

|  |  |
| --- | --- |
| **Use case name:** Add Student | |
| **Participating actors(s):** Admin | |
| **Preconditions:** Before the use case starts, the admin must log in the system. | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enters student’s details.  3. Actor confirms the details and save. | 2. Validates if data have been typed in the correct format and allow actor to confirm the details.  4. Displays a confirming message that student has been added. |
| **Postconditions:** Student has been added successfully. | |
| **Special Requirements:** None | |

|  |  |
| --- | --- |
| **Use case name:** UpdateStudent | |
| **Participating actors(s):** Admin | |
| **Preconditions:** Before the use case starts, the admin must log in the system and then search for the student before proceeding with the update | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enters the student ID and name.  3. Updates the required data and save. | 2. Displays the student details.  4. Presents a message confirming update. |
| **Postconditions:** Student has been updated successfully | |
| **Special Requirements:** None | |

|  |  |
| --- | --- |
| **Use case name:** DeleteStudent | |
| **Participating actors(s):** Admin | |
| **Preconditions:** Before the use case starts, the admin must log in the system and then search for the student before proceeding with the deletion | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enter student ID and name.  3. Selects delete. | 2. Retrieves the data, display student’s details and allow actor to delete.  4. Displays a confirming delete verification dialog. |
| **Postconditions:** Student has been deleted successfully | |
| **Special Requirements:** None | |

|  |  |
| --- | --- |
| **Use case name:** UploadStudent | |
| **Participating actors(s):** Admin | |
| **Preconditions:** Before the use case starts, the admin must log in the system | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
|  |  |
| **Postconditions:** Student has been uploaded successfully | |
| **Special Requirements:** None | |

|  |  |
| --- | --- |
| **Use case name:** Search Student | |
| **Participating actors(s):** Admin, Lecturer | |
| **Preconditions:** Before the use case starts, admin and lecturer must log in the system. | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enter student ID / name  3. The use case ends when the actor finds the student | 2. Retrieves and displays the data if it exists. |
| **Postconditions:**  The actors successfully find the respective student they are looking for. | |
| **Special Requirements:** Actors can either search student by name or ID at a time | |

Diagram

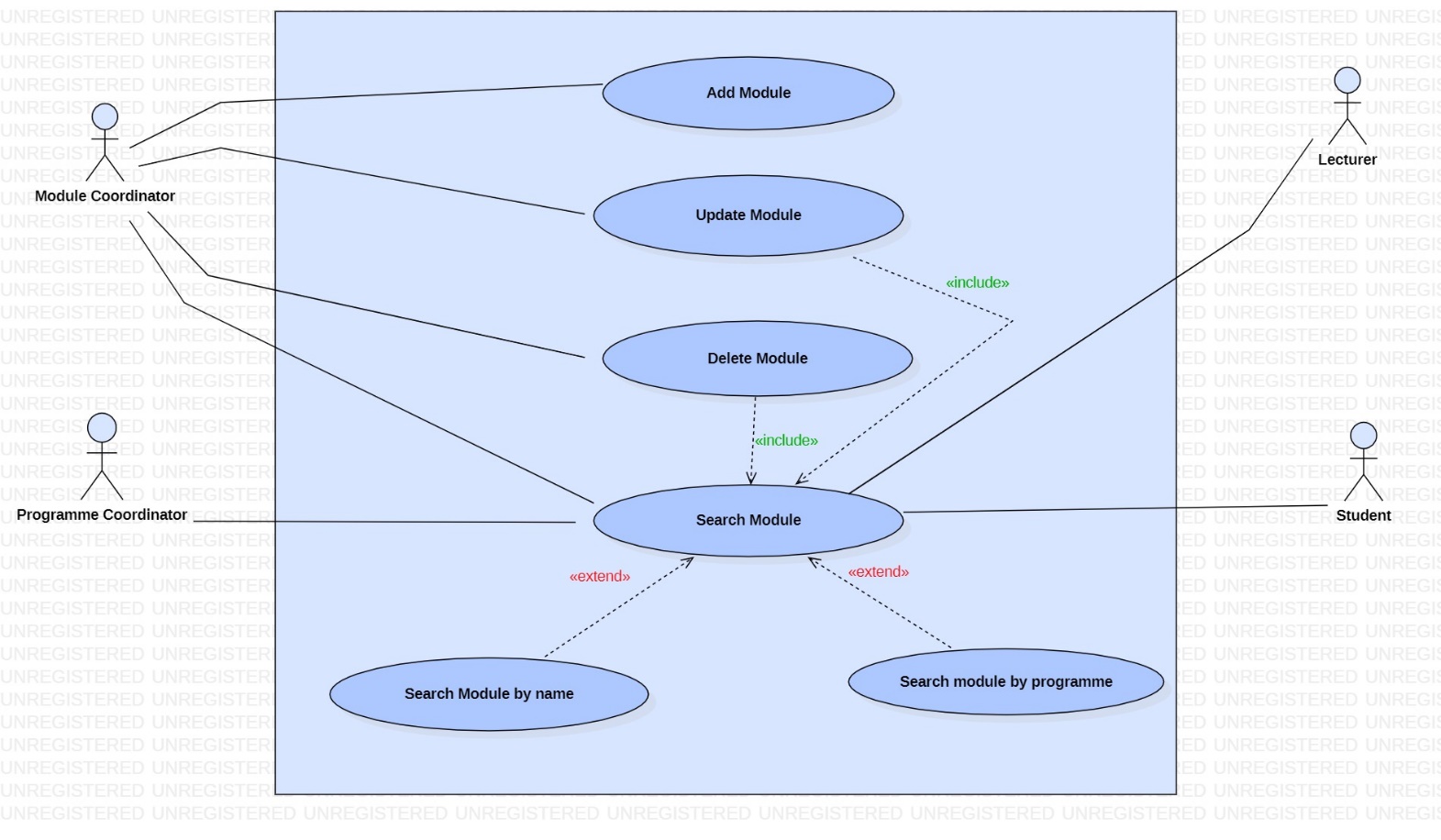
Description automatically generated

|  |  |
| --- | --- |
| **Use case name:** Add programme | |
| **Participating actors(s):** Programme Coordinator | |
| **Preconditions:** Before the use case starts, actor must log in the system | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enter the programme’s details  3. Actor confirms the details and save | 2. Validates if data have been typed in the correct format and allow actor to confirm the details.  4. Displays a confirming message that programme has been added. |
| **Postconditions:** Programmehas been added successfully | |
| **Special Requirements:** None | |

|  |  |
| --- | --- |
| **Use case name:** Update programme | |
| **Participating actors(s):** Programme Coordinator | |
| **Preconditions:** Before the use case starts, actor must log in the system and then search for the programme before proceeding with the update | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enter the programme’s details  3. Allow the actor to update the required data and save. | 2. Retrieves and displays the programme’s details  4. Displays a confirming message that programme has been updated. |
| **Postconditions:** Programmehas been updated successfully | |
| **Special Requirements:** None | |

|  |  |
| --- | --- |
| **Use case name:** Delete Programme | |
| **Participating actors(s):** Programme Coordinator | |
| **Preconditions:** Before the use case starts, the actor must log in the system and then actor must search for the programme before proceeding with the deletion | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enter programme’s details.  3. Selects delete. | 2. Retrieves the data, display programme’s details and allow actor to delete.  4. Displays a confirming delete verification dialog. |
| **Postconditions:** Programme has been deleted successfully | |
| **Special Requirements:** None | |

