FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGIES

FALL 2023

CA EXAMINATION

Assignment - CA - groups of 3 students

NDE HURICH DILAN nde.dilan@ictuniversity.edu.cm
ICTU20223351

NGANA NOA JUNIOR FREDERIC ABEL

ngana.fredericabel@ictuniversity.edu.cm

ICTU20223538

Mapa Tetong Caline Staelle

mapatetong.staelle@ictuniversity.edu.cm

ICTU20223062

1. Gather the necessary requirements from the stakeholders
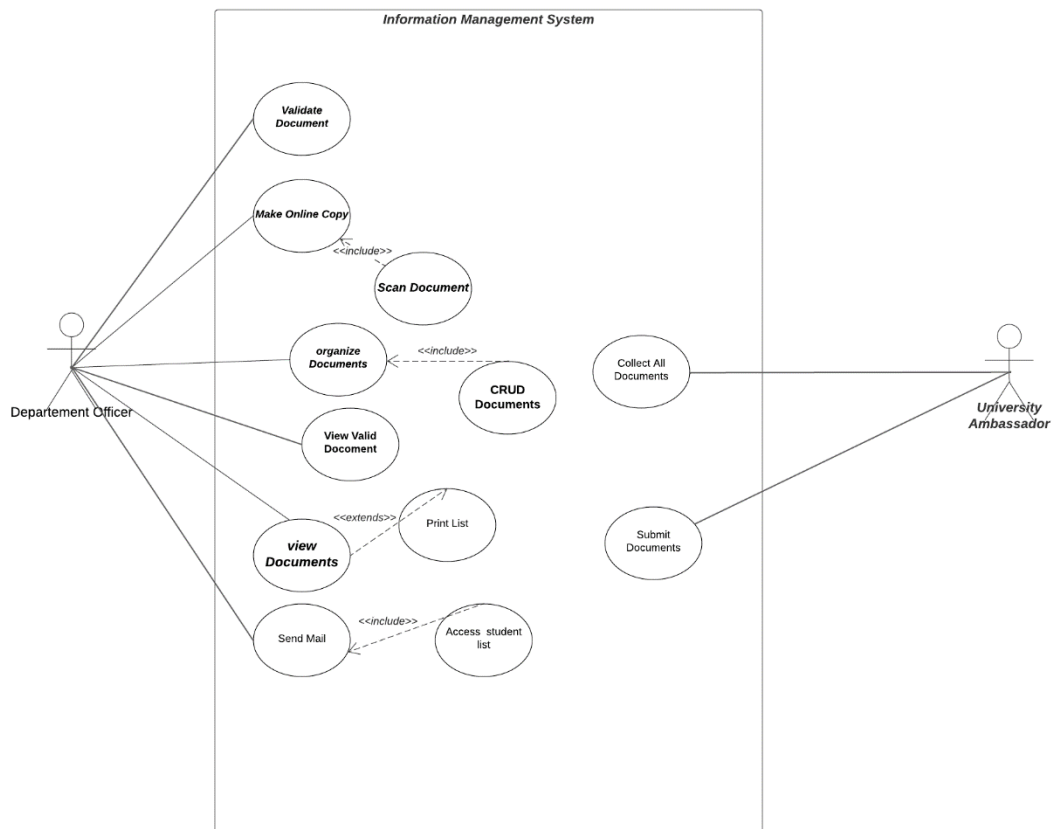
1. Validate Document (Done by the Department Officer):
- **User Requirements:**
    - The department officer should be able to initiate document validation.
    - The system should highlight errors or invalid information.
    - A notification should be sent to the user if validation fails.

2. Produce an Online Version of the Validated Document (Done by the Department Officer):
- **User Requirements:**
    - The department officer should have the option to scan a physical document.
    - The system should save the scanned file on the cloud.
    - Access to the online version should be restricted to authorized personnel.

3. View Document After Submission (Including Editing/Changing):
- **User Requirements:**
    - Users should be able to view their submitted documents.
    - Editing rights should be granted based on specific criteria.
    - Changes made to the document should be recorded for auditing purposes.

