

**CAPSTONE PROJECT REPORT**

**Report 3 – Software Requirement Specification**

– Hanoi, November 2023 –

**Table of Contents**

[I. Project Report 3](#_heading=h.gjdgxs)

[1. Status Report 3](#_heading=h.30j0zll)

[2. Team Involvements 3](#_heading=h.1fob9te)

[3. Issues/Suggestions 3](#_heading=h.3znysh7)

[II. Software Requirement Specification 4](#_heading=h.2et92p0)

[1. Overall Description 4](#_heading=h.tyjcwt)

[1.1 Product Overview 4](#_heading=h.3dy6vkm)

[1.2 Business Rules 4](#_heading=h.1t3h5sf)

[2. User Requirements 8](#_heading=h.4d34og8)

[2.1 Overview 8](#_heading=h.2s8eyo1)

[2.2.1 UC-01\_Register 14](#_heading=h.17dp8vu)

[2.2.2 UC-02\_Login 15](#_heading=h.lnxbz9)

[2.2.3 UC-03\_Reset password 16](#_heading=h.1ksv4uv)

[2.2.4 UC-04\_Change password 17](#_heading=h.9ws7shiqiwk)

[2.2.5 UC-05\_Update Subject 18](#_heading=h.zgzx30k7ovxf)

[2.2.5 UC-06\_Submit assignment 20](#_heading=h.tyk0hsdnh1xh)

[2.2.1 UC-07\_View user list 20](#_heading=h.2x23h7dter1e)

[2.2.2 UC-08\_Add new user 21](#_heading=h.au9eiv9wwcbq)

[3. Functional Requirements 22](#_heading=h.44sinio)

[3.1 System Functional Overview 22](#_heading=h.2jxsxqh)

[3.2 Authentication 26](#_heading=h.z337ya)

[3.3 Admin Feature 30](#_heading=h.3j2qqm3)

[4. Non-Functional Requirements 38](#_heading=h.1y810tw)

[4.1 External Interfaces 38](#_heading=h.4i7ojhp)

[4.2 Quality Attributes 38](#_heading=h.qsh70q)

# I. Project Report

## 1. Status Report

| **#** | **Work Item** | **Status** | **Notes (Work Item in Details)** |
| --- | --- | --- | --- |
| 1 |  | Pending |  |
| 2 |  | In Progress |  |
| 3 |  | Completed |  |

## 2. Team Involvements

| **#** | **Task** | **Member** | **Notes (Task Details, etc.)** |
| --- | --- | --- | --- |
| 1 |  | LongPV |  |
| 2 |  | DungHM |  |
| 3 |  | ChienDQ |  |
| 4 |  | DucTA |  |
| 5 |  | HaiTTT |  |

## 3. Issues/Suggestions

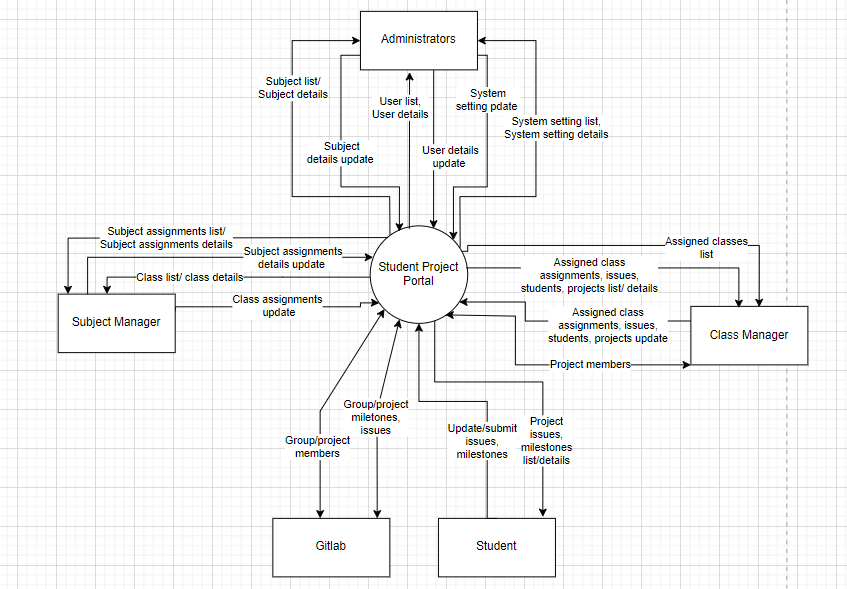
| **#** | **Issue** | **Status** | **Notes (Solution, Suggestion, etc.)** |
| --- | --- | --- | --- |
| 1 |  | Pending |  |
| 2 |  | In Progress |  |
| 3 |  | Completed |  |

# II. Software Requirement Specification

## 1. Overall Description

### 1.1 Product Overview

Student Project Portal Management will provide a new way of managing student's projects. Faculty only need to access the system; they can see all projects they take responsibility for and manage project progress through the system directly. They don't need to use email, saving them time and increasing their working efficiency in the same period. Students now only join the system to see all the tasks they need to complete and can quickly catch any change or requests, which will help with missing assignments or change requests like before.



>>

### 1.2 Business Rules

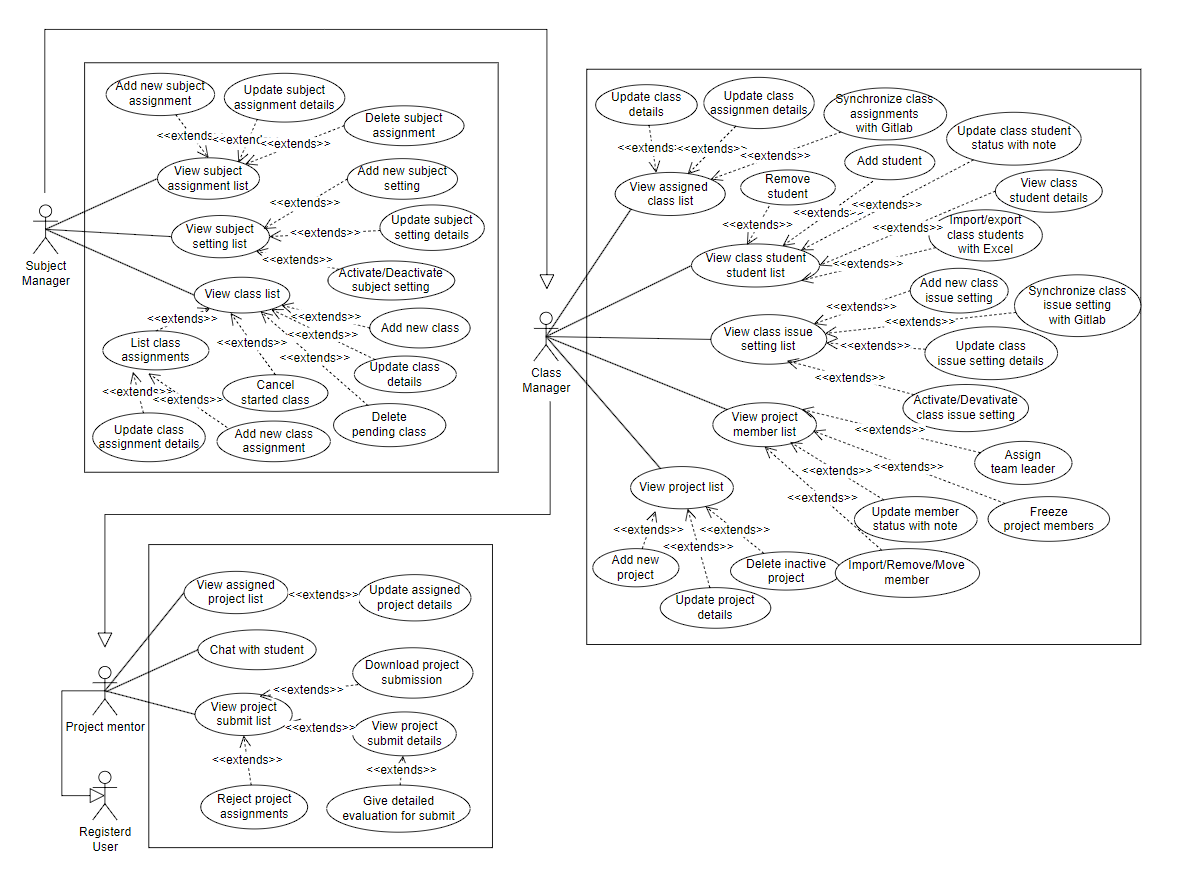
| **ID** | **Rule Definition** |
| --- | --- |
| BR-01 | User Registration and Authentication Users must provide a valid email address or mobile number for registration. |
| BR-02 | Email addresses must be verified by sending a verification code. |
| BR-03 | Only users with verified email addresses or mobile numbers can log in. |
| BR-04 | Users can log in using email, mobile, or Google login (for specific domains) |
| BR-05 | User Profile Users can view and edit their profiles, including full name, email, mobile, and avatar image. |
| BR-06 | Changing email or mobile requires verification. |
| BR-07 | Users can view their assigned user roles. |
| BR-08 | Users can reset their passwords if they have forgotten. |
| BR-09 | Password Management: Users can change their passwords. |
| BR-10 | Dashboard After logging in, users are directed to a dashboard. |
| BR-11 | The dashboard displays statistics and information related to their assigned projects |
| BR-12 | Student Students can manage project issues (list, add, update, batch update, import/export). |
| BR-13 | Customised issue statuses are supported, including a closing status. |
| BR-14 | Requirement issues can only be closed when linked to at least one non-requirement issue, and all linked issues are closed. |
| BR-15 | Project issues can be synchronised with GitLab. |
| BR-16 | Students can manage work updates (list, details, add, update). |
| BR-17 | BR-17 Students can view details of their team assignments, including submission information. |
| BR-18 | If eligible, students can submit or resubmit assignments. |
| BR-19 | Team Leader Team leaders can manage project milestones (list, add, update, delete pending). |
| BR-20 | Team leaders can manage project issue settings (types, statuses, work processes). |
| BR-21 | Team leaders can submit or resubmit project assignment results, including attachments and package descriptions. |
| BR-22 | Project Mentor Project mentors can manage assigned projects, including team leader changes (for both active and inactive projects). |
| BR-23 | Project mentors can evaluate project team assignments, providing grades and comments. |
| BR-24 | They can also evaluate the complexity and quality of work results. |
| BR-25 | In the case of the final assignment, they must re-evaluate submitted updated work. |
| BR-26 | Class Manager Class managers can manage assigned classes, including synchronisation settings. |
| BR-27 | They can update class assignments and synchronise them with GitLab milestones. |
| BR-28 | Class managers can manage class issue settings (types, statuses, work processes) |
| BR-29 | Class managers can manage class students, including import and export. |
| BR-30 | Class managers can manage projects within their classes. |
| BR-31 | Class managers can adjust project members, create new projects, and designate team leaders. |
| BR-32 | Subject Manager Subject managers can manage subject assignments. |
| BR-33 | Subject managers can set up subject-specific settings (work complexity levels, workloads, and quality levels). |
| BR-34 | Subject managers can manage classes, including adding new classes and defining assignments. |
| BR-35 | Administrator - Manage Subjects: Administrators can manage subjects, including activation/deactivation. |
| BR-36 | Administrators can manage users, including blocking/unblocking verified users. |
| BR-37 | Administrators can configure system settings (semesters, user roles, permitted email domains). |
| BR-38 | Use Google Firebase for mobile number verification via SMS. |
| BR-39 | User’s password must be encoded |

