

**CAPSTONE PROJECT REPORT**

**Report 3 – Software Requirement Specification**

– Hanoi, August 2023 –

**Table of Contents**

[I. Record of Changes 3](#_gjdgxs)

[II. Software Requirement Specification 4](#_30j0zll)

[1. Requirement Overview 4](#_1fob9te)

[1.1 Context Diagram 4](#_3znysh7)

[1.2 User Requirements 5](#_2et92p0)

[1.3 System Functionalities 5](#_tyjcwt)

[2. Functional Specifications 7](#_3dy6vkm)

[2.1 <<Feature/Workflow Name1>> 7](#_1t3h5sf)

[2.2 Common Feature 10](#_17dp8vu)

[2.3 Patron Feature 13](#_26in1rg)

[3. Non-Functional Requirements 16](#_1ksv4uv)

[3.1 External Interfaces 16](#_44sinio)

[3.2 Quality Attributes 16](#_2jxsxqh)

[4. Requirement Appendix 17](#_1y810tw)

[4.1 Business Rules 17](#_4i7ojhp)

[4.2 Common Requirements 17](#_2xcytpi)

[4.3 Application Messages List 17](#_1ci93xb)

[4.4 Other Requirements… 18](#_3whwml4)

# I. Record of Changes

| **Date** | **A\* M, D** | **In charge** | **Change Description** |
| --- | --- | --- | --- |
| 24/9/2023 | A | DucVD | add requirement overview |
| 24/9/2023 | A | DucVD | use case diagram |
| 24/9/2023 | A | PhongVD | screen flow |
| 25/9/2023 | A | TruongNQ | Screen Descriptions |
| 25/9/2023 | A | TruongNQ | Entities Description |
| 25/9/2023 | A | PhongVD | use case diagram |
| 25/9/2023 | A | PhongVD | Non UI-Function |
| 25/9/2023 | A | PhongVD | Actor |
| 25/9/2023 | A | DungND | Manager's UCs |
| 25/9/2023 | A | DucVD | Add function specification manager feature |
| 1/10/2023 | M | DungND | Manager Feature (Manage class) |
| 1/10/2023 | M | TruongNQ | Update Non-UI function |
| 1/10/2023 | A | TruongNQ | Student Feature |
| 1/10/2023 | M | DucVD | Modified use case diagram |
| 1/10/2923 | A | DucVD | Add entity diagram |
| 1/10/2023 | M | TruongNQ | Entities Description of Entity Relationship Diagram |
| 1/10/2023 | M | PhongVD | Teacher Feature ( Manage project) |
| 10/10 | A | DungND | UI Specification of Manager Feature (Update class) |
| 10/10 | A | DungND | UI Specification of Manager Feature (View class list) |
| 10/10 | A | DungND | UI Specification of Manager Feature (Add new class) |
| 10/10 | A | DungND | UI Specification of Manager Feature (Delete pending class) |
| 10/10 | A | DungND | UI Specification of Manager Feature (Cancel started class) |
| 10/10 | A | DungND | UI Specification of Manager Feature (Update class) |
| 10/10 | A | DungND | UI Specification of Manager Feature (View class list) |
| 15/10 | M | TruongNQ | Student Feature |
| 15/10 | M | DucVD | Modify Function specification of admin feature |
| 15/10 | A | DucVD | UI Specification of Manage Subject (View Subject) |
| 15/10 | A | DucVD | UI Specification of Manage Subject (View Detail Subject) |
| 15/10 | A | DucVD | UI Specification of Manage Subject (Add Subject) |
| 15/10 | A | DucVD | UI Specification of Manage Subject (Update Subject) |
| 15/10 | A | PhongVD | UI Specification of Teacher Feature (View Project) |
| 15/10 | A | PhongVD | UI Specification of Teacher Feature (Add project) |
| 15/10 | A | PhongVD | UI Specification of Teacher Feature (View Project Details) |
| 15/10 | A | PhongVD | UI Specification of Teacher Feature (Add project) |
| 15/10 | A | PhongVD | UI Specification of Teacher Feature (Update project) |
| 15/10 | A | PhongVD | UI Specification of Teacher Feature (View project team) |
| 15/10 | A | PhongVD | UI Specification of Teacher Feature (View member team) |
| 15/10 | A | PhongVD | UI Specification of Teacher Feature (Remove another team) |
| 15/10 | A | PhongVD | UI Specification of Teacher Feature (Update with note) |
| 5/11 | M | TruongNQ | Context Diagram |
| 5/11 | M | TruongNQ | Screen Description |
| 5/11 | M | DungND | Screen Flow |

\*A - Added M - Modified D - Deleted

# II. Software Requirement Specification

## 1. Requirement Overview

### 1.1 Context Diagram

The Issue Management System (IMS) is a comprehensive software solution designed to facilitate the management of student projects in SWP391, ISP392 courses, and SEP490 projects. IMS serves various user groups within an educational institution, including guests, users, students, team leaders, teachers, managers, and administrators

