**Assignment – CA – groups of 3 students**

1. Gather the necessary requirements from the stakeholders

**Stakeholder: Users (Students, Employees, Full-time and Part-time Lecturers)**

1. **Submit Document:**
   * Users should be able to fill an online form to submit various types of documents.
   * The submission process should include options for attaching documents.
   * The system should support the extension of submitting physical documents, possibly through printing the submitted document.
2. **Make Request to Edit Previous Submission:**
   * Users need the ability to request edits on previously submitted documents.
   * The system should capture the reasons for editing, and users should receive feedback on the status of their requests.
3. **View Document After Submission:**
   * Users should be able to view their submitted documents.
   * If granted, users should be able to edit the document after submission.
4. **Grant Access:**
   * Department officers should have the ability to grant or deny access to edit documents.
   * Officers need a mechanism to review and check the reasons provided by users for document edits.

**Stakeholder: Department Officers**

1. **Organize Documents:**
   * Department officers should be able to organize documents based on criteria such as faculty, department, and instructor.
   * The system should provide features for filtering, sorting, and searching documents efficiently.
2. **Validate the Submitted Document:**
   * Department officers should have the ability to validate submitted documents.
   * The validation process should include a review of the document content.
3. **View List of Validated Documents:**
   * Department officers need a feature to view a list of validated documents.
   * This list should be accessible and may be extended to include printing options.
4. **Sending Emails to Users with Accepted Documents:**
   * Department officers should be able to send notifications or emails to users whose documents have been accepted.
   * The system should facilitate access to the list of validated documents for email communication.

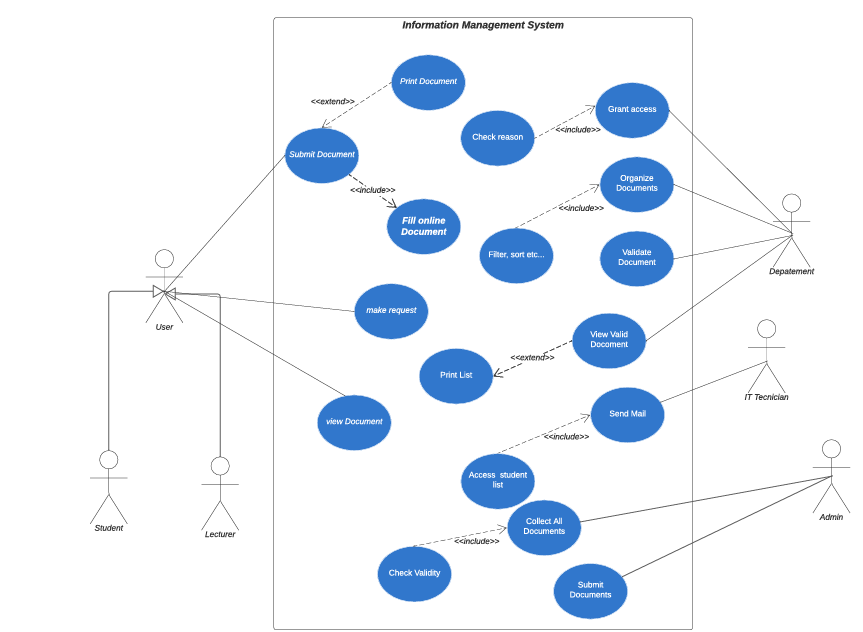
**Stakeholder: Admin/University Ambassador**

1. **Collect and Submit Validated Documents to Other University:**
   * The admin/university ambassador should have the ability to collect all validated documents.
   * The system should support the submission of these documents to another university as requested.

**General System Requirements**

1. **Document Storage and Retrieval:**
   * The system must efficiently store various types of documents.
   * Retrieval should be quick and organized based on different criteria.
2. **User Authentication:**
   * The system should have a secure user authentication mechanism to control access.
3. **Access Control:**
   * Role-based access control should be implemented to ensure that users and officers have appropriate permissions.
4. **Notification System:**
   * The system should include a notification mechanism to inform users about the status of their submissions or requests.
5. **Security and Privacy:**
   * The system must adhere to security standards to protect sensitive user data.
   * Privacy concerns should be addressed in accordance with applicable regulations.
6. **User-Friendly Interface:**
   * The user interface should be intuitive, allowing users and officers to navigate and use the system easily.
7. **Audit Trail:**
   * The system should maintain an audit trail to track user activities, especially for access and validation actions.
8. **Printing and Export:**
   * The system should support the printing of lists, validated documents, and other relevant information.
9. **Scalability:**
   * The system should be designed to handle a potentially large number of documents and users.
10. Design the use case diagram of the project

On the next page you will find the screenshot of our Use Case description Diagram followed by the Use cases descriptions



**1. Submit Document:**

1. **Use Case Name:** Submit Document
2. **Summary:** Users fill an online form to submit various documents. This can be extended to include the option of printing the document.
3. **Dependency:** -
4. **Actors:** User
5. **Preconditions:** User is logged in and has a document to submit.
6. **Description of Main Sequence:**
   1. User logs in to the system.
   2. User fills the online form with document details.
   3. User uploads the document.
   4. System saves the document and associated information.
7. **Description of Alternative Sequence:** User opts to print the submitted document.
8. **Nonfunctional Requirements:** Secure document storage, user-friendly form.
9. **Outstanding Questions:** How will the printing feature be implemented?

