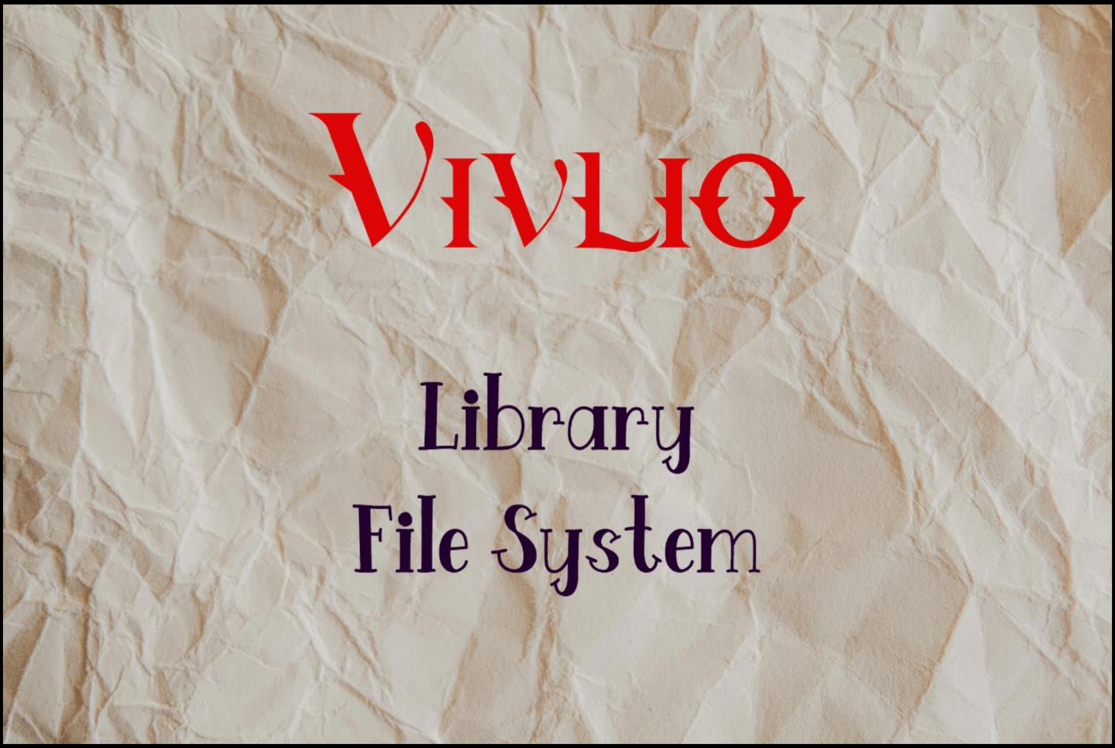
IT150 PROJECT FINAL REPORT



Vivlio

Library File System

Members:

Adarsh Kishore (201IT103)

Harish Gumnur (201IT223)

Poorna Hegde (201IT240)

Ranjana Kambhammettu (201IT247)

Abstract

This project is a library file system for a fictional library '*Vivlio*'. This system handles both the Customer/visitor and Administrator sections of the library. There is a welcome page that places authorization requests for Admin and Customer accounts. A Customer can do the following: borrowing a book, returning a book, buying a book, viewing history of bought and returned books and browsing the library's catalogue. An Administrator can do the following: Issuing a book, selling a book, revenue modification, modifying the catalogue, deleting a Customer. The state is stored across runs making it a reusable program. We have tried to simulate the real world to the best of our ability throughout our project.

Problem Definition

Functionality implemented:

As mentioned in our abstract, our program can be divided into two parts: Customer and Administration.

