

PROJECT NO: 14



DOCUMENT MANAGEMENT SYSTEM

OOM-MINI-PROJECT



PROJECT NO: 14

OUR TEAM

IIT2021189 VEDANSH RAGHUVANSHI

IIT2021159 ANUJ KUMAR

IIT2021015 SRAVAN

GUIDED BY :-

Prof. Op Vyas sir,

Bagesh Kumar Sir,

Ayush sihha sir,

Harshit Gupta sir.

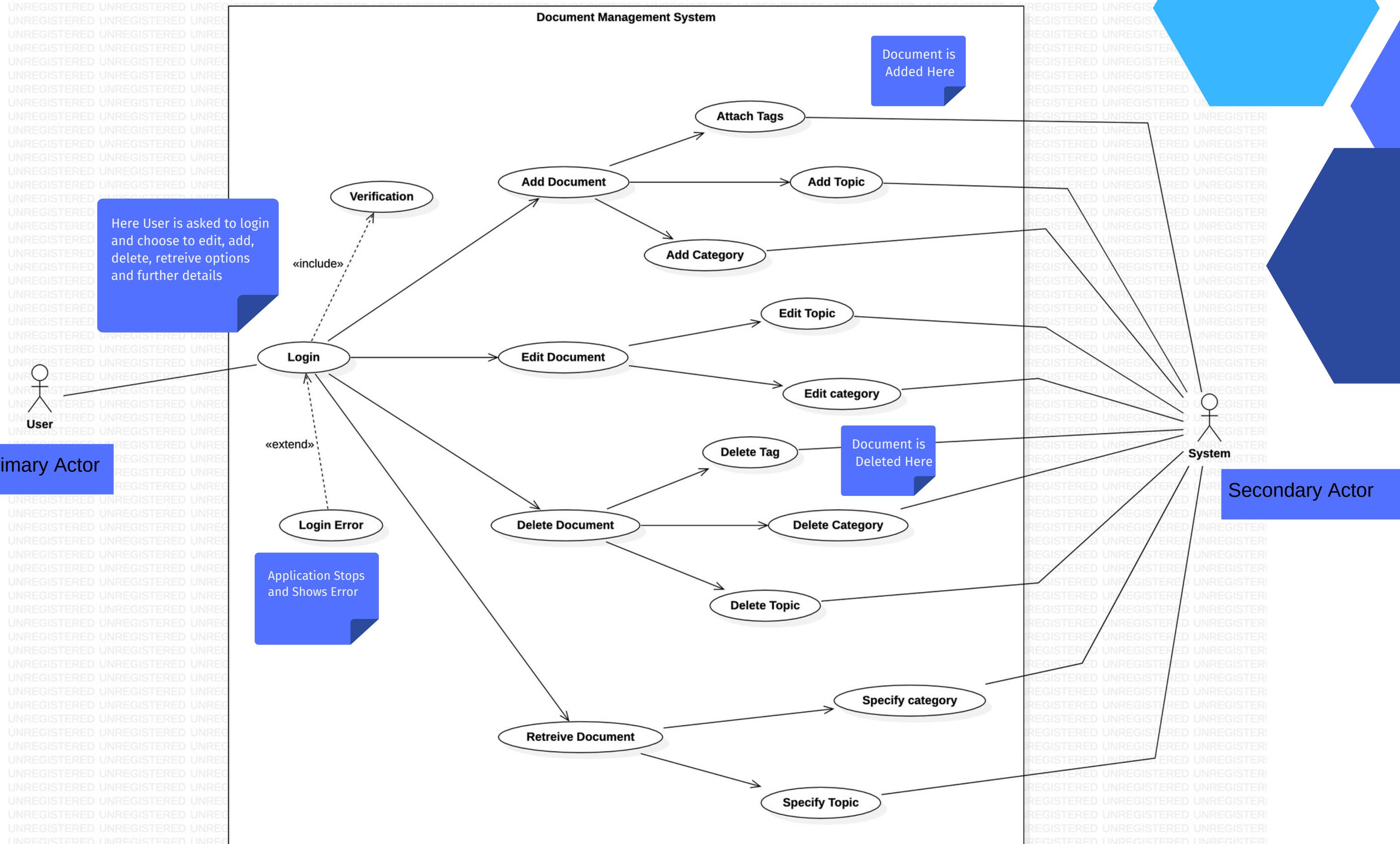


As the name suggests this is just a simple document management system providing basic functionalities like adding files, deleting files, modifying files etc., the purpose of this mini-project was not to make a full-fledged completely functional document management system, instead we just wanted to learn about basic java libraries and how to work with them.

