Access Management Framework

**Functional Requirement Specification Document**

Contents

[1. Introduction 3](#_Toc160207161)

[2. Scope 3](#_Toc160207162)

[3. Definitions 3](#_Toc160207163)

[4. Functional Requirements 4](#_Toc160207164)

[4.1. System Settings 4](#_Toc160207165)

[4.2. Module and Action Management 4](#_Toc160207166)

[4.3. Application Management 6](#_Toc160207167)

[4.4. Application Package Management 6](#_Toc160207168)

[4.5. Application Package Group Management 6](#_Toc160207169)

[4.6. Workspace Type Management ( Module Assignment) 6](#_Toc160207170)

[4.7. Member Management 6](#_Toc160207171)

[4.8. Role Management 6](#_Toc160207172)

[4.9. Role Group Management 6](#_Toc160207173)

[4.10. Workspace Management (User – Role Assignment) 6](#_Toc160207174)

[4.11. User Management (Workspace – Role Assignment) 6](#_Toc160207175)

[5. System Requirements 6](#_Toc160207176)

[6. Non-Functional Requirements 6](#_Toc160207177)

[7. Constraints 7](#_Toc160207178)

[8. Assumptions 7](#_Toc160207179)

[9. Dependencies 7](#_Toc160207180)

[10. Risks 7](#_Toc160207181)

[11. Glossary 7](#_Toc160207182)

[12. Appendix 7](#_Toc160207183)

[13. Version Log 7](#_Toc160207184)

1. Introduction

Access management is a critical aspect of software systems, ensuring that the functionalities are accessed only by authorized users. This document outlines the functional requirements for developing a generic software framework for Access Management, which can be utilized as a base for various software projects and products. The Access Management Framework (AMF) aims to provide robust access control mechanisms for both standalone systems and Software as a Service (SaaS) models.

1. Scope

The scope of this document encompasses the following key components:

**Inclusions**

* Definition of Modules, Actions, Applications, Application Packages, and Application Package Groups.
* Management of Members and Users.
* Management of Roles, Role Groups,
* Management of Workspace Types and Workspaces.
* Assignment of roles to users within workspaces.

**Exclusions**

**Application-Specific Implementations:**

* The AMF will not dictate the specific implementation details within individual software applications, focusing solely on providing a standardized framework for access management.

1. Definitions

|  |  |
| --- | --- |
| Module | A logical grouping of functionalities representing a specific process within a software system. |
| Action | Individual functionalities within a module whose accessibility needs to be controlled. |
| Application | A logical group of modules and their associated actions. |
| Application Package | A subset of module actions within an application. |
| Application Package Group | A collection of application packages bundled together. |
| Member | Subscriber of the system, which can be an organization or entity. |
| User | Entity accessing the system, associated with a member. |
| Role | A set of logically related actions assigned to users for access management. |
| Role Group | A bundle of roles assigned to users for convenience in workspace assignment. |
| Workspace Type | Structural division of the working area of an organization. |
| Workspace | An instance of a workspace type where users are assigned roles. |
| Standalone System | A system operating independently with a single member. |
| SaaS System | A system operating with multiple members subscribing to applications. |

1. Functional Requirements

Functional requirements for this system specify the specific functionalities and behaviours that must exhibit to full-fill its purpose.

* 1. System Settings

The system can be configured in two different modes:

* **SaaS (Software as a Service) System Mode:**

In SaaS system mode, the system can have any number of Members

* **Stand Alone System Mode:**

In Stand Alone System Mode, the system has one and only one Members

* 1. Module and Action Management

The Module and Action Management Utility is an integral part of the AMF. This utility is responsible for managing modules within the system, ensuring their proper integration, and maintaining an up-to-date database.

A **module** has the following attributes:

|  |  |
| --- | --- |
| Unique Numeric Identity | Each module is assigned a unique numeric identifier for identification and reference purposes. |
| Unique Name | A distinct name is allocated to each module for easy recognition and reference internally. |
| Display Name | Each module is assigned a unique name, intended for display within the user interface, ensuring clear identification and ease of use for the end-user. |
| Deprecation Flag | This flag indicates whether a module is deprecated or not. Deprecated modules can no longer be supported. |
| Marked for Deprecation Flag | This flag signifies whether a module is marked for deprecation in future updates. Modules marked for deprecation are still supported but will be phased out in subsequent releases. |
| Description | A brief description providing an overview of the module's functionality and purpose. |
| New Module Identity | Unique Identity of the replacement Module (if any), in case the module is marked for deprecation. |
| IsActive | The "IsActive" flag is utilized to temporarily deactivate a module, rendering it unavailable for use within the system until reactivated. |

A **Action** has the following attributes:

|  |  |
| --- | --- |
| Unique Numeric Identity | Each action is assigned a unique numeric identifier across modules for identification and reference purposes. |
| Unique Name | A distinct name is allocated to each action in a module for easy recognition and reference internally. |
| Display Name | Each action is assigned a unique name, intended for display within the user interface, ensuring clear identification and ease of use for the end-user. |
| Description | A brief description providing an overview of the action's functionality and purpose. |
| IsActive | The "IsActive" flag is utilized to temporarily deactivate an action, rendering it unavailable for use within the system until reactivated. |

The Module and Action Management Utility facilitates the following functionalities:

* **Addition of New Modules and Actions**

It allows for the incorporation of new modules and actions into the system by assigning them a unique numeric identity and a distinct name.

* **Updating Module and Action Information**

The utility enables the modification of module attributes such as deprecation status, marked for deprecation status, and description.

The utility enables the modification of action attributes such as name and description

* **Removal of Deprecated Modules and actions**

Deprecated modules can be flagged for removal from the system, ensuring optimal maintenance and efficiency.

* **Database Synchronization**

The utility ensures that the database containing module and action information remains synchronized with the latest changes, reflecting any updates or introductions of new modules and actions.

* 1. Application Management
  2. Application Package Management
  3. Application Package Group Management
  4. Workspace Type Management ( Module Assignment)
  5. Member Management
  6. Role Management
  7. Role Group Management
  8. Workspace Management (User – Role Assignment)
  9. User Management (Workspace – Role Assignment)

1. System Requirements

* The system shall be web-based and accessible via modern web browsers.
* It shall be built using scalable and maintainable technologies to accommodate future enhancements.
* The system shall ensure data security and privacy compliance measures.
* It shall provide user-friendly interfaces for ease of navigation and management.

1. Non-Functional Requirements
2. Constraints
3. Assumptions
4. Dependencies
5. Risks
6. Glossary
7. Appendix
8. Version Log

.