Product Description

# Product Summary

* **Title**:

Document Management System

* **Purpose**:

# Project Team

|  |  |  |
| --- | --- | --- |
| Name / Role | Student ID | Email |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Key Stakeholders

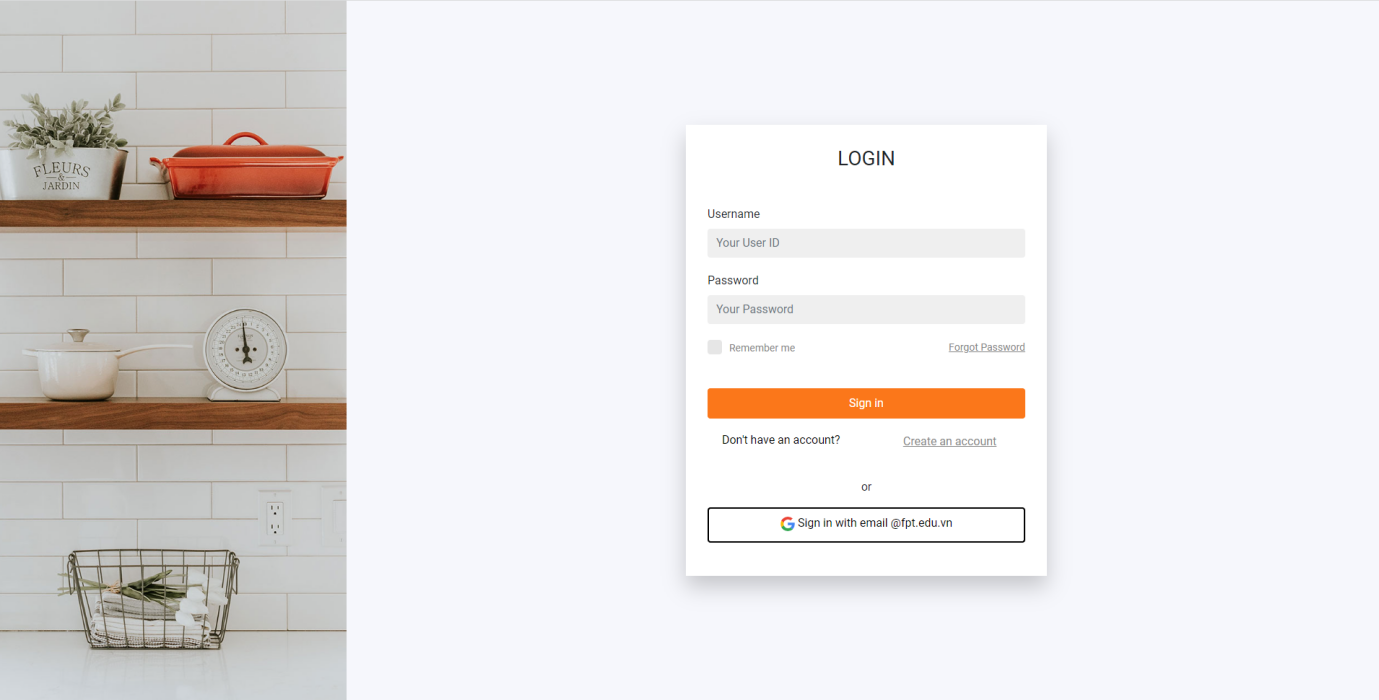
|  |  |
| --- | --- |
| **Stakeholders** | **Impact to the product** |
| Admin | View user’s feedback, view user’s account list, view user’s feedback details, delete feedback |
| Users (project owner) | Create project, add member to project, edit project, view project list, view project details, disable members in project |
| Users (not the owner of the project) | Create project, add member to project, edit project, view project list, view project details |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# Functional Requirements