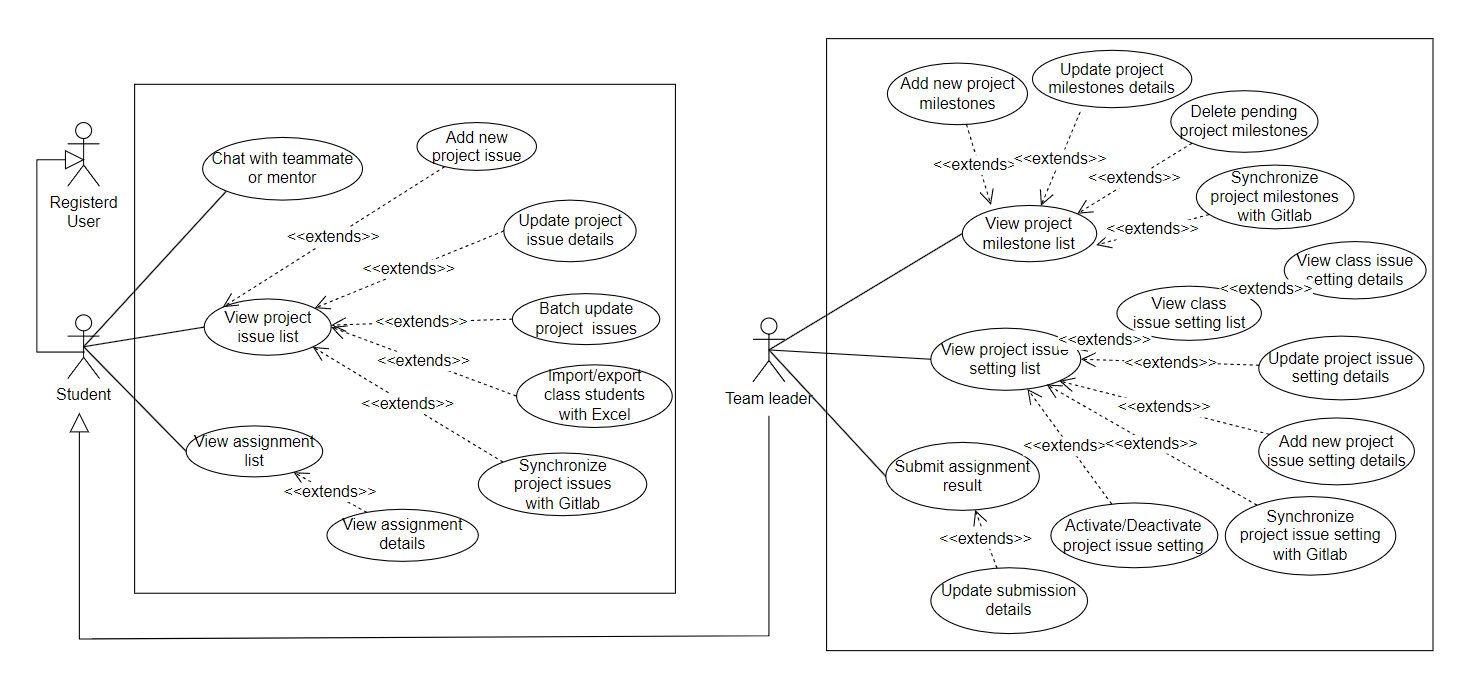
## 2. User Requirements

***(This is optional part)***

### 2.1 Overview

#### a. Use Case Diagram





#### b. System Actors

### 

| **#** | **Actor** | **Description** |
| --- | --- | --- |
| 1 | Guest | Non-registered visitor |
| 2 | Student | Enrolled learner in an educational program. |
| 3 | Team Leader | Student leading a project group. |
| 4 | Project Mentor | Experienced guide for student projects. |
| 5 | Class Manager | A person responsible for managing a class |
| 6 | Subject Manager | An individual who manages the curriculum and resources related to a specific academic subject |
| 7 | Admin | A person who manages and oversees administrative tasks to keep things running smoothly in an organization or system. |