|  |  |
| --- | --- |
| **Use case name:** Search Programme | |
| **Participating actors(s):** Programme coordinator, lecturer, student | |
| **Preconditions:** Before the use case starts, actors must log in the system. | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enter programme’s ID/ name/module  3. The use case ends when the actors find the programme | 2. Retrieves and displays the data if it exists. |
| **Postconditions:**  The actors successfully find the programme they are looking for. | |
| **Special Requirements:** Actors can eithersearch programme by name/ID/module at a time | |



|  |  |
| --- | --- |
| **Use case name:** Add Module | |
| **Participating actors(s):** Module Coordinator | |
| **Preconditions:** Before the use case starts, actor must log in the system | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enter the module’s details  3. Actor confirms the details and save | 2. Validates if data have been typed in the correct format and allow actor to confirm the details.  4. Displays a confirming message that module has been added. |
| **Postconditions:** Programmehas been added successfully | |
| **Special Requirements:** None | |

|  |  |
| --- | --- |
| **Use case name:** Update module | |
| **Participating actors(s):** Module Coordinator | |
| **Preconditions:** Before the use case starts, actor must log in the system and then actor must search for the module first before proceeding with the update | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enter the module’s details  3. Allow the actor to update the required data and save. | 2. Retrieves and displays the module’s details  4. Displays a confirming message that module has been updated. |
| **Postconditions:** Programmehas been updated successfully | |
| **Special Requirements:** None | |

|  |  |
| --- | --- |
| **Use case name:** Delete Module | |
| **Participating actors(s):** Module Coordinator | |
| **Preconditions:** Before the use case starts, the actor must log in the system and then search for the module before proceeding with the deletion | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enter module’s details.  3. Selects delete. | 2. Retrieves the data, display module’s details and allow actor to delete.  4. Displays a confirming delete verification dialog. |
| **Postconditions:** Module has been deleted successfully | |
| **Special Requirements:** none | |

|  |  |
| --- | --- |
| **Use case name:** Search Module | |
| **Participating actors(s):** Module Coordinator, Programme Coordinator, Lecturer, Students | |
| **Preconditions:** Before the use case starts, actors must log in the system. | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enter module’s name/programme  3. The use case ends when the actors find the module | 2. Retrieves and displays the data if it exists. |
| **Postconditions:**  The actors successfully find the programme they are looking for. | |
| **Special Requirements:** Actors can eithersearch the module by its name or programme at a time. | |

Diagram

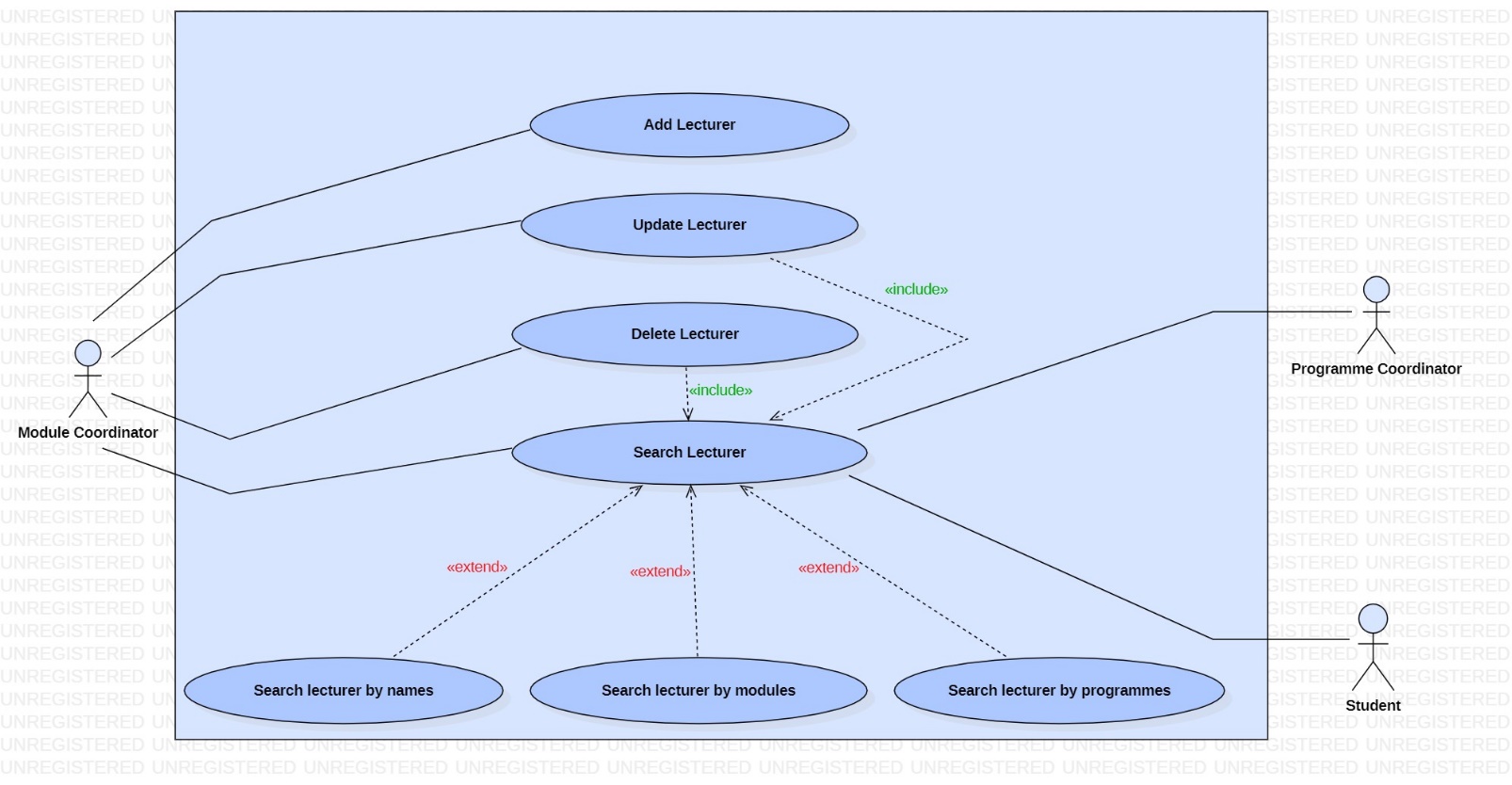
Description automatically generated

|  |  |
| --- | --- |
| **Use case name:** Add Cohort | |
| **Participating actors(s):** Programme Coordinator | |
| **Preconditions:** Before the use case starts, actor must log in the system | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enter the cohort’s details  3. Actor confirms the details and save | 2. Validates if data have been typed in the correct format and allow actor to confirm the details.  4. Displays a confirming message that module has been added. |
| **Postconditions:** Cohorthas been added successfully | |
| **Special Requirements:** None | |

|  |  |
| --- | --- |
| **Use case name:** Update Cohort | |
| **Participating actors(s):** Programme Coordinator | |
| **Preconditions:** Before the use case starts, actor must log in the system and must search for the cohort first and then proceed the update. | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enter the cohort’s details  3. Allow the actor to update the required data and save. | 2. Retrieves and displays the cohort’s details  4. Displays a confirming message that cohort has been updated. |
| **Postconditions:** Cohort has been updated successfully | |
| **Special Requirements:** None | |

|  |  |
| --- | --- |
| **Use case name:** Delete Cohort | |
| **Participating actors(s):** Programme Coordinator | |
| **Preconditions:** Before the use case starts, the actor must log in the system and then search for the module before proceeding with the deletion | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enter cohort’s details.  3. Selects delete. | 2. Retrieves the data, display cohort’s details and allow actor to delete.  4. Displays a confirming delete verification dialog. |
| **Postconditions:** Cohort has been deleted successfully | |
| **Special Requirements:** none | |