We Used Java and Java swing To use develop this application. As we had to interact with the OS to facilitate adding, updating, deleting files and directories through our application, we had to use a java Runtime object to help us with these tasks but that meant our project could no longer be run on windows as we had to use linux only commands and features like mkdir, rm, edit etc.

Overview of the Project

UseCase Diagram



Some Major Functional Requirements

UseCase - 1 : **Add Document**

- Allows User to Add Document File in the application.
- **Actor :** User ♂
- Once file added, user can access all features of the application.
- **Preconditions:** Internet Connectivity

UseCase - 2 : **Edit Document**

- Allows User to edit Document File in the application like editing the topic or category.
- **Actor :** User ♂
- Once file edited, File is saved in the application.
- **Preconditions:** User must upload or added some data or file in the application.
- Internet Connectivity

UseCase - 3 : **Delete Document**

- Allows User to delete Document File in the application.
- **Actor :** User ♂
- Once file deleted, There is no chance to undo it in the application.
- **Preconditions:** User must upload or added some data or file in the application.
- Internet Connectivity

UseCase - 4 : **Retreive Document**

- Allows User to retreive Document File in the application.
- **Actor :** User ♂
- Like user can specify topic or category.
- **Preconditions:** Internet Connectivity

UseCase - 5 :

User Login

- Allows User to login in the application.
- **Actor :** User ♂
- Once the user Logged in, user can access all features of the application.
- **Preconditions:** Internet Connectivity

UseCase - 6 :

