

EEE 210 Software Engineering

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Project Report**

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

Performed by Next Java Group

Shymyrbay Ayan

Azhigulov Dias

Toleubay Yeldar

Submitted on 22.04.2018

Table of Contents

**Introduction3**

**Objective3**

**Problem Statement3**

**Program Development3**

**Results4**

**Conclusion6**

**References…………………………………………………………..………………...7**

**Introduction**

In this project work, the team created their own Library Management System. The library has a user base and a list of books accessed from .xls file. Thus, only certified persons can access the library. The system, as its name suggests, allows users to search and sort books and consequently issue them. Every book has 7 fields: author, title, year, ISBN, publisher, LLC and finally stock. One can also track users’ accounts to identify which book was issued by them. In addition, the library features ability to obtain e-books’ URLs. It should be noticed that, throughout the project, the team implemented various fundamental Object-Oriented Programming concepts such as Inheritance, Polymorphism and Aggregation.

**Objectives**

* To design an interactive user interface that is attractive and user-friendly;
* To apply OOP concepts in practice;
* To develop new skills related to Software Engineering;

**Problem Statement**

As can be seen from the description above, the project is quite complex. Therefore, the team first identified key issues of the project such as how to access a .xls file and thus read/write to it. Also, another major issue is to build a global search algorithm which would find an appropriate book according to a text input in search bar. Consequently, one can notice that this algorithm has to deal with any types of inputs (ISBN, author name, title etc.). Furthermore, there has to be a sorting algorithm which implements sorting of books according to any of the 6 fields (stock is omitted). Finally, users should somehow issue a desired book and subsequently be able to return them which leads to a new set of problems such as user-database interaction.

**Program Development**

First of all, it should be mentioned that the entire project was performed in JavaFX Graphical User Interface (GUI) Development Platform. In addition, another library developed by Google called GSON is utilized due to its simplicity and functions it provides. Also, it facilitates the interactions with complex objects. Whereas JavaFX takes the burden of design aspects by supplying various functionalities and utilities to facilitate designing process.

Thus, initially, the team created DataAccess class which reads from the excel file and consequently write all the data in there to ArrayList of Books. Note that Book is the class from which various types of books are created according to their data fields in the excel file. Also, books are created sequentially as the DataAccess class reads more and more books. From here, one can notice Aggregation between DataAccess class and list of books. In this way, DataAccess “has” that list and there is indeed aggregational relationship. Additionally, there is a parent class called Controller which is common to all types of controllers that will be used throughout the project. Furthermore, as the project develops there will be another parent class called View which is responsible for creating new scenes or windows. From these lines one can notice that the program possesses Inheritance properties. Because certain child classes override some of the parent classes’ methods, Polymorphism also takes place in the program. Therefore, one can clearly see how the OOP concepts facilitate the coding aspects of the project.

Another topic to consider is which technique to use for sorting and searching. After days of discussion the team concluded that it is better to use QuickSort for sorting purposes and Linear search for global searching needs. A person can argue that linear search is very slow, and it is true, however, because the number of books in the library is not great, it is still quite fast. Despite this, it should be emphasized that Binary Search was used for integer data fields such as ISBN in order to allocate a particular book in the list. When it comes to sorting, QuickSort was the most suitable candidate as it has low time complexity (O(n\*log(n)). Also, it is easy to implement. It works as follows:



