Enterprise User Management Design

Version: 1.4

Dated: 21st June, 2016

Confidentiality and Copyright Statement

Information contained in this document is classified Altiux Confidential Proprietary. No person outside Altiux shall have access to the information contained in this document unless business needs dictate otherwise. It is the responsibility of the person knowing the information contained in this document to ensure confidentiality of information contained in it and preventing unauthorized access to this document at all times.

Contents

[1. Introduction 11](#_Toc454292164)

[1.1. Purpose 11](#_Toc454292165)

[1.2. Scope 11](#_Toc454292166)

[1.3. Acronyms 11](#_Toc454292167)

[1.4. References 11](#_Toc454292168)

[2. Block diagram for User Management 12](#_Toc454292169)

[2.1. High level Design for Enterprise User Management 12](#_Toc454292170)

[2.2. Enterprise User Login Management 12](#_Toc454292171)

[2.3. Enterprise Schema Diagram 13](#_Toc454292172)

[2.4. Multitenant Enterprise User Management Architecture 15](#_Toc454292173)

[3. Enterprise User Management Overview 15](#_Toc454292174)

[3.1. Enterprise User Management Mappings 16](#_Toc454292175)

[3.2. Operations 16](#_Toc454292176)

[3.2.1. Sites 16](#_Toc454292177)

[3.2.2. Site Status 16](#_Toc454292178)

[3.2.3. Authentication Source 16](#_Toc454292179)

[3.2.4. Site Advanced Users 17](#_Toc454292180)

[3.2.5. Site Scheduled 17](#_Toc454292181)

[3.2.6. Site Users 17](#_Toc454292182)

[3.2.7. Zones 17](#_Toc454292183)

[3.2.8. Places 17](#_Toc454292184)

[3.2.9. PINs 17](#_Toc454292185)

[3.2.10. Schedules 17](#_Toc454292186)

[3.2.11. Override 17](#_Toc454292187)

[3.2.12. Super Admins 17](#_Toc454292188)

[3.2.13. Site Super Admins 18](#_Toc454292189)

[3.3. Roles 18](#_Toc454292190)

[3.3.1. Enterprise Super Admin 18](#_Toc454292191)

[3.3.2. Enterprise Field Support Admin 18](#_Toc454292192)

[3.3.3. Enterprise Site Super Admin 18](#_Toc454292193)

[3.3.4. Enterprise Site Advanced User 18](#_Toc454292194)

[3.3.5. Enterprise Site Scheduled User 18](#_Toc454292195)

[3.3.6. Enterprise Site User 19](#_Toc454292196)

[3.3.7. Enterprise Manager 19](#_Toc454292197)

[3.3.8. Enterprise Assistant Manager 19](#_Toc454292198)

[3.3.9. Law Enforcement Officer(LEO) 19](#_Toc454292199)

[3.4. Roles and operations Diagram 19](#_Toc454292200)

[4. Configuration Settings 20](#_Toc454292201)

[4.1. Web.xml Settings 20](#_Toc454292202)

[4.2. Servlet-context settings 21](#_Toc454292203)

[4.3. User-beans 22](#_Toc454292204)

[4.4. User Properties File 25](#_Toc454292205)

[4.5. Views 25](#_Toc454292206)

[4.5.1. documentation.jsp 25](#_Toc454292207)

[4.5.2. home.jsp 27](#_Toc454292208)

[4.6. Resources 28](#_Toc454292209)

[4.6.1. Log4J Configuration 28](#_Toc454292210)

[5. Sequence Diagram 29](#_Toc454292211)

[5.1. User Request Management 29](#_Toc454292212)

[5.2. User Management 30](#_Toc454292213)

[5.3. Login Management 30](#_Toc454292214)

[5.4. Enterprise System Management 31](#_Toc454292215)

[6. Class Diagrams 32](#_Toc454292216)

[6.1. Controllers: 32](#_Toc454292217)

[6.1.1. EnterpriseSystemUserRequestController 32](#_Toc454292218)

[6.1.2. EnterpriseSystemLoginController 32](#_Toc454292219)

[6.1.3. EnterpriseSystemUserManagementController 32](#_Toc454292220)

[6.1.4. EnterpriseSiteLoginController 33](#_Toc454292221)

[6.1.5. EnterpriseSiteUserRequestController 33](#_Toc454292222)

[6.1.6. EnterpriseSiteUserMangementController 34](#_Toc454292223)

[6.2. DAO Interfaces: 34](#_Toc454292224)

[6.2.1. Enterprise System Login DAO 34](#_Toc454292225)

[6.2.2. Enterprise System User Request DAO 34](#_Toc454292226)

[6.2.3. Enterprise System User Management DAO 35](#_Toc454292227)

[6.2.4. Enterprise Site Login DAO 36](#_Toc454292228)

[6.2.5. Enterprise Site User Request DAO 36](#_Toc454292229)

[6.2.6. Enterprise Site User Management DAO 36](#_Toc454292230)

[6.3. DAO Implementations: 37](#_Toc454292231)

[6.3.1. Enterprise System Login DAO Impl 37](#_Toc454292232)

[6.3.2. Enterprise System User Management DAO Impl 38](#_Toc454292233)

[6.3.3. Enterprise System User Request DAO Impl 39](#_Toc454292234)

[6.3.4. Enterprise Site Login DAO Impl 39](#_Toc454292235)

[6.3.5. Enterprise Site User Management DAO Impl 40](#_Toc454292236)

[6.3.6. Enterprise Site User Request DAO Impl 40](#_Toc454292237)

[7. User Management Source Configuration: 41](#_Toc454292238)

[7.1. Java Resources: 41](#_Toc454292239)

[7.2. com.altiux.commons.errors 41](#_Toc454292240)

[7.2.1. Class: Error Response: 41](#_Toc454292241)

[7.2.1.1. Constructors 41](#_Toc454292242)

[7.2.1.2. Methods 41](#_Toc454292243)

[7.3. com.altiux.commons.exceptions 41](#_Toc454292244)

[7.3.1. AccessTokenSpecifiedException: 41](#_Toc454292245)

[7.3.1.1. Constructors 41](#_Toc454292246)

[7.3.2. Class: AccountNameInUseException: 42](#_Toc454292247)

[7.3.2.1. Constructors 42](#_Toc454292248)

[7.3.3. Class: DataInconsistentException: 42](#_Toc454292249)

[7.3.3.1. Constructors 42](#_Toc454292250)

[7.3.4. Class: EnterpriseSiteNotFoundException: 42](#_Toc454292251)

[7.3.4.1. Constructors 42](#_Toc454292252)

[7.3.5. Class: EnterpriseSiteNotActiveException: 43](#_Toc454292253)

[7.3.5.1. Constructors 43](#_Toc454292254)

[7.3.6. Class: EntityIDException: 43](#_Toc454292255)

[7.3.6.1. Constructors 43](#_Toc454292256)

[7.3.7. Class: EntityNameException: 43](#_Toc454292257)

[7.3.7.1. Constructors 43](#_Toc454292258)

[7.3.8. Class: InstanceIdNotProvidedException: 44](#_Toc454292259)

[7.3.8.1. Constructors 44](#_Toc454292260)

[7.3.9. Class: InvalidInputException: 44](#_Toc454292261)

[7.3.9.1. Constructors 44](#_Toc454292262)

[7.3.10. Class: LoginException: 44](#_Toc454292263)

[7.3.10.1. Constructors 44](#_Toc454292264)

[7.3.11. Class: NoDataFoundException: 45](#_Toc454292265)

[7.3.11.1. Constructors 45](#_Toc454292266)

[7.3.12. Class: PasswordSpecifiedException: 45](#_Toc454292267)

[7.3.12.1. Constructors 45](#_Toc454292268)

[7.3.13. Class: RequestBodyImproperException: 45](#_Toc454292269)

[7.3.13.1. Constructors 45](#_Toc454292270)

[7.3.14. Class: ServerInstanceNotFoundException: 46](#_Toc454292271)

[7.3.14.1. Constructors 46](#_Toc454292272)

[7.3.15. Class: UserAlreadyAvailableException: 46](#_Toc454292273)

[7.3.15.1. Constructors 46](#_Toc454292274)

[7.3.16. Class: UserNotFoundException: 46](#_Toc454292275)

[7.3.16.1. Constructors 46](#_Toc454292276)

[7.4. com.altiux.eum.esite.controller 47](#_Toc454292277)

[7.4.1. EnterpriseSiteLoginController 47](#_Toc454292278)

[7.4.1.1. Methods 47](#_Toc454292279)

[7.4.2. EnterpriseSiteUserManagementController 47](#_Toc454292280)

[7.4.2.1. Methods 47](#_Toc454292281)

[7.4.3. EnterpriseSiteUserRequestController 47](#_Toc454292282)

[7.4.3.1. Methods 47](#_Toc454292283)

[7.5. com.altiux.eum.esystem.controller 48](#_Toc454292284)

[7.5.1. EnterprirseSystemUserRequestController 48](#_Toc454292285)

[7.5.1.1. Methods 48](#_Toc454292286)

[7.5.2. EnterpriseSystemController 48](#_Toc454292287)

[7.5.2.1. Methods 48](#_Toc454292288)

[7.5.3. EnterpriseSystemLoginController 48](#_Toc454292289)

[7.5.3.1. Methods 48](#_Toc454292290)

[7.5.4. EnterpriseSystemUserManagementController 49](#_Toc454292291)

[7.5.4.1. Methods 49](#_Toc454292292)

[7.6. com.altiux.eum.esite.dao 49](#_Toc454292293)

[7.6.1. LoginDAO 49](#_Toc454292294)

[7.6.1.1. Methods 50](#_Toc454292295)

[7.6.2. UserManagementDAO 50](#_Toc454292296)

[7.6.2.1. Methods 50](#_Toc454292297)

[7.6.3. UserRequestDAO 51](#_Toc454292298)

[7.6.3.1. Methods 51](#_Toc454292299)

[7.7. com.altiux.eum.esystem.dao 51](#_Toc454292300)

[7.7.1. ConfigDAO 51](#_Toc454292301)

[7.7.1.1. Methods 51](#_Toc454292302)

[7.7.2. EnterpriseSystemDAO 51](#_Toc454292303)

[7.7.2.1. Methods 51](#_Toc454292304)

[7.7.3. EnterpriseSystemLoginDAO 52](#_Toc454292305)

[7.7.3.1. Methods 52](#_Toc454292306)

[7.7.4. UserManagementDAO 52](#_Toc454292307)

[7.7.4.1. Methods 52](#_Toc454292308)

[7.7.5. UserRequestDAO 54](#_Toc454292309)

[7.7.5.1. Methods 54](#_Toc454292310)

[7.8. com.altiux.eum.esite.dao.impl 54](#_Toc454292311)

[7.8.1. LoginDAOImpl 54](#_Toc454292312)

[7.8.1.1. Methods 54](#_Toc454292313)

[7.8.2. UserManagementDaoImpl 54](#_Toc454292314)

[7.8.2.1. Methods 54](#_Toc454292315)

[7.8.3. UserRequestDAOImpl 55](#_Toc454292316)

[7.8.3.1. Methods 55](#_Toc454292317)

[7.9. com.altiux.eum.esystem.dao.impl 56](#_Toc454292318)

[7.9.1. ConfigDAOImpl 56](#_Toc454292319)

[7.9.1.1. Methods 56](#_Toc454292320)

[7.9.2. EnterpriseSystemDAOImpl 56](#_Toc454292321)

[7.9.2.1. Methods 56](#_Toc454292322)

[7.9.3. EnterpriseSystemLoginDAOImpl 57](#_Toc454292323)

[7.9.3.1. Methods 57](#_Toc454292324)

[7.9.4. UserManagementDAOImpl 57](#_Toc454292325)

[7.9.4.1. Methods 57](#_Toc454292326)

[7.9.5. UserRegexUtil 58](#_Toc454292327)

[7.9.5.1. Methods 58](#_Toc454292328)

[7.9.6. UserRequestDAOImpl 58](#_Toc454292329)

[7.9.6.1. Methods 59](#_Toc454292330)

[7.10. com.altiux.rest.controllers 59](#_Toc454292331)

[7.10.1. ExampleDocumentationController 59](#_Toc454292332)

[7.10.1.1. Constructors 59](#_Toc454292333)

[7.10.1.2. Methods 59](#_Toc454292334)

[7.10.2. HomePageController 59](#_Toc454292335)

[7.10.2.1. Methods 59](#_Toc454292336)

[7.11. com.altiux.logger 60](#_Toc454292337)

[7.11.1. App\_logger 60](#_Toc454292338)

[7.11.1.1. Constructors 60](#_Toc454292339)

[7.11.1.2. Methods 60](#_Toc454292340)

[7.11.2. EModuleName 60](#_Toc454292341)

[7.11.2.1. Constants 60](#_Toc454292342)

[7.11.3. LoggerFactory 61](#_Toc454292343)

[7.11.3.1. Methods 61](#_Toc454292344)

[7.12. com.altiux.eum.utils 61](#_Toc454292345)

[7.12.1. AltiuxSessionFactory 61](#_Toc454292346)

[7.12.1.1. Methods 61](#_Toc454292347)

[7.12.2. EnterpriseSiteConstants 61](#_Toc454292348)

[7.12.2.1. Constants 61](#_Toc454292349)

[7.12.3. EnterpriseSystemInputValidator 62](#_Toc454292350)

[7.12.3.1. Methods 62](#_Toc454292351)

[7.12.4. EnterpriseSystemSessionFactory 62](#_Toc454292352)

[7.12.4.1. Methods 62](#_Toc454292353)

[7.13. com.altiux.eum.util 62](#_Toc454292354)

[7.13.1. EnterpriseSiteUtils 62](#_Toc454292355)

[7.13.1.1. Methods 62](#_Toc454292356)

[8. Database Schema Design 63](#_Toc454292357)

[8.1. Enterprise System 64](#_Toc454292358)

[8.1.1. Table Description 64](#_Toc454292359)

[8.1.2. Table Information 65](#_Toc454292360)

[8.2. Enterprise Site 65](#_Toc454292361)

[8.2.1. Table Description 65](#_Toc454292362)

[8.2.2. Table Information 66](#_Toc454292363)

[8.3. User 67](#_Toc454292364)

[8.3.1. Table Description 67](#_Toc454292365)

[8.3.2. Table Information 68](#_Toc454292366)

[8.4. Enterprise Site User Mapping 68](#_Toc454292367)

[8.4.1. Table Description 68](#_Toc454292368)

[8.4.2. Table Information 68](#_Toc454292369)

[8.5. Enterprise System User 69](#_Toc454292370)

[8.5.1. Table Description 69](#_Toc454292371)

[8.5.2. Table Information 69](#_Toc454292372)

[8.6. User Request 69](#_Toc454292373)

[8.6.1. Table Description 69](#_Toc454292374)

[8.6.2. Table Information 69](#_Toc454292375)

[8.7. Devices 70](#_Toc454292376)

[8.7.1. Table Description 70](#_Toc454292377)

[8.7.2. Table Information 70](#_Toc454292378)

[8.8. Gate Ways 71](#_Toc454292379)

[8.8.1. Table Description 71](#_Toc454292380)

[8.8.2. Table Information 71](#_Toc454292381)

[8.9. Roles 72](#_Toc454292382)

[8.9.1. Table Description 72](#_Toc454292383)

[8.9.2. Table Information 72](#_Toc454292384)

[8.10. Operations 72](#_Toc454292385)

[8.10.1. Table Description 72](#_Toc454292386)

[8.10.2. Table Information 72](#_Toc454292387)

[8.11. Permitted Operations 73](#_Toc454292388)

[8.11.1. Table Description 73](#_Toc454292389)

[8.11.2. Table Information 73](#_Toc454292390)

[8.12. Role Operations 73](#_Toc454292391)

[8.12.1. Table Description 73](#_Toc454292392)

[8.12.2. Table Information 73](#_Toc454292393)

[8.13. Violation Configuration 73](#_Toc454292394)

[8.13.1. Table Description 73](#_Toc454292395)

[8.13.2. Table Information 74](#_Toc454292396)

[8.14. Enterprise Site Phone Numbers 74](#_Toc454292397)

[8.14.1. Table Description 74](#_Toc454292398)

[8.14.2. Table Information 74](#_Toc454292399)

[8.15. Enterprise Site Emails 75](#_Toc454292400)

[8.15.1. Table Description 75](#_Toc454292401)

[8.15.2. Table Information 75](#_Toc454292402)

[8.16. Enterprise Site Photos 75](#_Toc454292403)

[8.16.1. Table Description 75](#_Toc454292404)

[8.16.2. Table Information 76](#_Toc454292405)

[8.17. User photos 76](#_Toc454292406)

[8.17.1. Table Description 76](#_Toc454292407)

[8.17.2. Table Information 76](#_Toc454292408)

[8.18. User Emails 76](#_Toc454292409)

[8.18.1. Table Description 76](#_Toc454292410)

[8.18.2. Table Information 77](#_Toc454292411)

[8.19. User Phone Numbers 77](#_Toc454292412)

[8.19.1. Table Description 77](#_Toc454292413)

[8.19.2. Table Information 77](#_Toc454292414)

Revision History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SI.NO** | **Version** | **Date** | **Author** | **Amendments** |
| 1 | 0.0a | 26-Apr-2016 | Arun Kumar V G | Draft version |
| 2 | 0.1 | 23-May-2016 | Arun Kumar V G | Draft version |
| 3 | 0.2 | 03-June-2016 | Arun Kumar V G | Draft version |
| 4 | 1.1 | 07-June-2016 | Arun Kumar V G | Release Version |
| 5 | 1.3 | 17-June-2016 | Arun Kumar V G | Release Version |
| 6 | 1.4 | 21-June-2016 | Arun Kumar V G | Release Version |

# Introduction

## Purpose

This document covers the Enterprise User Management application design aspects.

