**Yaşar University**

**Spring, 2020-2021**

**SE2224 - Software System Analysis**

**Final Project Report**

|  |  |
| --- | --- |
| **Student Name:** | **Tolga Çobanoğlu** |
| **Student No:** | **17070006032** |
| **Department Name:** | **Software Engineering** |
| **Course Section No:** | **2** |

**Table of Contents (Do not change the Section Names!)**

[**1** **Introduction** 3](#_Toc71375411)

[**2** **Requirements Definition** 3](#_Toc71375412)

[**2.1** **Functional Requirements** 3](#_Toc71375413)

[**2.2** **Nonfunctional Requirements** 3](#_Toc71375414)

[**3** **Use Case Analysis** 3](#_Toc71375415)

[**3.1** **Actors** 3](#_Toc71375416)

[**3.2** **Scenarios** 3](#_Toc71375417)

[**3.3** **Use Cases** 3](#_Toc71375418)

[**3.4** **Relationships among Actors and Use Cases** 3](#_Toc71375419)

[**3.5** **Use Case Diagram** 3](#_Toc71375420)

[**4** **Behavioral Models** 3](#_Toc71375421)

[**4.1** **Sequence Diagram and its Explanation** 3](#_Toc71375422)

[**5** **Structural Models** 3](#_Toc71375423)

[**5.1** **Class Diagram and its Explanation** 3](#_Toc71375424)

[**6** **Process Modeling** 4](#_Toc71375425)

[**6.1** **Data Flow Diagram (DFD) and its Explanation** 4](#_Toc71375426)

[**7** **Final Version(s) of the Graphical User Interface(s)** 4](#_Toc71375427)

[**8** **Conclusion and Future Work** 4](#_Toc71375428)

# **Introduction**

Our project is a library automation.We designed use Java language,Intellij IDE and text file.I used these things in our project simple graphical user interface,file operations like creating and writing file,object oriented paradigms and lots of software engineering principles.System includes three main components as login,homepage and transactions.In components didn't used drag and drop,all of are is placed by .setBounds() method.Program was designed for user to add new book,remove book,display of book's features and list books in library.The program is designed to add new books, remove books, view book properties and list the books in the library according to various details after the user logs in with user authentication.

# **Requirements Definition**

Functional requirements, these are the basic features that should be present in the system.

Non-functional requirements, defines the quality attribute of the system.

## **Functional Requirements**

Add a book,delete a book and text arrangement.

## **Nonfunctional Requirements**

Display information book,book cover and number of books that user finished in determined years and list books with prefered features.

# **Use Case Analysis**

## **Actors**

User and system.

## **Scenarios**

Add Book Scenarios  
User logs in system.Select add a book from menu and enter books attribute and click add button,information message box will open which include operation is successful.And book’s datas writing to in text file.Use case ends.

Display Book Cover Scenarios

User logs in system.Select Book Cover feature from menu and enter book’s id to see cover and click enter.Frame is opened and show to cover of book.Use case ends.

List Books with Category Scenarios  
User logs in.Select list books with category from menu and enter category name and system return all books in this category.Use case ends.

## **Use Cases**

User logs in to system and there are eight possible transactions and exit button on menu.These transactions are add a book,delete a book,display the information of a book with the book title,display the image of a book cover with the given book i,display number of books user has finished in a year and list books with given features.

Log in : User enter username and system control it,valid user or not.

Add a book : Take book’s feature as an input from user and write them in a text file.

Delete a book : Take book’s id as an input from user and delete this book’s on text file.

Display Information of book : Take book’s title as an input from user and return book’s detailed informations.

Display book covers : Take book’s id as an input from user and return this id’s book cover.

Display Number of book : Take year as an input from user and return number of finished book in this year.

List Books with Author name : Take author name as an input from user and return author book title and its description.

