**Name: Group 2**

**Team members**

**Adam Zacharias**

**Awet Goitom**

**Muna Teklemariam**

**Yonas Abrha**

**Sirak Ghebrehiwot**

***MIU-Schedule***

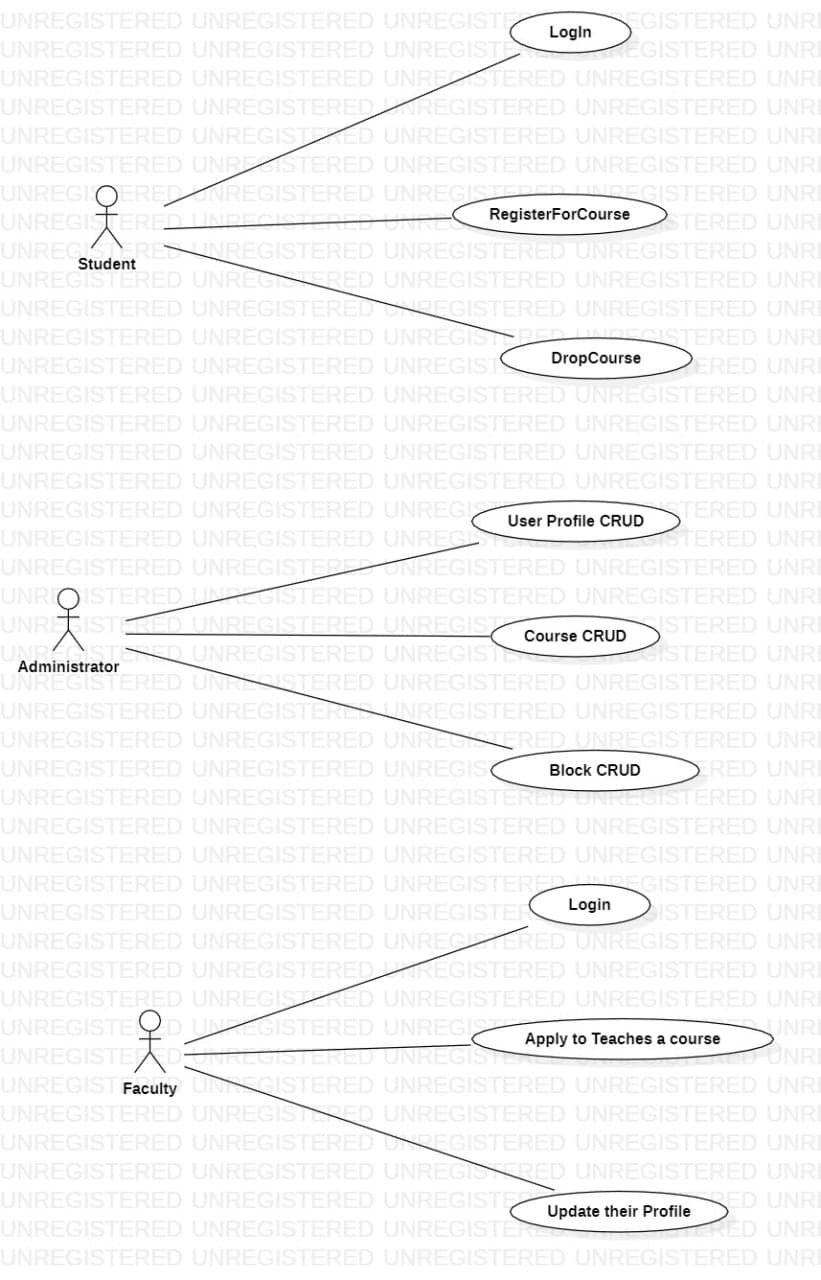
**Diagram

Description automatically generated**

**Use Case Description**

|  |  |  |
| --- | --- | --- |
| **Use Case Number: 1** | | |
| **Name course registration** | | |
| **Brief description** This use case allows the student to register courses | | |
| **Actors student** | | |
|  | | |
| **Preconditions** | | |
| The student must logged in to the system | | |
| **Flows of Events:** | | |
| **1. Basic Flows** | | |
| **1.1 Register course** | | |
| **Step** | **User Actions** | **System Actions** |
| 1 | The student calls the register course command | The system displays list of courses with their respective faculty member assigned to it, the block where the course is in and other attributes related to the course |
| 2 | The Admin fills out the form and requests the system to save the details  The students choose courses and requests the system to save the details | The system saves the changes made and it is still open for other updates the student can make until the course registration deadlines are met. |
| **Postconditions** | | |
| The student course list is persisted in the system | | |
|  | | |
| **Business Rules** | | |
| Course change request are not allow after deadlines are met. | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Number: 2** | | |
| **Name:** Course Request | | |
| **Brief description** This use case provides functions to change or drop a course that are already registered | | |
| **Actors:** Student | | |
|  | | |
| **Preconditions** | | |
| The student must log in to the system | | |
| **Flows of Events:** | | |
| **1. Basic Flows** | | |
| **1.0 Course Drop** | | |
| **Step** | **User Actions** | **System Actions** |
| 1 | The student calls the drop course command. | The system displays the course drop request form with the fields of first name, last name, email, course name, and the block which the course is offered and other related attributes. |
| 2 | The student fills out the form and submit the request. | The system verifies that all the required information is filled out in the form in order the student to successfully submit the drop course request form. |
