager and Lead of the project.
  + - * Return: PM, RM and lead details.

**New Implementation:**

* + - Make sure all older Business Rules are applied.
    - convert the SP to LINQ
      * Purpose:
      * Database Tables:
        + **HRMSProjectDB:**

Projects

ProjectManagers

* + - * Cross communication: Cross communication not required.
      * Return: The project related details.
    - API Details:
      * MicroService: Project
      * Controller: ProjectManagerController.cs
      * Method: GetReportingDetailsByProjectId
      * Path: {{HostName}}/project/api/v1/ProjectManager/ GetReportingDetailsByProjectId

## Project/SaveManagersToProject

* The above method is for Roles = "Program Manager, Department Head, HRA, HRM".

**Existing Implementation:**

* + - Input parameter for this method is as below

|  |  |
| --- | --- |
| **Parameter Name** | **Data Type** |
| projectData | ProjectData |

* + - **Controller**: ProjectController.cs
    - **Method:** SaveManagersToProject
    - This method used for assign managers to project.
    - Get Manager details from ProjectManager table and check whether the Manager record already exists or not.
    - If the Manager is different, then make the old record inactive and add new record to ProjectManager table.
    - Update AssociateAllocation and Employee table with new Managers.
* Purpose: Assigns PM, RM and Lead to project. Updates Allocation and Employee tables with new managers.
  + - * Tables: AssociateAllocation, Employees and ProjectManagers
      * Rules:
* Create entry for project manager record, update old record to inactive, update AssociateAllocation and Employees table.
  + - * Return: true/false

**New Implementation:**

* + - Make sure all older Business Rules are applied.
      * Database Tables:
        + **HRMSProjectDB:**

AssociateAllocation

ProjectManagers

* + - * + **HRMSOrganizationDB:**

Employee

* + - * Cross communication: Cross communication is required.
      * Return: true/false
    - API Details:
      * Project API.
* MicroService: Project
* Controller: ProjectManagerController.cs
* Method: SaveManagersToProject
* Path: {{HostName}}/project/api/v1/ProjectManager/ SaveManagersToProject
* Inputs:

|  |  |
| --- | --- |
| **Parameter Name** | **Data Type** |
| projectData | Project |

# REFERENCES