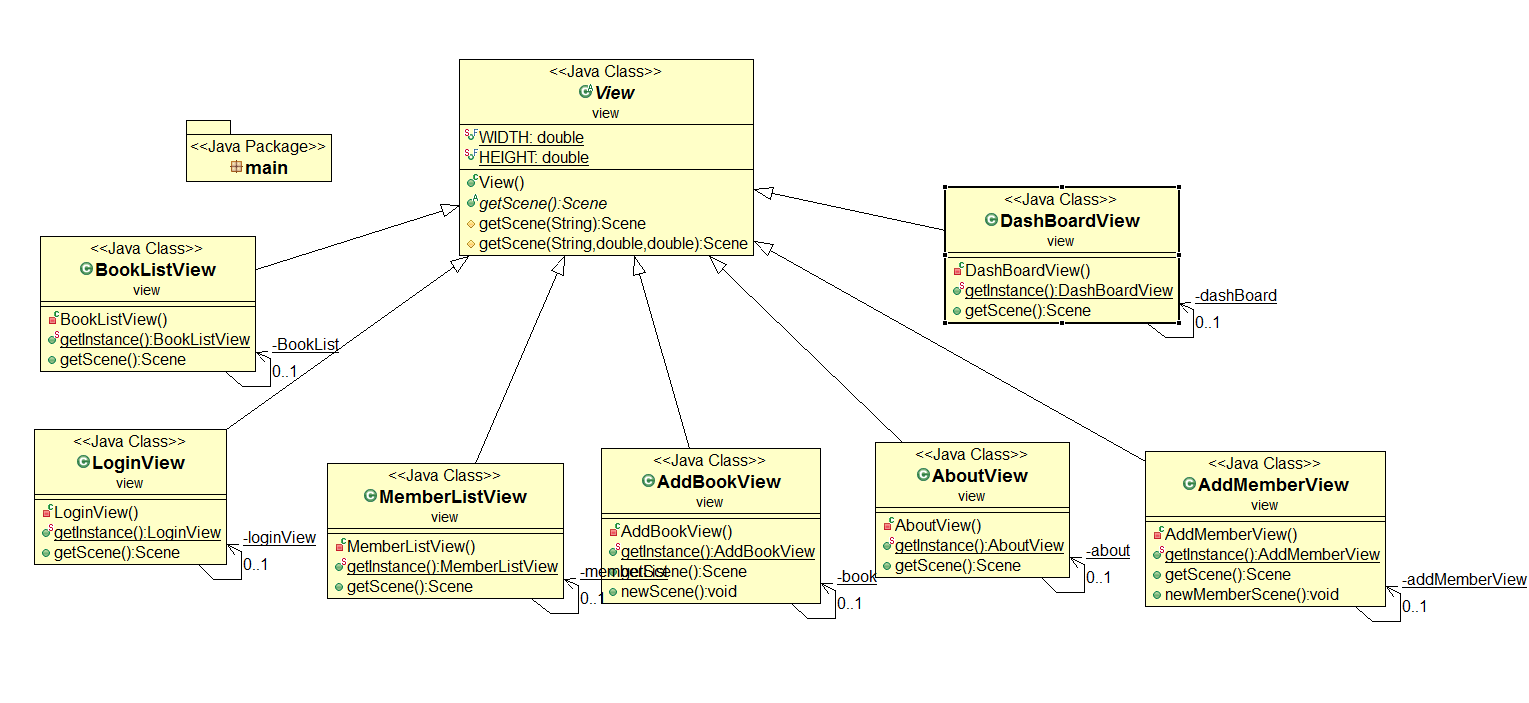
**Figure 1.** QuickSort Algorithm Working Principle [1]

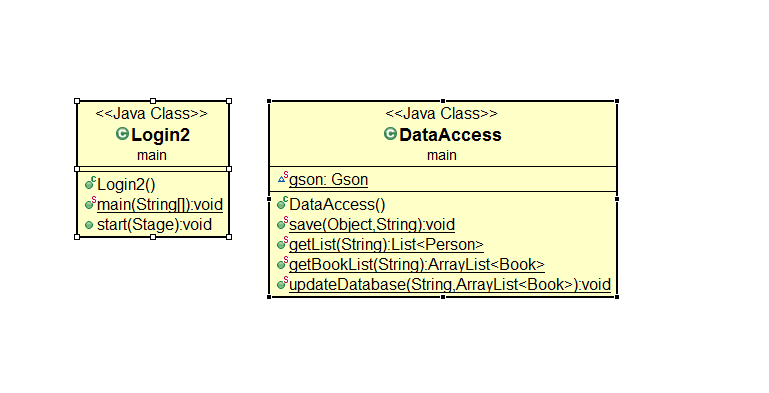
Exception handling is also a crucial part of any application. Hence it is of great importance to handle runtime errors and especially important to indicate the source of these exceptions. Consequently, the team applied exception handling to most of the methods in the project to quickly fix them.