4. Grant Access (Done by the Department Officer After Checking Reason):
- **User Requirements:**
    - Users should be able to request access to edit a document.
    - The department officer should review the reason for access.
    - Granting or denying access should be accompanied by a notification to the user.

5. Organize Documents by Faculty, Department, Instructor, and Other Criteria (Done by the Department Officer):
- **User Requirements:**
    - The department officer should be able to organize documents using various criteria.
    - Filtering and sorting options should be provided.
    - The system should allow for the creation of custom organization schemes.

6. View All Submitted Documents (Done by the Department Officer):
- **User Requirements:**
    - The department officer should have access to a list of all submitted documents.
    - The list should provide relevant details for each document.
    - Access should be secure and restricted to authorized personnel.

7. View All Validated Documents and Print a List (Done by the Department Officer):
- **User Requirements:**

- The department officer should have access to a list of all validated documents.
- The list should be printable for record-keeping purposes.
- The information presented should include document details and validation status.

8. Send Emails to Users Whose Documents Have Been Accepted (Done by the Department Officer):

- **User Requirements:**
  - The department officer should be able to send notifications to users whose documents have been accepted.
  - The notification should include access to the list of validated documents.

9. Admin (University Ambassador) Collects Validated Documents and Submits to Another University:

- **User Requirements:**
  - The admin (university ambassador) should be able to access all validated documents.
  - The system should facilitate the collection of validated documents for submission.
  - A confirmation mechanism should be in place after submission.

Common Requirements Across Use Cases:

- **Security:**
  - Access to sensitive information should be secure and role-based.
  - User authentication and authorization mechanisms should be implemented.
- **Audit Trail:**
  - Changes to documents, validation status, and access grants should be logged for auditing.
- **Usability:**
  - The user interface should be intuitive and user-friendly.
  - Necessary training materials or guides should be provided.
- **Integration:**
  - The system should be capable of integrating with scanning tools for document digitization.
  - Email notifications should be integrated seamlessly.
- **Performance:**
  - The system should handle a reasonable volume of documents without significant performance issues.
- **Compliance:**
  - The system should comply with data protection regulations and university policies.

2. <u>Design the use case diagram of the project</u>

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

Information Management System

- Validate Document
- Make Online Copy
- <<include>> Scan Document
- organize Documents
- <<include>> CRUD Documents
- Collect All Documents
- View Valid Document
- view Documents <<extends>> Print List
- Send Mail <<include>> Access student list
- Submit Documents

Departement Officer

University Ambassador

3. For each use case, write the use case description

Use Case 1: Validate Document (done by the department officer)

1. **Use Case Name: Validate Document**

2. **Summary: The department officer validates submitted documents by checking for errors or non-valid information.**

3. **Dependency: None**

4. **Actors: Department Officer**

5. **Preconditions: The document must be submitted for validation.**

6. **Description of Main Sequence:**

   - **The department officer reviews the submitted document.**

   - **Checks for errors or non-valid information.**

- Marks the document as validated if it meets the criteria.

7. **Description of Alternative Sequence:** If errors are found, the department officer communicates the issues to the user for correction.

8. **Non functional Requirements:** Efficient processing time, accurate validation.

9. **Outstanding Questions:** How to handle invalid documents?

Use Case 2: Produce an Online Version of the Validated Document (done by the department officer)

1. **Use Case Name:** Produce Online Version of Validated Document

2. **Summary:** The department officer scans the validated physical document and saves the scanned file on the cloud.

3. **Dependency:** Document must be validated.

4. **Actors:** Department Officer

5. **Preconditions:** The document must be validated.

6. **Description of Main Sequence:**

- The department officer scans the validated document.

- Saves the scanned file on the cloud.

7. **Description of Alternative Sequence:** None

8. **Non functional Requirements:** Secure cloud storage, efficient scanning process.

9. **Outstanding Questions:** What cloud storage system will be used?

Use Case 3: View Document After Submission

1. **Use Case Name:** View Document After Submission

2. **Summary:** Users can view submitted documents and edit/change them if granted the right to do so.