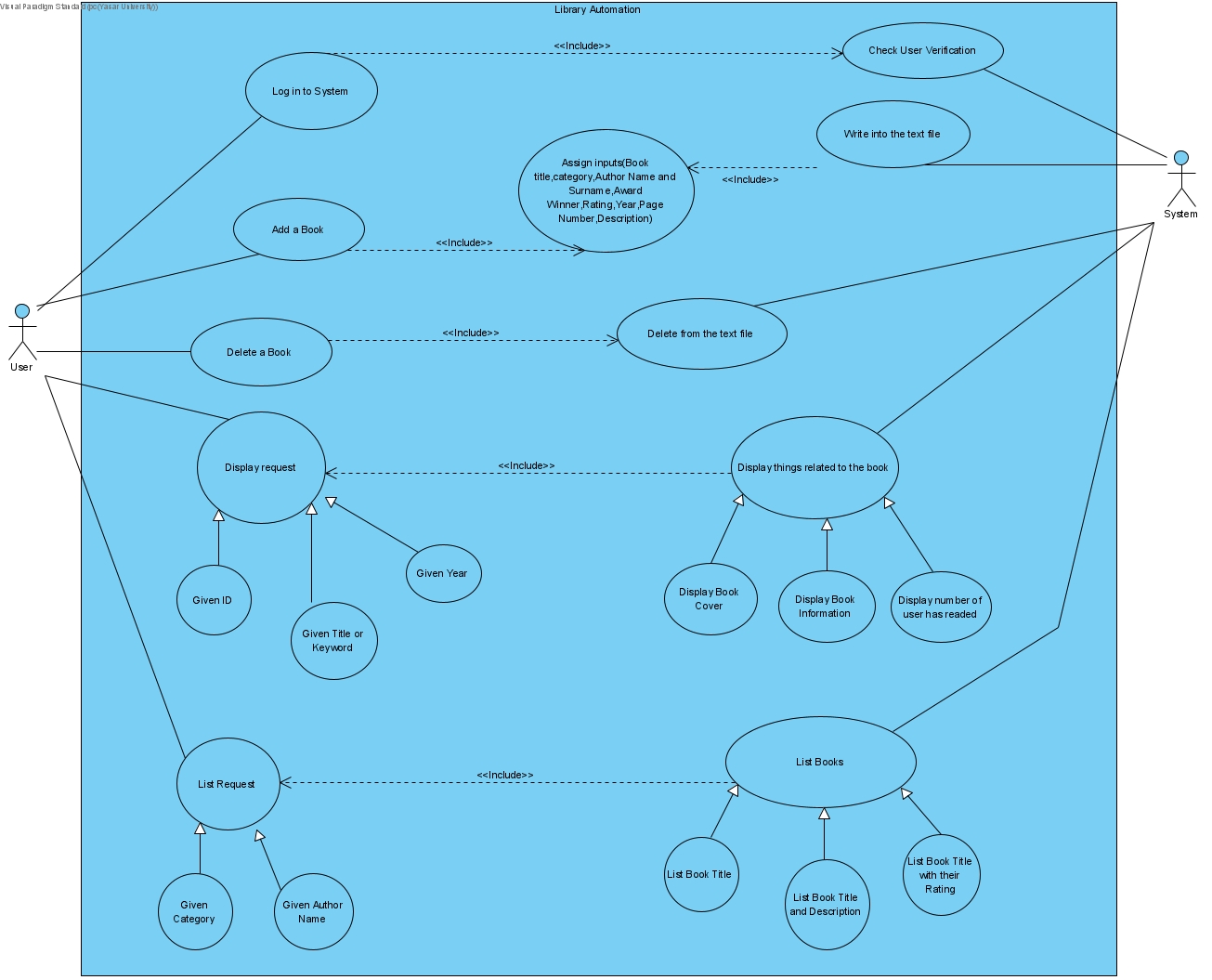
List Books with Category :Take category name as an input from user and return all books in this category.

List Books with Category and sorted with Rating :Take category name as an input from user and return all books in this category but sorted them with user rating.

## **Relationships among Actors and Use Cases**

Every use case scenarios are for user transactions. Relationships among an actor and use case is association. There are two relationship between them such as include and extend.