|  |  |
| --- | --- |
| **Use case name:** Search Cohort | |
| **Participating actors(s):** Programme Coordinator | |
| **Preconditions:** Before the use case starts, actors must log in the system. | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enter cohort’s name/ID/ Students’ name  3. The use case ends when the actors find the cohort | 2. Retrieves and displays the data if it exists. |
| **Postconditions:**  The actors successfully find the cohort they are looking for. | |
| **Special Requirements:** Actors can eithersearch the cohort by its name, ID or students’ names at a time. | |



|  |  |
| --- | --- |
| **Use case name:** Add Lecturer | |
| **Participating actors(s):** Module Coordinator | |
| **Preconditions:** Before the use case starts, actor must log in the system | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enter the lecturer’s details  3. Actor confirms the details and save | 2. Validates if data have been typed in the correct format and allow actor to confirm the details.  4. Displays a confirming message that lecturer has been added. |
| **Postconditions:** Lecturerhas been added successfully | |
| **Special Requirements:** None | |

|  |  |
| --- | --- |
| **Use case name:** Update Lecturer | |
| **Participating actors(s):** Module Coordinator | |
| **Preconditions:** Before the use case starts, actor must log in the system and must search for the lecturer first and then proceed with the update. | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enter the lecturer’s details  3. Allow the actor to update the required data and save. | 2. Retrieves and displays the cohort’s details  4. Displays a confirming message that lecturer has been updated. |
| **Postconditions:** Lecturer has been updated successfully | |
| **Special Requirements:** None | |

|  |  |
| --- | --- |
| **Use case name:** Delete Lecturer | |
| **Participating actors(s):** Module Coordinator | |
| **Preconditions:** Before the use case starts, the actor must log in the system and then search for the lecturer before proceeding with the deletion | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enter lecturer’s details.  3. Selects delete. | 2. Retrieves the data, display lecturer’s details and allow actor to delete.  4. Displays a confirming delete verification dialog. |
| **Postconditions:** Lecturer has been deleted successfully | |
| **Special Requirements:** none | |

|  |  |
| --- | --- |
| **Use case name:** Search Lecturer | |
| **Participating actors(s):** Module Coordinator, Programme Coordinator, Student | |
| **Preconditions:** Before the use case starts, actors must log in the system. | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enter lecturer’s name/ module/ programmes  3. The use case ends when the actors find the lecturer | 2. Retrieves and displays the data if it exists. |
| **Postconditions:**  The actors successfully find the lecturer they are looking for. | |
| **Special Requirements:** Actors can eithersearch the lecturer by his name/module or programme at a time. | |

Chart, diagram

Description automatically generated

|  |  |
| --- | --- |
| **Use case name:** Add Coursework | |
| **Participating actors(s):** Module Coordinator, lecturer | |
| **Preconditions:** Before the use case starts, actors must log in the system | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enter the coursework’s details  3. Actors confirm the details and save | 2. Validates if data have been typed in the correct format and allows actor to confirm the details.  4. Displays a confirming message that coursework has been added. |
| **Postconditions:** Courseworkhas been added successfully | |
| **Special Requirements:** None | |

|  |  |
| --- | --- |
| **Use case name:** Update Coursework | |
| **Participating actors(s):** Module Coordinator, Lecturer | |
| **Preconditions:** Before the use case starts, actors must log in the system and must search for the coursework first and then proceed with the update. | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enter the coursework’s details  3. Allow the actors to update the required data and save. | 2. Retrieves and displays the coursework’s details  4. Displays a confirming message that lecturer has been updated. |
| **Postconditions:** Coursework has been updated successfully | |
| **Special Requirements:** None | |

|  |  |
| --- | --- |
| **Use case name:** Delete Coursework | |
| **Participating actors(s):** Module Coordinator, Lecturer | |
| **Preconditions:** Before the use case starts, the actors must log in the system and then search for the coursework before proceeding with the deletion | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enter coursework’s details.  3. Selects delete. | 2. Retrieves the data, display coursework’s details and allow actor to delete.  4. Displays a confirming delete verification dialog. |
| **Postconditions:** Coursework has been deleted successfully | |
| **Special Requirements:** none | |

|  |  |
| --- | --- |
| **Use case name:** Search Coursework | |
| **Participating actors(s):** Module Coordinator, Lecturer | |
| **Preconditions:** Before the use case starts, actors must log in the system. | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enter coursework’s details  3. The use case ends when the actors find the coursework | 2. Retrieves and displays the data if it exists. |
| **Postconditions:**  The actors successfully find the coursework they are looking for. | |
| **Special Requirements:** Actors can eithersearch coursework by student/description/ lecturer/date at a time. | |

|  |  |
| --- | --- |
| **Use case name:** Add Re-sit Coursework | |
| **Participating actors(s):** Module Coordinator, lecturer | |
| **Preconditions:** Before the use case starts, actors must log in the system | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enter the re-sit coursework’s details  3. Actors confirm the details and save | 2. Validates if data have been typed in the correct format and allows actor to confirm the details.  4. Displays a confirming message that re-sit coursework has been added. |
| **Postconditions:** Resitcourseworkhas been added successfully | |
| **Special Requirements:** None | |

|  |  |
| --- | --- |
| **Use case name:** Update re-sit coursework | |
| **Participating actors(s):** Module Coordinator, Lecturer | |
| **Preconditions:** Before the use case starts, the actor must log in the system and then search for the re-sit before proceeding with the update. | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enter the re-sit coursework’s details  3. Allow the actors to update the required data and save. | 2. Retrieves and displays the re-sit coursework’s details  4. Displays a confirming message that the re-sit coursework has been updated. |
| **Postconditions:** Resit coursework has been updated successfully | |
| **Special Requirements:** None | |

|  |  |
| --- | --- |
| **Use case name:** Delete Resit coursework | |
| **Participating actors(s):** Module Coordinator, Lecturer | |
| **Preconditions:** Before the use case starts, the actor must log in the system and then search for the re-sit coursework before proceeding with the deletion | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enter re-sit coursework’s details.  3. Selects delete. | 2. Retrieves the data, display the re-sit coursework’s details, and allow actor to delete.  4. Displays a confirming delete verification dialog. |
| **Postconditions:** Resit coursework has been deleted successfully | |
| **Special Requirements:** none | |

|  |  |
| --- | --- |
| **Use case name:** Search re-sit coursework | |
| **Participating actors(s):** Module Coordinator, Lecturer | |
| **Preconditions:** Before the use case starts, actors must log in the system. | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enter coursework’s details  3. The use case ends when the actors find the re-sit coursework | 2. Retrieves and displays the data if it exists. |
| **Postconditions:**  The actors successfully find the coursework they are looking for. | |
| **Special Requirements:** Actors can eithersearch re-sit coursework by date/description/student/lecturer at a time. | |

