|  |  |  |
| --- | --- | --- |
| **Use Case Name** | View All Classes | |
| **Scenario** | View All Classes | |
| **Triggering Event** | Actors wanted to see all classes | |
| **Brief Description** | System shows all classes according to the actors’ respective roles, then each actor can view the classes’ details. | |
| **Actors** | Student, Lecturer, Administrative Department | |
| **Related Use Cases** | View Class Detail | |
| **Stakeholders** | Student, Lecturer, Administrative Department | |
| **Precondition** | System must validate whether the actor is Student, Lecturer, or Administrative Department | |
| **Postcondition** | Actor has viewed all classes and may proceed to see the classes’ detail. | |
| **Flow of Events** | **Actor** | **System** |
| 1. Open View All Classes Page | 1.1 Fetch all classes from database  1.2 Show all classes according to the actor’s role |
| **Exception Condition** | 1.2 If the actor is Student, then show only the classes which the student has enrolled in  1.2 If the actor is Lecturer, then show the classes that the lecturer teach  1.2 If the actor is Administrative Department, then show all classes in the database. | |

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Add forum post | |
| **Scenario** | Add forum post | |
| **Triggering Event** | Student or lecturer want to start a thread or reply to someone’s post | |
| **Brief Description** | Student or lecturer create a new post by choosing the class, fill in the form, and click the add button | |
| **Actors** | Student, Lecturer | |
| **Related Use Cases** | Might be invoked after *View Forum* use case | |
| **Stakeholders** | Student, Lecturer | |
| **Precondition** | Student and Lecturer classes must be available | |
| **Postcondition** | Forum post is submitted | |
| **Flow of Events** | **Actor** | **System** |
| 1. Student or Lecturer click Add New Post button  2. Student or Lecturer input post content  3. Student or Lecturer click submit button | 1.1 System show Add New Post form  3.1 System validate input  3.2 System insert data to database  3.3 System show success message |
| **Exception Condition** | 3.1 If input is not valid, then show error message and prompt user to reinput | |

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Delete Forum Post | |
| **Scenario** | Delete a forum post that Actor has created | |
| **Triggering Event** | Actor wanted to delete a forum post that they have created | |
| **Brief Description** | Student or lecturer click delete button on the post that they want to delete and then confirm the deletion so that the post is deleted completely | |
| **Actors** | Student, Lecturer | |
| **Related Use Cases** | Might be invoked after *View Forum* use case. Actor must have posted previously, so *Add Forum Post* use case is related as well | |
| **Stakeholders** | Student, Lecturer | |
| **Precondition** | Student’s or Lecturer’s forum post must be available | |
| **Postcondition** | Forum post is deleted | |
| **Flow of Events** | **Actor** | **System** |
| 1. Student or Lecturer open Forum detail page  2. Student or Lecturer click the delete button  3. Student or Lecturer confirm the deletion | 1.1 System show forum detail page  2.1 System show confirmation dialog  3.1 System delete the post from database  3.2 System show success message |
| **Exception Condition** | 2. Student or Lecturer didn’t confirm the deletion, so this process is cancelled. | |

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Submit Score | |
| **Scenario** | Submit Student Mid / Final exam score | |
| **Triggering Event** | After every exam, lecturer must mark their student exam and submit the score | |
| **Brief Description** | Lecturer can submit the score by inputting each of their student score | |
| **Actors** | Lecturer | |
| **Related Use Cases** | View Score | |
| **Stakeholders** | Lecturer, Student, Scoring Department | |
| **Precondition** | Lecturer has marked student’s exam | |
| **Postcondition** | Score is submitted | |
| **Flow of Events** | **Actor** | **System** |
| 1. Lecturer open Score Submission page  2. Lecturer input student score  3. Lecturer click submit button  4. Lecturer confirm submission | 1.1 System show Score Submission page  3.1 System show confirmation dialog  4.1 System update student score to database  4.2 System show success message |
| **Exception Condition** | 2. While there is more student score, keep repeat this step  4. If lecturer didn’t confirm the submission, then this process is cancelled. | |

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Update Course | |
| **Scenario** | Update Course | |
| **Triggering Event** | Academic Department wanted to revise the course syllabus of a specific course or its credits per semester. | |
| **Brief Description** | Academic Department update the course by clicking update button on the course that they want to update and input the data. | |
| **Actors** | Academic Department | |
| **Related Use Cases** | View All Course and View Course Detail | |
| **Stakeholders** | Academic Department, Student, Lecturer | |
| **Precondition** | Course must be available | |
| **Postcondition** | Course is updated | |
| **Flow of Events** | **Actor** | **System** |
| 1. Actor click update button on the selected course  2. Actor input course syllabus and course credit per semester  3. Actor click submit button | 1.1 System show update page  3.1 System validate input  3.2 System update data to database  3.3 System show success message |
| **Exception Condition** | 3.1 Inputted field is not valid, then system will show error message | |

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Create Assignment | |
| **Scenario** | Create Assignment | |
| **Triggering Event** | Lecturer wanted to create a new individual or group assignment for the student to do | |
| **Brief Description** | Lecturer will input all assignment details and submit it | |
| **Actors** | Lecturer | |
| **Related Use Cases** | Submit Assignment | |
| **Stakeholders** | Lecturer, Student | |
| **Precondition** | Class and students must be available | |
| **Postcondition** | Assignment is inserted to database | |
| **Flow of Events** | **Actor** | **System** |
| 1. Lecturer choose individual or group assignment  2. Lecturer input assignment title  3. Lecturer input assignment case  4. Lecturer click submit button | 4.1 System validate input  4.2 System insert assignment to database  4.3 System show success message |
| **Exception Condition** | 1. If lecturer choose group assignment, validate whether lecturer has created class group or not. If not, show error message and the process will be stopped. | |