Authorized Login

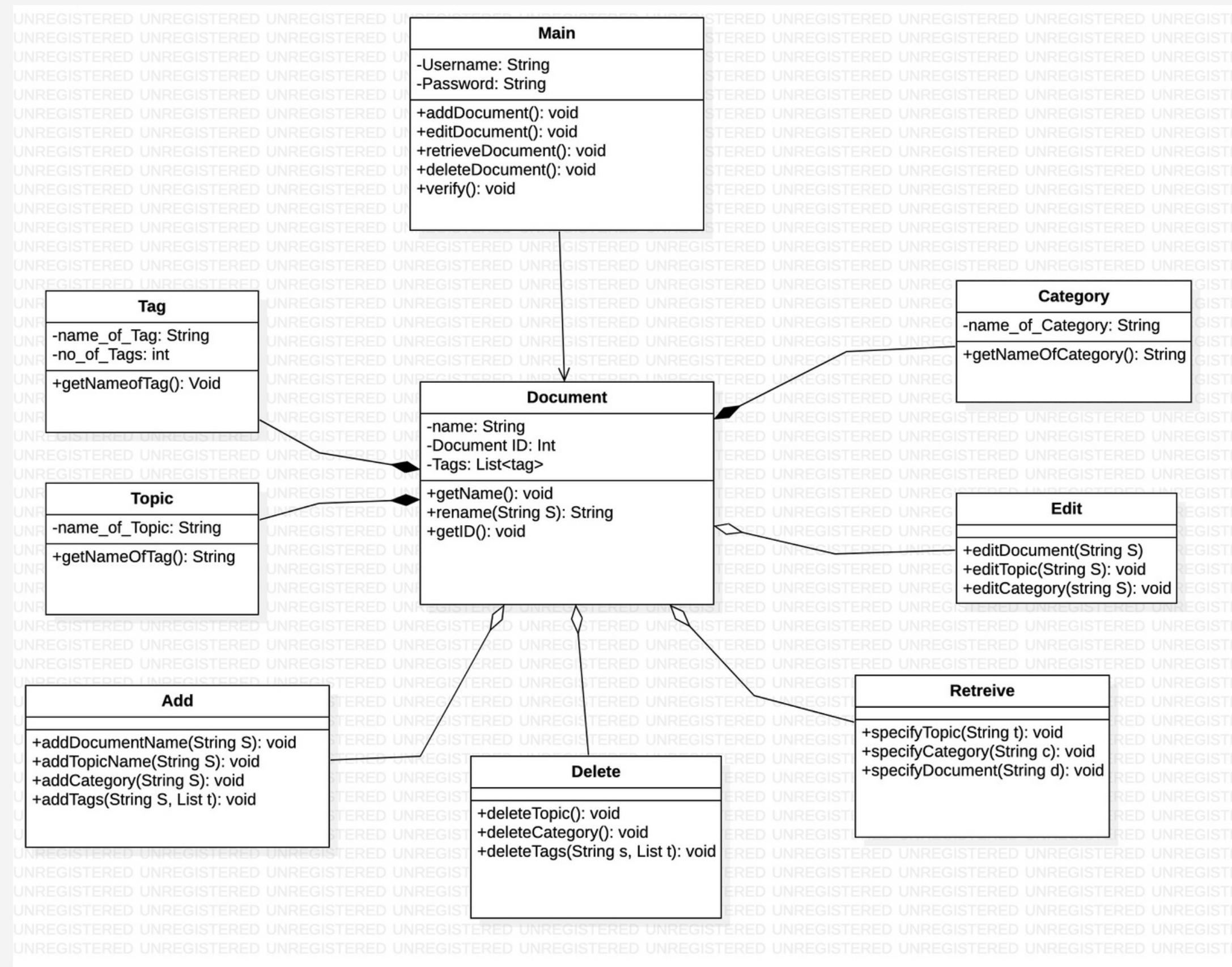
- Allows Admin to login in the application.
- **Actor :** Admin/system ♂
- checks for the authorization of login. Only for allowed mail ids.
- **Preconditions:** Internet Connectivity
- if Id or password incorrect. Shows error dialog box.

UseCase - 7:

Verification

- Allows Admin to verify the details submitted by the user .
- **Actor :** Admin / system ♂
- If the details are incorrect, an error message is shown on dialog box.
- **Preconditions:** User must send correct details to avoid errors or login failed.
- Internet Connectivity

