TOPIC

Manage and evaluate group presentations

| Group | |
| --- | --- |
| Group members | Nguyễn Bá Nhật Linh  Nguyễn Hoàng Khang  Trần Huy Hoàng  Nguyễn Phạm Nam An  Trang Quốc Đạt |
| Supervisor | Lại Đức Hùng |
| Project code | Manage and evaluate group presentations |

# Project Introduction

## Overview

* 1. Project Information
     1. Project name: Manage and evaluate group presentations
     2. Project code:
     3. Group name:
     4. Software type: Web application
  2. Project Team
     1. Supervisor

| Full Name | Email | Title |
| --- | --- | --- |
| Lâm Hữu Khánh Phương |  | Lecturer |
| Lại Đức Hùng |  | Lecturer |

* + 1. Team member

| Full name | Email | Role |
| --- | --- | --- |
| Trần Huy Hoàng |  |  |
| Nguyễn Hoàng Khang |  |  |
| Nguyễn Bá Nhật Linh | linhnbnse161462@fpt.edu.vn |  |
| Nguyễn Phạm Nam An | annpnse161461@fpt.edu.vn |  |
| Trang Quốc Đạt | dattqse161488@fpt.edu.vn |  |

## Product Background

The Project Registration System at FPT University is a software application designed to streamline the process of project registration for students. FPT University is a prominent educational institution known for its focus on technology and innovation. The system was developed to enhance the efficiency and effectiveness of project management within the university's academic programs.

The primary goal of the Project Registration System is to provide a centralized platform where students can submit project proposals and register for projects of their interest. It serves as a bridge between students, faculty members, and project supervisors, facilitating better communication and collaboration throughout the project lifecycle.

## Existing Systems

## Business Opportunity

In order to gain a deeper understanding of UniHome's customer needs, the research team noticed a number of issues that were of great concern to those selected to conduct the survey.

Through data and from the survey along with mining and research, we found the concerns of users, and teachers struggling including group management (allocating groups in class), member management, topic management for each class/subject), presentation management (notifying the presentation time for students, manage presentation time), application (manage the presentation application of each student) group, manage group member reports)

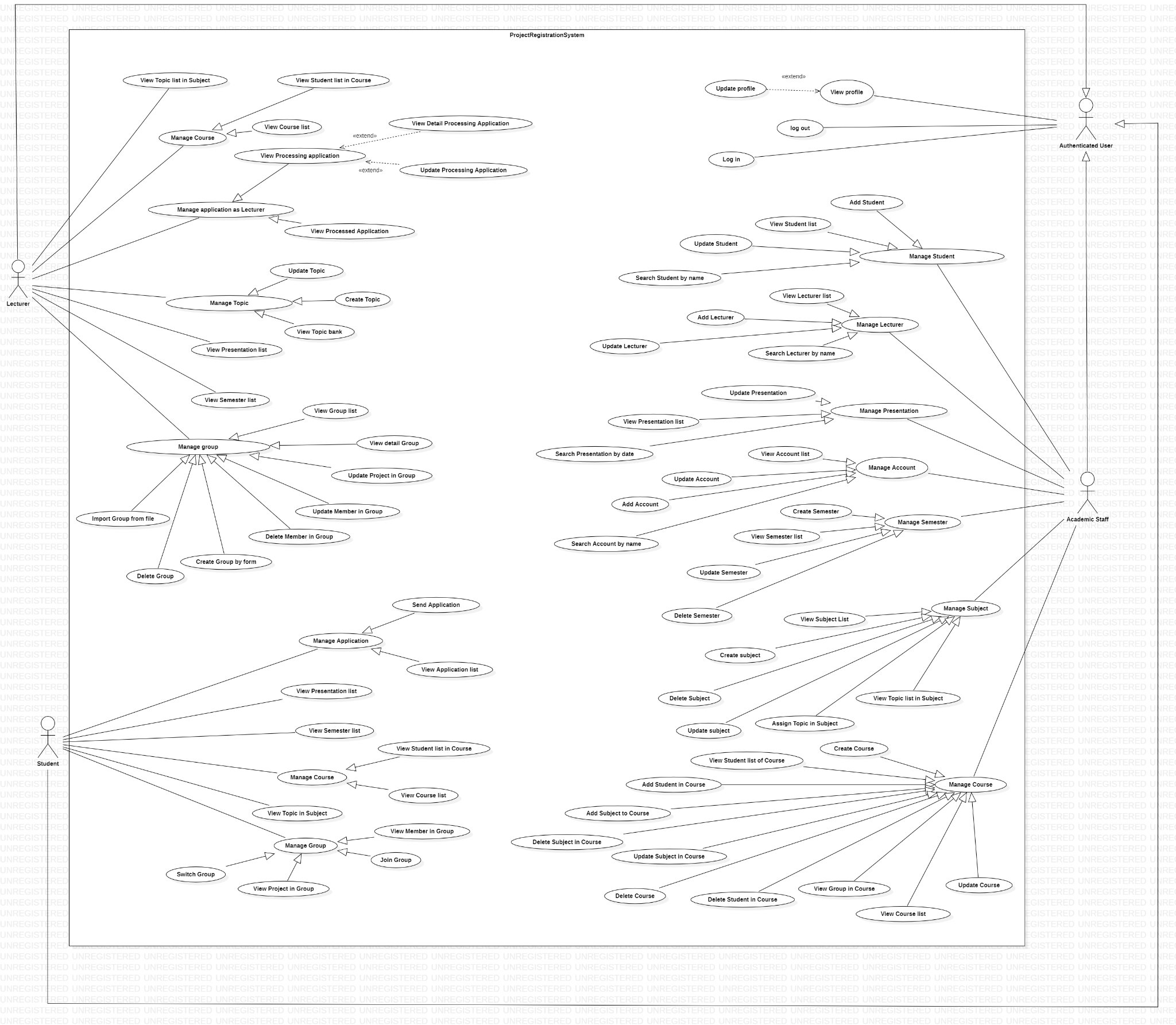
For students, after studying and considering student problems, they are group management (create groups, join groups, contact group members, report members), topic management ( view topics in each group, select topics), presentations (see presentation time, submit an application for presentation)

## Software Product Vision

The software product vision for the Project Registration System at FPT University is to create an efficient, user-friendly platform that streamlines project registration and management, fosters collaboration, ensures fair project allocation, and provides valuable insights for improved decision-making.

## Project Scope and Exclusions

* 1. Major Features



* + 1. Authenticated users: who use web applications, can choose either web application to use the following functions:
    2. Lecturer: Users of the student management page and projects on this website will use the following functions:
    3. Student: Users use this website to view and know subject information
    4. Academic Staff: Admins (login required) are capable of using the following functions:

1. Limitations and Exclusions

# Software Requirements Specification

## Overall Description

### User Requirements Overview

#### Authenticated users: who use web applications, can choose either web application to use the following functions:

* Log in
* Log out
* View Profile
* Update Profile
* Forget Password /

#### Lecturer: Users of the student management page and projects on this website will use the following functions:

* View Topic list in Subject/
* View Student list in Course
* View Course list
* View Processing Application
* View Detail Processing Application
* Update Processing Application
* View Processed Application
* Update Topic
* Create Topic
* View Topic Bank
* View Presentation list
* View Semester list/
* View Group list
* View Detail Group
* Update Project in Group
* Update Member in Group
* Delete Member in Group
* Delete Group
* Create Group by form
* Import Group by file

#### Student: Users use this website to view and know subject information

* Send Application
* View Application list
* View Presentation list
* View Semester list
* View Student list in Course
* View Course list
* View Topic in Subject
* View Member in Group
* Join Group
* View Project in Group
* Switch Group

Academic Staff: Admins (login required) are capable of using the following functions:

* Add Student
* View Student list
* Update Student
* Search Student by name
* View Lecturer list
* Add Lecturer
* Update Lecturer
* Search Lecturer by Name
* Update Presentation
* View Presentation list
* Search Presentation by Date
* View Account list
* Update Account
* Add Account
* Search Account by Name
* Create Semester
* View Semester list
* Update Semester
* Delete Semester
* View Subject list
* Create Subject
* Delete Subject
* Update Subject
* Assign Topic in Subject
* View Topic list int Subject
* Create Course
* View Student list of Course
* Add Student in Course
* Add Subject to Course
* Delete Subject in Course
* Update Subject in Course
* Delete Course
* Delete Student in Course
* View Group in Course
* View Course list
* Update Course

## User Requirements

### System Actors

| ID | Actor | Description |
| --- | --- | --- |
| 1 | Authenticated user | Users who have logged in the Capstone Project Required-Subject Checking and Student Group Registration web application |
| 2 | Lecturer | Users who have logged in Capstone Project Required-Subject Checking and Student Group Registration Lecturer web application |
| 3 | Student | Users who have logged in Capstone Project Required-Subject Checking and Student Group Registration Student web application |
| 4 | Admin | Users who have logged in Capstone Project Required-Subject Checking and Student Group Registration Admin web application |
| 5 | Viewer | Viewer is the person who will receive information about the presentation as well as evaluate it and ask questions. |

### Use Case Diagram

#### Authenticated User

| ID | Use Case | Primary Actors | Secondary Actors |
| --- | --- | --- | --- |
| UCU-01 | log in | Authenticated User |  |
| UCU-02 | log out | Authenticated User |  |
| UCU-03 | View Profile | Authenticated User |  |
| UCU-04 | Update Profile | Authenticated User |  |
| UCU-05 | Forget Password | Authenticated User |  |

#### Lecturer

| ID | Use Case | Primary Actors | Secondary Actors |
| --- | --- | --- | --- |
| UCL-01 | View Topic list in Subject | Lecturer |  |
| UCL-02 | View Student list in Course | Lecturer |  |
| UCL-03 | View Course list | Lecturer |  |
| UCL-04 | View Processing Application | Lecturer |  |
| UCL-05 | View Detail Processing Application | Lecturer |  |
| UCL-06 | Update Processing Application | Lecturer |  |
| UCL-07 | View Processed Application | Lecturer |  |
| UCL-08 | View Topic Bank | Lecturer |  |
| UCL-09 | Update Topic | Lecturer |  |
| UCL-10 | Create Topic | Lecturer |  |
| UCL-11 | View Presentation list | Lecturer |  |
| UCL-12 | View Semester list | Lecturer |  |
| UCL-13 | View Group list in Course | Lecturer |  |
| UCL-14 | View Detail Group | Lecturer |  |
| UCL-15 | Update Project in Group | Lecturer |  |
| UCL-16 | Update Member in Group | Lecturer |  |
| UCL-17 | Delete Member in Group | Lecturer |  |
| UCL-18 | Delete Group | Lecturer |  |
| UCL-19 | Create Group by form | Lecturer |  |
| UCL-20 | Import Group by file | Lecturer |  |

#### Student

| ID | Use Case | Primary Actors | Secondary Actors |
| --- | --- | --- | --- |
| UCS-01 | Send Application | Student |  |
| UCS-02 | View Application list | Student |  |
| UCS-03 | View Presentation list | Student |  |
| UCS-04 | View Semester list | Student |  |
| UCS-05 | View Student list in Course/ | Student |  |
| UCS-06 | View Course list/ | Student |  |
| UCS-07 | View Topic in Subject | Student |  |
| UCS-08 | View Member in Group | Student |  |
| UCS-09 | Join Group | Student |  |
| UCS-10 | View Project in Group | Student |  |
| UCS-11 | Switch Group | Student |  |

#### Academic Staff

| ID | Use Case | Primary Actors | Secondary Actors |
| --- | --- | --- | --- |
| UCA-01 | Add Student | Academic Staff |  |
| UCA-02 | View Student list | Academic Staff |  |
| UCA-03 | Update Student | Academic Staff |  |
| UCA-04 | Search Student by name | Academic Staff |  |
| UCA-05 | View Lecturer list | Academic Staff |  |
| UCA-06 | Add Lecturer | Academic Staff |  |
| UCA-07 | Update Lecturer | Academic Staff |  |
| UCA-08 | Search Lecturer by Name | Academic Staff |  |
| UCA-09 | Update Presentation | Academic Staff |  |
| UCA-10 | View Presentation list | Academic Staff |  |
| UCA-11 | Search Presentation by Date | Academic Staff |  |
| UCA-12 | View Account list | Academic Staff |  |
| UCA-13 | Update Account | Academic Staff |  |
| UCA-14 | Add Account | Academic Staff |  |
| UCA-15 | Search Account by Name | Academic Staff |  |
| UCA-16 | Create Semester | Academic Staff |  |
| UCA-17 | View Semester list | Academic Staff |  |
| UCA-18 | Update Semester | Academic Staff |  |
| UCA-19 | View Subject list | Academic Staff |  |
| UCA-20 | Create Subject | Academic Staff |  |
| UCA-21 | Delete Subject | Academic Staff |  |
| UCA-22 | Update Subject | Academic Staff |  |
| UCA-23 | Assign Topic in Subject | Academic Staff |  |
| UCA-24 | View Topic list int Subject | Academic Staff |  |
| UCA-25 | Create Course | Academic Staff |  |
| UCA-26 | View Student list of Course | Academic Staff |  |
| UCA-27 | Add Student in Course | Academic Staff |  |
| UCA-28 | Add Subject to Course | Academic Staff |  |
| UCA-29 | Delete Subject in Course | Academic Staff |  |
| UCA-30 | Update Subject in Course | Academic Staff |  |
| UCA-31 | Delete Course | Academic Staff |  |
| UCA-32 | Delete Student in Course | Academic Staff |  |
| UCA-33 | View Group in Course | Academic Staff |  |
| UCA-34 | View Course list | Academic Staff |  |
| UCA-35 | Update Course | Academic Staff |  |