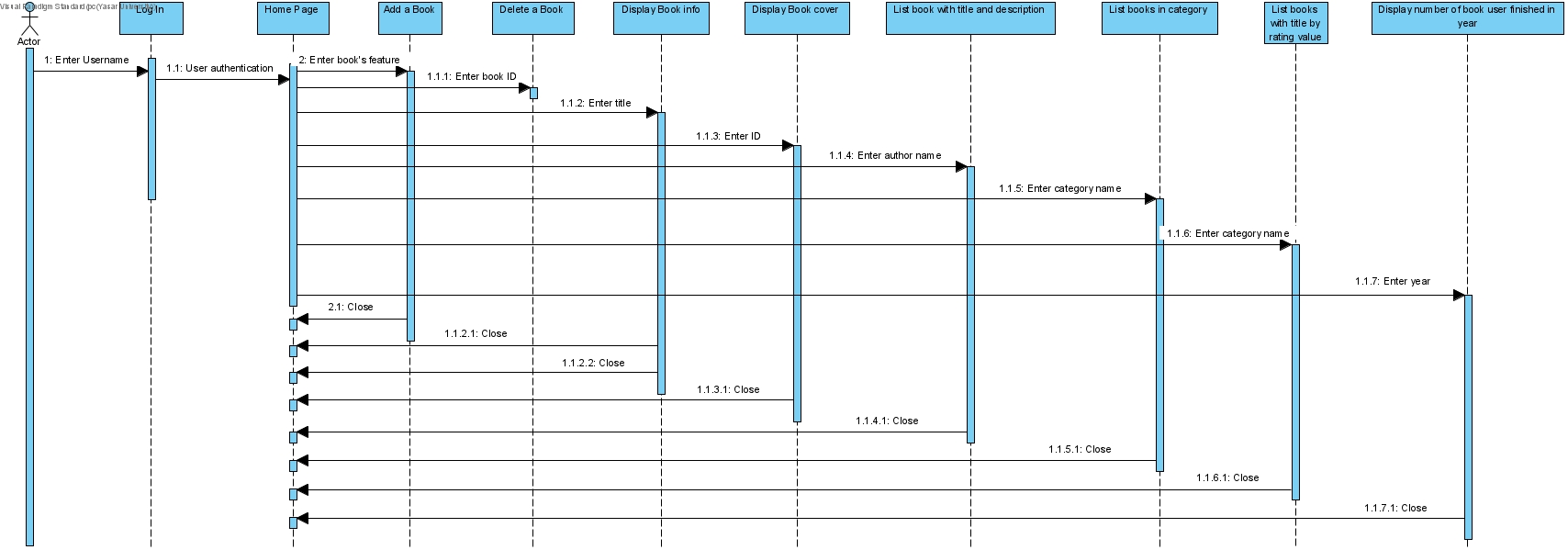
## **Use Case Diagram**



# **Behavioral Models**

## **Sequence Diagram and its Explanation**

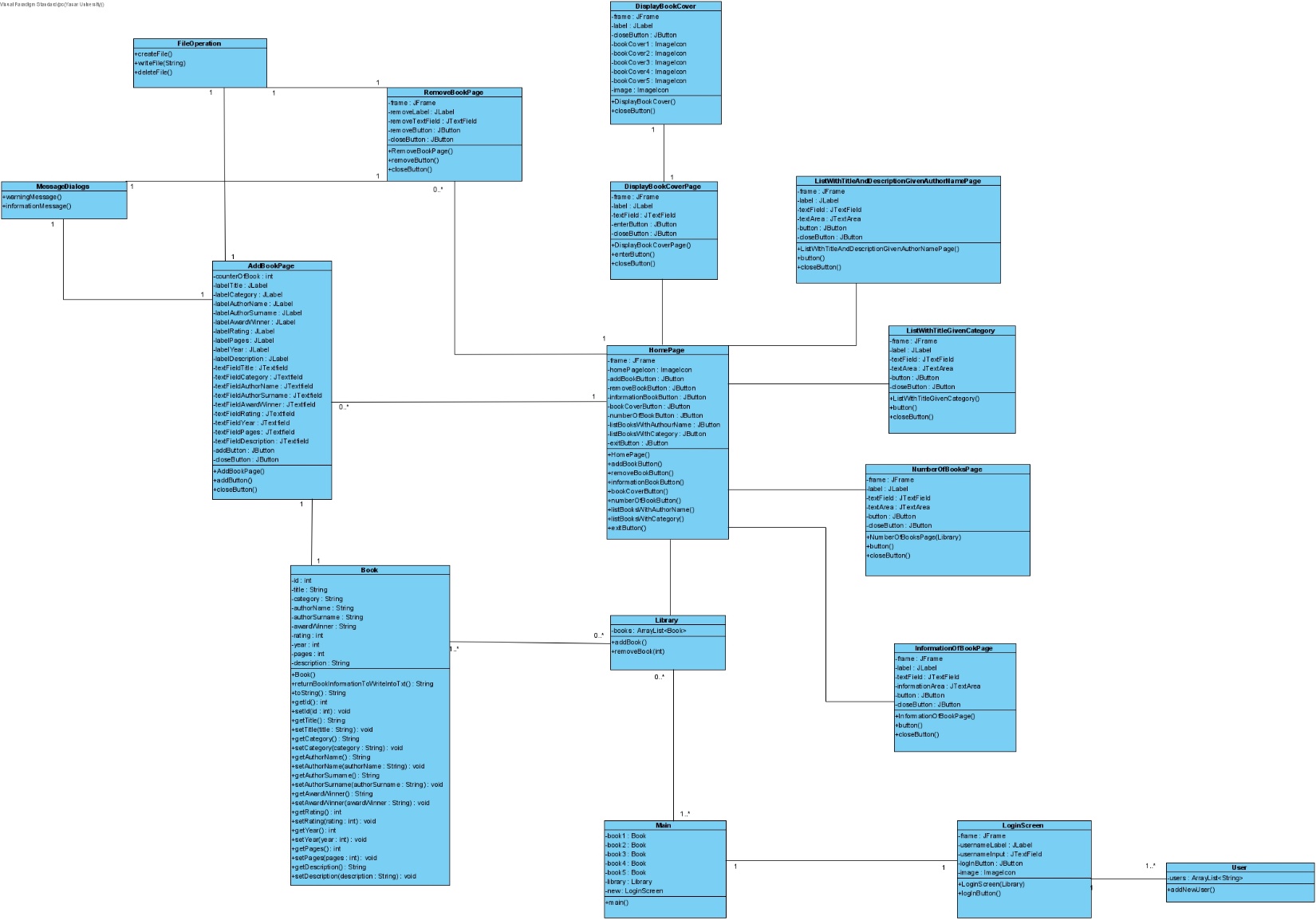
User logs in system.Homepage is open.User select any transaction or exit.Then, according to the selected operation, the object of that operation is formed and it performs the operation.