| **Postconditions** | | |
| The Course drop request is persisted in the system | | |
| **Business Rule** | | |
| Every required fields in the form has to be filled. | | |
|  | | |
| **1.1 Course change** | | |
| **Step** | **User Actions** | **System Actions** |
| 1 | The student calls the change course command. | The system displays the course change request form with the fields of first name, last name, email, course name to be dropped and course name to replace, and the block which the course is offered and other related attributes. |
| 2 | The student fills out the form and submit the request. | The system verifies that all the required information is filled out in the form in order the student to successfully submit the drop course request form. |
| **Postconditions** | | |
| The course change request is persisted in the system | | |
| **Business Rule** | | |
| Every required fields in the form has to be filled. | | |
|  | | |
|  | | |



|  |  |  |
| --- | --- | --- |
| **Use Case Number: 1** | | |
| **Name** User Profile CRUD | | |
| **Brief description** This use case allows the admin to create profiles for faculty and student | | |
| **Actors** Admin | | |
|  | | |
| **Preconditions** | | |
| The admin must logged in to the system | | |
| **Flows of Events:** | | |
| **1. Basic Flows** | | |
| **1.1.0 Create Faculty Profile** | | |
| **Step** | **User Actions** | **System Actions** |
| 1 | The admin calls the create faculty profile command | The system displays the faculty profile form with the fields for firstname, lastname, email, password, specializations, courses and blocks. |
| 2 | The Admin fills out the form and requests the system to save the details | The system verifies that there’s no other profile in the database with the same email address and saves the faculty and returns the success message on success or a fail message in case of failure. In case another profile exists with the email address, the system returns the message indicating a duplicate entry exists. |
| **Postconditions** | | |
| The staff profile is persisted in the system | | |
|  | | |
| **Business Rules** | | |
| No duplicate faculty profiles. A unique profile is identified by email address. | | |
|  | | |
| **1.1.1 Read/view Faculty Profile** | | |
| **Step** | **User Actions** | **System Actions** |

