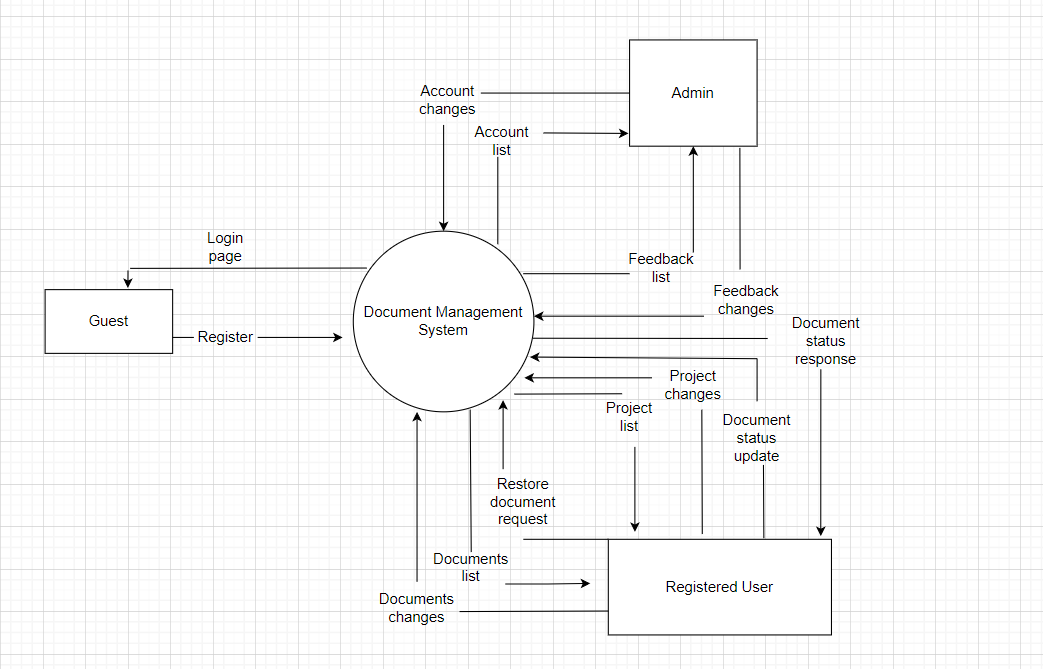
# Overview

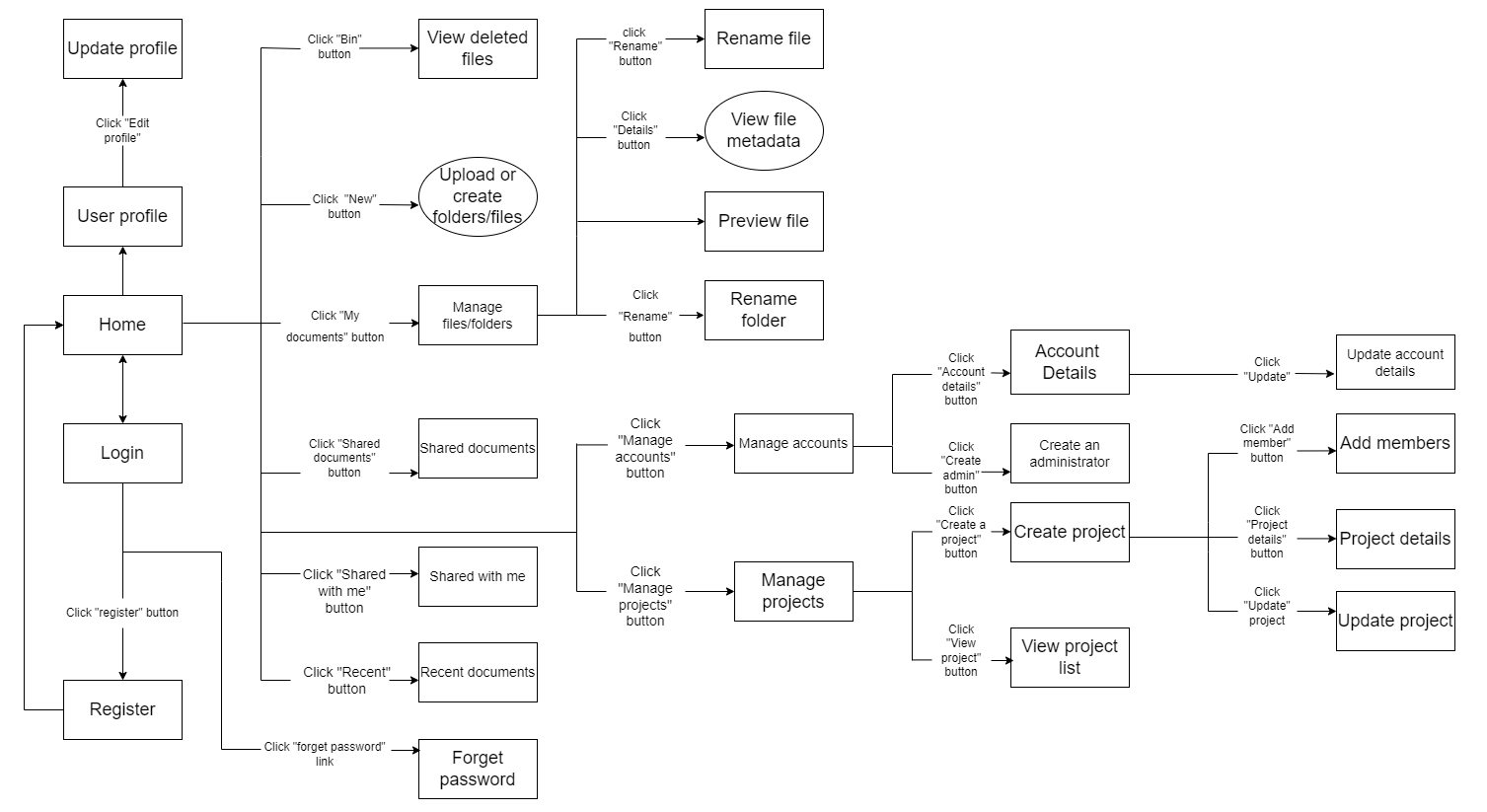
## Introduction

Document management system is designed to organize projects, business files and records digitally, whether they started out in paper form or were generated by software applications. Paper files are first converted to electronic format by scanning. This provides a more compact means of storage, our system will provide for each role lots of way to interact. For example, a guest will be able to view the public file, if you are a registered user, you will have tons of ways to play around with our system, for example you can view, create or upload, delete or update a folder, you can also do the same with files. And there is an admin who can manage all the account in the system.



## System Functions

### Screen Flow



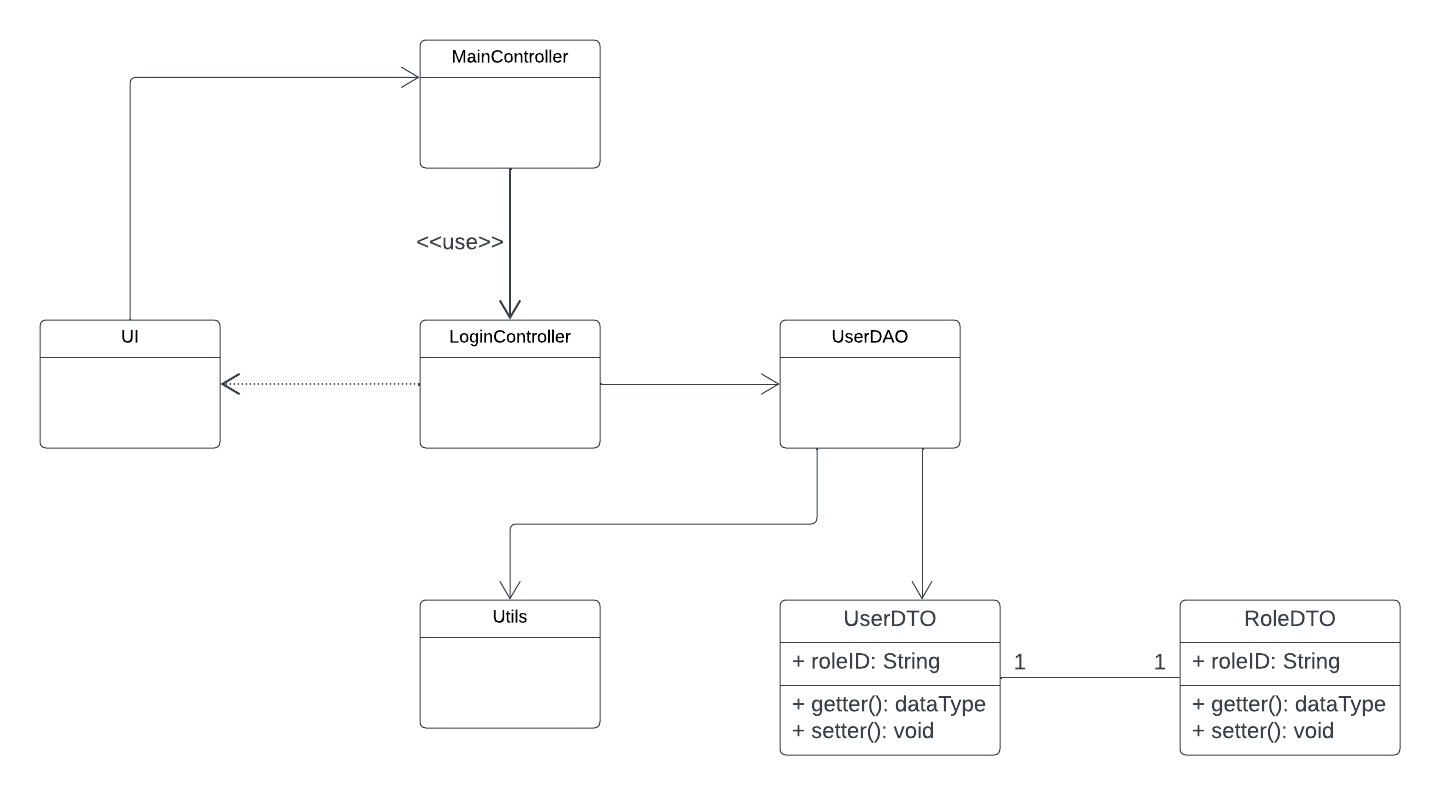
### Screen Details

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Feature** | **Screen** | **Description** |
| 1 | Authentication | Home | Show documents, options for each role |
| 2 | Authentication | Login | Allow users to type in their username, password |
| 3 | Authentication | Register | Create new users |
| 4 | Authentication | Forget password | Allow users to recover their password |
| 5 | Authentication | reCapcha | Check reCapcha |
| 6 | Account management | View user profile | Show all users’ information |
| 7 | Account management | Update profile | Allow users to change their profile |
| 8 | Account management | View account list | Show all the accounts have been registered |
| 9 | Account management | Search accounts | Allow administrator to search for a specific account |
| 10 | Account management | Account details | Show admin the details of selected account |
| 11 | Account management | Create administrator | Allow admin to create an account |
| 12 | Account management | Assign Administrator | Allow admin to assign his/her role to a user |
| 13 | Account management | Update account details | Allow admin to update the information of selected account |
| 14 | Account management | Assign Administrator role | Assign a user to become an administrator |