**2. Make Request to Edit Previous Submission:**

1. **Use Case Name:** Edit Document Request
2. **Summary:** Users request to edit a previously submitted document.
3. **Dependency:** Submit Document
4. **Actors:** User
5. **Preconditions:** User has a submitted document.
6. **Description of Main Sequence:**
   1. User selects the document to edit.
   2. User submits a request stating the reason for editing.
7. **Description of Alternative Sequence:** User cancels the request.
8. **Nonfunctional Requirements:** Edit request tracking, user notification.
9. **Outstanding Questions:** How will the system handle conflicting edit requests?

**3. View Document After Submission:**

1. **Use Case Name:** View Submitted Document
2. **Summary:** Users view submitted documents, with the option to edit if granted.
3. **Dependency:** Submit Document
4. **Actors:** User
5. **Preconditions:** User has a submitted document.
6. **Description of Main Sequence:**
   1. User selects the document to view.
   2. User views the document details.
7. **Description of Alternative Sequence:** User edits the document if granted access.
8. **Non-functional Requirements:** Access control, document viewing interface.
9. **Outstanding Questions:** How will access control be managed?

**4. Grant Access:**

1. **Use Case Name:** Grant Document Access
2. **Summary:** Department officer grants access to edit a document after checking the reason.
3. **Dependency:** View Document After Submission
4. **Actors:** Department Officer
5. **Preconditions:** Document edit request received.
6. **Description of Main Sequence:**
   1. Department officer reviews the edit request.
   2. Officer grants or denies access based on the reason provided.
7. **Description of Alternative Sequence:** Officer requests additional information.
8. **Non-functional Requirements:** Access control, audit trail.
9. **Outstanding Questions:** How will officers be notified of pending access requests?

**5. Organize Documents:**

1. **Use Case Name:** Organize Documents
2. **Summary:** Admin organizes documents with filtering, sorting, etc.
3. **Dependency:** -
4. **Actors:** Admin
5. **Preconditions:** Admin is logged in.
6. **Description of Main Sequence:**
   1. Admin accesses the document organization tools.
   2. Admin applies filters, sorts documents, etc.
7. **Description of Alternative Sequence:** Admin saves a custom document organization scheme.
8. **Non-functional Requirements:** Efficient document organization tools.
9. **Outstanding Questions:** What specific filters and sorting options are needed?

**6. Validate Submitted Document:**

1. **Use Case Name:** Validate Document
2. **Summary:** Department officer validates a submitted document.
3. **Dependency:** View Document After Submission
4. **Actors:** Department Officer
5. **Preconditions:** Document submitted and access granted.
6. **Description of Main Sequence:**
   1. Officer reviews the submitted document.
   2. Officer marks the document as validated.
7. **Description of Alternative Sequence:** Officer requests document modifications before validation.
8. **Non-functional Requirements:** Validation tracking, document status updates.
9. **Outstanding Questions:** How will validation be communicated to the user?

**7. View List of Validated Documents:**

1. **Use Case Name:** View Validated Documents
2. **Summary:** Department officer views a list of validated documents, with an option to print.
3. **Dependency:** Validate Submitted Document
4. **Actors:** Department Officer
5. **Preconditions:** Officer is logged in.
6. **Description of Main Sequence:**
   1. Officer accesses the list of validated documents.
   2. Officer views document details.
7. **Description of Alternative Sequence:** Officer prints the list of validated documents.
8. **Non-functional Requirements:** Efficient list retrieval, document printing.
9. **Outstanding Questions:** What details should be included in the list?