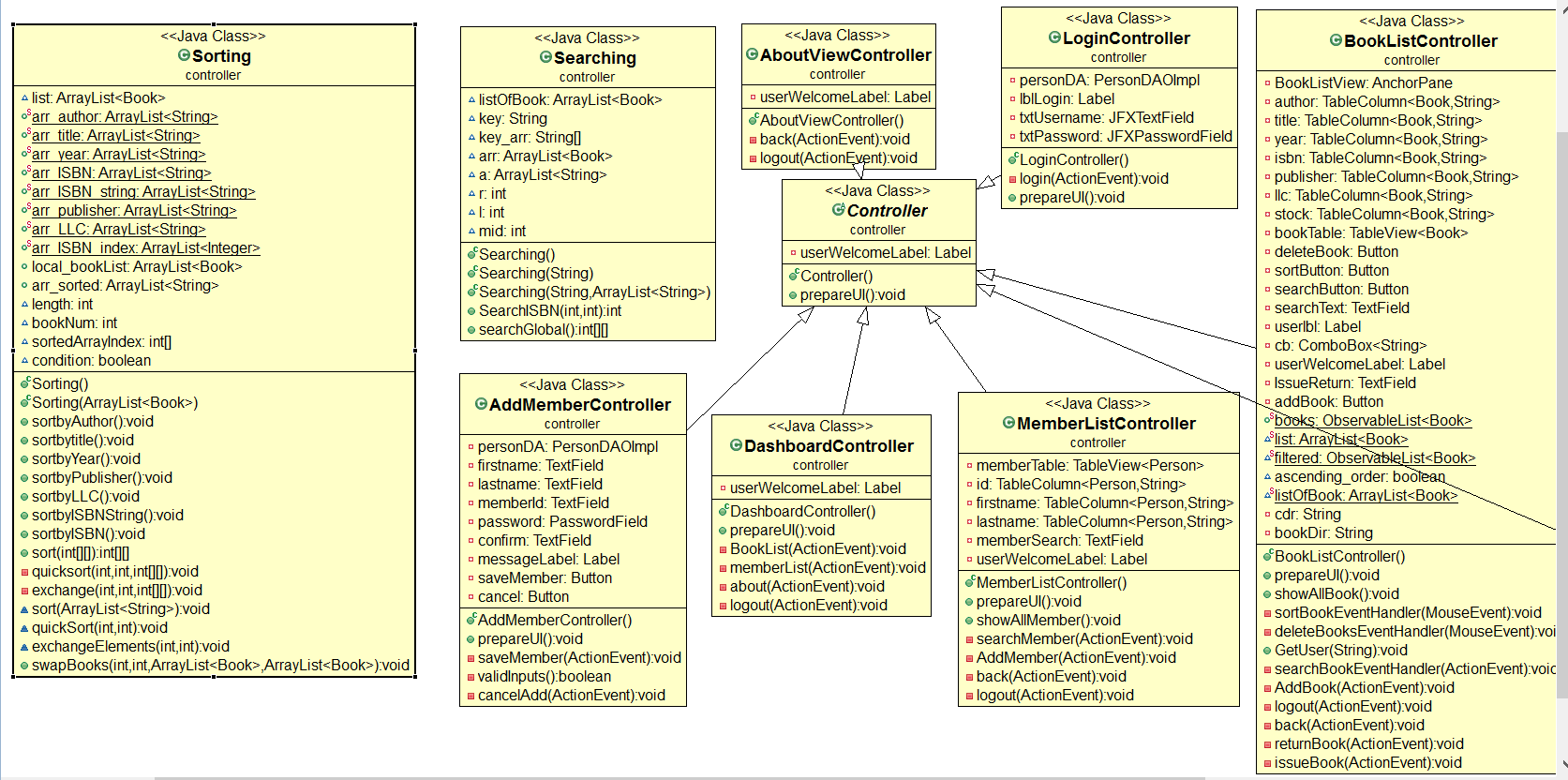
**Results**

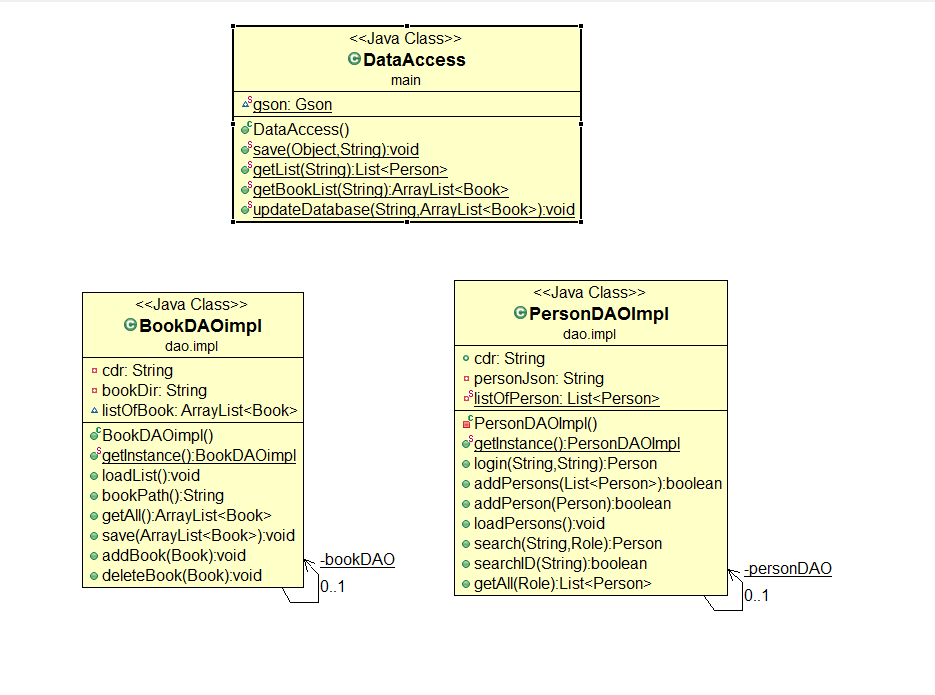
Following the implementation of the framework of the application, we turned our attention into graphical user interface. Consequently, our focus was on how to make the user experience as intuitive as possible. The overall final design of the Login, User and Admin windows is as follows:

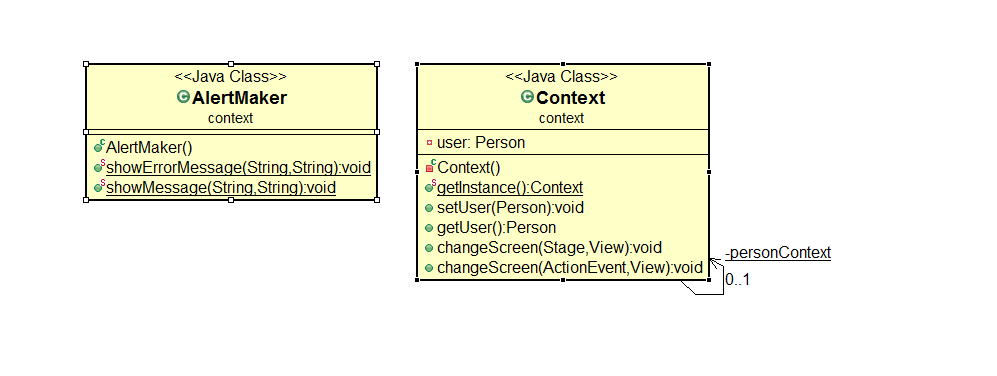
Below a set of UML diagrams can facilitate understanding of our coding structure and what constitutes the application.











**Conclusion**

To sum up, the core objectives were successfully achieved during the realization of the project. Thus, students developed a new set of skills related to Software Engineering. Additionally, the project tested the team’s ability to work on complex problems collaboratively. Thus, initially, the team was aiming to build a simple Library Management System, however, as the program progressed, additional features were added in order to make user experience as intuitive as possible. It should be emphasized that the library meets all the requirements and is based on Linear global search and Quick Sort algorithms. Also, the application is entirely developed on JavaFX thus transforming it to modern-looking app. Overall, the project facilitated students’ understanding of how the object-oriented programming operates and helps to improve coding experience.

**References**

[1]GeeksforGeeks. (2018). QuickSort - GeeksforGeeks. [online] Available at: https://www.geeksforgeeks.org/quick-sort/ [Accessed 20 Apr. 2018].