3. **Dependency:** Document must be submitted.

4. **Actors:** User

5. **Preconditions: The document must be submitted.**

6. **Description of Main Sequence:**

   - **Users access the system and locate their submitted document.**

   - **View the document details.**

   - **Edit/Change the document if granted permission.**

7. **Description of Alternative Sequence: If no permission is granted, the user can only view the document.**

8. **Non functional Requirements: User-friendly interface, secure editing permissions.**

9. **Outstanding Questions: How is permission granted for editing?**

4. Design the application flow chart
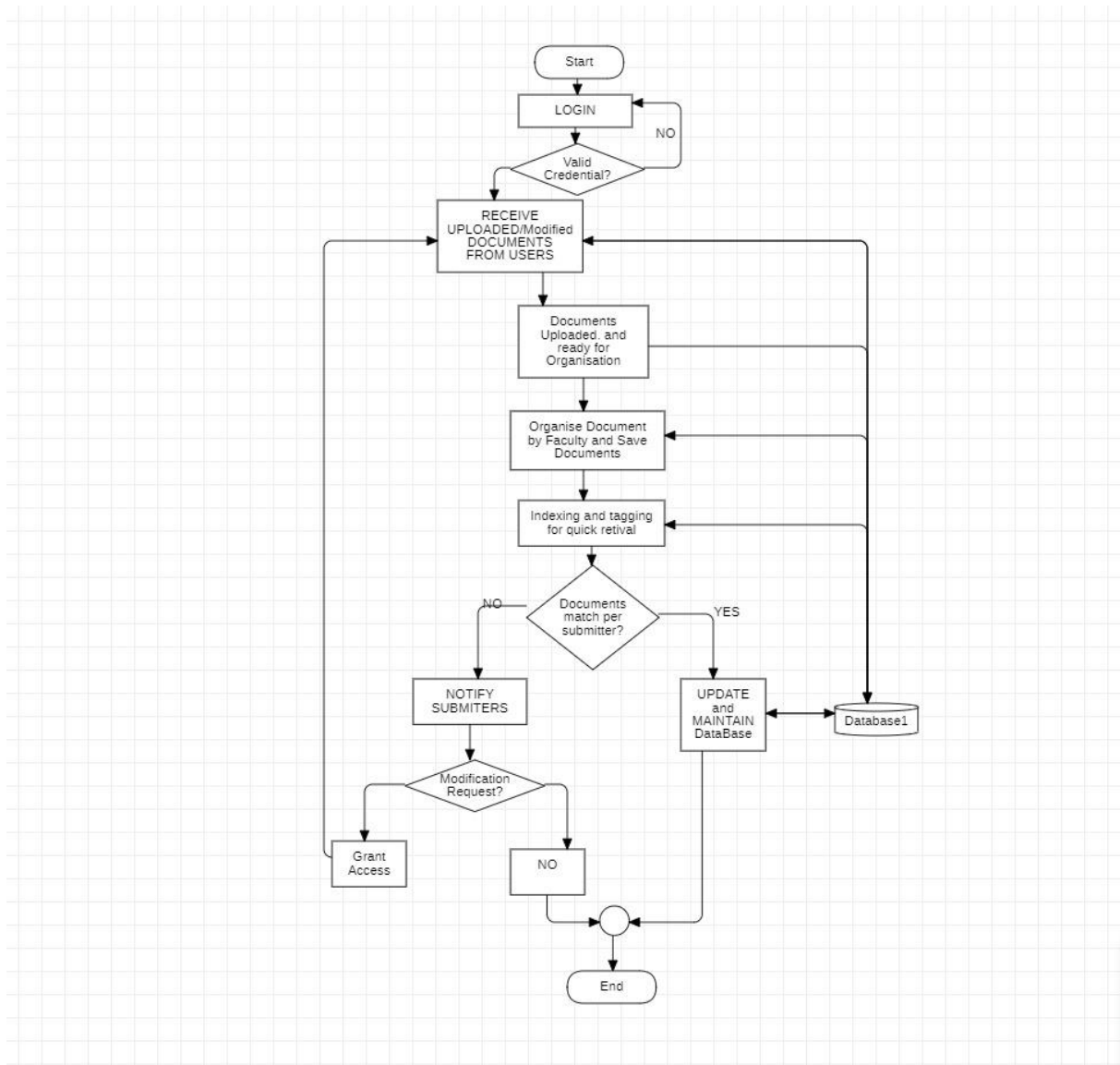
1. Design the class diagram of the application to be developed