* 1. Use Cases Diagram

Authenticated User

Lecturer

Student

Academic Staff

* 1. Usecase Detail

### Authenticated User

| ID and Name: | UCU-01: Login | | |
| --- | --- | --- | --- |
| Create by: | KhangNH | Date Created: | 21/06/2023 |
| Actor: | Authenticated User | | |
| Priority: | high | Frequency of Use: | Always |
| Trigger: | Actor wants to login Project Registration System | | |
| Description: | This feature allows users to login in Project Registration system | | |
| Pre-conditions: | Actor has authenticated into the Project Registration system. | | |
| Post-conditions: | Success: Actor successfully logged in and accessed the Home page. | | |
| Main Flow: | Step | Actor action | System response |
| 1 | Actor accesses Project Registration System website. | The system will display:  Project Registration System  Login   * Email (Email) (Exception 1) (Exception 2) * Password (String) (Exception 3) * "Login" button * "Forgot password" button |
| 2 | Actor fills in the email and password then clicks on the "Login" button. | Actor successfully logged in and accessed the Home page. |
| Alternative Flows: | N/A | | |
| Exception: | No | Cause | System Response |
| 1 | Enter an email that does not exist in the system. | The system displays the message "Incorrect email or password". |
| 2 | Enter the wrong email form. | The system displays the message "Enter the wrong email form.". |
| 3 | Enter the wrong password. | The system displays the message "Incorrect email or password". |
| 4 | Leave it blank and press the "Login" button. | The browser displays the message "Please fill out this field ". |
| Business Rules: | * Email must exist in the system. * Email must be in the correct form. (ex: [phuonglhk@fe.edu.vn](mailto:phuonglhk@fe.edu.vn) or hoangthse161468@fpt.edu.vn) | | |
| Assumptions: | Internet connection is available. | | |

| ID and Name: | UCU-02: Log out | | |
| --- | --- | --- | --- |
| Create by: | KhangNH | Date Created: | 21/06/2023 |
| Actor: | Authenticated User | | |
| Priority: | high | Frequency of Use: | Always |
| Trigger: | Actor wants to log out Project Registration System | | |
| Description: | This feature allows users to log out Project Registration system | | |
| Pre-conditions: | Actor has authenticated into the Project Registration system. | | |
| Post-conditions: | Success: Actor successfully logged out and accessed the Login page. | | |
| Main Flow: | Step | Actor action | System response |
| 1 | Actor clicks on the Logout icon | System redirects to Login screen |
| Alternative Flows: | N/A | | |
| Exception | N/A | | |
| Business Rules: | N/A | | |
| Assumptions: | Internet connection is available. | | |

| ID and Name: | UCU-03: View profile | | |
| --- | --- | --- | --- |
| Create by: | KhangNH | Date Created: | 21/06/2023 |
| Actor: | Authenticated User | | |
| Priority: | Medium | Frequency of Use: | sometimes |
| Trigger: | Actor wants to view his/her profile information. | | |
| Description: | This feature allows users to get profile information | | |
| Pre-conditions: | Actor has authenticated into the Project Registration system. | | |
| Post-conditions: | Success: Actor will be redirected to Profile screen of the website. | | |
| Main Flow: | Step | Actor action | System response |
| 1 | Actor clicks on the Profile icon. | The system will display the actor's profile:   * Image (image) * Student Code (String) * Name(String) * Birthday (Date) * Gender (String) * Phone(String) * Address(String) * Email (String) * "Update" button |
| Alternative Flows: | N/A | | |
| Exception | N/A | | |
| Business Rules: | N/A | | |
| Assumptions: | Internet connection is available. | | |

| ID and Name: | UCU-04: Update profile | | |
| --- | --- | --- | --- |
| Create by: | KhangNH | Date Created: | 21/06/2023 |
| Actor: | Authenticated User | | |
| Priority: | Medium | Frequency of Use: | sometimes |
| Trigger: | Actor wants to update actor profile information. | | |
| Description: | This feature allows users to update profile information. | | |
| Pre-conditions: | Actor has authenticated into the Project Registration system. | | |
| Post-conditions: | Success: The system displays the message "Update successfully" then reloads the Profile page. | | |
| Main Flow: | Step | Actor action | System response |
| 1 | Actor clicks on the Profile icon. | The system will display the actor's profile:   * Image (image) * Student Code (String) * Name(String) * Birthday (Date) * Gender (String) * Phone(String) * Address(String) * Email (String) * "Update" button |
| 2 | Actor fills in the information he wants to update and presses the "Update" button. | The system displays the message "Update successfully" then reloads the Profile page. |
| Alternative Flows: | N/A | | |
| Exception | No | Cause | System Response |
|  | 1 | Enter phone numbers < 10 and > 11 . | The system displays "Phone from 10 to 11 numbers" |
| Business Rules: | * Image, Student Code, Email can not update. * Phone from 10 to 11 numbers | | |
| Assumptions: | Internet connection is available. | | |

### 

| ID and Name: | UCU-05: Forget password | | | |
| --- | --- | --- | --- | --- |
| Create by: | KhangNH | Date Created: | | 26/06/2023 |
| Actor: | Authenticated User | | | |
| Priority: | Medium | Frequency of Use: | | sometimes |
| Trigger: | User initiates the "Forget Password" process by clicking on the "Forgot Password" l button on the login page. | | | |
| Description: | This use case allows users to reset their forgotten password by verifying their email, setting a new password, and gaining access to their account. | | | |
| Pre-conditions: | The user has an existing account in the system.  The user is unable to log in due to a forgotten password. | | | |
| Post-conditions: | User has successfully reset their password and can now log in using the updated password. | | | |
| Main Flow: | Step | Actor action | | System response |
| 1 | Actor taps on Forget password button | | The system shows the forget password screen include :  Area to fill “Enter your email”  Submit button |
| 2 | Actor fill in email in text | |  |
|  | 3 | Actor taps on “Submit” button | | The system prints out “Email verification complete!!” |
|  | 4 |  | | the system prints out a place to fill in the information "Reset Password" |
|  | 5 | Actors fill in new passwords  (Exception 1) | |  |
|  | 6 | Actor taps on “Reset” | | The system will transmit back to the login page |
| Alternative Flows: | N/A | | | |
| Exception | No | | Cause | System Response |
|  | 1 | | Actor re-enters old password | The system will display the message “Your password has at least one character and one number!!” |
| Business Rules: | The new password must have one character and a number | | | |
| Assumptions: | Internet connection is available. | | | |

### 

### 2.4.2 Lecturer

| ID and Name: | UCL-01: View Topic list in Subject | | |
| --- | --- | --- | --- |
| Create by: | AnNPN | Date Created: | 28/07/2023 |
| Actor: | Lecturer | | |
| Priority: | high | Frequency of Use: | sometimes |
| Trigger: | The lecturer wants to view the list of subject topics. | | |
| Description: | This feature allows the Lecturer to view a list of Topics of a particular subject. | | |
| Pre-conditions: | * The lecturer is logged into the system. * Subject must have a Topic. * The lecturer must be able to access the Course of the Subject. | | |
| Post-conditions: | Lecturer successfully displays the Topic list of a subject. | | |
| Main Flow: | Step | Actor action | System response |
| 1 | Lecturer accesses a specific course. | The system will display:   * “List Topic” button * “List Group” button * List of students in class. |
| 2 | The lecturer clicks on the “List Topic” button. | The system displays the Topic list table:   * No (int) * Topic Code (String) * Topic Name (String) * Short Description (String) * Full Description (String) * Lecturer Name (String) * Start Date (date) * Modify Date (date) |
| Alternative Flows: | N/A | | |
| Exception | N/A | | |
| Business Rules: | N/A | | |
| Assumptions: | Internet connection is available. | | |

### 

| ID and Name: | UCL-02: View Student list in Course | | |
| --- | --- | --- | --- |
| Create by: | AnNPN | Date Created: | 28/07/2023 |
| Actor: | Lecturer | | |
| Priority: | high | Frequency of Use: | sometimes |
| Trigger: | The lecturer wants to see the list of students in the course. | | |
| Description: | This feature allows the lecturer to view a list of students of a particular course. | | |
| Pre-conditions: | * The lecturer is logged into the system. * Course must have students. * Lecturer must be able to access the Course of the Subject. | | |
| Post-conditions: | Lecturer successfully displays the Student list of a course. | | |
| Main Flow: | Step | Actor action | System response |
| 1 | Lecturer accesses a specific course. | The system will display:   * “List Topic” button * “List Group” button * List of students in class:   + No (int)   + Student Code (String)   + Name (String)   + Birthday (date)   + Phone Number (String)   + Gender (String)   + Address (String)   + City (String)   + Email (String) |
| Alternative Flows: | N/A | | |
| Exception | N/A | | |
| Business Rules: | N/A | | |
| Assumptions: | Internet connection is available. | | |

### 

| ID and Name: | UCL-03: View Course list | | |
| --- | --- | --- | --- |
| Create by: | AnNPN | Date Created: | 28/07/2023 |
| Actor: | Lecturer | | |
| Priority: | high | Frequency of Use: | sometimes |
| Trigger: | The lecturer wants to view the class list. | | |
| Description: | This feature allows the Lecturer to view the class list of a particular subject. | | |
| Pre-conditions: | * The Lecturer is logged into the system. * Subject must have a Course. * The lecturer must access Semester. | | |
| Post-conditions: | Lecturer shows a successful course list. | | |
| Main Flow: | Step | Actor action | System response |
| 1 | Lecturer accesses a specific semester. | The system will display:   * Semester Name (String) * Start Date (date) * End Date (date) * Status (String) * List of subject code in the semester. * List of courses in the subject:   + Course Code (String)   + Subject Code (String)   + Subject Code (String)   + Lecturer Name (String)   + Start Date (date)   + End Date (date)   + “View” button |
| Alternative Flows: | N/A | | |
| Exception | N/A | | |
| Business Rules: | N/A | | |
| Assumptions: | Internet connection is available. | | |

### 

| ID and Name: | UCL-04: View Processing Application | | |
| --- | --- | --- | --- |
| Create by: | AnNPN | Date Created: | 28/07/2023 |
| Actor: | Lecturer | | |
| Priority: | high | Frequency of Use: | sometimes |
| Trigger: | The teacher wants to view the list of unapproved applications. | | |
| Description: | This feature allows the lecturer to view the list of unapproved Applications. | | |
| Pre-conditions: | * The Lecturer is logged into the system. * Lecturer must receive an application from the student. | | |
| Post-conditions: | Lecturer successfully displays the list of Unapproved Applications. | | |
| Main Flow: | Step | Actor action | System response |
| 1 | Lecturer clicks the “Application” button. | The system will display:   * “Processing Application” button * “Processed Application” button |
| 2 | Lecturer clicks the “Processing Application” button. | The system displays the Processing Application list.  Each Processing Application includes:   * No (int) * Application ID (int) * Type (String) * Subject Code (String) * Course Code (String) * Group Name (String) * Student Code (String) * Status: Processing (String) * Detail button |
| Alternative Flows: | N/A | | |
| Exception | N/A | | |
| Business Rules: | N/A | | |
| Assumptions: | Internet connection is available. | | |

| ID and Name: | UCL-05: View Detail Processing Application | | |
| --- | --- | --- | --- |
| Create by: | AnNPN | Date Created: | 28/07/2023 |
| Actor: | Lecturer | | |
| Priority: | high | Frequency of Use: | sometimes |
| Trigger: | Lecturer wants to view a detailed list of unapproved applications. | | |
| Description: | This feature allows the Lecturer to view the list of unapproved Applications with full information. | | |
| Pre-conditions: | * The Lecturer is logged into the system. * Lecturer must receive Application from Student. | | |
| Post-conditions: | Lecturer successfully displays the list of Unapproved Applications with full information. | | |
| Main Flow: | Step | Actor action | System response |
| 1 | Lecturer clicks the “Application” button. | The system will display:   * “Processing Application” button * “Processed Application” button |
| 2 | Lecturer clicks the “Processing Application” button. | The system displays the Processing Application list.  Each Processing Application includes:   * No (int) * Application ID (int) * Type (String) * Subject Code (String) * Course Code (String) * Group Name (String) * Student Code (String) * Status: Processing (String) * Detail button |
| 3 | Lecturer clicks the “Detail" button. | The system will display:   * No (int) * Application ID (int) * Type (String) * Subject Code (String) * Course Code (String) * Group Name (String) * Student Code (String) * Reason (String) * Lecturer’s Note (String) * Create Date (date) * Status : Processing or Processed (String) * “Update” button |
| Alternative Flows: | N/A | | |
| Exception | N/A | | |
| Business Rules: | N/A | | |
| Assumptions: | Internet connection is available. | | |