# Code Designs

## Authentication/Log in

### Class Diagram



### Class Specifications

#### UserDAO Class

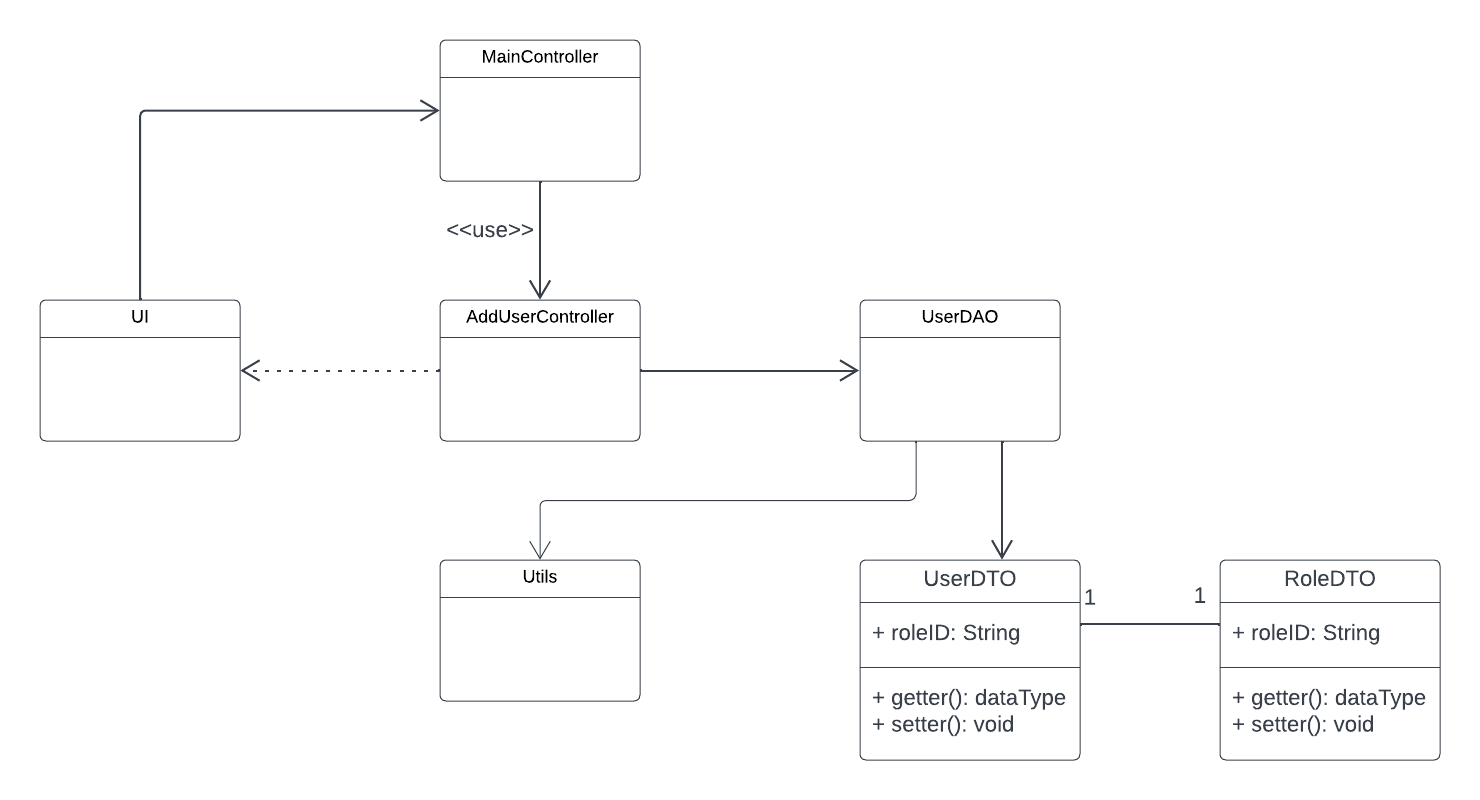
|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | checkLogin() | Call database to check user id and password. If user id and password are existed, return an object userDTO. |

#### LoginController Class

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | processRequest() | Receive user id and password from UI. Call function checkLogin(). If user id, password are existed, and if role is user, redirect to Home Page. If user id, password are existed, and if role is admin, redirect to Admin Page. |

## Authentication/Registration

### Class Diagram

****

### Class Specifications

#### UserDAO Class

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | createUser() | Call database to insert a new account to database. |
| 02 | checkUserExisted() | Call database to check user id is existed or not. |
| 03 | isDeleted() | Call database to check account is deleted or not. |
| 04 | checkEmailExisted | Call database to check email is deleted or not. |