First Class Diagram

User *(abstract)*
- -unique identifier:string
- -Name:string
- -Phone Number:int
- -Email Address:string

Document
- - documentName:string
- - documentFormatType:string
- - ownerID:string
- -isMatchingPhysical:bool
- -faculty:string
- -department:string
- +save(string):void

has a 1 ... 1

Admin *(abstract)*
- adminID:string
- -Name:string
- -Phone Number:int
- -Email Address:string
- +printList():string
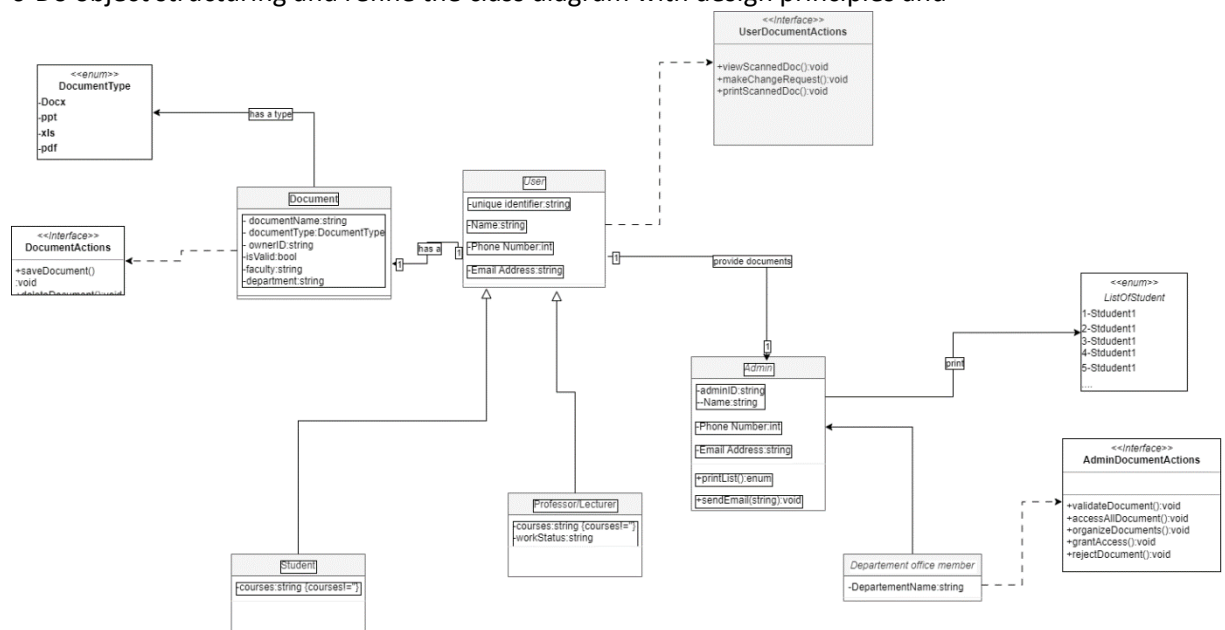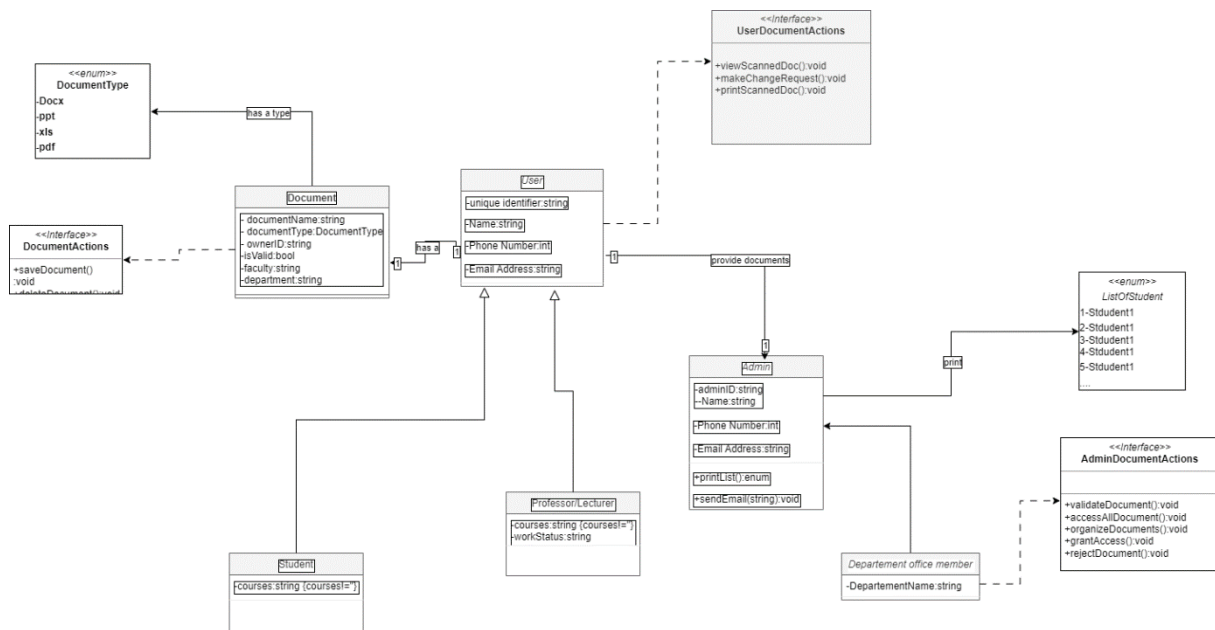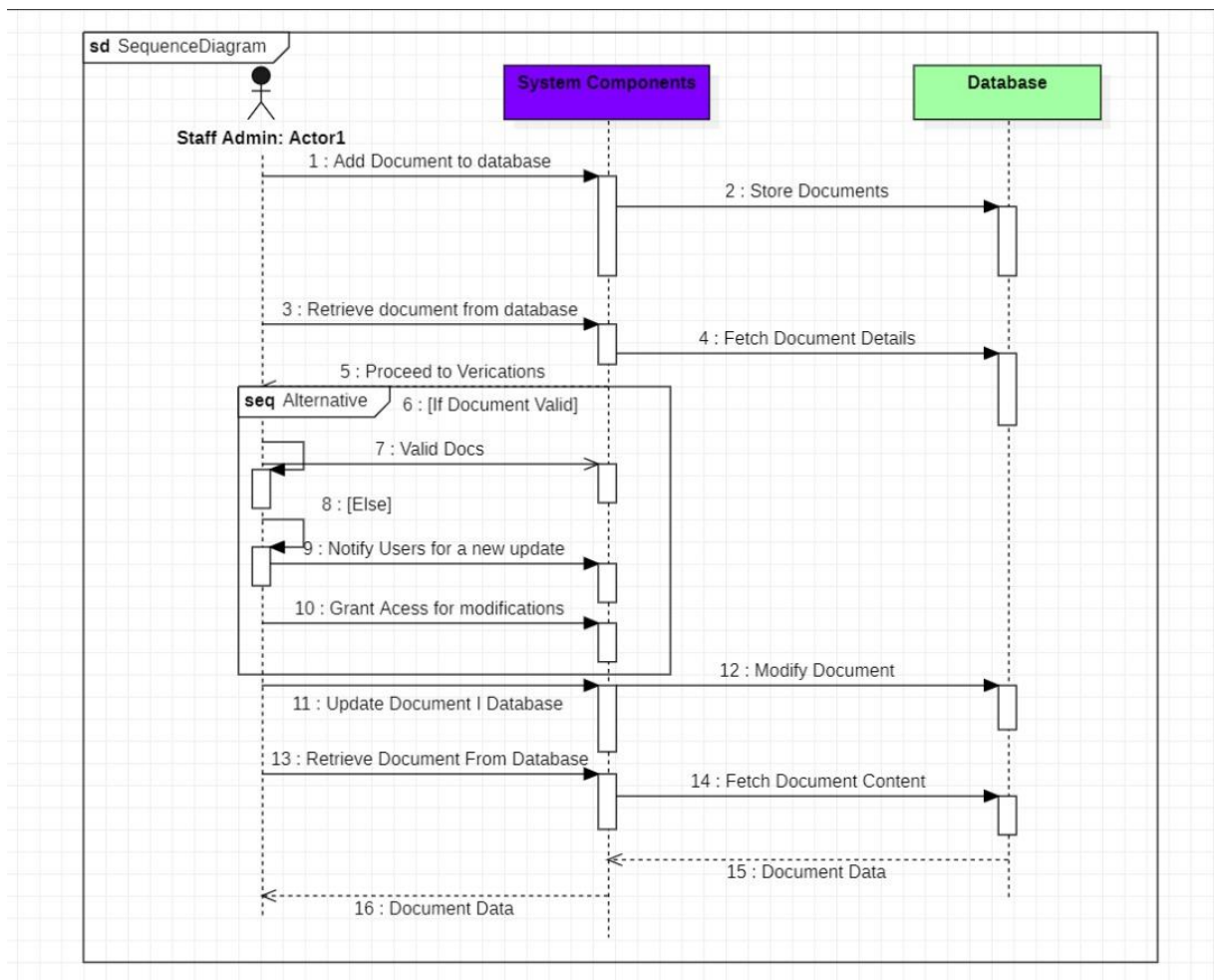
1 — provide documents — 1