#### c. Use Cases List

#### 

| **#** | **Feature** | **Use case** | **Use case description** |
| --- | --- | --- | --- |
| 1 | Common | View page landing | Users can access the landing page of the application, providing an initial view and introduction to the system. |
| 2 | Common | Register an account | New users can register for an account by providing required information, such as username, password, email, and personal details. Verify mail/phone |
| 3 | Common | Login to the system | Users can log in to the system by providing their username and password or via Google login page, as long as the input email's domain is listed in the predefined domain list |
| 4 | Common | Reset password | Users can reset their password in case they forget their password |
| 5 | Common | Change password | Users have the option to change their account passwords for security reasons. |
| 6 | Common | Logout | User can logs out of the system |
| 7 | Common | View/Edit profile | Users can view their own user profile within the system, allowing them to see and edit their personal information and settings. |
| 8 | Admin | View user list | Displays a list of all users |
| 9 | Admin | Add new user | Add an user with username and password |
| 10 | Admin | Update user details | Update information of an user (display name, username, ...) |
| 11 | Admin | Activate/Deactivate user | Activate or deactivate for an user to prevent or allow users to use the system |
| 12 | Admin | View subject list | Displays a list of all subjects |
| 13 | Admin | Add new subject | Add a new subject to system |
| 14 | Admin | Update subject details | Activate or deactivate for a subject |
| 15 | Admin | Activate/Deactivate subject | Update details, documents,... of a subject |
| 16 | Admin | View setting list | Displays a list of all settings |
| 17 | Admin | Add new setting | Add a new setting to system |
| 18 | Admin | Update setting details | Update details of a setting |
| 19 | Admin | Activate/deactivate setting | Activate or deactivate a setting |
| 20 | Subject Manager | View subject setting list | Displays a list of all subject setting |
| 21 | Subject Manager | Activate/Deactivate subject setting | Activate or deactivate a subject setting |
| 22 | Subject Manager | Update subject setting details | Update details of a subject setting |
| 23 | Subject Manager | Add new subject setting | Add a new subject setting to system |
| 24 | Subject Manager | View subject assignment list | Displays a list of all subject assignments |
| 25 | Subject Manager | Add new subject assignment | Add a new subject assignment to system |
| 26 | Subject Manager | Update subject assignment details | Update details of a subject assignment |
| 27 | Subject Manager | Delete subject assignment | Delete a subject assignment |
| 28 | Subject Manager | View class list | Displays a list of all classes that learn this subject |
| 29 | Subject Manager | Add new class | Add a new class that learn this subject |
| 30 | Subject Manager | Update class details | Update details of a class |
| 31 | Subject Manager | Cancel started class | Cancel a class that has started |
| 32 | Subject Manager | Delete pending class | delete a class that is pending |
| 33 | Subject Manager | View class  assignment list | Displays a list of all assignments of a class that learn this subject |
| 34 | Subject Manager | Add new class  assignment | Setup assignments for a class that learn this subject |
| 35 | Subject Manager | Update class  assignment details | Update assignments of subject for all class learn this subject |
| 36 | Class Manager | View assigned  class list | Displays a list of all classes that the class manager manages |
| 37 | Class Manager | Update class details | Update information of a class |
| 38 | Class Manager | Synchronise class information  with Gitlab | Synchronise information of a class from teacher Gitlab's group |
| 39 | Class Manager | View class assignment list | Displays a list of all assignments of a class |
| 40 | Class Manager | Update class assignment details | Update assignments for a class that learn this subject |
| 41 | Class Manager | Synchronise class assignments  with Gitlab | Synchronise assignments of a class from Gitlab |
| 42 | Class Manager | View class student list | Displays a list of all student of class |
| 43 | Class Manager | View class student details | View details of a student of the class |
| 44 | Class Manager | Add student | Add to a project |
| 45 | Class Manager | Remove student | Remove student from a project |
| 46 | Class Manager | Update class student status with note | Update status, note of a student in class |
| 47 | Class Manager | Import/export class students  with Excel | Import or export student list with Excel |
| 48 | Class Manager | View class issue setting list | Displays a list of all issues of class |
| 49 | Class Manager | Add new class issue setting | Add a new issue setting of a class |
| 50 | Class Manager | Update class  issue setting details | Update issue setting of a class |
| 51 | Class Manager | Activate/Deactivate class issue setting | Update status of issue setting in a class |
| 52 | Class Manager | Synchronise class issue settings  with Gitlab | Synchronise class issue setting with Gitlab |
| 53 | Class Manager | View project list | Display all project in class |
| 54 | Class Manager | Add new project | Create a new project for class |
| 55 | Class Manager | Update project details | Update details of a project |
| 56 | Class Manager | Delete inactive project | Delete project that is inactive |
| 57 | Class Manager | View project member list | Display all member in project |
| 58 | Class Manager | Assign  team leader | Select a leader for project |
| 59 | Class Manager | Freeze  project members | Freeze a member in a project group |
| 60 | Class Manager | Update member  status with note | Update status, note of a student in class |
| 61 | Class Manager | Import/Remove/Move member | Change member in project or import list member of project |
| 61 | Project Mentor | View assigned project list | View list assigned project |
| 62 | Project Mentor | Update assigned project details | Update assigned project details |
| 63 | Project Mentor | View project submission list | View list project submission |
| 64 | Project Mentor | Download project submission | Download project submission |
| 65 | Project Mentor | View project submit details | View project submission details |
| 66 | Project Mentor | Reject project assignments | Reject a project assignment |
| 67 | Project Mentor | Give detailed evaluation for submit | Comment and evaluate submission |
| 68 | Team Leader | View project milestone list | View list project milestones |
| 69 | Team Leader | Add new project milestones | Add new project milestones |
| 70 | Team Leader | Update project milestones details | Update project milestones details |
| 71 | Team Leader | Delete pending project milestones | Pending delete request for project milestone |
| 72 | Team Leader | Synchronise project milestones with Gitlab | Synchronise project milestones with the Gitlab project milestones |
| 73 | Team Leader | View class issue setting list | View list class issue settings |
| 74 | Team Leader | View class issue setting details | View and Update class issue setting details |
| 75 | Team Leader | View project issue setting list | View list project issue settings |
| 76 | Team Leader | Update project issue setting details | Update current project issue setting details |
| 77 | Team Leader | Add new project issue setting details | Add new project issue setting details |
| 78 | Team Leader | Activate/Deactivate project issue setting | Activate/Deactivate project issue setting |
| 79 | Team Leader | Synchronise project issue setting with Gitlab | Synchronise project setting in gitlab |
| 80 | Team Leader | Submit assignment result | View subject assignment result |
| 81 | Team Leader | Update submission details | Update submission details |
| 82 | Student | View project issue list | View list project issues |
| 83 | Student | Add new  project issue | Add new project issue |
| 84 | Student | Update project issue details | Update project issue details |
| 85 | Student | Batch update  project issues | Update batch project issues |
| 86 | Student | Import/export class students with Excel | Import / Export list student or class in excel |
| 87 | Student | Synchronise project issue with Gitlab | Synchronise project issue with the Gitlab project issues |
| 88 | Student | View work list | View list to do work |
| 89 | Student | Add new work | Add new work in the to do list |
| 90 | Student | Update work details | Update work details in the to do list |
| 91 | Student | Delete works | Delete work in to do list |
| 92 | Student | View assignment list | View student's assignment |
| 93 | Student | View assignment details | View student's assignment details |
| 94 | Student | Chat with teammate or mentor | Allows users to exchange messages, seek assistance, share information, and collaborate effectively |

### 2.2.1 UC-01\_Register

#### a. Functionalities

| UC ID and Name: | **UC-1\_Register** | | |
| --- | --- | --- | --- |
| Created By: | LongPV | Date Created: | 28/Oct/2023 |
| Primary Actor: | Amin, Subject manager, Class Manager, Team Leader, Student, Guest | Secondary Actors: | None |
| Trigger: | Click a button Register | | |
| Description: | Signup for a new account | | |
| Preconditions: | 1. PRE-1. The user email/phone does not register | | |
| Postconditions: | 1. POST-1. The user has an account to login into system | | |
| Normal Flow: | 1. **2.0 Register an account** 2. 1. User selects the account registration function on the main screen. 3. 2. The system displays a text box allowing the user to create an 4. fullname, username, password. 5. 3. User fills in the account name and password he wants to create. 6. **4. The system checks the characters entered in the username and** 7. **password fields.** 8. 5. User selects “Register”. 9. **6. The system checks the account name and password with the** 10. **data on the system. If it exists, go to the registration page..** 11. 7. The system checks the email domain with the data on the system. If it does not exist, add the account and password to the system. 12. 8.1 The user receives the OTP from the inputted phone. 13. 8.2 The user receives a link to verify from inputted email. 14. 9.1 The web moves to enter the OTP screen. 15. 9.2 User clicks the links then redirects to his/her dashboard. 16. 10. The user enters the OTP. 17. 11. The user clicks the “Confirm” button. 18. 12 The system moves to his/her dashboard. | | |
| Alternative Flows: | **2.1 Register**  **4. Username containing special characters**  1. The system displays the message "Invalid username or password"  below the text box of the registered name.  2. User re-enters the username.  3. Go back to step 4.  **2.2 Register**  **6. The registered name matches the data on the system**  1. The system displays the message "Registered name is already in  use".  2. User re-enters the registered name and password.  3. Go back to step 6.  7. The system checks the email domain with the data on the system. If it does not exist, add the account and password to the system.  1. The system displays the message "Email domain is not allowed".  2. User re-enters the registered name and password.  3. Go back to step 6. | | |
| Exceptions: | N/A | | |
| Priority: | High, Must Have | | |
| Frequency of Use: | N/A | | |
| Business Rules: | FR1, FR2 | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

#### b. Business Rules

| **ID** | **Business Rule** | **Business Rule Description** |
| --- | --- | --- |
| FR1 | Password Encoding | User’s password must be encoded |
| FR2 | Authenticate email or phone number | User account will not active if   * ·    His/her already not authenticate |

### 2.2.2 UC-02\_Login

#### a. Functionalities

| UC ID and Name: | **UC-02\_Login to the system** | | |
| --- | --- | --- | --- |
| Created By: | LongPV | Date Created: | 28/Oct/2023 |
| Primary Actor: | Amin, Subject manager, Class Manager, Team Leader, Student | Secondary Actors: | None |
| Trigger: | User clicks Login button | | |
| Description: | As a user, I want to be able to log into the system so that I can use the system’s authenticated features and access my personalized account. | | |
| Preconditions: | PRE-1. User account has been created & authorised | | |
| Postconditions: | POST-1. User logs in the system successfully | | |
| Normal Flow | **3.0 Login to the system**  1. User accesses the User Login screen  2. User types in the login details or choose other login options (see 3.1)  3. User clicks the Login button  4. System validates the login details (see 3.0.E1)  5. System allows user to access  6. System tracks user’s success login to the Activity Log  7. System accesses the Dash Board | | |
| Alternative Flows: | ***3.1 Google Login***  1. User chooses to login system using Google account  2. System redirects the user to the Google’s Login screen  3. User types in the Google account details and chooses to login  4. Google validates user’s login information successfully and redirect him/her back to the system  5. Return to step 5 of normal flow. | | |
| Exceptions: | ***3.0.E1 System can’t authenticate the user***  1. The Error Message screen is shown to the user  2. User cancels the logging in *=> UC stops, change to UC-1\_View Landing Page*  3. User clicks “Forgot Password?” link *=> change to UC-4\_Reset Password*  4. User clicks “Register” link *=> change to UC-2\_Register* | | |
| Priority: | Must Have | | |
| Frequency of Use: | High | | |
| Business Rules: | FR1, FR2 | | |
| Other Information: | 1. N/A | | |
| Assumptions: | N/A | | |