Diagram

Description automatically generated

|  |  |
| --- | --- |
| **Use case name:** Add Exam | |
| **Participating actors(s):** Faculties | |
| **Preconditions:** Before the use case starts, actors must log in the system | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enter the exam’s details  3. Actors confirm the details and save | 2. Validates if data have been typed in the correct format and allows actor to confirm the details.  4. Displays a confirming message that exam has been added. |
| **Postconditions:** Examhas been added successfully | |
| **Special Requirements:** None | |

|  |  |
| --- | --- |
| **Use case name:** Update Exam | |
| **Participating actors(s):** Faculties | |
| **Preconditions:** Before the use case starts, actors must log in the system and must search for the exam first and then proceed with the update. | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enter the exam’s details  3. Allow the actors to update the required data and save. | 2. Retrieves and displays the exam’s details  4. Displays a confirming message that lecturer has been updated. |
| **Postconditions:** Exam has been updated successfully | |
| **Special Requirements:** None | |

|  |  |
| --- | --- |
| **Use case name:** Delete Exam | |
| **Participating actors(s):** Faculties | |
| **Preconditions:** Before the use case starts, the actors must log in the system and then search for the exam before proceeding with the deletion | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enter exam’s details.  3. Selects delete. | 2. Retrieves the data, exam’s details and allow actor to delete.  4. Displays a confirming delete verification dialog. |
| **Postconditions:** Exam has been deleted successfully | |
| **Special Requirements:** none | |

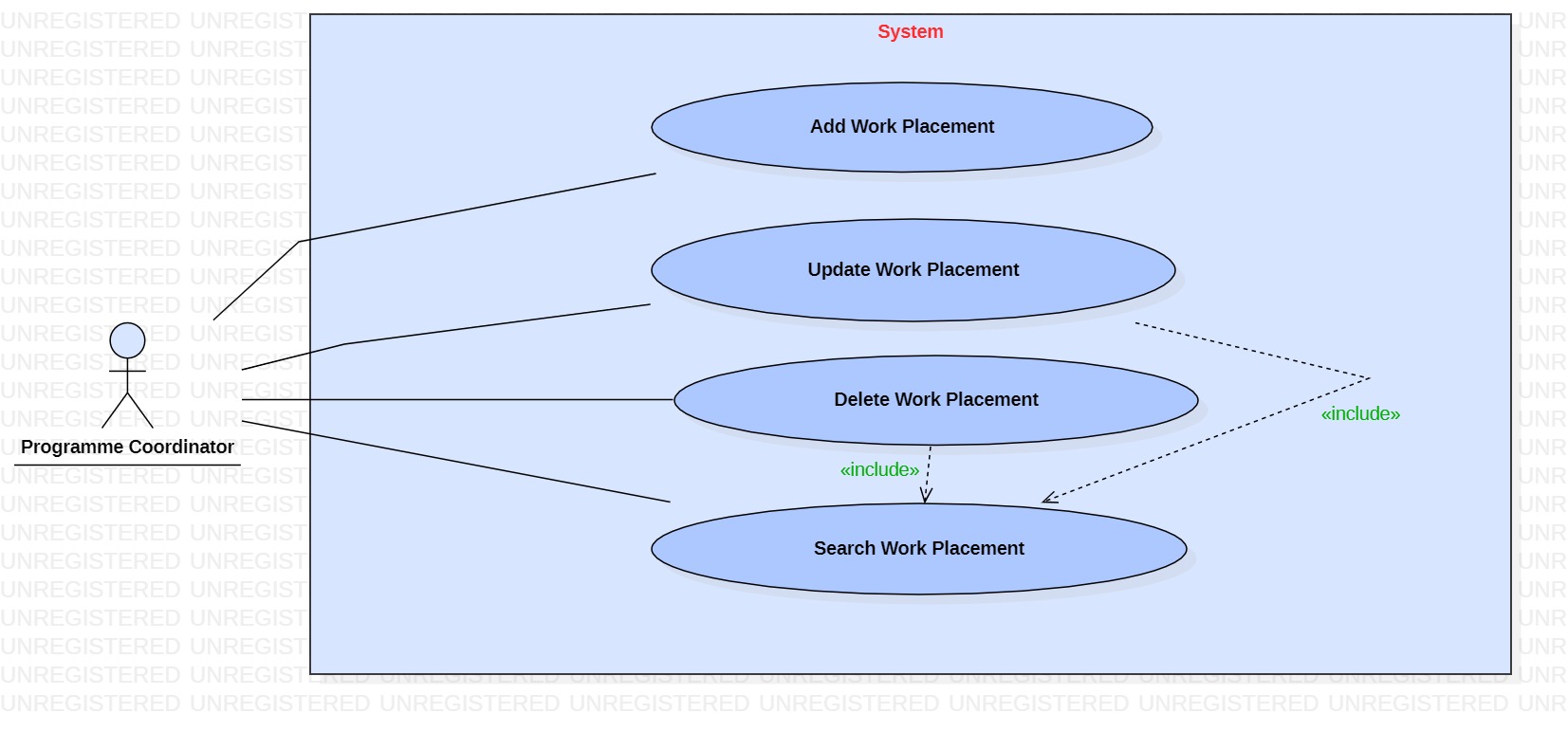
|  |  |
| --- | --- |
| **Use case name:** Search Exam | |
| **Participating actors(s):** Module Coordinator, Student | |
| **Preconditions:** Before the use case starts, actors must log in the system. | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enter exam’s details  3. The use case ends when the actors find the exam | 2. Retrieves and displays the data if it exists. |
| **Postconditions:**  The actors successfully find the exam they are looking for. | |
| **Special Requirements:** Actors can either  Search Exam by date, module, student, programme, cohort at a time | |

|  |  |
| --- | --- |
| **Use case name:** Add Re-sit Exam | |
| **Participating actors(s):** Module Coordinator, Programme Coordinator | |
| **Preconditions:** Before the use case starts, actors must log in the system | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enter the re-sit exam’s details  3. Actors confirm the details and save | 2. Validates if data have been typed in the correct format and allows actor to confirm the details.  4. Displays a confirming message that re-sit exam has been added. |
| **Postconditions:** Re-sitexam has been added successfully | |
| **Special Requirements:** None | |

|  |  |
| --- | --- |
| **Use case name:** Update re-sit Exam | |
| **Participating actors(s):** Module Coordinator, Programme Coordinator | |
| **Preconditions:** Before the use case starts, actors must log in the system and must search for the re-sit exam first and then proceed with the update. | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enter the re-sit exam’s details  3. Allow the actors to update the required data and save. | 2. Retrieves and displays the re-sit exam’s details  4. Displays a confirming message that the re-sit exam has been updated. |
| **Postconditions:** Re-sit exam has been updated successfully | |
| **Special Requirements:** None | |

|  |  |
| --- | --- |
| **Use case name:** Delete Resit Exam | |
| **Participating actors(s):** Module Coordinator, Programme Coordinator | |
| **Preconditions:** Before the use case starts, the actor must log in the system and then search for the re-sit exam before proceeding with the deletion | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enter re-sit exam s details.  3. Selects delete. | 2. Retrieves the data, display the re-sit exam’s details, and allow actor to delete.  4. Displays a confirming delete verification dialog. |
| **Postconditions:** Resit exam has been deleted successfully | |
| **Special Requirements:** none | |

|  |  |
| --- | --- |
| **Use case name:** Search re-sit Exam | |
| **Participating actors(s):** Module Coordinator, Programme Coordinator, Student | |
| **Preconditions:** Before the use case starts, actors must log in the system. | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enter exam’s details  3. The use case ends when the actors find the re-sit exam | 2. Retrieves and displays the data if it exists. |
| **Postconditions:**  The actors successfully find the exam they are looking for. | |
| **Special Requirements:** Actors can eithersearch re-sit exam by date, module, student, programme, cohort at a time. | |



|  |  |
| --- | --- |
| **Use case name:** Add Work placement | |
| **Participating actors(s):** Programme Coordinator | |
| **Preconditions:** Before the use case starts, actor must log in the system | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enter the work placement’s details  3. Actor confirms the details and save | 2. Validates if data have been typed in the correct format and allow actor to confirm the details.  4. Displays a confirming message that work placement has been added. |
| **Postconditions:** Work placementhas been added successfully | |
| **Special Requirements:** None | |