Customer features:

1. A Customer will be able to browse through the catalogue of books. They will be able to search based on title, author, price, genre, ISBN and availability.
2. A Customer can borrow books from the library. A Customer will be able to borrow a maximum of 5 books at a time. The Customer will search through the books first and then select a book for borrowing.
3. A Customer will return the borrowed books. If the book has not been returned within 14 days, then a fine will be imposed.
4. A Customer can buy books from the library.
5. A Customer can read through a user manual at any time.
6. A Customer can view the history of returned and bought books.
7. A Customer can view the list of books currently borrowed.

Admin features:

1. An Admin upon logging in can see the library details like revenue, number of books on rent and number of books sold.
2. An Admin can see the list of all Customers and all books.
3. An Admin can see the complete details of any books.
4. An Admin can add, delete or modify a book.
5. An Admin can see the details of any Customer.
6. An Admin can remove a Customer's account.

Supplementary features:

1. There will be an option of logging in/out for both Admin and Customers to protect their accounts. (Authorization is necessary to access details)
2. For buying and paying fine, there will be a payment system that takes the credit/debit card details with suitable checking and updates the revenue.
3. To store all the details there will be .dat files.

Methodology

1. How did we go about solving the problem?

1. We have implemented 10 classes for the system:
   - **Person**: This is an abstract class with the functionality for login credentials of any person in the library. The classes Customer and Admin inherit from this class.
   - **Admin**: This is a date type that implements an Administrator of the library who can issue and sell books. It also keeps track of all the Customers and book in the library, and also the revenue generated.
   - **Customer**: This is a data type that implement any visitor in the library that can borrow, buy or return a book. It also contains information about a Customer, like a list of all the books that they has bought and borrowed. It also has an object of a Transaction class.
   - **Transaction**: This deals with the payment, fines and dates related to a Customer's transactions of borrowing, buying and returning.
   - **Book**: This implements a real-life book with all its properties like title, authors, genres, price and ISBN. It also stores its own list of Customers who have bought and borrowed the book. The Customer and Admin deal with an object of this data type in their functions.
   - **Catalogue**: This implements a catalogue for the Customer to browse through and filter all the books according to his/her own choice. It also implements the storage of all the books in the library by using data files.
   - **AuthService**: This is a class that authorizes the Login and Signup of a Customer and also the Login of the Admin in the library by creating an authentication object. It also used to differentiate between a Customer and Admin while logging in.
   - **AdminOptions**: This displays all the functions that an Admin can choose to do when he or she logs into their own account. It also regularly displays all the library details, for example, the number of books on rent, the number of books sold and the revenue generated.
   - **CustomerOptions**: This implements the functions to be display all the options that a Customer can choose to do when they logs into their own account by giving their username and password.
   - **Main**: This contains the main() function for running the application by first displaying a welcome page for login or sign up.

2. We have used **.dat files** to store all the information. A dat file is a binary file, and hence objects of a date type can be written into it and read from it. The classes Person, Book and Transaction implement the Serializable interface. This lets you write objects of these types into the file by converting the objects into a stream of bytes.

3. We have used **Java** OOP language, **git** to collaborate amongst ourselves, **GitHub** to host all of our code and collab tools, and **Markdown** to write out instructions in the ReadMe file.
4. Used OOP concepts implemented – **Inheritance** (Customer and Admin inherit the Person class), **Encapsulation**, **Abstraction**, Runtime and Compile time **Polymorphism** (Overloading and Inheritance).