## Authentication

### Login

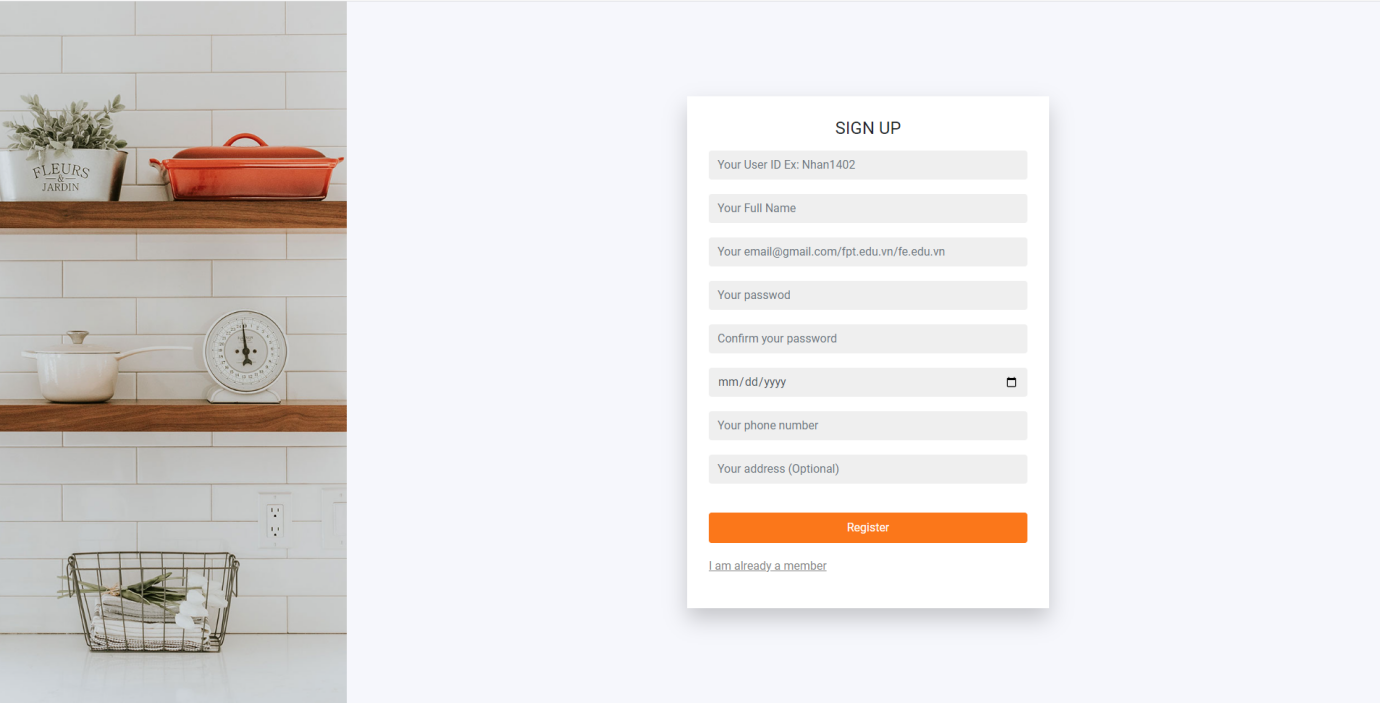
* Function trigger: When user click the button “Sign in” on the “login.jsp” page.
* Function description: Roles: Users/Administrators, allow the to sign into the system that they have registered by sending a request consists of a parameter “action” with value “Sign in” to the “MainController” servlet to process.
* Screen layout:



* Function details: In order to sign in, they must register an account, unless they are an administrator. If they sign in with the “User” role, the system will send them to the dashboard page where they can interact with the documents management system. And an administrator will be sent to “Admin” page in which they can manage all the accounts which had been registered in the system. If the user’s ID or password are invalid, an alert will pop up for notification.