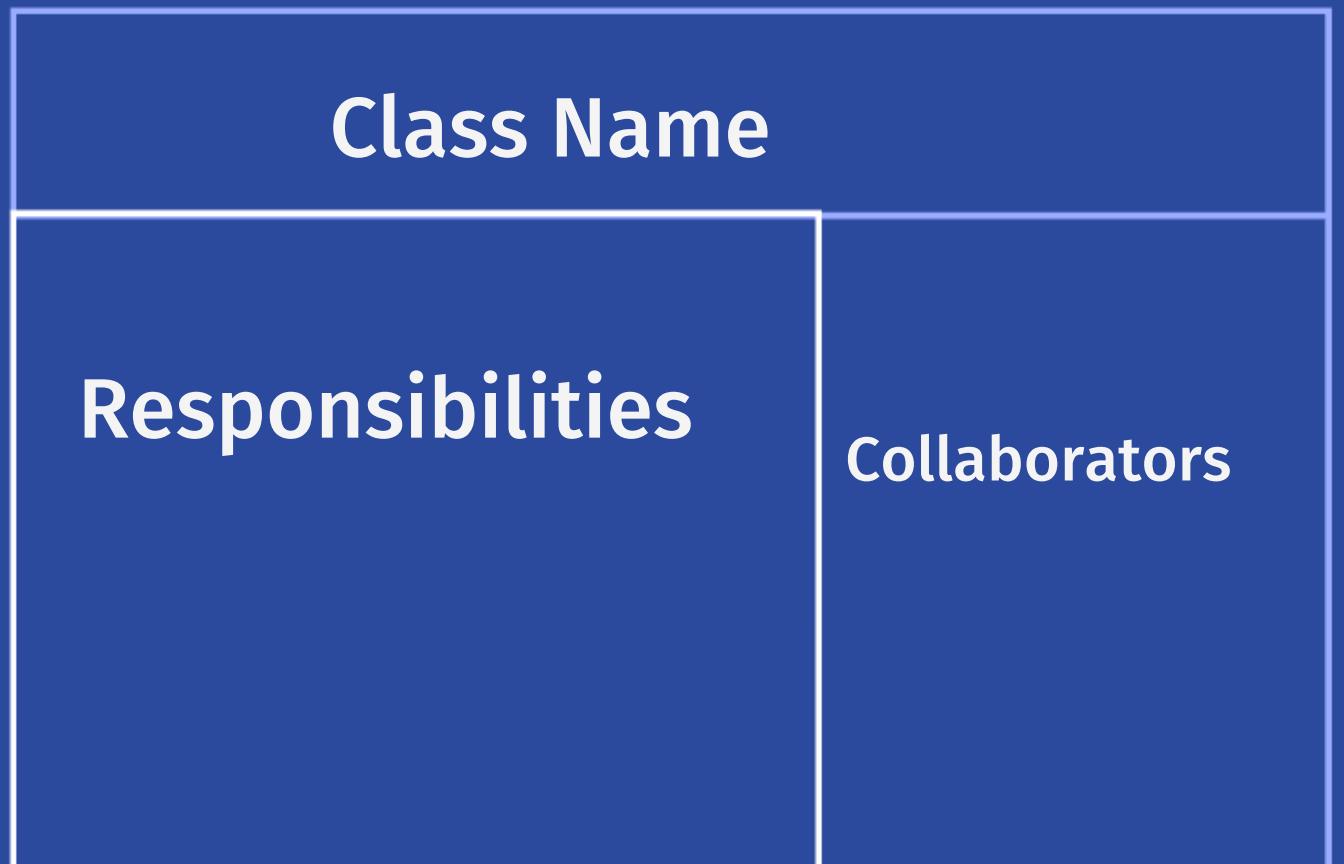
Class Diagram



CRC Diagram

Class-responsibility-collaboration (CRC) cards are a brainstorming tool used in the design of object-oriented software. To create a CRC card, you can begin by writing out a scenario which identifies the major actors and actions which the actors do.

Only write out actions and actors specific to that particular scenario. Nouns should turn into the classes of the card, verbs typically turn into the responsibilities of the card, and collaborators are the other cards with which the card will be interacting.



Main

- Responsible for choosing b/w options of actions on Document
- These actions are typically Add Edit and Delete

- Document
- AddDocument
- EditDocument
- DeleteDocument
- RetreiveDocument

Document

- Responsible for holding data
- Responsible for having tags, topic category
- Responsible for actions on documents

- Tag
- Category
- Topic

Tag

- Responsible for attaching tags to a document
- Responsible for holding record of each tag for each document

Document

- Responsible for defining the category of each Document

Document

- Document

Topic

- Responsible for attaching a topic to each document

- Document

Add Document

- Responsible for adding topic, category and tags for each document added

- Document
- Topic
- Category
- Tag

Edit Document

- Responsible for updating and changing topic , category and tags for each document

