Design Document

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| Serial no | Content | Page no |
| 1 | Purpose | 1 |
| 2 | User Case | 1 |
| 3 | Flow Chart | 2 |
| 4 | Database Design | 3 |
| 5 | Required Objects, Services, and Libraries | 5 |
| 6 | Technical Specifications | 6 |
| 7 | Controller class | 8 |

# Purpose

The document illustrates the low-level design details of role management module of the **User role** application. The document can be used by developers to implement the following functionalities,

* The administrator can add new role/update role of role functionality and store it in the database.

# User Case

The following are the use cases for which the use case diagram would be

Add New

ADD ROLE

ADMIN

# Flow Chart

The following are flow charts illustrate the process flow of the two use cases.

* **Add Batch**

User/Admin

Add User/admin

Validate the User/Admindetails

Display the Success message

Successfully Done

No

Yes

Invalid credentials

CRUD Done Successfully

Add/delete/update/post-User-Role

# Database Design

**Role Table:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Table Name** | **Column Name** | **Column Discription** | **Data Type** | **Data Size** | **Primary Key** | **Foreign Key** |
| Role\_Id | role\_id | Column is used for storing the Role-Id | Int |  | yes | - |
| Role\_Name | role\_name | Column is used for storing the Role-Name | varchar2 | 30 | - | - |
| Role\_Description | role\_description | Column is used for storing the Role-Description | varchar2 | 50 | - | - |
| Is\_Active | is\_active | Column is used for storing the Ia-Active | varchar2 | 30 | - | - |
| Created\_By | created\_by | Column is used for storing the Created-By | varchar2 | 30 | - | - |
| Updated\_By | updated\_by | Column is used for storing the Update-By | varchar2 | 30 | - | - |
| Created\_On | created\_on | Column is used for storing the Created-By | Date | - | - | - |
| Updated\_On | updated\_on | Column is used for storing the Update-on | Date | - | - | - |

**User Table:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Table Name | Column Name | Column Discription | Data Type | Data Size | Primary Key | Foreign Key |
| User\_Id | user\_id | Column is used for storing the User-Id | Int |  | Yes | Yes |
| User\_Code | user\_code | Column is used for storing the User-code | varchar | 50 | - | - |
| User\_phone | user\_phone | Column is used for storing the User-Phone | Int |  | - | - |
| User\_Email | user\_email | Column is used for storing the Email Id | Varchar | 50 | - | - |
| Pan\_No | pan\_no | Column is used for storing the Pan-no | Varchar | 50 | - | - |
| User\_Password | user\_password | Column is used for storing the Password | Varchar | 50 | - | - |
| Is\_Active | is\_active | Column is used for storing the Is-Active | varchar | 50 | - | - |

**User\_Role Table:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Table Name | Column Name | Column Discription | Data Type | Data Size | Primary Key | Foreign Key |
| User\_Role\_Id | user\_role\_id | Column is used for storing the user\_role\_id | Int |  | Yes | - |
| User\_Id | user\_id | Column is used for storing the user\_id | Int |  | - | Yes |
| Role\_Id | role\_id | Column is used for storing the role\_id | Int |  | - | Yes |
| Is\_Active | is\_active | Column is used for storing the is\_active | Varchar | 50 | - | - |
| Created\_By | created\_by | Column is used for storing the created\_by | Varchar | 50 | - | - |
| Updated\_By | updated\_by | Column is used for storing the updated\_by | Varchar | 50 | - | - |
| Created\_On | created\_on | Column is used for storing the created\_on | Date | - | - | - |
| Updated\_On | updated\_on | Column is used for storing the updated\_on | Date | - |  |  |

# Required Objects, Services, and Libraries

* 1. Common Components

|  |  |
| --- | --- |
| **Component Name** | **Component Description** |
| User-Dao | User component used for creating the appropriate DAO objects. This needs to be implemented using the DAO Factory pattern. |
| Service Layer | User service layer component used for creating the appropriate service objects. |
| User Exception | This is the exception class which will be used to throw any User exceptions which needs to be thrown as result of any user rule failure or validation error. |
| User\_Role Exception | This is the exceptions class used to throw any exception which arises in the system other than the user exceptions. |

* 1. UI Standard

The following are the UI standards

* Error Message should be in screen.
* Error Message should be in screen.
* All API should follows valid parameters.
* Date and Time Stored correctly every addition of role or user.
  1. Package Details