## Scope

This caters to the high level information so as to meet the objective of making the Enterprise User Management.

## Acronyms

|  |  |
| --- | --- |
| EUM | Enterprise User Management |
| LEO | Law Enforcement Officer |
| AM | Asst. Manager |

## References

# Block diagram for User Management

## High level Design for Enterprise User Management

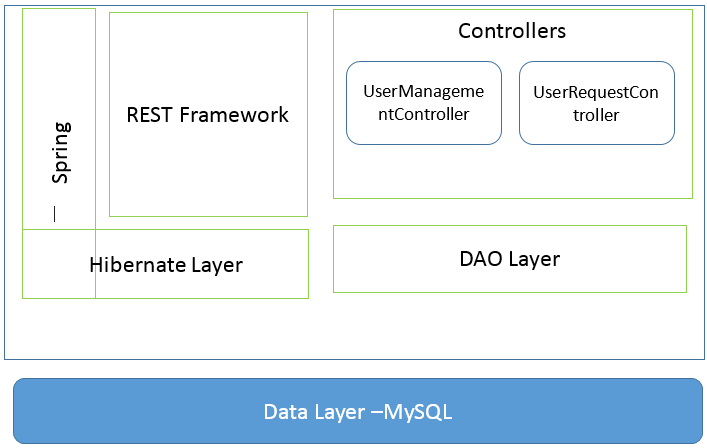


Figure: Block Diagram for Enterprise User Management

## Enterprise User Login Management

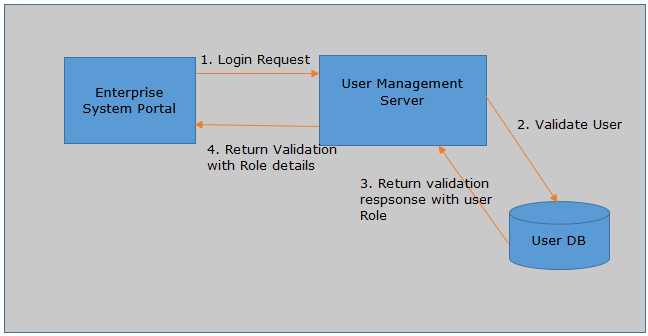


Figure: Block Diagram for Enterprise User Login Management

## Enterprise Schema Diagram

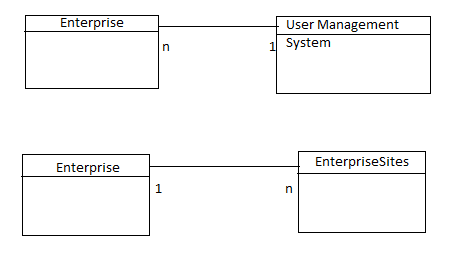


Figure A: Enterprise User Management System.

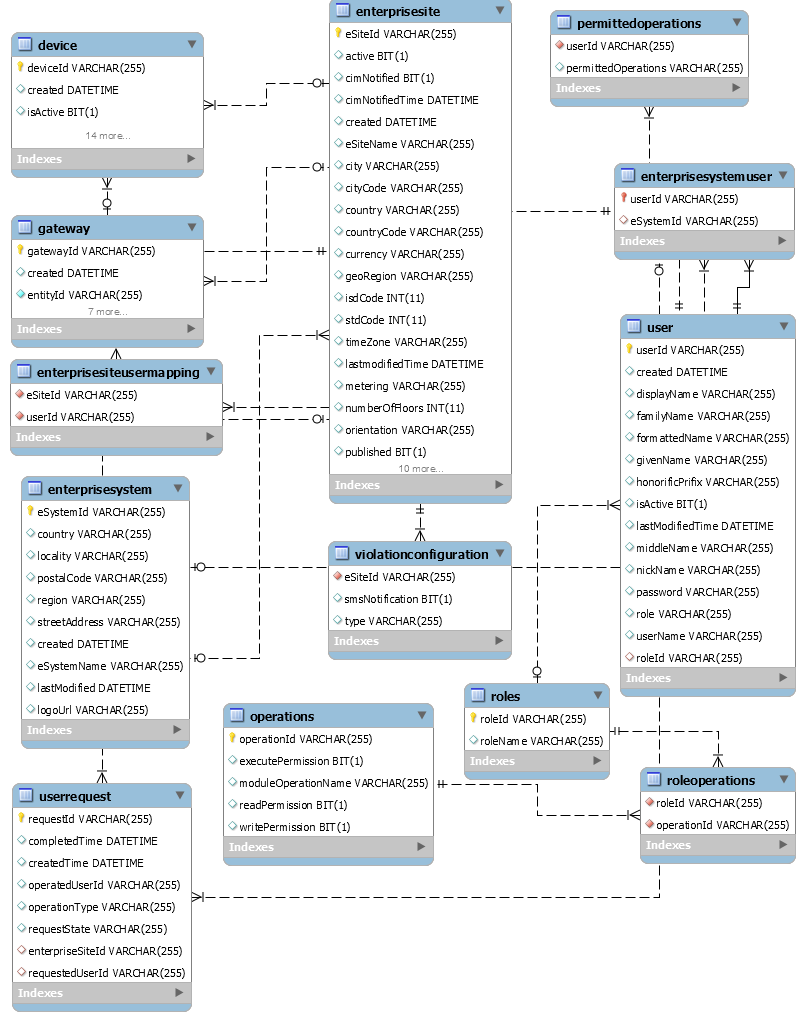


Figure B: User Management with User Request Handling Schema

## Multitenant Enterprise User Management Architecture

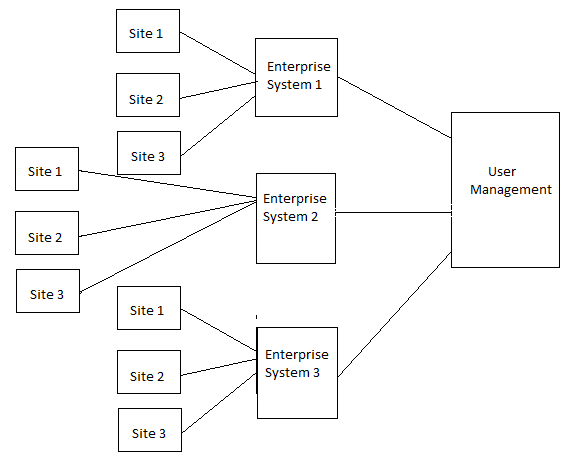


Figure: Multitenant Architecture of Enterprise User Management

# Enterprise User Management Overview

Enterprise User management consists of broad level of operations and management. Below are the design assumptions made.

* User will be basically associated with one Enterprise. So User cannot belong to more than one Enterprise.
* User can be associated with many Enterprise Sites. Same User can look after the activities of more than one Enterprise Sites.
* Each User has associated with one Role. As of now it cannot be changed.
* User can be Enterprise Super Admin, Enterprise Field Support Admin, Enterprise Site Super Admin, Enterprise Advanced User, Enterprise Scheduled User, Enterprise Site User, Enterprise Manager, Enterprise Asst. Manager or Law Enforcement Officer. These are the different roles as of now.
* Each Role has n number of operations that he can perform.
* Enterprise Super Admin is responsible managing entire Enterprise System. He is the owner of the Enterprise System.
* Each Manager can be associated with one or more Enterprise Sites.
* Each Asst. Manager can only be associated with one Enterprise Sites only.
* Each Law Enforcement Officer can be associated with many Enterprise Sites.
* Law Enforcement Manager is responsible to enforce the Enterprise Sites rules and regulations.
* Each operation has 4 levels of access with view, read and full permissions.
* Create – Create the and manipulates Enterprise System date
* Read - Read Only Mode on UI
* Update – Update the existing resources
* Delete- Delete the resources

## Enterprise User Management Mappings

User Management Mapping consists of below mappings

* Users
* Roles
* Operations
* User Requests
* Enterprise System
* Enterprise Sites
* Mapping between Users and Roles
* Mapping between Roles and Operations
* Mapping between Users and Enterprise
* Mapping between Users and Enterprise Sites
* Mapping between Enterprise System and Enterprise Sites
* Mapping between Enterprise System and User Requests
* Mapping between User and User Requests

## Operations

Operations in the Enterprise User Management System are defined as below.

### Sites

This Operation is to manage Enterprise Sites, has Different level of access like Create, Read, Update and Delete Operations. The Assigned user will be able to access with these CRUD operations. Enterprise Super Admin will have the full access to the system, and will responsible for updating the Site Information.

### Site Status

This Operation is to manage Enterprise Sites Status and has Different level of access like Create, Read, Update and Delete Operations. Enterprise Super Admin and Field Admin will have the full access to the system, and will responsible for updating the Site Status Information.

### Authentication Source

This Operation is to update and retrieve the Authentication Source information and has Different level of access like Create, Read, Update and Delete Operations. The Enterprise Super Admin will be able to access with perform CRUD operations.

### Site Advanced Users

This Operation is to manage Enterprise Sites Status and has Different level of access like Create, Read, Update and Delete Operations. The Assigned user will be able to access with these CRUD operations. The Enterprise Super Admin and Enterprise Field Admin will able to modify the Site Advanced user’s information.

### Site Scheduled

This Operation is to manage Enterprise Sites Scheduled activities and has Different level of access like Create, Read, Update and Delete Operations. The Assigned user will be able to access with these CRUD operations. All the admin and managers will be able to manipulated these operations

### Site Users

This Operation is to manage Enterprise Sites Users Information and has Different level of access like Create, Read, Update and Delete Operations. The Enterprise Super admin and Enterprise Field Admin will be able to access with to do these CRUD operations.

### Zones

This Operation is to manage Enterprise Zone Status and has Different level of access like Create, Read, Update and Delete Operations. The Enterprise Super admin, Field Admin will be able to access and to do these CRUD operations.

### Places

This Operation is to manage Enterprise Sites Places and has Different level of access like Create, Read, Update and Delete Operations. The Enterprise Super admin, Field Admin will be able to access and to do these CRUD operations

### PINs

This Operation is to manage Enterprise Sites PINs Information and has Different level of access like Create, Read, Update and Delete Operations. All the Admins are having access to modify the information.

### Schedules

This Operation is to Schedules activities belongs to an Enterprise Site and has Different level of access like Create, Read, Update and Delete Operations. All the Admins are having access to modify the information.

### Override

This Operation is to update or Override Enterprise Sites all the field information and has Different level of access like Create, Read, Update and Delete Operations. The Assigned user will be able to access with these CRUD operations. All the Admins are having access to modify the information.

### Super Admins

This Operation is to role to update Enterprise system Super Admin Roles and permissions. It has Different level of access like Create, Read, Update and Delete Operations. The Assigned user will be able to access with these CRUD operations. The Enterprise Super admin will be having full access to the system.

### Site Super Admins

This Operation is to update Enterprise Site Super Admin Roles and permissions. It has Different level of access like Create, Read, Update and Delete Operations. The Enterprise Super admin will be having full access to the system.

## Roles

The Roles of the Enterprise System are defined as below.

### Enterprise Super Admin

The Enterprise Super Admin will be First and Super level Access to the Enterprise User Management System.

* Super Admin can manage one Enterprise Systems.
* Super Admin can manage any number of Enterprise Sites.
* Super Admin will be having the supreme access to the all the operations and management.
* Super Admin will be having permissions to handle all the user requests.
* Super Admin can cancel and create the Enterprise systems.

### Enterprise Field Support Admin

The Enterprise Filed Support Admin will be having the Second level and Super level Access to the Enterprise site Management System.

* Support Admin is responsible for handling all the Sites.
* Support Admin is having the Full access to the Enterprise Systems.
* Support Admin is responsible for handling the activities related schedules, sites, Site Status, and user management.

### Enterprise Site Super Admin

The Enterprise Site Super Admin will be having the Supreme access at the Enterprise Site level.

* Site Super Admin is responsible for handling one enterprise site.
* Site Super Admin should be able to raise the requests to Enterprise Site Management.
* Site Super Admin is responsible for handling the activities related schedules, sites, Site Status, and user management.

### Enterprise Site Advanced User

The Enterprise Site Super Admin will be having the Second level access to the Enterprise Sites.

* Site Advanced User responsible for handling one enterprise site.
* Site Advanced User should be able to raise the requests to Enterprise Site Management.
* Site Advanced user is responsible for handling the activities related schedules, sites, Site Status, zones, Places, Pins and user management.

### Enterprise Site Scheduled User

The Enterprise Site Super Admin will be having the operator level access at the Enterprise Site.

* Scheduled User should be able to manage the Scenes, Schedules and Override information.
* Scheduled User will be managing only one enterprise system.

### Enterprise Site User

The Enterprise Site User will have the last level of access to Enterprise sites. The enterprise site can manage any only of Enterprise System.

* Site user can manage more than only one Enterprise site.
* Site user will having read only access all the operations in the system.
* Site user does not have any update related operations.

### Enterprise Manager

The Enterprise Manager is second level super user for Enterprise Systems. The enterprise manager can manage any only of Enterprise System.

* Manager can manage more than one Enterprise sites.
* Manager is Responsible for raising user requests.
* Manager Handles entire Enterprise site management.

### Enterprise Assistant Manager

The Enterprise Assistant Manager is Third level super user for Enterprise Systems.

* Asst. Manager Responsible for handling Enterprise site operations such as, Inventory, Dashboard, configuration, violations, Live view.
* Asst. Manager can only be associated with one Enterprise site.

### Law Enforcement Officer(LEO)

The Law Enforcement Officer is part of Advisory committee. LEO will be monitoring all the Site activities, Violations and other monitoring aspects.

* Law Enforcement Officer Responsible for forcing rules and regulation for the Enterprise sites, and Enterprise system.
* LEO can manage any number of Enterprise Sites which belong to one system.
* LEO can be only associated to one Enterprise System.
* LEO can be associated with many Enterprise sites.
* LEO is able to access the violations, Reports, Live view and Dashboard

## Roles and operations Diagram

Below Diagrams specifies the permissions assigned to an individual role.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Permission/UserType | Enterprise  Admin | Enterprise Field Support Admin | Site Super Admin | Site  Advanced User | Site Scheduled User |
| Sites | CRUD | CRUD |  |  |  |
| Site Status | RU | RU |  |  |  |
| Authentication Source | CRUD |  |  |  |  |
| Super Admins | CRUD |  |  |  |  |
| Site Super Admin | CRUD | CRUD | CRUD |  |  |
| Site Advanced users | CRUD | CRUD | CRUD | CRUD |  |
| Site Scheduled | CRUD | CRUD | CRUD | CRUD |  |
| Site Users | CRUD | CRUD | CRUD | CRUD |  |
| Zones | CRUD | CRUD | RU | RU |  |
| Places | CRUD | CRUD | RU | RU |  |
| PINs | CRUD | CRUD | RU | RU |  |
| Schedules | CRUD | CRUD | RU | RU | RU |
| Scenes | CRUD | CRUD | CRUD | CRUD | CRUD |
| Override | CRUD | CRUD | CRUD | CRUD | CRUD |

Figure 1. A. Roles and Operations Assigned to User

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Permission/UserType | Site User | Manager | Assistant Manager | Law Enforcement Officer |
| Sites |  |  |  |  |
| Site Status |  |  |  |  |
| Authentication Source |  |  |  |  |
| Super Admins |  |  |  |  |
| Site Super Admin |  |  |  |  |
| Site Advanced users |  |  |  |  |
| Site Scheduled |  |  |  |  |
| Site Users |  | RU | RU | RU |
| Zones |  | RU | RU | RU |
| Places |  |  | RU | RU |
| PINs |  | RU | RU | RU |
| Schedules |  | RU | RU | RU |
| Scenes |  | RU | RU | RU |
| Override | CRUD | CRUD | CRUD | CRUD |

Figure 1. B. Roles and Operations Assigned to User

# Configuration Settings

## Web.xml Settings

This section explains about the web.xml configuration.