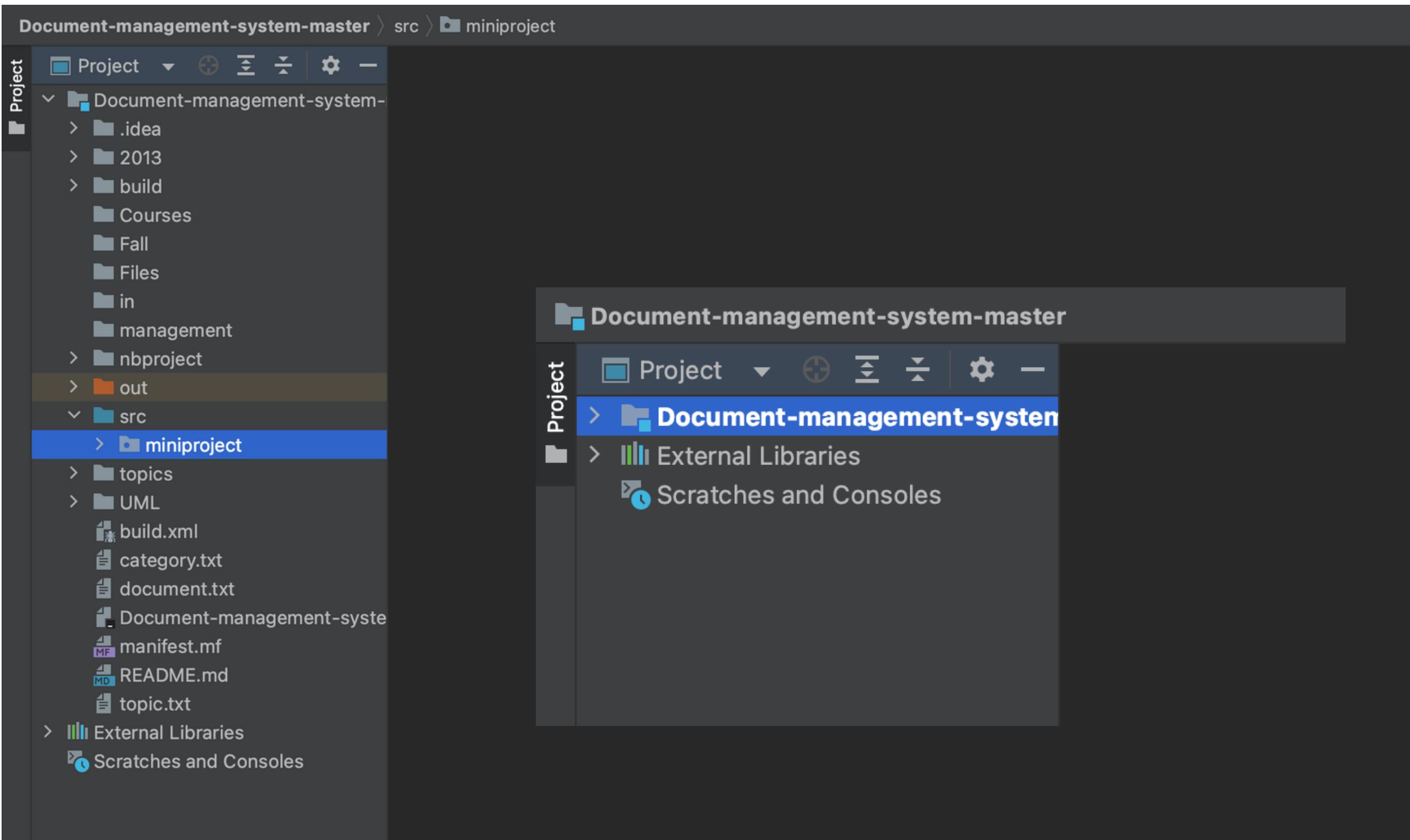
- Document
- Topic
- Category
- Tag

Delete Document