# **Structural Models**

## **Class Diagram and its Explanation**

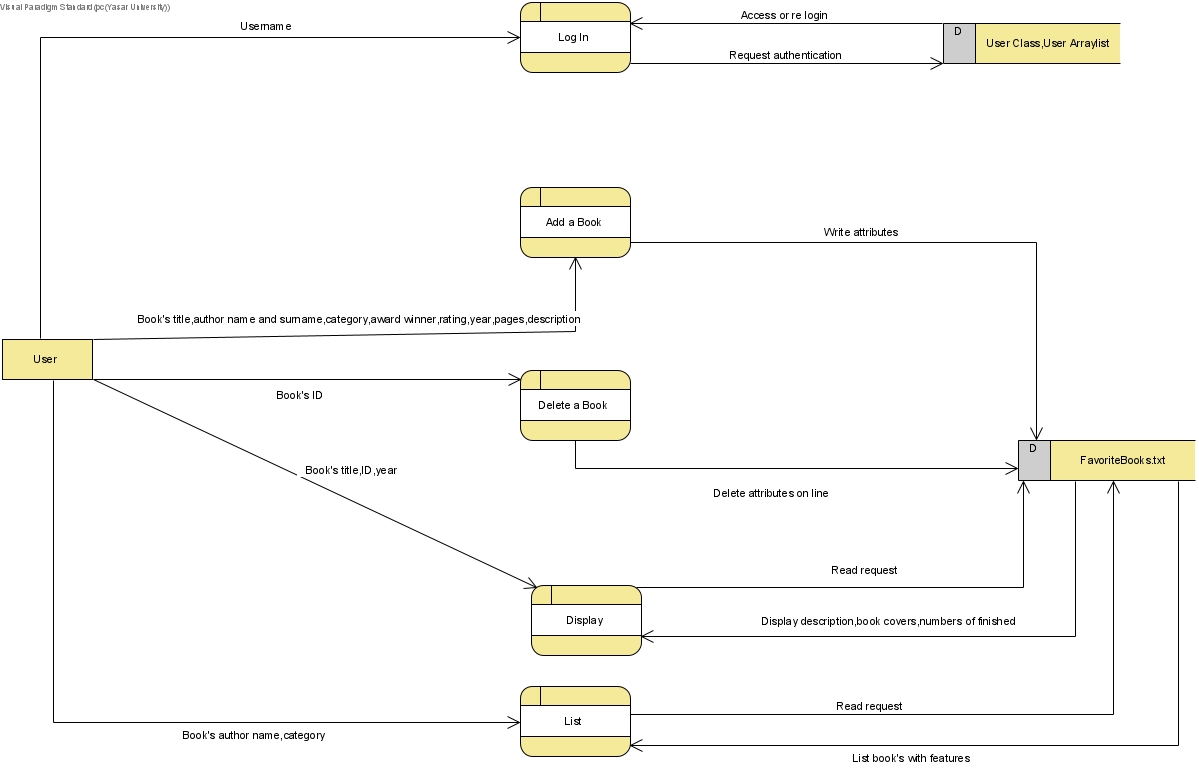
Explain class by one by :  
  