|  |  |
| --- | --- |
| **Use case name:** Update Work placement | |
| **Participating actors(s):** Programme Coordinator | |
| **Preconditions:** Before the use case starts, actor must log in the system and must search for the work placement first and then proceed with the update. | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enter the work placement’s details  3. Allow the actor to update the required data and save. | 2. Retrieves and displays the cohort’s details  4. Displays a confirming message that work placement has been updated. |
| **Postconditions:** Work placement has been updated successfully | |
| **Special Requirements:** None | |

|  |  |
| --- | --- |
| **Use case name:** Delete Work placement | |
| **Participating actors(s):** Programme Coordinator | |
| **Preconditions:** Before the use case starts, the actor must log in the system and then search for the work placement before proceeding with the deletion | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enter work placement’s details.  3. Selects delete. | 2. Retrieves the data, display work placement’s details and allow actor to delete.  4. Displays a confirming delete verification dialog. |
| **Postconditions:** Work placement has been deleted successfully | |
| **Special Requirements:** none | |

|  |  |
| --- | --- |
| **Use case name:** Search Work placement | |
| **Participating actors(s):** Programme Coordinator | |
| **Preconditions:** Before the use case starts, actors must log in the system. | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enter work placement  3. The use case ends when the actors find the work placement | 2. Retrieves and displays the data if it exists. |
| **Postconditions:**  The actors successfully find the work placement they are looking for. | |
| **Special Requirements:** None | |

A close-up of a document

Description automatically generated with low confidence

|  |  |
| --- | --- |
| **Use case name:** Add Project | |
| **Participating actors(s):** Module Coordinator, Lecturer | |
| **Preconditions:** Before the use case starts, actor must log in the system | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enter the project’s details  3. Actor confirms the details and save | 2. Validates if data have been typed in the correct format and allow actor to confirm the details.  4. Displays a confirming message that project has been added. |
| **Postconditions:** Project has been added successfully | |
| **Special Requirements:** None | |

|  |  |
| --- | --- |
| **Use case name:** Update Project | |
| **Participating actors(s):** Module Coordinator, Lecturer | |
| **Preconditions:** Before the use case starts, actor must log in the system and must search for the project first and then proceed with the update. | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enter the project’s details  3. Allow the actor to update the required data and save. | 2. Retrieves and displays the project’s details  4. Displays a confirming message that project has been updated. |
| **Postconditions:** Project has been updated successfully | |
| **Special Requirements:** None | |

|  |  |
| --- | --- |
| **Use case name:** Delete Project | |
| **Participating actors(s):** Module Coordinator, Lecturer | |
| **Preconditions:** Before the use case starts, the actor must log in the system and then search for the project before proceeding with the deletion | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enter project’s details.  3. Selects delete. | 2. Retrieves the data, display project s details and allow actor to delete.  4. Displays a confirming delete verification dialog. |
| **Postconditions:** Project has been deleted successfully | |
| **Special Requirements:** none | |

|  |  |
| --- | --- |
| **Use case name:** Search Project | |
| **Participating actors(s):** Module Coordinator, Lecturer | |
| **Preconditions:** Before the use case starts, actors must log in the system. | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enter project’s details  3. The use case ends when the actors find the project | 2. Retrieves and displays the data if it exists. |
| **Postconditions:**  The actors successfully find the project they are looking for. | |
| **Special Requirements:** None | |

|  |  |
| --- | --- |
| **Use case name:** Post coursework | |
| **Participating actors(s):** Module Coordinator, Lecturer | |
| **Preconditions:** Before the use case starts, actors must log in the system. | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enter coursework’s details  3. Actors confirm the details and save | 2. Dialog pops up to confirm the posting of the coursework  4. Displays a confirming message that coursework has been posted. |
| **Postconditions:**  The actors successfully post the coursework | |
| **Special Requirements:** None | |

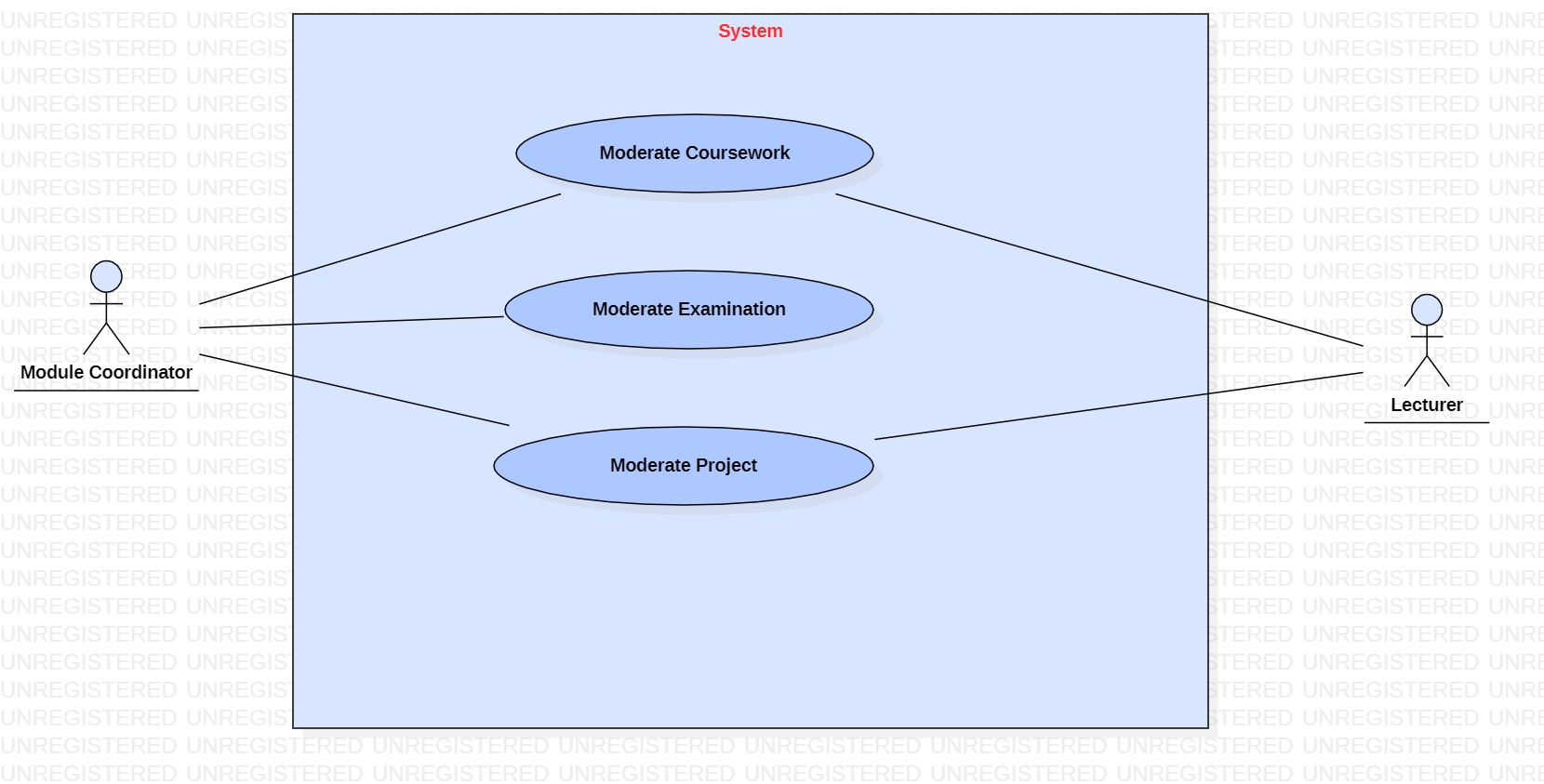
|  |  |
| --- | --- |
| **Use case name:** Post exam | |
| **Participating actors(s):** Module Coordinator, Lecturer | |
| **Preconditions:** Before the use case starts, actors must log in the system. | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Input exam’s details  3. Actors confirm the details and save. | 2. Pop of a dialog box to confirm the posting of the exam  4. Displays a confirming message that exam has been posted. |
| **Postconditions:**  The actors successfully post the exam | |
| **Special Requirements:** None | |