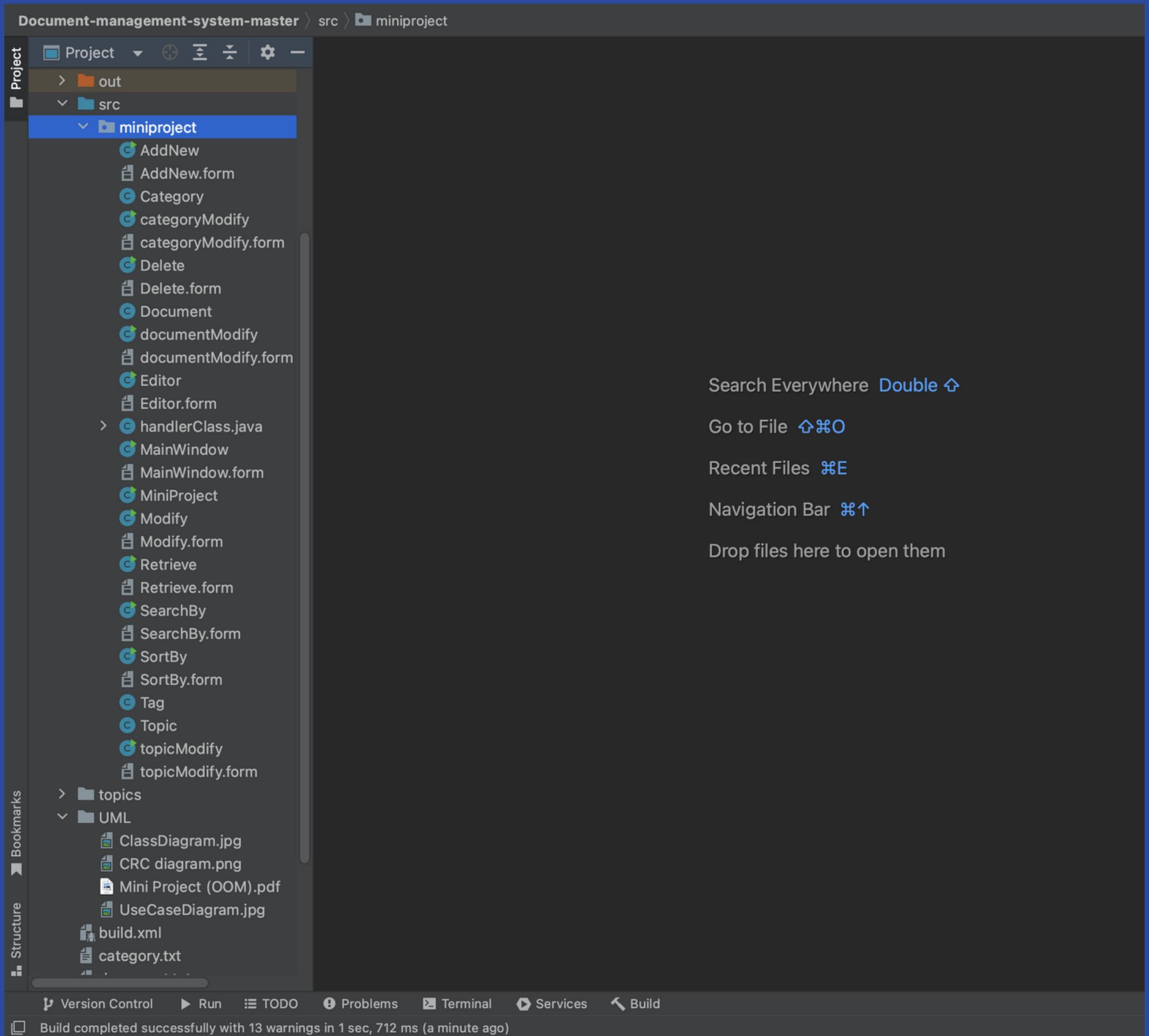
- Responsible for deleting all contents of a document