| ID and Name: | UCL-06: Update Processing Application | | |
| --- | --- | --- | --- |
| Create by: | AnNPN | Date Created: | 28/07/2023 |
| Actor: | Lecturer | | |
| Priority: | high | Frequency of Use: | sometimes |
| Trigger: | Lecturer wants to update an unapproved application. | | |
| Description: | This feature allows teachers to update Status and Note for Processing Application. | | |
| Pre-conditions: | * The Lecturer is logged into the system. * Lecturer must receive Application from Student. * Lecturer must view details of an unapproved application. | | |
| Post-conditions: | Lecturer successfully updates the Processing Application and converts it to a Processed Application. | | |
| Main Flow: | Step | Actor action | System response |
| 1 | Lecturer clicks the “Detail” button in the Processing Application. | * Display system: * No (int) * Application ID (int) * Type (String) * Subject Code (String) * Course Code (String) * Group Name (String) * Student Code (String) * Reason (String) * Lecturer’s Note (String) * Create Date (date) * Status : Processing or Processed (String) (Exception 1) * Update button |
| 2 | Lecturer fills in Lecturer's Note and selects Status as Approve or Refuse and then clicks on the “Update” button. | The system displays the message "Update Successfully". |
| Alternative Flows: | N/A | | |
| Exception | No | Cause | System Response |
| 1 | Lecturer does not select Status. | The system displays the message " Please choose Approve or Refuse status!!!". |
| Business Rules: | Status must select Approve or Refuse. | | |
| Assumptions: | Internet connection is available. | | |

| ID and Name: | UCL-07: View Processed Application | | |
| --- | --- | --- | --- |
| Create by: | AnNPN | Date Created: | 28/07/2023 |
| Actor: | Lecturer | | |
| Priority: | high | Frequency of Use: | sometimes |
| Trigger: | Lecturer wants to view the list of approved applications. | | |
| Description: | This feature allows the Lecturer to view the list of approved Applications. | | |
| Pre-conditions: | * The Lecturer is logged into the system. * Lecturer must receive Application from Student. | | |
| Post-conditions: | Lecturer successfully displays the list of approved Applications. | | |
| Main Flow: | Step | Actor action | System response |
| 1 | Lecturer clicks the “Application" button. | The system will display:   * “Processing Application” button * “Processed Application” button |
| 2 | Lecturer clicks the “Processed Application" button. | The system displays the Processing Application list.  Each Processing Application includes:   * No (int) * Application ID (int) * Type (String) * Subject Code (String) * Course Code (String) * Group Name (String) * Student Code (String) * Reason (String) * Lecturer’s Note (String) * Create Date (date) * Process Date (date) * Status: Processing (String) |
| Alternative Flows: | N/A | | |
| Exception | N/A | | |
| Business Rules: | N/A | | |
| Assumptions: | Internet connection is available. | | |

| ID and Name: | UCL-08: View Topic Bank | | |
| --- | --- | --- | --- |
| Create by: | AnNPN | Date Created: | 28/07/2023 |
| Actor: | Lecturer | | |
| Priority: | high | Frequency of Use: | sometimes |
| Trigger: | Lecturer wants to view the list of topics in the topic bank. | | |
| Description: | This feature allows the lecturer to view the list of Topics in the Topic Bank. | | |
| Pre-conditions: | * The Lecturer is logged into the system. * Topic Bank must have Topic. | | |
| Post-conditions: | Lecturer successfully displayed the topic list in Topic Bank. | | |
| Main Flow: | Step | Actor action | System response |
| 1 | Lecturer clicks the “Source Topic" button. | The system displays a list of topics.  Each topic includes:   * No (int) * Topic Code (String) * Topic Name (String) * Short Description (String) * Full Description (String) * Lecturer Name (String) * Update button   Create button |
| Alternative Flows: | N/A | | |
| Exception | N/A | | |
| Business Rules: | N/A | | |
| Assumptions: | Internet connection is available. | | |

| ID and Name: | UCL-09: Update Topic | | |
| --- | --- | --- | --- |
| Create by: | AnNPN | Date Created: | 28/07/2023 |
| Actor: | Lecturer | | |
| Priority: | high | Frequency of Use: | sometimes |
| Trigger: | Lecturer wants to update a topic in the topic bank. | | |
| Description: | This feature allows the lecturer to update the topic in the Topic Bank. | | |
| Pre-conditions: | * The Lecturer is logged into the system. * Topic Bank must have Topic. | | |
| Post-conditions: | Lecturer successfully updated the topic in Topic Bank. | | |
| Main Flow: | Step | Actor action | System response |
| 1 | Lecturer clicks the “Source Topic" button. | The system displays a list of topics.  Each topic includes:   * No (int) * Topic Code (String) * Topic Name (String) * Short Description (String) * Full Description (String) * Lecturer Name (String) * Update button   Create button |
| 2 | Lecturer fills in the information that needs to be updated in Topic Name, Short Description, Full Description and clicks on the “Update" button. | The system displays the message "Update Successfully!!". |
| Alternative Flows: | N/A | | |
| Exception | No | Cause | System response |
| 1 | Lecturer updates other Lecturer's topics. | The system displays the message “Do not update Topics of other Lecturer!!!”. |
| Business Rules: | Topic ID must be a standard form. (ex: HRM-501)  Topic ID is not duplicate. | | |
| Assumptions: | Internet connection is available. | | |

| ID and Name: | UCL-10: Create Topic | | |
| --- | --- | --- | --- |
| Create by: | AnNPN | Date Created: | 28/07/2023 |
| Actor: | Lecturer | | |
| Priority: | high | Frequency of Use: | sometimes |
| Trigger: | Lecturer wants to create a topic in the topic bank. | | |
| Description: | This feature allows the lecturer to create the topic in the Topic Bank. | | |
| Pre-conditions: | * The Lecturer is logged into the system. * The lecturer is logged in to the system. | | |
| Post-conditions: | Lecturer successfully created the topic in Topic Bank. | | |
| Main Flow: | Step | Actor action | System response |
| 1 | Lecturer clicks the “Source Topic" button. | The system displays a list of topics.  Each topic includes:   * No (int) * Topic Code (String) * Topic Name (String) * Short Description (String) * Full Description (String) * Lecturer Name (String) * Update button   Create button |
| 2 | Lecturer clicks the Create button. | The system will display:   * Topic Code (String) (Exception 1) (Exception 2) * Topic Name (String) * Short Description (String) * Full Description (String) * “Create” button * Reset button |
| 3 | Lecturer fills in Topic Code, Topic Name, Short Description, Full Description and clicks the “Create” button. | The system displays the message "Create Successfully!!". |
| Alternative Flows: | N/A | | |
| Exception | No | Cause | System response |
| 1 | Lecturer incorrectly filled in the Topic ID form. | The system displays the message “Topic Code is not Match !!!”. |
| 2 | Lecturer fills in the duplicate Topic ID. | The system displays the message “Topic Code is duplicated!!!”. |
| Business Rules: | N/A | | |
| Assumptions: | Internet connection is available. | | |

| ID and Name: | UCL-11: View List Presentation | | |
| --- | --- | --- | --- |
| Create by: | AnNPN | Date Created: | 18/07/2023 |
| Actor: | Lecturer | | |
| Priority: | high | Frequency of Use: | Often |
| Trigger: | The lecturer requests to view list presentation | | |
| Description: | This use case describes the process by which a lecturer views a presentation list. | | |
| Pre-conditions: | * The Lecturer is logged into the system. * The lecturer is logged into the system. | | |
| Post-conditions: | The lecturer can view the list of presentations. | | |
| Main Flow: | Step | Actor action | System response |
| 1 | The lecturer accesses the "Presentation" section. | The system displays a list of presentation announcements.  Each presentation notice includes:   * Presentation Code (String) * Subject Code (String) * Course Code (String) * Group ID (String) * Group Name (String) * Room (int) * Present Date (date) * Time (String) |
| Exception | N/A | | |
| Alternative Flows: | N/A | | |
| Business Rules: | N/A | | |
| Assumptions: | Internet connection is available. | | |

| ID and Name: | UCL-12: View Semester List | | |
| --- | --- | --- | --- |
| Create by: | AnNPN | Date Created: | 19/07/2023 |
| Actor: | Lecturer | | |
| Priority: | High | Frequency of Use: | Usually |
| Trigger: | Lecturer wants to view the semester list. | | |
| Description: | This feature allows the Lecturer to view a list of semesters. | | |
| Pre-conditions: | * The Lecturer is logged into the system. * The institution has defined existing semesters for lecturer to view and select. | | |
| Post-conditions: | The lecturer is presented with a list of available semesters on the homepage for enrollment or further actions. | | |
| Main Flow: | Step | Actor action | System response |
| 1 | Lecturer logs into the system. | The system automatically displays a list of available semesters on the homepage:   * Semester Name (String) * Start Date (date) * End Date (date) * Status (String) |
| Alternative Flows: | N/A | | |
| Exception | N/A | | |
| Business Rules: | N/A | | |
| Assumptions: | Internet connection is available. | | |

| ID and Name: | UCL-13: View Group list in Course | | |
| --- | --- | --- | --- |
| Create by: | AnNPN | Date Created: | 28/07/2023 |
| Actor: | Lecturer | | |
| Priority: | high | Frequency of Use: | sometimes |
| Trigger: | Lecturer performs the " View Group list in Course" process by clicking the List Group button on the Course Detail page. | | |
| Description: | This feature allows the Lecturer to view a list of groups of a particular course. | | |
| Pre-conditions: | * The Lecturer is logged into the system. * Course must have a group. * Lecturer must be able to access the Course of the Subject. | | |
| Post-conditions: | Lecturer successfully displays the group list of a course. | | |
| Main Flow: | Step | Actor action | System response |
| 1 | Lecturer accesses a specific course. | The system will display:   * “List Topic” button * “List Group” button * List of students in class. |
| 2 | Lecturer clicks on the “List Group" button. | The system will display:   * Lecturer Name (String) * The system displays the group list table:   + No (int)   + Group Code (String)   + Group Name (String)   + Topic Code (String)   + View Detail button   + Delete button * Create Group button * Import Group button |
| Alternative Flows: | N/A | | |
| Exception | N/A | | |
| Business Rules: | N/A | | |
| Assumptions: | Internet connection is available. | | |

| ID and Name: | UCL-14: View Detail Group | | |
| --- | --- | --- | --- |
| Create by: | AnNPN | Date Created: | 28/07/2023 |
| Actor: | Lecturer | | |
| Priority: | high | Frequency of Use: | sometimes |
| Trigger: | Lecturer wants to view details of a group. | | |
| Description: | This feature allows the Lecturer to view the project and the members of the group. | | |
| Pre-conditions: | * The Lecturer is logged into the system. * Group must have a Project. * Group must have a Member. | | |
| Post-conditions: | Lecturer successfully displays the group project and the list of members of the group. | | |
| Main Flow: | Step | Actor action | System response |
| 1 | Lecturer accesses a specific course. | The system will display:   * “List Topic” button * “List Group” button * List of students in class. |
| 2 | Lecturer clicks on the “List Group" button. | The system will display:   * Lecturer Name (String) * The system displays the group list table:   + No (int)   + Group Code (String)   + Group Name (String)   + Topic Code (String)   + View Detail button   + Delete button * Create Group button * Import Group button |
| 3 | Lecturer clicks on the ‘View Detail' button. | The system will display:   * Project of the group:   + Topic Code (String)   + Topic Name (String)   + Context (String)   + Actors (String)   + Function Requirement (String)   + Notes (String)   + Update Project Topic button * List of group members:   + No (int)   + Student Code (String)   + Student Name (String)   + isLeader: LD or MB (String)   + Start Date (date)   + Update button   + Delete button |
| Alternative Flows: | N/A | | |
| Exception | N/A | | |
| Business Rules: | N/A | | |
| Assumptions: | Internet connection is available. | | |

| ID and Name: | UCL-15: Update Project in Group | | |
| --- | --- | --- | --- |
| Create by: | AnNPN | Date Created: | 28/07/2023 |
| Actor: | Lecturer | | |
| Priority: | high | Frequency of Use: | sometimes |
| Trigger: | Lecturer wants to update a group's project. | | |
| Description: | This feature allows the Lecturer to update the project topic of a specific group. | | |
| Pre-conditions: | * The Lecturer is logged into the system. * Group must have a Project. * Group must have a Member. | | |
| Post-conditions: | Lecturer successfully updates the project topic and displays the project of that group again. | | |
| Main Flow: | Step | Actor action | System response |
| 1 | Lecturer accesses a specific course. | The system will display:   * “List Topic” button * “List Group” button * List of students in class. |
| 2 | Lecturer clicks on the “List Group" button. | The system will display:   * Lecturer Name (String) * The system displays the group list table:   + No (int)   + Group Code (String)   + Group Name (String)   + Topic Code (String)   + View Detail button   + Delete button * Create Group button * Import Group button |
| 3 | Lecturer clicks on the “View Detail" button. | The system will display:   * Project of the group:   + Topic Code (String)   + Topic Name (String)   + Context (String)   + Actors (String)   + Function Requirement (String)   + Notes (String)   + Update Project Topic button * List of group members:   + No (int)   + Student Code (String)   + Student Name (String)   + isLeader: LD or MB (String)   + Start Date (date)   + Update button   + Delete button |
|  | 4 | Lecturer fills in Topic Code and clicks the “Update Project Topic" button. | The system displays the message “Update Successfully !!”. |
| Alternative Flows: | N/A | | |
| Exception | No | Cause | System response |
| 1 | If the Lecturer fills in the wrong form of Topic Code. | The system displays the message “Topic is not Match !!!”. |
| 2 | If the Lecturer fills in the Topic Code, it is not in the subject's topic list. | The system displays the message “Topic Code could not be found!!!”. |
| Business Rules: | Topic Code must follow the form. (ex: HRM-501)  Topic Code must be in the topic list of the subject. | | |
| Assumptions: | Internet connection is available. | | |

