FEU Institute of Technology
COLLEGE OF ENGINEERING • COLLEGE OF COMPUTER STUDIES

College of Computer Studies

DATA STRUCTURES AND ALGORITHMS

(CCS0015L)

BookFlow: A Book Borrowing System

GRADE

Submitted by:

Arada, Aaron Randolph
BSCSSE
TN03

Boo, James Gabriel
BSCSSE
TN03

Januya, Joaquin Elizandro
BSCSSE
TN03

Legara, Dashtin Kyle
BSCSSE
TN03

Marfil, Ezekiel
BSCSSE
TN03

Submitted to:

FCALMENIANA

I. Introduction

The system that the programmers created is a Book Borrowing System. The system implements the use of various Data Structures and Algorithms discussed in the course. The specific system was chosen to provide a solution for inefficient, slow, and unreliable manual recording that libraries rely on. The programmers aim to create a Book Borrowing System that is efficient and accurate by implementing various Data Structures and Algorithms such as: Linked Lists, Vectors, Pointers, File-Handling, and several others, which would ensure an efficient and reliable data collection, processing, and presentation. These functions were implemented by using Abstract Data Types (ADTs) and Standard Template Libraries to ensure a proper Abstraction and Encapsulation of the implementation details.

II. Description of the Project

The Book Borrowing System is a digital platform that allows users to borrow, return, and search for books within the library's catalog. The system records all user transactions and book availability in real-time, ensuring that records are always up to date.

III. Objectives

The main objective of this project is to develop an efficient and user-friendly Book Borrowing Catalog System that improves how library resources are managed and accessed. Specifically, the system aims to:

1. Replace manual recording with a digital catalog to minimize human error and reduce time spent on logging transactions.
2. Implement key data structures and algorithms (such as arrays, linked lists, stack, and trees) to manage book and user records effectively.
3. Enable real-time tracking of book availability, borrowing status, and return history.
4. Allow users to search and browse the catalog using efficient search algorithms.
5. Provide administrative functions for managing the catalog, such as adding or removing books and viewing borrowing history.
6. Enhance user experience by providing a fast, reliable, and organized interface for borrowing and returning books.
7. Easier access to books data

IV. Significance of the Study

The Book Borrowing System would resolve the issues of the traditional manual recording methods commonly used in older library systems. The system created would ensure a quick and accurate retrieval, processing, and presentation of data, which is beneficial for individuals who often use the library to borrow, return, or search for books. Specifically, the beneficiaries of the system are the following:

1. **Librarians.** The ease of use that the book borrowing system provides is beneficial for librarians as it allows an efficient and accurate access to data that is necessary to keep track of the books that are borrowed and returned by library users.
2. **Students and Library Users.** The system would be beneficial for individuals, mainly students, who often use the library to search, borrow, and return books. Due to the efficiency and simplicity that the system provides, the users would be able to conserve more time and effort.
3. **Institution and Other Libraries.** The system would be beneficial for the institution and other libraries as the system allows for an accurate tracking of the individuals who borrow a book. This is important to ensure that the books are preserved and properly returned each time it is borrowed.

V. Project Details

Table of Functions

Function Name	Scope (Local / Global)	Purpose
database_file()	**Local**	**It initializes the book list file if not found**
LoadFromFile()	**Local**	**Loads books from txt file (library_data.txt) into a linked list**
SaveToFile()	**Local**	**Saves the linked list of books back to the txt file**
Menu()	**Local**	**Displays the main menu options**
ShowBook()	**Local**	**Displays all books and their details**

BorrowBook()	**Local**	**Handles borrowing of books with ID check and ban check**
ReturnBook()	**Local**	**Handles returning of books and issues penalty for late return**
SearchBook()	**Local**	**Searches the linked list for a book by its ID using sequential search**
history()	**Local**	**Displays borrow history from file**
checkName()	**Local**	**Validates that the borrower name**
getCurrentDate()	**Local**	**Returns the current date in MM/DD/YYYY format**
Alis()	**Local**	**Handles the programs exit confirmation**
sortingData()	**Local**	**Sorts the books by the availability status using bubble sort, borrowed first then available**
date_time()	**Local**	**display the current date and exact time**
isBanned()	**Local**	**Open “Banned_data.txt” file in read, check if the file is properly opened, and check if there is a matching ID number and save it to file.**

Table of Classes

Class Name	Scope (Public /Private)	Purpose
BOOK	**Public**	**Main class for managing books, borrowers, borrowing, and returns**

Table of Data Structures/ADT

Identifier Name	Type of Data Structure/ ADT Operations	Purpose
bookList	dequeue<books>	Temporary book list initialization before file loading
head	books* (linked list	Head pointer to dynamically loaded book records
borrowHistory	stack<string>	Keeps track of latest borrow/return events (LIFO)
books	struct	Stores information of individual book records
borrower	struct	Temporarily stores

Table of Algorithms

Algorithm Name	In what part of the program was it applied	Purpose
Sequential Search	SearchBook(), isBanned()	Find the specific book using a book ID and check if the borrower is banned.
Stack	history(), BorrowBook(), ReturnBook()	Saves the record of latest borred books, and deletes if returned.
Bubble Sort	sortingData()	Sorts the books via availability status.
Book Availability Check	BorrowBook, ReturnBook	Checks if a book is available or borrowed before processing action.