#### b. Business Rules

| **ID** | **Business Rule** | **Business Rule Description** |
| --- | --- | --- |
| FR1 | Password Encoding | User’s password must be encoded |
| FR2 | Invalid Logging In | User can’t be authenticated to login the system if below cases   * His/her logging-in details are incorrect * His/her account has not been verified * His/her account has been locked or blocked |

### 

### 2.2.3 UC-03\_Reset password

| UC ID and Name: | **UC-3\_Reset password** | | |
| --- | --- | --- | --- |
| Created By: | DungHM | Date Created: | 28/Oct/2023 |
| Primary Actor: | Registered User | Secondary Actors: | None |
| Trigger: | When users click on the “Forgot Password” button in the Login screen | | |
| Description: | The user input his/her email to receive OTP code. This code is sent to the user's email for his/her password resetting. | | |
| Preconditions: | PRE-1. The user account is already existed  PRE-2. The user account is active | | |
| Postconditions: | POST-1. The user can log in with the new password | | |
| Normal Flow: | **3.0 Reset password**  1. The user clicks “Forgot password?”  2. The user enters an email/phone.  3. The user clicks the “Continue” button.  4. The system checks the email that the user entered.  **5. The user receives the OTP from inputted email/phone.**  6. The web moves to enter the OTP screen.  **7. The user enters the OTP.**  8. The user clicks the “Confirm” button.  9. The system checks the input OTP that the user entered.  10. The web moves to reset the password screen.  11. The user enters the new password and confirms the new password, then clicks the “Save” button.  12. The system checks the valid password that the user entered.  15. When the user clicks OK, the dialog is closed and the web returns to the home page | | |
| Alternative Flows: | **3.1 Reset password**  **5. User did not receive the confirmation code**  1. The system displays the message “Confirmation code sent”.  2. User re-enters the confirmation code.  3. Go back to step 6.  **3.2 Reset password**  **7. User entered wrong confirmation code**  1. The system displays the message "Invalid confirmation code".  2. User re-enters the confirmation code.  3. Go back to step 8. | | |
| Exceptions: | N/A | | |
| Priority: | High, Must Have | | |
| Frequency of Use: | Low | | |
| Business Rules: | BR - 08, BR - 39 | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

### 2.2.4 UC-04\_Change password

| **Use Case Specification/Description** | | | |
| --- | --- | --- | --- |
| **Use Case ID:** | **UC-4\_Change password** | | |
| **Use Case Name:** | Edit Profile | | |
| **Created By:** | Đàm Quang Chiến | **Date Created:** | **11/09/2023** |
| **Primary Actor:** | User | **Secondary Actors:** |  |
| **Trigger:** | The user indicates that he/she wants to change their password account. | | |
| **Description:** | This use case allows users to change account passwords. | | |
| **Preconditions:** | - Users must have their account on the website. | | |
| **Post–conditions:** | - Display a screen where users can change their account password. | | |
| **Normal Flow:** | 1. The user logs in to the website.  2. The user clicks on the profile icon and chooses the view profile.  3. The system displays the change password buttons in the view profile screen. | | |
| **Alternative Flows:** |  | | |
| **Exceptions:** | -If users don't have an account they can't use this function. | | |
| **Priority:** | High | | |
| **Frequency of Use:** | 10 transactions/day | | |
| **Business Rules:** | BR-09 | | |
| **Other Information:** |  | | |
| **Assumptions:** |  | | |

### 2.2.5 UC-05\_Update Subject

| **Use Case Specification/Description** | | | |
| --- | --- | --- | --- |
| **Use Case ID:** | **UC-5\_Update subject** | | |
| **Use Case Name:** | Update subject | | |
| **Created By:** | Đàm Quang Chiến | **Date Created:** | 11/09/2023 |
| **Primary Actor:** | Admin, | **Secondary Actors:** | Lecturer |
| **Trigger:** | When admin accesses managed subjects. When accessing the list of all subjects, sections such as adding, editing, deleting, etc. will appear. admin navigates to the "Update subject" section. | | |
| **Description:** | This use case describes the process by which the administrator accesses the subject list and can update information related to the subject such as: subject name, course code, detailed information, lecturer, number of course hours. study,... Administrators need to confirm their correct information to use this function. | | |
| **Preconditions:** | +The system is working and accessible.  +The admin is authenticated and has admin level permissions.  +Data related to subjects is available and up-to-date. | | |
| **Post–conditions:** | + The admin has successfully updated the selection subject.  + Any changes made to subjects are accurately reflected in the list.  + Security measures and access controls are enforced to protect subject data. | | |
| **Normal flow:** | + The admin logs into the system using their information  + The admin is presented with the main dashboard or landing page upon successful authentication.  + The admin navigates to the section or module of the application where subject management is available. This could be an "Admin Dashboard" or "Subjects" section.  + The system displays the "Update subject" screen, showing a list of information an admin needs to edit including: subject name, subject ID, detailed information, lecturer, time number of subjects,... then the admin presses submit button.  + The admin can also perform actions on subjects from this view, such as editing subject details, deleting subjects, or adding new subjects. | | |
| **Alternative Flows:** |  | | |
| **Exceptions:** | + If the technical system or data is not available, the administrator cannot update the subject | | |
| **Priority:** | + High (The necessary feature) | | |
| **Business Rules:** | BR-35 | | |
| **Other Information:** | = | | |
| **Assumptions:** |  | | |

### 2.2.5 UC-06\_Submit assignment

#### 

| UC ID and Name: | **UC-6\_Submit assignment** | | |
| --- | --- | --- | --- |
| Created By: | DungHM | Date Created: | 28/Oct/2023 |
| Primary Actor: | Team leader | Secondary Actors: | None |
| Trigger: | Clicks “Submit assignment” button on ”Assignment Details” screen | | |
| Description: | *As a team leader, I want to submit assignments to the project mentor so that the project mentor can view and evaluate.* | | |
| Preconditions: | PRE-1. User account have Team leader role | | |
| Postconditions: | POST-1. Upload assignment successfully *to system* | | |
| Normal Flow | **6.0 Submit assignment**  1. Clicks “Submit assignment” button on ”Assignment details” screen  2. Choose file to upload  3. Click Submit | | |
| Alternative Flows: | None | | |
| Exceptions: | None | | |
| Priority: | Must have | | |
| Frequency of Use: | High | | |
| Business Rules: | BR - 21 | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

### 2.2.1 UC-07\_View user list

#### a. Functional Description

| UC ID and Name: | **UC-7\_View user list** | | |
| --- | --- | --- | --- |
| Created By: | DucTA | Date Created: | 28/Oct/2023 |
| Primary Actor: | Admin | Secondary Actors: | None |
| Trigger: | Clicks “User” button from the sidebar | | |
| Description: | *As an admin, I want to be able to view all users of the system so that I can manage all users.* | | |
| Preconditions: | PRE-1. User account have admin role | | |
| Postconditions: | POST-1. Show a list of all users in the system and their information. | | |
| Normal Flow | **7.0 List users**  1. User clicks the “User” in the sidebar  2. System show a list of all users | | |
| Alternative Flows: | None | | |
| Exceptions: | None | | |
| Priority: | Must Have | | |
| Frequency of Use: | Medium | | |
| Business Rules: | None | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

#### b. Business Rules

None

#### 

### 2.2.2 UC-08\_Add new user

#### a. Functional Description

| UC ID and Name: | **UC-8\_Add new user** | | |
| --- | --- | --- | --- |
| Created By: | DucTA | Date Created: | 18/Sep/2023 |
| Primary Actor: | Admin | Secondary Actors: | None |
| Trigger: | Clicks *Add new user* button in User list screen | | |
| Description: | *As an admin, I want to be able to add new users so that I can add new users like a new admin account, quickly create users without registering.* | | |
| Preconditions: | PRE-1. User account have admin role | | |
| Postconditions: | POST-1. Add a new user to the system successfully | | |
| Normal Flow | **8.0 Add new user**  1. User clicks the “Add new user” button.  2. The web displays a form for the admin to fill out the user's details.  3. User fills out the form and clicks the “Add” button.  4. A new user has been added to the system. | | |
| Alternative Flows: | **8.1. Email of a new user existing in the system or email domain isn’t appropriate.**  1. Fill other email that isn’t existed in system  2. Return to step 4 of normal flow.  **8.2. Phone number of new user existed in system**  1. Fill other phone that isn’t existed in system  2. Return to step 4 of normal flow. | | |
| Exceptions: | None | | |
| Priority: | Medium | | |
| Frequency of Use: | Medium | | |
| Business Rules: | FR1 | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

#### b. Business Rules

| **ID** | **Business Rule** | **Business Rule Description** |
| --- | --- | --- |
| FR1 | Password Encoding | User’s password must be encoded |