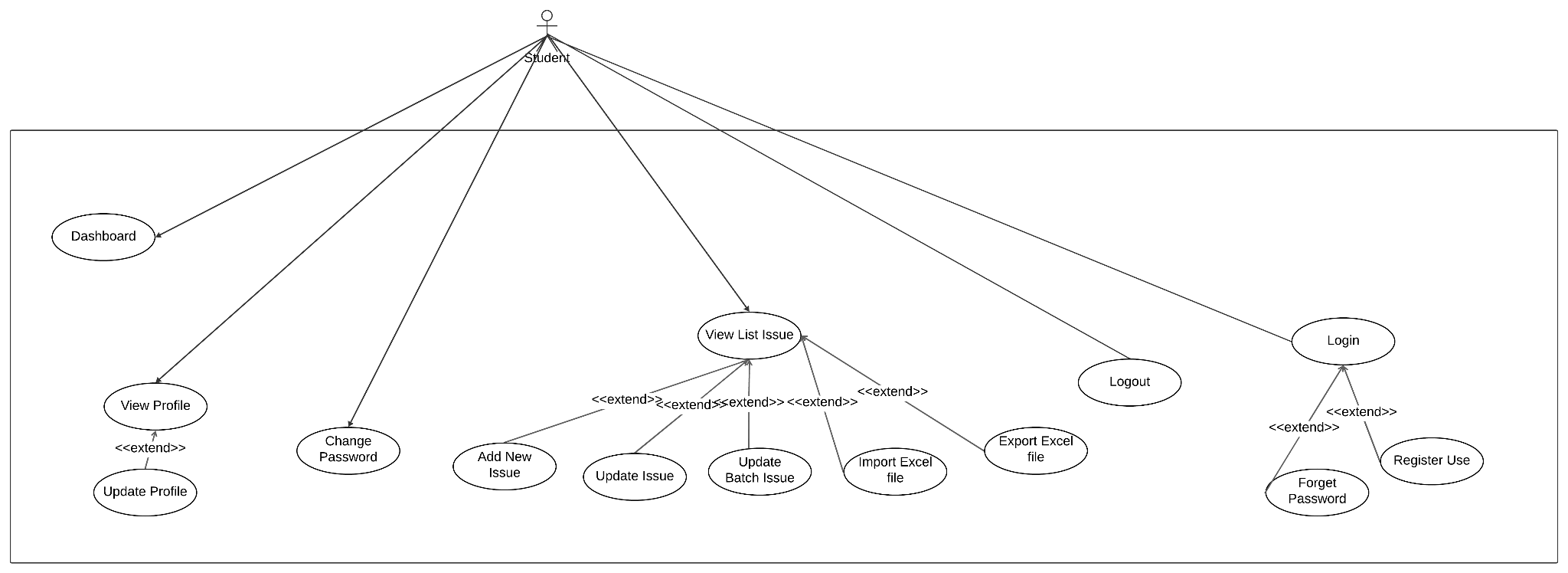
### 

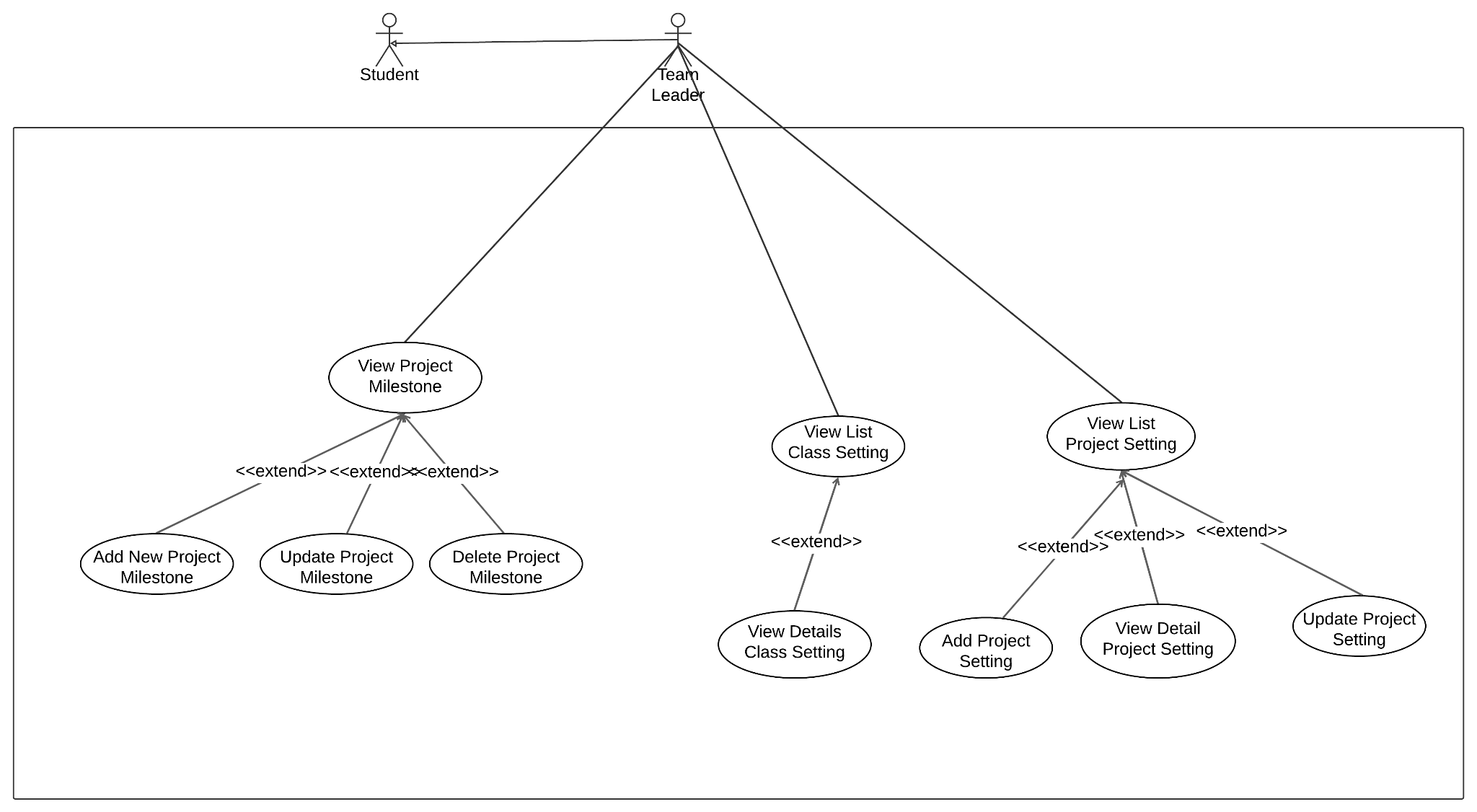
### 1.2 User Requirements

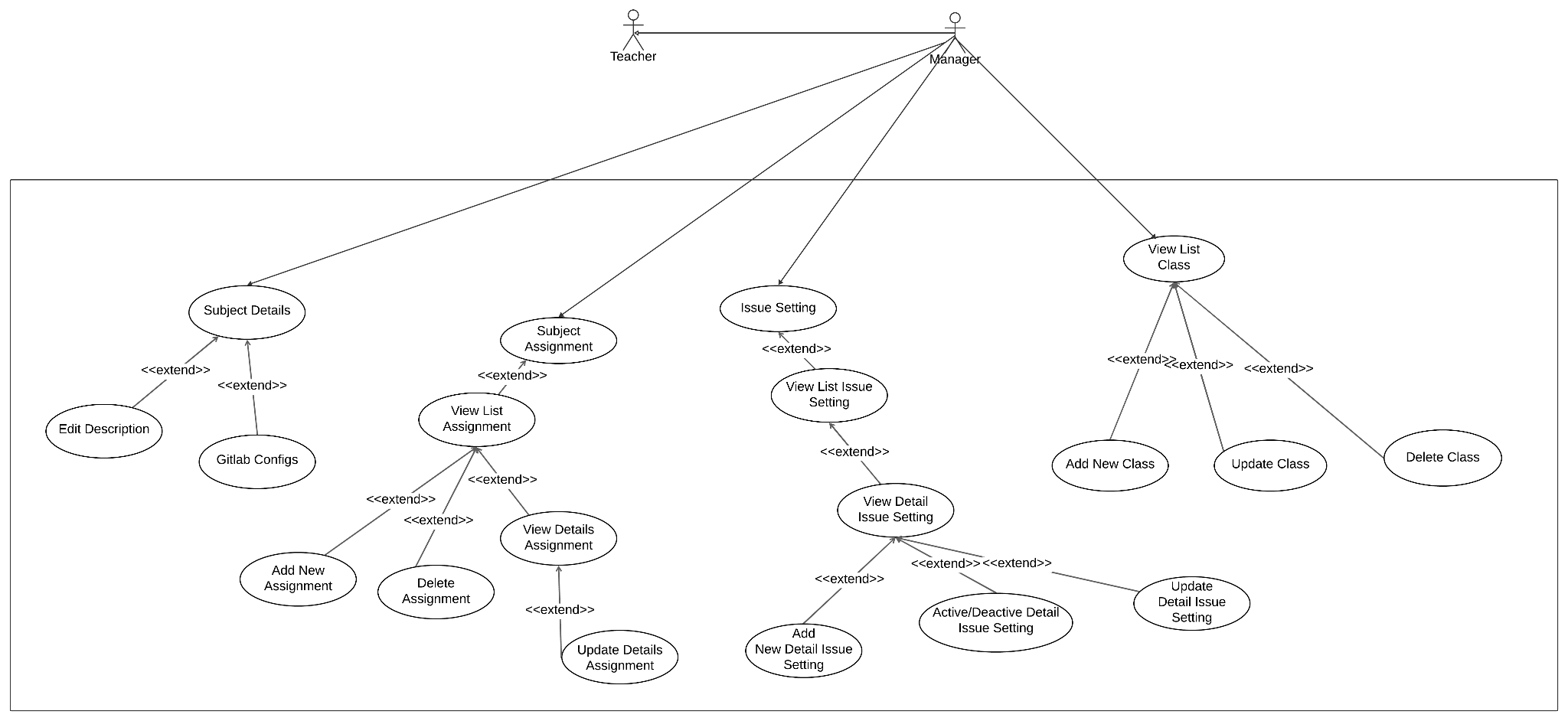
#### 1.2.1 Actors

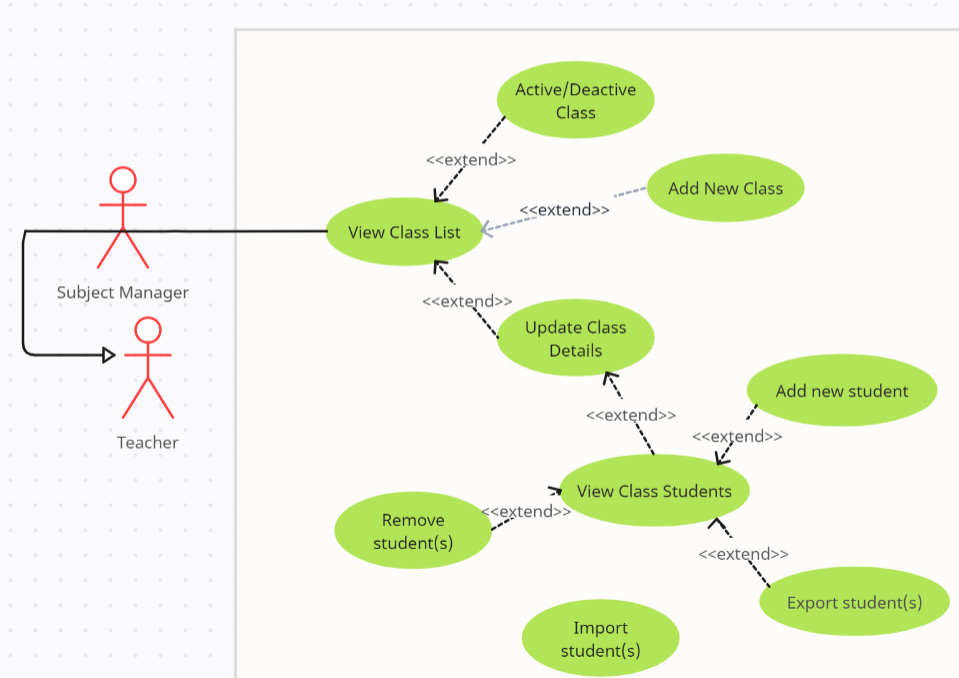
| **#** | **Actor** | **Description** |
| --- | --- | --- |
| 1 | Student | Students have functionalities related to managing project issues, including viewing lists, adding new issues, updating issues, importing/exporting from/to Excel files, and synchronizing project issues with GitLab projects. |
| 2 | Team Leader | Student team leaders have the capability to manage project milestones, including viewing lists, adding new milestones, updating pending milestones, and synchronizing project milestones with GitLab project milestones. They also have the ability to manage project issue settings. |
| 3 | Teacher | Teachers can manage assigned classes, update class details, change class statuses, manage milestones, and manage class students. They can also view and update details of a project and manage project members within teams. |
| 4 | Manager | Subject managers can manage assigned subjects, edit subject details, manage subject assignments, and manage issue settings for subjects. They can also manage classes and their details. |
| 5 | Admin | System administrators have the authority to manage subjects, manage users, and manage system settings such as roles, semesters, and permitted email domains. |
| 6 | Guest | Guests only can view the landing page and register a user account if they use the system. |
| 7 | GitLab | Synchronize the project |

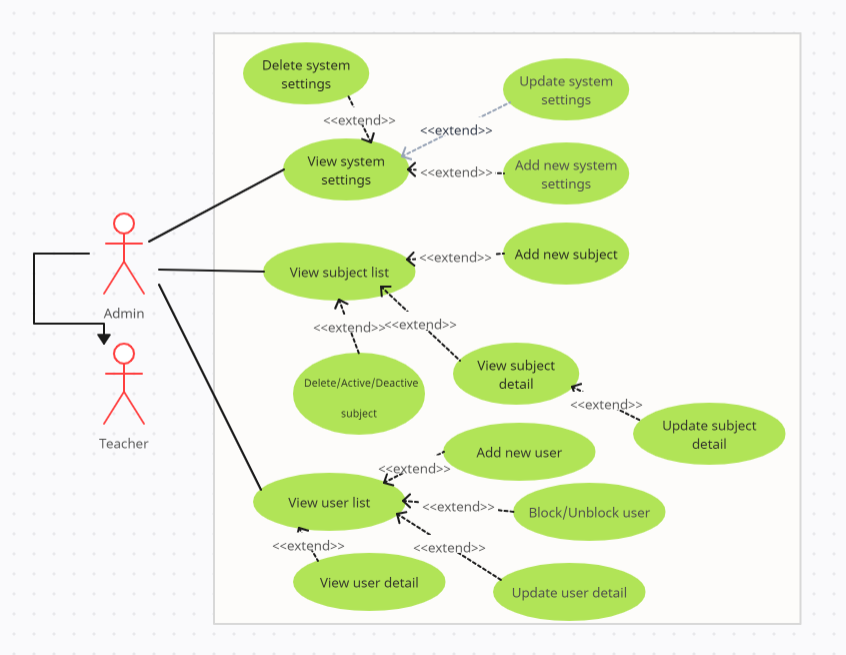
#### 1.2.2 Use Cases

**

**

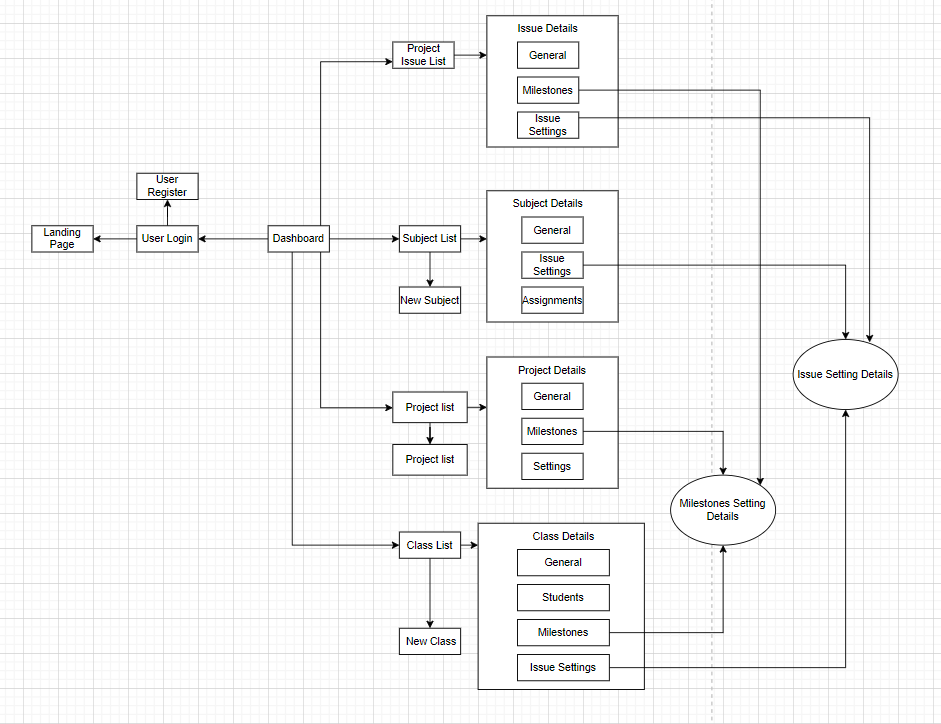
**





### 1.3 System Functionalities

#### 1.3.1 Screens Flow



#### 1.3.2 Screen Descriptions

*[Provide the descriptions for the screens in the Screens Flow above]*

| **#** | **Feature** | **Screen** | **Description** |
| --- | --- | --- | --- |
| 1 | Authentication and Management | User Login | This function allows users to enter their credentials and log in to their accounts. It typically includes fields for username/email and password, along with a login button. |
| 2 | Authentication and Management | User Register | This function enables users to create a new account. It usually includes fields for necessary information such as username, email, password, and additional details. |
| 3 | Project Management | Project List | This Screen displays a list of all projects with filtering/search options. |
| 4 | Admin Future | Subject List | This screen empowers administrators to manage subjects. |
| 5 | Teacher Future | Class List | This Screen displays a list of all classes |
| 6 | Manage Future | Class Detail | This Screen displays classes detail |
| 7 | Manage Future | Project Detail | This Screen displays Project detail |
| 8 | Issue Tracking | Milestone setting detail | This screen displays a list of milestone detail. |
| 9 | Issue Tracking | Issue setting detail | This screen displays a list of project issues. Students can view the details of each issue, including its status, priority, assigned personnel, and any related discussion or updates. |
| 10 | Manage Future | Subject List | This Screen displays a list of Subject |
| 11 | Manage Future | Issue List | This Screen displays a list of Issue |

#### 1.3.3 Screen Authorization

#### 

| **Screen** | **Guest** | **Leader** | **Manager** | **Admin** | **Teacher** | **Student** |
| --- | --- | --- | --- | --- | --- | --- |
| Home Page | X | X | X | X | X | X |
| Login | X | X | X | X | X | X |
| Register | X | X | X | X | X | X |
| User Profile |  | X | X | X | X | X |
| Change Password |  | X | X | X | X | X |
| Dashboard |  | X | X | X | X | X |
| Project Dashboard |  | X | X | X | X | X |
| Manage Project issues |  | X |  |  |  | X |
| -View list project issues |  | X |  |  |  | X |
| - Add new project issues |  | X |  |  |  | X |
| -Update project issues |  | X |  |  |  | X |
| -Batch update |  | X |  |  |  | X |
| -Synchronise project issues |  | X |  |  |  | X |
| Mange project milston |  | X |  |  |  |  |
| -View list pending milstion |  | X |  |  |  |  |
| -Add list pending milston |  | X |  |  |  |  |
| -Update pending milston |  | X |  |  |  |  |
| -Delete pending milston |  | X |  |  |  |  |
| -Synchronize project milston |  | X |  |  |  |  |
| Manage project issue setting |  | X |  |  |  |  |
| - view a list and details of relevant class settings |  | X |  | X |  |  |
| -Add new setting |  | X |  | X |  |  |
| -View update detail project setting |  | X |  |  |  |  |
| -Synchronization of project settings |  | X |  |  |  |  |
| Manage Assigned Classes |  |  | X | X | X |  |
| - View list; update class details |  |  | X | X | X |  |
| -Change a class to Running status |  |  | X | X | X |  |
| - Manage milestones |  |  | X | X | X |  |
| - Add New, Update, delete customised class milestones |  |  | X | X | X |  |
| -Update subject milestones |  |  | X | X | X |  |
| -Manage issue settings |  |  | X | X | X |  |
| -Manage class students |  |  | X | X | X |  |
| Manage projects |  |  |  |  | X |  |
| -View list of project teams |  |  |  |  | X |  |
| Export current teams' member list |  |  |  |  | X |  |
| import new project allocations |  |  |  |  | X |  |
| Remove or move a member |  |  |  |  | X |  |
| Make a project member as the only team leader |  |  |  |  | X |  |
| Update a team member's status with note |  |  |  |  | X |  |
| Manage classes |  |  | X |  |  |  |
| view list, add new, update, delete pending class(es) |  |  | X |  |  |  |
| cancel started class(es) |  |  | X |  |  |  |
| Manage subjects: |  |  |  | X |  |  |
| view list, view details, add new, delete/activate/deactivate |  |  |  | X |  |  |

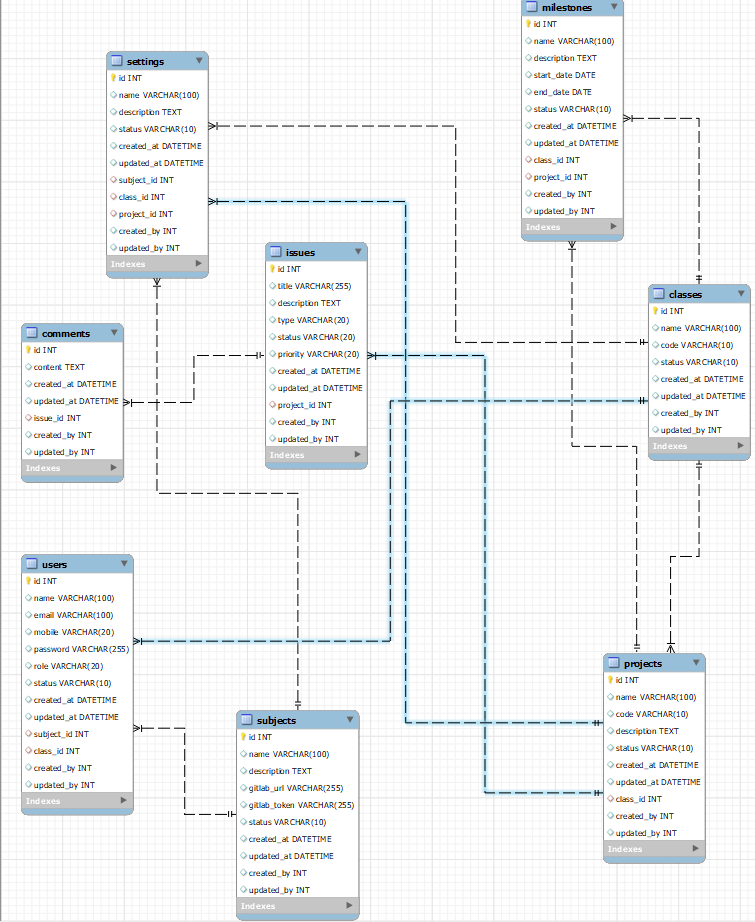
#### 1.3.4 Non-UI Functions

*[Provide the descriptions for the non-screen system functions, i.e batch/cron job, service, API, etc.]*

| **#** | **Feature** | **System Function** | **Description** |
| --- | --- | --- | --- |
| 1 | Admin | User Management | Administering user accounts, roles, and permissions is another important non-UI function. It allows administrators to control who can access and perform actions within the system. |
| 2 | Admin | Reporting and Analytics | The system may offer reporting and analytics features that allow users to generate custom reports and gain insights into issue trends and performance metrics. |
| 3 | Teacher | Issue Assignment | Teachers can assign issues or tasks to students, indicating deadlines and priorities for classroom or coursework management. |
| 4 | Synchronize project milestones | Synchronized | Synchronize project milestones with the GitLab project milestones |
| 5 | Synchronize class milestones & class issue settings | Synchronized | Synchronize class milestones & class issue settings with the GitLab group milestones & labels |
| 6 | Synchronize subject issue settings | Synchronized | Synchronize subject issue settings with the GitLab group labels |

#### 1.3.5 Entity Relationship Diagram

*[Provide the entity relationship diagram and the entity descriptions in the table format as below]*



**Entities Description**

| **#** | **Entity** | **Description** |
| --- | --- | --- |
| 1 | User | They can have different roles such as administrators, managers, teachers, students, or other stakeholders. Users log in to the system to create, manage, and track issues, collaborate with others, and configure their personal settings. |
| 2 | Subject | Subjects help categorize and organize issues based on the area or domain they pertain to. Subjects can be linked to specific classes, projects, or assignments, providing a way to filter and manage related issues. |
| 3 | Class | Class used for education, classes might be associated with teachers, students, and specific subjects. They can be used to manage and track issues related to a particular class or course. |
| 4 | Project | They can represent software development projects, research projects, or any other type of organized effort. Issues related to a project can be categorized, prioritized, and assigned to team members for resolution. |
| 5 | Comment | Users can add comments to issues to provide status updates, discuss solutions, or share insights. Comments facilitate communication and collaboration among team members working on an issue. |
| 6 | Issue | They represent problems, tasks, or items that require attention, tracking, and resolution. Issues have attributes such as status, priority, assignee, and a description to help manage their lifecycle. |
| 7 | Setting | Users, particularly administrators, can customize system settings, such as notifications, access controls, workflows, and integration settings, to align the system with their organizational needs. |
| 8 | Milestone | They help break down large projects into manageable phases or achievements. Milestones can be associated with specific issues or projects, and their completion often marks important progress in the overall effort. |