### Registration

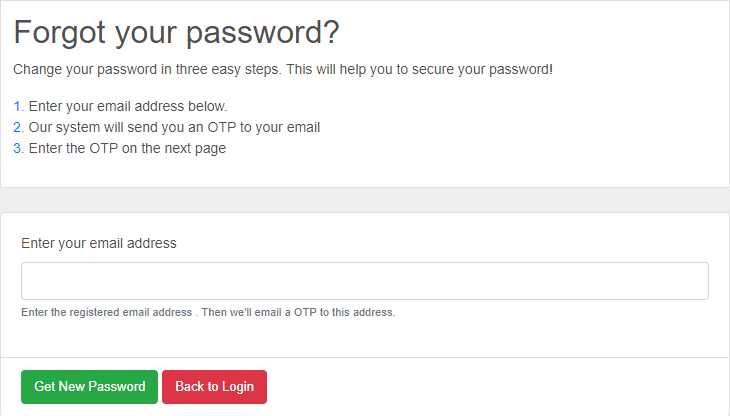
* Function trigger: When user click the “Create an account” link on the “login.jsp” page, the “registration.jsp” page will show up. After that, click the button “Sign up” to complete your service.
* Function description: Guests are allowed to create a new account, they have to fill in registration form with their personal information. All the information parameters and the “action”, value “Register” will be sent to the “MainController” to handle.
* Function layout:



* Function details: In order to sign up, guests have to fill in all their personal information. The system will validate each of the information to make sure they are all appropriate. For example, you can not register an existed user ID and some error messages will be shown on the form. After finishing with no errors, an alert will pop up for notification.

### Forgot Password

* Function trigger: When the users/administrators click the “Forgot your password?” link on the “login.jsp” page. The “forgOTPassword.jsp” will show up.
* Function description: Users/Administrators are allowed to retrieve their password. An OTP code will be sent to the mail you used to register, after checking the OTP, you will need to set a new password, confirm it and click the “Confirm” button.
* Function layout:



* Function details: In order to retrieve you password, you must have an registered account in system. After checking that your email/userID is existed, the system will send to you an email contains an OTP code, if you enter an invalid OTP code, an error message will show up. After that, you need to set up and confirm your new password, if they do not match, an error will show up. Finishing all the steps, your new password is ready to go.

## System Handler

### Email Sending

* Function trigger: When you type in the email in order to retrieve your password, or when you share a document to someone else, an email will be sent to them for notification.
* Function description: It is a non-screen function, it will be automatically handled by our system
* Function details: You have to add some of the java Mail libraries so as to implement it, after that you have to define all the classes and methods. When you forgot your password, an OTP code will be sent to your mail, or you want to share your document to someone else, an email will be sent to them.

## Project Management

### Create new project

* Function trigger: When user click the button “Project” on the “index.jsp” page. , the “projectlist.jsp” page will show up in the “projectlist.jsp” page click the button “create project” to enter “createProject.jsp”page
* Function description: Users are allowed to create a new project, they have to fill in form with their information. All the information parameters and the “action”, value “Create” will be sent to the “CreateProjectController” to handle.
* Function layout:
* Function details: In order to create project, users have to fill in all their information. The system will validate each of the information to make sure they are all appropriate. Error messages will be shown on the form when it occurs. After finishing with no errors, user will return to the project list.

### View project list

* Function trigger: When user click the button “Project” on the “index.jsp” page. , the “projectlist.jsp” page will show up.
* Function description: Users allow to see all the projects . All the information parameters will be sent to the “ViewProjectController” to handle.
* Function layout:
* Function details: In order to view project, users have to create the project first. The system will validate each of the information to make sure they are all appropriate. Error messages will be shown on the form when it occurs. After finishing with no errors, user will return to the project list.

### Update project

* Function trigger: When user click the button “Project” on the “index.jsp” page. , the “projectlist.jsp” page will show up in the “projectlist.jsp” page click the button “update” to enter “updateProject.jsp”page
* Function description: Users allow to edit all the project. All the information parameters will be sent to the “ProjectUpdateController” to handle
* Function layout:
* Function details: In order to update project, users have to create the project first and users have to fill in all their information. The system will validate each of the information to make sure they are all appropriate. Error messages will be shown on the form when it occurs. After finishing with no errors, user will return to the project list.

### Project details

* Function trigger: When users click on the button “Details” on the “projects.jsp” page, the details of a project will be shown.
* Function description: When users click the button “Details”, a parameter “action” with value “projectDetails” will be sent to the “MainController” to handle.
* Function layout:
* Function details: In order to view project details, you have to be a member of it. If not, the system will redirect you back to the “index.jsp” page. Otherwise, all the information of that project will be loaded on to the “projectdetails.jsp”. Error messages will be shown on the page when it occurs.