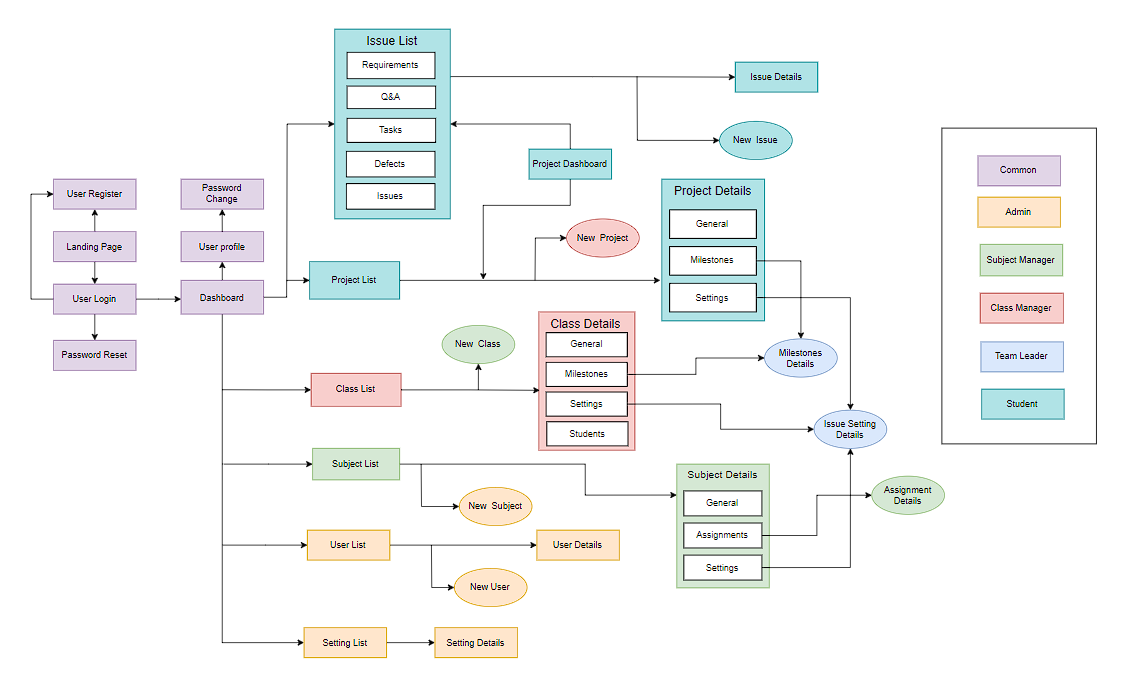
### 

#### 

## 3. Functional Requirements

### 3.1 System Functional Overview

#### a. Screen Flow



#### b. Screen Description

| **#** | **Feature** | **Screen** | **Description** |
| --- | --- | --- | --- |
| 1 | Common | Landing Page | View Landing Page - Login - Register |
| 2 | Common | User Register | Register a new account - Enter Full Name, Email or Phone Number, Password to register |
| 3 | Common | User Login | Enter email/phone number & password to login - Enter login details |
| 4 | Common | Password Reset | Reset password via Email: - Users enter their email to receive a password reset link, clicking the link then they can set a new password Reset password via SMS: - Users provide their phone number to get an OTP. After entering the OTP, they can set a new password |
| 5 | Common | Password Change | Allows the user to change password by inputting current password as well as new password (twice) |
| 6 | Common | User profile | View user profile information and update profile  - Update fullname or avatar  - Update email or phone number (before verify new email or phone number) |
| 7 | Admin | Admin Dashboard | Display total student, teacher, subject, …  Allows admin go to screen user list, subject list, setting list |
| 8 | Admin | User List | View all user  - Can change status of each user  - Go to details user |
| 9 | Admin | New user | Add new user into the system |
| 10 | Admin | User Details | Admin can view and change user detail |
| 11 | Admin | Subject List | View all subject are management  - Can change status of each subject  - Go to subject detail |
| 12 | Admin | New subject | Add new subject are management |
| 13 | Admin | Subject details | This screen allows to update all subject are management |
| 14 | Admin | Setting List | View all setting  - Can view and change status and add each setting |
| 15 | Admin | Setting Details | Update semester from system |
| 16 | Subject Manager | Subject Assignment List | Displays a list of all subject assignments |
| 17 | Subject Manager | New Subject Assignment | Add a new subject assignment to system |
| 18 | Subject Manager | Subject Assignment Details | Update assignments of subject for all class learn this subject |
| 19 | Subject Manager | Class List | Displays a list of all classes that learn this subject |
| 20 | Subject Manager | Class Details | Update details of a class |
| 21 | Subject Manager | New Class | Add a new class that learn this subject |
| 22 | Subject Manager | Subject Setting List | Displays a list of all subject setting |
| 23 | Subject Manager | Subject Setting Details | Update details of a subject setting |
| 24 | Class Manager | Project List | Display all project in class |
| 25 | Class Manager | New Project | Add a new project |
| 26 | Class Manager | Project Details | Update details of a project |
| 27 | Class Manager | Project Member List | Display all member of project group in class |
| 28 | Class Manager | Project Member Change | Add/remove student between project group |
| 29 | Class Manager | Assigned Class List | Displays a list of all classes that the class manager manages |
| 30 | Class Manager | Assigned Class Details | Update information of a class |
| 31 | Class Manager | Class Assignment Details | Displays a list of all assignments of a class that learn this subject |
| 32 | Class Manager | Class Issue Setting List | Displays a list of all issues of class |
| 33 | Class Manager | New Class Issue Setting | Add new class issue setting |
| 34 | Class Manager | Class Issue Setting Details | Update details of issue setting in a class |
| 35 | Class Manager | Class Student List | Displays a list of all student of class |
| 36 | Class Manager | Class Student Change | Add/remove student of class |