## 2. Functional Specifications

2.1 Admin Feature

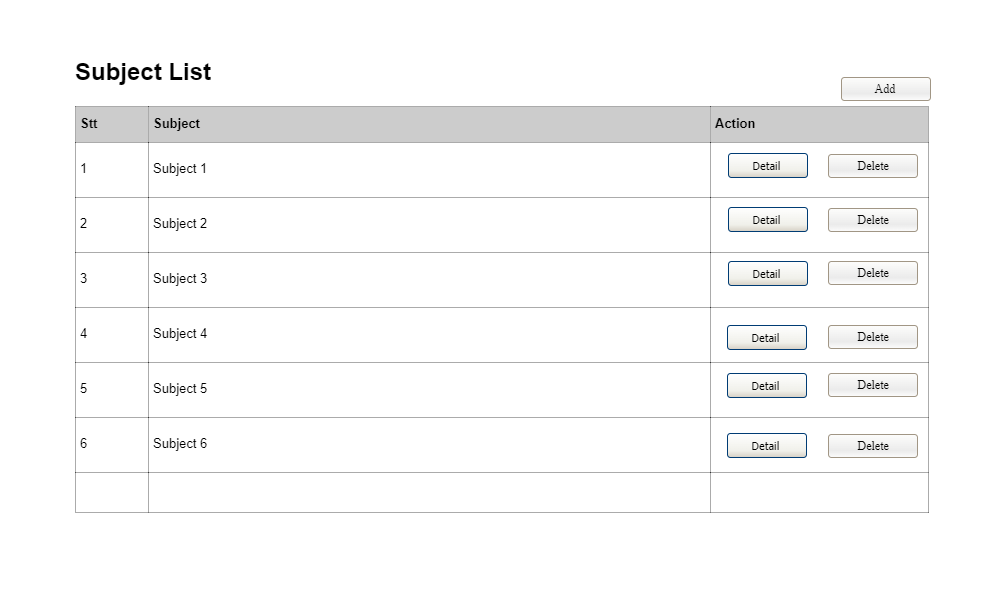
#### 2.1.1 Subject Setting

##### 2.1.1.1 Subject View

###### **a. UI Specifications**

### View subject list

Screen provide view all the subject list

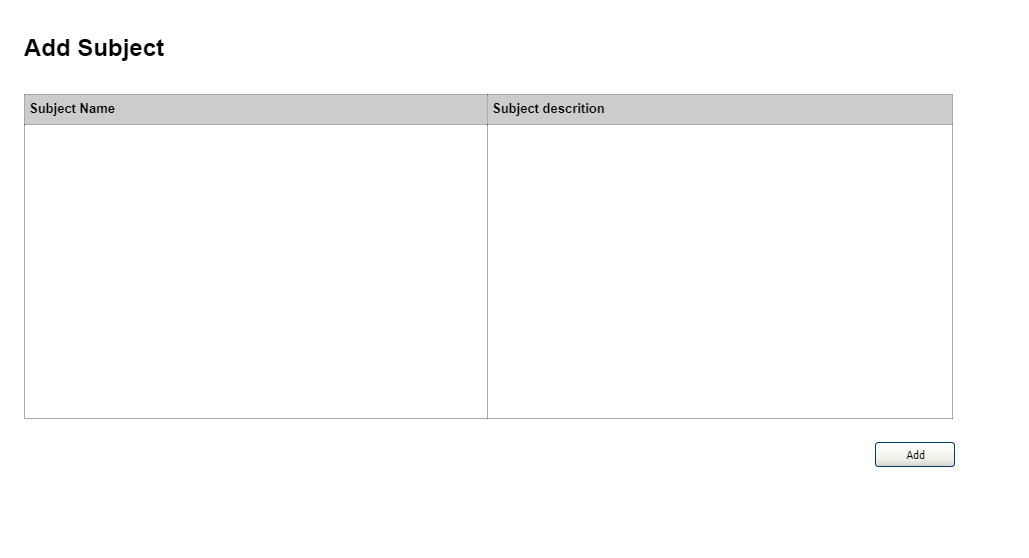


| Field Name | Field Type | Description |
| --- | --- | --- |
| *View subject list* | | |
| Stt | Number | Provide column order |
| Subject | Text | Provide name of the subject |
| Detail | Button | When you click go to detail screen |
| Delete | Button | Delete the subject |
| Add | Button | When you click the add button go to add screen |

### 

### Add subject list

Screen provide to add subject to subject list



### 

| Field Name | Field Type | Description |
| --- | --- | --- |
| *Add Subject* | | |
| Subject Name | Text | Enter Subject name |
| Subject Description | Text | Enter description of the subject |
| Add | Button | Click add button add new subject to subject list |

### 

###### **b. UC Specifications**

#### UC001\_ View Subject List

| UC ID and Name: | UC001 - View Subject List | | |
| --- | --- | --- | --- |
| Created By: | DucVD | Date Created: | 24/9/2023 |
| Primary Actor: | Administrator | Secondary Actors: |  |
| Trigger: | Administrator logs in and accesses the subject management interface. | | |
| Description: | This use case describes how an Administrator views a list of subjects within the system. | | |
| Preconditions: | The Administrator is authenticated and authorized to access the subject management functionality.  Subjects are already defined in the system. | | |
| Postconditions: | The subject list is displayed to the Administrator. | | |
| Normal Flow: | Administrator selects "manage subject" from the dashboard.  The system displays a paginated list of all subjects. | | |
| Alternative Flows: | N/A | | |
| Exceptions: | If the Administrator lacks proper authorization, access to subject management is denied. | | |
| Priority: | High | | |
| Frequency of Use: | Frequent | | |
| Business Rules: | * Only authorized users should be able to access the subject list. * Subjects should be organized and categorized for easy navigation. * The system should provide an efficient and responsive way to view subjects, especially when there is a large number of them. | | |
| Other Information: | The "View Subject List" feature may be linked to other parts of the Issue Management System, such as issue creation or editing, to allow users to associate subjects with specific issues. | | |
| Assumptions: | * Users have the necessary training to understand and use the "View Subject List" feature. * Subjects are managed and updated by authorized personnel within the organization. | | |

#### UC002\_ Add new subject

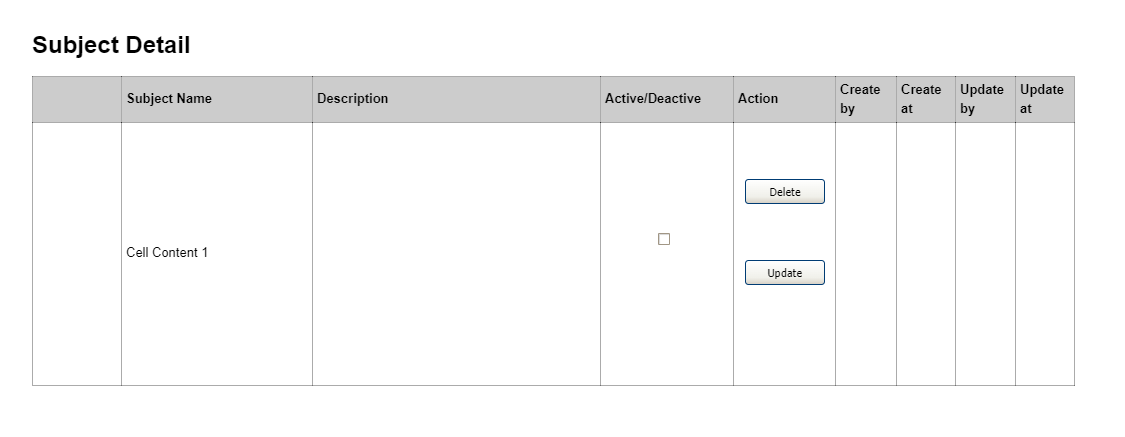
| UC ID and Name: | UC002 - Add new Subject List | | |
| --- | --- | --- | --- |
| Created By: | DucVD | Date Created: | 24/9/2023 |
| Primary Actor: | Administrator | Secondary Actors: |  |
| Trigger: | Administrator selects "Add New Subject" from the subject management menu. | | |
| Description: | This use case describes how an Administrator adds a new subject to the system. | | |
| Preconditions: | The Administrator is authenticated and authorized to access the subject management functionality. | | |
| Postconditions: | The new subject is added to the system. | | |
| Normal Flow: | Administrator selects "Add New Subject" from the subject management menu.  The system presents a form to input subject details.  Administrator fills in the required fields (name and description) and submits the form.  The system validates the data.  If the data is valid, the system adds the new subject to the database. | | |
| Alternative Flows: | N/A | | |
| Exceptions: | If there is an issue with data validation during the addition of a subject, an error message is displayed.  If the Administrator lacks proper authorization, access to subject management is denied. | | |
| Priority: | High | | |
| Frequency of Use: | Low to Moderate | | |
| Business Rules: | * Users with the necessary permissions should be able to add new subjects. * The system should enforce rules for naming and categorizing subjects to maintain consistency. * New subjects should be adequately documented with descriptions and attributes. | | |
| Other Information: | The "Add New Subject List" feature may be closely integrated with other parts of the Issue Management System, such as issue creation and editing, allowing users to easily associate newly created subjects with issues. | | |
| Assumptions: | * Users have the required training to understand and use the "Add New Subject List" feature. * The organization encourages users to keep the subject list well-structured and up-to-date. | | |

#### 2.1.2 Subject Detail

##### 2.1.1.2 View Subject Details

###### **a. UI Specifications**

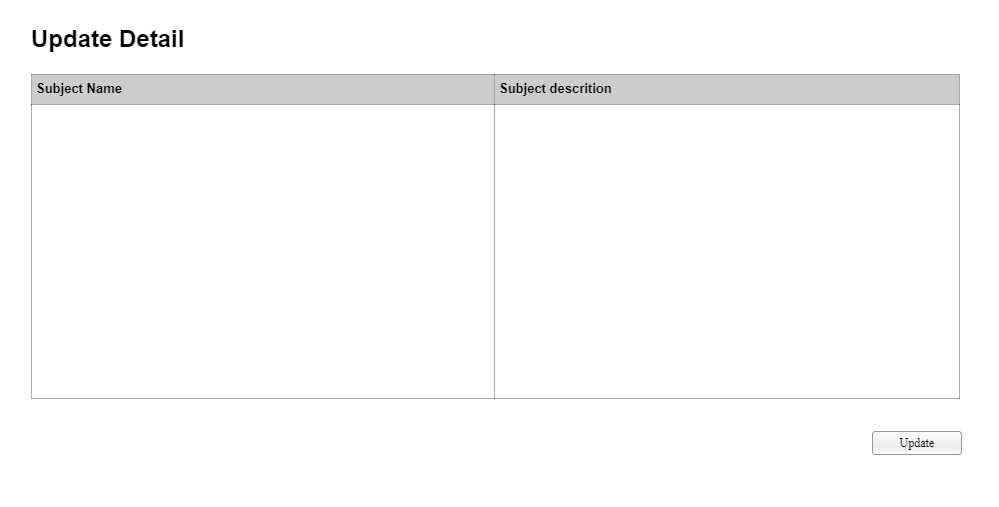
### View Subject Details



| Field Name | Field Type | Description |
| --- | --- | --- |
| *Field Group Name* | | |
| Subject Name | Text | Provide Subject name |
| Description | Text | Provide the description of the subject |
| Active/Deactive | Checkbox | Active or deactivate the subject |
| Delete | Button | Delete the subject |
| Update | Button | Go to the update subject screen |

### 

### Update Subject



### 

| Field Name | Field Type | Description |
| --- | --- | --- |
| *Field Group Name* | | |
| Subject Name | Text | Enter subject name |
| Subject Description | Text | Enter Description |
| Update | Button | Update the subject detail |

### 

###### **b. UC Specifications**

#### 2.1.2 UC002\_ View Subject Detail

| UC ID and Name: | UC002 - View Subject Detail | | |
| --- | --- | --- | --- |
| Created By: | DucVD | Date Created: | 24/9/2023 |
| Primary Actor: | Administrator | Secondary Actors: |  |
| Trigger: | Administrator selects a subject from the subject list | | |
| Description: | This use case describes how an Administrator views a list of subjects within the system. | | |
| Preconditions: | The Administrator is authenticated and authorized to access the subject management functionality.  Subjects are already defined in the system.  The Administrator has selected a subject from the subject list. | | |
| Postconditions: | The detailed information of the selected subject is displayed to the Administrator. | | |
| Normal Flow: | Administrator selects a subject from the list. The system displays detailed information about the selected subject, including name, description, activation status, and other relevant attributes. | | |
| Alternative Flows: | N/A | | |
| Exceptions: | If the Administrator lacks proper authorization, access to subject management is denied.  If the selected subject does not exist or has been deleted, an error message is displayed. | | |
| Priority: | High | | |
| Frequency of Use: | Frequent | | |
| Business Rules: | * Access to subject details should be restricted to authorized users. * Subject details should include all relevant information such as name, description, attributes, and a list of associated issues. * The system should provide an organized and user-friendly interface for viewing subject details. | | |
| Other Information: | * The "View Subject Detail" feature may offer options for users to take actions related to the subject, such as editing or linking it to additional issues. * It may include options for exporting subject details for reporting and analysis purposes. | | |
| Assumptions: | * Users have the necessary training to understand and use the "View Subject Detail" feature. * Subjects and their associated issues are managed and updated by authorized personnel within the organization. | | |

#### 2.1.4 UC004\_ Delete subject

| UC ID and Name: | UC004 - Delete Subject | | |
| --- | --- | --- | --- |
| Created By: | DucVD | Date Created: | 24/9/2023 |
| Primary Actor: | Administrator | Secondary Actors: |  |
| Trigger: | Administrator selects a subject from the subject list and chooses "Delete Subject." | | |
| Description: | This use case describes how an Administrator deletes a subject from the system. | | |
| Preconditions: | The Administrator is authenticated and authorized to access the subject management functionality.  Subjects are already defined in the system. | | |
| Postconditions: | The selected subject is deleted from the system.. | | |
| Normal Flow: | Administrator selects a subject from the list.  The system displays the subject's details.  Administrator chooses "Delete Subject."  The system displays a confirmation dialog.  If confirmed, the system deletes the subject. | | |
| Alternative Flows: | N/A | | |
| Exceptions: | If the Administrator lacks proper authorization, access to subject management is denied.  If the selected subject does not exist or has been deleted, an error message is displayed. | | |
| Priority: | High | | |
| Frequency of Use: | Low to Moderate | | |
| Business Rules: | * Access to subject deletion should be restricted to authorized users to prevent accidental or unauthorized removal. * The system may require users to confirm their intent to delete a subject to minimize accidental deletions. | | |
| Other Information: | * Users may be given the option to reassign associated issues to another subject or delete them when deleting a subject. * The "Delete Subject" feature should be carefully designed to minimize the risk of accidental deletions, as removing a subject can impact the organization's issue categorization. | | |
| Assumptions: | * Users have the necessary training to understand and use the "Delete Subject" feature responsibly. * Users have a clear understanding of when it is appropriate to delete a subject and how it may impact associated issues. | | |