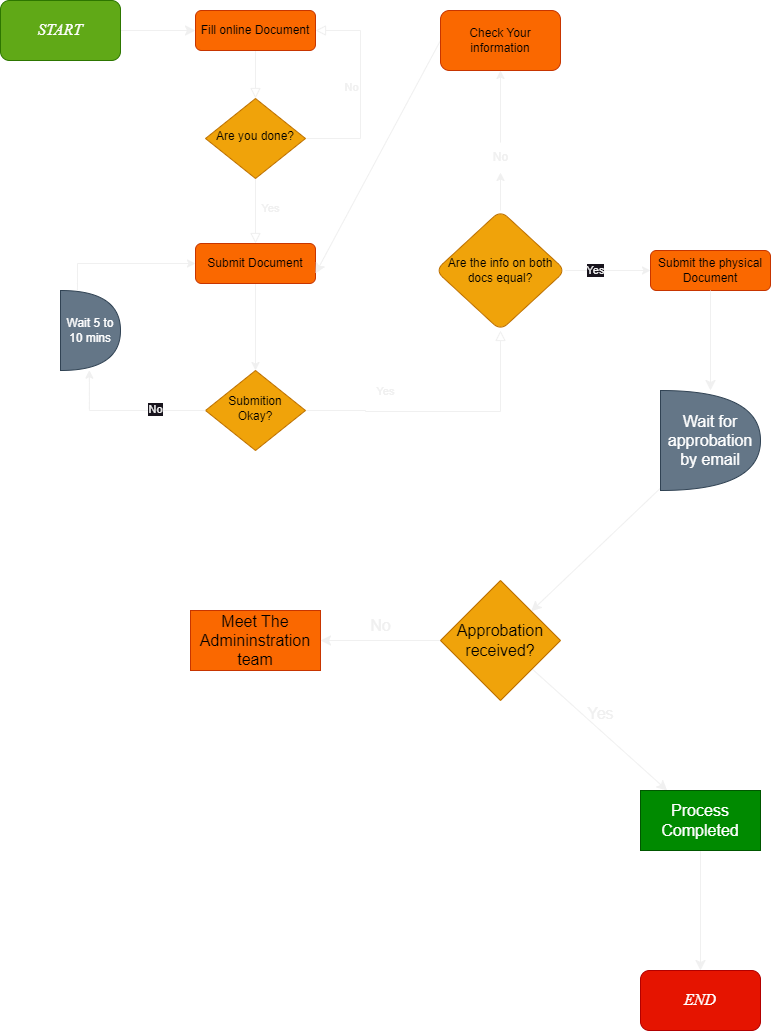
**8. Send Emails to Users with Accepted Documents:**

1. **Use Case Name:** Notify Accepted Documents
2. **Summary:** System sends emails to users whose documents have been accepted, with access to the list.
3. **Dependency:** View List of Validated Documents
4. **Actors:** System
5. **Preconditions:** Validated documents exist.
6. **Description of Main Sequence:**
   1. System generates an email list.
   2. System sends emails with access links.
7. **Description of Alternative Sequence:** -
8. **Nonfunctional Requirements:** Email delivery, secure access links.
9. **Outstanding Questions:** How often will the notifications be sent?

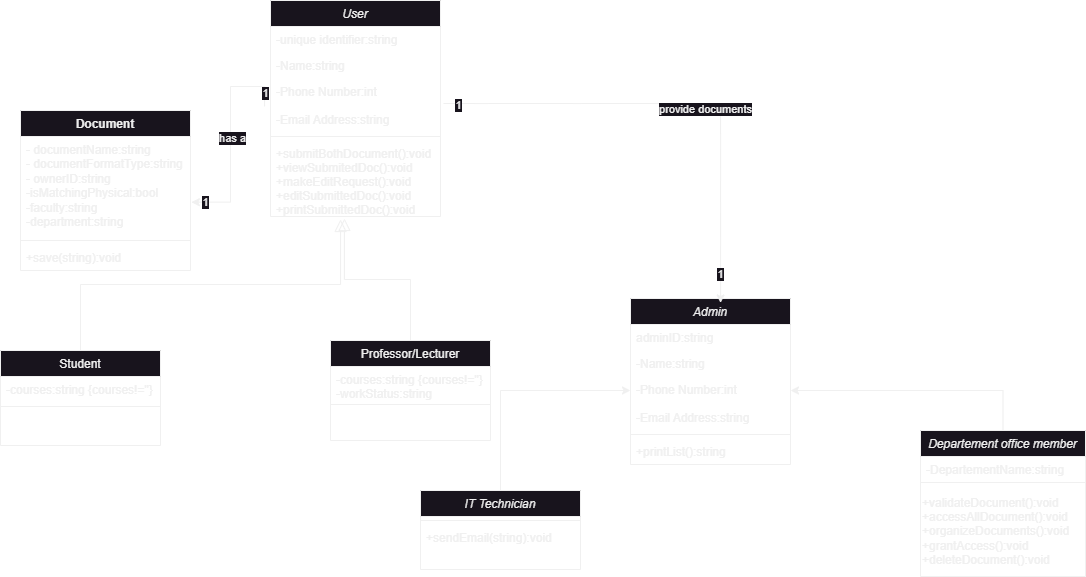
**9. Collect and Submit Validated Documents to Other University:**

1. **Use Case Name:** Submit Validated Documents
2. **Summary:** Admin or university ambassador collects and submits validated documents to another university.
3. **Dependency:** View List of Validated Documents
4. **Actors:** Admin or University Ambassador
5. **Preconditions:** Validated documents exist.
6. **Description of Main Sequence:**
   1. Admin accesses the list of validated documents.
   2. Admin selects documents for submission.
   3. Admin submits documents to the other university.
7. **Description of Alternative Sequence:** Admin requests additional information for submission.
8. **Nonfunctional Requirements:** Secure submission, tracking of submitted documents.
9. **Outstanding Questions:** What format will the documents be submitted in?

