Document Management System

Lawrence Mensah

Huzaifa Anas

Tommy Lin

The project we worked on contained the following classes:

Main

Category

CategoryRepository

Document

DocumentRepository

Topic

TopicRepository

TopicService

We used a separate GUI folder to store the window and display classes. The size, color and buttons are all referenced in the Gui class.

A screenshot of a cell phone

Description automatically generated

For this project, we started with the document class as the original foundation of our project. We did this because it would contain everything of importance, like the tags, storage folder, category, and topic ID. We kept the other concepts, tags, categories, and topics separate though as a user might just want to edit those specifically or look at those. Moving forward then we made interfaces to be repositories for the list, and hence the origin of the name. Furthermore, with this, we were then able to have our I/O in the third connecting chain through having feature like DocumentService implements List, which is building off Class Document -> DocumentRepository interface. In this way, we were able to see the logical evolution. First, we needed to know what really, we are doing. Then we needed a place to store it. Lastly, then we needed to implement this storage location, and then be able to take inputs in and out. Now to achieve this goal we first, had our instance variables in our classes, like the document. Then we followed it with methods for getters & setters to protect them. For our repositories, we had arrayLists, which would then take in specific information like TagID, for our storage purposes. Then we followed up by our TopicService class implements TopicRepository interface, with findBy to get our specific values, and then through the GUI we eventually show it in a visually pleasing way. This all comes together as the culmination you see at the end.

The methods in this project in the main setup() helps the main connect with the rest of the classes. findByID(int) helps system find and locate a numbered category as findByCategoryName locates by name. We use getters and setters throughout the project in order to reference mainApp.java class. The project is mostly coded in Strings as documents usually are arrange in alphabetical orientation.

In order to use the management system you must change directory inside the terminal and type ‘javac mainApp.java’ afterwards the user shall enter ‘java mainApp’ in order to compile the system. Once the system loads, due to the gui implemented on the stack, the user is shown three buttons to arrange their filed folders inside. The creation and deletion of the folder have not been installed as the program is running on a beta version for a smaller business records.