#### 2.1.4 UC004\_ Active subject

| UC ID and Name: | UC004 - Active Subject | | |
| --- | --- | --- | --- |
| Created By: | DucVD | Date Created: | 24/9/2023 |
| Primary Actor: | Administrator | Secondary Actors: |  |
| Trigger: | Administrator selects a subject from the subject list and chooses "Activate Subject." | | |
| Description: | This use case describes how an Administrator activates an inactive subject in the system. | | |
| Preconditions: | The Administrator is authenticated and authorized to access the subject management functionality.  Subjects are already defined in the system. | | |
| Postconditions: | The selected subject is activated. | | |
| Normal Flow: | Administrator selects a subject from the list.  The system displays the subject's details.  Administrator chooses "Activate Subject."  The system toggles the activation status of the subject. | | |
| Alternative Flows: | N/A | | |
| Exceptions: | If the Administrator lacks proper authorization, access to subject management is denied.  If the selected subject does not exist or has been deleted, an error message is displayed. | | |
| Priority: | High | | |
| Frequency of Use: | Low to Moderate | | |
| Business Rules: | * Access to the "Active Subject" feature should be restricted to authorized users with relevant permissions to prevent unauthorized changes. * The system should maintain an audit trail or history of subject status changes for accountability and documentation | | |
| Other Information: | * Inactive subjects may be hidden from the main subject list view, reducing clutter while keeping historical subject data accessible. * Users should have a clear understanding of when it is appropriate to mark a subject as active or inactive and how this change impacts issue management. | | |
| Assumptions: | * Users have the necessary training to understand and use the "Active Subject" feature responsibly. * Users have a clear understanding of when it is appropriate to change the status of a subject and how this change may impact the organization's issue management workflow. | | |

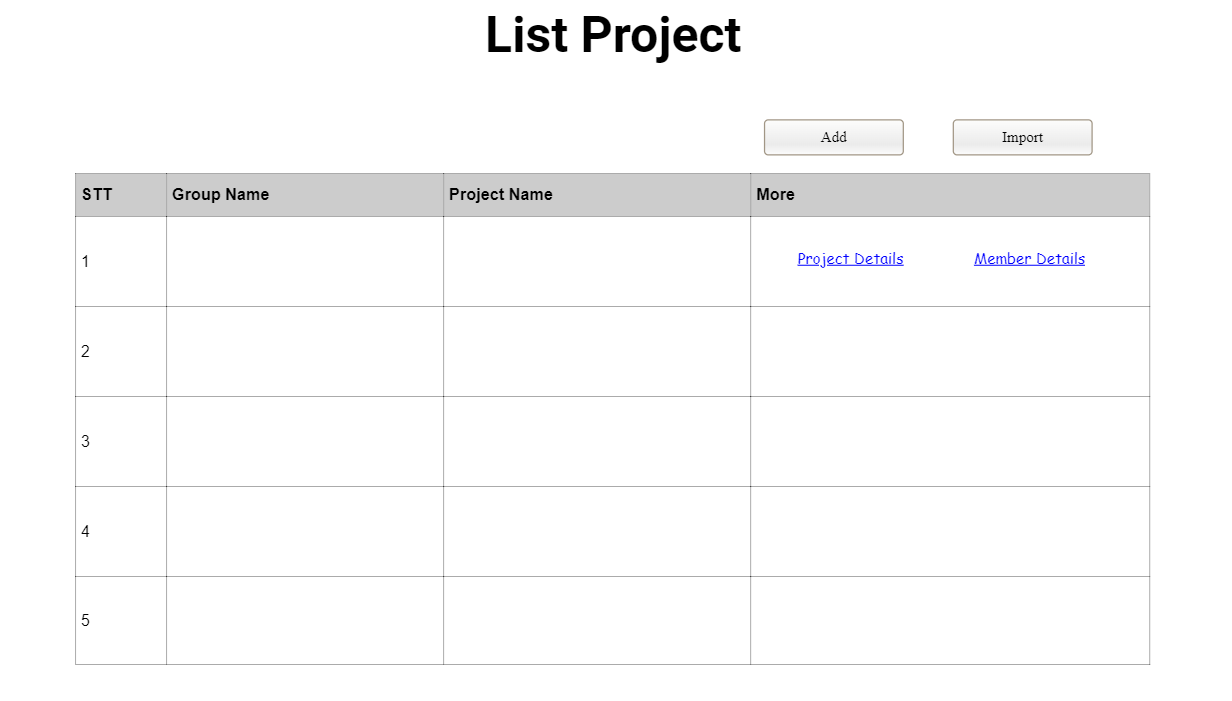
### 2.2 Teacher Feature

#### 2.2.1 Project Setting

##### 2.2.1.1 Project view

###### **a. UI Specifications**

This screen allows the teacher can see list project when logged successfully



#### 

| Field Name: | Field Type | Description: |
| --- | --- | --- |
| View list project |  |  |
| STT | Text | This is show column orders |
| Group Name | Text | This is show name of group |
| Project Name | Text | This is show name of project |
| More | Text | Include 2 hyperlink |
| Project Details | Hyperlink | Teacher clicks to hyperlink and will be redirected to project details page |
| Member Details | Hyperlink | Teacher clicks to hyperlink and will be redirected to member details page |
| Add | Button | Allow teacher add more project to list |
| Import | Button |  |

### 

###### **b. UC Specifications**

***UC\_09: View List project***

#### 

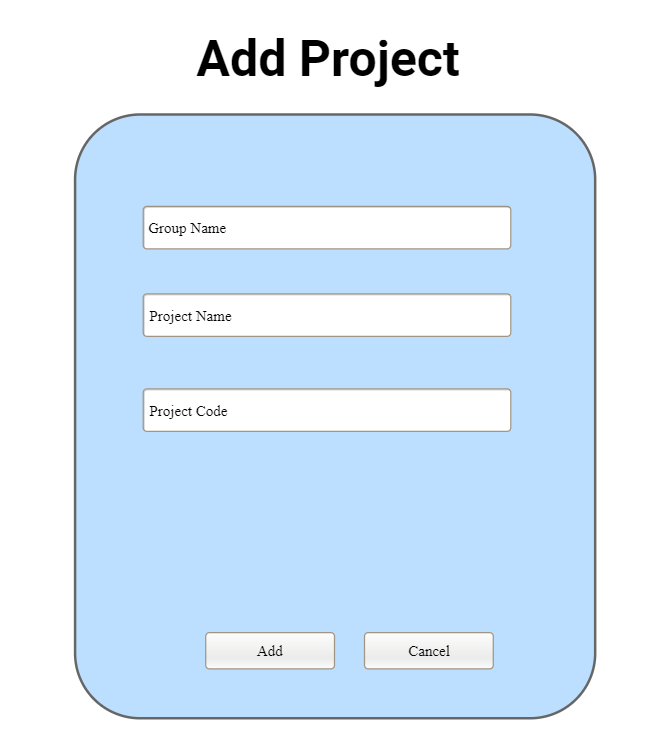
| UC ID and Name: | UC -001 \_View list project | | |
| --- | --- | --- | --- |
| Created By: | PhongVD | Date Created: | 25/19/2023 |
| Primary Actor: | Teacher | Secondary Actors: | None |
| Trigger: | The teacher wants to view the list of project teams and the members within each team. | | |
| Description: | This use case outlines the steps for the teacher to view the list of project teams and the members associated with each team. | | |
| Preconditions | * The teacher must be logged into the Project Management System. | | |
| Postconditions: | The teacher is presented with a list of project teams, and for each team, the system displays the list of members.. | | |
| Normal Flow: | * The teacher logs into the Project Management System. * The teacher navigates to the "Manage Projects" section. * The teacher selects the option to "View List of Project Teams and Members." * The system retrieves and displays a list of project teams. * For each project team, the system displays the list of members, including their names and roles. | | |
| Alternative Flows: | None | | |
| Exceptions: | None | | |
| Priority: | High | | |
| Frequency of Use: | Often | | |
| Business Rules: | * Only authorized users, such as the teacher, can access and view the list of project teams and their members. * The list of project teams and members is dynamically generated from the system's data. | | |
| Other Information: | This use case is essential for the teacher to have an overview of the project teams and the distribution of members, which aids in team management and resource allocation. | | |

#### 

##### 2.2.1.2 <<Add project>>

###### **a. UI Specifications**

This screen allow the teacher add more project



#### 

| Field Name: | Field Type | Description: |
| --- | --- | --- |
| Add project |  |  |
| Group Name | TextBox | Enter group name |
| Project Name | TextBox | Enter project name |
| Project Code | TextBox | Enter project code |
| Add | Button | Add project to list |
| Cancel | Button | Cancel add project |

### 

###### **b. UC Specifications**

#### UC\_08: Add New Project

#### 

| UC ID and Name: | UC -008\_Add New Project | | |
| --- | --- | --- | --- |
| Created By: | PhongVD | Date Created: | 25/9/2023 |
| Primary Actor: | Teacher | Secondary Actors: | None |
| Trigger: | the teacher, who has successfully logged into the system and selected their assigned class, chooses the option to add a new project to the class.. | | |
| Description: | This use case describes how a teacher can create and add a new project to their assigned class. During this process, the teacher can select waiting class students to participate in the new project. | | |
| Preconditions: | * The teacher has successfully logged into the system. * The teacher is assigned to a class | | |
| Postconditions: | A new project is created within the class, and the selected students are allocated to the project. | | |
| Normal Flow: | * The teacher logs in to the system and selects the class they are responsible for. * The teacher chooses the option to add a new project. * The system presents a form for creating the new project, including fields for group name, project code, project name, and description. * The teacher fills out the necessary information for the new project. * The teacher selects waiting class students from a list to participate in the new project. * The teacher confirms the creation of the new project. * The system creates the project, assigns the selected students to it, and saves the project details. | | |
| Alternative Flows: | None | | |
| Exceptions: | * If there are issues with creating the new project (e.g., missing information), the system prompts the teacher to correct the issues before saving. * If the teacher decides to cancel the creation of the new project, they can do so without saving the project. | | |
| Priority: | High | | |
| Frequency of Use: | Occasional | | |
| Business Rules: | Teachers should be able to create new projects within their assigned class and allocate students to them. | | |
| Other Information: | This use case is essential for teachers to initiate new projects and manage student allocations. | | |
| Assumptions: | * The teacher has the necessary permissions to add new projects. * Students are available and waiting for assignments within the teacher's class. | | |

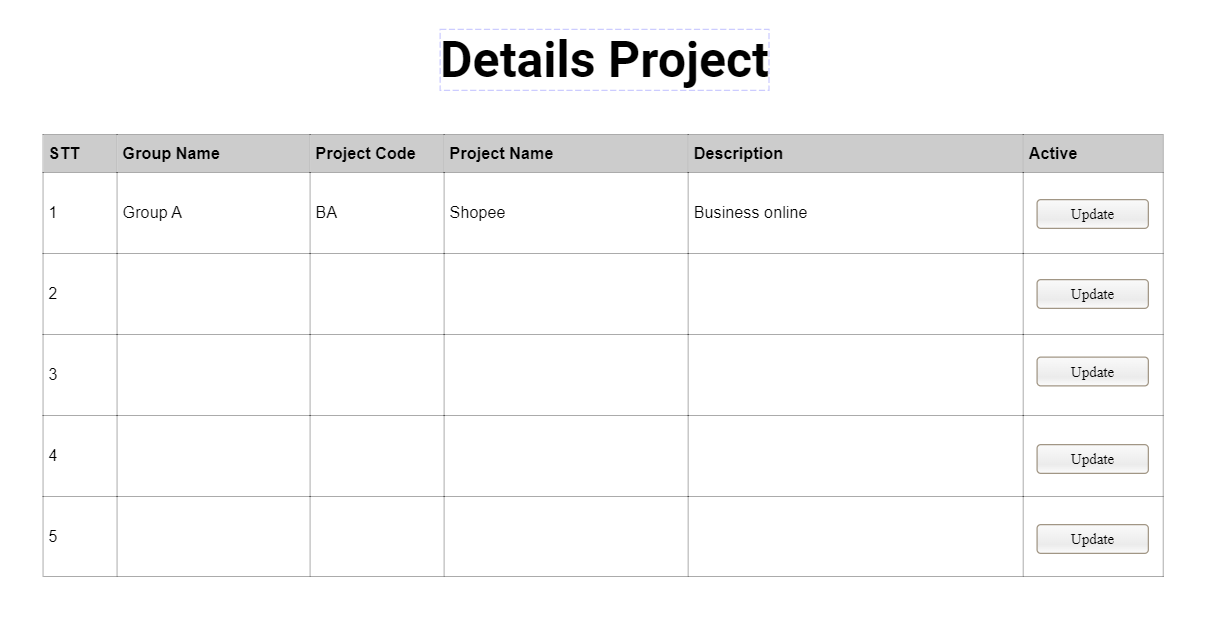
#### 

#### 2.2.2 Detail Project

##### 2.2.2. Project Detail View

###### **a. UI Specifications**

This screen show details of projects



#### 

| Field Name: | Field Type | Description: |
| --- | --- | --- |
| View Project Details |  |  |
| STT | Text | This is show column orders: |
| Group Name | Text | This is show name of group |
| Project Code | Text | This is show code of project |
| Project Name | Text | This is show name of project |
| Description | Text | This is show description of project |
| Active | Text | Including button |
| Update | Button | Edit project |

### 

###### **b. UC Specifications**

#### 

#### UC\_06: View Details of a Project

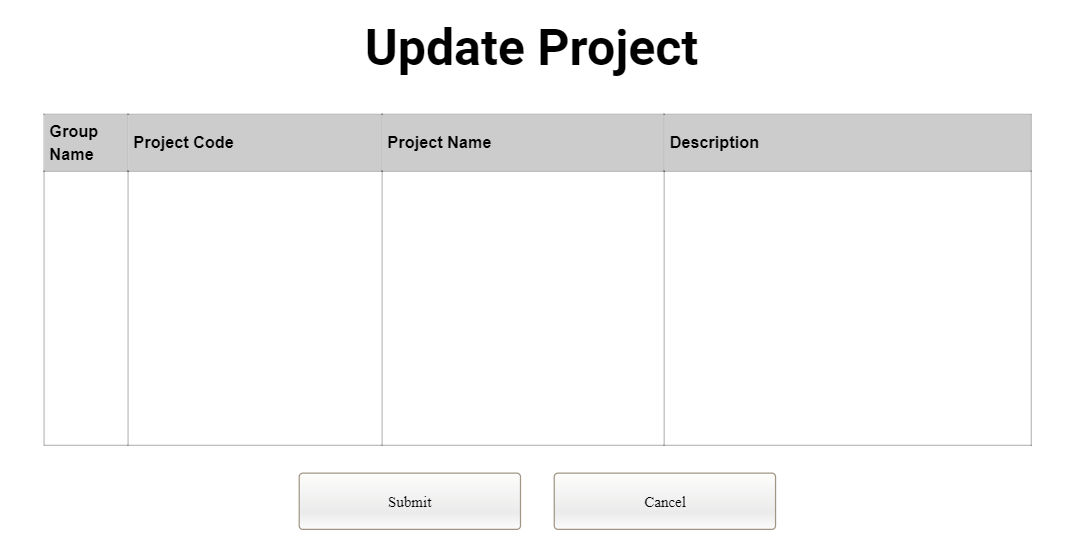
| UC ID and Name: | UC -006 \_View Details of a Project | | |
| --- | --- | --- | --- |
| Created By: | PhongVD | Date Created: | 25/9/2023 |
| Primary Actor: | Teacher | Secondary Actors: | None |
| Trigger: | The teacher has successfully logged into the system and selected their assigned class, chooses a specific project from the list of projects within the class and selects the option to view the details of the chosen project. | | |
| Description: | This use case describes how a teacher can view the details of a specific project within their assigned class. | | |
| Preconditions: | * The teacher has successfully logged into the system. * The teacher is assigned to a class. * The selected project exists within the class. | | |
| Postconditions: | The teacher can view the details of the selected project, including group name, project code, project name, and description. | | |
| Normal Flow: | * The teacher logs in to the system and selects the class they are responsible for. * The system displays the list of projects within the class. * The teacher clicks on a project in the list to select it. * The teacher chooses the option to view the details of the project. * The system retrieves and displays the project details, including group name, project code, project name, and description. | | |
| Alternative Flows: | None | | |
| Exceptions: | If the selected project does not exist or cannot be accessed for any reason, the system displays an error message. | | |
| Priority: | High | | |
| Frequency of Use: | Frequent | | |
| Business Rules: | Teachers should be able to view the details of any project within their assigned class. | | |
| Other Information: | This use case is crucial for teachers to understand the specifics of individual projects. | | |
| Assumptions: | * The teacher has the necessary permissions to access project details. * Projects exist within the teacher's class. | | |