| ID and Name: | UCL-16: Update Member in Group | | |
| --- | --- | --- | --- |
| Create by: | AnNPN | Date Created: | 28/07/2023 |
| Actor: | Lecturer | | |
| Priority: | high | Frequency of Use: | sometimes |
| Trigger: | Lecturer wants to update members of a group. | | |
| Description: | This feature allows the Lecturer to update isLeader of the group member of a specific group. | | |
| Pre-conditions: | * The Lecturer is logged into the system. * Group must have a Project. * Group must have a Member. | | |
| Post-conditions: | Lecturer successfully updates isLeader of the group member and displays the member list of that group again. | | |
| Main Flow: | Step | Actor action | System response |
| 1 | Lecturer accesses a specific course. | The system will display:   * “List Topic” button * “List Group” button * List of students in class. |
| 2 | Lecturer clicks on the “List Group" button. | The system will display:   * Lecturer Name (String) * The system displays the group list table:   + No (int)   + Group Code (String)   + Group Name (String)   + Topic Code (String)   + View Detail button   + Delete button * Create Group button * Import Group button |
| 3 | Lecturer clicks on the “View Detail" button. | The system will display:   * Project of the group:   + Topic Code (String)   + Topic Name (String)   + Context (String)   + Actors (String)   + Function Requirement (String)   + Notes (String)   + Update Project Topic button * List of group members:   + No (int)   + Student Code (String)   + Student Name (String)   + isLeader: LD or MB (String)   + Start Date (date)   + Update button   + Delete button |
|  | 4 | Lecturer selects isLeader for the member and presses the “Update” button. | The system displays the message “Update Successfully !!”. |
| Alternative Flows: | N/A | | |
| Exception | No | Cause | System response |
| 1 | If the lecturer fills in the wrong form of Topic Code. | The system displays the message “Topic is not Match !!!”. |
| 2 | If the Lecturer fills in the Topic Code, it is not in the subject's topic list. | The system displays the message “Topic Code could not be found!!!”. |
| Business Rules: | Topic Code must follow the form. (ex: HRM-501)  Topic Code must be in the topic list of the subject. | | |
| Assumptions: | Internet connection is available. | | |

| ID and Name: | UCL-17: Delete Member in Group | | |
| --- | --- | --- | --- |
| Create by: | AnNPN | Date Created: | 28/07/2023 |
| Actor: | Lecturer | | |
| Priority: | high | Frequency of Use: | sometimes |
| Trigger: | Lecturer wants to delete members of a group. | | |
| Description: | This feature allows the Lecturer to delete the group member of a specific group. | | |
| Pre-conditions: | * The Lecturer is logged into the system. * Group must have a Project. * Group must have a Member. | | |
| Post-conditions: | Lecturer successfully deletes the group member and displays the member list of that group again. | | |
| Main Flow: | Step | Actor action | System response |
| 1 | Lecturer accesses a specific course. | The system will display:   * “List Topic” button * “List Group” button * List of students in class. |
| 2 | Lecturer clicks on the List Group button. | The system will display:   * Lecturer Name (String) * The system displays the group list table:   + No (int)   + Group Code (String)   + Group Name (String)   + Topic Code (String)   + View Detail button   + Delete button * Create Group button * Import Group button |
| 3 | Lecturer clicks on the View Detail button. | The system will display:   * Project of the group:   + Topic Code (String)   + Topic Name (String)   + Context (String)   + Actors (String)   + Function Requirement (String)   + Notes (String)   + Update Project Topic button * List of group members:   + No (int)   + Student Code (String)   + Student Name (String)   + isLeader: LD or MB (String)   + Start Date (date)   + Update button   + Delete button |
|  | 4 | Lecturer clicks on the “Delete" button on the group member list table. | The system displays the message “Delete Successfully !!”. |
| Alternative Flows: | N/A | | |
| Exception | No | Cause | System response |
| 1 | If the lecturer deletes a member of a group that has only one member. | The system displays the message “Group has one member can not deleted !!”. |
| Business Rules: | Group must have more than one member. | | |
| Assumptions: | Internet connection is available. | | |

| ID and Name: | UCL-18: Delete Group | | |
| --- | --- | --- | --- |
| Create by: | AnNPN | Date Created: | 28/07/2023 |
| Actor: | Lecturer | | |
| Priority: | high | Frequency of Use: | sometimes |
| Trigger: | Lecturer wants to delete a group of the course. | | |
| Description: | This feature allows the Lecturer to delete a group of a particular course. | | |
| Pre-conditions: | * The Lecturer is logged into the system. * The course must have a group. * Lecturer must be able to access the Course of the Subject. | | |
| Post-conditions: | Lecturer successfully displays the group list of a course. | | |
| Main Flow: | Step | Actor action | System response |
| 1 | Lecturer accesses a specific course. | The system will display:   * “List Topic” button * “List Group” button * List of students in class. |
| 2 | Lecturer clicks on the “List Group" button. | The system will display:   * Lecturer Name (String) * The system displays the group list table:   + No (int)   + Group Code (String)   + Group Name (String)   + Topic Code (String)   + View Detail button   + Delete button * Create Group button * Import Group button |
|  | 3 | Lecturer clicks on the “Delete” button. | The system displays the message “Delete Successfully !!”. |
| Alternative Flows: | N/A | | |
| Exception | N/A | | |
| Business Rules: | N/A | | |
| Assumptions: | Internet connection is available. | | |

| ID and Name: | UCL-19: Create Group by form | | |
| --- | --- | --- | --- |
| Create by: | AnNPN | Date Created: | 28/07/2023 |
| Actor: | Lecturer | | |
| Priority: | high | Frequency of Use: | sometimes |
| Trigger: | Lecturer wants to create a group in the course by form. | | |
| Description: | This feature allows the Lecturer to create a group of a particular course by form. | | |
| Pre-conditions: | * The Lecturer is logged into the system. * The course must have a group. * Lecturer must be able to access the Course of the Subject. | | |
| Post-conditions: | Lecturer successfully displays the group list of a course. | | |
| Main Flow: | Step | Actor action | System response |
| 1 | Lecturer accesses a specific course. | The system will display:   * “List Topic” button * “List Group” button * List of students in class. |
| 2 | Lecturer clicks on the List Group button. | The system will display:   * Lecturer Name (String) * The system displays the group list table:   + No (int)   + Group Code (String)   + Group Name (String)   + Topic Code (String)   + View Detail button   + Delete button * Create Group button * Import Group button |
|  | 3 | Lecturer clicks on the “Create Group" button. | Display system:  + Group Name (String) (Exception 1)  + Student Code (String)  (Exception 2) (Exception 3) (Exception 4)  + Topic Code (String) (Exception 5) (Exception 6) (Exception 7)  + Context (String)  + Actors (String)  + Function Requirement (String)  + Note (String) |
|  | 4 | Lecturer fills in the information and clicks on the "Create" button. | The system displays the message “Create Successfully !!”. |
| Alternative Flows: | N/A | | |
| Exception | No | Cause | System response |
| 1 | Lecturer fills in the existing Group Name. | The system displays the message “Group Name is duplicated!!!”. |
| 2 | Lecturer entered the wrong Student Code form. | The system displays the message “Student Code is not Match !!!”. |
| 3 | Lecturer fills in Student Code not in course. | The system displays the message “Student is not in Course!!!”. |
| 4 | Lecturer fills in the student Code of students who already have a group. | The system displays the message “Student has a Group !!”. |
| 5 | Lecturer filled in the Topic Code in the wrong form. | The system displays the message “Topic Code is not Match !!!. |
| 6 | Lecturer fills in the Topic Code that is not in the subject. | The system displays the message “Topic Code could not be found!!!”. |
| 7 | Lecturer fills in the Topic Code that has been registered by another group. | The system displays the message “Topic is assign!!!”. |
| Business Rules: | Group Name cannot be duplicated.  Student Code must fill in the correct form. (ex: SE16-123)  Student Code must be included in the course.  Student Code does not have a group.  Topic Code must fill in the correct form. (ex: HRM-501)  Topic Code must be in subject.  Topic Code has not been registered. | | |
| Assumptions: | Internet connection is available. | | |

| ID and Name: | UCL-20: Import Group by file | | |
| --- | --- | --- | --- |
| Create by: | AnNPN | Date Created: | 28/07/2023 |
| Actor: | Lecturer | | |
| Priority: | high | Frequency of Use: | sometimes |
| Trigger: | Lecturer wants to create a group in the course by file. | | |
| Description: | This feature allows the Lecturer to create a group of a particular course by file. | | |
| Pre-conditions: | * The Lecturer is logged into the system. * The course must have a group. * Lecturer must be able to access the Course of the Subject. | | |
| Post-conditions: | Lecturer successfully displays the group list of a course. | | |
| Main Flow: | Step | Actor action | System response |
| 1 | Lecturer accesses a specific course. | The system will display:   * “List Topic” button * “List Group” button * List of students in class. |
| 2 | Lecturer clicks on the List Group button. | The system will display:   * Lecturer Name (String) * The system displays the group list table:   + No (int)   + Group Code (String)   + Group Name (String)   + Topic Code (String)   + View Detail button   + Delete button * Create Group button * Import Group button |
|  | 3 | Lecturer clicks on the “Import Group" button. | The system will display:  + Choose file button  + Upload button |
|  | 4 | Lecturer fills in the information and clicks on the "Create" button. | The system displays the message “Create Successfully !!”. |
| Alternative Flows: | N/A | | |
| Exception | No | Cause | System response |
| 1 | Group creation failed. | The system displays the message “Cannot create group”. |
| 2 | Insert project failed. | The system displays the message “Cannot insert project group”. |
| Business Rules: | NA | | |
| Assumptions: | Internet connection is available. | | |

### 2.4.3 Academic Staff

| ID and Name: | UCA-01 Add Student | | | | |
| --- | --- | --- | --- | --- | --- |
| Create by: | Linhnbn | | Date Created: | 20/6/2023 | |
| Actor: | Academic Staff | | | | |
| Priority: | High | | Frequency of Use: | Often | |
| Trigger: | Academic staff wants to add students to the system | | | | |
| Description: | This use case will help Academic staff add students to the system | | | | |
| Pre-conditions: | The actor has logged into the system. | | | | |
| Post-conditions: | Academic Staff will be able to successfully add a student to the system | | | | |
| Main Flow: | Step | Actor action | | | System response |
| 1 | Academic staff click ‘’Student’’ in the sidebar | | | The system will display a list of students already in the system |
|  | 2 | Academic staff click ‘’add Student’’ button | | | The system presents a form for entering the details of the new student including:   * Student ID(String) * Name(String) * Email(String) * Major:(String)   ‘’Add’’ button |
|  | 3 | Academic Staff enters the student's information including:   * Student ID(String) * Name(String) * Email(String) * Major:(String) | | |  |
|  | 4 | Academic Staff click the ‘’Add’’ button | | | The system displays a confirmation message to the Academic Staff member, ‘’Add successfully!’’ |
| Exception: | No | Cause | | | System Response |
|  | 1 | Input Student ID wrong format | | | The system displays a message to the Academic Staff member, ‘’ID wrong format’’ |
|  | 2 | The Email not exist in the system | | | The system displays a message to the Academic Staff member, ‘’Email is not exsited!!’’ |
|  | 3 | The Email already create into the system, duplicate email | | | The system displays a message to the Academic Staff member, ‘’Email is exsited!!’’ |
|  | 4 | The ID is already create into the system, duplicate ID | | | The system displays a message to the Academic Staff member, ‘’ID duplicate’’ |
|  | 5 | Input Email wrong format | | | The system displays a message to the Academic Staff member, ‘’Email is wrong format!!’’ |
| Alternative Flows: | N/A | | | | |
| Business Rules: | * The student ID must be unique and should follow a specific format determined by the institution. * The Email must exist in the system * The student’s Email must have a tail “@fpt.edu.vn” * Students do not duplicate email and ID * Major must exist in the system | | | | |
| Assumptions: | Internet connection is available. | | | | |

| ID and Name: | UCA-02: View Student list | | | | |
| --- | --- | --- | --- | --- | --- |
| Create by: | Linhnbn | | Date Created: | 20/6/2023 | |
| Actor: | Academic Staff | | | | |
| Priority: | high | | Frequency of Use: | Often | |
| Trigger: | Academic staff want to view the students list in the system | | | | |
| Description: | This use case will help Academic view the students list already in the system | | | | |
| Pre-conditions: | The actor has logged into the system. | | | | |
| Post-conditions: | Academic Staff will be able to successfully view the students list | | | | |
| Main Flow: | Step | Actor action | | | System response |
| 1 | Academic staff click ‘’Student’’ in the sidebar | | | The system will display  a list of students already in the system, search bar and ‘’Add Student‘’ button  List containing column:   * No(Number) * Student Code(String * Name(String) * Birthday(Date) * Phone Number(Number) * Gender(String) * Address(String) * City(String) * Major(String) * “Update” button |
| Exception: | No | Cause | | | System Response |
| Alternative Flows: | N/A | | | | |
| Business Rules: |  | | | | |
| Assumptions: | Internet connection is available. | | | | |

| ID and Name: | UCA-03: Update Student | | | | |
| --- | --- | --- | --- | --- | --- |
| Create by: | Linhnbn | | Date Created: | 20/6/2023 | |
| Actor: | Academic Staff | | | | |
| Priority: | High | | Frequency of Use: | Often | |
| Trigger: | Academic staff wants to update a student in the students list in the system | | | | |
| Description: | This use case will help Academic update a student the students list already in the system | | | | |
| Pre-conditions: | The actor has logged into the system. | | | | |
| Post-conditions: | Academic Staff will be able to successfully update a student in the students list | | | | |
| Main Flow: | Step | Actor action | | | System response |
| 1 | Academic staff click ‘’Student’’ in the sidebar | | | The system will display  a list of students already in the system, search bar and ‘’Add Student‘’ button  List containing column:   * No(Number) * Student Code(String * Name(String) * Birthday(Date) * Phone Number(Number) * Gender(String) * Address(String) * City(String) * Major(String) * “Update” button |
|  | 2 | Academic Staff can update column:   * Student name (String) | | |  |
|  | 3 | Academic Staff change info a student and click ‘’Update’’ button | | | System will display the message ‘’Update successfully’’ |
| Exception: | No | Cause | | | System Response |
| Alternative Flows: | N/A | | | | |
| Business Rules: | Student names cannot be blank | | | | |
| Assumptions: | Internet connection is available. | | | | |