The following are the package details of the objects

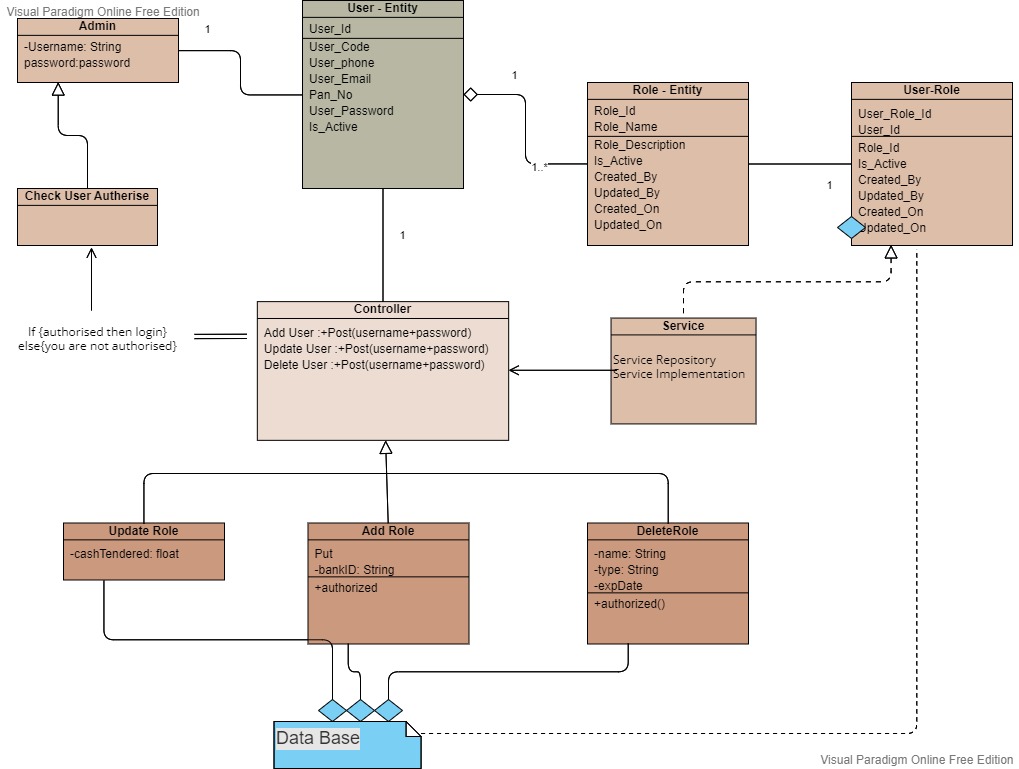
|  |  |
| --- | --- |
| **Objects** | **Package details** |
| Role Objects | com. userrole. role |
| Service Objects | com. userrole.object |
| Service Impel Objects | Service Impel Objects |
| DAO | com. userrole.dao |
| DAO Interface | com. userrole.dao.interfaces |
| DAO classes | com. userrole .dao |
| Controller | com userrole.controller |
| Exceptions | com. userrole.exceptions |
|  |  |

# 

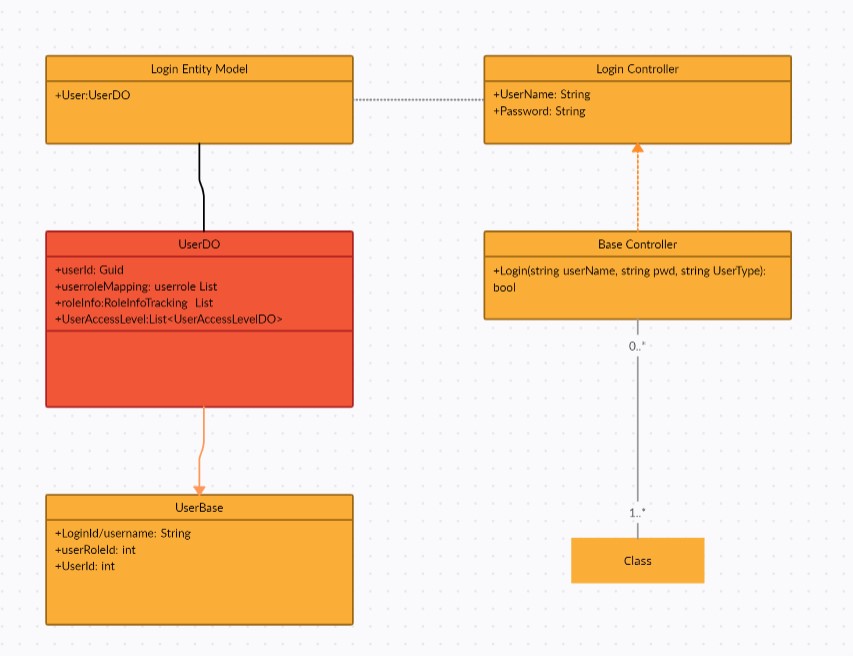
# Technical Specifications

## 6.1 UI Layer

Refer the requirement document for the prototype on how the screen should be designed.



## 6.1 Controller Layer



# Controller class:

|  |  |
| --- | --- |
| Class Name | UserController |
| Class Description | Request Controller for all request |

|  |  |
| --- | --- |
| Method Name | addRole/update role/delete role/create new role |
| Method Description | Methods are used to perform CRUD operation on UserRole and role |
| Method Input | User\_role |
| Method Output | Objects/ boolean |
| Algorithm | This invokes the add role method/ update role method  Delete role method/ Create role method |
| Validations | Not Null |

# DAO Layer:

|  |  |
| --- | --- |
| Class Name | DaoLayer |
| Class Description | This dao is used for adding new batches into the system. |

|  |  |  |
| --- | --- | --- |
| Method Name | | addDetails |
| Method Description | | Used for storing the roles details in the system as a new record |
| Method Input | | Role add |
| Method Output | | Boolean |
| Validations | | NA |
| Exception | Throws SQL Exception – when any underlying exception arises. | |

# Service class:

|  |  |
| --- | --- |
| Class Name | UserService |
| Class Description | Provide service for all request |

|  |  |
| --- | --- |
| Method Name | addRole /update role/delete role/create Method new role |
| Method Description | Methods are used to perform CRUD operation on UserRole and role |
| Method Input | User\_role |
| Method Output | Objects/ boolean |
| Algorithm | This invokes the add role method/ update role method  Delete role method/ Create role method |
| Validations | Not Null |