### Detailed Design

Detailed Design of Edu Unity Desk is given below.

### Use case Diagram

Use case diagram of Edu Unity Desk.



Fig 4.3. 1: Use Case Diagram

### Use Case Fully Dressed Format

Use case fully dressed format is given below.

#### Add Teacher

Table 4.3.2.1 | UC-1 Add Teacher

|  |  |
| --- | --- |
| Use case ID: | UC-001 |
| Use case: | Add Teacher |
| Actors: | Admin, Teacher |
| Preconditions: | Admin is an authorized user with login access to the system. |
| Basic Flow | |
| System Response:   1. System displays login page. 2. System shows dashboard. 3. System displays Teacher Module options. 4. System displays the Add Teacher form. 5. System validates the information entered. 6. System confirms the successful addition of the new teacher by showing a success message. | Actor Actions:   1. Admin logs into the system. 2. Admin navigates to the Teacher Module section. 3. Admin clicks on Add New Teacher. 4. Admin enters the following teacher details: 5. Admin submits the form. |
| Alternative Flow: | * 1. If the Admin enters invalid data (e.g., incorrect email format or duplicate email):      1. System prompts Admin to correct the errors before submission. |
| Post Condition: | The new teacher’s information is successfully stored in the system, making it accessible for further actions. |

#### Update Teacher

Table 4.3.2.2 | UC-2 Update Teacher

|  |  |
| --- | --- |
| Use case ID: | UC-002 |
| Use case: | Update Teacher |
| Actors: | Admin, Teacher |
| Preconditions: | Admin is an authorized user with login access to the system. |
| Basic Flow | |
| System Response:   1. System displays the main dashboard. 2. System displays options within the Teacher Module. 3. System displays the Add Teacher" form with fields to enter teacher details. 4. System validates the entered data and saves the teacher's information if all data is valid. 5. System displays a success message to confirm the addition of the new teacher. | Actor Actions:   1. Admin logs into the system. 2. Admin navigates to the Teacher Module section. 3. Admin clicks on Add New Teacher. 4. Admin submits the form. |
| Alternative Flow: | * 1. If the Admin enters invalid data (e.g., incorrect email format or duplicate email):      1. System prompts Admin to correct errors before submission. |
| Post Condition: | The new teacher’s information is successfully saved in the system. |

#### Add Student

Table 4.3.2.3 | UC-3 Add Student

|  |  |
| --- | --- |
| Use case ID: | UC-003 |
| Use case: | Add Student |
| Actors: | Admin, Student |
| Preconditions: | The Admin and Teacher are authorized users with access to the system.  The Teacher record to be updated exists in the system. |
| Basic Flow | |
| System Response:   1. System displays the main dashboard. 2. System shows options within the Student Module. 3. System displays a form to add a new student. 4. System provides fields for entering student details. 5. System validates the entered information. 6. System saves the new student's information if all data is valid. 7. System displays a success message confirming the student was added. | Actor Actions:   1. Admin logs into the system. 2. Admin navigates to the Student Module. 3. Admin clicks on Add New Student. 4. Admin enters the following details: 5. Admin submits the form. |
| Alternative Flow: | * 1. If the Admin enters an invalid email format:      1. System prompts the Admin to correct the email format before submission. |
| Post Condition: | The new student’s information is successfully saved in the system and can access relevant modules, interact with  teachers, and engage with the coordination module. |

#### Update Student

Table 4.3.2.4 | UC-4 Update Student

|  |  |
| --- | --- |
| Use case ID: | UC-003 |
| Use case: | Update Student |
| Actors: | Admin, Student |
| Preconditions: | The Admin and Student are authorized users with access to the system.  The student record to be updated exists in the system. |
| Basic Flow | |
| System Response:   1. System displays the main dashboard. 2. System displays options within the Student Module. 3. System shows the student's profile page. 4. System allows updates in some fields 5. System validates the updated information 6. System saves the updated information if all data is valid. 7. System displays a success message confirming the student’s profile has been updated. | Actor Actions:   1. Admin logs into the system. 2. Admin navigates to the Student Module section. 3. Admin go to the student’s profile. 4. Admin clicks on Update" on the student’s profile. 5. Admin updates necessary fields 6. Admin submits the form. |
| Alternative Flow: | * 1. If the Admin enters invalid data (e.g., incorrect email format):      1. System prompts the Admin to correct errors before submission. |
| Post Condition: | The updated student information is successfully saved in the system.  The updated details are now available for future reference and interactions within the system. |

#### Communication with Coordination

Table 4.3.2.5 | UC-5 Communication with Coordination

|  |  |
| --- | --- |
| Use case ID: | UC-005 |
| Use case: | Communication with Coordination |
| Actors: | Admin, Students |
| Preconditions: | Student is an authorized user with login access to the system. |
| Basic Flow | |
| System Response:   1. System displays the main dashboard. 2. System shows options for available modules. 3. System displays options within the module. 4. System automatically generates a voucher number for tracking the query. 5. System displays a form for composing a message. 6. System saves the message and notifies the coordination team about the new query. 7. System displays a success message confirming that   the query has been sent. | Actor Actions:   1. Student logs into the system. 2. Student navigates to the Communication with Coordination module. 3. Student clicks on New Message. 4. Student compose the message. 5. Student reviews and clicks on submit query. |
| Post Condition: | The query is successfully logged in the system.  The coordination team is notified and can respond to the student's message. |