| ID and Name: | UCA-04: Search Student by name | | | | |
| --- | --- | --- | --- | --- | --- |
| Create by: | Linhnbn | | Date Created: | 20/6/2023 | |
| Actor: | Academic Staff | | | | |
| Priority: | High | | Frequency of Use: | Often | |
| Trigger: | Academic staff wants to search a student in the students list in the system | | | | |
| Description: | This use case will help Academic search a student the students list already in the system | | | | |
| Pre-conditions: | The actor has logged into the system. | | | | |
| Post-conditions: | Academic Staff will be able to successfully update a student in the students list | | | | |
| Main Flow: | Step | Actor action | | | System response |
| 1 | Academic staff click ‘’Student’’ in the sidebar | | | The system will display  a list of students already in the system, search bar and ‘’Add Student‘’ button  List containing column:   * No(Number) * Student Code(String * Name(String) * Birthday(Date) * Phone Number(Number) * Gender(String) * Address(String) * City(String) * Major(String) * “Update” button |
|  | 2 | Academic Staff input a character in search bar | | |  |
|  | 3 | Academic Staff click ‘’Search’’ button | | | The system will display a list of students by name containing the character the Academic Staff searched for  List containing column:   * No(Number) * Student Code(String * Name(String) * Birthday(Date) * Phone Number(Number) * Gender(String) * Address(String) * City(String) * Major(String) * “Update” button |
| Exception: | No | Cause | | | System Response |
|  | 1 | the character entered by the Academic Staff does not match any student's name | | | System will display the message ‘’No Record’’ |
| Alternative Flows: | N/A | | | | |
| Business Rules: | * The character entered by the Academic Staff must match student's name | | | | |
| Assumptions: | Internet connection is available. | | | | |

| ID and Name: | UCA-05: View Lecturer list | | | | |
| --- | --- | --- | --- | --- | --- |
| Create by: | Linhnbn | | Date Created: | 20/6/2023 | |
| Actor: | Academic Staff | | | | |
| Priority: | High | | Frequency of Use: | Often | |
| Trigger: | Academic staff want to view the lecturer list in the system | | | | |
| Description: | This use case will help Academic view the lecturer list already in the system | | | | |
| Pre-conditions: | The actor has logged into the system. | | | | |
| Post-conditions: | Academic Staff will be able to successfully view the lecturer list | | | | |
| Main Flow: | Step | Actor action | | | System response |
| 1 | Academic staff click ‘’Lecturer’’ in the sidebar | | | The system will display  a list of lecturers already in the system, search bar, and ‘’Add Lecturer‘’ button  List containing column:   * No(String) * Lecture Code (String) * Name (String) * Birthday(Number) * Phone Number(Number) * Gender(String) * Address(String) * City(String) * “Update” button |
| Exception: | No | Cause | | | System Response |
| Alternative Flows: | N/A | | | | |
| Business Rules: | N/A | | | | |
| Assumptions: | Internet connection is available. | | | | |

| ID and Name: | UCA-06 Add Lecturer | | | | |
| --- | --- | --- | --- | --- | --- |
| Create by | Linhnbn | | Date Created: | 20/6/2023 | |
| Actor: | Academic Staff | | | | |
| Priority: | High | | Frequency of Use: | Often | |
| Trigger: | Academic staff wants to add lecturers to the system | | | | |
| Description: | This use case will help Academic staff add lecturer to the system | | | | |
| Pre-conditions: | The actor has logged into the system. | | | | |
| Post-conditions: | Academic Staff will be able to successfully add a lecturer to the system | | | | |
| Main Flow: | Step | Actor action | | | System response |
| 1 | Academic staff click ‘’Lecturer’’ in the sidebar | | | The system will display a list of lecturer already in the system |
|  | 2 | Academic staff click ‘’add Lecturer’’ button | | | The system presents a form for entering the details of the new lecturer including:   * Lecturer ID(String) * Name(String) * Email(String)   ‘’Add’’ button |
|  | 3 | The system presents a form for entering the details of the new lecturer including:   * Lecturer ID(String) * Name(String) * Email(String) | | |  |
|  | 4 | Academic Staff click the ‘’Add’’ button | | | The system displays a confirmation message to the Academic Staff member, ‘’Add successfully!’’ |
| Exception: | No | Cause | | | System Response |
|  | 1 | Input Lecturer ID wrong form | | | The system displays a message to the Academic Staff member, ‘’LectureID is wrong form !!!’’ |
|  | 2 | Input Email wrong form | | | The system displays a message to the Academic Staff member, ‘’Email is wrong form !!!’’ |
|  | 3 | The Email not exist in the system | | | The system displays a message to the Academic Staff member, ‘’Email is not exsited!!’’ |
|  | 4 | The Email already create into the system, duplicate email | | | The system displays a message to the Academic Staff member, ‘’Email is exsited!!’’ |
|  | 5 | The ID is already create into the system, duplicate ID | | | The system displays a message to the Academic Staff member, ‘’ID duplicate’’ |
| Alternative Flows: | N/A | | | | |
| Business Rules: | * The lecturer ID must be unique and should follow a specific format determined by the institution. * The Email must exist in the system * The lecturer’s Email must have a tail “@fe.edu.vn” * Lecturer does not duplicate | | | | |
| Assumptions: | Internet connection is available. | | | | |

| ID and Name: | UCA-07: Update Lecturer | | | | |
| --- | --- | --- | --- | --- | --- |
| Create by: | Linhnbn | | Date Created: | 20/6/2023 | |
| Actor: | Academic Staff | | | | |
| Priority: | High | | Frequency of Use: | Often | |
| Trigger: | Academic staff wants to update a lecturer in the students list in the system | | | | |
| Description: | This use case will help Academic update a lecturer in the lecturer list is already in the system | | | | |
| Pre-conditions: | The actor has logged into the system. | | | | |
| Post-conditions: | Academic Staff will be able to successfully update a lecturer in the lecturer list | | | | |
| Main Flow: | Step | Actor action | | | System response |
| 1 | Academic staff click ‘’Lecturer’’ in the sidebar | | | The system will display  a list of lecturers already in the system, search bar, and ‘’Add Lecturer‘’ button  List containing column:   * No(String) * Lecture Code (String) * Name (String) * Birthday(Number) * Phone Number(Number) * Gender(String) * Address(String) * City(String) * “Update” button |
|  | 2 | Academic Staff can update column:   * Lecturer name | | |  |
|  | 3 | Academic Staff change info a lecturert and click ‘’Update’’ button | | | System will display the message ‘’Update successfully’’ |
| Exception: | No | Cause | | | System Response |
| Alternative Flows: | N/A | | | | |
| Business Rules: | N/A | | | | |
| Assumptions: | Internet connection is available. | | | | |

| ID and Name: | UCA-08: Search Lecturer by name | | | | |
| --- | --- | --- | --- | --- | --- |
| Create by | Linhnbn | | Date Created: | 20/6/2023 | |
| Actor: | Academic Staff | | | | |
| Priority: | High | | Frequency of Use: | Often | |
| Trigger: | Academic staff wants to search a lecturer in the lecturer list in the system | | | | |
| Description: | This use case will help Academic search a lecturer the lecturer list already in the system | | | | |
| Pre-conditions: | The actor has logged into the system. | | | | |
| Post-conditions: | Academic Staff will be able to successfully search a lecturer in the lecturer list | | | | |
| Main Flow: | Step | Actor action | | | System response |
| 1 | Academic staff click ‘’Lecturer’’ in the sidebar | | | The system will display  a list of lecturers already in the system, search bar, and ‘’Add Lecturer‘’ button  List containing column:   * No(String) * Lecture Code (String) * Name (String) * Birthday(Number) * Phone Number(Number) * Gender(String) * Address(String) * City(String) * “Update” button |
|  | 2 | Academic Staff input a character in the search bar | | |  |
|  | 3 | Academic Staff click ‘’Search’’ button | | | The system will display a list of lecturer by name containing the character the Academic Staff searched for  List containing column:   * No(String) * Lecture Code (String) * Name (String) * Birthday(Number) * Phone Number(Number) * Gender(String) * Address(String) * City(String) * “Update” button |
| Exception: | No | Cause | | | System Response |
| Alternative Flows: | N/A | | | | |
| Business Rules: | N/A | | | | |
| Assumptions: | Internet connection is available. | | | | |

| ID and Name: | UCA-09: Update Presentation | | | | |
| --- | --- | --- | --- | --- | --- |
| Create by | Linhnbn | | Date Created: | 20/6/2023 | |
| Actor: | Academic Staff | | | | |
| Priority: | High | | Frequency of Use: | Often | |
| Trigger: | Academic staff wants to update a presentation in the presentation list in the system | | | | |
| Description: | This use case will help Academic update a presentation in the presentation list | | | | |
| Pre-conditions: | The actor has logged into the system. | | | | |
| Post-conditions: | Academic Staff will be able to successfully update a presentation in the presentation list | | | | |
| Main Flow: | Step | Actor action | | | System response |
| 1 | Academic staff click ‘’Presentation’’ in the sidebar | | | The system will display  a list of presentations, search bar  List containing column:   * ID * Type * Course Code * Group Code * Present Date * Room * Create Date * Status * Subject Code * Time * Publish * Update |
|  | 2 | Academic Staff can update column:   * Present Date * Room * Time | | |  |
|  | 3 | Academic Staff change info a Presentation and click ‘’Update’’ button | | | System will display the message ‘’Update successfully’’ |
| Exception: | No | Cause | | | System Response |
| Alternative Flows: | N/A | | | | |
| Business Rules: |  | | | | |
| Assumptions: | Internet connection is available. | | | | |

| ID and Name: | UCA-10: View Presentation List | | | | |
| --- | --- | --- | --- | --- | --- |
| Create by | Linhnbn | | Date Created: | 20/6/2023 | |
| Actor: | Academic Staff | | | | |
| Priority: | High | | Frequency of Use: | Often | |
| Trigger: | Academic staff wants to view presentation list in the system | | | | |
| Description: | This use case will help Academic view presentation list in this semester | | | | |
| Pre-conditions: | The actor has logged into the system. | | | | |
| Post-conditions: | Academic Staff will be able to successfully view the presentation list this semester | | | | |
| Main Flow: | Step | Actor action | | | System response |
| 1 | Academic staff click ‘’Presentation’’ in the sidebar | | | The system will display  a list of presentations, search bar  List containing column:   * ID * Type * Course Code * Group Code * Present Date * Room * Create Date * Status * Subject Code * Time * Publish * Update |
| Exception: | No | Cause | | | System Response |
| Alternative Flows: | N/A | | | | |
| Business Rules: |  | | | | |
| Assumptions: | Internet connection is available. | | | | |

| ID and Name: | UCA-11: Search Presentation by Date | | | | |
| --- | --- | --- | --- | --- | --- |
| Create by | Linhnbn | | Date Created: | 20/6/2023 | |
| Actor: | Academic Staff | | | | |
| Priority: | High | | Frequency of Use: | Often | |
| Trigger: | Academic staff wants to search for a presentation in the presentation list in the system | | | | |
| Description: | This use case will help Academic search a presentation the presentation list | | | | |
| Pre-conditions: | The actor has logged into the system. | | | | |
| Post-conditions: | Academic Staff will be able to successfully search for a presentation in the presentation list | | | | |
| Main Flow: | Step | Actor action | | | System response |
| 1 | Academic staff click ‘presentation’’ in the sidebar | | | The system will display  a list of presentations, a search bar  List containing column:   * ID * Type * Course Code * Group Code * Present Date * Room * Create Date * Status * Subject Code * Time * Publish * Update |
|  | 2 | Academic Staff input a date in the search bar | | |  |
|  | 3 | Academic Staff click ‘’Search’’ button | | | The system will display a list of presentation by date containing the date in ‘’time’’ column the Academic Staff searched for  List containing column:   * ID * Type * Course Code * Group Code * Present Date * Room * Create Date * Status * Subject Code * Time * Publish * Update |
| Exception: | No | Cause | | | System Response |
| Alternative Flows: | N/A | | | | |
| Business Rules: | N/A | | | | |
| Assumptions: | Internet connection is available. | | | | |

| ID and Name: | UCA-12: View Account list | | | | |
| --- | --- | --- | --- | --- | --- |
| Create by: | Linhnbn | | Date Created: | 20/6/2023 | |
| Actor: | Academic Staff | | | | |
| Priority: | High | | Frequency of Use: | Often | |
| Trigger: | Academic staff wants to view the account list in the system | | | | |
| Description: | This use case will help Academic view the account list already in the system | | | | |
| Pre-conditions: | The actor has logged into the system. | | | | |
| Post-conditions: | Academic Staff will be able to successfully view the account list | | | | |
| Main Flow: | Step | Actor action | | | System response |
| 1 | Academic staff click ‘’account’’ in the sidebar | | | The system will display  a list of accounts already in the system, a search bar, and ‘’Add account‘’ button  List containing column:   * No.(String) * Account Code (String) * Email (String) * Role(String) * Full Name(String) * Status (Boolen) * Update button |
| Exception: | No | Cause | | | System Response |
| Alternative Flows: | N/A | | | | |
| Business Rules: |  | | | | |
| Assumptions: | Internet connection is available. | | | | |