Student
- -courses:string {courses!=''}

Professor/Lecturer
- -courses:string {courses!=''}
- -workStatus:string

University's ambassador
- +collectAllDocuments():void
- +submitAllDocuments():void

Departement office member
- -DepartementName:string
- +validateDocument():void
- +accessAllDocument():void
- +organizeDocuments():void
- +grantAccess():void
- +deleteDocument():void

6-Do object structuring and refine the class diagram with design principles and

Second Class Diagram

<<enum>> DocumentType
- -Docx
- -ppt
- -xls
- -pdf

has a type

<<Interface>> UserDocumentActions
- +viewScannedDoc():void
- +makeChangeRequest():void
- +printScannedDoc():void

<<Interface>> DocumentActions
- +saveDocument() :void
- -deleteDocument():void

Document
- -documentName:string
- -documentType:DocumentType
- -ownerID:string
- -isValid:bool
- -faculty:string
- -department:string

User *(abstract)*
- -unique identifier:string
- -Name:string
- -Phone Number:int
- -Email Address:string

has a   provide documents

<<enum>> ListOfStudent
- 1-Stdudent1
- 2-Stdudent1
- 3-Stdudent1
- 4-Stdudent1
- 5-Stdudent1
- ...

Admin *(abstract)*
- -adminID:string
- -Name:string
- -Phone Number:int
- -Email Address:string
- +printList():enum
- +sendEmail(string):void

print

<<Interface>> AdminDocumentActions
- +validateDocument():void
- +accessAllDocument():void
- +organizeDocuments():void
- +grantAccess():void
- +rejectDocument():void