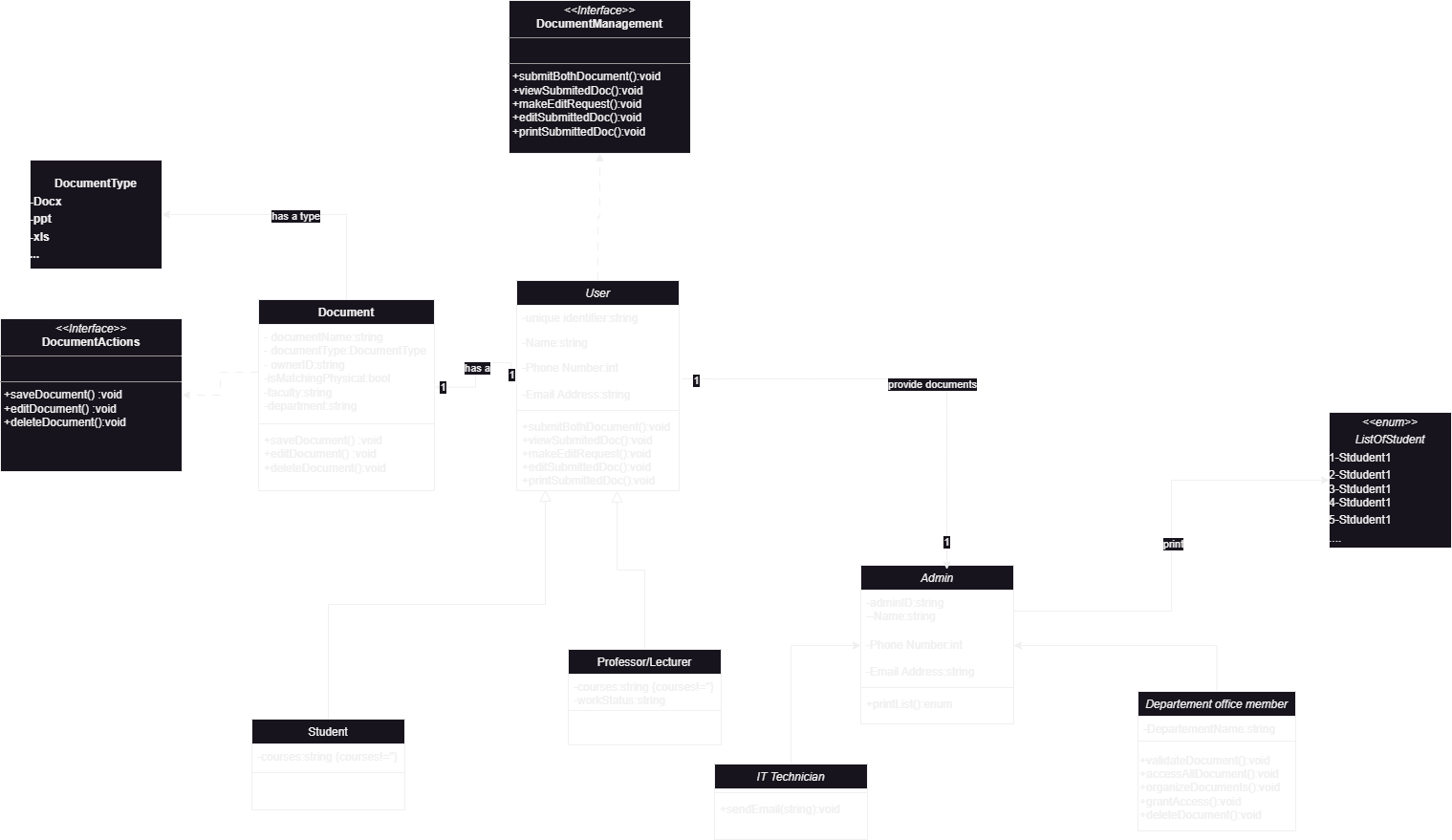
4. Design the application flow chart



1. Design the class diagram of the application to be developed



1. Do object structuring and refine the class diagram with design principles and 7. Refine the class diagram with the use of design patterns



1. Design the sequence diagram for the use cases