|  |  |
| --- | --- |
| **Use case name:** Post project | |
| **Participating actors(s):** Module Coordinator, Lecturer | |
| **Preconditions:** Before the use case starts, actors must log in the system. | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Input project’s details  3. Actors confirm the details and save | 2. Pop up of the dialog box to confirm the posting of the project  4. Displays a confirming message that project has been posted. |
| **Postconditions:**  The actors successfully post the project | |
| **Special Requirements:** None | |

|  |  |
| --- | --- |
| **Use case name:** Un-post coursework | |
| **Participating actors(s):** Module Coordinator, Lecturer | |
| **Preconditions:** Before the use case starts, actors must log in the system. | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enter coursework details  3. Actor confirm the details | 2. Retrieves and displays the data if it exists and shows a dialog box to confirm the un-posting of the coursework  4. Displays a message that coursework has been un-posted. |
| **Postconditions:**  The actors successfully un-post the coursework | |
| **Special Requirements:** None | |

|  |  |
| --- | --- |
| **Use case name:** Un-post exam | |
| **Participating actors(s):** Module Coordinator | |
| **Preconditions:** Before the use case starts, actors must log in the system. | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enter exam’s details  3. Actor confirm the details | 2. Retrieves and displays the data if it exists and shows a dialog box to confirm the un-posting of the exam  4. Displays a message that exam has been un-posted. |
| **Postconditions:**  The actors successfully un-post the exam | |
| **Special Requirements:** None | |

|  |  |
| --- | --- |
| **Use case name:** Un-post project | |
| **Participating actors(s):** Module Coordinator, Lecturer | |
| **Preconditions:** Before the use case starts, actors must log in the system. | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enter project’s details  3. Actor confirm the details | 2. Retrieves and displays the data if it exists and shows a dialog box to confirm the un-posting of the project  4. Displays a message that project has been un-posted. |
| **Postconditions:**  The actors successfully un-post the project | |
| **Special Requirements:** None | |



|  |  |
| --- | --- |
| **Use case name:** Moderate Coursework | |
| **Participating actors(s):** Module Coordinator, Lecturer | |
| **Preconditions:** Before the use case starts, actor must log in the system | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enter details to moderate coursework  3. Actor confirms the details and save | 2. Validates if data have been typed in the correct format and allow actor to confirm the details.  4. Displays a confirming message that coursework has been moderated. |
| **Postconditions:** Courseworkhas been moderated successfully | |
| **Special Requirements:** Retrieve coursework details to moderate | |

|  |  |
| --- | --- |
| **Use case name:** Moderate Examination | |
| **Participating actors(s):** Module Coordinator | |
| **Preconditions:** Before the use case starts, actor must log in the system | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enter details to moderate examination  3. Actor confirms the details and save | 2. Validates if data have been typed in the correct format and allow actor to confirm the details.  4. Displays a confirming message that examination has been moderated. |
| **Postconditions:** Examination has been moderated successfully | |
| **Special Requirements:** Retrieve exam details to moderate | |

|  |  |
| --- | --- |
| **Use case name:** Moderate Project | |
| **Participating actors(s):** Module Coordinator, Lecturer | |
| **Preconditions:** Before the use case starts, actor must log in the system | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enter details to moderate project  3. Actor confirms the details and save | 2. Validates if data have been typed in the correct format and allow actor to confirm the details.  4. Displays a confirming message that project has been moderated. |
| **Postconditions:** Project has been moderated successfully | |
| **Special Requirements:** Retrieve project details to moderate | |

Diagram

Description automatically generated

|  |  |
| --- | --- |
| **Use case name:** Generate Board of Examiners report | |
| **Participating actors(s):** Programme Coordinator | |
| **Preconditions:** Before the use case starts, actor must log in the system | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enter the details of Board of Examiners report  3. Actor confirms the details and save | 2. Validates if data have been typed in the correct format and allow actor to confirm the details.  4. Displays a confirming message that report of Board of Examiners has been generated |
| **Postconditions:** Board of Examiners has been generated successfully | |
| **Special Requirements:** None | |

|  |  |
| --- | --- |
| **Use case name:** Update Exam after BOE | |
| **Participating actors(s):** Module Coordinator | |
| **Preconditions:** Before the use case starts, actor must log in the system and must search for the exam first and then proceed with the update. | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enter the exam’s details as per BOE  3. Allow the actor to update the required data and save. | 2. Retrieves and displays the exam’s details as per BOE  4. Displays a confirming message that exam has been updated. |
| **Postconditions:** Exam after BOE has been updated successfully | |
| **Special Requirements:** None | |

|  |  |
| --- | --- |
| **Use case name:** Update Project after BOE | |
| **Participating actors(s):** Module Coordinator | |
| **Preconditions:** Before the use case starts, actor must log in the system and must search for the project first and then proceed with the update. | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enter the project’s details as per BOE  3. Allow the actor to update the required data and save. | 2. Retrieves and displays the project’s details as per BOE  4. Displays a confirming message that project has been updated. |
| **Postconditions:** Project after BOE has been updated successfully | |
| **Special Requirements:** None | |

|  |  |
| --- | --- |
| **Use case name:** Update Coursework after BOE | |
| **Participating actors(s):** Module Coordinator | |
| **Preconditions:** Before the use case starts, actor must log in the system and must search for the exam first and then proceed with the update. | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enter the coursework’s details as per BOE  3. Allow the actor to update the required data and save. | 2. Retrieves and displays the coursework’s details as per BOE  4. Displays a confirming message that coursework has been updated. |
| **Postconditions:** Coursework after BOE has been updated successfully | |
| **Special Requirements:** None | |

Diagram

Description automatically generated

|  |  |
| --- | --- |
| **Use case name:** Generate results | |
| **Participating actors(s):** Faculties | |
| **Preconditions:** Before the use case starts, actor must log in the system | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enter the results of students  3. Actor confirms the details and save | 2. Validates if data have been typed in the correct format and allow actor to confirm the details.  4. Displays a confirming message that results has been generated |
| **Postconditions:** Results has been generated successfully | |
| **Special Requirements:** None | |

|  |  |
| --- | --- |
| **Use case name:** Check results by students | |
| **Participating actors(s):** Faculties | |
| **Preconditions:** Before the use case starts, actor must log in the system | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enter student’s details  3. The use case ends when the actors find the results of students. | 2. Retrieves and displays the data if it exists. |
| **Postconditions:** The actors successfully find the results they are looking for. | |
| **Special Requirements:** Actors can search the results by student’s details at a time | |

|  |  |
| --- | --- |
| **Use case name:** Generate Transcript | |
| **Participating actors(s):** Faculties | |
| **Preconditions:** Before the use case starts, actor must log in the system | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enter the transcript’s details  3. Actor confirms the details and save | 2. Validates if data have been typed in the correct format and allow actor to confirm the details.  4. Displays a confirming message that Transcript has been generated |
| **Postconditions:** Transcript has been generated successfully | |
| **Special Requirements:** None | |