Professor/Lecturer
- -courses:string {courses!=''}
- -workStatus:string

Student
- -courses:string {courses!=''}

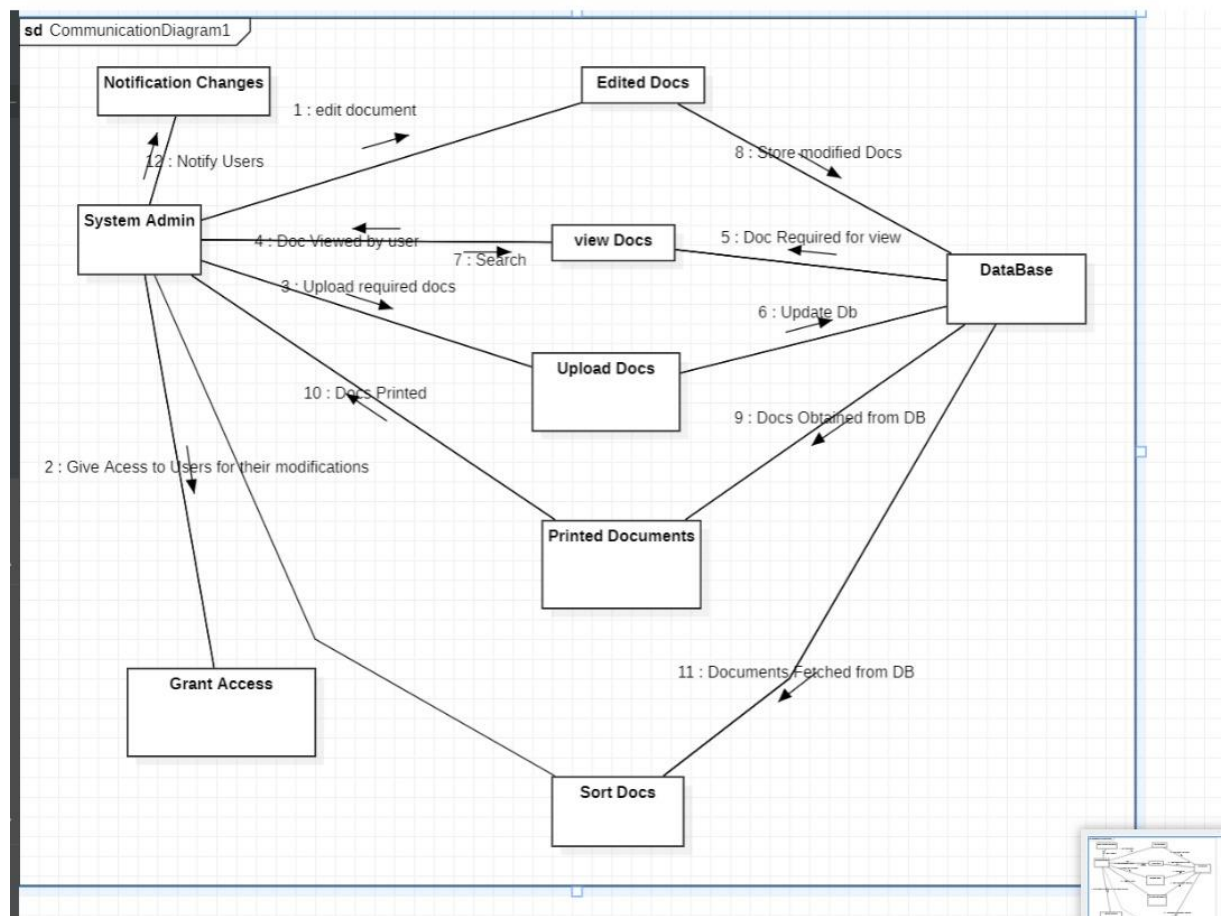**Departement office member**
- -DepartementName:string

---

7. Refine the class diagram with the use of design patterns

8-Design the sequence diagram for the use cases

9-Convert the sequence diagram into collaboration diagram

sd CommunicationDiagram1

Notification Changes

1 : edit document

12 : Notify Users

Edited Docs

8 : Store modified Docs

System Admin

4 : Doc Viewed by user

7 : Search

view Docs

5 : Doc Required for view

DataBase

3 : Upload required docs

6 : Update Db

Upload Docs

10 : Docs Printed

9 : Docs Obtained from DB

2 : Give Acess to Users for their modifications

Printed Documents

Grant Access

11 : Documents Fetched from DB

Sort Docs

10. Chose three objects from object structuring and design the state transition diagram

ADMIN

Login

Receives documents

validates documents

Grants access

Prints list