| ID and Name: | UCA-13: Update Account | | | | |
| --- | --- | --- | --- | --- | --- |
| Create by: | Linhnbn | | Date Created: | 20/6/2023 | |
| Actor: | Academic Staff | | | | |
| Priority: | High | | Frequency of Use: | Often | |
| Trigger: | Academic staff wants to update an account in the account list in the system | | | | |
| Description: | This use case will help Academic update an account the account list already in the system | | | | |
| Pre-conditions: | The actor has logged into the system. | | | | |
| Post-conditions: | Academic Staff will be able to successfully update an account in the account list | | | | |
| Main Flow: | Step | Actor action | | | System response |
| 1 | Academic staff click ‘’ account’’ in the sidebar | | | The system will display  a list of accounts already in the system, a search bar, and ‘’Add account‘’ button  List containing column:   * No.(String) * Account Code (String) * Email (String) * Role(String) * Full Name(String) * Status (Boolen) * Update button |
|  | 2 | Academic Staff can update column:   * Role(String) * Full Name(String) * Status (Boolen) | | |  |
|  | 3 | Academic Staff change info an account and click ‘’Update’’ button | | | System will display the message ‘’Update successfully’’ |
| Exception: | No | Cause | | | System Response |
| Alternative Flows: | N/A | | | | |
| Business Rules: |  | | | | |
| Assumptions: | Internet connection is available. | | | | |

| **ID and Name:** | UCA-14: Add Account from file | | |
| --- | --- | --- | --- |
| Create by: | Linhnbn | Date Created: | 20/06/2023 |
| Actor: | Academic staff | | |
| Priority: | high | Frequency of Use: | often |
| Trigger: | The actor wants to add the accounts. | | |
| Description: | This use case outlines the process add more accounts to the system | | |
| Pre-conditions: | The Academic staff is logged into the system.  Accounts exist within the system.  The account information has been approved and added by the academic staff | | |
| Post-conditions: | Academic Staff will be able to successfully add more accounts to the system | | |
| Main Flow: | Step | Actor action | System response |
| 1 | Academic staff tap the button ”Account” in the sidebar | The system will display  a list of accounts already in the system, a search bar, and ‘’Add account form file‘’ button  List containing column:   * No.(String) * Account Code (String) * Email (String) * Role(String) * Full Name(String) * Status (Boolen) * Update button |
|  | 2 | Academic staff tap the button ” Add account from file” in the sidebar | The system displays the sub-tab:  Input your file to add Account Button:   * Choose file * Upload |
| Exception | No | Cause | System Response |
|  | N/A | N/A | N/A |
| Alternative Flows: | N/A | | |
| Business Rules: | Note in file .xlsx: +column: Email, password, Full Name, Role, Account Code, Status Email must true to form: abc@fpt.edu.vn  Password must has one character and one number | | |
| Assumptions: | Internet connection is available. | | |

| ID and Name: | UCA-15: Search Account by Name | | | | |
| --- | --- | --- | --- | --- | --- |
| Create by | Linhnbn | | Date Created: | 20/6/2023 | |
| Actor: | Academic Staff | | | | |
| Priority: | High | | Frequency of Use: | Often | |
| Trigger: | Academic staff wants to search for an Account in the Account list in the system | | | | |
| Description: | This use case will help Academic search an Account the Account list already in the system | | | | |
| Pre-conditions: | The actor has logged into the system. | | | | |
| Post-conditions: | Academic Staff will be able to successfully search for an Account in the Account list | | | | |
| Main Flow: | Step | Actor action | | | System response |
| 1 | Academic staff click ‘’Account’’ in the sidebar | | | The system will display  a list of accounts already in the system, a search bar, and ‘’Add account form file‘’ button  List containing column:   * No.(String) * Account Code (String) * Email (String) * Role(String) * Full Name(String) * Status (Boolen) * Update button |
|  | 2 | Academic Staff input a character in the search bar | | |  |
|  | 3 | Academic Staff click ‘’Search’’ button | | | The system will display a list of accounts by full name containing the character the Academic Staff searched for  List containing column:   * No.(String) * Account Code (String) * Email(String) * Role(String) * Full Name(String) * Status(String) * Update button |
| Exception: | No | Cause | | | System Response |
| Alternative Flows: | N/A | | | | |
| Business Rules: | N/A | | | | |
| Assumptions: | Internet connection is available. | | | | |

| ID and Name: | UCA-16: Create Semester | | | | |
| --- | --- | --- | --- | --- | --- |
| Create by | Linhnbn | | Date Created: | 20/6/2023 | |
| Actor: | Academic Staff | | | | |
| Priority: | High | | Frequency of Use: | Often | |
| Trigger: | Academic staff wants to Create a semester to the system | | | | |
| Description: | This use case will help Academic staff add students to the system | | | | |
| Pre-conditions: | The actor has logged into the system. | | | | |
| Post-conditions: | Academic Staff will be able to successfully create a semester to the system | | | | |
| Main Flow: | Step | Actor action | | | System response |
| 1 | Academic staff click ‘’Semester’’ in the sidebar | | | The system will display a list of semesters already in the system |
|  | 2 | Academic staff click ‘’add Semster’’ button | | | The system presents a form for entering the details of the new student including:   * Semester ID (int) * Year(int) * Name(String) * Status(String) * Start date(date) * Ends date(date)   ‘’Add’’ button |
|  | 3 | Academic Staff enters the semester's information including:   * Semester ID (int) * Year(int) * Name(String) * Status(String) * Start date(date) * Ends date(date) | | |  |
|  | 4 | Academic Staff click the ‘’Add’’ button | | | The system displays a confirmation message to the Academic Staff member, ‘’Add successfully!’’ |
| Exception: | No | Cause | | | System Response |
|  |  |  | | |  |
| Alternative Flows: | N/A | | | | |
| Business Rules: | * Semester ID do not duplicate * Semester Name do not duplicate * Only one semester has status available | | | | |
| Assumptions: | Internet connection is available. | | | | |

| ID and Name: | UCA-17: View Semester List | | | | |
| --- | --- | --- | --- | --- | --- |
| Create by | Linhnbn | | Date Created: | 20/6/2023 | |
| Actor: | Academic Staff | | | | |
| Priority: | High | | Frequency of Use: | Often | |
| Trigger: | Academic staff wants to view a semester to the system | | | | |
| Description: | This use case will help Academic staff view semester list already in the system | | | | |
| Pre-conditions: | The actor has logged into the system. | | | | |
| Post-conditions: | Academic Staff will be able to successfully view the semester list already in the system | | | | |
| Main Flow: | Step | Actor action | | | System response |
| 1 | Academic staff click ‘’Semester’’ in the sidebar | | | The system will display a list of semesters already in the system  List containing column:   * Semester ID (int) * Year(int) * Name(String) * Status(String) * Start date(date) * Ends date(date) * Update button * Delete button |
| Exception: | No | Cause | | | System Response |
| Alternative Flows: | N/A | | | | |
| Business Rules: | N/A | | | | |
| Assumptions: | Internet connection is available. | | | | |

| ID and Name: | UCA-18: Update Semester | | | | |
| --- | --- | --- | --- | --- | --- |
| Create by | Linhnbn | | Date Created: | 20/6/2023 | |
| Actor: | Academic Staff | | | | |
| Priority: | High | | Frequency of Use: | Often | |
| Trigger: | Academic staff wants to update a semester in the semester list | | | | |
| Description: | This use case will help Academic staff update a semester in the semester list already in the system | | | | |
| Pre-conditions: | The actor has logged into the system. | | | | |
| Post-conditions: | Academic Staff will be able to successfully update a semester in the semester list already in the system | | | | |
| Main Flow: | Step | Actor action | | | System response |
| 1 | Academic staff click ‘’Semester’’ in the sidebar | | | The system will display a list of semesters already in the system  List containing column:   * Semester ID (int) * Year(int) * Name(String) * Status(String) * Start date(date) * Ends date(date) * Update button * Delete button |
|  | 2 | Academic Staff can update column:   * Year(int) * Name(String) * Status(String) * Start date(date) * Ends date(date) | | |  |
|  | 3 | Academic Staff change info a semester and click ‘’Update’’ button | | | System will display the message ‘’Update successfully’’ |
| Exception: | No | Cause | | | System Response |
| Alternative Flows: | N/A | | | | |
| Business Rules: | * Semester Name do not duplicate * Only one semester has status available | | | | |
| Assumptions: | Internet connection is available. | | | | |

| ID and Name: | UCA-19: View Subject List | | | |
| --- | --- | --- | --- | --- |
| Create by: | HoangTH | | Date Created: | 20/06/2023 |
| Actor: | Academic staff | | | |
| Priority: | high | | Frequency of Use: | often |
| Trigger: | The actor wants to view list of subject. | | | |
| Description: | This use case describes the process of viewing the list of subjects in a semester | | | |
| Pre-conditions: | The Academic staff is logged into the system.  The system displays a list of available subject. | | | |
| Post-conditions: | The system will display the information of the semester's class in the data | | | |
| Main Flow: | Step | | Actor action | System response |
| 1 | | Academic staff tap the button ”view” in the box semester | The system displays the semesters about :  List subject and course in subject |
| Exception | No | | Cause | System response |
|  | N/A | | N/A | N/A |
| Alternative Flows: | | N/A | | |
| Business Rules: | | N/A | | |
| Assumptions: | | Internet connection is available. | | |

| ID and Name: | UCA-20: Create Subject | | |
| --- | --- | --- | --- |
| Create by: | HoangTH | Date Created: | 20/06/2023 |
| Actor: | Academic staff | | |
| Priority: | high | Frequency of Use: | often |
| Trigger: | The actor wants to create subject. | | |
| Description: | This use case describes the process of creating more Subject | | |
| Pre-conditions: | The Academic staff is logged into the system.  Admin must provide valid course creation information | | |
| Post-conditions: | A new subject is created in the system with all the provided details accurately recorded.  The subject is now available for course offerings and other administrative tasks. | | |
| Main Flow: | Step | Actor action | System response |
| 1 | Academic staff tap the button icon “+” in box subject | The system displays the form create subject about :   * Subject ID(String) Exception 1 * Subject Name(String) * Description(String) * Credit(Number) |
| Exception | No | Cause | System response |
|  | 1 | Subject ID is duplication | The subject id has exist |
| Alternative Flows: | N/A | | |
| Business Rules: | N/A | | |
| Assumptions: | Internet connection is available. | | |

| ID and Name: | UCA-21: Delete Subject | | | |
| --- | --- | --- | --- | --- |
| Create by: | HoangTH | | Date Created: | 20/06/2023 |
| Actor: | Academic staff | | | |
| Priority: | high | | Frequency of Use: | often |
| Trigger: | The actor wants to delete subject. | | | |
| Description: | This use case describes the course deletion a subject | | | |
| Pre-conditions: | The Academic staff is logged into the system.  Admin must provide valid course creation information | | | |
| Post-conditions: | System information is deleted successfully  The system will not appear the deleted period’s information in the data | | | |
| Main Flow: | Step | | Actor action | System response |
| 1 | | Academic staff tap the button ”add subject” in the box Subject | The system displays the list subject about :   * Subject ID(string) * Subject Name(string) * Description(String) * Credit(number) * Delete button * Add button |
|  | 2 | | Academic staff tap the button ”delete subject” in subject they want | The system displays the message "delete successfull” |
| Exception | No | | Cause | System response |
|  | N/A | | N/A | N/A |
| Alternative Flows: | | N/A | | |
| Business Rules: | | N/A | | |
| Assumptions: | | Internet connection is available. | | |

| ID and Name: | **UCA-22: View assign topic in subject** | | |
| --- | --- | --- | --- |
| Create by: | HoangTH | Date Created: | 21/06/2023 |
| Actor: | Academic staff | | |
| Priority: | Medium | Frequency of Use: | sometimes |
| Trigger: | The actor wants to update subject. | | |
| Description: | This use case describes the course update a subject | | |
| Pre-conditions: | The Academic staff is logged into the system.  Before updating, the subject list needs to exist in the system.  The administrator needs to provide new information to update the subject list.  The updated information needs to be checked for validity before being applied to the subject list. | | |
| Post-conditions: | The subject list has been successfully updated by the administrator  Information about the subject list has been stored and updated in the database | | |
| Main Flow: | Step | Actor action | System response |
| 1 | Academic staff tap the button ”view” of box course in the subject | The system displays the list subject about list student :  · Student code(String)  · Name (String)  · Birthday(date)  · Phone number (String)  · Gender(string)  · Address(string)  · City(string)  · Email(string)  · Delete button  · Add student button  · List topic button  · List Group button |
| 2 | Academic staff tap the list topic button | The system display the list topic :   * · No(int) * · Topic code (String) * · Topic name (String) * · Short Description (String) * · Full description(String) * · Lecturer name (String) * · Start date (date) * · Modify date (date) * · Delete topic button * · Add topic button |
|  | 3 | Academic staff tap the button ”add topic” on the navbar | The system displays the list topic assigned to admin add:   * · Topic code (String) * · Topic name (String) * · Lecturer name (String) * · Short Description (String) * · Full description(String) * · Add button |
| Alternative Flows: | N/A | | |
| Exception | N/A | | |
| Business Rules: | N/A | | |
| Assumptions: | Internet connection is available. | | |