Diagram

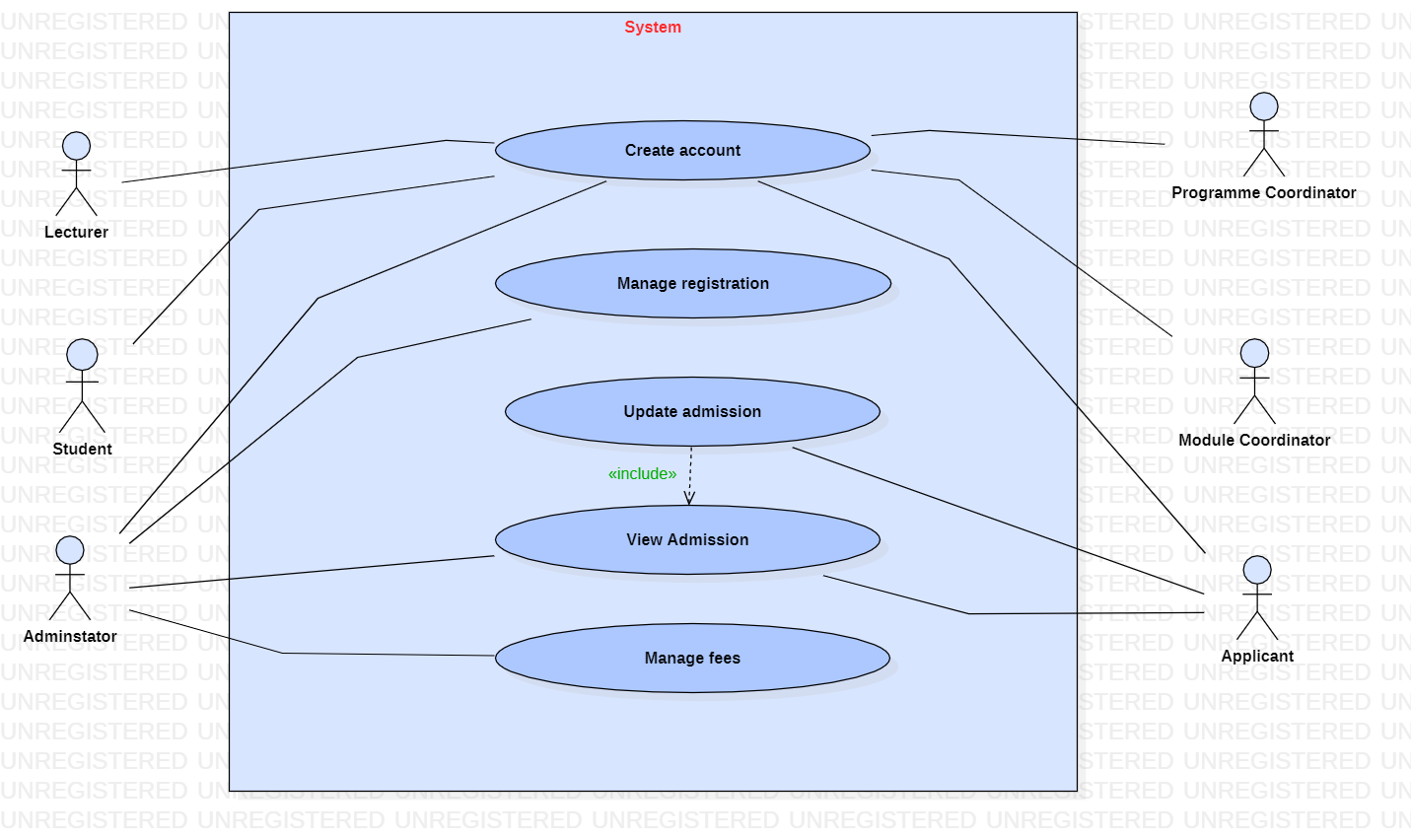
Description automatically generated

|  |  |
| --- | --- |
| **Use case name:** Add attendance | |
| **Participating actors(s):** Lecturer | |
| **Preconditions:** Before the use case starts, actor must log in the system | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enter the attendance’s details  3. Actor confirms the details and save | 2. Validates if data have been typed in the correct format and allow actor to confirm the details.  4. Displays a confirming message that attendance has been added. |
| **Postconditions:** Attendance has been added successfully | |
| **Special Requirements:** None | |

|  |  |
| --- | --- |
| **Use case name:** Update attendance | |
| **Participating actors(s):**  Lecturer | |
| **Preconditions:** Before the use case starts, actor must log in the system and then search for the attendance before proceeding with the update | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enter the attendance’s details  3. Allow the actor to update the required data and save. | 2. Retrieves and displays the attendance’s details  4. Displays a confirming message that attendance has been updated. |
| **Postconditions:** Attendance has been updated successfully | |
| **Special Requirements:** None | |

|  |  |
| --- | --- |
| **Use case name:** Delete attendance | |
| **Participating actors(s):** Lecturer | |
| **Preconditions:** Before the use case starts, the actor must log in the system and then actor must search for the attendance before proceeding with the deletion | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enter attendance’s details.  3. Selects delete. | 2. Retrieves the data, display attendance’s details and allow actor to delete.  4. Displays a confirming delete verification dialog. |
| **Postconditions:** Attendance has been deleted successfully | |
| **Special Requirements:** None | |

|  |  |
| --- | --- |
| **Use case name:** Search attendance | |
| **Participating actors(s):** Modulecoordinator, lecturer | |
| **Preconditions:** Before the use case starts, actors must log in the system. | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enter students’ ID  3. The use case ends when the actors find the attendance. | 2. Retrieves and displays the data if it exists. |
| **Postconditions:**  The actors successfully find the attendance they are looking for. | |
| **Special Requirements:** Actors cansearch attendance by students’ ID at a time | |



