System configuration

**Low Level Design Document**

Contents

[1 Introduction 3](#_Toc155711723)

[ Purpose Of the Document 3](#_Toc155711724)

[ Scope 3](#_Toc155711725)

[2 Module Overview 3](#_Toc155711726)

[3 Architectural Diagram 4](#_Toc155711727)

[4 Data Structure 4](#_Toc155711728)

[5 Algorithms and Logic 6](#_Toc155711729)

[6 Interfaces 8](#_Toc155711730)

[7 Error Handling 8](#_Toc155711731)

[8 Performance Considerations 8](#_Toc155711732)

[9 Security Considerations 8](#_Toc155711733)

[10 Testing Strategy 8](#_Toc155711734)

[11 Dependencies 8](#_Toc155711735)

[12 Configuration Management 8](#_Toc155711736)

[13 Documentation References 8](#_Toc155711737)

[14 Glossary 8](#_Toc155711738)

[15 Revision History 9](#_Toc155711739)

# Introduction

* Purpose Of the Document

This document serves as a detailed guide for the development team to implement the low-level design of the System Configuration of AMF. It outlines the specific components, data flow, algorithms, and interfaces required to realize the functionality of managing System Configuation within the system.

* Scope

The scope of this LLD document is focused on providing a detailed design for the System Configuration module of AMF. It encompasses the following aspects:

* Module Overview
* Architectural Diagram
* Data Structure
* Algorithms and Logic
* Interfaces
* Error Handling
* Performance Considerations
* Security Considerations
* Testing Strategy
* Dependencies
* Configuration Management

# Module Overview

The System Configuration module defines:

* The different Entities that determine how the system is structured
  + - **Application:** An Application is a logical group of Modules that enables management of information of specific nature
    - **Module:** A Module is an independent component that can be easily integrated into an application of the system. A module specifies a set of functions which can be performed on a specific set of information.
    - **Action:** An action is the defined piece of task that can be performed in a module.
    - **Member:** A Member is an organization or an individual who manages its information in one or more Applications in the system.
    - **Action Group:** Action Groups are logical grouping of actions that standardizes and facilitates the assignment of multiple actions to members.
* The different Configurations that determine how the system behaves :
  + - **System Mode:** The System Mode determines whether the system is a
      * Stand Alone System
      * SaaS system.
    - **Multi Action Group:** The Multi Action Group behaviour determines whether a Member can be assigned multiple Action Groups or must be restricted to only one Action Group.

# Architectural Diagram

# Data Structure

This section describes the attributes of different Entities of system configuration and their relationships.

|  |  |  |  |
| --- | --- | --- | --- |
| **Entity Name** | Application | | |
| **Entity Type** | Table | | |
| **Purpose / Description** | Stores the list of Applications configured in the system | | |
| **Database Type** | PostgreSQL | | |
| **Database Name** | System Database | | |
| **References** |  | | |
| **Attribute Name** | **Purpose / Description** | **Data Type** | **Constraints** |
| Code | An Alpha Numeric value that is used internally to uniquely identify an application. | String | Primary Key |
| Name | A string that uniquely identifies an application. Used for public display. | string | Unique |
| Description | A brief description of the application | Text |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Entity Name** | Module | | |
| **Entity Type** | Table | | |
| **Purpose / Description** | Stores the list of Modules available in the System. | | |
| **Database Type** | PostgreSQL | | |
| **Database Name** | System Database | | |
| **References** |  | | |
| **Attribute Name** | **Purpose / Description** | **Data Type** | **Constraints** |
| Code | An unique No for a module | String | Primary Key |
| Name | A name for a module | string | Unique |
| Description | A small note about module | Text |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Entity Name** | Member | | |
| **Entity Type** | Table | | |
| **Purpose / Description** |  | | |
| **Database Type** | PostgreSQL | | |
| **Database Name** | System Database | | |
| **References** |  | | |
| **Attribute Name** | **Purpose / Description** | **Data Type** | **Constraints** |
| Code | An unique No for member | String | Primary Key |
| Name | A name for a member | string | Unique |
| Created On | When member is created | DateTime |  |
| Description | A small note about member | Text |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Entity Name** | ActionGroup | | |
| **Entity Type** | Table | | |
| **Purpose / Description** |  | | |
| **Database Type** | PostgreSQL | | |
| **Database Name** | System Database | | |
| **References** |  | | |
| **Attribute Name** | **Purpose / Description** | **Data Type** | **Constraints** |
| Code | An unique No for action group | String | Primary Key |
| Member Code | Member information | String | Foreign Key |
| Name | A name for an action group | string | Unique |
| Description | A small note about action group | Text |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Entity Name** | Action | | |
| **Entity Type** | Table | | |
| **Purpose / Description** |  | | |
| **Database Type** | PostgreSQL | | |
| **Database Name** | System Database | | |
| **References** |  | | |
| **Attribute Name** | **Purpose / Description** | **Data Type** | **Constraints** |
| Code | An unique No for an action | String | Primary Key |
| ApplicationCode |  | String | Foreign Key |
| ModuleCode |  | String | Foreign Key |
| ActionGroupCode |  | String | Foreign Key |
| Name | A name for an action | string | Unique |
| Description | A small note about action | Text |  |

# Algorithms and Logic

**Algorithms**: Create Application

Description:

This algorithm will store application information

Input:

Application Code, Name, Description

Output:

Boolean indication of Application stored or not.

**Algorithms**: Create Module

Description:

This algorithm will store module information

Input:

Module Code, Name, Description

Output:

Boolean indication of Module stored or not.

**Algorithms**: Create Member

Description:

This algorithm will store member information

Input:

Member Code, Name, Created On, Description

Output:

Boolean indication of member stored or not.

**Algorithms**: Create Action

Description:

This algorithm will store Action information

Input:

Action Code, Name, Module Code, Action Group, Description

Output:

Boolean indication of Action stored or not.

**Algorithms**: Create ActionGroup

Description:

This algorithm will store Action Group information

Input:

Action Group Code, Name, Member Code, Description

Output:

Boolean indication of Action Group stored or not.

# Interfaces

# Error Handling

# Performance Considerations

# Security Considerations

# Testing Strategy

# Dependencies

# Configuration Management

# Documentation References

# Glossary

|  |  |
| --- | --- |
| **Term** | **Definition** |
|  |  |
| **Action** | An action is the defined piece of task that can be performed in a module. |
| **Action Group** | Action Groups are logical grouping of actions that standardizes and facilitates the assignment of multiple actions to members. |
| **Application** | An Application is a logical group of Modules that enables management of information of specific nature |
| **Member** | A Member is an organization or an individual who manages its information in one or more Applications in the system. |
| **Module** | A Module is an independent component that can be easily integrated into an application of the system.  A module specifies a set of functions which can be performed on a specific set of information. |
|  |  |

# Revision History