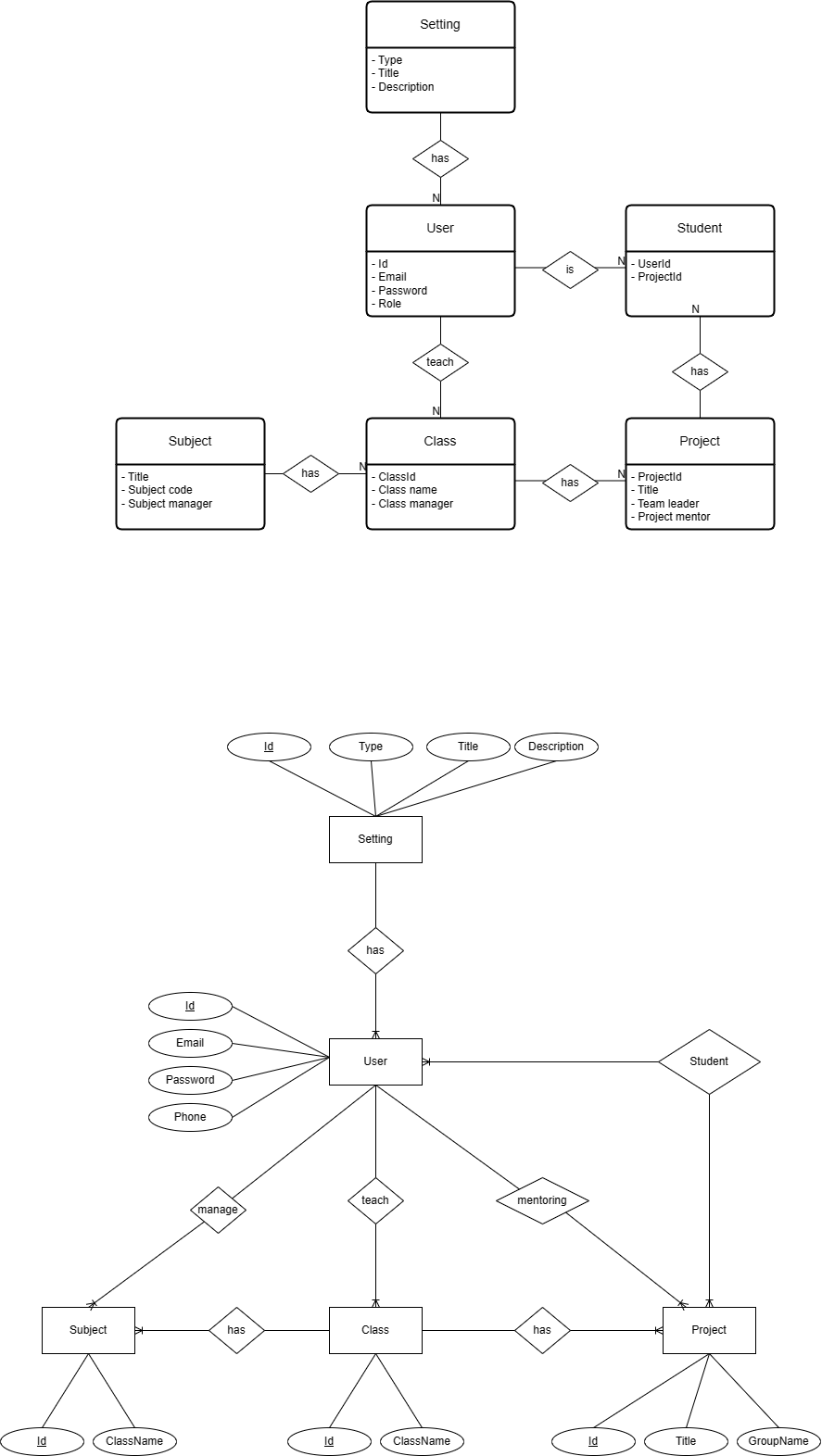
#### c. Screen Authorization

| **Screen** | **Guest** | **Student** | **Leader** | **Project Mentor** | **Class Manager** | **Subject Manager** | **Admin** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Landing Page | X | X | X | X | X | X | X |
| User Register | X | X | X | X | X | X | X |
| User Login |  | X | X | X | X | X | X |
| Password Reset |  | X | X | X | X | X | X |
| Password Change |  | X | X | X | X | X | X |
| User profile |  | X | X | X | X | X | X |
| Admin Dashboard |  |  |  |  |  |  | X |
| Users List |  |  |  |  |  |  | X |
| New user |  |  |  |  |  |  | X |
| User Details |  |  |  |  |  |  | X |
| Subjects List |  |  |  |  |  |  | X |
| New subject |  |  |  |  |  |  | X |
| Subject details |  |  |  |  |  |  | X |
| Setting List |  |  |  |  |  |  | X |
| Setting Details |  |  |  |  |  |  | X |
| Subject Assignment List |  |  |  |  |  | X |  |
| New Subject Assignment |  |  |  |  |  | X |  |
| Subject Assignment Details |  |  |  |  |  | X |  |
| Class List |  |  |  |  |  | X |  |
| Class Details |  |  |  |  |  | X |  |
| New Class |  |  |  |  |  | X |  |
| Subject Setting List |  |  |  |  |  | X |  |
| Subject Setting Details |  |  |  |  |  | X |  |
| Project List |  |  |  |  | X |  |  |
| New Project |  |  |  |  | X |  |  |
| Project Details |  |  |  |  | X |  |  |
| Project Member List |  |  |  |  | X |  |  |
| Project Member Change |  |  |  |  | X |  |  |
| Assigned Class List |  |  |  |  | X |  |  |
| Assigned Class Details |  |  |  |  | X |  |  |
| Class Assignment Details |  |  |  |  | X |  |  |
| Class Issue Setting List |  |  |  |  | X |  |  |
| New Class Issue Setting |  |  |  |  | X |  |  |
| Class Issue Setting Details |  |  |  |  | X |  |  |
| Class Student List |  |  |  |  | X |  |  |
| Class Student Change |  |  |  |  | X |  |  |

#### d. Non-Screen Functions

| **#** | **Feature** | **System Function** | **Description** |
| --- | --- | --- | --- |
| 1 | Common | Logout | The process of signing out or disconnecting from a system or account to secure personal data |
| 2 | Common | Authorization | Authorize user into the system functions based on his/her role |

#### e. Entity Relationship Diagram

**

**Entities List**

| **#** | **Entity** | **Description** |
| --- | --- | --- |
| 1 | User |  |
| 2 | Subject |  |
| 3 | Class |  |
| 4 | Project |  |
| 5 | Student |  |
| 6 | Setting |  |

### 3.2 Authentication

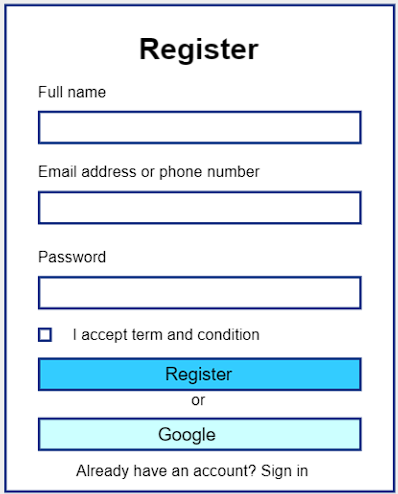
#### a. User Register

This screen allows user to be authenticated to the system screens/functionalities.

Related use cases:

[UC-1\_Register](https://docs.google.com/document/d/12WUVF9drpY1HSYgpNq1JY2UvNiHSz3SUNO5U4dRYRJM/edit#heading=h.1pgrrkc)

##### UI Design



| **Field Name** | **Field Type** | **Description** |
| --- | --- | --- |
| Fullname\* | Text Box | Enter full name |
| Username\* | Text Box | Enter Email or Phone number |
| Password\* | Password Box | Enter password |
| Register | Button | Click to Register account |
| Google | Button | Click to sign in or sign up with google |
| Sign in | Hyperlink | Click to go Login Page |

##### Database Access

| **Table** | **CRUD** | **Description** |
| --- | --- | --- |
| User | C | Insert new user into database |

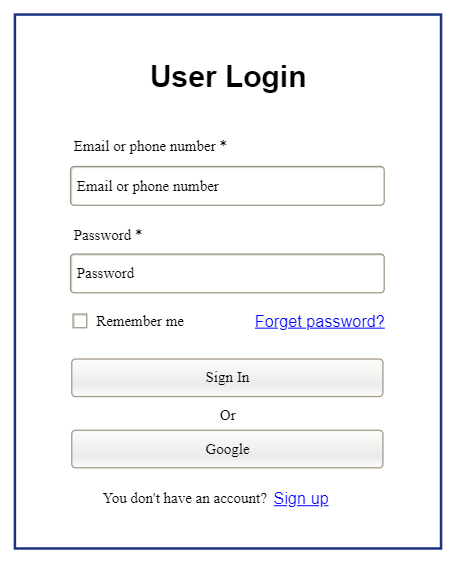
#### b. User Login

This screen allows users change their password.

Related use cases:

[UC-2\_Login](https://docs.google.com/document/d/12WUVF9drpY1HSYgpNq1JY2UvNiHSz3SUNO5U4dRYRJM/edit#heading=h.49gfa85)

##### UI Design



| **Field Name** | **Field Type** | **Description** |
| --- | --- | --- |
| Email or phone number\* | Text Box | This is for user to input valid email address/ valid phone number for logging in |
| Password\* | Password Box | This is for user to input password for logging in |
| Login | Button | User clicks to authenticate him/herself into the system with provided email/phone & password |
| Remember me | Check Box | User ticks to remember login information |
| Forgot Password? | Hyperlink | User clicks to redirect to the Password Reset page for resetting his/her forgot password |
| Login with Google | Hyperlink | Allow user to login with his/her Google account |
| Sign up | Hyperlink | User clicks to redirect to the User Register page for registering new user account to access the system |

##### Database Access

| **Table** | **CRUD** | **Description** |
| --- | --- | --- |
| User | CR | Find User By Email & Password information  Find User By Phone & Password information  Check Existed Email (login with google)  Create new user (login with google) |
| Setting | R | Check Email Domain |

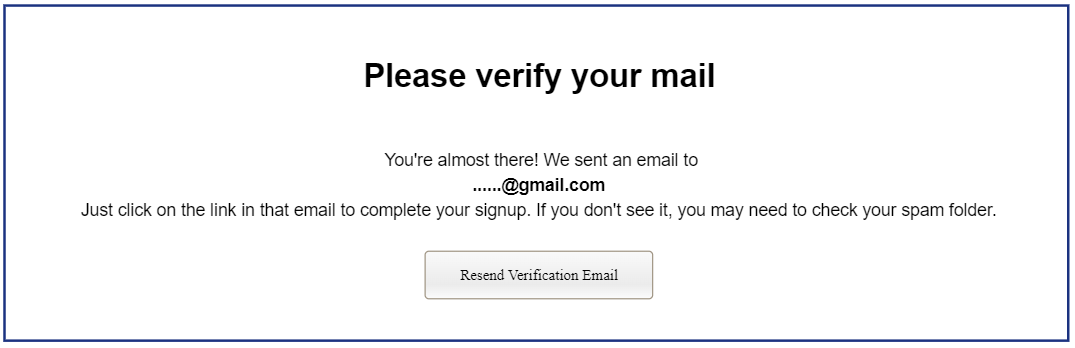
#### c. Password Reset

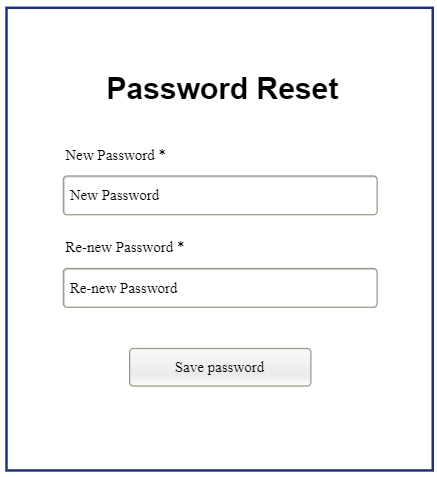
**Function Trigger:** When the user clicks the Forgot Password button.

**Function Description:** This function allows users to regain access to their accounts in case they forget their passwords or suspect that their accounts have been compromised.