- Document
- Topic
- Category
- Tag

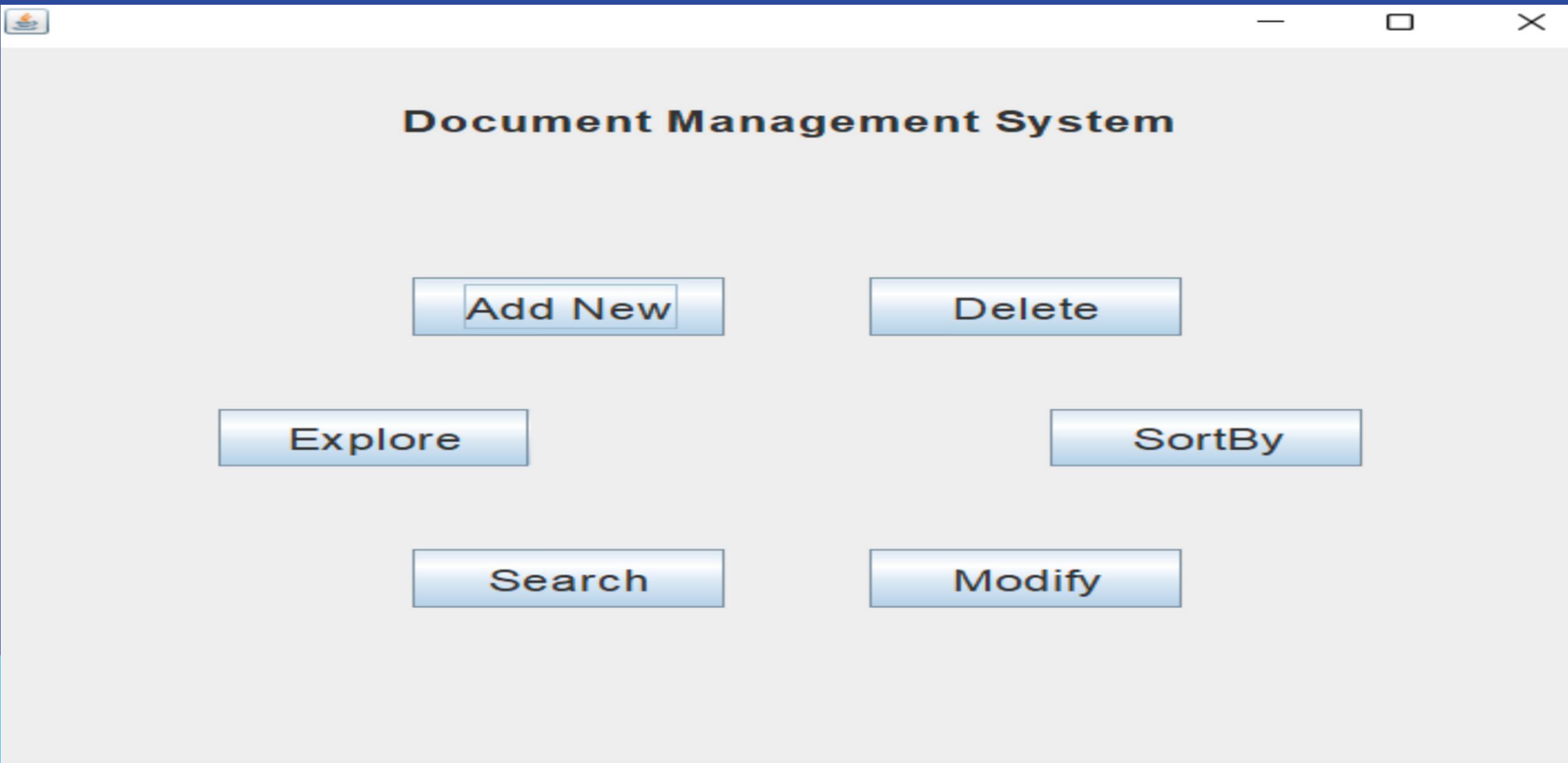
CODE :



Implementation of code



MainWindow (UI)





Add A Document



Name :

Category Name :

Topic Name :

Tags (Separated By Commas) :