Code Snippet:

*<?***xml version="1.0" encoding="UTF-8"***?>*<**web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xmlns="http://java.sun.com/xml/ns/javaee"  
 xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app\_2\_5.xsd"  
 version="2.5"**>  
 <**display-name**>User Management Service</**display-name**>  
 <**context-param**>  
 <**param-name**>contextConfigLocation</**param-name**>  
 <**param-value**>  
 /WEB-INF/spring/user-beans.xml  
 </**param-value**>  
 </**context-param**>  
  
 <**listener**>  
 <**listener-class**>org.springframework.web.context.ContextLoaderListener</**listener-class**>  
 </**listener**>  
  
   
  
 <**servlet**>  
 <**servlet-name**>appServlet</**servlet-name**>  
 <**servlet-class**>org.springframework.web.servlet.DispatcherServlet</**servlet-class**>  
 <**init-param**>  
 <**param-name**>contextConfigLocation</**param-name**>  
 <**param-value**>/WEB-INF/spring/appServlet/servlet-context.xml</**param-value**>  
 </**init-param**>  
 <**load-on-startup**>1</**load-on-startup**>  
 </**servlet**>  
  
 <**servlet-mapping**>  
 <**servlet-name**>appServlet</**servlet-name**>  
 <**url-pattern**>/</**url-pattern**>  
 </**servlet-mapping**>  
  
</**web-app**>

Steps:

1. Add the Dispatcher servlet configuration

<**servlet**>  
 <**servlet-name**>appServlet</**servlet-name**>  
 <**servlet-class**>org.springframework.web.servlet.DispatcherServlet</**servlet-class**>  
 <**init-param**>  
 <**param-name**>contextConfigLocation</**param-name**>  
 <**param-value**>/WEB-INF/spring/appServlet/servlet-context.xml</**param-value**>  
 </**init-param**>  
 <**load-on-startup**>1</**load-on-startup**>  
 </**servlet**>

1. Add the URI Mapping as shown below

<**servlet-mapping**>  
 <**servlet-name**>appServlet</**servlet-name**>  
 <**url-pattern**>/</**url-pattern**>  
 </**servlet-mapping**>

1. Save and close the web.xml file.

## Servlet-context settings

This section explains about the servlet-context.xml configuration.

Code Snippet:

*<?***xml version="1.0" encoding="UTF-8"***?>*<**beans:beans xmlns="http://www.springframework.org/schema/mvc"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xmlns:beans="http://www.springframework.org/schema/beans"  
 xmlns:context="http://www.springframework.org/schema/context"  
 xsi:schemaLocation="http://www.springframework.org/schema/mvc http://www.springframework.org/schema/mvc/spring-mvc-4.0.xsd  
 http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-4.0.xsd  
 http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-4.0.xsd  
 http://www.springframework.org/schema/task http://www.springframework.org/schema/task/spring-task-4.1.xsd"**>  
  
 *<!-- DispatcherServlet Context: defines this servlet's request-processing infrastructure -->  
  
 <!-- Enables the Spring MVC @Controller programming model -->* <**annotation-driven**>  
 <**message-converters**>  
 *<!--<beans:bean class="org.springframework.http.converter.StringHttpMessageConverter"/>-->  
 <!--<beans:bean class="org.springframework.http.converter.ByteArrayHttpMessageConverter"/>-->* <**beans:bean class="org.springframework.http.converter.json.MappingJackson2HttpMessageConverter"**>  
 <**beans:property name="objectMapper"**>  
 <**beans:bean class="com.knappsack.swagger4springweb.util.ScalaObjectMapper"**/>  
 </**beans:property**>  
 </**beans:bean**>  
 </**message-converters**>  
 </**annotation-driven**>  
 <**default-servlet-handler**/>  
  
 *<!-- Handles HTTP GET requests for /resources/\*\* by efficiently serving up static resources in the ${webappRoot}/resources directory -->* <**resources mapping="/resources/\*\*" location="/resources/"** />  
  
 *<!-- Resolves views selected for rendering by @Controllers to .jsp resources in the /WEB-INF/views directory -->* <**beans:bean class="org.springframework.web.servlet.view.InternalResourceViewResolver"**>  
 <**beans:property name="prefix" value="/WEB-INF/views/"** />  
 <**beans:property name="suffix" value=".jsp"** />  
 </**beans:bean**>  
   
 <**context:component-scan base-package="com.altiux.rest.controllers"** />  
 <**context:component-scan base-package="com.altiux.eum"** />  
  
   
</**beans:beans**>

Steps:

1. In this file add the controllers for component scanning as follows

<**beans:bean class="org.springframework.web.servlet.view.InternalResourceViewResolver"**>  
 <**beans:property name="prefix" value="/WEB-INF/views/"** />  
 <**beans:property name="suffix" value=".jsp"** />  
 </**beans:bean**>  
   
 <**context:component-scan base-package="com.altiux.rest.controllers"** />  
 <**context:component-scan base-package="com.altiux.eum"** />  
  
   
</**beans:beans**>

1. Save and close the servlet-context file.

## User-beans

This section explains about the user-beans.xml configuration.

Code Snippet:

*<?***xml version='1.0' encoding='UTF-8'** *?>*<**beans xmlns="http://www.springframework.org/schema/beans"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:task="http://www.springframework.org/schema/task"  
 xmlns:util="http://www.springframework.org/schema/util" xmlns:context="http://www.springframework.org/schema/context"  
 xmlns:mvc="http://www.springframework.org/schema/mvc" xmlns:aop="http://www.springframework.org/schema/aop"  
 xmlns:tx="http://www.springframework.org/schema/tx" xmlns:p="http://www.springframework.org/schema/p"  
 xsi:schemaLocation="  
 http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd  
 http://www.springframework.org/schema/task http://www.springframework.org/schema/task/spring-task.xsd  
 http://www.springframework.org/schema/util http://www.springframework.org/schema/util/spring-util.xsd  
 http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context.xsd  
 http://www.springframework.org/schema/mvc http://www.springframework.org/schema/mvc/spring-mvc.xsd  
 http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop.xsd  
 http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx.xsd"**>  
  
 *<!-- Root Context: defines shared resources visible to all other web components -->* <**task:annotation-driven** />  
  
 <**aop:aspectj-autoproxy proxy-target-class="true"** />  
  
 <**bean id="propertyConfigurer"  
 class="org.springframework.beans.factory.config.PropertyPlaceholderConfigurer"**>  
 <**property name="location" value="/WEB-INF/user.properties"**></**property**>  
 </**bean**>  
  
  
 *<!—User management Factory Bean -->* <**bean id="userDataSourceBean"  
 class="org.springframework.jdbc.datasource.DriverManagerDataSource"**>  
 <**property name="driverClassName" value="${sqldb\_user\_driver}"** />  
 <**property name="url" value="${sqldb\_user\_url}"** />  
 <**property name="username" value="${sqldb\_user\_username}"** />  
 <**property name="password" value="${sqldb\_user\_password}"** />  
 </**bean**>  
  
 <**bean id="enterpriseSessionFactoryBean"  
 class="org.springframework.orm.hibernate3.annotation.AnnotationSessionFactoryBean"**>  
 <**property name="dataSource" ref=" userDataSourceBean "** />  
 <**property name="packagesToScan"**>  
 <**list**>  
 <**value**>com.altiux.eum.entities</**value**>  
 </**list**>  
 </**property**>  
 <**property name="hibernateProperties"**>  
 <**props**>  
 <**prop key="hibernate.hbm2ddl.auto"**>update</**prop**>  
 <**prop key="hibernate.dialect"**>org.hibernate.dialect.MySQLDialect</**prop**>  
 <**prop key="hibernate.current\_session\_context\_class"**>thread</**prop**>  
 <**prop key="hibernate.show\_sql"**>true</**prop**>  
 </**props**>  
 </**property**>  
 </**bean**>  
  
<**bean id="esSessionFactoryBean"  
 class="com.altiux.eum.util.EnterpriseSystemSessionFactory" autowire="byName"** />  
  
 <**bean id="esInputValidator"  
 class="com.altiux.eum.util.EnterpriseSystemInputValidator" autowire="byType"** />  
  
 <**bean id="configSessionFactory"  
 class="com.altiux.eum.esite.util.ConfigSessionFactory" autowire="byName"** />  
 <**bean id="configInputValidator"  
 class="com.altiux.eum.esite.util.ConfigInputValidator" autowire="byType"** />  
  
  
 <**bean id="psUserManagementDao"  
 class="com.altiux.eum.esite.dao.impl.UserManagementDaoImpl" autowire="byType"** />  
  
 <**bean id="psUserRequestDao"  
 class="com.altiux.eum.esite.dao.impl.UserRequestDAOImpl" autowire="byType"** />  
  
 <**bean id="pcUserManagementDao"  
 class="com.altiux.eum.esystem.dao.impl.UserManagementDAOImpl" autowire="byType"** />  
  
 <**bean id="pcUserRequestDao"  
 class="com.altiux.eum.esystem.dao.impl.UserRequestDAOImpl" autowire="byType"** />  
  
 <**bean id="configDAO" class="com.altiux.eum.esystem.dao.impl.ConfigDAOImpl"  
 autowire="byType"** />  
  
 <**bean id="userManagementBootUpConfigurationBean"  
 class="com.altiux.eum.esystem.dao.UserManagementBootUpConfiguration" autowire="byType"** />  
  
 <**bean id="applicationConfigInitializerBean"  
 class="com.altiux.eum.esystem.dao.ApplicationConfigInitializer" autowire="byType"** />  
  
 <**bean id="serverConfigBean"  
 class="com.altiux.eum.esystem.dto.ApplicationConfigDTO"**>  
 <**property name="appName" value="${appName}"** />  
 <**property name="id" value="${appID}"** />  
 <**property name="ip" value="${appIP}"** />  
 <**property name="port" value="${appPort}"** />  
 </**bean**>   
  
 <**bean id="loginDAO"  
 class="com.altiux.eum.esite.dao.impl.LoginDAOImpl"  
 autowire="byType"** />  
  
 <**bean id="pcloginDAO" class="com.altiux.eum.esystem.dao.impl.EnterpriseSystemLoginDAOImpl"  
 autowire="byType"** />  
</**beans**>

Steps:

1. In this file, add the all the bean configurations required to load by the spring container.

Eg:

<**bean id="pcloginDAO" class="com.altiux.eum.esystem.dao.impl.EnterpriseSystemLoginDAOImpl"  
 autowire="byType"** />

1. Save and close the user-beans.xml file.

## User Properties File

This section explains about the User Property File configurations

Properties File content:

*# EUM DB configuration*

*//Db Username***sqldb\_EUM\_username**=**root**

*//Db password* **sqldb\_EUM\_password**=**root**

*//Db mysql database to connect* **sqldb\_EUM\_url**=**jdbc:mysql://localhost:3306/user\_management**

*//Db mysql driver* **sqldb\_EUM\_driver**=**com.mysql.jdbc.Driver***# Server Config*

*//application Name***appName**=**EnterpriseUserManagement**

*//application identifier* **appID**=**1008**

*//application server ip address* **appIP**=**172.16.1.119**

*//application server port* **appPort**=**8080**

Steps:

1. In this file, add the properties related to the bean configuration.

Eg: Data base properties.

*//Db Username***sqldb\_EUM\_username**=**root**

*//Db password* **sqldb\_EUM\_password**=**root**

*//Db mysql database to connect* **sqldb\_EUM\_url**=**jdbc:mysql://localhost:3306/user\_management**

*//Db mysql driver* **sqldb\_EUM\_driver**=**com.mysql.jdbc.Driver**

1. After defining all the properties required to load the bean.xml, save and close the file.

## Views

Views are used in the Swagger Configuration, please find the below details.

### documentation.jsp

This file used internally by the swagger components for rendering the APIs

Code Snippet:

<%@ **page contentType**="**text/html;charset=UTF-8**" **language**="**java**" %>  
  
<%@ **taglib uri**="**http://java.sun.com/jsp/jstl/core**" **prefix**="**c**" %>  
<%@ **page session**="**false**" %>  
<**html**>  
  
<**head**>  
 <**title**>Enterprise User Management REST API Documentation</**title**>  
 <**link href='http://fonts.googleapis.com/css?family=Droid+Sans:400,700' rel='stylesheet' type='text/css'**/>  
  
 <**link href='**<**c:url value='/resources/swagger-ui/css/reset.css'** />**' media='screen'  
 rel='stylesheet' type='text/css'**/>  
 <**link href='**<**c:url value='/resources/swagger-ui/css/screen.css'** />**' media='screen' rel='stylesheet'  
 type='text/css'**/>  
 <**link href='**<**c:url value='/resources/swagger-ui/css/reset.css'** />**' media='print'  
 rel='stylesheet' type='text/css'**/>  
 <**link href='**<**c:url value='/resources/swagger-ui/css/screen.css'** />**' media='print' rel='stylesheet'  
 type='text/css'**/>  
 <**script type="text/javascript" src="**<**c:url value='/resources/swagger-ui/lib/shred.bundle.js'**/>**"**></**script**>  
 <**script src='**<**c:url value='/resources/swagger-ui/lib/jquery-1.8.0.min.js'** />**' type='text/javascript'**></**script**>  
 <**script src='**<**c:url value='/resources/swagger-ui/lib/jquery.slideto.min.js'** />**'  
 type='text/javascript'**></**script**>  
 <**script src='**<**c:url value='/resources/swagger-ui/lib/jquery.wiggle.min.js'** />**'  
 type='text/javascript'**></**script**>  
 <**script src='**<**c:url value='/resources/swagger-ui/lib/jquery.ba-bbq.min.js'** />**'  
 type='text/javascript'**></**script**>  
 <**script src='**<**c:url value='/resources/swagger-ui/lib/handlebars-1.0.0.js'** />**'  
 type='text/javascript'**></**script**>  
 <**script src='**<**c:url value='/resources/swagger-ui/lib/underscore-min.js'** />**' type='text/javascript'**></**script**>  
 <**script src='**<**c:url value='/resources/swagger-ui/lib/backbone-min.js'** />**' type='text/javascript'**></**script**>  
 <**script src='**<**c:url value='/resources/swagger-ui/lib/swagger.js'** />**' type='text/javascript'**></**script**>  
 <**script src='**<**c:url value='/resources/swagger-ui/swagger-ui.js'** />**' type='text/javascript'**></**script**>  
 <**script src='**<**c:url value='/resources/swagger-ui/lib/highlight.7.3.pack.js'** />**'  
 type='text/javascript'**></**script**>  
 <**script src='**<**c:url value='/resources/swagger-ui/lib/swagger-oauth.js'** />**' type='text/javascript'**></**script**>  
  
 <**script type="text/javascript" th:inline="javascript"**>  
 **$**(**document**).**ready**(**function** () {  
  
 *displaySwaggerDocuments*();  
  
 **function** *displaySwaggerDocuments*() {  
 **var** resourceUrl = **'**<**c:url value="/documentation/resourceList"**/>**'**;  
 **window**.**swaggerUi** = **new SwaggerUi**({  
 **url**: resourceUrl,  
 **dom\_id**: **"swagger-ui-container"**,  
 **supportedSubmitMethods**: [**'get'**, **'post'**, **'put'**, **'delete'**],  
 onComplete: **function** (swaggerApi, swaggerUi) {  
 **if** (**console**) {  
 **console**.log(**"Loaded SwaggerUI"**)  
 **console**.log(swaggerApi);  
 **console**.log(swaggerUi);  
 }  
 **$**(**'pre code'**).each(**function** (i, e) {  
 ***hljs***.highlightBlock(e)  
 });  
 **if**(**typeof** *initOAuth* == **"function"**) {  
 */\*  
 initOAuth({  
 clientId: "your-client-id",  
 realm: "your-realms",  
 appName: "your-app-name"  
 });  
 \*/* }  
 },  
 onFailure: **function** (data) {  
 **if** (**console**) {  
 **console**.log(**"Unable to Load SwaggerUI"**);  
 **console**.log(data);  
 }  
 },  
 **docExpansion**: **"none"** });  
  
 **$**(**'#input\_apiKey'**).change(**function**() {  
 **var** key = **$**(**'#input\_apiKey'**)[0].value;  
 log(**"key: "** + key);  
 **if**(key && key.trim() != **""**) {  
 log(**"added key "** + key);  
 **window**.authorizations.add(**"key"**, **new** *ApiKeyAuthorization*(**"api\_key"**, key, **"query"**));  
 }  
 });  
 **window**.**swaggerUi**.load();  
 }  
 });  
 </**script**>  
</**head**>  
  
<**body class="swagger-section"**>  
<**div id='header'**>  
 <**div class="swagger-ui-wrap"**>  
  
 </**div**>  
</**div**>  
  
<**div id="message-bar" class="swagger-ui-wrap"**>**&nbsp;**</**div**>  
<**div id="swagger-ui-container" class="swagger-ui-wrap"**></**div**>  
</**body**>  
  
</**html**>

Steps:

1. This File is used by the Swagger for API documentation internally.
2. This is the template file and not required to change any lines.