#### AddUserController Class

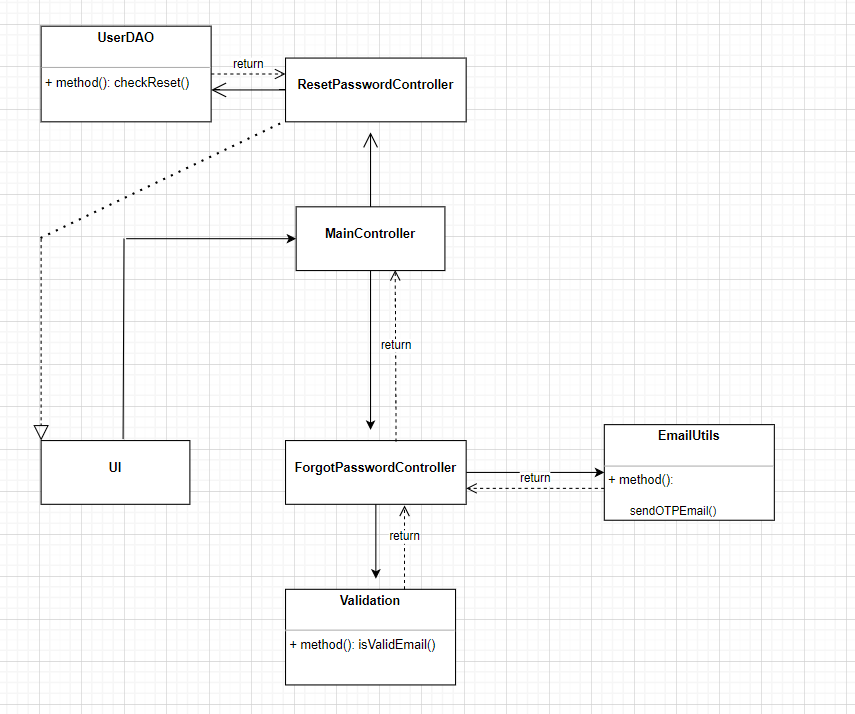
|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | processRequest() | Receive data from UI.  Call function checkUserExisted() and isDeleted().Check validation of data and show error in UI.  Call function createUser(). If success, show notification and redirect to Login Page. If fail, show error, notification, and let user enter data again. |

#### Validation Class

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | isValidEmail () | Validation the input |
| 02 | isValidPhoneNumber () | Validation the input |
| 03 | isValidUserID () | Validation the input |
| 04 | isValidPassword() | Validation the input |

## Authentication/Forgot Password

### Class Diagram:



### Class Specifications

#### Validation Class

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | isValidEmail() | Validation the input |

#### EmailUtils Class

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | sendOTPEmail() | Send OTP to email |

#### ForgotPasswordController Class

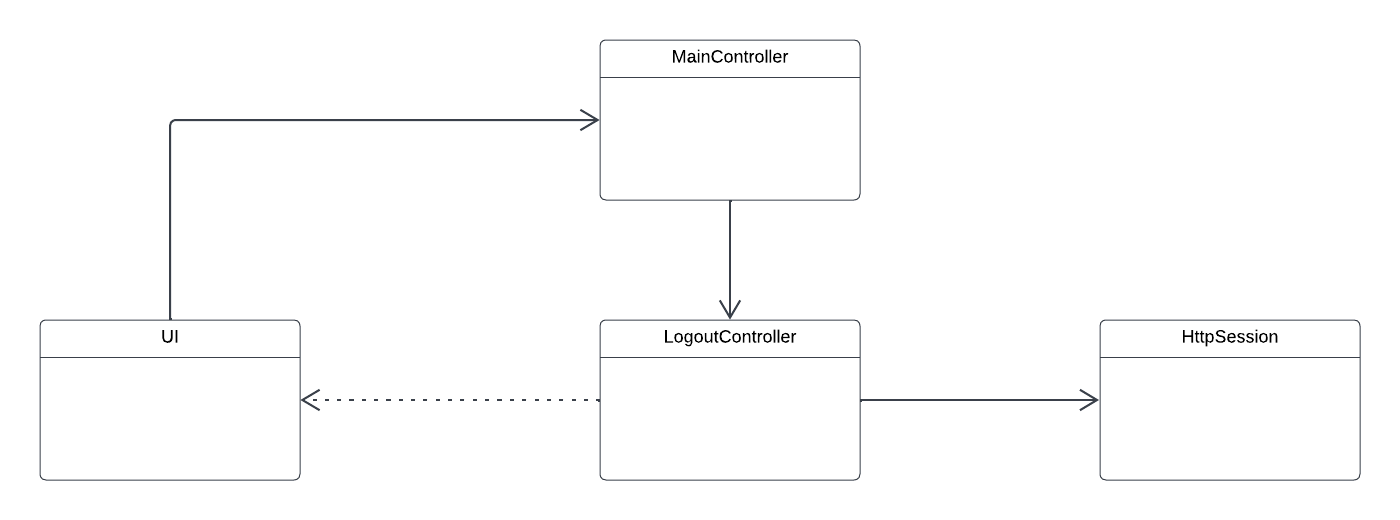
|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | processRequest() | Receive data from UI.  Call function isValidEmail ().  Call function sendOTPEmail(). If success, redirect to Enter OTP Page. If fail, show error and let user input again. |

#### ValidateOTPController Class

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | processRequest() | Receive data from UI.  Compare OTP in input and OTP in mail. If success, redirect to Enter New Password Page. If fail, show error and let user input again. |