Add

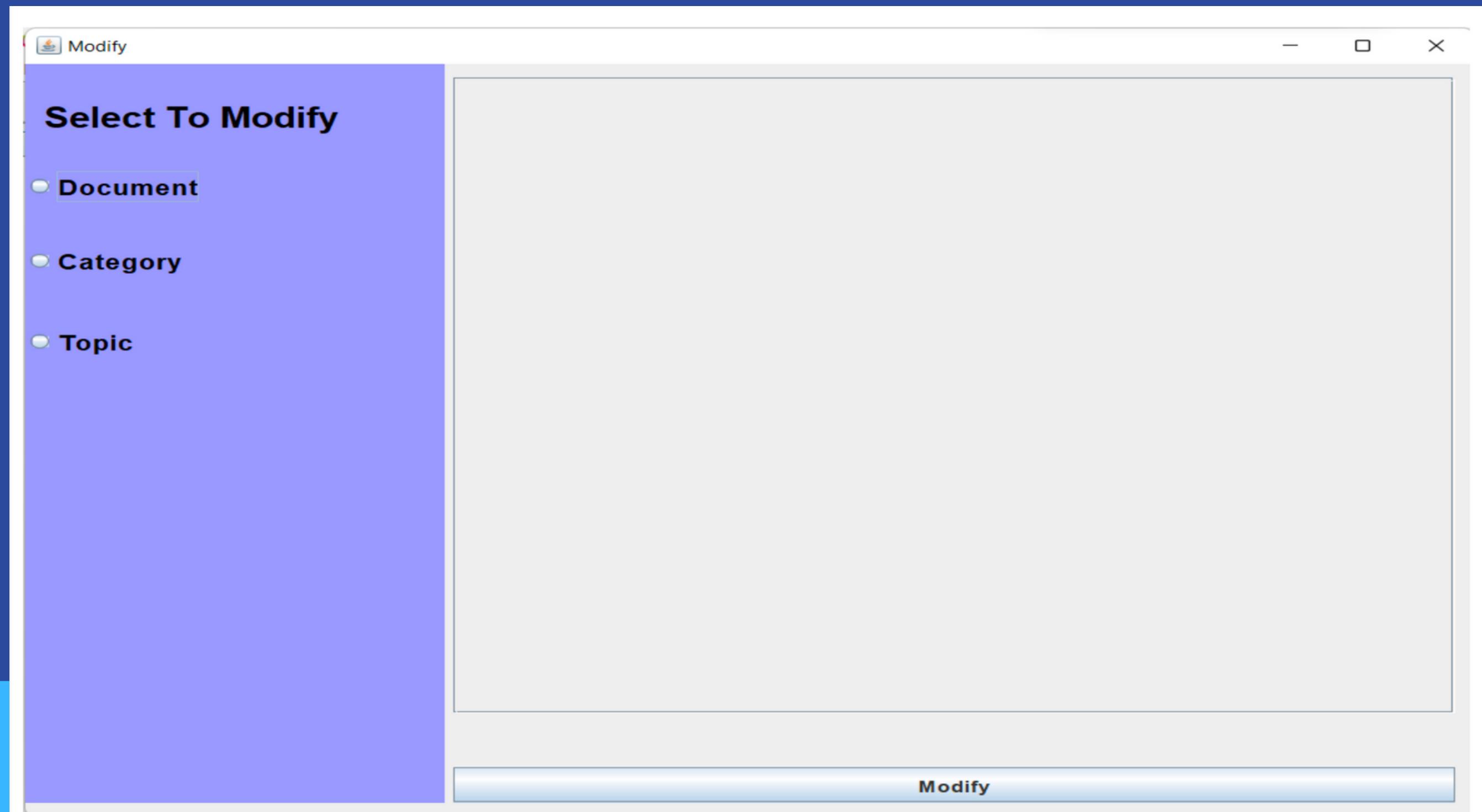




Select To Explore

- Document
- Category
- Tag
- Topic

Open



 Search

Search By :

Name

Category

Topic

Tag

Search

Edit **Delete**



Explore Our System

Sort By :

Name

Category

Date Modified

Topic

| Name | Path |
|--------------|---------------|
| t1d1.txt | topics/topic1 |
| t1d2.txt | topics/topic1 |
| t1d3.txt | topics/topic1 |
| same.txt | topics/topic1 |
| t2d1.txt | topics/topic2 |
| t2d2.txt | topics/topic2 |
| t2d3.txt | topics/topic2 |
| same.txt | topics/topic2 |
| t3d1.txt | topics/topic3 |
| t3d2.txt | topics/topic3 |
| same.txt | topics/topic3 |
| document.pdf | topics/None |
| document.pdf | topics/None |
| document.pdf | topics/None |
| hello | topics/none |

Open

Object Oriented Methodologies

THANK YOU

FOR EVERYTHING.



Prof. O.P.Vyas sir



Bagesh sir



Ayush Sinha sir



Harshit sir