#### 

##### 2.2.2.2 Update project

###### **a. UI Specifications**

This screen allows the teacher can edit project



#### 

| Field Name: | Field Type | Description: |
| --- | --- | --- |
| Update project |  |  |
| Group Name | TextBox | Enter to edit name of group |
| Project Code | TextBox | Enter to edit code of project |
| Project Name | TextBox | Enter to edit name of project |
| Description | TextBox | Enter to edit description of project |
| Submit | Button | Submit to update project |
| Cancel | Button | Cancel edit project |

### 

###### **b. UC Specifications**

#### 

| UC ID and Name: | UC -007 \_Update details of a project | | |
| --- | --- | --- | --- |
| Created By: | PhongVD | Date Created: | 25/9/2023 |
| Primary Actor: | Teacher | Secondary Actors: | None |
| Trigger: | when the teacher, having successfully logged into the system and selected their assigned class, chooses a specific project from the list of projects within the class and selects the option to update the details of the chosen project. | | |
| Description: | This use case describes how a teacher can update the details of a specific project within their assigned class, including the group name, project code, project name, and description. | | |
| Preconditions: | * The teacher has successfully logged into the system. * The teacher is assigned to a class. * The selected project exists within the class. | | |
| Postconditions: | The project's details, including group name, project code, project name, and description, are updated in the system. | | |
| Normal Flow: | * The teacher logs in to the system and selects the class they are responsible for. * The system displays the list of projects within the class. * The teacher clicks on a project in the list to select it. * The teacher chooses the option to update the details of the project. * The system presents a form allowing the teacher to edit the project's details, including group name, project code, project name, and description. * The teacher makes the desired changes to the project's details. * The teacher confirms the changes. * The system updates the project's details in the system. | | |
| Alternative Flows: | None | | |
| Exceptions: | * If the selected project does not exist or cannot be accessed for any reason, the system displays an error message. * If there are issues with updating the project details (e.g., invalid data), the system prompts the teacher to correct the issues before saving. | | |
| Priority: | High | | |
| Frequency of Use: | Occasional | | |
| Business Rules: | Teachers should be able to modify the details of any project within their assigned class. | | |
| Other Information: | This use case is crucial for teachers to keep project information up-to-date and aligned with class requirements. | | |
| Assumptions: | * The teacher has the necessary permissions to update project details. * Projects exist within the teacher's class. | | |

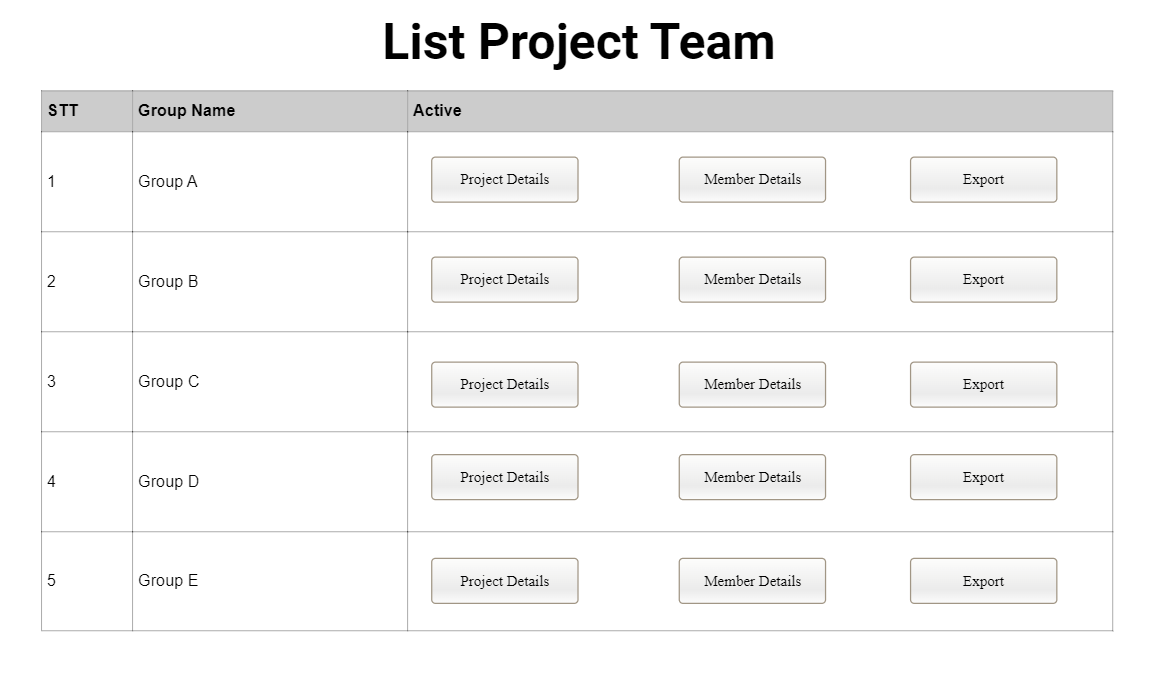
#### 2.2.3 Project Team View

##### 2.2.3.1 <<View Project Team And Member>>

##### 

###### **a. UI Specifications**

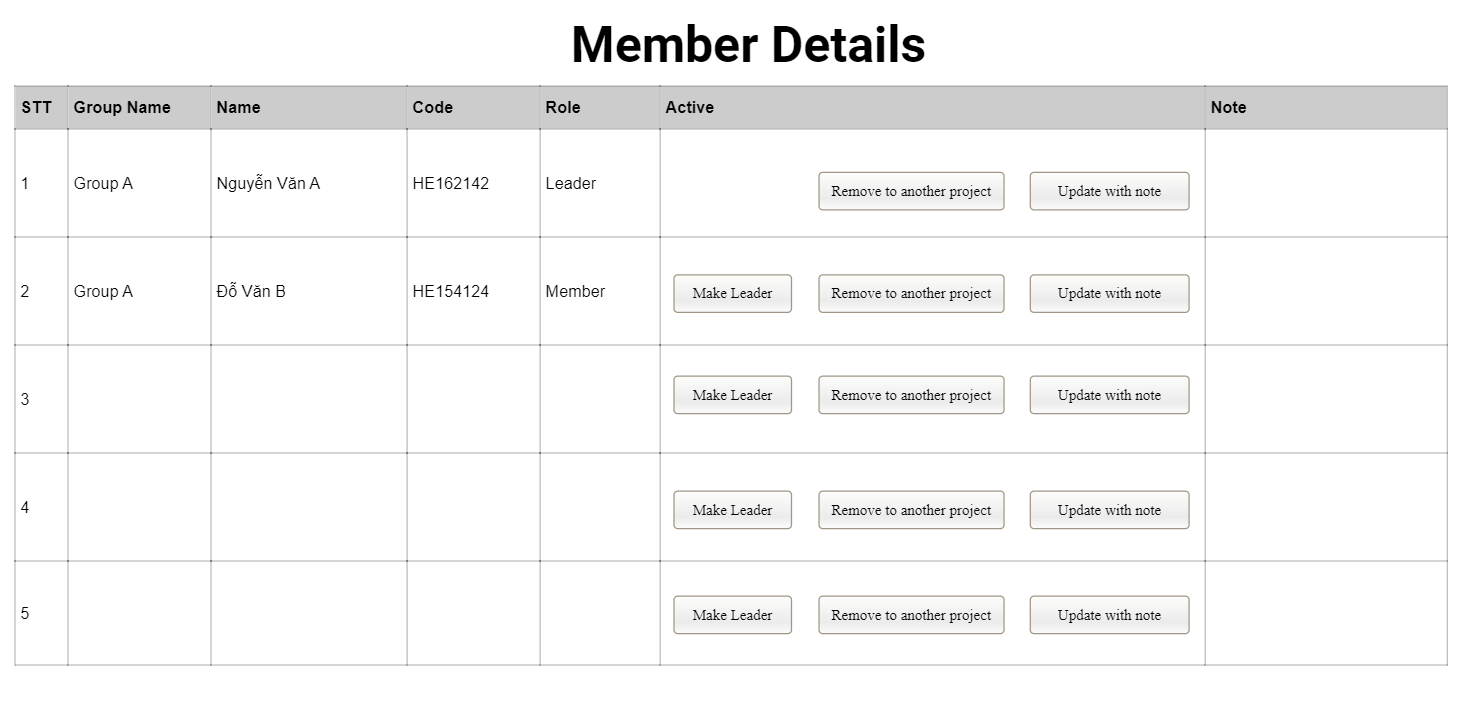
This screen allows the teacher can see list project team and members



#### 

| Field Name: | Field Type | Description: |
| --- | --- | --- |
| View Project Team |  |  |
| STT | Text | This is show column orders |
| Group Name | Text | This is show name of group |
| Active | Text | Including 3 button |
| Project Details | Button | The teacher click the button then switches to the project details page |
| Member Details | Button | The teacher click the button then switches to the member details page |
| Export | Button | The teacher click the button then export to file |

### 



#### 

| Field Name: | Field Type | Description: |
| --- | --- | --- |
| View Member Details |  |  |
| STT | Text | This is show column orders |
| Group Name | Text | This is show name of group |
| Name | Text | This is show name of member |
| Code | Text | This is show code of member |
| Active | Text | Including 3 button |
| Note | Text | This is show note by teacher for member |
| Make leader | Button | Choose the leader of group |
| Remove to another project | Button | Remove member to other project |
| Update with note | Button | Update by write text for member |

#### UC002\_Export Currently Team's Member List

#### 

| UC ID and Name: | UC -002 \_Export Currently Team's Member List | | |
| --- | --- | --- | --- |
| Created By: | PhongVD | Date Created: | 25/19/2023 |
| Primary Actor: | Teacher | Secondary Actors: | None |
| Trigger: | The teacher decides to export the current team's member list | | |
| Description: | This use case describes the steps for the Project Manager to export the list of members | | |
| Preconditions | * The teacher must be logged into the Project Management System. * At least one project team must exist in the system. | | |
| Postconditions: | The member list for the specified project team is successfully exported. | | |
| Normal Flow: | * The teacher logs into the Project Management System. * The teacher navigates to the "Manage Projects" section. * The teacher selects the project team for which they want to export the member list. * The teacher chooses the option to export the member list, specifying that it should include UC ID and Name. * The system generates and provides the exported list in a preferred format. | | |
| Alternative Flows: | None | | |
| Exceptions: | * If there are no members in the selected project team, a notification is displayed, and no export is performed. * If the system encounters technical difficulties during the export process, an error message is shown, and the export is aborted. | | |
| Priority: | Medium | | |
| Frequency of Use: | Sometimes | | |
| Business Rules: | * Only authorized users, such as the Project Manager, can perform this export operation. * The exported list must contain the UC ID and Name of each member in the selected project team. * The exported file format can vary based on system requirements and user preferences. | | |
| Other Information: | The exported member list can be valuable for reporting, team communication, and tracking project resources. | | |

#### 

#### UC004\_Make a project member as the only team leader

#### 

| UC ID and Name: | UC -004 Make a project member as the only team leader | | |
| --- | --- | --- | --- |
| Created By: | PhongVD | Date Created: | 25/19/2023 |
| Primary Actor: | Teacher | Secondary Actors: | None |
| Trigger: | The teacher wants to designate a member within a project team as the Team Leader. | | |
| Description: | This use case outlines the process of assigning a project member as the only team leader for a project within the project management system. | | |
| Preconditions | * The teacher must be logged . * The teacher has the necessary permissions to make this designation. * The project exists.. | | |
| Postconditions: | The designated member is successfully assigned as the sole team leader for the project. | | |
| Normal Flow: | * The teacher logs into the project management system. * The teacher navigates to the "Manage Projects" section. * The teacher selects the project for which they want to assign a team leader. * The teacher selects the project member they want to make the only team leader. * The teacher chooses the "Assign as Team Leader" action. * The system prompts the teacher to confirm the action. * Upon confirmation, the system updates the project's team leader to the selected member. | | |
| Alternative Flows: | * If the selected member is already a team leader for the project or is part of a leadership team, the system should provide a warning or error message and ask the user to confirm the change. * If the teacher does not have the necessary permissions, the system should deny the action and display an error message. | | |
| Exceptions: | If there are technical issues, such as server errors, the system should display a generic error message and allow the user to retry or contact support. | | |
| Priority: | High | | |
| Frequency of Use: | Often | | |
| Business Rules: | * Only teachers can initiate this action. * A project can have only one team leader, and designating a new one should replace the existing team leader. | | |
| Other Information: | * This use case is integral to maintaining an organized project structure and ensuring clear leadership roles within each project. It is crucial for effective project management and team coordination. | | |

###### **b. UC Specifications**

#### UC001\_View list of project teams and members

#### 

| UC ID and Name: | UC -001 \_View list project teams and members | | |
| --- | --- | --- | --- |
| Created By: | PhongVD | Date Created: | 25/19/2023 |
| Primary Actor: | Teacher | Secondary Actors: | None |
| Trigger: | The teacher logged into the system, accesses the system's project management section specifically for the class they are responsible for. | | |
| Description: | This use case describes how a teacher can view a list of projects team and members teams within their assigned class. | | |
| Preconditions | * The teacher has successfully logged into the system. * The teacher is assigned to a class. * Projects and students exist within the class. | | |
| Postconditions: | The teacher can see a list of project teams and members within the class. | | |
| Normal Flow: | * The teacher logs in to the system and selects the class they are responsible for. * The system displays the list of projects and members within the class, including details such as group names, project codes, project names, and descriptions,members, name, members codes,... | | |
| Alternative Flows: | None | | |
| Exceptions: | If there are technical issues with retrieving the list of project teams and members, the system displays an error message, and the teacher may need to retry or contact support. | | |
| Priority: | High | | |
| Frequency of Use: | Often | | |
| Business Rules: | The list of project teams and members should include only the teams within the teacher's assigned class. | | |
| Other Information: | This use case is essential for teachers to get an overview of project teams and members within their class. | | |

