|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case Number: 1** | | | |  |
| **Name** | |  | Update Profile |  |
|  |  | |  |  |
| **Brief description** | | | This use case allows the faculty to update profiles for faculty and student | |
|  |  | |  |  |
| **Actors** | | | Faculty |  |
|  |  | |  |  |
|  |  |  |  |  |
| **Preconditions** | | |  |  |
| The faculty must logged in to the system | | | |  |
|  |  | |  |  |
| **Flows of Events:** | | |  |  |
| **1. Basic Flows** | | |  |  |
|  |  | | |  |
| **1.1.0 Update Faculty Profile** | | | |  |
|  |  |  |  |  |
| **Step** |  |  | **User Actions** | **System Actions** |
|  |  |  | |  |
| 1 |  | The faculty calls the update faculty profile command | | The system displays an editable faculty profile  form pre-populated with the faculty profile details |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  | |  |
| 2 |  | The admin updates the fields they want to update | | The system updates the record and returns the  success message or a fail message on exception. |
|  |  | and requests system to save the new details | |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  | |  |  |
| **Postconditions** | | |  |  |
| The faculty profile is persisted in the system | | | |  |



**Business Rules**

Faculty profiles should be filled properly. A unique profile is identified by email address.

**1.1.1 Read/view Faculty Profile**

|  |  |  |
| --- | --- | --- |
| **Step** | **User Actions** | **System Actions** |
|  |  |  |

|  |  |  |
| --- | --- | --- |
|  |  |  |
| 1 | The faculty logs in to view a their view | 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. |
|  |  |  |
|  |  |  |
| **Postconditions** | |  |
| The faculty profile will be updated | |  |
|  |  |  |
| **Business Rule** | |  |
| The email field should be unwritable. | |  |
|  |  |  |
|  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Use Case Number: 2** | | | |  | |
| **Name** | | | Course Schedule |  | |
|  | |  | |  | |
| **Brief description** This use case provides a functionality for viewing courses | | | | | |
|  | |  |  |  | |
| **Actors** | | | Faculty |  | |
|  | |  | |  | |
|  | |  |  |  | |
| **Preconditions** | | | |  | |
| The faculty must logged in to the system | | | |  | |
|  | |  | |  | |
| **Flows of Events:** | | | |  | |
| **1. Basic Flows** | | | |  | |
|  | |  | |  | |
| **1.1 Read/view Course** | | | |  | |
|  | |  |  |  | |
| **Step** | |  | **User Actions** | **System Actions** | |
|  | |  |  |  | |
| 1 | |  | The faculty requests for a list of courses | The system returns a list of all courses. | |
|  | |  |  |  | |
| 2 | |  | The faculty selects to view a course they want from the list | The system returns the course object as a string | |
|  | |  |  | with course code, and course name | |
|  | |  | |  | |
| **Postconditions** | | | | |  | |
| The course details will be viewed | | | | |  | |
|  |  | | | |  | |
| **Business Rule** | | | | |  | |
| The course code should be unwritable. | | | | |  | |
|  |  | | | |  | |
|  |  | | | |  | |