

**CST1150 Portfolio**

**Assignment 4 - Report**

**2023 Feb Intake**

**Arusha Ramessur**

**M00940320**

**Project: Simple Library System**

1. Programming concepts and structures used:

* The program defines a class called ‘LibrarySystem’ which extends the JFrame class, representing the main graphical user interface for the library system. Then 2 other classes which are the ‘Book’ class and ‘Subsciber’ class.
* The program imports various packages using the import keyword to access classes and functionalities from those packages, such as javax.swing, java.io, java.util and java.time.LocalDate.
* The program uses ArrayList<Book> and Vector<Subscriber> to store and manage collections of book and subscriber objects, respectively. These classes provide dynamic arrays with additional methods for manipulation.
* The program defines several methods, such as AddBook(), AddSubscriber(), BorrowBook(), ReturnBook(), SearchBooks(), SearchAuthor(), SearchSubscribers(), SortBooks(), SortSubscribersID(), SortSubscribersName(), SaveBooksToFile(), SaveSubscribersToFile(), DisplayBooksFromFile(), DisplaySubscribersFromFile(), createButton(), and HandleButtonClick(). These methods do specific tasks in the library system. Then we have getter and setters methods in both Book and Subscriber class which allow access to and modification of the attributes.
* The program uses try-catch statements to handle potential exceptions that may occur during the execution program. For example, NumberFormatException is caught when parsing integers from user input, and IOException is caught when reading or writing files.
* The program also uses the javax.swing package to create a graphical user interface for the library system. It includes buttons, text input dialogs, and message dialogs provided by JOptionPane to interact with the user.
* The program includes methods to read and write data to files using classes such as PrintWriter, FileWriter, BufferedReader, and FileReader. It allows the system to write, read and save records to files and display data from files.
* The program contain if-else statements and loops (for, while) to control the flow of execution and perform operations based on conditions.

1. Provide clear details of what you started off to do in your project:

At the beginning of assignment 2, I created a library system program with a simple graphical user interface using JOption Pane. The initial steps involved creating the LibrarySystem class which serves as the main class. Then the Book and Subscriber classes were created to represent book and subscriber objects respectively. The Book class and Subscriber class were designed with appropriate attributes and methods. These attributes included BookID, BookName, Author, PublicationDate, Qty, Section for books, and SubID, Name, Surname, ContactNumber for subscribers. Methods were defined to perform operations such as adding books and subscribers, searching, borrowing, and returning books, as well as displaying book and subscriber details.

Then I continued to build on that for assignment 4 by adding various functionalities such as searching for books and subscribers, sorting books and subscribers, saving records to files, and displaying records from files. Next, I added a GUI interface using Java Swing to create the Menu of the library system. It consisted of multiple buttons that were linked to the methods so that users were able to access them with just a click. Lastly, I added try and catch statement to handle exceptions in the program.

1. The problems faced in implementing as you progressed and how the problems were resolved:

Initially, there was a conflict between using a simple array and an ArrayList. To resolve this, the array object was converted to an ArrayList object to pass it to the ArrayList.

The book name search, author search, and surname search were case-sensitive, causing the program not to find books and subscribers when using or not using capital letters. To overcome this, the equalsIgnoreCase() method was used to perform case-insensitive comparisons.

Another problem I got was, the records written to the file were not properly formatted and separated. To address this, a StringBuilder was used to construct the complete record as a string, including commas or other delimiters as needed, and then written to the file.

Then the program was overwriting the existing records in the files when saving records. To append the records, the FileWriter constructor was modified to include true as the second argument, enabling append mode.

It took some trial and error adjustments to find the appropriate settings for buttons and spacing to achieve the desired scale and layout of the GUI interface.

1. Reflection on the project work done and how you could improve it. What other concepts could you add to enhance your project:

Overall, the project has been successfully implemented, fulfilling the initial goals of creating a library system program with a GUI. The program allows users to perform various library-related tasks such as adding books and subscribers, searching for records, borrowing and returning books, searching and sorting records, and writing, reading and saving/displaying records from files. The GUI interface provides a user-friendly experience, and the program handles user input validation and file operations effectively.

Other concepts that could be implemented: I could incorporate more GUI components beyond the main menu of the library system to interact with more functionalities and enhance the visual appeal and usability of the GUI by improving the layout, color scheme, and font of the program. Additionally, I could implement other features that allow users to retrieve existing records, enabling them to modify, update or delete books and subscribers details as needed.

Provide a flowchart to show the structure and navigation of your project.