## Authentication/Log out

### Class Diagram

****

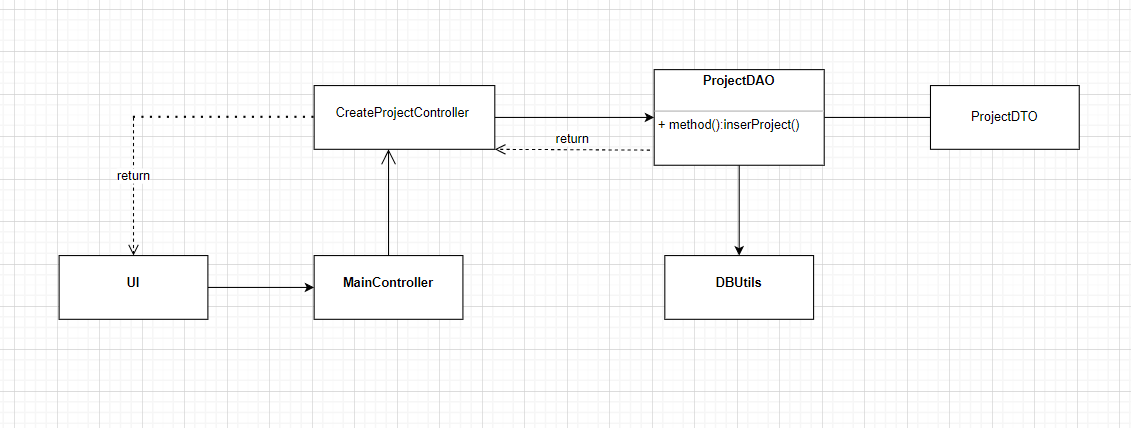
### Class Specifications

#### LogoutController Class

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | processRequest() | Invalidate session that contain user info. Redirect to Login Page |

## Project Management/Create Project

### Class Diagram



### Class Specifications

#### CreateProjectController Class

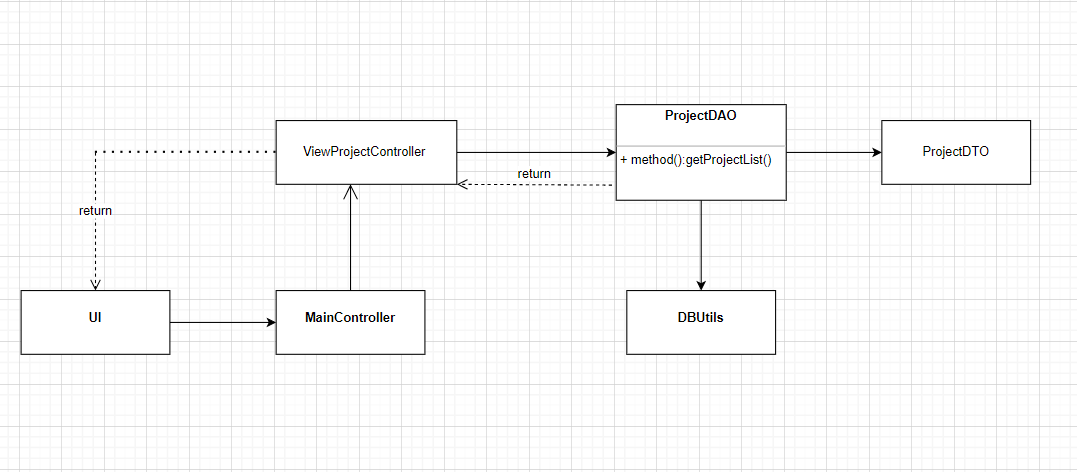
|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | processRequest() | Get data from request session. Call insertProject (). If success, show notification. If fail, show error. |

#### ProjectDAO Class

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | insertProject () | Call database to insert a new account. |

## Project Management/View Project List

### Class Diagram



### Class Specifications

#### ProjectDAO Class

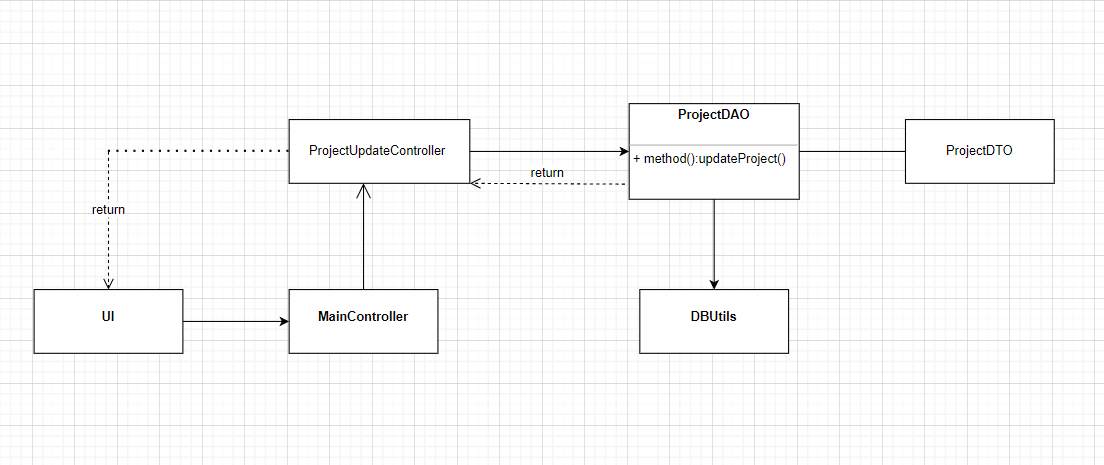
|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | getProjectList() | Users allow to see all the projects . All the information parameters will be sent to the “ViewProjectController” to handle. |

#### ViewProjectController Class

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | doGet() | Dispatcher project list to show. |