### home.jsp

This file used internally by the swagger components as a home page.

Code Snippet:

<%@ **page contentType**="**text/html;charset=UTF-8**" **language**="**java**" %>  
  
<%@ **taglib uri**="**http://java.sun.com/jsp/jstl/core**" **prefix**="**c**" %>  
<%@ **page session**="**false**" %>  
<**html**>  
<**head**>  
 <**title**>Home</**title**>  
</**head**>  
<**body**>  
<**h1**>  
 Altiux Enterprise User Management Service  
</**h1**>  
  
<**p**>  
 <**a href='**<**c:url value="/documentation/"**/>**'**>View API documentation</**a**>  
</**p**>  
  
<**P**> The time on the server is **${**serverTime**}**. </**P**>  
</**body**>  
</**html**>

Steps:

1. This File is used by the Swagger for API documentation internally.
2. Add the Modification required by the welcome page.

Eg:

<**h1**>Altiux Enterprise User Management Service</**h1**>

1. Save the page and close the file.

## Resources

This folder will be used to place all the resources files for configuration it contains below files.

### Log4J Configuration

Code Snippet:

*<?***xml version="1.0" encoding="UTF-8"***?>***<!DOCTYPE log4j:configuration PUBLIC "-//APACHE//DTD LOG4J 1.2//EN" "log4j.dtd"*>***<**log4j:configuration xmlns:log4j="http://jakarta.apache.org/log4j/"**>  
  
 *<!-- Appenders -->* <**appender name="console" class="org.apache.log4j.ConsoleAppender"**>  
 <**param name="Target" value="System.out"** />  
 <**layout class="org.apache.log4j.PatternLayout"**>  
 <**param name="ConversionPattern" value="%-5p: %c - %m%n"** />  
 </**layout**>  
 </**appender**>  
   
 *<!-- Application Loggers -->* <**logger name="com.knappsack.swagger4springweb"**>  
 <**level value="info"** />  
 </**logger**>  
   
 *<!-- 3rdparty Loggers -->* <**logger name="org.springframework.core"**>  
 <**level value="info"** />  
 </**logger**>  
   
 <**logger name="org.springframework.beans"**>  
 <**level value="info"** />  
 </**logger**>  
   
 <**logger name="org.springframework.context"**>  
 <**level value="info"** />  
 </**logger**>  
  
 <**logger name="org.springframework.web"**>  
 <**level value="info"** />  
 </**logger**>  
  
 *<!-- Root Logger -->* <**root**>  
 <**priority value="warn"** />  
 <**appender-ref ref="console"** />  
 </**root**>  
   
</**log4j:configuration**>

Steps:

1. This File is used by the Swagger for API documentation internally.
2. This is the template file used to add the Log4J properties.

Eg:

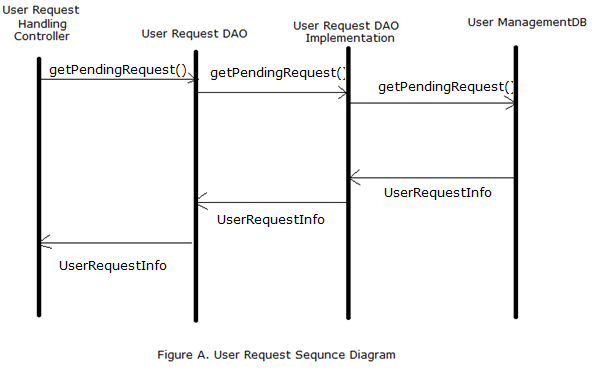
<**logger name="org.springframework.context"**>  
 <**level value="info"** />  
 </**logger**>

1. After the defining level for the log4j close the file.

# Sequence Diagram

## User Request Management

This below diagram talks about the Basic flow of User Request Handling flow.

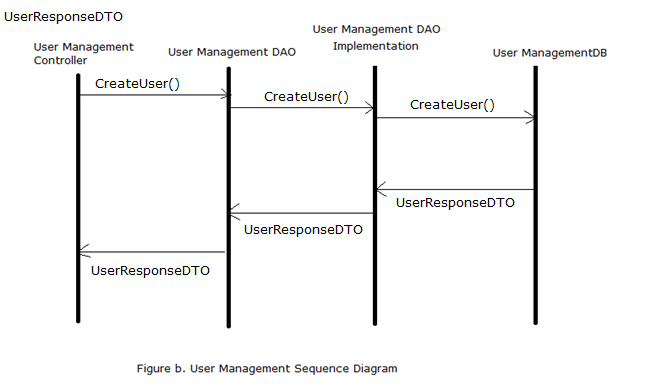


Steps:

1. User Request Controller invokes the DAO methods of User Request DAO.
2. User Request DAO methods queries to the Database.
3. The Model object from the DB will be will retrieved as a part of response.
4. The Controller will render it as Json response.

## User Management

This below diagram talks about the Basic flow of User management flow.

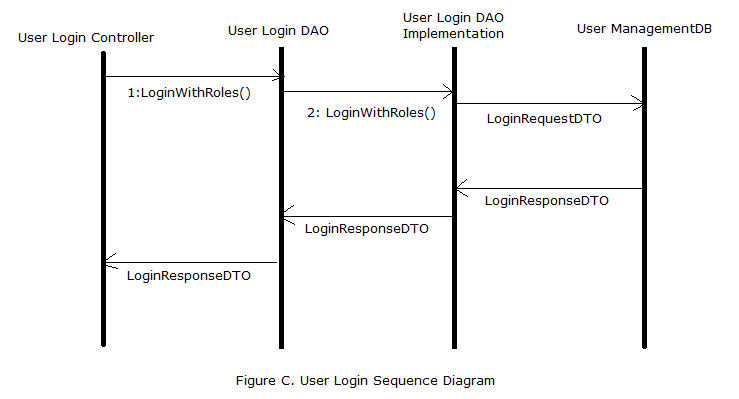


Steps:

1. User Management Controller invokes the DAO methods of User Management DAO.
2. User Management DAO methods queries to the Database.
3. The Model object from the DB will be will retrieved as a part of response.
4. The Controller will render it as Json response.

## Login Management

This below diagram talks about the Basic flow of User Login management flow.

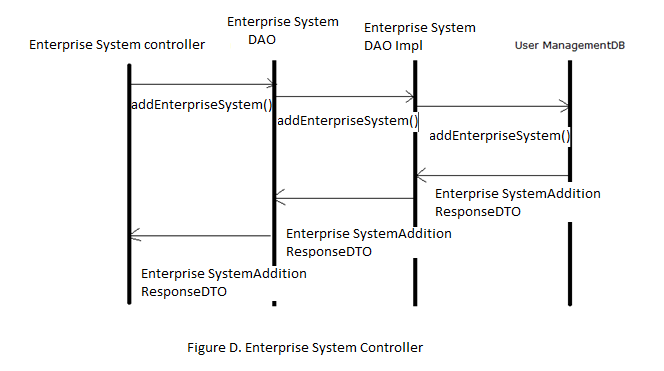


Steps:

1. User Login Controller invokes the DAO methods of User Login DAO.
2. User Login DAO methods queries to the Database, to validate the user name and password.
3. The Model object from the DB will be will retrieved as a part of response.
4. The Controller will render it as Json response.

## Enterprise System Management

This below diagram talks about the Basic flow of Enterprise System management flow.



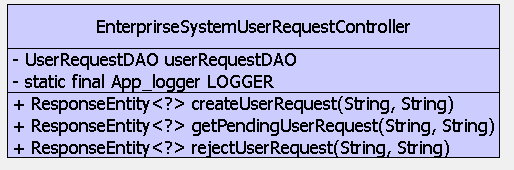
Steps:

1. Enterprise System Controller invokes the DAO methods of Enterprise System DAO.
2. Enterprise System DAO methods queries to the Database, to add the the enterprise system to the database.
3. In the Database the Enterprise system information will be persisted.
4. The Model object from the DB will be will retrieved as a part of response.
5. The Controller will render it as Json response.

# Class Diagrams

## Controllers:

### EnterpriseSystemUserRequestController



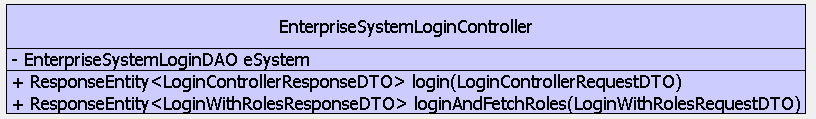
Methods:

1.public ResponseEntity<?> createUserRequest()

2.public ResponseEntity<?> deleteUserRequest()

3.public ResponseEntity<?> getPendingUserRequest()

### EnterpriseSystemLoginController

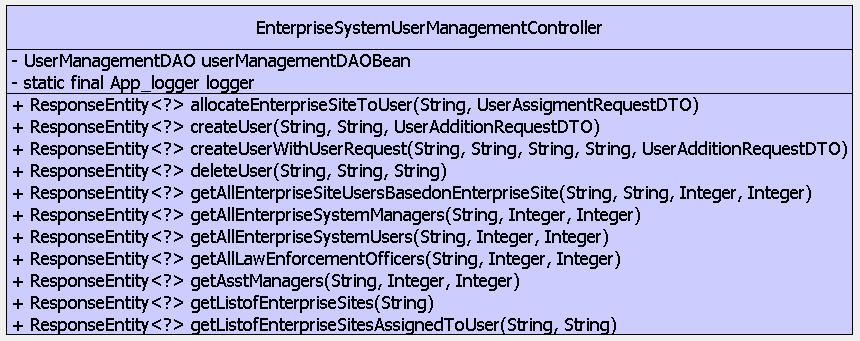


Methods:

public ResponseEntity<LoginControllerResponseDTO> login()

public ResponseEntity<LoginWithRolesResponseDTO> loginAndFetchRoles()

### EnterpriseSystemUserManagementController



Methods:

Public ResponseEntity<?> getAllEnterpriseSystemUsers()

public ResponseEntity<?> getAllEnterpriseSystemManagers()

public ResponseEntity<?> createUser()

public ResponseEntity<?> createUserWithUserRequest()

public ResponseEntity<?> deleteUser()

public ResponseEntity<?> getAllLawEnforcementOfficers()

public ResponseEntity<?> getAllEnterpriseSiteUsersBasedonEnterpriseSite()

public ResponseEntity<?> allocateEnterpriseSiteToUser()

public ResponseEntity<?> getListofEnterpriseSites()

public ResponseEntity<?> getAsstManagers()

public ResponseEntity<?> getListofEnterpriseSitesAssignedToUser()

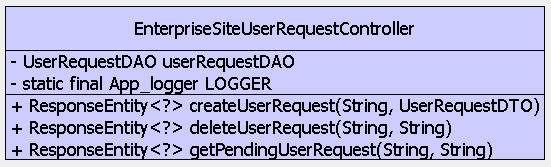
### EnterpriseSiteLoginController



Methods:

public ResponseEntity<?> login()

### EnterpriseSiteUserRequestController



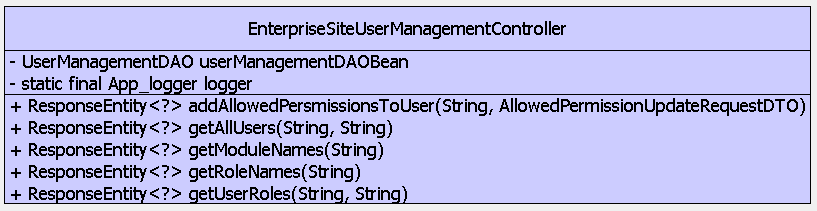
Methods:

public ResponseEntity<?> createUserRequest()

public ResponseEntity<?> deleteUserRequest()

public ResponseEntity<?> getPendingUserRequest()

### EnterpriseSiteUserMangementController



Methods:

public ResponseEntity<?> getUserRoles()

public ResponseEntity<?> addAllowedPersmissionsToUser()

public ResponseEntity<?> getAllUsers()

public ResponseEntity<?> getModuleNames()

public ResponseEntity<?> getRoleNames()

## DAO Interfaces:

### Enterprise System Login DAO

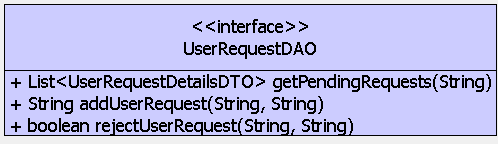


Abstract Methods:

Public LoginControllerResponseDTO login(LoginControllerRequestDTO requestDTO)

public LoginWithRolesResponseDTO loginWithRolesResponse(LoginWithRolesRequestDTO requestDTO)

### Enterprise System User Request DAO



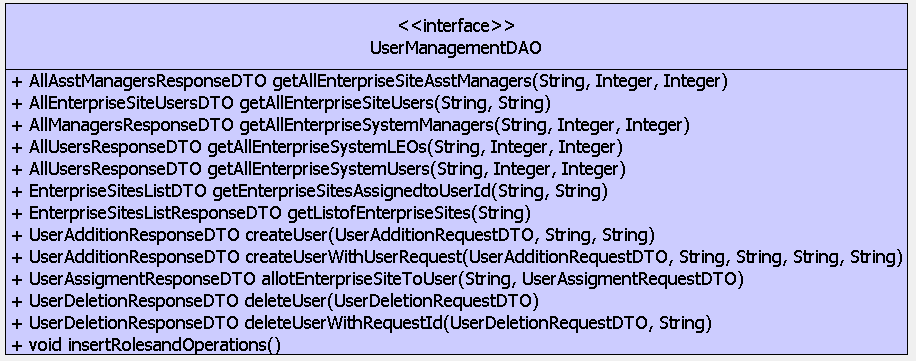
Abstract Methods:

public boolean rejectUserRequest(String systemId, String requestId) throws InvalidInputException

public List<UserRequestDetailsDTO> getPendingRequests(String enterpriseSystemId) throws InvalidInputException

public String addUserRequest(String enterpriseSystemId,String enterpriseSiteId) throws InvalidInputException

### Enterprise System User Management DAO



Abstract Methods:

public void insertRolesandOperations()

public UserDeletionResponseDTO deleteUser(UserDeletionRequestDTO dto) throws UserNotFoundException, InvalidInputException

public UserDeletionResponseDTO deleteUserWithRequestId(UserDeletionRequestDTO dto, String userRequestId) throws UserNotFoundException

public AllUsersResponseDTO getAllEnterpriseSystemLEOs(String enterpriseSystemId, Integer offset, Integer limit) throws InvalidInputException;

public AllAsstManagersResponseDTO getAllEnterpriseSiteAsstManagers(String enterpriseSystemId,Integer offset, Integer limit) throws InvalidInputException;

public AllEnterpriseSiteUsersDTO getAllEnterpriseSiteUsers(String enterpriseSiteId, String enterpriseSystemId) throws InvalidInputException;

public UserAssigmentResponseDTO allotEnterpriseSiteToUser(String systemId,UserAssigmentRequestDTO dto) throws InvalidInputException;

public EnterpriseSitesListResponseDTO getListofEnterpriseSites(String enterpriseSystemId) throws InvalidInputException;

public UserAdditionResponseDTO createUserWithUserRequest(UserAdditionRequestDTO requestDTO, String enterpriseSystemId, String enterpriseSiteId,String userRequestId, String adminId) throws UserAlreadyAvailableException, InvalidInputException;

public EnterpriseSitesListDTO getEnterpriseSitesAssignedtoUserId(String enterpriseSystemId,String userId)throws InvalidInputException, UserNotFoundException;

public UserAdditionResponseDTO createUser(UserAdditionRequestDTO requestDTO, String enterpriseSystemId, String adminId)throws UserAlreadyAvailableException, InvalidInputException;

public AllUsersResponseDTO getAllEnterpriseSystemUsers(String enterpriseSystemId, Integer offset, Integer limit)throws InvalidInputException;

public AllManagersResponseDTO getAllEnterpriseSystemManagers(String enterpriseSystemId, Integer offset, Integer limit)throws InvalidInputException;