#### 

#### 

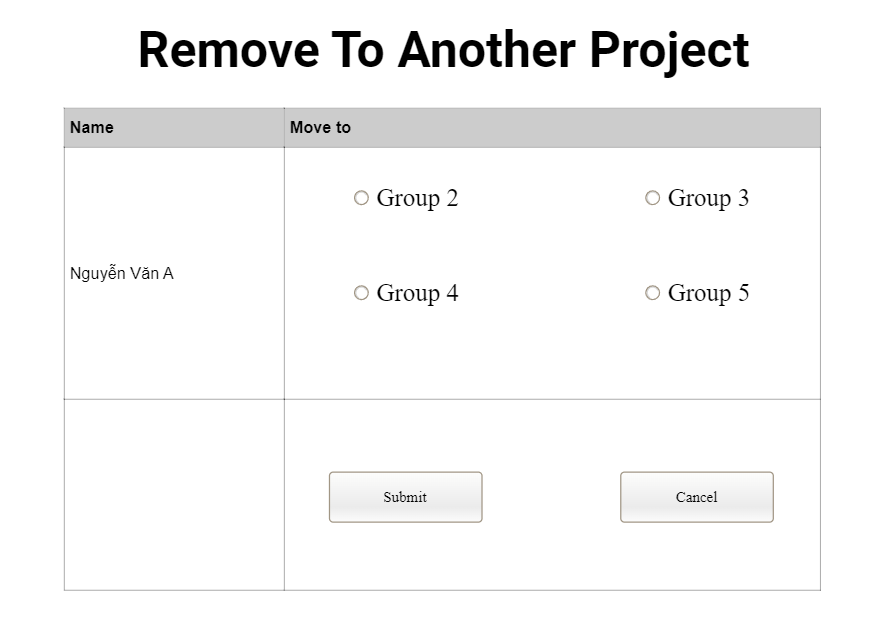
##### 2.2.3.2 <<Remove to another project>>

###### **a. UI Specifications**

This screen allows the teacher can remove member to another project

#### 

#### 



#### 

| Field Name: | Field Type | Description: |
| --- | --- | --- |
| Remove to another project |  |  |
| Name | Text | This is show name of member |
| Move to | Text | Including radio button |
| Name group | Radio Button | Select the group you want to remove |
| Submit | Button | Submit to remove |
| Cancel | Button | Cancel to remove |

###### 

###### **b. UC Specifications**

#### UC003\_Remove or Move a Member from One Project to Another Project

#### 

#### 

| UC ID and Name: | UC -003 \_Remove or Move a Member from One Project to Another Project | | |
| --- | --- | --- | --- |
| Created By: | PhongVD | Date Created: | 25/19/2023 |
| Primary Actor: | Teacher | Secondary Actors: | None |
| Trigger: | The teacher needs to reassign a project team member to a different project or remove them from their current project. | | |
| Description: | This use case describes the process of removing or moving a member from one project to another project within the project management system.. | | |
| Preconditions | * The user must be logged in as a Teacher. * The user has appropriate permissions to make these changes. * The target project exists. | | |
| Postconditions: | The member is successfully removed from the source project and assigned to the target project, or they are completely removed from the system. | | |
| Normal Flow: | * The teacher logs into the project management system. * The teacher navigates to the "Manage Projects" section. * The teacher selects the source project from which they want to remove or move a member. * The teacher selects the member they want to remove or move. * The teacher chooses the "Remove" or "Move" action. * If the "Move" action is chosen:   + The teacher selects the target project for the member. * The teacher confirms the action. * The system updates the member's assignment to the project as per the user's request. | | |
| Alternative Flows: | None | | |
| Exceptions: | If there are technical issues, such as server errors, the system should display a generic error message and allow the user to retry or contact support. | | |
| Priority: | High | | |
| Frequency of Use: | Often | | |
| Business Rules: | * Only teachers can initiate this action. * Team leaders cannot be removed from a project without appointing a new team leader. | | |
| Other Information: | * This use case is part of a larger system that manages project teams, members, and their assignments. It is essential for maintaining an up-to-date and efficient project management environment. | | |

#### 

##### 2.2.3.4 <<Update with note>>

##### 

###### **a. UI Specifications**

This screen allows the teacher can write note for member



#### 

| Field Name: | Field Type | Description: |
| --- | --- | --- |
| Update with note |  |  |
| Text | TextBox | Enter note |
| Submit | Button | Submit to note for member |
| Cancel | Button | Cancel to note for member |

###### 

###### **b. UC Specifications**

#### UC005\_Update a Team Member's Status with Note

#### 

| UC ID and Name: | UC -005 Update a Team Member's Status with Note | | |
| --- | --- | --- | --- |
| Created By: | PhongVD | Date Created: | 25/19/2023 |
| Primary Actor: | Teacher | Secondary Actors: | None |
| Trigger: | The need to update the status of a team member and attach a note to provide additional context or information. | | |
| Description: | This use case defines the process of updating a team member's status and adding a note to their profile within the project management system. | | |
| Preconditions | * The teacher must be logged . * The teacher has the necessary permissions to update team member statuses. * The team member's profile exists within the system. | | |
| Postconditions: | The team member's status is successfully updated, and the note is attached to their profile. | | |
| Normal Flow: | * The teacher logs into the project management system. * The teacher navigates to the "Manage Team" or "Team Members" section. * The teacher selects the team member whose status they want to update. * The teacher chooses the "Update Status" action. * The teacher selects the new status from a predefined list (e.g., Active, Inactive, On Leave, In Progress). * The teacher enters a note in a designated field to provide additional context or information related to the status change. * The teacher confirms the action. * The system updates the team member's status and attaches the note to their profile. | | |
| Alternative Flows: | * If the selected status does not align with the predefined options, the system should prompt the user to choose a valid status. * If the user does not have the necessary permissions, the system should deny the action and display an error message. | | |
| Exceptions: | If there are technical issues, such as server errors, the system should display a generic error message and allow the user to retry or contact support. | | |
| Priority: | Medium | | |
| Frequency of Use: | Usually | | |
| Business Rules: | * Only teachers can initiate this action. * Status options should be predefined to maintain consistency. | | |
| Other Information: | * This use case is essential for keeping the team member information up to date and for providing context on their status changes. It contributes to effective communication and management within the project team. | | |

#### 

#### UC\_09: Select Waiting Class Students into Project

#### 

| UC ID and Name: | UC -009\_Select Waiting Class Students into Project | | |
| --- | --- | --- | --- |
| Created By: | PhongVD | Date Created: | 25/9/2023 |
| Primary Actor: | Teacher | Secondary Actors: | None |
| Trigger: | The teacher, during the process of adding a new project or updating an existing one, reaches the stage where they need to select students for the project, and they choose to select waiting class students from the available list for project allocation. | | |
| Description: | This use case describes how a teacher can select students from a list of waiting class students and allocate them to a specific project within their assigned class. | | |
| Preconditions: | * The teacher has successfully logged into the system. * The teacher is in the process of adding a new project or updating an existing one. * Waiting class students are available for assignment within the class. | | |
| Postconditions: | Selected waiting class students are allocated to the project. | | |
| Normal Flow: | * The teacher is either creating a new project or updating an existing one. * The teacher reaches the stage in the process where they need to select students for the project. * The system provides a list of waiting class students who are available for assignment. * The teacher selects the desired students from the list to participate in the project. * The teacher confirms the selection. * The system allocates the selected students to the project. | | |
| Alternative Flows: | None | | |
| Exceptions: | * If there are no waiting class students available for assignment, the system informs the teacher and suggests checking the class's student enrollment status. | | |
| Priority: | Medium | | |
| Frequency of Use: | Occasional | | |
| Business Rules: | Teachers should be able to select waiting class students and allocate them to specific projects within their assigned class. | | |
| Other Information: | This use case is crucial for teachers to form project teams by selecting the appropriate students. | | |
| Assumptions: | * The teacher has the necessary permissions to select waiting class students for project allocation. * Waiting class students are available and waiting for assignments within the teacher's class. | | |

—-----------------------------------

### 2.3 Student Feature

#### 2.3.1 Issue List

###### **a. UI Specifications**

### 

#### 

| Field Name: | Field Type | Description: |
| --- | --- | --- |
| Issue List |  |  |
| Issue ID | Text | show column orders |
| Title | Text | show the title of the issue |
| Status | Text | Show status of the issue |

### 

###### **b. UC Specifications**

#### UC-1\_View list issue

| ID and Name: | **UC-1 View list issue** | | |
| --- | --- | --- | --- |
| Created By: | TruongNQ | Date Created: | 1/10/2023 |
| Primary Actor: | Student | Secondary Actors: | None |
| Description: | Student want to view list issue | | |
| Trigger: | The trigger for a student to use the "View List Issue" feature could be the need to check their course assignments, grades, or any academic-related issues. | | |
| Preconditions: | Preconditions for a student to use this feature may include having access to a student portal, being enrolled in courses, and being logged into their student account. | | |
| Postconditions: | Postconditions for a student using the "View List Issue" feature could include viewing a list of their upcoming assignments, seeing their current grades, or getting information about academic deadlines. | | |
| Normal Flow: | A student would typically log into their student portal, navigate to the relevant section (e.g., "My Courses" or "Grades"), and select the "View List Issue" feature. The system would then display the requested information. | | |
| Alternative Flows: | A student might involve scenarios like requesting additional details on a particular assignment, filtering the list of issues, or accessing the feature through a mobile app. | | |
| Exceptions: | include encountering errors when trying to access the feature due to technical issues or being denied access due to enrollment problems. | | |
| Priority: | High | | |
| Frequency of Use: | Often | | |
| Business Rules: | include validation checks to ensure the accuracy and consistency of batch updates, as well as authorization requirements for performing batch updates. | | |
| Other Information: | Additional information could include integration with other student tools (e.g., a learning management system), support resources for students using the feature, or any upcoming changes or updates to the feature. | | |
| Assumptions: | Assumptions made in a student context might include assuming that students have access to the necessary technology (e.g., a computer or smartphone) and that they have been properly enrolled in their courses. | | |

**Business Rules**

None

#### 2.3.2. Add new issue

###### **a. UI Specifications**

### 

### 

| Field Name: | Field Type | Description: |
| --- | --- | --- |
| Add new issue |  |  |
| Title | Text | the title of the issue |
| Description | Text | the detail of the issue |
| Status | list | Show the status of the issue |
| Submit | button | Submit the issue |

### 

###### **b. UC Specifications**

#### UC-2\_Add new issue

| ID and Name: | **UC-2 Add new issue** | | |
| --- | --- | --- | --- |
| Created By: | TruongNQ | Date Created: | 1/10/2023 |
| Primary Actor: | Student | Secondary Actors: | None |
| Description: | Student want to add new issue | | |
| Trigger: | Use the "Add New Issue" feature could be encountering a problem or issue related to their coursework, such as a technical issue with an online assignment submission or a question about their enrollment status. | | |
| Preconditions: | When use this feature may include having access to a student portal, being logged into their student account, and being currently enrolled in a course or program. | | |
| Postconditions: | Using the "Add New Issue" feature could include successfully submitting their issue or request, receiving a confirmation message, and having the issue recorded for administrative review or resolution. | | |
| Normal Flow: | * Log into the student portal. * Navigate to the "Support" or "Help Center" section. * Select the "Add New Issue" option. * Fill out a form that includes details about the issue, such as a description, relevant course information, and contact details. * Attach any necessary documents or screenshots. * Submit the issue. * Receive a confirmation message indicating that the issue has been submitted for review. | | |
| Alternative Flows: | The option to contact support through other means, such as email or phone, instead of using the online form.  In some cases, there may be a chatbot or virtual assistant available to assist with issue reporting. | | |
| Exceptions: | * If there are technical issues preventing a student from submitting the form, an error message should be displayed with troubleshooting instructions. * If the issue reporting form is unavailable due to maintenance, students should be notified in advance. | | |
| Priority: | High | | |
| Business Rules: | Business rules may include guidelines for issue categorization, response times, and escalation procedures for more urgent issues. | | |
| Other Information: | Additional information could include details about the support team responsible for handling reported issues, any service level agreements (SLAs) related to issue resolution, and available communication channels for follow-up. | | |
| Assumptions: | Assumptions made in a student context might include assuming that students have access to the necessary technology to submit issues and that they are familiar with the process of using the student portal. | | |

**Business Rules**

None

#### 2.3.3 Update Issue

###### **a. UI Specifications**

### 

| Field Name: | Field Type | Description: |
| --- | --- | --- |
| Update issue |  |  |
| Issue ID | Text | Show the column order |
| Title | Text | the title of the issue |
| Description | Text | the detail of the issue |
| Status | list | Show the status of the issue |
| Update | button | Update the issue |

### 

###### **b. UC Specifications**

#### UC-3\_Update issue

| ID and Name: | **UC-3 Update project issue** | | |
| --- | --- | --- | --- |
| Created By: | TruongNQ | Date Created: | 1/10/2023 |
| Primary Actor: | Student | Secondary Actors: | None |
| Description: | Student want to add new issue | | |
| Trigger: | Use the "Add New Issue" feature could be encountering a problem or issue related to their coursework, such as a technical issue with an online assignment submission or a question about their enrollment status. | | |
| Preconditions: | When use this feature may include having access to a student portal, being logged into their student account, and being currently enrolled in a course or program. | | |
| Postconditions: | Using the "Add New Issue" feature could include successfully submitting their issue or request, receiving a confirmation message, and having the issue recorded for administrative review or resolution. | | |
| Normal Flow: | * Log into the student portal. * Navigate to the "Support" or "Help Center" section. * Select the "Add New Issue" option. * Fill out a form that includes details about the issue, such as a description, relevant course information, and contact details. * Attach any necessary documents or screenshots. * Submit the issue. * Receive a confirmation message indicating that the issue has been submitted for review. | | |
| Alternative Flows: | The option to contact support through other means, such as email or phone, instead of using the online form.  In some cases, there may be a chatbot or virtual assistant available to assist with issue reporting. | | |
| Exceptions: | * If there are technical issues preventing a student from submitting the form, an error message should be displayed with troubleshooting instructions. * If the issue reporting form is unavailable due to maintenance, students should be notified in advance. | | |
| Priority: | High | | |
| Business Rules: | Business rules may include guidelines for issue categorization, response times, and escalation procedures for more urgent issues. | | |
| Other Information: | Additional information could include details about the support team responsible for handling reported issues, any service level agreements (SLAs) related to issue resolution, and available communication channels for follow-up. | | |
| Assumptions: | Assumptions made in a student context might include assuming that students have access to the necessary technology to submit updates, that they are aware of the process for requesting updates, and that they have reported an issue previously. | | |

#### 2.3.4 Batch Update

###### **a. UI Specifications**

### 

### 

| Field Name: | Field Type | Description: |
| --- | --- | --- |
| Batch Update |  |  |
| Issue1 | CheckBox | Show the update issue |
| Issue2 | CheckBox | Show the update issue |
| New Status | Text | the status of the issue |
| Update Selected Issue | Button | Update the issue |

### 

###### **b. UC Specifications**

#### UC-4\_Batch update

| ID and Name: | **UC-4 Batch update** | | |
| --- | --- | --- | --- |
| Created By: | TruongNQ | Date Created: | 1/10/2023 |
| Primary Actor: | Student | Secondary Actors: | None |
| Description: | This function provides information about how students can use this feature to update multiple pieces of academic or administrative information simultaneously. | | |
| Trigger: | To make changes to multiple course enrollments, update contact information, or perform other bulk updates to their academic records. | | |
| Preconditions: | This feature may include having access to a student portal, being logged into their student account, and having the necessary permissions to perform batch updates. | | |
| Postconditions: | Include successfully applying the requested updates to the specified records, receiving a confirmation message, and having the updated information reflected in their academic profile. | | |
| Normal Flow: | * Log into the student portal. * Navigate to the "Account Settings" or "Profile Management" section. * Select the "Batch Update" option. * Choose the type of updates to perform (e.g., course enrollments, contact information). * Provide the necessary details or data for the batch update (e.g., upload a CSV file with updated information). * Review and confirm the changes to be made. * Submit the batch update request. * Receive a confirmation message indicating that the batch update request has been received and will be processed. | | |
| Alternative Flows: | Provide options for students to preview changes before final submission.  Some batch updates may require administrative approval before they are applied. | | |
| Exceptions: | * If there are technical issues preventing a student from submitting the batch update request, an error message should be displayed with troubleshooting instructions. * If the batch update request cannot be processed due to validation errors or conflicts, students should be provided with guidance on resolving the issues. | | |
| Priority: | Often | | |
| Business Rules: | include validation checks to ensure the accuracy and consistency of batch updates, as well as authorization requirements for performing batch updates. | | |
| Other Information: | Additional information could include details about the support team responsible for processing batch updates, any processing times associated with batch update requests, and available support resources for students using the feature. | | |
| Assumptions: | Assumptions made in a student context might include assuming that students have access to the necessary technology to submit batch update requests, that they are aware of the process for performing batch updates, and that they have the necessary permissions to make the requested changes. | | |