2. How did we solve each problem (mentioned in the previous section: Problem Definition)

1. **Authorization**: The AuthService object differentiates between a Customer and an Admin while logging in by checking the usernames. If the username contains a digit, it's a Customer. If not, it's an Admin. When a Customer signs up, if they hasn't added any digits in their username, then their age is appended by default.
2. **Selecting a Book**: The Customer has many ways to select a book. They can enter their preferred title, author(s), genre(s), ISBN(s), and maximum price. This request is taken by the Catalogue, which calls its filter functions. The booksList ArrayList is traversed through, and if the conditions match, the filtered books are returned as an ArrayList and displayed to the Customer. If the inputs don't match with any book in the library, an appropriate message is displayed, and the Customer can browse again.
3. **Borrow a Book**: The Customer requests to borrow a book after selecting it. If the copies of the book are available, then it checks if the Customer hasn't exceeded the maximum borrow limit. If successful, it passes control to the Admin who is a window between the Customer and transaction. The transaction is initiated by the Admin, and if successful, the Customer is added to the list of borrowers for that book. The book is also added to the borrowed books list of Customers and the number of copies of that book in the library is decreased. Thus, the transaction of borrowing is complete.
4. **Return a Book**: The Customer requests to return a book. Only the books that they has borrowed are displayed for them to choose. The control now passes to the Admin. The Admin initiates the return book transaction. It then adds back the book to the library database by increasing its number of copies. It also checks the date it had been borrowed. and calculates fine if it has been over 14 days. The fine has to be paid through card payment. The control now passes to Transaction for payment. It then removes the Customer from the list of borrowers for that book, and it also removes the book from the list of borrowed books of the Customer. Thus, the transaction of returning is complete.
5. **Buy a Book**: The Customer requests to buy a book after selecting it. If the copies of the book are available, it passes control to the Admin who is a window between Customer and transaction. The transaction is initiated by the Admin, and after entering the proper card details, if transaction is successful, the Customer is added to the list of buyers for that book. The book is also added to the bought books list of Customer and the number of

copies of that book in the library is decreased Thus, the transaction of buying is complete.

6. **User Manual**: If the Customer wants to read the instructions, they can select the User Manual option. A function is called, and it displays all the instructions on the screen.
7. **View History**: History for every Customer has been implemented by using a HashMap. It is an instance variable of the Customer object. The book is the key, and the value stored is an ArrayList of dates (for borrowing: borrow and return date, for buying: bought date). It's updated every time the Customer returns a book or buys a book. If there's only one date in an entry's value, it implies that the book has been bought. Based on this, the history details are displayed.
8. **View list of currently borrowed books**: The Customer can also view a list of the books borrowed currently, and also the last date to return them (by adding 14 days to the borrow date, which is got from the transaction object of that Customer). These books are contained in the borrowedList instance variable of the Customer object.

Admin Specific functions:

9. **List of all Customers:** When the Admin requests to do this, a list of all the Customers is got from the Customer class and then displaying them on the screen with all their details.
10. **List of all Books:** The Admin can view a list of all the books in the library by requesting the Catalogue class. This returns the list, which is then displayed. The number of copies of each book left is also displayed.
11. **Complete details of a Book:** The Admin enters the name of the book. This book is then searched for in the booksList of the Catalogue class. If it exists, the book's characteristic and transaction details are displayed. If there's no book, an appropriate message is given.
12. **Modify Details/Add Book**: The Admin enters the characteristic details of the book (title, authors, genres, price and ISBN). The entered ISBNs are first checked to be valid. A new Book object is created. If the ISBN entered already exists, then the book's details are modified. If it's a new ISBN, then the new book is added by calling the saveBooks(ArrayList<Book> books) function of Catalogue, which writes into the dat file. This is implemented by keeping the ISBN as a final variable, and hence it can't be modified again and is used as a unique key.
13. **Removing a Book**: The Admin requests to remove a book. There's a check for all the currently borrowed books in the library by checking Customers' lists of borrowed books. If the entered book matches any book in these lists, then it's not removed. If there's no match, it's removed successfully by accessing the booksList in Catalogue, and calling the saveBooks() function, which writes into the dat file.
14. **Removing a Customer**: After the Admin's request If the entered Customer's *isReturned* ArrayList (in its transaction instance variable) has an element that is *false*, the Customer isn't removed as that implies that they hasn't returned a book. If all the elements are *true*, the account is removed successfully by

the Admin, by accessing the CustomerList in Customer, and removing it. The saveCustomer() function is called, which writes into the dat file.