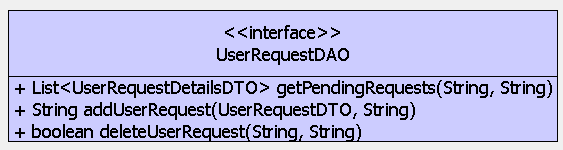
### Enterprise Site Login DAO



Abstract Methods:

public LoginSuccessDTO login(LoginCredentialsDTO requestDTO) throws InvalidInputException, EnterpriseSiteNotActiveException

### Enterprise Site User Request DAO



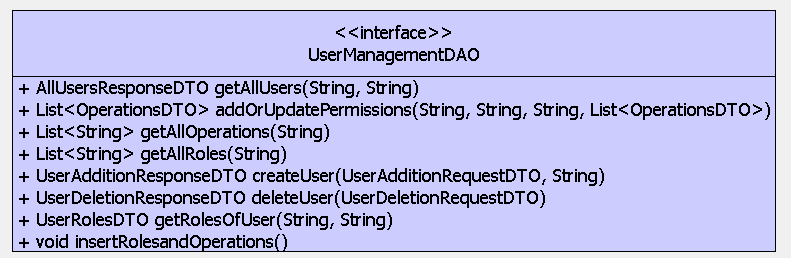
Abstract Methods:

public boolean rejectUserRequest(String SystemId, String requestId) throws InvalidInputException

public List<UserRequestDetailsDTO> getPendingRequests(String enterpriseSystemId) throws InvalidInputException

public String addUserRequest(String enterpriseSystemId,String enterpriseSiteId) throws InvalidInputException

### Enterprise Site User Management DAO



Abstract Methods:

public void insertRolesandOperations()

public List<String> getAllRoles(String enterpriseSiteId)

public List<String> getAllOperations(String enterpriseSiteId )

public UserAdditionResponseDTO createUser(UserAdditionRequestDTO requestDTO,

String enterpriseSiteId)

public UserDeletionResponseDTO deleteUser(UserDeletionRequestDTO dto)

public List<OperationsDTO> addOrUpdatePermissions(String enterpriseId,String userId, String adminUserId,

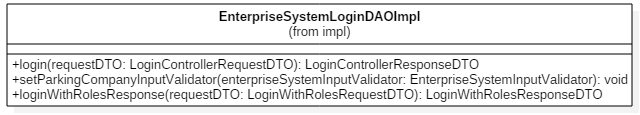
List<OperationsDTO> allowedPermissions)

public AllUsersResponseDTO getAllUsers(String enterpriseSiteId, String adminUserId)

public UserRolesDTO getRolesOfUser(String userId, String siteId)

## DAO Implementations:

### Enterprise System Login DAO Impl



Methods:

Public LoginControllerResponseDTO login(LoginControllerRequestDTO requestDTO)

public LoginWithRolesResponseDTO loginWithRolesResponse(LoginWithRolesRequestDTO requestDTO)

### Enterprise System User Management DAO Impl



Methods:

public void insertRolesandOperations()

public UserDeletionResponseDTO deleteUser(UserDeletionRequestDTO dto) throws UserNotFoundException, InvalidInputException

public UserDeletionResponseDTO deleteUserWithRequestId(UserDeletionRequestDTO dto, String userRequestId) throws UserNotFoundException

public AllUsersResponseDTO getAllEnterpriseSystemLEOs(String enterpriseSystemId, Integer offset, Integer limit) throws InvalidInputException;

public AllAsstManagersResponseDTO getAllEnterpriseSiteAsstManagers(String enterpriseSystemId,Integer offset, Integer limit) throws InvalidInputException;

public AllEnterpriseSiteUsersDTO getAllEnterpriseSiteUsers(String enterpriseSiteId, String enterpriseSystemId) throws InvalidInputException;

public UserAssigmentResponseDTO allotEnterpriseSiteToUser(String systemId,UserAssigmentRequestDTO dto) throws InvalidInputException;

public EnterpriseSitesListResponseDTO getListofEnterpriseSites(String enterpriseSystemId) throws InvalidInputException;

public UserAdditionResponseDTO createUserWithUserRequest(UserAdditionRequestDTO requestDTO, String enterpriseSystemId, String enterpriseSiteId,String userRequestId, String adminId) throws UserAlreadyAvailableException, InvalidInputException;

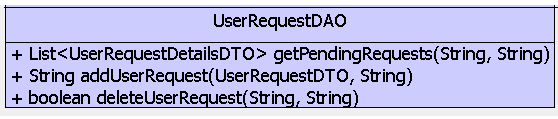
public EnterpriseSitesListDTO getEnterpriseSitesAssignedtoUserId(String enterpriseSystemId,String userId)throws InvalidInputException, UserNotFoundException;

public UserAdditionResponseDTO createUser(UserAdditionRequestDTO requestDTO, String enterpriseSystemId, String adminId)throws UserAlreadyAvailableException, InvalidInputException;

public AllUsersResponseDTO getAllEnterpriseSystemUsers(String enterpriseSystemId, Integer offset, Integer limit)throws InvalidInputException;

public AllManagersResponseDTO getAllEnterpriseSystemManagers(String enterpriseSystemId, Integer offset, Integer limit)throws InvalidInputException

### Enterprise System User Request DAO Impl



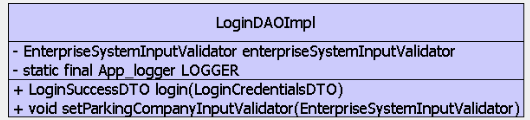
Methods:

public boolean rejectUserRequest(String systemId, String requestId) throws InvalidInputException

public List<UserRequestDetailsDTO> getPendingRequests(String enterpriseSystemId) throws InvalidInputException

public String addUserRequest(String enterpriseSystemId,String enterpriseSiteId) throws InvalidInputException

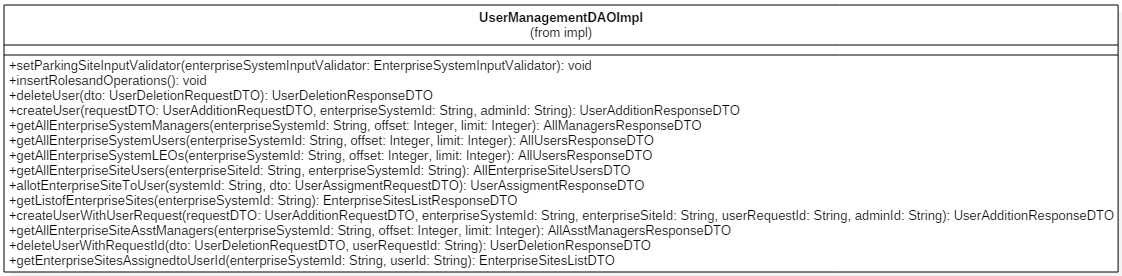
### Enterprise Site Login DAO Impl



Methods:

public LoginSuccessDTO login(LoginCredentialsDTO requestDTO) throws InvalidInputException, EnterpriseSiteNotActiveException

### Enterprise Site User Management DAO Impl



Methods:

public void insertRolesandOperations()

public List<String> getAllRoles(String enterpriseSiteId)

public List<String> getAllOperations(String enterpriseSiteId )

public UserAdditionResponseDTO createUser(UserAdditionRequestDTO requestDTO,

String enterpriseSiteId)

public UserDeletionResponseDTO deleteUser(UserDeletionRequestDTO dto)

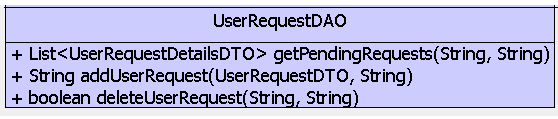
public List<OperationsDTO> addOrUpdatePermissions(String enterpriseId,String userId, String adminUserId,

List<OperationsDTO> allowedPermissions)

public AllUsersResponseDTO getAllUsers(String enterpriseSiteId, String adminUserId)

public UserRolesDTO getRolesOfUser(String userId, String siteId)

### Enterprise Site User Request DAO Impl



Methods:

public boolean rejectUserRequest(String systemId, String requestId) throws InvalidInputException

public List<UserRequestDetailsDTO> getPendingRequests(String enterpriseSystemId) throws InvalidInputException

public String addUserRequest(String enterpriseSystemId,String enterpriseSiteId) throws InvalidInputException

# User Management Source Configuration:

In this section, we can find the information about the package Structure of the Java code and Web code documentation.

Package Structure:

src

|

* Main

|

- Java

- Resources

- web app

## Java Resources:

Package wise Documentation:

## com.altiux.commons.errors

In this package the error handling classes are defined.

### Class: Error Response:

#### Constructors

|  |  |
| --- | --- |
| **Constructors** | |
| Constructors | Definition |
| [ErrorResponse](file:///D:\workspace-eum\EnterpriseUserManagement\EnterpriseUserManagementApp\doc\com\altiux\commons\errors\ErrorResponse.html#ErrorResponse--)() | Instantiates a new error response. |
| [ErrorResponse](file:///D:\workspace-eum\EnterpriseUserManagement\EnterpriseUserManagementApp\doc\com\altiux\commons\errors\ErrorResponse.html#ErrorResponse-int-java.lang.String-)(int errorCode, String errorMessage) | Instantiates a new error response. |

#### Methods

|  |  |
| --- | --- |
| **Methods** | |
| Constructors | Definition |
| public int getErrorCode() | Gets the error code. |
| public void setErrorCode(int errorCode) | Sets the error code. |
| public String getErrorMessage() | Gets the error message. |
| public void setErrorMessage  (String errorMessage) | Sets the error message. |

## com.altiux.commons.exceptions

In this package the all the exception handling classes are defined.

### AccessTokenSpecifiedException:

#### Constructors

|  |  |
| --- | --- |
| **Constructors** | |
| Constructors | Definition |
| AccessTokenSpecifiedException () | Instantiates an Access Token Specified Exception. |
| AccessTokenSpecifiedException(String message, Throwable cause,boolean enableSuppression, boolean writableStackTrace) | Instantiates an Access Token Specified Exception. |
| AccessTokenSpecifiedException(String message) | Instantiates an Access Token Specified Exception. |
| AccessTokenSpecifiedException(Throwable cause) | Instantiates an Access Token Specified Exception. |

### Class: AccountNameInUseException:

#### Constructors

|  |  |
| --- | --- |
| **Constructors** | |
| Constructors | Definition |
| AccountNameInUseException() | Instantiates an Account Name in Use Exception. |
| AccountNameInUseException (String message, Throwable cause,boolean enableSuppression, boolean writableStackTrace) | Instantiates an Account Name in Use Exception. |
| AccountNameInUseException (String message) | Instantiates an Account Name in Use Exception. |
| AccountNameInUseException (Throwable cause) | Instantiates an Account Name in Use Exception. |

### Class: DataInconsistentException:

#### Constructors

|  |  |
| --- | --- |
| **Constructors** | |
| Constructors | Definition |
| DataInconsistentException () | Instantiates a Data Inconsistent Exception. |
| DataInconsistentException (String message, Throwable cause,boolean enableSuppression, boolean writableStackTrace) | Instantiates a Data Inconsistent Exception. |
| DataInconsistentException (String message) | Instantiates a Data Inconsistent Exception. |
| DataInconsistentException (Throwable cause) | Instantiates a Data Inconsistent Exception. |

### Class: EnterpriseSiteNotFoundException:

#### Constructors

|  |  |
| --- | --- |
| **Constructors** | |
| Constructors | Definition |
| EnterpriseSiteNotFoundException () | Instantiates an Enterprise Site Not Found Exception. |
| EnterpriseSiteNotFoundException (String message, Throwable cause,boolean enableSuppression, boolean writableStackTrace) | Instantiates an Enterprise Site Not Found Exception. |
| EnterpriseSiteNotFoundException (String message) | Instantiates an Enterprise Site Not Found Exception. |
| EnterpriseSiteNotFoundException (Throwable cause) | Instantiates an Enterprise Site Not Found Exception. |

### Class: EnterpriseSiteNotActiveException:

#### Constructors

|  |  |
| --- | --- |
| **Constructors** | |
| Constructors | Definition |
| EnterpriseSiteNotActiveException () | Instantiates an Enterprise Site Not Active Exception. |
| EnterpriseSiteNotActiveException (String message, Throwable cause,boolean enableSuppression, boolean writableStackTrace) | Instantiates an Enterprise Site Not Active Exception. |
| EnterpriseSiteNotActiveException (String message) | Instantiates an Enterprise Site Not Active Exception. |
| EnterpriseSiteNotActiveException (Throwable cause) | Instantiates an Enterprise Site Not Active Exception. |

### Class: EntityIDException:

#### Constructors

|  |  |
| --- | --- |
| **Constructors** | |
| Constructors | Definition |
| EntityIDException () | Instantiates an Entity Id Exception. |
| EntityIDException (String message, Throwable cause,boolean enableSuppression, boolean writableStackTrace) | Instantiates an Entity Id Exception. |
| EntityIDException (String message) | Instantiates an Entity Id Exception. |
| EntityIDException (Throwable cause) | Instantiates an Entity Id Exception. |

### Class: EntityNameException:

#### Constructors

|  |  |
| --- | --- |
| **Constructors** | |
| Constructors | Definition |
| EntityNameException () | Instantiates an Entity Name Exception. |
| EntityNameException (String message, Throwable cause,boolean enableSuppression, boolean writableStackTrace) | Instantiates an Entity Name Exception. |
| EntityNameException (String message) | Instantiates an Entity Name Exception. |
| EntityNameException (Throwable cause) | Instantiates an Entity Name Exception. |

### Class: InstanceIdNotProvidedException:

#### Constructors

|  |  |
| --- | --- |
| **Constructors** | |
| Constructors | Definition |
| InstanceIdNotProvidedException () | Instantiates an Instance Id Not Provided Exception. |
| InstanceIdNotProvidedException (String message, Throwable cause,boolean enableSuppression, boolean writableStackTrace) | Instantiates an Instance Id Not Provided Exception. |
| InstanceIdNotProvidedException (String message) | Instantiates an Instance Id Not Provided Exception. |
| InstanceIdNotProvidedException (Throwable cause) | Instantiates an Instance Id Not Provided Exception. |

### Class: InvalidInputException:

#### Constructors

|  |  |
| --- | --- |
| **Constructors** | |
| Constructors | Definition |
| InvalidInputException () | Instantiates an Invalid Input Exception. |
| InvalidInputException (String message, Throwable cause,boolean enableSuppression, boolean writableStackTrace) | Instantiates an Invalid Input Exception. |
| InvalidInputException (String message) | Instantiates an Invalid Input Exception. |
| InvalidInputException (Throwable cause) | Instantiates an Invalid Input Exception. |

### Class: LoginException:

#### Constructors

|  |  |
| --- | --- |
| **Constructors** | |
| Constructors | Definition |
| LoginException () | Instantiates a Login Exception. |
| LoginException (String message, Throwable cause,boolean enableSuppression, boolean writableStackTrace) | Instantiates a Login Exception. |
| LoginException (String message) | Instantiates a Login Exception. |
| LoginException (Throwable cause) | Instantiates a Login Exception. |

### Class: NoDataFoundException:

#### Constructors

|  |  |
| --- | --- |
| **Constructors** | |
| Constructors | Definition |
| NoDataFoundException () | Instantiates a No Data Found Exception. |
| NoDataFoundException (String message, Throwable cause,boolean enableSuppression, boolean writableStackTrace) | Instantiates a No Data Found Exception. |
| NoDataFoundException (String message) | Instantiates a No Data Found Exception. |
| NoDataFoundException (Throwable cause) | Instantiates a No Data Found Exception. |