**Screen Layout:**







**Function Details:**

* In the Login page, if the user clicked on forget password, it will navigate to reset the password page.
* After the user input their email, if the email exists, the link will be sent to this email.
* When the user clinks to the link in the email, the web will redirect to the page that can create a new password

### 3.3 Admin Feature

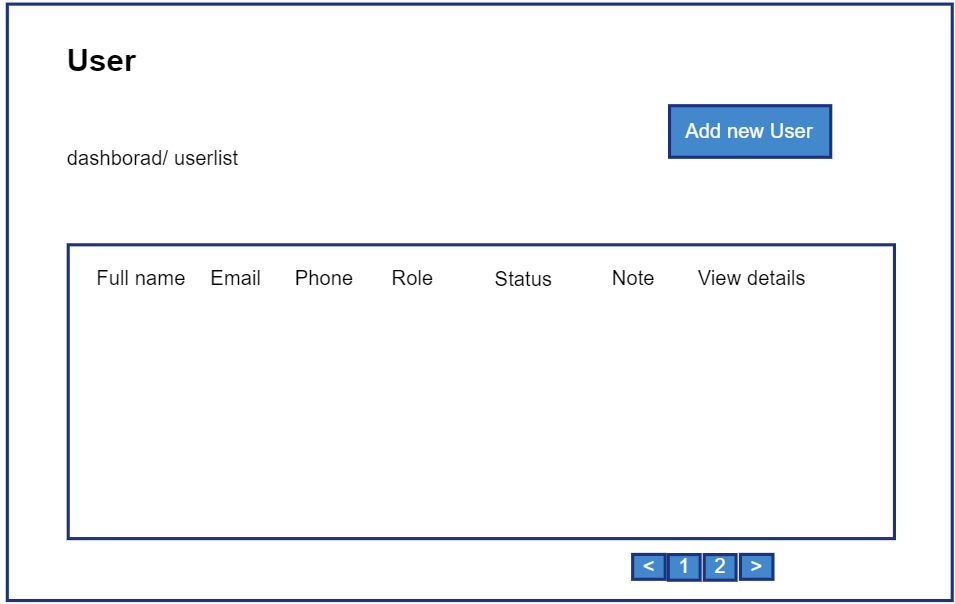
#### a. User List

This screen allows admins to see the list of users

Related use cases:

* UC-7\_View user list

##### UI Design



| **Field Name** | **Field Type** | **Description** |
| --- | --- | --- |
| Add New | Hyperlink | Click to open New User page for adding new user(master data) |
| ***Data Table*** | | |
| Fullname | Text | Show full name of user |
| Email | Text | Show email of user |
| Phone | Text | Show phone of user |
| ***Data Status*** | | |
| Activate/ Deactivate | On/Off toggle | Shown when the data status is inactive. This is to activate/deactivate the relevant user (master data) |
| ***Data Actions*** |  |  |
| Edit | icon | Click to open the User Details page for updating the relevant user (master data) |

##### Database Access

| **Table** | **CRUD** | **Description** |
| --- | --- | --- |
| User | RU | Query the list of current users from the database  Update status of a specific user |

***SQL Commands:***

SELECT *\** FROM *user*;

UPDATE *user*  
SET *status = ?*

WHERE *id = ?*

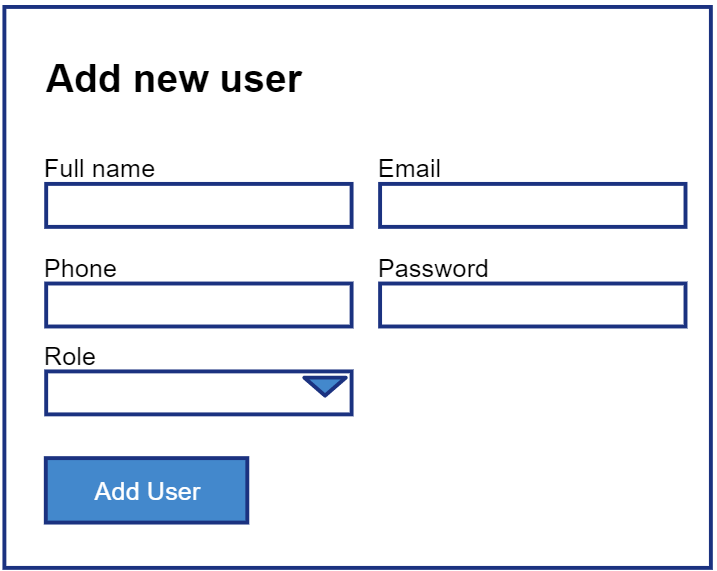
#### b. New User

This screen allows admins to add a new user.

Related use cases:

* UC-8\_Add new user

##### UI Design

****

| **Field Name** | **Field Type** | **Description** |
| --- | --- | --- |
| Fullname | Text Box | Input full name of new user |
| Email | Text Box | Input email of new user |
| Phone | Text Box | Input phone of new user |
| Role | Select box | Select role of new user |
| Add user | Button | Click to add new user |

##### Database Access

| **Table** | **CRUD** | **Description** |
| --- | --- | --- |
| User | C | Insert new user into database |

***SQL Commands:***

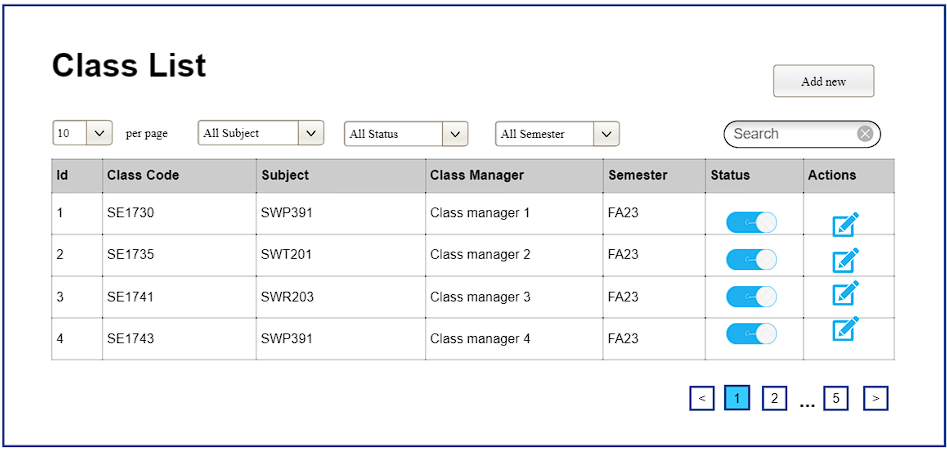
INSERT INTO *User* (full\_name, email, phone, role\_id)  
VALUES (*?, ?, ?, ?*);

#### c. Class List

**Function Trigger:** When the user clicks the Class button in the sidebar..

**Function Description:** This function allows users to view list of classes

**Screen Layout:**

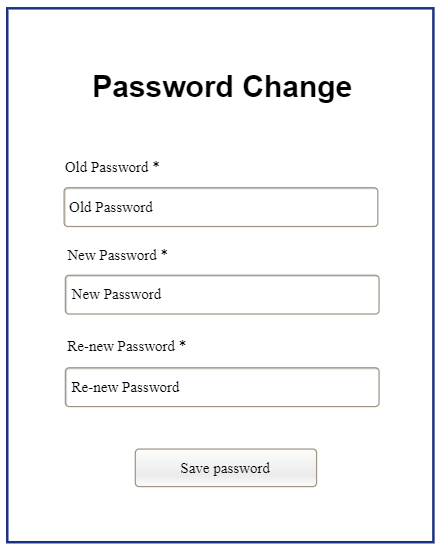


**Function Details:**

* In the Class List page, the user can search class by name, subject, manager, …
* The user also can filter class by subject, status, semester,..

#### d. Change password

**UI Design**



| **Field Name** | **Field Type** | **Description** |
| --- | --- | --- |
| Old password \* | Password Box | This is for user to input old password |
| New Password\* | Password Box | This is for user to input new password |
| ReNew Password\* | Password Box | This is for user to input confirm new password |
| Save password | Button | User clicks to update new password |

##### Database Access

| **Table** | **CRUD** | **Description** |
| --- | --- | --- |
| User | U | Update new password |