## Project Management/Update Project

### Class Diagram



### Class Specifications

#### ProjectDAO Class

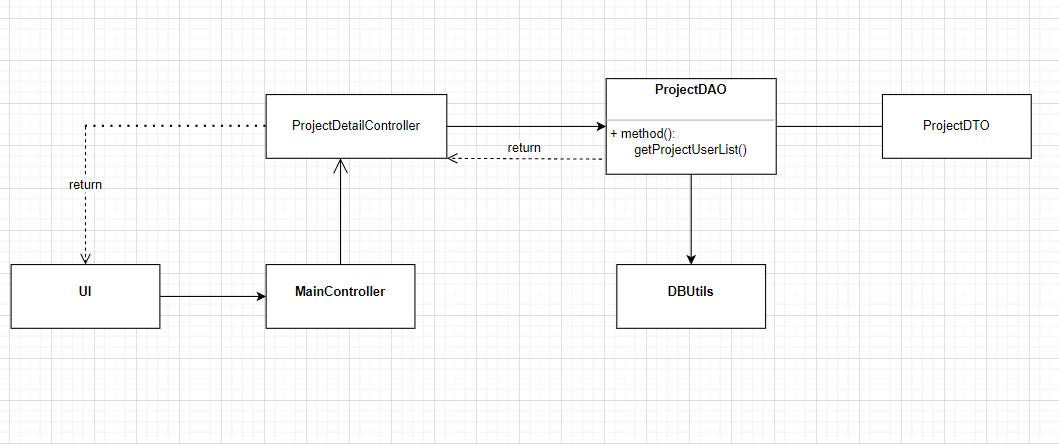
|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | updateProject() | Call database top update selected project. |

#### ProjectUpdateController Class

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | processRequest() |  |

## Project Management/ Project Detail

### Class Diagram



### Class Specifications

#### ProjectDAO Class

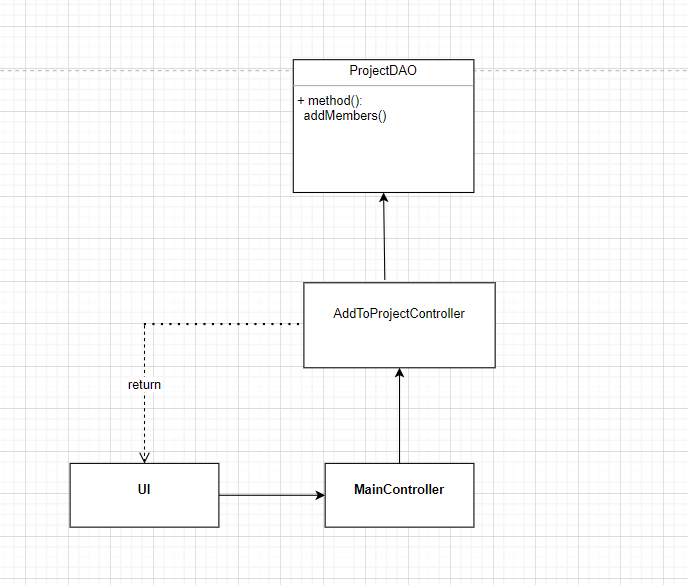
|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | getProjectUserList () | Call database to get project details by user id. |

#### ProjectDetailController Class

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | processRequest() | Get user id from request scope. Call function getProjectUserList (). Redirect to Project Details Page. |

## Project Management/ Add members to project

### Class Diagram



### Class Specifications

#### ProjectDAO Class

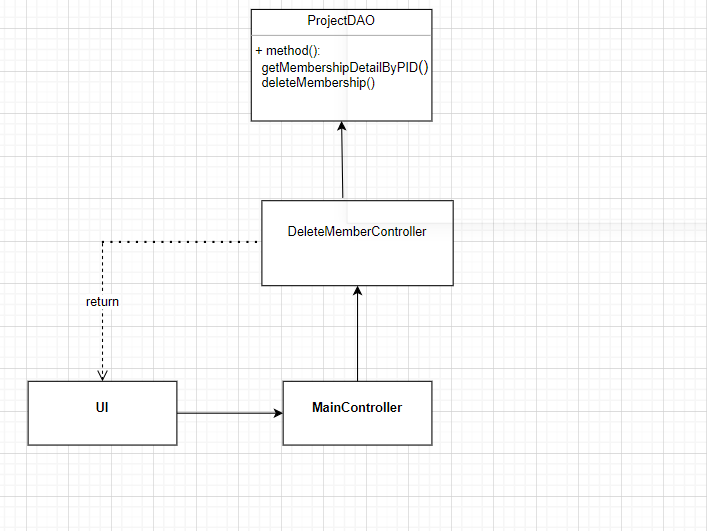
|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | addMembers() | Add member to project. |

#### AddToProjectController Class

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | processRequest() | Call function addMembers(). Redirect to "searchUsers.jsp". |

## Project Management / Disable members in project

### Class Diagram



### Class Specifications

#### ProjectDAO Class

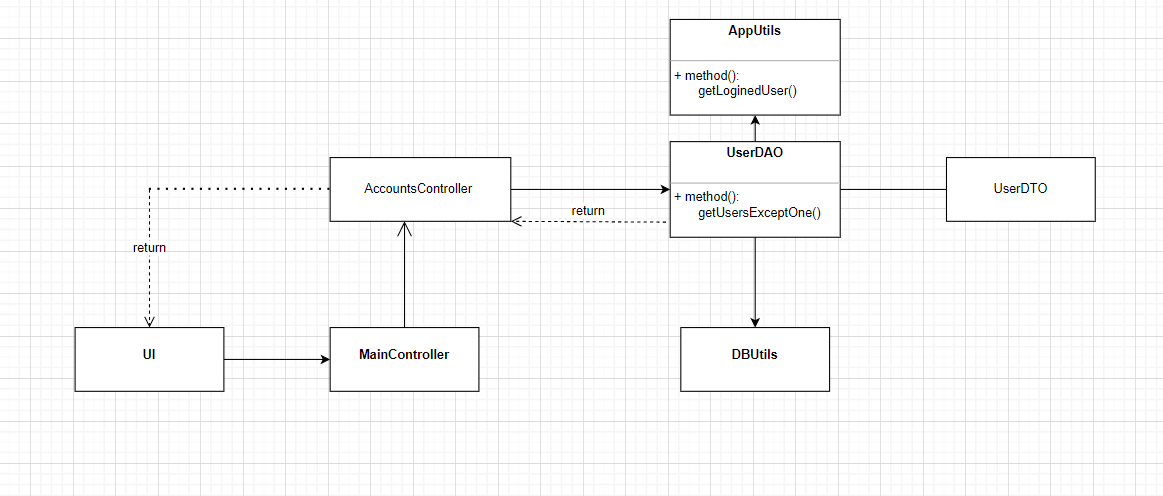
|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | getMembershipDetailByPID() | Get member detail by projectID. |
| 02 | deleteMembership() | Delete member. |