15. **View the History of a Customer**: The Admin enters a Customer's username. If this account exists (checked by using the list in Customer), showHistory() function is called from CustomerOptions, which was also used while displaying history to a Customer and it's displayed. If there's no account, an appropriate message is displayed.

UML Diagrams:

Google Drive Link:
https://drive.google.com/drive/folders/1it684RHwyYmf1CoWOKHq2b8Sww2lehZK?usp=sharing

GitHub Repository Link:

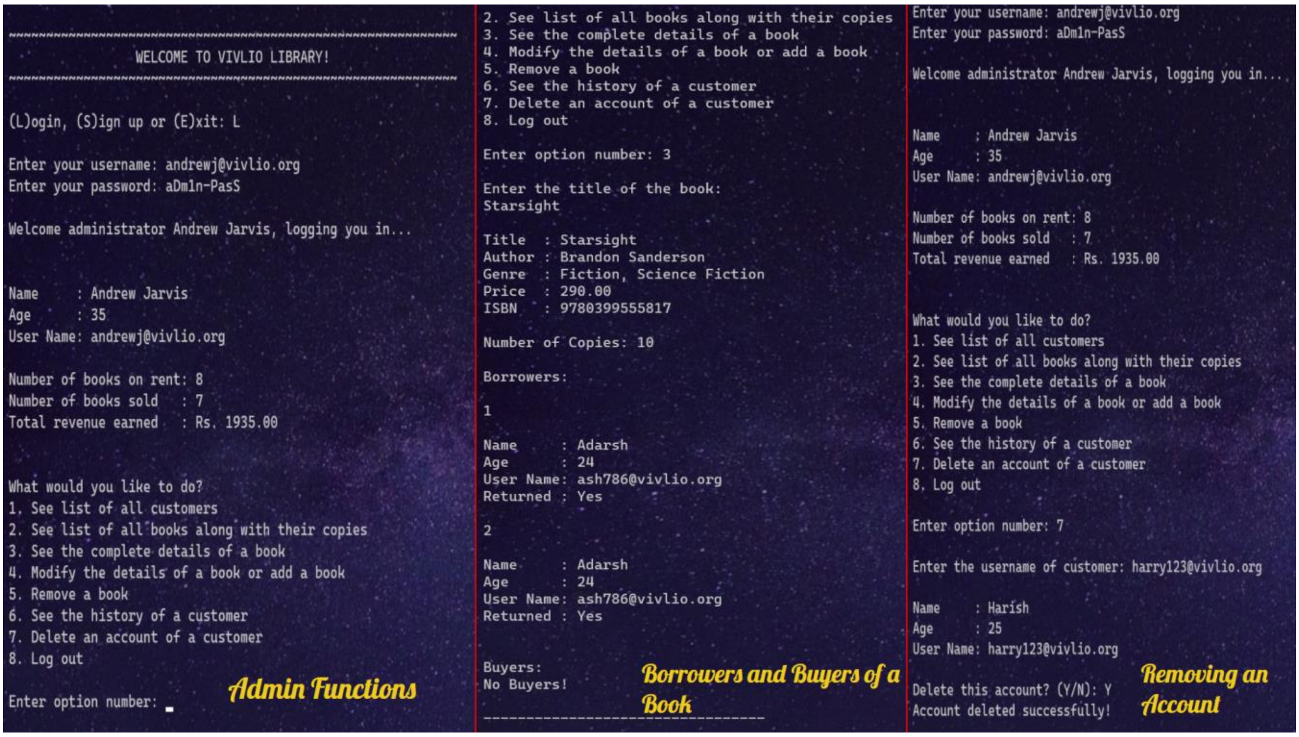
https://github.com/ashy-boy786/IT-Project

Results

The project '*Vivlio*' works successfully. All the features were tested and worked on Ubuntu command line and Windows PowerShell with zero warnings.