***SQL Commands:***

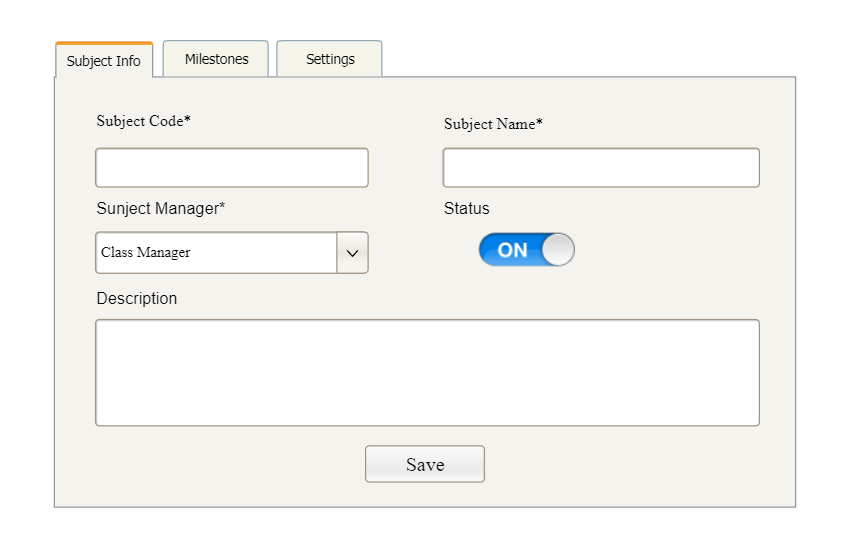
UPDATE *user*  
SET *password=?*

WHERE *id = ?*

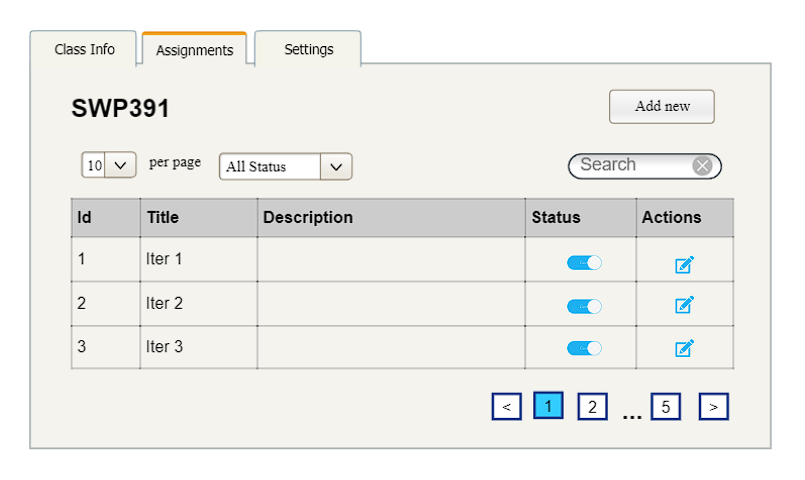
#### 

#### e. Update subject

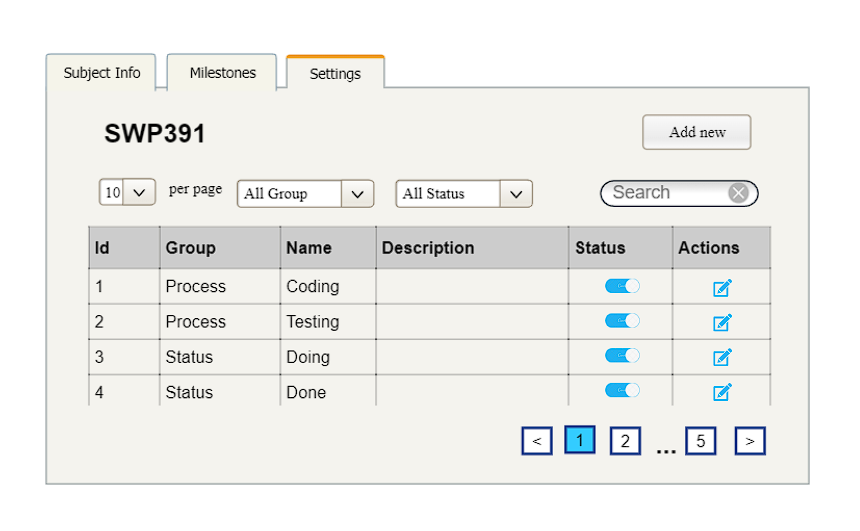
##### UI Design



| **Field Name** | **Field Type** | **Description** |
| --- | --- | --- |
| Subject Code | String | Code of the subject |
| Subject Name | String | Name of the subject |
| Subject Manager | String | Manager of the subject |
| Status | Select box | Status of the subject |
| Update Subject | Button | Click to update subject details |



| **Field Name** | **Field Type** | **Description** |
| --- | --- | --- |
| Add New Subject Assignment | Hyperlink | Click to open the Subject Details page for adding new subject assignment (master data) |
| ***Data Table*** | | |
| Subject Code | Text | Code of the subject |
| Subject Name | Text | Name of the subject |
| Assignment Title | Text | Title of the assignment |
| Subject Description | Text | Description of the assignment |
| ***Data Actions*** |  |  |
| View details | Hyperlink | Click to open the Subject Details page for updating the relevant Subject Assignment (master data) |



| **Field Name** | **Field Type** | **Description** |
| --- | --- | --- |
| Add New | Hyperlink | Click to open the Subject Setting Details page for adding new subject setting (master data) |
| ***Data Table*** | | |
| Subject | Text | Name of subject that subject manager manages |
| Type | Text | Type of the subject setting |
| Tittle | Text | Name of subject setting |
| Display Order | Integer | Display order of the setting: the order of the subject setting type, displayed among the list of subject settings with the same type |
| ***Data Status*** | | |
| Activate/ Deactivate | On/Off toggle | Shown when the data status is inactive. This is to activate/deactivate the relevant subject setting (master data) |
| ***Data Actions*** |  |  |
| Edit | icon | Click to open the Subject Setting Details page for updating the relevant setting (master data) |

##### Database Access*SQL Commands:*

| **Table** | **CRUD** | **Description** |
| --- | --- | --- |
| Subject | RU | Query specific subject details from the database  Update specific subject details |
| Assignment | RU | Query the list of current Subject Assignment from the database  Update status of a specific SubjectAssignment |
| Subject, User | R | Query the manager of the Subject Assignment |
| Subject\_setting | RU | Query the list of current subject settings of the subject that subject manager manages from the database  Update status of a specific setting setting |

***SQL Commands:***

SELECT *\** FROM *subject*

WHERE *id=?*

UPDATE *subject*  
SET *subjectName*=*?*,*subjectCode*=*?*, subjectManager = ?, status = ?  
WHERE *id = ?*;

SELECT *\** FROM *subject\_assignment*;

SELECT *\** FROM subject;

SELECT *ss.id, s.subject\_code, ss.type\_id, ss.setting\_title, ss.display\_order, ss.status*

FROM *subject\_setting ss join subject s on ss.subject\_is = s.id*

WHERE *s.subject\_manager\_id = ?*

UPDATE subject\_*setting*  
SET *status = ?*

WHERE *id = ?*

## 4. Non-Functional Requirements

### 4.1 External Interfaces

#### a. User Interfaces

**REQUIREMENT**: The student project portal’s UI must be user friendly, easily recognized and easily get used to.

**SCALE**: Will be measured in terms of the interface’s designs and references.

**GOAL**: Create an intuitive and user-friendly interface that provides a seamless and efficient experience for users, promoting easy navigation, accessibility, and satisfaction with the system.

**STRETCH**: In addition to the primary goal, there is a stretch goal to implement redundancy and fault-tolerance measures to achieve a minimum system uptime of 99% during day hours (22 hours per day). This higher level of reliability ensures that the system is available for use almost continuously, with minimal downtime.

### 4.2 Quality Attributes

#### a. Usability

**REQUIREMENT**: The student project portal must be highly reliable, with maximum uptime and efficient failure recovery.

**AMBITION**: The ambition is to ensure that the system operates without frequent disruptions and provides efficient recovery mechanisms in case of failures.

**SCALE**: Reliability will be measured in terms of the following metrics:

**METER**: Reliability assessment will involve:

**GOAL**: The primary goal is to achieve a minimum system uptime of 90% during day hours (22 hours per day). This means that the system should be available and functional for at least 90% of the day.

**STRETCH**: In addition to the primary goal, there is a stretch goal to implement redundancy and fault-tolerance measures to achieve a minimum system uptime of 99% during day hours (22 hours per day). This higher level of reliability ensures that the system is available for use almost continuously, with minimal downtime.

#### b. Reliability

**TAG:** Reliability.Requirement

**AMBITION:** Ensure a highly reliable system with maximum uptime and efficient failure recovery.

**SCALE:** Measure reliability in terms of system uptime, repair time, and Concurrent User Capacity (CCU).

**METER:** Reliability assessment will include continuous monitoring and real-world failure recovery testing.

**GOAL:** Achieve a minimum system uptime of 90% during day hours (22 hours/day).

**STRETCH:** Implement redundancy and fault-tolerance measures to achieve a minimum system uptime of 99% during day hours (22 hours/day).

#### 

#### c. Performance

**TAG**: UserLoadCapacity

**AMBITION:** Ensure the Student Project Portal is scalable and offers optimal performance for its users.

**SCALE**: Concurrent users supported and response time (seconds) for common operations.

**METER**: Conduct load testing with simulated concurrent access for up to 5,000 users.

Measure response times for operations such as logging in, accessing a project, and submitting an assignment under peak load conditions.

**GOAL**: Support up to 5,000 concurrent users without degradation in system performance.

Achieve a response time of less than or equal to 3 seconds for common operations under peak load conditions.

**STRETCH**: Ensure the system is scalable to accommodate a 25% increase in user load annually without requiring significant architectural changes.

**BASE USER PLATFORM DEFINED**: Standard user platform with a quad-core processor, 8GB RAM, modern web browser, and a network connection speed of at least 50 Mbps.

### 