**Business Rules**

None

#### 2.3.5 Synchronise project issue

###### **a. UI Specifications**

### 

### 

| Field Name: | Field Type | Description: |
| --- | --- | --- |
| Batch Update |  |  |
| Start Synchronization | Button | Show Synchronise project issue |

### 

###### **b. UC Specifications**

#### UC-5\_Synchronise project issues

| ID and Name: | **UC-5 Synchronise project issues** | | |
| --- | --- | --- | --- |
| Created By: | TruongNQ | Date Created: | 1/10/2023 |
| Primary Actor: | Student | Secondary Actors: | None |
| Description: | This function provides information about how students can use this feature to ensure that their project-related issues or tasks are up-to-date and synchronized across relevant platforms or tools. | | |
| Trigger: | To update project-related information, such as tasks, deadlines, or progress, to ensure consistency and accuracy across various project management tools or platforms. | | |
| Preconditions: | This feature may include having active projects or assignments that involve multiple tools or platforms for task management, having access to these tools, and being logged into their student accounts. | | |
| Postconditions: | Include successfully updating project-related information across the selected tools or platforms, ensuring that all project data is consistent and accurate. | | |
| Normal Flow: | * Log into the student portal. * Navigate to the "Projects" or "Task Management" section. * Select the "Synchronize Project Issues" option. * Choose the project or projects that need synchronization. * Select the external tools or platforms (e.g., project management software, calendars) that need to be updated. * Review and confirm the synchronization settings and data to be updated. * Initiate the synchronization process. * Receive a confirmation message indicating that the synchronization has been completed successfully. | | |
| Alternative Flows: | Allow students to configure synchronization schedules for automatic updates.  Students may have the option to manually resolve conflicts or discrepancies that arise during synchronization. | | |
| Exceptions: | * If there are technical issues preventing a student from initiating synchronization, an error message should be displayed with troubleshooting instructions. * In case synchronization with external tools or platforms fails, students should be provided with guidance on resolving the issue or contacting support. | | |
| Priority: | High | | |
| Business Rules: | include rules for data mapping and transformation during synchronization, ensuring data privacy and security, and handling conflicts or data discrepancies. | | |
| Other Information: | Include details about the types of external tools or platforms that can be synchronized, any limitations or restrictions on synchronization, and available support resources for students using the feature. | | |
| Assumptions: | Assumptions made in a student context might include assuming that students have access to the necessary project management tools, that they are familiar with the synchronization process, and that they have active projects or assignments requiring synchronization. | | |

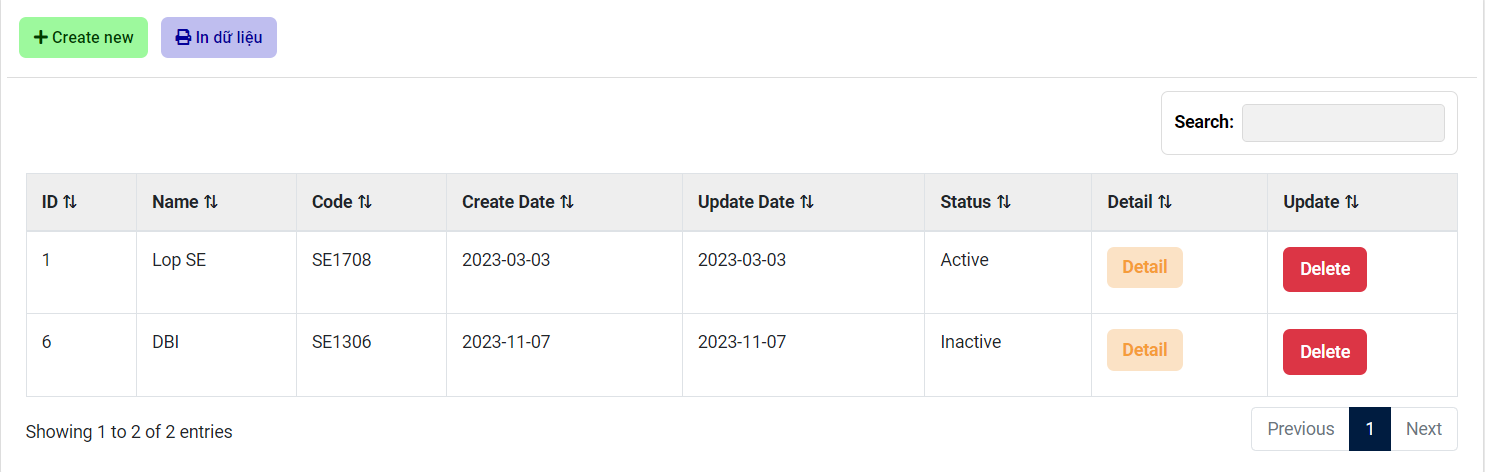
### 2.4 Manager Feature

#### 2.4.5. UC-5 View class list

##### a.UI Specifications

**Class List**

This screen allows the manager to view the class list.



| **Field Name** | **Field Type** | **Description** |
| --- | --- | --- |
| Class Code | Text box | This is for manager to input valid class code to search |
| Search | Button | User clicks to search the class code in class list |
| Delete pending | Button | User clicks to delete the pending class |
| Update | Button | User clicks to update class |
| Add new | Button | User clicks to add a new class |
| Class table | Table | Show information of class: class code, teacher, description, status |

##### b.UC Specifications

| ID and Name: | **UC-5 View class list** | | |
| --- | --- | --- | --- |
| Created By: | DungND | Date Created | 26/9/2023 |
| Primary Actor: | Manager | Secondary Actors | None |
| Description: | Manager want to view class list | | |
| Trigger: | Manager selects the "View class list" option from the manager dashboard. | | |
| Preconditions: | PRE-1. Manager is logged into IMS.  PRE-2. The manager account has been authorized | | |
| Postconditions: | POST-1. The class list displays successfully | | |
| Normal Flow: | Manager account login successfully.  Manager selects "view class list" from the dashboard.  The system displays a paginated list of all classes. | | |
| Alternative Flows: | If there are no issue classes defined in the system, a message is displayed indicating that there are no classes to view. | | |
| Exceptions: | Class data could not be loaded | | |
| Priority: | High | | |
| Frequency of Use: | Frequent | | |
| Business Rules | * Users can only view the issue classes and their descriptions; they cannot modify or delete classes from this view. * The display of issue classes should respect any privacy or security settings defined in the system. | | |
| Other Information: | * The class list interface should be user-friendly, facilitating easy navigation and understanding of the different issue classes. * Users may have the option to search, filter, or sort the issue classes based on specific criteria. | | |
| Assumptions: | * Issue classes have been defined and categorized within the system. * Users have the necessary permissions and access rights to view issue classes. | | |

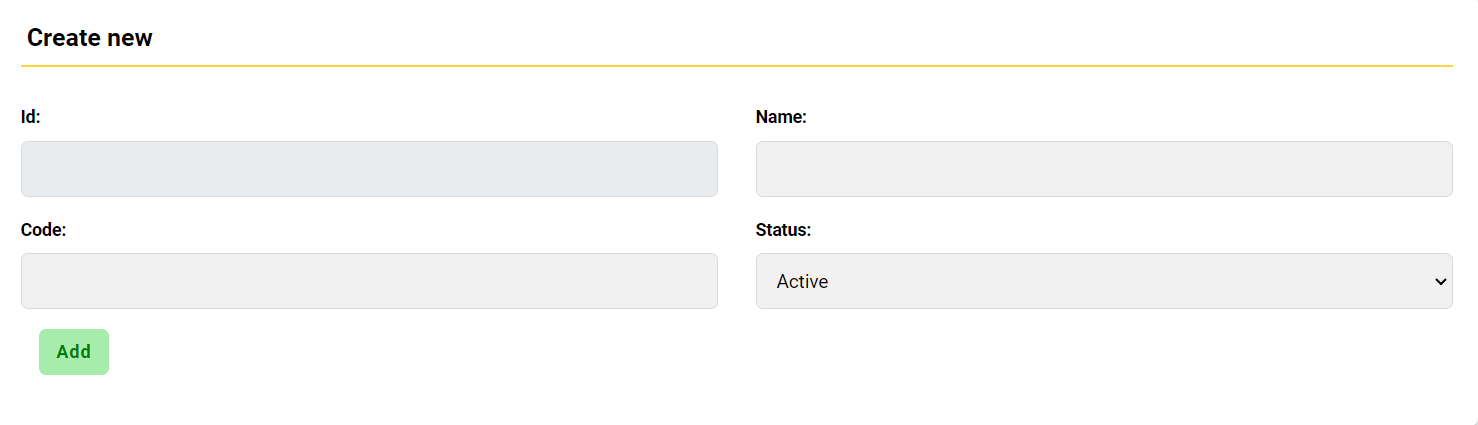
#### 

#### 2.4.6 UC-6 Add a new class

##### a.UI Specifications

**Class Information**

This screen allows the manager to update the class.



| **Field Name** | **Field Type** | **Description** |
| --- | --- | --- |
| Class Code | Text box | This is for manager to input valid class code |
| Teacher | Combo box | This is for manager to select teacher for class |
| Description | Text box | This is for manager to input description |
| Add | Button | User clicks to add a new class |

##### b.UC Specifications

#### 

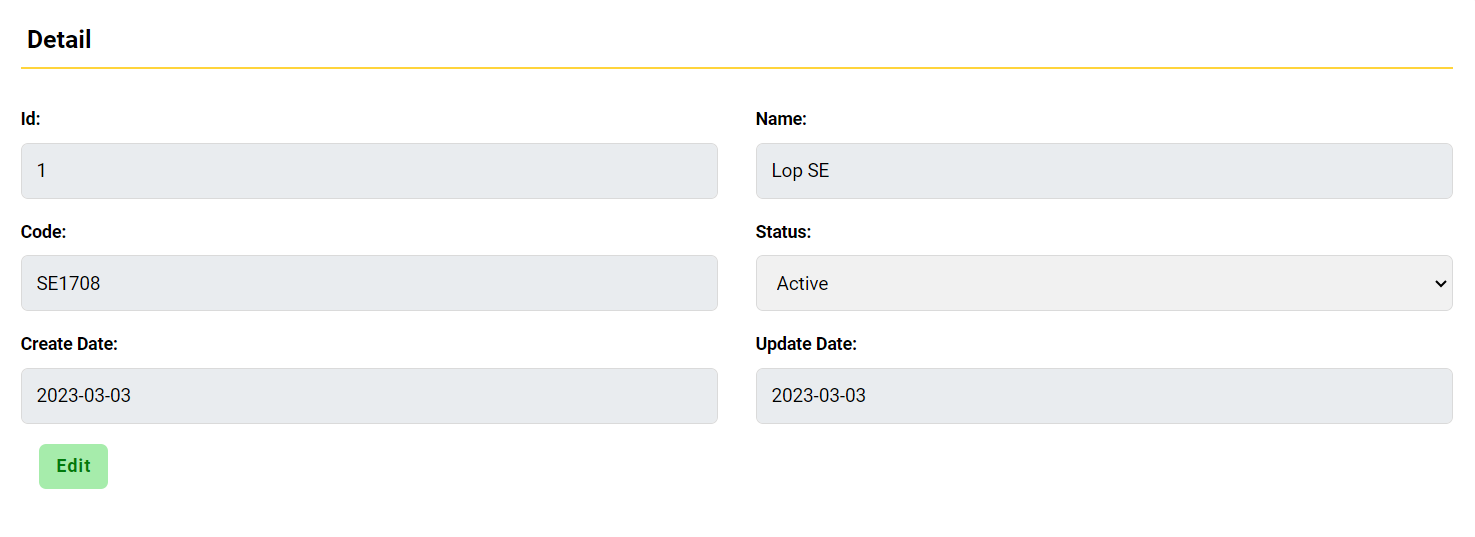
| ID and Name: | **UC-6 Add new class** | | |
| --- | --- | --- | --- |
| Created By: | DungND | Date Created | 26/9/2023 |
| Primary Actor: | Manager | Secondary Actors | None |
| Description: | Manager want to add new class into list | | |
| Trigger: | Manager selects “Add a new class” in manager dashboard | | |
| Preconditions: | PRE-1. Manager is logged into IMS.  PRE-2. The manager account has been authorized  PRE-3. Manager can view class list | | |
| Postconditions: | POST-1. A new class added successfully into list | | |
| Normal Flow: | Manager account login successfully.  Manager selects "view class list" from the dashboard.  Manager selects “Add new“ on the right top of the list .  Manager fills in the required fields (code, name, start date,teacher, members,status) and clicks the button “Add”.  The system validates the data.  List displays new class. | | |
| Alternative Flows: | If the user decides not to proceed with creating a new class after reaching the confirmation step, they can choose to cancel the process, and no new class is created. | | |
| Exceptions: | Manger import invalid data  Data of new class after clicking the button “Add” wasn’t displayed in list | | |
| Priority: | High | | |
| Frequency of Use: | Often | | |
| Business Rules | * Only authorized users with the appropriate permissions can create new issue classes. * Depending on system settings, certain fields may be marked as required, such as class name. | | |
| Other Information: | * The interface for adding a new class should be user-friendly, with clear fields and options for input. * Users may have the option to provide additional information or attributes for the new class, depending on system requirements. | | |
| Assumptions: | * The new class name is unique and does not conflict with existing classes. * Users have the necessary permissions and access rights to create new issue classes. | | |

#### 

#### 2.4.7 UC-7 Update class

##### a.UI Specifications

**Class Information**

This screen allows the manager to update the class.****

| **Field Name** | **Field Type** | **Description** |
| --- | --- | --- |
| Class Code | Text box | This is for manager to input valid class code |
| Teacher | Combo box | This is for manager to select teacher for class |
| Description | Text box | This is for manager to input description |
| Update | Button | User clicks to update new information of class |

##### b.UC Specifications

| ID and Name: | **UC-7 Update class details** | | |
| --- | --- | --- | --- |
| Created By: | DungND | Date Created | 26/9/2023 |
| Primary Actor: | Manager | Secondary Actors | None |
| Description: | Manager want to update class information | | |
| Trigger: | Manager selects the information of the class details in the class list that he/she wants to update. | | |
| Preconditions: | PRE-1. Manager is logged into IMS.  PRE-2. The manager account has been authorized | | |
| Postconditions: | POST-1. Notification of successful update.  POST-2. Display new information of the class in the class list | | |
| Normal Flow: | Manager account login successfully.  Manager selects "view class list" from the dashboard.  Manager selects one class by code or name.  The class shows information details.  Manager selects information that he/she want to update  Manager click button Save to save new data.  The system displays a notification of successful update.  The class displays new information. | | |
| Alternative Flows: | If the user decides not to proceed with updating the class after reaching the confirmation step, they can choose to cancel the process, and no changes are made to the class. | | |
| Exceptions: | Notification of successful update but data didn’t change  Notification of successful update but new data inserted wrong position | | |
| Priority: | Must have | | |
| Frequency of Use: | Often | | |
| Business Rules | * Only authorized users with the appropriate permissions can update issue classes. * Depending on system settings, certain fields may be restricted from being edited. | | |
| Other Information: | * The interface for updating a class should include a confirmation step to prevent accidental changes. * Users may have the option to provide a reason for the update, which can be logged for reference. | | |
| Assumptions: | * The class being updated exists within the system. * Users have the necessary permissions and access rights to update issue classes. | | |