#### DeleteMemberController Class

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | processRequest() | Call function getMembershipDetailByPID() and deleteMembership(). Redirect to "projectDetail.jsp". |

## Account Management / View account list

### Class Diagram



### Class Specifications

#### AccountsController Class

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | processRequest() | Call function getLoginedUser() to get current user in session scope. Then call getUsersExceptOne() to get all user except current user login. |

#### AppUtils Class

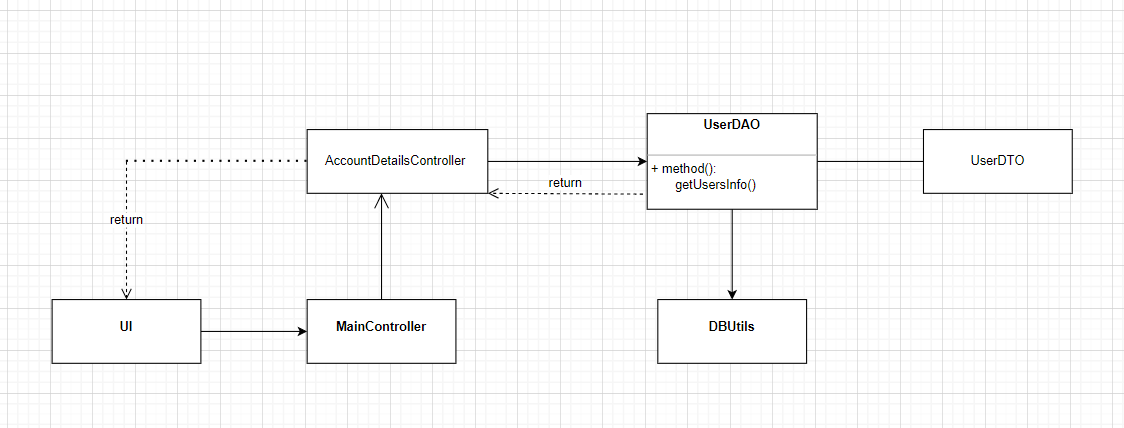
|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | getLoginedUser() | Get current user login. |

#### UserDAO Class

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | getUsersExceptOne () | Call database to get all user except current user login. |

## Account Management / View user profile

### Class Diagram



### Class Specifications

#### AccountDetailsController Class

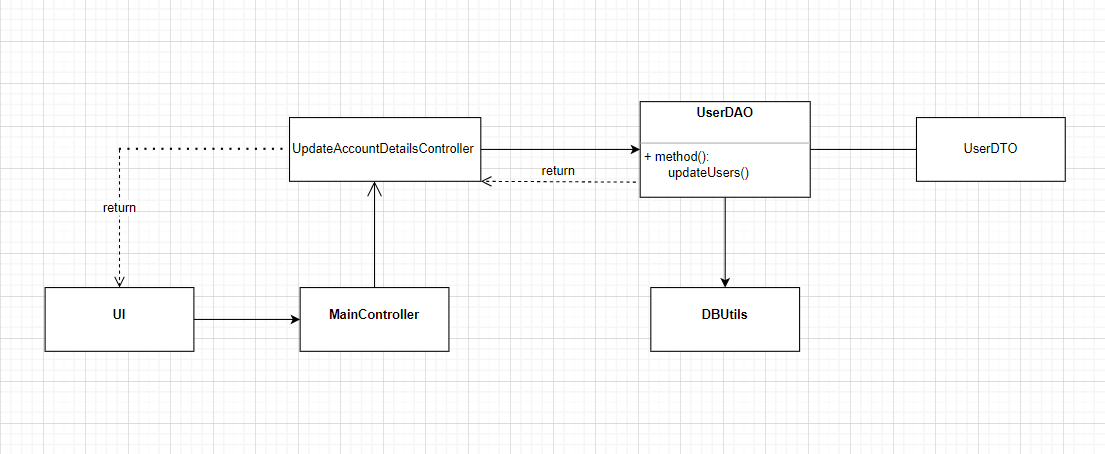
|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | processRequest() | Get user id from request scope. Call function getUserInfo()to get all information of user. Redirect to Edit Account Page. |

#### UserDAO Class

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | getUserInfo () | Call database to get all information of user by user id. |