When the program in run, a welcome page is displayed with 3 options: (L)ogin, (S)ign up and (E)xit. When you enter 'S', the program allows you to sign up, collects a one-time membership fee and registers the user. A list of instructions and the login credentials are displayed. Upon logging in as a Customer, a menu of options is displayed to the Customer. The menu has the following options: View Catalogue, Borrow A Book, Return A Book, Buy a Book, View List of Books You've Currently Borrowed, View History, User Manual and Log Out. These options work as explained in Methodology.

When the user logs in as the Admin, the number of books rented, sold and total revenue is displayed. A menu is also displayed with the following options: See list of all Customers, See list of all books along with their copies, See the complete details of a book, Modify the details of a book or add a book, Remove a book, See the history of a Customer, Delete an account of a Customer and Log out. These options work as explained in Methodology.



```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                WELCOME TO VIVLIO LIBRARY!
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

(L)ogin, (S)ign up or (E)xit: L

Enter your username: andrewj@vivlio.org
Enter your password: aDm1n-PasS

Welcome administrator Andrew Jarvis, logging you in...


Name     : Andrew Jarvis
Age      : 35
User Name: andrewj@vivlio.org

Number of books on rent: 8
Number of books sold   : 7
Total revenue earned   : Rs. 1935.00


What would you like to do?
1. See list of all customers
2. See list of all books along with their copies
3. See the complete details of a book
4. Modify the details of a book or add a book
5. Remove a book
6. See the history of a customer
7. Delete an account of a customer
8. Log out

Enter option number: _
```

Admin Functions

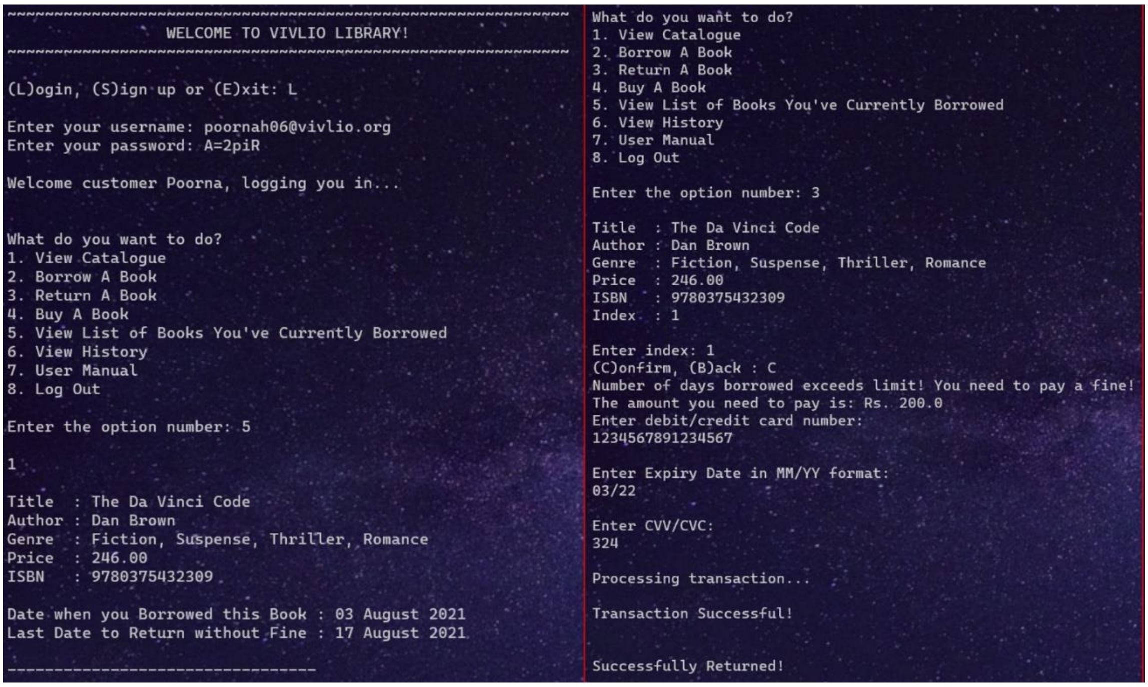
```
2. See list of all books along with their copies
3. See the complete details of a book
4. Modify the details of a book or add a book
5. Remove a book
6. See the history of a customer
7. Delete an account of a customer
8. Log out

Enter option number: 3

Enter the title of the book:
Starsight

Title  : Starsight
Author : Brandon Sanderson
Genre  : Fiction, Science Fiction
Price  : 290.00
ISBN   : 9780399555817

Number of Copies: 10

Borrowers:

1

Name      : Adarsh
Age       : 24
User Name: ash786@vivlio.org
Returned : Yes

2

Name      : Adarsh
Age       : 24
User Name: ash786@vivlio.org
Returned : Yes


Buyers:
No Buyers!
--------------------------------
```