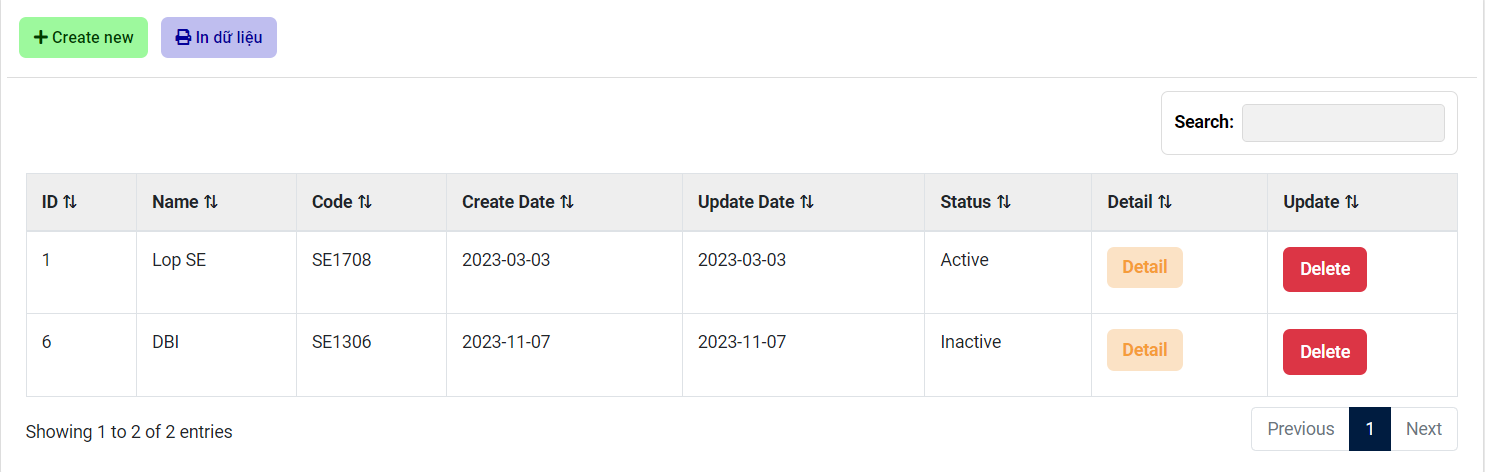
#### 

#### 2.4.8 UC-8 Delete pending class

##### a.UI Specifications

**Class List**

This screen allows the manager to delete the pending class.



| **Field Name** | **Field Type** | **Description** |
| --- | --- | --- |
| Class Code | Text box | This is for manager to input valid class code to search |
| Search | Button | User clicks to search the class code in class list |
| Delete pending | Button | User clicks to delete the pending class |
| Update | Button | User clicks to update class |
| Add new | Button | User clicks to add a new class |
| Class table | Table | Show information of class: class code, teacher, description, status |

##### b.UC Specifications

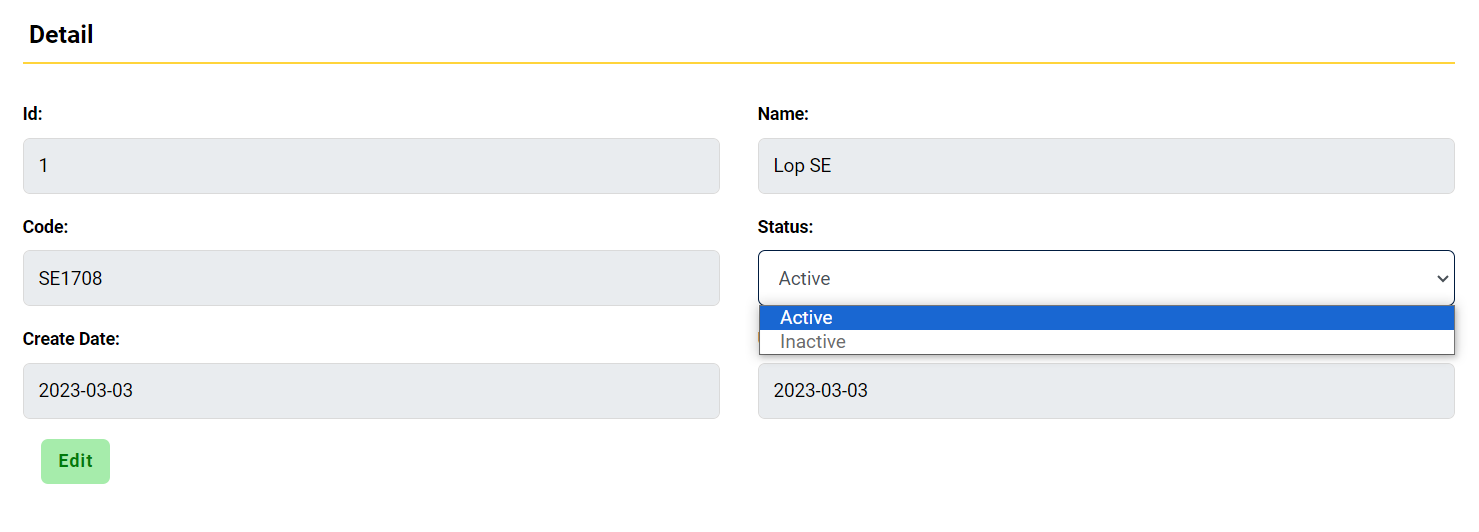
| UC ID and Name: | **UC-8 Delete pending class** | | |
| --- | --- | --- | --- |
| Created By: | DungND | Date Created: | 26/9/2023 |
| Primary Actor: | Manager | Secondary Actors: | None |
| Trigger: | Manager selects a class from the class list and click button "Delete pending" | | |
| Description: | This use case describes how an Manager deletes a pending class from the system. | | |
| Preconditions: | PRE-1. Manager is logged into IMS.  PRE-2. The manager account has been authorized  PRE-3. Manager can view class list  PRE-4. Classes are already defined in the system. | | |
| Postconditions: | The selected pending class is deleted from the system. | | |
| Normal Flow: | Manager selects a class from the list.  The system displays the class details.  Manager checks the deadline to handle it.  If it is out of date, the manager click button “Delete pending”  The system displays a confirmation dialog “Are you sure to delete this pending class?”.  If confirmed, the manager clicks the button “YES”.  The system deletes the pending class.  The system reloads a new class list. | | |
| Alternative Flows: | If the user decides not to proceed with deleting the pending class after reaching the confirmation step, they can choose to cancel the process, and no deletion occurs. | | |
| Exceptions: | If the selected class does not exist or has been deleted, an error message is displayed.  Deadline has not passed yet. | | |
| Priority: | High | | |
| Frequency of Use: | sometimes | | |
| Business Rules: | * Only authorized users with the appropriate permissions can delete pending classes. * The system may have specific criteria for when a class is considered "pending" or for deletion. | | |
| Other Information: | * The interface for deleting pending classes should include a confirmation step to prevent accidental deletions. * Users may have the option to provide a reason for the deletion, which can be logged for reference. | | |
| Assumptions: | * The pending class to be deleted exists in the system but has not been fully added. * Users have the necessary permissions and access rights to delete pending classes. | | |

#### 2.4.9 UC-9 Cancel started class

##### a.UI Specifications

**Class List**

This screen allows the manager to cancel started class.



| **Field Name** | **Field Type** | **Description** |
| --- | --- | --- |
| Class Code | Text box | This is for manager to input valid class code to search |
| Search | Button | User clicks to search the class code in class list |
| Delete pending | Button | User clicks to delete the pending class |
| Update | Button | User clicks to update class |
| Add new | Button | User clicks to add a new class |
| Class table | Table | Show information of class: class code, teacher, description, status |

##### b.UC Specifications

| UC ID and Name: | **UC-9 Cancel started class** | | |
| --- | --- | --- | --- |
| Created By: | DungND | Date Created: | 26/9/2023 |
| Primary Actor: | Manager | Secondary Actors: | None |
| Trigger: | Manager selects a class from the class list and clicks the radio button "canceled" in Status. | | |
| Description: | This use case describes how a Manager cancels started class from the system. | | |
| Preconditions: | PRE-1. Manager is logged into IMS.  PRE-2. The manager account has been authorized  PRE-3. Manager can view class list  PRE-4. Classes are already defined in the system. | | |
| Postconditions: | The selected class is canceled from the system. | | |
| Normal Flow: | Manager selects a class from the list.  The system displays the class details.  Manager clicks the radio button “canceled” in status.  The system changed the status of the class to “canceled”. | | |
| Alternative Flows: | If the user decides not to proceed with canceling the started class after reaching the confirmation step, they can choose to cancel the process, and no cancellation occurs. | | |
| Exceptions: | If the selected class does not exist or has been deleted, an error message is displayed. | | |
| Priority: | High | | |
| Frequency of Use: | sometimes | | |
| Business Rules: | * Only authorized users with the appropriate permissions can cancel started classes. * The system may have specific criteria for when a class is considered "started" or for cancellation. | | |
| Other Information: | * The interface for canceling started classes should include a confirmation step to prevent accidental cancellations. * Users may have the option to provide a reason for the cancellation, which can be logged for reference. | | |
| Assumptions: | * The started class to be canceled exists in the system and is in progress. * Users have the necessary permissions and access rights to cancel started classes. | | |

## 3. Non-Functional Requirements

### 3.1 External Interfaces

*[This section provides information to ensure that the system will communicate properly with users and with external hardware or software/system elements.]*

**Performance:** Ensure that the system can handle a certain number of concurrent external requests without degrading performance. Response times for external interactions should meet predefined criteria.

**Security:** Protect external interfaces from security threats and vulnerabilities. This includes data encryption, access control, and protection against common security risks.

**Reliability**: External interfaces should be reliable and available, with minimal downtime. Non-functional requirements may include availability targets and disaster recovery plans.

**Interoperability:** The system must be able to communicate and exchange data with external systems seamlessly. Non-functional requirements may specify data formats, protocols, and compatibility.

**Compliance:** Ensure that the system complies with relevant regulations and industry standards, such as GDPR, HIPAA, or ISO standards. Non-functional requirements should address regulatory compliance.

**Logging and Monitoring:** The system should log interactions with external interfaces for auditing and monitoring. Non-functional requirements may include log retention and reporting capabilities.

### 3.2 Quality Attributes

*[List all the required system characteristics (quality attributes) specification. Some of the possible attributes are provided with the guide/descriptions are mentioned here]*

#### 3.2.1 Usability

* User Interface Design: The user interface should be intuitive, user-friendly, and consistent, following established usability guidelines.
* Accessibility: Ensure the system is accessible to users with disabilities, meeting WCAG standards, and other relevant accessibility guidelines.
* Training and Documentation: Provide user documentation and training resources to help users effectively use the system.

#### 3.2.2 Performance

* Response Time: The system should respond to user requests within X seconds, ensuring a smooth user experience.
* Throughput: The system should handle a minimum of X concurrent users or issues per minute without performance degradation.
* Scalability: The system should be able to scale horizontally and/or vertically to handle increased loads during peak times.

#### 3.2.3 Security

* Data Encryption: All data transmitted over the network should be encrypted using SSL/TLS to protect against eavesdropping.
* Access Control: Implement role-based access control to restrict system access based on user roles and permissions.
* Authentication: Implement strong user authentication methods, including multi-factor authentication if required.
* Data Protection: Ensure sensitive user data is stored securely and that the system is protected against common security vulnerabilities, such as SQL injection and cross-site scripting (XSS).

#### 3.2.4 Maintainability

* Code Quality: Maintain a high code quality standard with well-documented code, adherence to coding standards, and comments where necessary.
* Version Control: Implement version control to track changes and manage codebase updates.

## 4. Requirement Appendix

### 4.1 Business Rules

*[Provide common business rules that you must follow. The information can be provided in the table format as the sample below]*

| **ID** | **Rule Definition** |
| --- | --- |
| BR-RA-001 | This format includes clear and concise language, metadata fields (e.g., requirement ID, creation date, author, and status), and appropriate headings or sections for categorizing and describing each requirement. |
| BR-RA-002 | Users must be able to access and review previous versions of requirements, and the system should maintain a change history to track modifications. |
| BR-RA-003 | This workflow should specify the responsible parties for reviewing and approving requirements, and the system should have mechanisms for notifying these parties about pending actions. |
| BR-RA-004 | This process should require the documentation of change requests, their review by relevant stakeholders, and approval before implementing changes. |
| BR-RA-011 | The system should provide advanced search and filter options to allow users to find specific requirements based on various criteria, such as keywords, categories, or dates. |
| BR-RA-012 | Access to these requirements should be restricted to authorized users, and appropriate data security measures should be implemented to prevent unauthorized access or data breaches. |
| BR-RA-024 | These backups should be scheduled at defined intervals and should include mechanisms for data recovery. |

### 4.2 Common Requirements

*[Fill all the common requirements here..]*

**Document Inclusion:**

* All project requirements must be documented within the Requirement Appendix.
* Each requirement should include a unique identifier, a clear description, and any relevant metadata, such as the date of creation and the author.

**Organization and Categorization:**

* Requirements within the Requirement Appendix should be organized and categorized based on the project's structure and priorities.
* Users should be able to easily locate and navigate through requirements, possibly through a hierarchical or tagging system.

**Requirement Format:**

* All requirements in the Requirement Appendix must adhere to a predefined format, ensuring consistency in documentation.
* This format may include standardized sections for specifying the requirement's purpose, acceptance criteria, and dependencies.

**Version Control:**

* The Requirement Appendix should maintain a version control system to track changes to requirements.
* Users should be able to view the history of each requirement and access previous versions.

**Search and Retrieval:**

* Users should be able to search for and retrieve requirements in the Requirement Appendix efficiently.
* The system should offer search functionality with options to filter and sort requirements based on various criteria.

**Data Security:**

* Sensitive or confidential requirements in the Requirement Appendix should be protected.
* The system should employ security measures, such as encryption and access control, to safeguard these requirements.

### 4.3 Application Messages List

| **#** | **Message code** | **Message Type** | **Context** | **Content** |
| --- | --- | --- | --- | --- |
| 1 | REQ001 | Information | Requirement Creation | *"A new requirement has been successfully created."* |
| 2 | REQ002 | Information | Requirement Update | *"The requirement has been updated and saved."* |
| 3 | REQ003 | Information | Requirement Deletion | "The requirement has been deleted." |
| 4 | REQ004 | Warning | Requirement Deletion | *"Caution: This action will permanently delete the selected requirement. Confirm the deletion."* |
| 5 | REQ005 | Error | Requirement Creation | *"Failed to create the requirement. Please check your input and try again."* |
| 6 | REQ006 | Error | Requirement Update | *"Failed to update the requirement. Please check your input and try again."* |

### 