## Account Management / Update account

### Class Diagram



### Class Specifications

#### UpdateAccountDetailsController Class

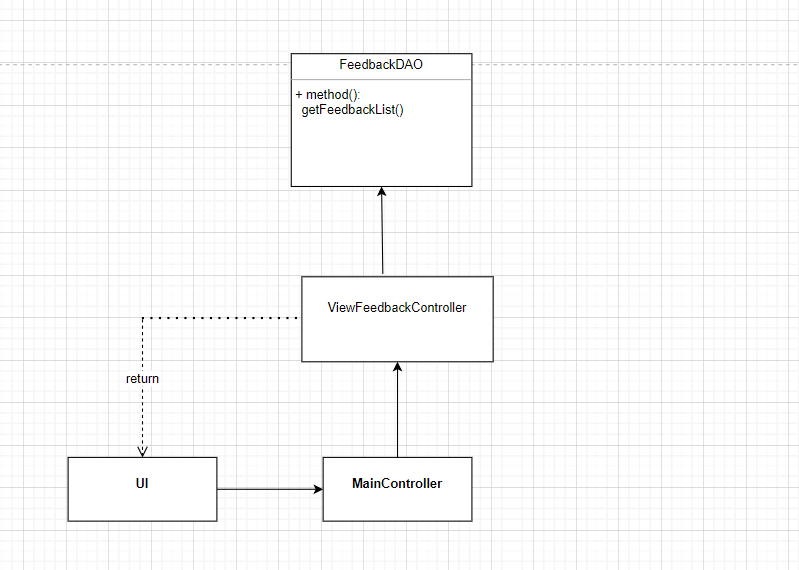
|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | processRequest() | Get data from request scope and check validation. Call function  updateUser() to update all information of user. Show notification if success, show error and let user input again if fail. |

#### UserDAO Class

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | updateUser() | Call database to update all information of user. |

## Account Management / View feedback list with role ‘admin’

### Class Diagram



### Class Specifications

#### ViewFeedbackController Class

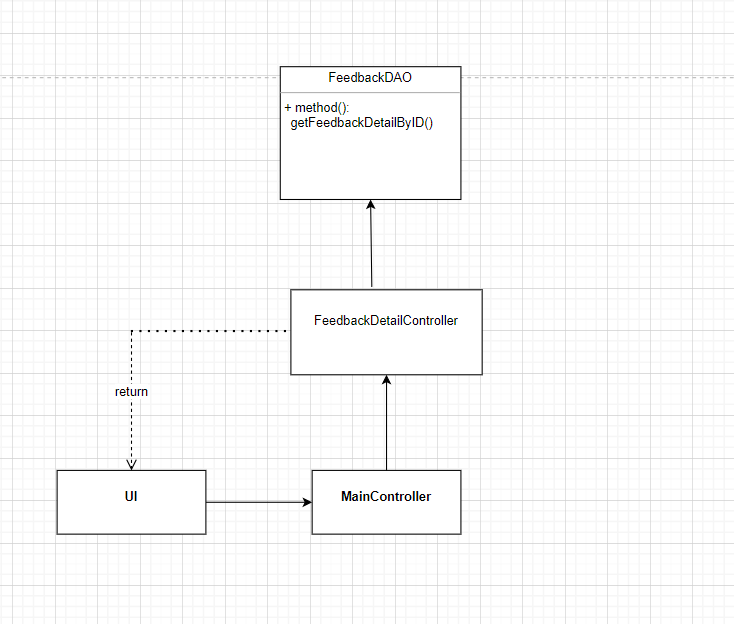
|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | processRequest() | Call getFeedbackList(). Redirect to “viewFeedback.jsp”. |

#### FeedbackDAO Class

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | getFeedbackList() | Get feedback lists. |

## Account Management / View feedback details with role ‘admin’

## Class Diagram



### Class Specifications

#### FeedbackDetailController Class

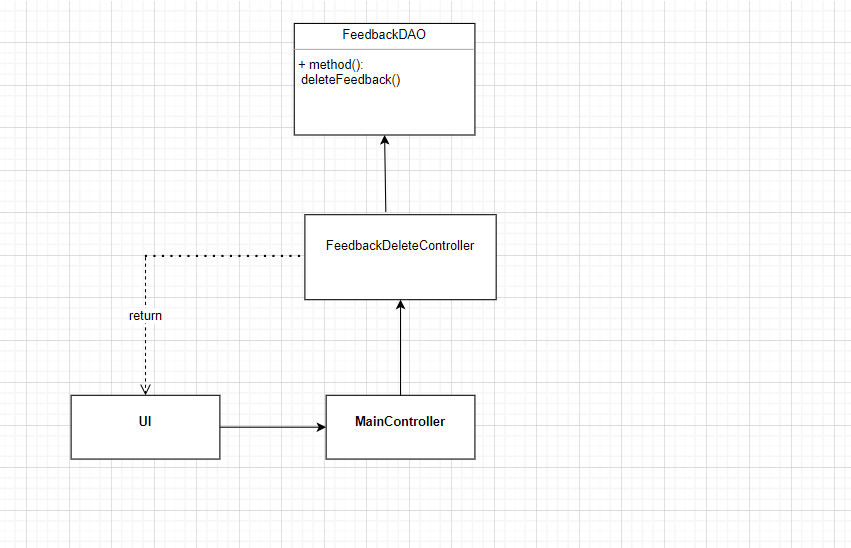
|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | processRequest() | Call getFeedbackDetailByID(). Redirect to “feedbackDetail.jsp”. |

#### FeedbackDAO Class

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | getFeedbackDetailByID() | Get feedback details by feedbackID. |

## Account Management / Delete feedback with role ‘admin’

### Class Diagram



### Class Specifications

#### FeedbackDeleteController Class

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | processRequest() | Call deleteFeedback(). Redirect to “viewFeedback.jsp”. |

#### FeedbackDAO Class

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | deleteFeedback() | Delete a feedback. |