### Class: PasswordSpecifiedException:

#### Constructors

|  |  |
| --- | --- |
| **Constructors** | |
| Constructors | Definition |
| PasswordSpecifiedException () | Instantiates a Password Specified Exception. |
| PasswordSpecifiedException (String message, Throwable cause,boolean enableSuppression, boolean writableStackTrace) | Instantiates a Password Specified Exception. |
| PasswordSpecifiedException (String message) | Instantiates a Password Specified Exception. |
| PasswordSpecifiedException (Throwable cause) | Instantiates a Password Specified Exception. |

### Class: RequestBodyImproperException:

#### Constructors

|  |  |
| --- | --- |
| **Constructors** | |
| Constructors | Definition |
| RequestBodyImproperException () | Instantiates a Request Body Improper Exception. |
| RequestBodyImproperException (String message, Throwable cause,boolean enableSuppression, boolean writableStackTrace) | Instantiates a Request Body Improper Exception. |
| RequestBodyImproperException (String message) | Instantiates a Request Body Improper Exception. |
| NoDataFou RequestBodyImproperExceptionndException (Throwable cause) | Instantiates a Request Body Improper Exception. |

### Class: ServerInstanceNotFoundException:

#### Constructors

|  |  |
| --- | --- |
| **Constructors** | |
| Constructors | Definition |
| ServerInstanceNotFoundException () | Instantiates a Server Instance Not Found Exception. |
| ServerInstanceNotFoundException (String message, Throwable cause,boolean enableSuppression, boolean writableStackTrace) | Instantiates a Server Instance Not Found Exception. |
| ServerInstanceNotFoundException (String message) | Instantiates a Server Instance Not Found Exception. |
| NoDataFoundException (Throwable cause) | Instantiates a Server Instance Not Found Exception. |

### Class: UserAlreadyAvailableException:

#### Constructors

|  |  |
| --- | --- |
| **Constructors** | |
| Constructors | Definition |
| UserAlreadyAvailableException () | Instantiates a User Already Available Exception. |
| UserAlreadyAvailableException (String message, Throwable cause,boolean enableSuppression, boolean writableStackTrace) | Instantiates a User Already Available Exception. |
| UserAlreadyAvailableException (String message) | Instantiates a User Already Available Exception. |
| UserAlreadyAvailableException (Throwable cause) | Instantiates a User Already Available Exception. |

### Class: UserNotFoundException:

#### Constructors

|  |  |
| --- | --- |
| **Constructors** | |
| Constructors | Definition |
| UserNotFoundException () | Instantiates a User Not Found Exception. |
| UserNotFoundException (String message, Throwable cause,boolean enableSuppression, boolean writableStackTrace) | Instantiates a User Not Found Exception. |
| UserNotFoundException (String message) | Instantiates a User Not Found Exception. |
| UserNotFoundException (Throwable cause) | Instantiates a User Not Found Exception. |

## com.altiux.eum.esite.controller

This Package contains the all the controllers which belongs to enterprise site

### EnterpriseSiteLoginController

This Interface is used to handle the user Login management methods

#### Methods

|  |  |
| --- | --- |
| **Methods** | |
| Methods | Definition |
| public ResponseEntity<?> login() | This method is used to validate the existing user to login for the enterprise site. |

### EnterpriseSiteUserManagementController

This Interface is used to handle the user Login management methods

#### Methods

|  |  |
| --- | --- |
| **Methods** | |
| Methods | Definition |
| public ResponseEntity<?> getUserRoles() | This method is used to fetch roles and permissions belongs to the particular user. |
| public ResponseEntity<?> addAllowedPersmissionsToUser() | This method is used to add the allowed permissions to the user. |
| public ResponseEntity<?> getAllUsers() | This method is used to fetch all the users belonging to the enterprise site. |
| public ResponseEntity<?> getModuleNames() | This method is used to fetch all the oiperations and permissions in the enterprise system. |
| public ResponseEntity<?> getRoleNames() | This method is used to fetch all the roles in the enterprise system. |

### EnterpriseSiteUserRequestController

This Interface is used to handle the user Login management methods

#### Methods

|  |  |
| --- | --- |
| **Methods** | |
| Methods | Definition |
| public ResponseEntity<?> createUserRequest() | This method is used to create a new user request. |
| public ResponseEntity<?> deleteUserRequest() | This method is used to create a delete/cancel user request. |
| public ResponseEntity<?> getPendingUserRequest() | This method is used fetch the list of pending user requests. |

## com.altiux.eum.esystem.controller

This Package contains the all the controllers which belongs to enterprise site

### EnterprirseSystemUserRequestController

This Interface is used to handle the user Login management methods

#### Methods

|  |  |
| --- | --- |
| **Methods** | |
| Methods | Definition |
| public ResponseEntity<?> createUserRequest() | This method is used to create an new user request |
| public ResponseEntity<?> deleteUserRequest() | This method is used to delete/reject an new user request |
| public ResponseEntity<?> getPendingUserRequest() | This method is used to fetch all the pending user request |

### EnterpriseSystemController

This Interface is used to handle the user Login management methods

#### Methods

|  |  |
| --- | --- |
| **Methods** | |
| Methods | Definition |
| public ResponseEntity<?> addEnterpriseSystem() | This method is used to add new Enterprise System. |
| public ResponseEntity<?> addEnterpriseSite() | This method is used to add new Enterprise Site. |
| public ResponseEntity<?> activeOrDeactiveSystem() | This method is used to activate/Deactivate Enterprise System. |
| public ResponseEntity<?> activeOrDeactiveSite() | This method is used to activate/Deactivate Enterprise Site. |
| public ResponseEntity<?> insertOperations() | This method is used to add new operations to the system. |
| public ResponseEntity<?> insertRole() | This method is used to add new Role with operations and permissions. |

### EnterpriseSystemLoginController

This Interface is used to handle the user Login management methods

#### Methods

|  |  |
| --- | --- |
| **Methods** | |
| Methods | Definition |
| public ResponseEntity  <LoginControllerResponseDTO> login() | This method is used to validate the existing user to login for the enterprise system. |
| public ResponseEntity<LoginWithRolesResponseDTO> loginAndFetchRoles() | This method is used to validate the existing user to login for the enterprise system, also fetches the user with roles. |

### EnterpriseSystemUserManagementController

This Interface is used to handle the user Login management methods

#### Methods

|  |  |
| --- | --- |
| **Methods** | |
| Methods | Definition |
| Public  ResponseEntity<?> getAllEnterpriseSystemUsers() | This method is used to fetch all the enterprise system users existing in the enterprise system. |
| public  ResponseEntity<?> getAllEnterpriseSystemManagers() | This method is used to fetch all the enterprise system managers existing in the enterprise system. |
| public ResponseEntity<?> createUser() | This method is used to create a new user, to the enterprise system. |
| public  ResponseEntity<?> createUserWithUserRequest() | This method is used to create a new user with user request, to the enterprise system. |
| public ResponseEntity<?> deleteUser() | This method is used to delete the existing user in the system. |
| public  ResponseEntity<?> getAllLawEnforcementOfficers() | This method is used to fetch all the enterprise system law enforcement users, existing in the enterprise system. |
| public  ResponseEntity<?> getAllEnterpriseSiteUsers  BasedonEnterpriseSite() | This method is used to fetch all the enterprise sites users based on the enterprise site. |
| public  ResponseEntity<?> allocateEnterpriseSiteToUser() | This method is used to allocate the enterprise site to a user belongs existing in the enterprise system. |
| public  ResponseEntity<?> getListofEnterpriseSites() | This method is used to fetch all the enterprise sites existing in the enterprise system. |
| public  ResponseEntity<?> getAsstManagers() | This method is used to fetch all the enterprise system assistant managers existing in the enterprise system. |
| public  ResponseEntity<?> getListofEnterpriseSitesAssignedToUser() | This method is used to fetch all the enterprise sites assigned to the user existing in the enterprise system. |

## com.altiux.eum.esite.dao

This Package contains the all the DAO Interfaces which belongs to enterprise site

### LoginDAO

This Interface is used to handle the user Login management methods

#### Methods

|  |  |
| --- | --- |
| **Methods** | |
| Methods | Definition |
| public LoginSuccessDTO login  (LoginCredentialsDTO requestDTO)  throws InvalidInputException,  EnterpriseSiteNotActiveException,  UserNotFoundException | This method is used to validate the existing user to login for the enterprise site. |

### UserManagementDAO

This interface is used to handle the user management handling methods

#### Methods

|  |  |
| --- | --- |
| **Methods** | |
| Methods | Definition |
| public void insertRolesandOperations(); | This method is used to insert the roles and operations into the existing enterprise system. |
| public List<String>  getAllRoles(String enterpriseSiteId); | This method is used to fetch the roles enterprise system. |
| public List<String> getAllOperations(String  enterpriseSiteId ) throws InvalidInputException; | This method is used to all the operations belong to the system. |
| public UserAdditionResponseDTO createUser  (UserAdditionRequestDTO requestDTO,  String enterpriseSiteId)  throws UserAlreadyAvailableException,  InvalidInputException; | This method is used to create and new user in the system. |
| public UserDeletionResponseDTO deleteUser  (UserDeletionRequestDTO dto) throws  UserNotFoundException, InvalidInputException; | This method is used to delete existing user in the system. |
| public List<OperationsDTO>  addOrUpdatePermissions(String enterpriseId,  String userId, String adminUserId,  List<OperationsDTO> allowedPermissions)  throws InvalidInputException; | This method is used to update the user with new operations and permissions for the user. |
| public AllUsersResponseDTO getAllUsers  (String enterpriseSiteId,  String adminUserId)  throws InvalidInputException; | This method is used to fetch the list of users belonging to the system. |
| public UserRolesDTO getRolesOfUser  (String userId, String psId)  throws InvalidInputException; | This method is used to fetch list of operations belongs to the particular user. |

### UserRequestDAO

This interface is used to handle the user request handling methods

#### Methods

|  |  |
| --- | --- |
| **Methods** | |
| Methods | Definition |
| ApplicationConfigDTO  getApplicationConfiguration(String appname); | This method is used to fetch the application configuration information. |
| boolean addApplicationConfiguration  (final ApplicationConfigDTO configDTO); | This method is used to add the application configuration information. |
| boolean updateApplicationConfiguration  (final ApplicationConfigDTO configDTO); | This method is used to update the application configuration information. |

## com.altiux.eum.esystem.dao

This Package contains the all the DAO Interfaces which belongs to enterprise system

### ConfigDAO

This interface Handles the Configuration related Enterprise APP

#### Methods

|  |  |
| --- | --- |
| **Methods** | |
| Methods | Definition |
| ApplicationConfigDTO  getApplicationConfiguration(String appname); | This method is used to fetch the application configuration information. |
| boolean addApplicationConfiguration  (final ApplicationConfigDTO configDTO); | This method is used to add the application configuration information. |
| boolean updateApplicationConfiguration  (final ApplicationConfigDTO configDTO); | This method is used to update the application configuration information. |

### EnterpriseSystemDAO

This interface is used to handle the Enterprise system and Enterprise site related operations

#### Methods

|  |  |
| --- | --- |
| **Methods** | |
| Methods | Definition |
| public EnterpriseSystemAdditionResponseDTO  addEnterpriseSystem  (EnterpriseSystemAdditionRequestDTO dto) | This method is used to create the enterprise system with the given enterprise request id. |
| public EnterpriseSiteAdditionResponseDTO  addEnterpriseSite(String systemId,EnterpriseSiteAdditionRequestDTO dto) throws InvalidInputException; | This method is used to create an enterprise site. |
| public EnterpriseSiteStatusRepsonse  activeOrDectiveSite(String siteId, String status) throws InvalidInputException; | This method is used to activate or deactivate the enterprise system. |
| public EnterpriseSystemStatusRepsonse  activeOrDectiveSystem(String systemId, String status) throws InvalidInputException; | This method is used to activate or deactivate the enterprise site. |
| public OperationsInputResponseDTO  insertOperations(List<OperationsDTO> operations); | This method is used to add the new operations into the existing system. |
| public RolesDTO InsertRoles(RolesDTO dto); | This method is used to add the new Roles with operations belongs to the user into the existing system. |

### EnterpriseSystemLoginDAO

This interface used the handle the Enterprise System Login Management Related methods.

#### Methods

|  |  |
| --- | --- |
| **Methods** | |
| Methods | Definition |
| Public  LoginControllerResponseDTO login(LoginControllerRequestDTO requestDTO) | This method is used to validate the existing user for the enterprise system. In case validation fails the exception will be thrown. |
| public  LoginWithRolesResponseDTO loginWithRolesResponse  (LoginWithRolesRequestDTO requestDTO) | This method is used to validate the existing user for the enterprise system, along with it fetches the roles and operations belongs to the user. In case validation fails the exception will be thrown. |

### UserManagementDAO

This interface used the handle the user Management Related methods.

#### Methods

|  |  |
| --- | --- |
| **Methods** | |
| Methods | Definition |
| public void insertRolesandOperations() | This method is used to insert the roles and operations into the database. |
| public UserDeletionResponseDTO deleteUser(UserDeletionRequestDTO dto) throws UserNotFoundException, InvalidInputException | This method is used to the delete existing user in the enterprise system. |
| public UserDeletionResponseDTO deleteUserWithRequestId(UserDeletionRequestDTO dto, String userRequestId) throws UserNotFoundException; | This method is used to delete existing user in the enterprise system with user request. |
| public AllUsersResponseDTO getAllEnterpriseSystemLEOs(String enterpriseSystemId, Integer offset, Integer limit) throws InvalidInputException; | This method is used to fetch all the enterprise system Law Enforcement officers. |
| public AllAsstManagersResponseDTO getAllEnterpriseSiteAsstManagers(String enterpriseSystemId,Integer offset, Integer limit) throws InvalidInputException; | This method is used to fetch all the enterprise system assistant managers. |
| public AllEnterpriseSiteUsersDTO getAllEnterpriseSiteUsers(String enterpriseSiteId, String enterpriseSystemId) throws InvalidInputException; | This method is used to fetch all the enterprise site users. |
| public UserAssigmentResponseDTO allotEnterpriseSiteToUser(String systemId,UserAssigmentRequestDTO dto) throws InvalidInputException; | This method is used to assign the enterprise site belongs to the user. |
| public EnterpriseSitesListResponseDTO getListofEnterpriseSites(String enterpriseSystemId) throws InvalidInputException; | This method is used to fetch the list of enterprise sites. |
| public UserAdditionResponseDTO createUserWithUserRequest(UserAdditionRequestDTO requestDTO, String enterpriseSystemId,  String enterpriseSiteId,String userRequestId, String adminId) throws UserAlreadyAvailableException, InvalidInputException; | This method is used to create and user with Enterprise system with the user request raised by the Manager. Incase user already exists, it throws the user already available exception. |
| public EnterpriseSitesListDTO getEnterpriseSitesAssignedtoUserId(String enterpriseSystemId,String userId)  throws InvalidInputException, UserNotFoundException; | This method is used to fetch enterprise sites belongs to user. Throws the exception in case user not available. |
| public UserAdditionResponseDTO createUser(UserAdditionRequestDTO requestDTO, String enterpriseSystemId, String adminId)throws UserAlreadyAvailableException, InvalidInputException; | This method is used to create and user with Enterprise system. Incase user already exists, it throws the user already available exception. |
| public AllUsersResponseDTO getAllEnterpriseSystemUsers(String enterpriseSystemId, Integer offset, Integer limit)throws InvalidInputException; | This method is used to fetch all the enterprise site users. |
| public AllManagersResponseDTO getAllEnterpriseSystemManagers(String enterpriseSystemId, Integer offset, Integer limit)throws InvalidInputException; | This method is used to fetch all the enterprise system managers. |

### UserRequestDAO

This interface used the handle the user requests handling methods.