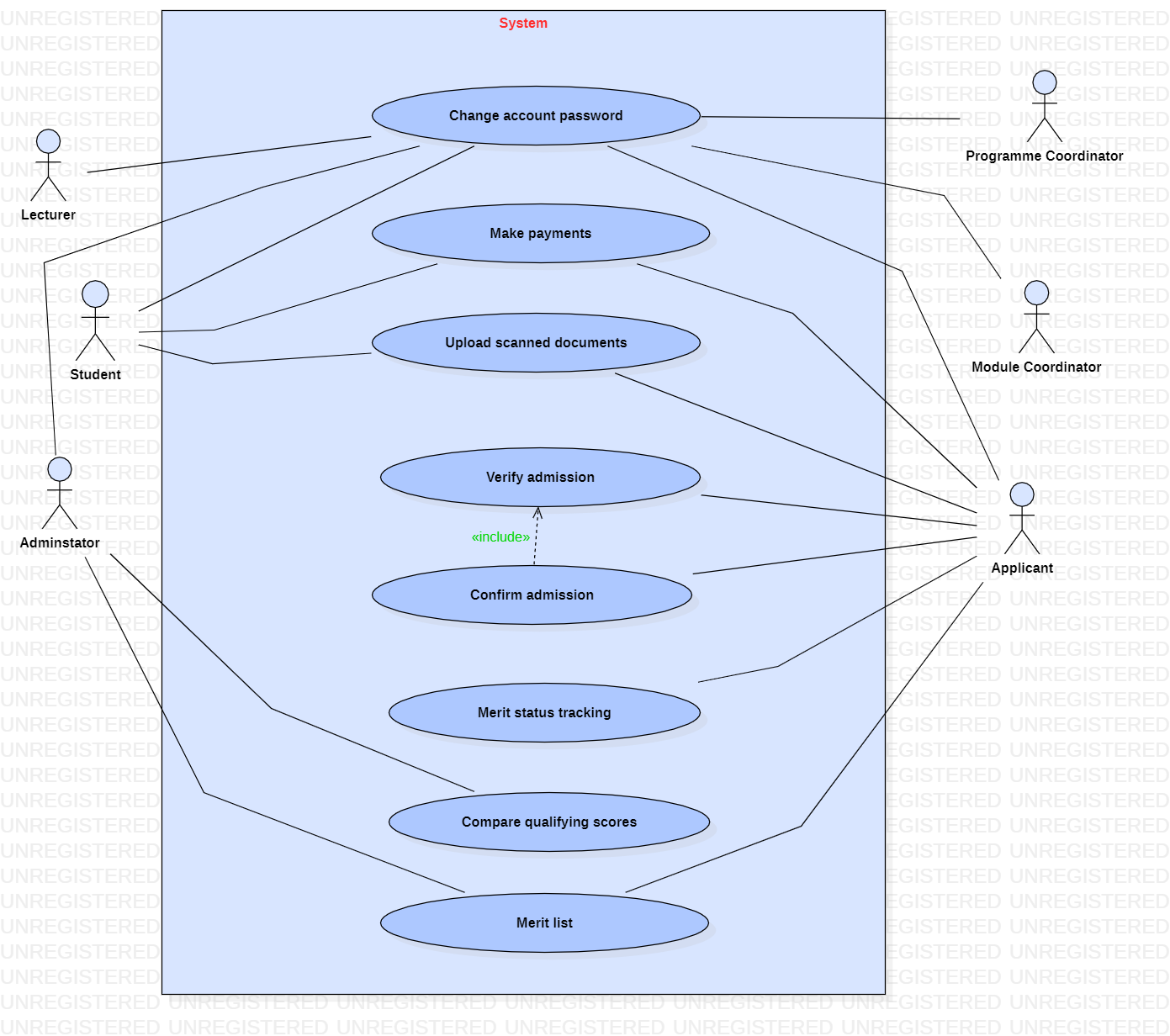
|  |  |
| --- | --- |
| **Use case name:** Create Account | |
| **Participating actors(s):** Lecturer, Student, Administrator, Programme Coordinator, Module Coordinator, Applicant | |
| **Preconditions:** None | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enter the details for creation of the account  3. Actor confirms the details and save | 2. Validates if data have been typed in the correct format and allow actor to confirm the details.  4. Displays a confirming message that account has been created |
| **Postconditions:** Account has been created successfully | |
| **Special Requirements:** None | |

|  |  |
| --- | --- |
| **Use case name:** Manage registration | |
| **Participating actors(s):** Administrator | |
| **Preconditions:** Before the use case starts, actor must log in the system | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enter registration’s details  3. The use case ends when registration has been managed | 2. Retrieves and displays the registration details |
| **Postconditions:** Registration has been managed successfully. | |
| **Special Requirements:** None | |

|  |  |
| --- | --- |
| **Use case name:** Update admission | |
| **Participating actors(s):** Applicant | |
| **Preconditions:** Before the use case starts, actors must log in the system and must view the admission first before proceeding with the update. | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enter admission’s details  3. Allow the actor to update the required data and save. | 2. Retrieves and displays the data if it exists.  4. Displays a confirming message that the admission has been updated |
| **Postconditions:**  The admission has been updated successfully | |
| **Special Requirements:** None | |

|  |  |
| --- | --- |
| **Use case name:** View admission | |
| **Participating actors(s):** Applicant, Administrator | |
| **Preconditions:** Before the use case starts, actors must log in the system and must view the admission first before proceeding with the update. | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enter admission’s details  3. The use case ends when the actor has viewed the admission | 2. Retrieves and displays the data if it exists. |
| **Postconditions:**  None | |
| **Special Requirements:** None | |

|  |  |
| --- | --- |
| **Use case name:** Manage fees | |
| **Participating actors(s):** Administrator | |
| **Preconditions:** Before the use case starts, actors must log in the system | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1.Enter admission’s details  3. Actor confirms and save | 2.System shows if fees have been received and allow actor to confirm |
| **Postconditions:**  Fees has been managed successfully | |
| **Special Requirements:** None | |



|  |  |
| --- | --- |
| **Use case name:** Change account password | |
| **Participating actors(s):** Lecturer, Student, Administrator, Programme Coordinator, Module Coordinator, Applicant | |
| **Preconditions:** Actors must log in the system first | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1.Sign in and enter account’s details (with current account password)  3. Allow the actor to change the account password and save | 2. Retrieves and display data  4. Displays a confirming message that account’s password has been changed |
| **Postconditions:** Password of the account has been changed successfully | |
| **Special Requirements:** None | |
| **Use case name:** Make payments | |
| **Participating actors(s):** Students, Applicants | |
| **Preconditions:** Before the use case starts, actors must log in the system | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1.Enter payment’s details  3. Actor select the mode of payment and confirm the transaction | 2.Display the mode of payment  4. Shows a confirming message that payment has been done |
| **Postconditions:**  Payment of fees has been successfully made | |
| **Special Requirements:** None | |

|  |  |
| --- | --- |
| **Use case name:** Upload scanned documents | |
| **Participating actors(s):** Applicants, students | |
| **Preconditions:** Actors must log in the system first | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1.Select the required scanned document and upload it.  3. Actor confirms and save. | 2. A pop of the dialog box to confirm the uploading of the documents  3. displays a confirming message that documents have been uploaded |
| **Postconditions:** Documents have been uploaded successfully | |
| **Special Requirements:** None | |

|  |  |
| --- | --- |
| **Use case name:** Verify admission | |
| **Participating actors(s):** Applicant | |
| **Preconditions:** Before the use case starts, actors must log in the system | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enter admission’s details  3. The use case ends once actor has verified the admission | 2. Retrieves and displays the data if it exists. |
| **Postconditions:**  None | |
| **Special Requirements:** None | |

|  |  |
| --- | --- |
| **Use case name:** Confirm admission | |
| **Participating actors(s):** Applicant | |
| **Preconditions:** Before the use case starts, actors must log in the system and must verify the admission first before proceeding with the confirmation | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1. Enter admission’s details  3. Actor verifies and the confirm the admission | 2. Retrieves and displays the data if it exists and allow actor to confirm  4. Displays a message that admission has been confirmed |
| **Postconditions:**  The admission has been confirmed successfully | |
| **Special Requirements:** None | |

|  |  |
| --- | --- |
| **Use case name:** Merit status tracking | |
| **Participating actors(s):** Applicants | |
| **Preconditions:** Actor must log in the system first | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1.Enter tracking details  3.Use case ends once actor viewed the status of the tracking | 2. Displays the status of the tracking merit |
| **Postconditions:** None | |
| **Special Requirements:** None | |

|  |  |
| --- | --- |
| **Use case name:** Compare qualifying scores | |
| **Participating actors(s):** Administrator | |
| **Preconditions:** Actor must log in the system first | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1.Enter the admission details  3.Actor then compares the qualifying scores | 2. retrieves and display the admission results. |
| **Postconditions:** None | |
| **Special Requirements:** None | |

|  |  |
| --- | --- |
| **Use case name:** Merit List | |
| **Participating actors(s):** Administrator, Applicant | |
| **Preconditions:** Actors must log in the system first | |
| **Flow of events** | |
| **Actor Action** | **System Response** |
| 1.Enter exam/ admission details.  3.Actors can view the merit list with the high achiever student / applicant at the top. | 2. retrieves and display the admission results. |
| **Postconditions:** None | |
| **Special Requirements:** None | |