### Share documents to project:

* Function trigger: When users click on the “Add documents” on the “projects.jsp” page. A drop down with three options will be shown for the user to choose.
* Function description: When the users click the “Create folder” option, it will send the parameter “action” with value “createFoler”, on the “Upload folder” the value will change to “uploadFolder” and “Upload file”, it will be “uploadFile”. All of that will be sent to the “MainController” to handle.
* Function layout:
* Function details: It allow a member of that project to share documents to others to view, others that are not member, will be redirected back to the “index.jsp” page. And all the documents’ permissions will be set to “Edit” mode for all members to work on it.

### Add members to project

* Function trigger: When the users click on the button “Add members” on the “projects.jsp” page.
* Function description: When it happens, the parameter “action” with value “addMember” will be sent to the “MainController” to handle.
* Function layout:
* Function details: Only the manger of that project can add new members, otherwise the button will be invisible, the manager needs to type in the member’s userID or email to proceed.

### Disable members in project

* Function trigger: When the manager click on the button “Remove” in the “projectdetais.jsp” page.
* Function description: It allows the manager to remove any members from the current project. A parameter “action” with value “removeMember” will be sent to the “MainController” to handle.
* Function layout:
* Function details: Only the manager of that project can remove members, otherwise the button will be invisible, errors will be shown on the page when it occurs.

## Account Management:

### View accounts list

* Function trigger: When the administrator click “Accounts” on the side bar.
* Function description: It will show all the accounts that have been registered to our system, a parameter “action” with value “Accounts” will be sent to the “MainController” to handle.
* Function layout:
* Function details: It will load all information of all accounts, including the account whose role is “Admin” onto the “accounts.jsp” page. Error will be shown on the page when it occurs.

### View user profile

* Function trigger: When the users click on the button “Profile” on “index.jsp” page.
* Function description: It allow users to see if the information they registered with is correct or not. If not, they can change it. A parameter “action” with value “Profile” will be sent to the “MainController” to handle.
* Function layout:
* Function details: It will load the information of a user and show it on the “profile.jsp” page, users will check if the information is accurate and they can continue to update it. Errors will be shown on the page when it occurs.

### Search accounts

* Function trigger: When admin type in the information they want to search in the search bar on the “accounts.jsp” page.
* Function description: It allows admin to search for the the user which contains the parameter value “search”, and it will be sent to the “MainController” to handle.
* Function layout:
* Function details: Administrator can type in the information of an account they want to search. All the accounts that match the parameter will be shown on the “accounts.jsp” page. Errors will be shown on the page when it occurs.

### Update account

* Function trigger: When users click on the button “Update” on the “profile.jsp” page.
* Function description: It allows user to change their information if it is incorrect. A parameter “action” with value “Update” will be sent to “MainController” to handle.
* Function layout:
* Function details: All the information the users change will be sent to the “UpdateUserProfileController”, if it meets the validated requirement, the “profile.jsp” will be shown for user to check again. If not, errors will be shown to the incorrect input.

### Create administrator

* Function trigger:
* Function description:
* Function layout:
* Function details:

### Send feedback

* Function trigger: When users click on the button “Feedback” on the “index.jsp” page.
* Function description: It allows users to send feedback they want the app to improve on. A parameter “action” with value “Feedback” will be sent to “MainController” to handle.
* Function layout:
* Function details: The feedback sent will be handled by administrator, when admin click the button “Received”, a notification will be sent to the user who sent it, content and type of feedback will be sent to “FeedbackController” to handle. Errors will be shown when it occurs.

### Disable account

* Function trigger: When admin click on the button “Remove” on the “accounts.jsp” page.
* Function description: It will disable the account that admin decided on, when the user try to log in with it, a alert will show up says “Your account has been disabled. Please contact us for further notice”. A parameter “action” with value “Delete” will be sent to the “MainController” to handle .
* Function layout:

Then:

* Function details: It will completely delete the account from the system, we just simply disable if you violate some of our policies, you can always contact the admin to enable it again. All of the requests will be proceed by the “DeleteAccountController”. Errors will be shown when it occurs.

# Non-Functional Requirements