|  |  |  |
| --- | --- | --- |
| 1 | The admin selects to view a list of faculty profiles | The system returns a list of all faculty profiles. |
| 2 | The admin selects to view a profile of one of the faculty from the list of profiles | The system returns the profile of the faculty as a string with the names (firstname and lastname), email, specializations, courses and the blocks of preference. |
|  | | |
| **1.1.2 Update Faculty Profile** | | |
| **Step** | **User Actions** | **System Action** |
| 1 | The admin selects to view a list of faculty profiles | The system returns a list of all faculty profiles. |
| 2 | The admin selects the profile of the faculty they want to update | The system displays an editable faculty profile form pre-populated with the faculty profile details |
| 3 | The admin updates the fields they want to update and requests system to save the new details | The system updates the record and returns the success message or a fail message on exception. |
| **Postconditions** | | |
| The staff profile will be updated | | |
| **Business Rule** | | |
| The email field should be unwritable. | | |
|  | | |
| **1.1.3 Delete Faculty Profile** | | |
| **Step** | **User Actions** | **System Actions** |
| 1 | The admin selects to view a list of faculty profiles | The system returns a list of all faculty profiles. |
| 2 | The admin selects to delete a faculty profile from a list of faculty profiles | The system displays a confirmation dialogue window |
| 3 | The admin selects OK on the confirmation dialog window to confirm deleting the profile | The system confirms the faculty is not assigned to any sections or blocks and deletes the profile. The system returns message a success message on success or a failure message. The system should return a message indicating the profile could not be deleted because it is already assigned to some blocks and sections |
| **Postconditions** | | |
| The staff profile will be deleted | | |
| **Business Rule** | | |
| Only a faculty profile that is not assigned to any blocks or sections. | | |
| **1.2.0 Create Student Profile** | | |

|  |  |  |
| --- | --- | --- |
| **Step** | **User Action** | **System Action** |
| 1 | The admin calls the create student profile command | The system displays the student profile form with firstname, lastname, registration number, email and initial password |
| 2 | The Admin fills out the form and requests the system to save the details | The system checks if no other student profile has the same registration number and email and saves the student, and displays a success message. The system returns a fail message on exception or a duplicate entry message in case of a duplicate. |
| **Postconditions** | | |
| The student profile is persisted in the system | | |
|  | | |
| **Business Rules** | | |
| No duplicate student profiles. A unique profile is identified by email address and registration number. | | |
|  | | |
| **1.2.1 Read/view Student Profile** | | |
| **Step** | **User Actions** | **System Actions** |
| 1 | The admin selects to view a list of student profiles | The system returns a list of all student profiles. |
| 2 | The admin selects the profile they want to view. | The system returns the profile of the student as a string with names (firstname, lastname), registration number and email. |
| **1.2.2 Update Student Profile** | | |
| **Step** | **User Actions** | **System Action** |
| 1 | The admin selects to view a list of student profiles | The system returns a list of all student profiles. |
| 2 | The admin selects the profile of the student they want to update | The system displays back an editable form with the student profile details pre-filled. |
| 3 | The admin updates the parts they want to update and requests system to save the new details | The system updates the record and returns a success message or a fail message on exception. |
| **Postconditions** | | |
| The student profile will be updated | | |
| **Business Rule** | | |
| The registration number and email fields should be unwritable | | |
|  | | |
| **1.2.3 Delete Student Profile** | | |

|  |  |  |
| --- | --- | --- |
| **Step** | **User Actions** | **System Actions** |
| 1 | The admin selects to view a list of student profiles | The system returns a list of all student profiles. |
| 2 | The admin selects to delete a student profile they want | The system displays a confirmation dialogue window |
| 3 | The admin selects OK on the confirmation dialog window to confirm deleting the profile | The system checks to confirm the student is not assigned to any sections and deletes the profile and returns a success message or a fail message on failure. It must also return a specific message when the student is assigned to some sections. |
| **Postconditions** | | |
| The student profile will be deleted | | |
| **Business Rule** | | |
| The student must have not been assigned to any of the sections. | | |

**1, Name: Teaches course**

Faculty teaches a course for students registered in that section.

**2, Actor**

Faculty

**3, Precondition**

Faculty must have to login before he/she use the system.

**4, Basic Flow**

|  |  |
| --- | --- |
| **User Action** | **System Response** |
| Faculty applies to teach a course, | The system checks if there is open spot for the applied course and send the request to administrator to be approved. |
| Faculty update profile | The system receives the update and store in database. |
| Faculty can teach course | The system assigns the section to faculty and also the system provide the faculty with the student list. |