#### Methods

|  |  |
| --- | --- |
| **Methods** | |
| Methods | Definition |
| public boolean rejectUserRequest(String systemId, String requestId) throws InvalidInputException | This method is used to reject the user request sent. This operation is handled by Enterprise Admin. |
| public List<UserRequestDetailsDTO> getPendingRequests(String enterpriseSystemId) throws InvalidInputException | This method is used to fetch the number of pending user requests. |
| public String addUserRequest(String enterpriseSystemId,String enterpriseSiteId) throws InvalidInputException | This method is used to create an new user request for the enterprise system. |

## com.altiux.eum.esite.dao.impl

This Package contains the all the DAO Implementation which belongs to enterprise site

### LoginDAOImpl

This class is used to handle the user Login management methods

#### Methods

|  |  |
| --- | --- |
| **Methods** | |
| Methods | Definition |
| public LoginSuccessDTO login  (LoginCredentialsDTO requestDTO)  throws InvalidInputException,  EnterpriseSiteNotActiveException,  UserNotFoundException | This method is used to validate the existing user to login for the enterprise site. |

### UserManagementDaoImpl

This class is used to handle the user management handling methods

#### Methods

|  |  |
| --- | --- |
| **Methods** | |
| Methods | Definition |
| public void insertRolesandOperations(); | This method is used to insert the roles and operations into the existing enterprise system. |
| public List<String>  getAllRoles(String enterpriseSiteId); | This method is used to fetch the roles enterprise system. |
| public List<String> getAllOperations(String  enterpriseSiteId ) throws InvalidInputException; | This method is used to all the operations belong to the system. |
| public UserAdditionResponseDTO createUser  (UserAdditionRequestDTO requestDTO,  String enterpriseSiteId)  throws UserAlreadyAvailableException,  InvalidInputException; | This method is used to create and new user in the system. |
| public UserDeletionResponseDTO deleteUser  (UserDeletionRequestDTO dto) throws  UserNotFoundException, InvalidInputException; | This method is used to delete existing user in the system. |
| public List<OperationsDTO>  addOrUpdatePermissions(String enterpriseId,  String userId, String adminUserId,  List<OperationsDTO> allowedPermissions)  throws InvalidInputException; | This method is used to update the user with new operations and permissions for the user. |
| public AllUsersResponseDTO getAllUsers  (String enterpriseSiteId,  String adminUserId)  throws InvalidInputException; | This method is used to fetch the list of users belonging to the system. |
| public UserRolesDTO getRolesOfUser  (String userId, String psId)  throws InvalidInputException; | This method is used to fetch list of operations belongs to the particular user. |

### UserRequestDAOImpl

This class is used to handle the user request handling methods

#### Methods

|  |  |
| --- | --- |
| **Methods** | |
| Methods | Definition |
| ApplicationConfigDTO  getApplicationConfiguration(String appname); | This method is used to fetch the application configuration information. |
| boolean addApplicationConfiguration  (final ApplicationConfigDTO configDTO); | This method is used to add the application configuration information. |
| boolean updateApplicationConfiguration  (final ApplicationConfigDTO configDTO); | This method is used to update the application configuration information. |

## com.altiux.eum.esystem.dao.impl

This Package contains the all the DAO Implementation which belongs to enterprise system

### ConfigDAOImpl

This Class Handles the Configuration related Enterprise APP

#### Methods

|  |  |
| --- | --- |
| **Methods** | |
| Methods | Definition |
| ApplicationConfigDTO  getApplicationConfiguration(String appname); | This method is used to fetch the application configuration information. |
| boolean addApplicationConfiguration  (final ApplicationConfigDTO configDTO); | This method is used to add the application configuration information. |
| boolean updateApplicationConfiguration  (final ApplicationConfigDTO configDTO); | This method is used to update the application configuration information. |

### EnterpriseSystemDAOImpl

This class is used to handle the Enterprise system and Enterprise site related operations

#### Methods

|  |  |
| --- | --- |
| **Methods** | |
| Methods | Definition |
| public EnterpriseSystemAdditionResponseDTO  addEnterpriseSystem  (EnterpriseSystemAdditionRequestDTO dto) | This method is used to create the enterprise system with the given enterprise request id. |
| public EnterpriseSiteAdditionResponseDTO  addEnterpriseSite(String systemId,EnterpriseSiteAdditionRequestDTO dto) throws InvalidInputException; | This method is used to create an enterprise site. |
| public EnterpriseSiteStatusRepsonse  activeOrDectiveSite(String siteId, String status) throws InvalidInputException; | This method is used to activate or deactivate the enterprise system. |
| public EnterpriseSystemStatusRepsonse  activeOrDectiveSystem(String systemId, String status) throws InvalidInputException; | This method is used to activate or deactivate the enterprise site. |
| public OperationsInputResponseDTO  insertOperations(List<OperationsDTO> operations); | This method is used to add the new operations into the existing system. |
| public RolesDTO InsertRoles(RolesDTO dto); | This method is used to add the new Roles with operations belongs to the user into the existing system. |

### EnterpriseSystemLoginDAOImpl

This class used the handle the Enterprise System Login Management Related methods.

#### Methods

|  |  |
| --- | --- |
| **Methods** | |
| Methods | Definition |
| Public  LoginControllerResponseDTO login(LoginControllerRequestDTO requestDTO) | This method is used to validate the existing user for the enterprise system. In case validation fails the exception will be thrown. |
| public  LoginWithRolesResponseDTO loginWithRolesResponse  (LoginWithRolesRequestDTO requestDTO) | This method is used to validate the existing user for the enterprise system, along with it fetches the roles and operations belongs to the user. In case validation fails the exception will be thrown. |

### UserManagementDAOImpl

This class used the handle the user Management Related methods.

#### Methods

|  |  |
| --- | --- |
| **Methods** | |
| Methods | Definition |
| public void insertRolesandOperations() | This method is used to insert the roles and operations into the database. |
| public UserDeletionResponseDTO deleteUser(UserDeletionRequestDTO dto) throws UserNotFoundException, InvalidInputException | This method is used to the delete existing user in the enterprise system. |
| public UserDeletionResponseDTO deleteUserWithRequestId(UserDeletionRequestDTO dto, String userRequestId) throws UserNotFoundException; | This method is used to delete existing user in the enterprise system with user request. |
| public AllUsersResponseDTO getAllEnterpriseSystemLEOs(String enterpriseSystemId, Integer offset, Integer limit) throws InvalidInputException; | This method is used to fetch all the enterprise system Law Enforcement officers. |
| public AllAsstManagersResponseDTO getAllEnterpriseSiteAsstManagers(String enterpriseSystemId,Integer offset, Integer limit) throws InvalidInputException; | This method is used to fetch all the enterprise system assistant managers. |
| public AllEnterpriseSiteUsersDTO getAllEnterpriseSiteUsers(String enterpriseSiteId, String enterpriseSystemId) throws InvalidInputException; | This method is used to fetch all the enterprise site users. |
| public UserAssigmentResponseDTO allotEnterpriseSiteToUser(String systemId,UserAssigmentRequestDTO dto) throws InvalidInputException; | This method is used to assign the enterprise site belongs to the user. |
| public EnterpriseSitesListResponseDTO getListofEnterpriseSites(String enterpriseSystemId) throws InvalidInputException; | This method is used to fetch the list of enterprise sites. |
| public UserAdditionResponseDTO createUserWithUserRequest(UserAdditionRequestDTO requestDTO, String enterpriseSystemId,  String enterpriseSiteId,String userRequestId, String adminId) throws UserAlreadyAvailableException, InvalidInputException; | This method is used to create and user with Enterprise system with the user request raised by the Manager. Incase user already exists, it throws the user already available exception. |
| public EnterpriseSitesListDTO getEnterpriseSitesAssignedtoUserId(String enterpriseSystemId,String userId)  throws InvalidInputException, UserNotFoundException; | This method is used to fetch enterprise sites belongs to user. Throws the exception in case user not available. |
| public UserAdditionResponseDTO createUser(UserAdditionRequestDTO requestDTO, String enterpriseSystemId, String adminId)throws UserAlreadyAvailableException, InvalidInputException; | This method is used to create and user with Enterprise system. Incase user already exists, it throws the user already available exception. |
| public AllUsersResponseDTO getAllEnterpriseSystemUsers(String enterpriseSystemId, Integer offset, Integer limit)throws InvalidInputException; | This method is used to fetch all the enterprise site users. |
| public AllManagersResponseDTO getAllEnterpriseSystemManagers(String enterpriseSystemId, Integer offset, Integer limit)throws InvalidInputException; | This method is used to fetch all the enterprise system managers. |

### UserRegexUtil

This class used for handling validation using regular expressions.

#### Methods

|  |  |
| --- | --- |
| **Methods** | |
| Methods | Definition |
| public static boolean isAlphaNumeric(String s) | This method is used to validate the Regular Expressions for the alphanumeric Stings. |
| public static boolean validateUserName(String userName) | This method is used to validate the username. |

### UserRequestDAOImpl

This class used the handle the user requests handling methods.

#### Methods

|  |  |
| --- | --- |
| **Methods** | |
| Methods | Definition |
| public boolean rejectUserRequest(String systemId, String requestId) throws InvalidInputException | This method is used to reject the user request sent. This operation is handled by Enterprise Admin. |
| public List<UserRequestDetailsDTO> getPendingRequests(String enterpriseSystemId) throws InvalidInputException | This method is used to fetch the number of pending user requests. |
| public String addUserRequest(String enterpriseSystemId,String enterpriseSiteId) throws InvalidInputException | This method is used to create an new user request for the enterprise system. |

## com.altiux.rest.controllers

These controllers are used by the swagger module internally to handle the UI.

### ExampleDocumentationController

This controller is used to render the documentation page by the Swagger UI.

#### Constructors

|  |  |
| --- | --- |
| **Constructors** | |
| Constructors | Definition |
| public ExampleDocumentationController() | Instantiates a new ExampleDocumentationController. Sets up the all the controllers which are required document for swagger inrernally. |

#### Methods

|  |  |
| --- | --- |
| **Methods** | |
| Methods | Definition |
| public String documentation() | This method is used to bring documentation page on to the UI. Which renders the basic UI path. |

### HomePageController

This controller is used to render the documentation page by the Swagger UI.

#### Methods

|  |  |
| --- | --- |
| **Methods** | |
| Methods | Definition |
| public String home(Locale locale, Model model) | This method is used to bring display page on to the UI. Which renders the basic Home page of Swagger. |

## com.altiux.logger

This package contains the classes related to Log4J configurations.

### App\_logger

This App Logger class will contains methods related Logging API

#### Constructors

|  |  |
| --- | --- |
| **Constructors** | |
| Constructors | Definition |
| public App\_logger () | Instantiates a new App\_Logger. Sets up the logger properties. |

#### Methods

|  |  |
| --- | --- |
| **Methods** | |
| Methods | Definition |
| public void info(String className,String method, String message) | This API is used to set the information level in the Logging mechanism. |
| public void error(String className,String method, String message) | This API is used to set the Error level in the Logging mechanism. |
| public void debug(String className,String method, String message) | This API is used to set the Debug level in the Logging mechanism. |
| public void warn(String className,String method, String message) | This API is used to set the Warn level in the Logging mechanism. |

### EModuleName

This is an enum used to set the logger levels for different modules.

#### Constants

|  |  |
| --- | --- |
| **Constants** | |
| Constants | Definition |
| CONTROLLER | This is used to indicate the CONTROLLER level |
| MODEL | This is used to indicate the MODEL level |
| PERSISTENCE | This is used to indicate the PERSISTENCE level |
| COMMUNICATION | This is used to indicate the COMMUNICATION level |
| MQTT | This is used to indicate the MQTT level |
| WEBSOCKETS | This is used to indicate the WEBSOCKETS level |
| UTILITY | This is used to indicate the UTILITY level |
| DATASERVICE | This is used to indicate the DATASERVICE level |
| CONFIG | This is used to indicate the CONFIG level |

### LoggerFactory

This is an factory class used to the fetch the different levels of the Logger Information.

#### Methods

|  |  |
| --- | --- |
| **Methods** | |
| Methods | Definition |
| public static App\_logger getLogger(EModuleName moduleName) | This method is used to fetch the different levels of Information as mentioned in the Emodule levels.  Also sets up the different levels such as DEBUG, INFO,WARN, ERROR. |

## com.altiux.eum.utils

This package consists of the util classes required to set the Session Factory.

### AltiuxSessionFactory

This class is used to set the Session factory at the spring – hibernate integration level.

#### Methods

|  |  |
| --- | --- |
| **Methods** | |
| Methods | Definition |
| public void setLocalSessionFactoryBean(  AnnotationSessionFactoryBean localSessionFactoryBean) | This method is used to set the annotation bean injection required for the setting up of annotation session factory bean. |
| public SessionFactory getSessionFactory() | This method is used to fetch the annotation session factory. |

### EnterpriseSiteConstants

#### Constants

|  |  |
| --- | --- |
| **Constants** | |
| Constants | Definition |
| ENTERPRISE\_ADMIN | Defined The Enterprise Admin |
| MANAGER | Defined The Enterprise Manager |
| ASST\_MANAGER | Defined The Enterprise Assistant Manager |
| LAW\_ENFORCEMENT\_OFFICER | Defined The Enterprise Law Enforcement Officer |
| USER | Defined The Enterprise User |
| STATUS\_INACTIVE | Defined The Status Inactive |
| STATUS\_ACTIVE | Defined The Status Active |
| STATUS\_ADDED | Defined The Status Added |
| STATUS\_PUBLISHED | Defined The Status Published |
| STATUS\_UNPUBLISHED | Defined The Status Unpublished |
| STATUS\_SYNC | Defined The Status Sync |
| TYPE\_ACTIVIVATE | Defined The Type Activate |
| TYPE\_DEACTIVATE | Defined The Type Deactivate |
| TYPE\_PUBLISH | Defined The Type Publish |
| TYPE\_UNPUBLISH | Defined The Type Unpublish |

### EnterpriseSystemInputValidator

This class is used to set the Session factory at the spring – hibernate integration level.

#### Methods

|  |  |
| --- | --- |
| **Methods** | |
| Methods | Definition |
| public void setLocalSessionFactoryBean(  AnnotationSessionFactoryBean localSessionFactoryBean) | This method is used to set the annotation bean injection required for the setting up of annotation session factory bean. |
| public SessionFactory getSessionFactory() | This method is used to fetch the annotation session factory. |

### EnterpriseSystemSessionFactory

This class is used to set the Session factory at the spring – hibernate integration level.

#### Methods

|  |  |
| --- | --- |
| **Methods** | |
| Methods | Definition |
| public  SessionFactory getSessionFactory() | This method is used to fetch the session factory bean. |
| public void setEnterpriseSessionFactoryBean  (SessionFactory sessionFactory) | This method is used to set the session bean. |

## com.altiux.eum.util

This Package contains the all the utility Implementation which belongs to enterprise site

### EnterpriseSiteUtils

This class is used to handle the Enterprise Site Util for REST APIs methods

#### Methods

|  |  |
| --- | --- |
| **Methods** | |
| Methods | Definition |
| public static MultiValueMap<String, String> getHeadersForGetAPI() | This method is used to add the static data for the http GET API |
| public static MultiValueMap<String, String> getHeadersForPostAPI() | This method is used to add the static data for the http POST API |

# Database Schema Design

This is section defines the Enterprise User Management Database schema. Which comprises of the different entities.

1. Enterprise System
2. Enterprise Site
3. User
4. User Requests
5. Devices
6. Gateways
7. Roles
8. Operations
9. Permitted Operations
10. Role Operations
11. Enterprise Site User Mapping
12. Enterprise System User
13. Violation Configuration
14. Enterprise Site Emails
15. Enterprise Site Photos
16. Enterprise Site Phone Numbers
17. User Emails
18. User Phone Numbers
19. User Photos

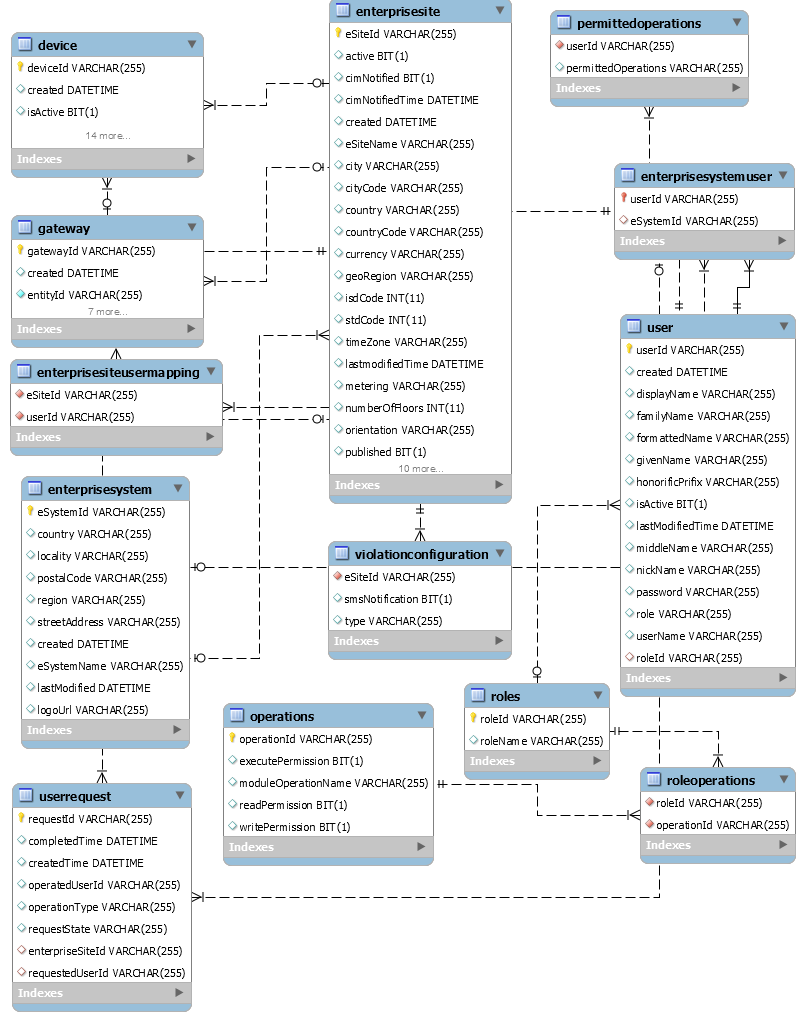
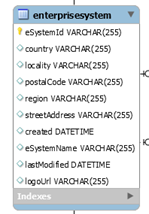


Figure: Enterprise User Management Database schema.