| ID and Name: | **UCA-23: View topic in subject** | | |
| --- | --- | --- | --- |
| Create by: | HoangTH | Date Created: | 21/06/2023 |
| Actor: | Academic staff | | |
| Priority: | high | Frequency of Use: | often |
| Trigger: | The actor wants to view topic in subject. | | |
| Description: | This use case describes looking at the topics of that course | | |
| Pre-conditions: | The Academic staff is logged into the system.  must have the topic's information available in the system | | |
| Post-conditions: | Information about the topic has been stored and updated in the database | | |
| Main Flow: | Step | Actor action | System response |
| 1 | Academic staff tap the button ”view” in the course box | The system displays the button in navbar about :   * Add student * List topic * List Group * List Student |
| 2 | Academic staff tap the button ”List topic” in the navbar | The system display the list topic :   * No(int) * Topic code (String) * Topic name (String) * Short Description (String) * Full description(String) * Lecturer name (String) * Start date (date) * Modify date (date) * Delete topic button * Add topic button |
| Alternative Flows: | N/A | | |
| Exception | N/A | | |
| Business Rules: | N/A | | |
| Assumptions: | Internet connection is available. | | |

| ID and Name: | UCA-24: Create course | | |
| --- | --- | --- | --- |
| Create by | HoangTH | Date Created: | 20/6/2023 |
| Actor: | Academic Staff | | |
| Priority: | High | Frequency of Use: | often |
| Trigger: | The actor request to create course | | |
| Description: | This use case will help actor create course in a subject | | |
| Pre-conditions: | The actor has logged into the system.  The actor must complete the form to create a course | | |
| Post-conditions: | the notification system has created the course | | |
| Main Flow: | Step | Actor action | System response |
| 1 | The actor clicks on the icon “+” button on the navbar subject | The system display the list course of each subject:   * Subject code (String) * Lecturer name (String) * Course (String) * Start date (Date) * End date (Date) * Status (String) * Update button * Delete button * Add class button |
|  | 2 | The actor clicks on the add class button | The system display the fields course of each subject:   * Course ID(String)   (Exception 1)   * Course name (String) * Start date (date) * End date (date) * Lecturer name (String) * Create course button |
|  | 3 | Students will click the "Create course" button | The system will print the message line ‘’created course successfully’’ |
| Exception: | No | Cause | System Response |
|  | 1 | Course ID wrong form | The system show message “Course Id is wrong” |
| Alternative Flows: | N/A | | |
| Business Rules: | · course name must be written in the correct format: NJS-1701   * Semester must exist | | |
| Assumptions: | Internet connection is available. | | |

| ID and Name: | UCA-25: view student list of course | | |
| --- | --- | --- | --- |
| Create by | HoangTH | Date Created: | 20/6/2023 |
| Actor: | Academic Staff | | |
| Priority: | High | Frequency of Use: | often |
| Trigger: | The actor request to view student list of course | | |
| Description: | This use case will help actor view student list in a course | | |
| Pre-conditions: | The actor has logged into the system.  The information about student list in course | | |
| Post-conditions: | The system returns student information | | |
| Main Flow: | Step | Actor action | System response |
| 1 | Academic staff tap the button ”view” in the course box | The system displays the list subject about list student :   * Student code(String) * Name (String) * Birthday(date) * Phone number (String) * Gender(string) * Address(string) * City(string) * Email(string) * Delete button * Add student button * List topic button * List Group button |
| Exception: | No | Cause | System Response |
|  | N/A | N/A | N/A |
| Alternative Flows: | N/A | | |
| Business Rules: | N/A | | |
| Assumptions: | Internet connection is available. | | |

| ID and Name: | UCA-26: Add student in course | | |
| --- | --- | --- | --- |
| Create by | HoangTH | Date Created: | 20/6/2023 |
| Actor: | Academic Staff | | |
| Priority: | High | Frequency of Use: | often |
| Trigger: | The actor request to add student in course | | |
| Description: | This use case will help actor add more student in a course | | |
| Pre-conditions: | The actor has logged into the system.  There must be an excel file containing all the information that students need to input | | |
| Post-conditions: | The system returns student information | | |
| Main Flow: | Step | Actor action | System response |
| 1 | Academic staff tap the button ”view” in the course box | The system displays the list subject about list student :   * Student code(String) * Name (String) * Birthday(date) * Phone number (String) * Gender(string) * Address(string) * City(string) * Email(string) * Delete button * Add student button * List topic button * List Group button |
|  | 2 | Academic staff tap the button ”Add Student” in the navbar | The system display add student :   * Choose file * upload |
|  | 3 | Academic staff tap the button ”choose file” and choose the file want to up |  |
|  | 4 | Academic staff tap the button ”Upload” | The system show the message “Upload successfull” |
| Exception: | No | Cause | System Response |
|  | N/A | N/A | N/A |
| Alternative Flows: | N/A | | |
| Business Rules: | · The student ID must be unique and should follow a specific format determined by the institution.  · The program information should match the available programs and meet any prerequisites or requirements set by the institution.  · The student’s Email must have a tail “@fpt.edu.vn” | | |
| Assumptions: | Internet connection is available. | | |

| ID and Name: | UCA-27: Add subject to course | | |
| --- | --- | --- | --- |
| Create by | HoangTH | Date Created: | 20/6/2023 |
| Actor: | Academic Staff | | |
| Priority: | High | Frequency of Use: | often |
| Trigger: | The actor request to add subject to course | | |
| Description: | This use case will help actor add more subject in the semester | | |
| Pre-conditions: | The actor has logged into the system.  The information about subject has been available in the system | | |
| Post-conditions: | The system returns student information | | |
| Main Flow: | Step | Actor action | System response |
| 1 | Academic staff tap the button ”add subject” on the navbar semester | The system displays the list subject about list subject :   * No(int) * Subject code(String) * Subject name(String) * Description(String) * Lecturer name (String) * Course(String) * Credit (number) * Add button |
|  | 2 | Academic staff tap the button ”Add Subject” in the navbar | The system display the navbar has information about that subject |
| Exception: | No | Cause | System Response |
|  | N/A | N/A | N/A |
| Alternative Flows: | N/A | | |
| Business Rules: | N/A | | |
| Assumptions: | Internet connection is available. | | |

| ID and Name: | UCA-28: Delete course | | |
| --- | --- | --- | --- |
| Create by | HoangTH | Date Created: | 20/6/2023 |
| Actor: | Academic Staff | | |
| Priority: | High | Frequency of Use: | often |
| Trigger: | The actor request to delete course | | |
| Description: | This use case will help actor delete course | | |
| Pre-conditions: | The actor has logged into the system.  The information must has been available | | |
| Post-conditions: | The system remove course information | | |
| Main Flow: | Step | Actor action | System response |
| 1 | Academic staff tap the icon button ”+” on the navbar course course | The system display the list course of each subject:   * Subject code (String) * Lecturer name (String) * Course (String) * Start date (Date) * End date (Date) * Status (String) * Update button * Delete button * Add class button |
|  | 2 | Academic staff tap the delete button | The information about this course will be clear by system |
| Exception: | No | Cause | System Response |
|  | N/A | N/A | N/A |
| Alternative Flows: | N/A | | |
| Business Rules: | N/A | | |
| Assumptions: | Internet connection is available. | | |

| ID and Name: | UCA-29: delete student | | |
| --- | --- | --- | --- |
| Create by | HoangTH | Date Created: | 20/6/2023 |
| Actor: | Academic Staff | | |
| Priority: | High | Frequency of Use: | often |
| Trigger: | The actor request to view student list of course | | |
| Description: | This use case will help actor view student list in a course | | |
| Pre-conditions: | The actor has logged into the system.  The information about student list in course has available | | |
| Post-conditions: | The system remove student information | | |
| Main Flow: | Step | Actor action | System response |
| 1 | Academic staff tap the button ”view” of box course in the subject | The system displays the list subject about list student :  · Student code(String)  · Name (String)  · Birthday(date)  · Phone number (String)  · Gender(string)  · Address(string)  · City(string)  · Email(string)  · Delete button  · Add student button  · List topic button  · List Group button |
|  | 2 | Academic staff tap the delete button | The information about this Student will be clear by system |
| Exception: | No | Cause | System Response |
|  | N/A | N/A | N/A |
| Alternative Flows: | N/A | | |
| Business Rules: | N/A | | |
| Assumptions: | Internet connection is available. | | |

| ID and Name: | UCA-30: View group in course | | |
| --- | --- | --- | --- |
| Create by | HoangTH | Date Created: | 20/6/2023 |
| Actor: | Academic Staff | | |
| Priority: | High | Frequency of Use: | often |
| Trigger: | The actor request to add student in course | | |
| Description: | This use case will help actor add more student in a course | | |
| Pre-conditions: | The actor has logged into the system.  There must be an excel file containing all the information that students need to input | | |
| Post-conditions: | The system returns student information | | |
| Main Flow: | Step | Actor action | System response |
| 1 | Academic staff tap the button ”view” in the course box | The system displays the list subject about list student :   * Student code(String) * Name (String) * Birthday(date) * Phone number (String) * Gender(string) * Address(string) * City(string) * Email(string) * Delete button * Add student button * List topic button * List Group button |
|  | 2 | Academic staff tap the button ”List Group” in the navbar | The system display list group :   * No(int) * Group code(String) * Topic code (String) * View Detail button |
| Exception: | No | Cause | System Response |
|  | N/A | N/A | N/A |
| Alternative Flows: | N/A | | |
| Business Rules: | N/A | | |
| Assumptions: | Internet connection is available. | | |

| ID and Name: | UCA-31:view list of course | | |
| --- | --- | --- | --- |
| Create by | HoangTH | Date Created: | 20/6/2023 |
| Actor: | Academic Staff | | |
| Priority: | High | Frequency of Use: | often |
| Trigger: | The actor request to view list of course | | |
| Description: | This use case will help actor view list of course | | |
| Pre-conditions: | The actor has logged into the system.  The information about course list in course | | |
| Post-conditions: | The system returns course information | | |
| Main Flow: | Step | Actor action | System response |
| 1 | Academic staff tap the button ”view” in the semester box | The system displays the button in course:   * List Subject * List course in subject |
| Exception: | No | Cause | System Response |
|  | N/A | N/A | N/A |
| Alternative Flows: | N/A | | |
| Business Rules: | N/A | | |
| Assumptions: | Internet connection is available. | | |

### 2.4.4 Student

| ID and Name: | UCS-01 : Send Application | | |
| --- | --- | --- | --- |
| Create by: | KhangNH | Date Created: | 26/06/2023 |
| Actor: | Student | | |
| Priority: | high | Frequency of Use: | often |
| Trigger: | The student sends an application for a report or registers for a presentation. | | |
| Description: | This use case describes the process of a student sending an application for a report or registering for a presentation. | | |
| Pre-conditions: | The student is logged into the system.  The application period for the program or opportunity is open. | | |
| Post-conditions: | The application is successfully submitted. | | |
| Main Flow: | Step | Actor action | System response |
| 1 | The student access the "Send Application" section | The system displays the type of the application:  Report  Presentation |
| 2 | The student chooses the type for the application. | The system displays the form of the application:  Student ID(String) (Exception 1)  Type (String)  Topic ID (String) (Exception 2)  Subject ID(String)  Course ID (String)  Group Name (String) (Exception 3)  Reason (String)  Lecturer’s Note (String) |
| 3 | The actor fills in the information and clicks on the “Send" button. | The system confirms the application was sent successfully |
| Exception: | No | Cause | System Response |
| 1 | The Student Code is not correct | The system response “Student Code is not correct” |
| 2 | The Topic ID is not correct | The system response “Topic does not exist” |
| 3 | The Group Name is not correct | The system response “Group Name is not correct” |
| Alternative Flows: | N/A |  |  |
| Business Rules: | Student ID must be a standard form. (ex: SE16123)  Must choose type  Group Name has a length greater than 3 and less than 15 characters.  The group name must exist.  Topic ID must be a standard form. (ex: HRM501)  Subject ID must be a standard form. (ex: SWT301)  Course ID must be a standard form. (ex: NJS1701)  Reason has a length greater than 5 and less than 50 characters.  Process Note has a length greater than 5 and less than 50 characters. | | |
| Assumptions: | Internet connection is available. | | |

| ID and Name: | UCS-02: View List of Applications | | |
| --- | --- | --- | --- |
| Create by: | KhangNh | Date Created: | 21/06/2023 |
| Actor: | Student | | |
| Priority: | Medium | Frequency of Use: | often |
| Trigger: | Actor wants to view list of application. | | |
| Description: | This feature allows users to view list of application | | |
| Pre-conditions: | The actor must be logged into the system.  The actor must have the necessary permissions to view the list of applications.. | | |
| Post-conditions: | Success: The lecturer has reviewed the applications and performed the necessary actions.  The application status and notifications are updated accordingly. | | |
| Main Flow: | Step | Actor action | System response |
| 1 | Actor navigates to homepage | The system displays toolbars |
| 2 | Actor taps on Application toolbar | The system show table include ;  Application ID (String)  Student Code(String)  Create Date(Date)  Type(String)  Reason(String)  Process note(String)  Status(String) |
| Alternative Flows: | N/A | | |
| Exception | N/A | | |
| Business Rules: | N/A | | |
| Assumptions: | Internet connection is available. | | |