Book : Store book’s data in a object.  
Library : Store one more book’s object in a single structure in Library’s books arraylist.  
User : Store one more user’s username in a single structure in User’s users arraylist.  
LoginScreen:GUI class for login screen and starting automation from here.  
HomePage:GUI class and menu page  
AddBookPage: GUI class for Add a book feature.  
RemoveBookPage: GUI class for Delete a book feature.  
InformationOfBookPage : GUI class for display the information of a book feature.  
DisplayBookCover : GUI class for store an ImageIcon and includes showing images method.  
DisplayBookCoverPage: GUI class for ask to user which book cover you want to display.  
NumberOfBooksPage : GUI class for display the number of books that the user has finished reading in the given year.  
ListWithTitleAndDescriptionGivenAuthorNamePage : GUI class for list the books (book title and short description) with respect to the given author name.  
ListWithTitleGivenCategory : GUI class for list the books (book title) with respect to the given category.  
FileOperation : File operation class to create,write and delete file.  
MessageDialogs : Includes updated version of ready JOptionPane.showMessageDialog method.  
Main : To run system.



# **Process Modeling**

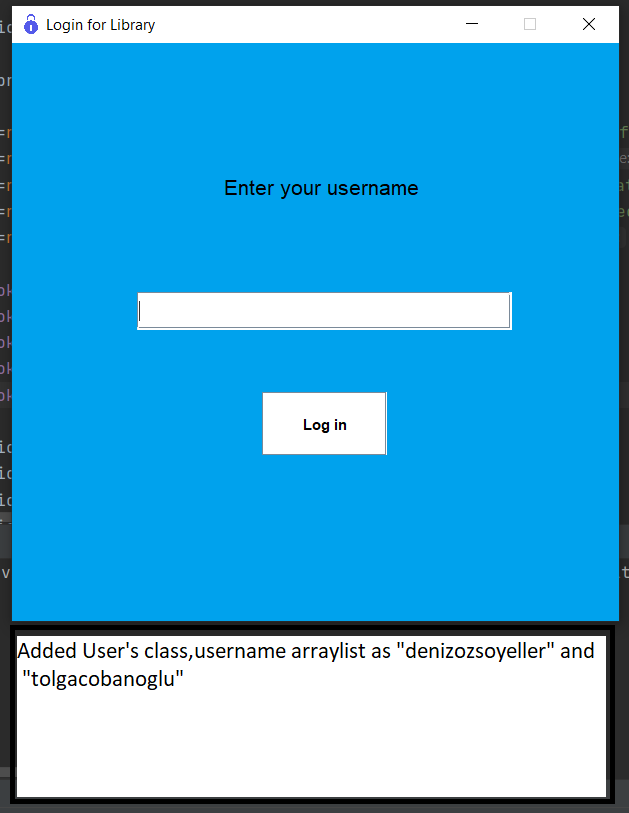
## **Data Flow Diagram (DFD) and its Explanation**

User enters username to login, login process controls the input valid or invalid and access or re log in.  
User enters book’s feature as input to add a book process and book process writes them into text file.  
User enters book’s ID as input to delete a book process and process delete things in text file.  
User enters book’s title and category to display process,display process send read request after from text file return and displayfeature.  
User enters book’s author name and category to list process,the process send read request from text file and list books.