Late Return Detection	**ReturnBook()**	**Compares return time with borrow time (7 days) to trigger ban, in our system 1 day is equivalent to 1.**

Division of Labor

Function Name	Author/Developer	Date Completed
database_file()	**Arada, Aaron Randolph**	**July 12, 2025**
LoadFromFile()	**Legara, Dashtin Kyle**	**July 12, 2025**
SaveToFile()	**Arada, Aaron Randolph**	**July 12, 2025**
Menu()	**Marfil, Ezekiel O.**	**July 12, 2025**
main()	**Boo, James Gabriel**	**July 12, 2025**
ShowBook()	**Januya, Joaquin Elizandro B.**	**July 12, 2025**
BorrowBook()	**Marfil, Ezekiel O.**	**July 12, 2025**
ReturnBook()	**Marfil, Ezekiel O.**	**July 14, 2025**
findBookById()	**Boo, James Gabriel**	**July 14, 2025**
history()	**Arada, Aaron Randolph**	**July 14, 2025**
checkName()	**Legara, Dashtin Kyle**	**July 15, 2025**
getCurrentDate()	**Boo, James Gabriel**	**July 15, 2025**
Alis()	**Boo, James Gabriel**	**July 17, 2025**
getCurrentDateTime()	**Marfil, Ezekiel O.**	**July 17, 2025**
sortingData()	**Januya, Joaquin Elizandro B.**	**July 17, 2025**
date_time()	**Januya, Joaquin Elizandro B.**	**July 17, 2025**
isBanned()	**Legara, Dashtin Kyle**	**July 17, 2025**

VI. DATA ITEMS SPECIFICATION

Data Item	**Input/ Output**	**Data Type**	**Length**	**Decimal**	**Formula used**	**Allowed Value**
book_id	Input/Output	string	4	-	-	B001-B020
b.book_title	Output	string	-	-	-	-
b.book_author	Output	string	-	-	-	-
b.book_availability	Output	string	-	-	-	-
currentDate	Output	string	10	-	date_time()	-
name	Input	string	-	-	-	alphabetical
b.borrower_name	Input	string	-	-	-	alphabetical
b.borrower_ID	Input	int	9	-	-	numeric
line	Output	string	-	-	-	-
b.BookID	Input/Output	string	4	-	-	B001-B020
bookBorrowed	Output	string	-	-	-	-
needswapped	-	-	-	-	-	-
b.confirm	Input	char	1	-	-	Y, N
b.again	Input	char	1	-	-	Y, N
wow	-	bool	-	-	-	-
borrowHistory	Output	stack<string>	-	-	-	-
id	Output	string	4	-	-	-
title	Output	string	-	-	-	-
author	Output	string	-	-	-	-

status	Output	string	-	-	-	-
se	Input	string	4	-	-	B001-B020
current_dt	Output	time_t	10	-	time(0)	-
save_time	Output	time_t	10	-	sizeof(time_t)	-
inputBookID	input	string	4	-	-	alphanumeric
found	-	bool	-	-	-	-
validDate	-	bool	-	-	-	-
r.borrower_name	Input	string	-	-	-	alphabetical
r.borrower_ID	Input	int	9	-	-	alphanumeric
bookID	Input	string	4	-	-	B001-B020
r.confirm	Input	char	1	-	-	Y, N
r.again	Input	char	1	-	-	Y, N
e.again	Input	char	1	-	-	Y,N
option	Input	int	1	-	-	1-6
obj	-	BOOK	-	-	-	-

VII. Screen Output

This should include the screen outputs of the features of the project along with detailed explanations.

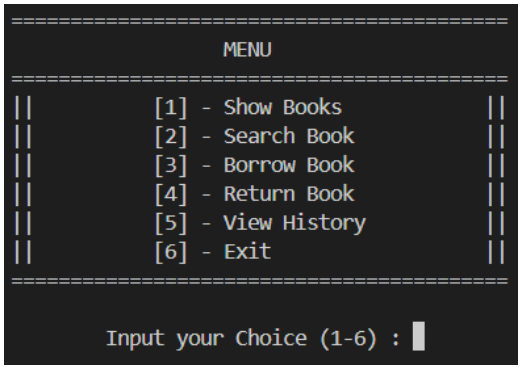

Figure 1. Menu for BookFlow

Figure 1 shows the menu for BookFlow that include the following:

1. **Show Books**
2. **Search Book**

3. **Borrow Book**
4. **Return Book**
5. **View History**
6. **Exit**

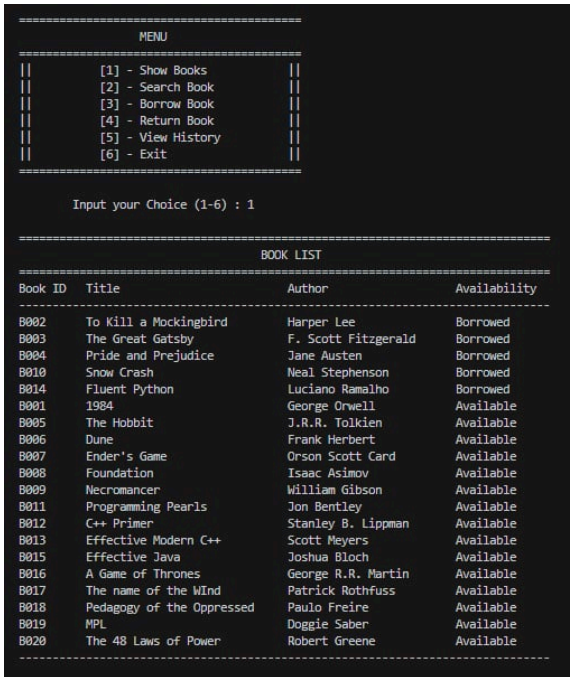

Figure 2 presents the option 1 (showbooks) output

Displays all the book titles with their corresponding authors, id's, and availability.

Figure 3 presents the option 2 (search book) output

It will ask the user to input a Book ID, then checks the database for a matching Book ID. If a match is found, the system will display the book information. If no match is found, the system will output "Book not found."

Figure 4 presents the option 3 (borrow book) output

It will ask the user to input their name, a 9-digit ID number, and the Book ID of the book they want to borrow. The system will notify the user to return the book within 7 days to avoid being banned, then it will ask for confirmation.

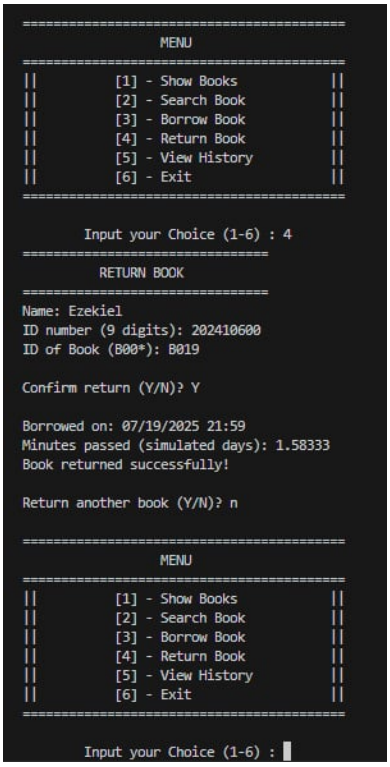

Figure 5 presents the option 4 (return book) output

It will ask the user to input their name, a 9-digit ID number, and the Book ID of the book they want to return. Then it checks whether the return is late. If it is, the user's ID will be banned from using the system.

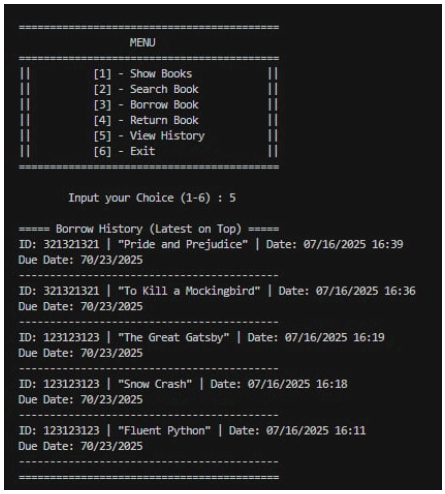

Figure 6 presents the option 5 (view history) output

This will display the borrowing history, showing each book by its ID, along with the student ID of the borrower and the corresponding return date.

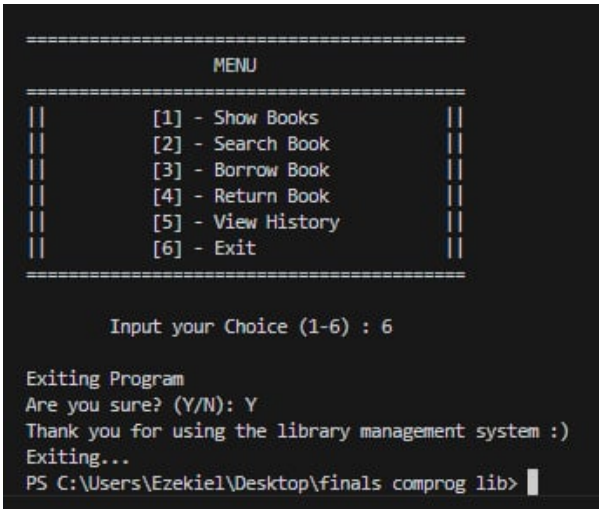

Figure 7 presents the option 6 (exit) output

It will ask the user for confirmation to exit the system.

VIII. Source Code

Upload both .cpp and .txt of your code.