#### Add Course

Table 4.3.2.6 | UC-6 Add Course

|  |  |
| --- | --- |
| Use case ID: | UC-006 |
| Use case: | Add Course |
| Actors: | Admin |
| Preconditions: | Admin or is an authorized user.  The course to be updated exists in the system. |
| Basic Flow | |
| System Response:   1. System displays available modules. 2. System shows options within the Course module. 3. System prompts the user to enter course details. 4. System validates the course information. 5. System saves the course details and displays a success message confirming the course   was added. | Actor Actions:   1. Admin navigates to the Course module. 2. Admin selects the Add Course" option. 3. Admin begins entering details as prompted. 4. Admin confirms the addition of the course. |
| Alternative Flow: | * 1. If required information is missing:      1. System prompts the user to complete all fields before proceeding. |
| Post Condition: | The new course is successfully added to the platform. Course details are now visible to students, teachers, and coordinators.  Authorized users can manage and interact with the course information. |

#### Update Course

Table 4.3.2.7 | UC-7 Update Course

|  |  |
| --- | --- |
| Use case ID: | UC-007 |
| Use case: | Update Course |
| Actors: | Admin |
| Preconditions: | Admin is an authorized user.  The course to be updated exists in the system. |
| Basic Flow | |
| System Response:   1. System displays available modules. 2. System shows options within the Course Management module. 3. System displays a list of existing courses. 4. System displays the current course details. 5. System provides editable fields for course details. 6. System validates the updated course information. 7. System saves the updated information and displays a success message confirming that the course details have been updated. | Actor Actions:   1. Admin navigates to the Course module. 2. Admin selects the Update Course option. 3. Admin searches for and selects the course to be updated. 4. Admin reviews the existing course information. 5. Admin updates course details as needed. 6. Admin confirms the updates. |
| Alternative Flow: | * 1. If required information is missing:      1. **System** prompts the user to complete all necessary fields before proceeding. Admin fills in any missing information and reattempts submission. |
| Post Condition: | The new course is successfully added to the platform. Course details are now visible to students, teachers, and coordinators.  Authorized users can manage and interact with the course information. |

#### Notice Board

Table 4.3.2.8 | UC-8 Notice Board

|  |  |
| --- | --- |
| Use case ID: | UC-008 |
| Use case: | Notice Board |
| Actors: | Admin, Teachers, Students |
| Preconditions: | The admin must be logged into the system with the appropriate permissions.  The students and teachers must also be logged in to view the notices. |
| Basic Flow | |
| Admin Actions:   1. Logging in 2. Posting messages 3. Reviewing messages 4. Managing attachments. | Student Actions:   1. Logging in 2. Viewing messages 3. Clicking on messages to view details. |
| Alternative Flow: | * 1. If the admin does not fill in the title field, the system shows an error message stating, "Please fill in the title field."   2. If the admin does not fill in the content field, the system shows an error message stating, Please fill in the   content field. |
| Post Condition: | The details of the posted message will be displayed on the Notice Board of the system. |

#### Allocate Voucher Number

Table 4.3.2.9 | UC-9 Allocate Voucher Number

|  |  |
| --- | --- |
| Use case ID: | UC-009 |
| Use case: | Allocate Voucher Number |
| Actors: | Admin, Students |
| Preconditions: | The admin and the student must be authorized and logged into the system.  The student must post a query before a voucher number can be allocated. |
| Basic Flow | |
| Admin Actions:   1. Logging in 2. Viewing and managing queries 3. Marking queries as seen 4. Responding to queries | Student Actions:   1. Logging in 2. Posting queries 3. Viewing voucher numbers. |
| Post Condition: | The system sends the generated voucher number to the student.  The query is visible in the system with its status. |

#### Favorite Course

Table 4.3.2.10 | UC-10 Favorite Course

|  |  |
| --- | --- |
| Use case ID: | UC-010 |
| Use case: | Favorite Course |
| Actors: | Students |
| Preconditions: | The student is authenticated and logged into the system.  The student has access to the course management module. The course is already added to the system. |
| Basic Flow | |
| System Response:   1. The system checks if the course is already in the student's favorites list. 2. ​   If not, the system adds the selected course to the student's favorites.   1. ​   The system updates the student's favorite courses list.   1. ​   The system displays a confirmation message that the course has been added to the favorites.   1. ​   The updated favorites list is now visible to the student. | Student Actions:   1. The student navigates to the Courses" module. 2. The student selects a course from the available list. 3. The student clicks Add to Favorites. |
| Post Condition: | The selected course is added to the student's list of favorite courses.  The updated favorite courses list is visible to the student. |