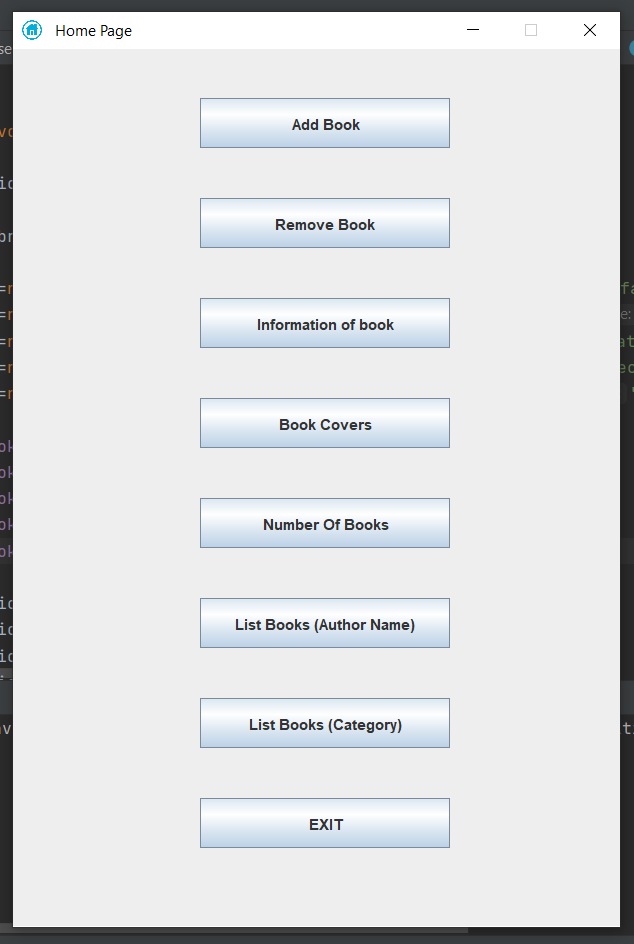


# **Final Version(s) of the Graphical User Interface(s)**

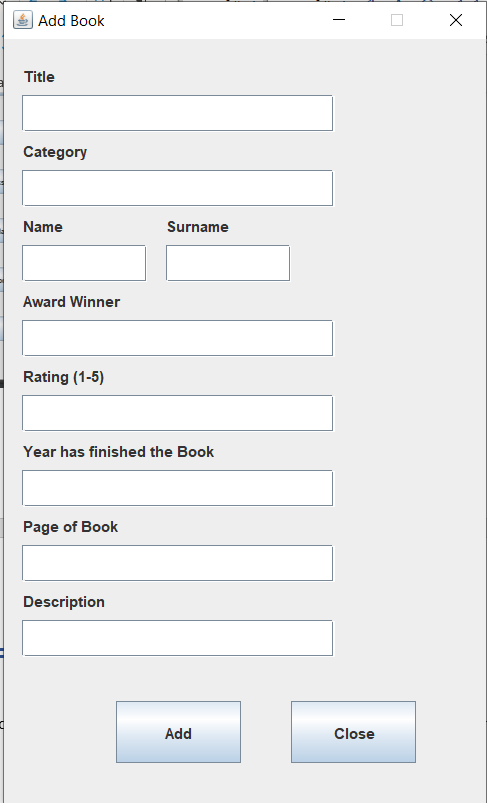
LOGIN SCREEN



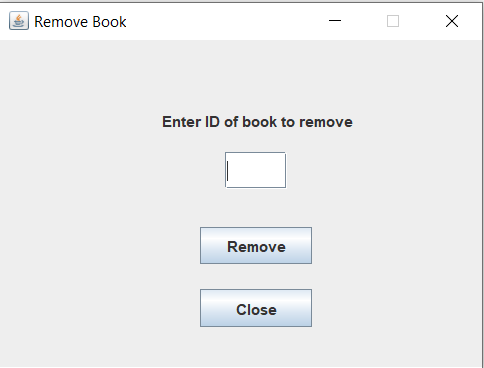
HOMEPAGE or MENU



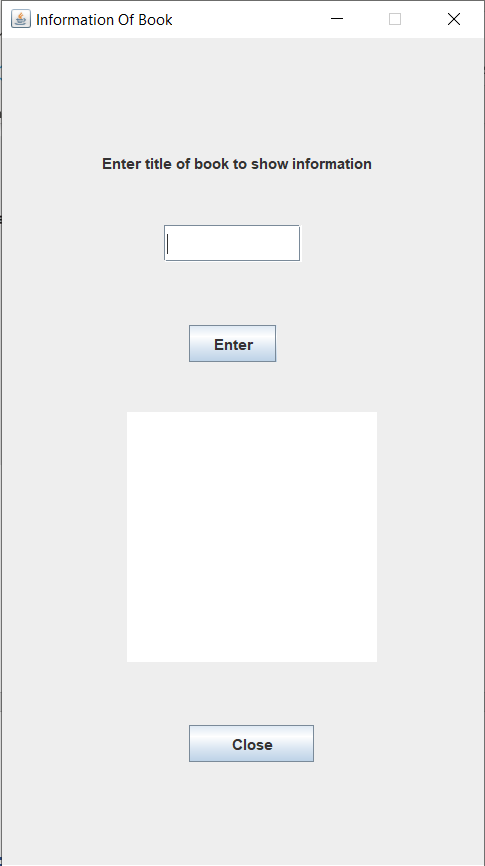
ADD BOOK FEATURE



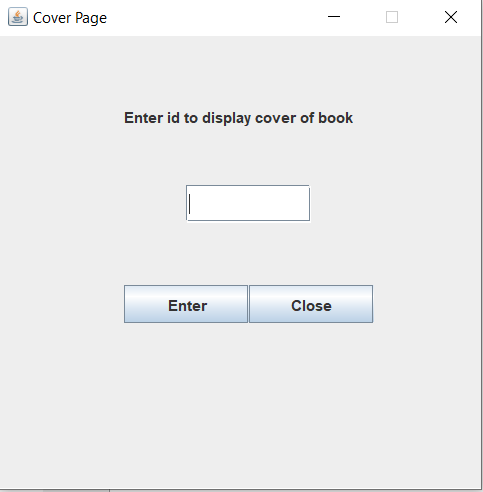
DELETE BOOK FEATURE



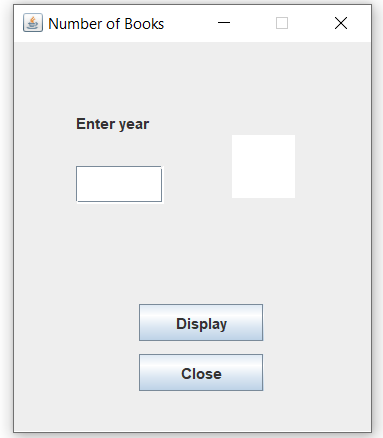
INFORMATION OF BOOK FEATURE



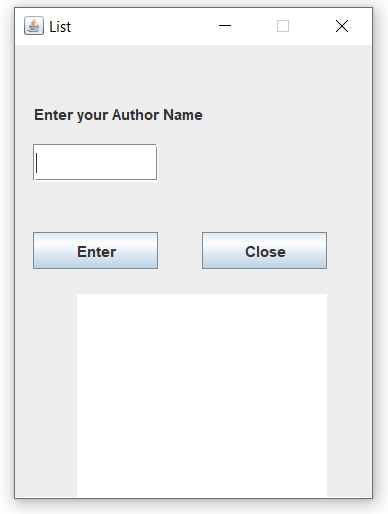
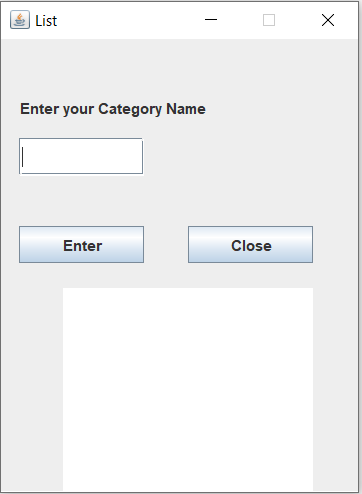
DISPLAY BOOK COVER FEATURE



USER HAS FINISHED READING IN YEAR FEATURE



LIST FEATURES

# **Conclusion and Future Work**

As a result, I wanted to design all the operations and the layout according to the object-oriented programming paradigms. In order to meet the innovations that will occur in the system with the least cost, I functionalized all the operations and took into account the reusability of the codes. I aimed to build a more dynamic code structure by not using the drag and drop method. If I have to generalize, I realized how data transfer works in complex structures, how to establish relationships between class structures and the importance of agile before writing code.

The features that I want to develop for my project are to use a database instead of a text file, to add a reading analysis on an annual or monthly basis, for example, you read 30 pages of books this month, you read 3 books a year. With the data I will get from the internet using the API, daily random book recommendation and displaying it with the book title and description. When you add a new book, the ability to share it on social media accounts, such as location reporting in the Swarm application or sharing the music you listen to on Spotify in the Instagram application.