## Enterprise System

### Table Description

This Table has the information related to Enterprise System and its Associations.



### Table Information

Below are the fields used to define the Enterprise System

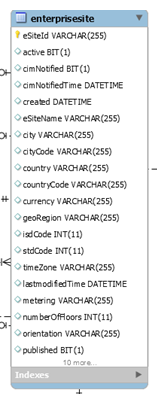
|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Description | Key | Type |
| Enterprise System Id | This is the id for the enterprise system that may own n number of enterprise sites. | Primary key | Varchar |
| active | Holds the Information whether, system is in the active or not. |  | Boolean |
| country | Holds the name of the country in which the Enterprise system is located. It is not unique. More than one Enterprise System can be located in a country, hence it cannot be primary key |  | Varchar |
| locality | Holds the name of the locality in which the Enterprise system is located. |  | Varchar |
| Postal Code | Holds the name of the Postal Code in which the Enterprise system is located. |  | Varchar |
| region | Holds the name of the Region in which the Enterprise system is located. |  | Varchar |
| Street address | Holds the name of the Street Address in which the Enterprise system is located. |  | Varchar |
| created | Holds the name of the Time, in which the Enterprise system is created. |  | Date Time |
| Last modified | Holds the name of the Time, in which the Enterprise system is last modified or updated. |  | Date Time |
| Login Url | Holds the login url of the Enterprise system. |  | Varchar |

## Enterprise Site

Below are the fields used to defines the Enterprise Sites

### Table Description

This Table has the information related to Enterprise Sites and its Associations.



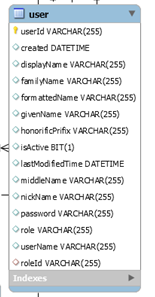
### Table Information

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Description | Key | Type |
| Enterprise Site Id | This is the id for the enterprise site that may own n number of gateways. | Primary key | Varchar |
| Enterprise Site Name | This Holds the name of the Enterprise Site. |  | Varchar |
| active | Holds the name of the Information whether, site is in the active or not. |  | Boolean |
| country | Holds the name of the country in which the Enterprise system is located. It is not unique. More than one Enterprise System can be located in a country, hence it cannot be primary key |  | Varchar |
| Geo Region | Holds the value of the Geo region in which Enterprise Site is located |  | Varchar |
| latitude | Holds the value of the Geological Latitude in which Enterprise Site is located |  | Double |
| longitude | Holds the value of the Geological Longitude in which Enterprise Site is located |  | Double |
| altitude | Holds the value of the Geological Altitude value in which Enterprise Site is located |  | Double |
| Demarked | Holds the value whether enterprise site is demarked or not |  | Varchar |
| Country Code | Holds the name of the Country Code in which the Enterprise Site is located. |  | Varchar |
| locality | Holds the name of the locality in which the Enterprise system is located. |  | Varchar |
| City | Holds the name of the City in which the Enterprise Site is located. |  | Varchar |
| City Code | Holds the City Code in which the Enterprise Site is located. |  | Varchar |
| Postal Code | Holds the name of the Postal Code in which the Enterprise Site is located. |  | Varchar |
| Orientation | Holds the name of the Postal Code in which the Enterprise Site is located. |  | Varchar |
| Number of Floors | Holds the name of the Postal Code in which the Enterprise Site is located. |  | Integer |
| Time Zone | Holds the name of the Postal Code in which the Enterprise Site is located. |  | Varchar |
| ISD Code | Holds the name of the ISD Code in which the Enterprise Site is located. |  | Integer |
| STD Code | Holds the name of the STD Code in which the Enterprise Site is located. |  | Integer |
| Facility Type | Holds the Facility type of the location |  | Varchar |
| region | Holds the name of the Region in which the Enterprise site is located. |  | Varchar |
| Street address | Holds the name of the Street Address in which the Enterprise site is located. |  | Varchar |
| created | Holds the name of the Time, in which the Enterprise site is created. |  | Date Time |
| Last modified | Holds the name of the Time, in which the Enterprise site is last modified or updated. |  | Date Time |
| Enterprise System Id | Holds the value of value of parent enterprise system, to which enterprise site belongs. | Foreign Key | Varchar |

## User

### Table Description

This Table has the information related to user and its Associations.



### Table Information

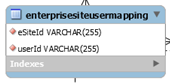
Below are the fields used to define the Enterprise System user.

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Description | Key | Type |
| UserId | The user id defines the unique identifier along with its primary key in the database. | Primary key | Varchar |
| User Name | The user name holds the value of the username. |  | Varchar |
| Display Name | Holds the display name of the user |  | Varchar |
| Family Name | Holds the Family name of the user |  | Varchar |
| Formatted Name | Holds the Formatted name of the user |  | Varchar |
| Given Name | Holds the Given name of the user |  | Varchar |
| Honorific Prefix | Holds the Honorific prefix value of the user, such as Mr., Mrs. or shri. |  | Varchar |
| Middle Name | Holds the Middle name of the user |  | Varchar |
| Nick Name | Holds the Nick name of the user |  | Varchar |
| password | Holds the Password value of the user |  | Varchar |
| role | Holds the Password value of the user |  | Varchar |
| is Active | Defines the Whether user is active or inactive for the system. |  | Bit |
| created | Holds the date and time, on which user has created. |  | Date Time |
| Last modified | Holds the date and time, on which user has been last modified. |  | Date Time |

## Enterprise Site User Mapping

### Table Description

This Table has the information related to Enterprise User mapping.



### Table Information

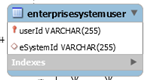
Below are the fields used to define the Enterprise Site User Mapping.

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Description | Key | Type |
| Enterprise Site Id | This field holds the id for the enterprise site. | Foreign Key | Varchar |
| UserId | This field holds the id for the enterprise site user, to which he is managing. | Foreign Key | Varchar |

## Enterprise System User

### Table Description

This Table has the information related to Enterprise System User and its Associations.



### Table Information

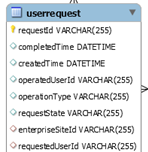
Below are the fields used to define the Enterprise Site User Mapping.

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Description | Key | Type |
| Enterprise System Id | This field holds the id for the enterprise site. | Foreign Key | Varchar |
| UserId | This field holds the id for the enterprise site user, to which he is managing. | Foreign Key | Varchar |

## User Request

### Table Description

This Table has the information related to User Request and its Associations.



### Table Information

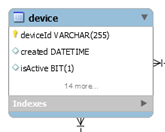
Below are the fields used to define the User Request information.

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Description | Key | Type |
| Request Id | This holds the id for the User request | Primary key | Varchar |
| Enterprise Site Id | This holds the id for the Enterprise Site |  | Varchar |
| Requested User Id | This holds the id for the user, who raised the request. |  | Varchar |
| Operation Type | Holds the value of the operations to be completed. |  | Varchar |
| Request State | Holds the status of the user request. |  | Varchar |
| Created Time | Holds the Date and Time where the user request created. |  | Date Time |
| Completed Time | Holds the Date and Time where the user request completed. |  | Date Time |
| Operated User Id | This holds the id for the user, to whom permissions are update. |  | Varchar |

## Devices

### Table Description

This Table has the information related to Enterprise System Device and its Associations.



### Table Information

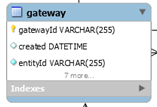
Below are the fields used to define the Enterprise System Device

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Description | Key | Type |
| Device Id | Each device must be given an id to identify them. | Primary key | varchar |
| created | The date and time of creation of the table |  | Date time |
| Is Active | It is a bit to know whether the device is active or not  If the bit is on, then it is active else inactive |  | Bit (interpreted as boolean) |
| Last Modified time | Holds the date and time when table was last modified |  | Date time |
| altitude | The height of the enterprise site |  | Double |
| latitude | The distance (north and south) from the equator |  | Double |
| longitude | The distance that is east west on the earth surface |  | Double |
| Manufacturer Name | Name of the manufacturing company that manufactured the device |  | Varchar |
| Serial no | The number that is given to each device  Manufacturing Year  Year of manufacture of the device |  | Integer |
| Node Id | The id for the node in the device which is used for communication purpose |  | Varchar |
| Node Name | The name of the node |  | Varchar |
| password | Holds the password to access the device |  | Varchar |
| Photo Type | The gateway through which the devices sends signal can be photographed  The type of this photo can be mentioned in this column |  | Varchar |
| Photo Url | The url for the photographs can be given for future access |  | Varchar |
| Gateway Id | The id of the gateway through which the device transfers the message  Information from many sensors will be sent to a single gateway | Foreign key | Varchar |
| Enterprise Site Id | The id of the Enterprise Site where the device is implemented | Foreign key | Varchar |

## Gate Ways

### Table Description

This Table has the information related to Enterprise System Gate ways and its Associations.



### Table Information

Below are the fields used to define the Enterprise System Gate ways.

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Description | Key | Type |
| Gateway Id | The id of the gateway through which the device transfers the message  Information from many sensors will be sent to a single gateway | Primary key | Varchar |
| created | The date and time of creation of the table |  | Date time |
| Entity Id | Holds the entity id for gateway | Unique key | Varchar |
| Entity Name | The name of the entity |  | Varchar |
| Last Modified | Holds the date and time when table was recently modified |  | Date time |
| Mac Address | The mac address for the gateway is specified  Information is passed using this address | Unique key | Varchar |
| Manufacturing Name | Name of the manufacturing company that manufactured the gateway |  | Varchar |
| Manufacture Year | Year of manufacture of the device |  | Integer |
| Serial no | The serial number of the gateway is provided | Foreign key | Varchar |
| Enterprise Site Id | The id of the Enterprise site where the device is implemented | Foreign key | Varchar |

## Roles

### Table Description

This Table has the information related to Enterprise System Roles.



### Table Information

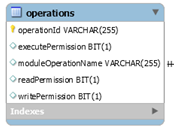
Below are the fields used to define the Enterprise System Roles.

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Description | Key | Type |
| Role Id | Holds the Identifier of the Role | Primary Key | Varchar |
| Role Name | Holds the Name of the Role |  | Varchar |

## Operations

### Table Description

This Table has the information related to Enterprise Operations Information.



### Table Information

Below are the fields used to define the Enterprise System Operations

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Description | Key | Type |
| Operation id | Holds the identifier of the operation. | Primary key | Varchar |
| Module Operation name | Holds the name of the operation. |  | Varchar |
| Execute permissions | Holds the Execute permission of the operation. |  | BIT |
| Read permissions | Holds the Read permission of the operation. |  | BIT |
| Write permissions | Holds the Write permission of the operation. |  | BIT |

## Permitted Operations

### Table Description

This Table has the information related to Enterprise Roles and Permitted Operations of the user.



### Table Information

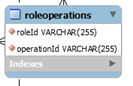
Below are the fields used to define the Enterprise Roles and permitted Operations

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Description | Key | Type |
| user Id | Holds the information related to the user Id | Foreign Key | Varchar |
| Operation Id | Holds the information related to the Operation Id | Foreign Key | Varchar |

## Role Operations

### Table Description

This Table has the information related to Enterprise Roles and Operations Mappings.



### Table Information

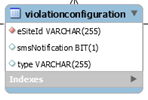
Below are the fields used to define the Role Operations.

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Description | Key | Type |
| Role Id | Holds the information related to the Role Id | Foreign Key | Varchar |
| Operation Id | Holds the information related to the Operation Id | Foreign Key | Varchar |

## Violation Configuration

### Table Description

This Table has the information related to Enterprise Violation and its types.



### Table Information

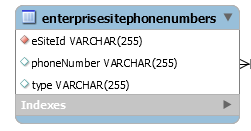
Below are the fields used to define the Enterprise Violation Information.

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Description | Key | Type |
| Enterprise Site Id | This is the id for the enterprise Site. | Primary key | Varchar |
| SMS Notifications | Contains the basic information related to SMS Notifications |  | Varchar |
| type | Contains the violation information |  | Varchar |

## Enterprise Site Phone Numbers

### Table Description

This Table has the information related to Enterprise Site Photos.



### Table Information

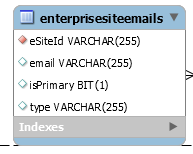
Below are the fields used to define the Enterprise Site Photos Information.

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Description | Key | Type |
| eSite Id | This is the id for the enterprise Site. | Foreign key | Varchar |
| Phone Number | Holds the phone number of the Enterprise Sites |  | Varchar |
| type | Holds the information of type of email   * Personal * Office |  | Varchar |

## Enterprise Site Emails

### Table Description

This Table has the information related to Enterprise Site Emails and its types.



### Table Information

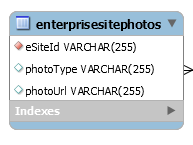
Below are the fields used to define the Enterprise Site Emails Information.

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Description | Key | Type |
| eSite Id | This is the id for the enterprise Site. | Primary key | Varchar |
| email | Contains the basic information related to Emails of enterprise site. |  | Varchar |
| Is Primary | Defines the email is primary or not  It contains a bit (interpreted as Boolean) to indicated primary or not  If the bit is on, then it is primary else not primary |  | Bit (interpreted as Boolean) |
| type | Holds the type of the email   * Primary * Secondary |  | Varchar |

## Enterprise Site Photos

### Table Description

This Table has the information related to Enterprise Site Photos.



### Table Information

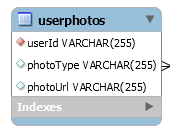
Below are the fields used to define the Enterprise Site Photos Information.

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Description | Key | Type |
| esiteId | Holds the value of the identifier of each authorized enterprise site | Foreign key | Varchar |
| Photo Type | Holds the type of the photo of Enterprise Site.  The types are:   * Thumbnail * Profile * Full size |  | Varchar |
| Photo Url | Holds the url in which the Sites photo has been stored |  | Varchar |

## User photos

### Table Description

This Table has the information related to User photos.



### Table Information

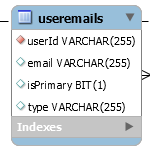
Below are the fields used to define the User photos.

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Description | Key | Type |
| userId | Holds the value of the identifier of each authorized user of the Enterprise System | Foreign key | Varchar |
| Photo Type | Holds the type of the photo of user.  The types are:   * Thumbnail * Profile * Full size |  | Varchar |
| Photo Url | Holds the url in which the user’s photo has been stored |  | Varchar |

## User Emails

### Table Description

This Table has the information related to user Emails.



### Table Information

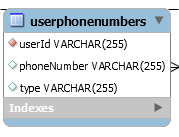
Below are the fields used to define the user Emails Information.

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Description | Key | Type |
| userId | Holds the value of the identifier of each authorized user of the Enterprise System | Foreign key | Varchar |
| email | Holds the email id of each user |  | Varchar |
| is Primary | Defines the email is primary or not  It contains a bit (interpreted as Boolean) to indicated primary or not  If the bit is on, then it is primary else not primary |  | Bit (interpreted as Boolean) |
| type | Defines the type of the email id.  The types are:   * Personal * Office |  | Varchar |

## User Phone Numbers

### Table Description

This Table has the information related to User Phone Numbers.



### Table Information

Below are the fields used to define the user Phone Numbers Information.

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Description | Key | Type |
| userId | Holds the value of the identifier of each authorized user of the Enterprise System. | Foreign key | Varchar |
| Phone Number | Holds the phone number of the users |  | Varchar |
| type | Holds the type of the phone number.  The types are:   * Mobile * Home * Office |  | Varchar |