| ID and Name: | UCS-03 : View List Presentation | | |
| --- | --- | --- | --- |
| Create by: | KhangNH | Date Created: | 20/06/2023 |
| Actor: | Student | | |
| Priority: | high | Frequency of Use: | Often |
| Trigger: | The student requests to view list presentation | | |
| Description: | This use case describes the process by which a student views a presentation list. | | |
| Pre-conditions: | The student is logged into the system. | | |
| Post-conditions: | The student can view the list of presentations. | | |
| Main Flow: | Step | Actor action | System response |
| 1 | The student accesses the "Presentation" section. | The system displays a list of presentation announcements.  Each presentation notice includes:  Presentation ID (String)  Subject ID(String)  Course ID(String)  Group ID(String)  Topic Name (String)  Present Date (String)  Time (String)  View Content introduction button  Grade button  Feedback button  View Question |
| Exception | N/A | | |
| Alternative Flows: | N/A | | |
| Business Rules: | N/A | | |
| Assumptions: | Internet connection is available. | | |

| ID and Name: | UCS-04: View Semester List | | |
| --- | --- | --- | --- |
| Create by: | KhangNH | Date Created: | 26/07/2023 |
| Actor: | Student | | |
| Priority: | High | Frequency of Use: | Usually |
| Trigger: | Student logs into the system and accesses the homepage, triggering the automatic display of a list of available semesters on the homepage. | | |
| Description: | This feature allows the student to view a list semester. | | |
| Pre-conditions: | The student is logged into the system.  The institution has defined existing semesters for students to view and select. | | |
| Post-conditions: | The student is presented with a list of available semesters on the homepage for enrollment or further actions. | | |
| Main Flow: | Step | Actor action | System response |
| 1 | Student logs into the system. | The system automatically displays a list of available semesters on the homepage:  · Semester Name(String)  · StartDate(Date)  · EndDate(Date)  · Status(String) |
| Alternative Flows: | N/A | | |
| Exception | N/A | | |
| Business Rules: | N/A | | |
| Assumptions: | Internet connection is available. | | |

| ID and Name: | UCS-05: View Student list in Course | | |
| --- | --- | --- | --- |
| Create by: | KhangNH | Date Created: | 26/07/2023 |
| Actor: | Student | | |
| Priority: | high | Frequency of Use: | sometimes |
| Trigger: | Student performs the " View Student list in Course" process by clicking the List Student button on the Course Detail page. | | |
| Description: | This feature allows the student to view a list of Students of a particular course. | | |
| Pre-conditions: | Course must have Student.  Lecturer must be able to access the Course of the Subject. | | |
| Post-conditions: | Lecturer successfully displays the Student list of a course. | | |
| Main Flow: | Step | Actor action | System response |
| 1 | Lecturer accesses a specific course. | Display system:  + List Student button  + List Topic button  + List Group button |
| 2 | Lecturer clicks on the List Student button. | Display system:  Lecturer Name (String)  The system displays the student list table:  + No (int)  + Student Code (String)  + Name (String)  + Birthday (date)  + Phone Number (String)  + Gender (String)  + Address (String)  + City (String)  + Email (String) |
| Alternative Flows: | N/A | | |
| Exception | N/A | | |
| Business Rules: | N/A | | |
| Assumptions: | Internet connection is available. | | |

| ID and Name: | UCS-06: View Course list | | |
| --- | --- | --- | --- |
| Create by: | KhangNH | Date Created: | 26/07/2023 |
| Actor: | Student | | |
| Priority: | high | Frequency of Use: | sometimes |
| Trigger: | Student performs the "View Course list" process by accessing a semester. | | |
| Description: | This feature allows the student to view the class list of a particular subject. | | |
| Pre-conditions: | The Student is logged into the system.  The user has the necessary permissions or role to view the course list.  There are existing courses available in the system. | | |
| Post-conditions: | Student is presented with a list of available courses in the system, including course names, course codes, instructors, and other relevant information. | | |
| Main Flow: | Step | Actor action | System response |
| 1 | Student accesses a specific semester. | Display system:  Semester Name (String)  Start Date (date)  End Date (date)  The system displays the list of courses:  Each course includes:  + Course Code (String)  + Subject Code (String)  + Start Date (date)  + End Date (date) |
| Alternative Flows: | N/A | | |
| Exception | N/A | | |
| Business Rules: | N/A | | |
| Assumptions: | Internet connection is available. | | |

| ID and Name: | UCS-07: View Topic list in Subject | | |
| --- | --- | --- | --- |
| Create by: | KhangNH | Date Created: | 28/07/2023 |
| Actor: | Student | | |
| Priority: | high | Frequency of Use: | sometimes |
| Trigger: | Student performs the "View Topic list in Subject" process by clicking the List Topic button on the Course Detail page. | | |
| Description: | This feature allows the Student to view a list of Topics of a particular subject. | | |
| Pre-conditions: | The user is logged into the system.  The user has the necessary permissions to access and view topics related to subjects.  There are existing subjects and associated topics available in the system. | | |
| Post-conditions: | The user is presented with a list of topics related to the selected subject.  The topic list includes details such as topic names, topic codes, descriptions, and any other relevant information.  The user can further interact with the displayed topic list, such as accessing detailed information about specific topics or selecting a topic for further actions. | | |
| Main Flow: | Step | Actor action | System response |
| 1 | Student accesses a specific course. | Display system:  + List Student button  + List Topic button  + List Group button |
| 2 | Student clicks on the List Topic button. | The system displays the Topic list table:  + No (int)  + Topic Code (String)  + Topic Name (String)  + Short Description (String)  + Full Description (String)  + Lecturer Name (String)  + Start Date (date)  + Modify Date (date) |
| Alternative Flows: | N/A | | |
| Exception | N/A | | |
| Business Rules: | N/A | | |
| Assumptions: | Internet connection is available. | | |

| ID and Name: | UCS-08: View Member in Group | | |
| --- | --- | --- | --- |
| Create by: | KhangNH | Date Created: | 26/07/2023 |
| Actor: | Student | | |
| Priority: | High | Frequency of Use: | Usually |
| Trigger: | The student want to view the member in Group | | |
| Description: | This feature allows the student to view member in group. | | |
| Pre-conditions: | The student is logged into the system.  The student is a member of the group they want to view.  The group to be viewed exists in the system. | | |
| Post-conditions: | The student successfully views the list of members in the specific group they are part of.  The student can see the names and roles of all other members in the group. | | |
| Main Flow: | Step | Actor action | System response |
| 1 | Student logs into the system. | The system automatically displays a list of available semesters on the homepage:  · Semester Name(String)  · StartDate(Date)  · EndDate(Date)  · Status(String) |
|  | 2 | students click to view a semester | The system will navigate to a page to view all subjects in 1 semester include:  Course Name(String)  StartDate(Date)  EndDate(Date)  Subject Name(String)  Lecturer Name(String)  Detail Button |
|  | 3 | Students select the detail button of a subject | Display system includes:  · Course Name(String)  · Lecturer Name(String)  · StartDate(Date)  · EndDate(Date)    Subtab include :  · List Student  · List Topic  · List Group |
|  | 4 | Student taps on “List Group” | The system include:  SubjectName(String)  Lecturer Name (String)  Topic(String)  and show more table:  StudentNoRoll(int)  StudentName(String)  GroupName(String)  StartDate(Date)  Major(String)  isLeader(String)  “Switch Group” button |
| Alternative Flows: | N/A | | |
| Exception | N/A | | |
| Business Rules: | N/A | | |
| Assumptions: | Internet connection is available. | | |

| ID and Name: | UCS-10: View Project in Group | | |
| --- | --- | --- | --- |
| Create by: | KhangNH | Date Created: | 26/07/2023 |
| Actor: | Student | | |
| Priority: | High | Frequency of Use: | Usually |
| Trigger: | The student want to view the project in Group | | |
| Description: | This feature allows the student to view project in group. | | |
| Pre-conditions: | The student is logged into the system.  The student is a member of the group they want to view project.  The project to be viewed exists in the system. | | |
| Post-conditions: | The student successfully views the list of members in the specific group they are part of.  The student can see the names and roles of all other members in the group. | | |
| Main Flow: | Step | Actor action | System response |
| 1 | Student logs into the system. | The system automatically displays a list of available semesters on the homepage:  · Semester Name(String)  · StartDate(Date)  · EndDate(Date)  · Status(String) |
|  | 2 | students click to view a semester | The system will navigate to a page to view all subjects in 1 semester include:  Course Name(String)  StartDate(Date)  EndDate(Date)  Subject Name(String)  Lecturer Name(String)  Detail Button |
|  | 3 | Students select the detail button of a subject | Display system includes:  · Course Name(String)  · Lecturer Name(String)  · StartDate(Date)  · EndDate(Date)    Subtab include :  · List Student  · List Topic  · List Group |
|  | 4 | Student taps on “List Group” | The system include:   * SubjectName(String) * Lecturer Name (String) * Topic(String) * TopicID(String) * TopicName(String) * Context(String) * Actors(String) * Requirement(String) * Note(String) |
| Alternative Flows: | N/A | | |
| Exception | N/A | | |
| Business Rules: | N/A | | |
| Assumptions: | Internet connection is available. | | |

| ID and Name: | UCS-11: Switch Group | | | | |
| --- | --- | --- | --- | --- | --- |
| Create by: | KhangNH | | Date Created: | 26/07/2023 | |
| Actor: | Student | | | | |
| Priority: | High | | Frequency of Use: | Usually | |
| Trigger: | The student want to switch group | | | | |
| Description: | This feature allows the student to switch group. | | | | |
| Pre-conditions: | The student is logged into the system.  The student want to switch  The group to be viewed exists in the system. | | | | |
| Post-conditions: | The student successfully switch group they want. | | | | |
| Main Flow: | Step | | Actor action | System response | |
| 1 | | Student logs into the system. | The system automatically displays a list of available semesters on the homepage:  · Semester Name(String)  · StartDate(Date)  · EndDate(Date)  · Status(String) | |
|  | 2 | | students click to view a semester | The system will navigate to a page to view all subjects in 1 semester include:  Course Name(String)  StartDate(Date)  EndDate(Date)  Subject Name(String)  Lecturer Name(String)  Detail Button | |
|  | 3 | | Students select the detail button of a subject | Display system includes:  · Course Name(String)  · Lecturer Name(String)  · StartDate(Date)  · EndDate(Date)    Subtab include :  · List Student  · List Topic  · List Group | |
|  | 4 | | Student taps on “List Group” | The system include:  SubjectName(String)  Lecturer Name (String)  Topic(String)  and show more table:  StudentNoRoll(int)  StudentName(String)  GroupName(String)  StartDate(Date)  Major(String)  isLeader(String)  “Switch Group” button | |
|  | 5 | | Student presses on “Switch Group” | The system will show the table include:  StudentNoRoll(int)  StudentName(String)  GroupName(String)  StartDate(Date)  Major(String)  isLeader(String)  Code(String)  “View Detail” button  “Switch Group” button | |
|  | 6 | | Student fill group’s code which student want to switch |  | |
|  | 7 | | Student click button “Switch” | The system transfers students to the group chosen by the student  (exception 1) | |
| Alternative Flows: | N/A | | | | |
| Exception | No | Cause | | | System Response |
|  | 1 | Students fill in the wrong code | | | The system show message “This group code is false or does not exist” |
| Business Rules: | Each group's code is unique | | | | |
| Assumptions: | Internet connection is available. | | | | |

## NON-FUNCTIONAL REQUIREMENT

### Usebality

#### Ease of use:

* The system must have a simple and user-friendly user interface.
* Users should be able to learn and use the system quickly without extensive training.
* The usability of the user interface should be highly rated.

#### Compatibility:

* The system should be compatible with popular web browsers such as Chrome, Firefox.
* The user interface should display correctly and be compatible across different devices, including desktop computers, laptops, and Ipad.

#### Convenience and user experience:

* The system should provide user-friendly and convenient features such as search, information filtering, and data sorting.
* The user interface should have an appealing and visually pleasing design, making it easy for users to interact with and understand information.

### Reliability

#### Availability:

* The system should have a high level of availability to ensure uninterrupted access for users.
* The system downtime should be minimised, and scheduled maintenance should be communicated in advance to users.

#### Fault tolerance:

* The system should be designed to handle and recover from failures gracefully.
* It should have mechanisms in place to detect and handle errors, ensuring minimal impact on users' activities.

#### Data integrity:

* The system should ensure the integrity of data, preventing data loss or corruption.
* Adequate backup and recovery mechanisms should be implemented to safeguard data in case of system failures.

#### Performance under load:

* The system should be able to handle a significant number of concurrent users and process registration requests efficiently.
* It should be designed to scale and adapt to increased user demand without significant degradation in performance.

### Performance

#### Response time:

* The system must have fast response times to promptly meet user requests.
* The maximum response time for each action in the system should not exceed an acceptable limit.

#### Load handling:

* The system must be able to handle simultaneous registration requests from multiple users.
* The system's ability to handle high loads must be ensured to avoid overloading situations.

#### Consistency:

* The system must ensure data consistency throughout the presentation registration process.
* Data related to presentation registration must be updated synchronously and without conflicts.

#### System efficiency:

* The system must effectively utilise resources such as memory, CPU, network bandwidth, etc.
* The system's performance should meet requirements in terms of processing time and efficient resource utilisation.

### Supportability

#### Maintainability:

* The system must be easily maintainable to facilitate updates, bug fixes, and enhancements.
* Changes to the system should be easy to implement without causing disruptions or extensive downtime.

#### Scalability:

* The system must have the ability to scale up or down to accommodate changes in user demand and presentation registrations.
* It should handle increased loads without sacrificing performance or stability.

#### Technical Support:

* The system should provide comprehensive technical support to users and administrators.
* Support channels such as online help documentation, user forums, and responsive customer service should be available.