Borrowers and Buyers of a Book

```
Enter your username: andrewj@vivlio.org
Enter your password: aDm1n-PasS

Welcome administrator Andrew Jarvis, logging you in...


Name     : Andrew Jarvis
Age      : 35
User Name: andrewj@vivlio.org

Number of books on rent: 8
Number of books sold   : 7
Total revenue earned   : Rs. 1935.00


What would you like to do?
1. See list of all customers
2. See list of all books along with their copies
3. See the complete details of a book
4. Modify the details of a book or add a book
5. Remove a book
6. See the history of a customer
7. Delete an account of a customer
8. Log out

Enter option number: 7

Enter the username of customer: harry123@vivlio.org

Name     : Harish
Age      : 25
User Name: harry123@vivlio.org

Delete this account? (Y/N): Y
Account deleted successfully!
```

Removing an Account

This set of screen shots displays a run few features in the admin. After logging in, the revenue, number of books on rent and number of books sold are displayed along with the menu. Option 3 lets the admin see the complete details of a book (example here: Starsight). The admin can delete an account if there are no dues.



```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                WELCOME TO VIVLIO LIBRARY!
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

(L)ogin, (S)ign up or (E)xit: L

Enter your username: poornah06@vivlio.org
Enter your password: A=2piR

Welcome customer Poorna, logging you in...


What do you want to do?
1. View Catalogue
2. Borrow A Book
3. Return A Book
4. Buy A Book
5. View List of Books You've Currently Borrowed
6. View History
7. User Manual
8. Log Out

Enter the option number: 5

1

Title  : The Da Vinci Code
Author : Dan Brown
Genre  : Fiction, Suspense, Thriller, Romance
Price  : 246.00
ISBN   : 9780375432309

Date when you Borrowed this Book : 03 August 2021
Last Date to Return without Fine : 17 August 2021

--------------------------------
```

```
What do you want to do?
1. View Catalogue
2. Borrow A Book
3. Return A Book
4. Buy A Book
5. View List of Books You've Currently Borrowed
6. View History
7. User Manual
8. Log Out

Enter the option number: 3

Title  : The Da Vinci Code
Author : Dan Brown
Genre  : Fiction, Suspense, Thriller, Romance
Price  : 246.00
ISBN   : 9780375432309
Index  : 1

Enter index: 1
(C)onfirm, (B)ack : C
Number of days borrowed exceeds limit! You need to pay a fine!
The amount you need to pay is: Rs. 200.0
Enter debit/credit card number:
1234567891234567

Enter Expiry Date in MM/YY format:
03/22

Enter CVV/CVC:
324

Processing transaction...

Transaction Successful!


Successfully Returned!
```

In customer, after logging in, the menu is displayed. When option 5 is chosen, the list of books borrowed is displayed. When option 3 is chosen, the borrowed book can be returned, but since the 14 days limit has been exceeded, a fine needs to be paid before returning. This leads to the payment section. After paying, the returning transaction is completed.