

CC202 SYSTEM DOCUMENTATION

By Iloilo Shareholders (Group 6)

Passed to: Sir. John Cairo Minerva

Group Members:

Leader: Dolutan, Alfonso

Jurquina, Joaquin Daniel

Morena, Tim Matthias

Introduction:

This paper serves as an in-depth official documentation in the making of the Iloilo Shareholders (Group) Library System. This is a Library Management System created by the Iloilo Shareholders. The Iloilo Shareholders library system contains both admin and guest functions, with the admin only accessed with a username and password. The admin functions are to add, delete, and update books. While the guest functions are to rent, return, and search books.

System Overview:

Struct

- We implemented a “struct” (also known as Structure in our system.
- According to w3schools, it’s a way to group several related variables into one place.
- For example, books usually contain a: Title, Author, Publishing Date, and Genre.

Main Login

- We implemented a main login system which has an If - Else Statement.
- This would be where the user would be prompted to choose either logging in as an admin or as a guest, which have different functions.
- Admin function is linked behind a username and password checker, while guest functions do not.

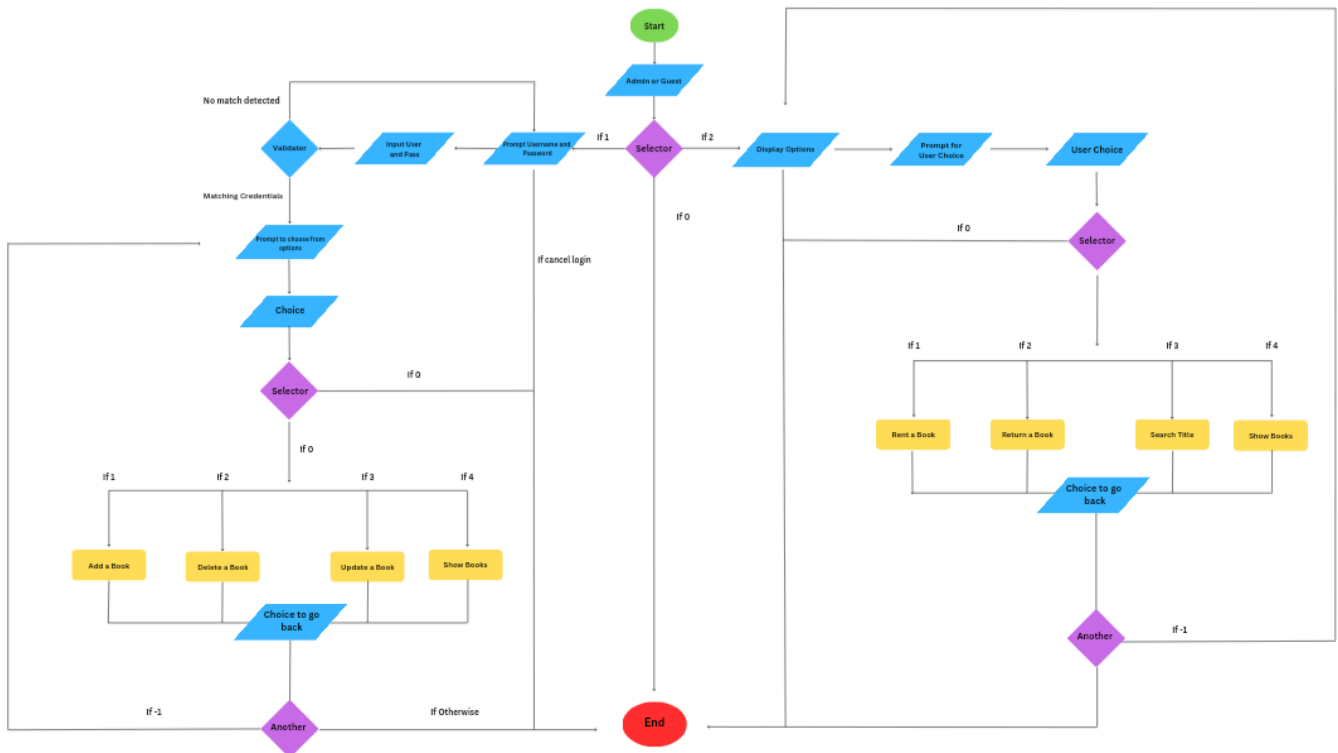
Admin

- We implemented a switch statement for our admin menu.
- The switch statements here would all lead into function declarations.
- This is all in a do-while loop so that in case the user wants to do other functions limited ONLY to the guest menu.

Guest

- We implemented a similar menu for our guest menu
- The switch statements here would all lead into function declarations
- This is all in a do-while loop so that in case the user wants to do other functions limited ONLY to the guest menu

Flowchart:



Pseudocode:

Main Login

- User will be prompted to select 3 options:
 - Enter "1" for Admin Login
 - Enter "2" for Guest Login
 - Enter "0" to Exit Program
- If User enters "1", the program will ask the user to input a username and password. If User's input string matches the string of username and password in the program, the user will be directed to the admin menu.
- If the User's string input doesn't match the string of username and password in the program, the program enters a loop where the user has to keep re-entering their credentials until they match, or enter the sentinel integer to exit the program.
- If User enters "2", there will be no username and password prompt, the user will be directed immediately to the guest menu.
- Finally, if User enters 0, the program will end.

Admin Login

1. If credentials match, the user is prompted to choose the following options:
 - Enter “1” to add a new book to the library.
 - Enter “2” to delete a book from the library.
 - Enter “3” to edit current books in the library.
 - Enter “4” to view book status
 - Enter “0” to exit program
2. After each option is run, the program will be sent in a do-while loop asking the user if he wants to do another action. (Note: This option doesn’t apply for Option “0”)
3. These options are mostly done with the help of “struct” and “vector.” These two data types are the main parts of our admin functions which we will discuss later

Admin Functions:

- *AddBooks ();*
 - i. This process has two parts that is done first in the main switch statement, and in a separate function called “addBooks();”
 1. First, our program would call the struct by naming the variable “newBook”. Next, our program will prompt users to enter information about the book. This information includes the Title, Author, Date Published, and Genre. The availability status is automatically set to “Available”.
 2. Once the user has inputted values in the prompts, the program will call out the addBooks function to print these inputs to the text file.
 3. In our addBooks(); function, this will mostly contain formatting code on adding user input into the text file. With using text files, we have to include <fstream> in our header files.
 4. We use fstream to open our text file as write only (ios::out) to write the user’s input into the text file. We opted to have a simple but strict format in the text file to avoid confusion for our later functions.
 5. In the start of our function, we initialize a variable called “bookcount”. This integer will be in charge of numbering the current books. With this, we opened our text file as read only (ios::in) and inputted a getline(while) loop. This while loop will read all lines in the text file, however if the loop detects a “-----” line or as we call it a separator line, will increment bookcount++.
 6. After the while loop, don’t forget to close the fstream variable. After closing, we can finally format the user input for a new book.
 7. We opted to use a simple but strict format for the text file:
Title:
Author:
Date:
Genre:

Status:

8. After status, is where we put our separator mentioned awhile ago. This separator is important for 2 reasons: To separate book information from each other to avoid confusion, and to serve as some sort of condition statement. Once the program has multiple books, it will use the separator line as an indicator that a new book has been added, hence why incrementing bookcount on detection.
- *showBooks ();*
 - i. This program has one goal, to display all the lines in the book file
 1. The program uses a while loop that uses the getline function to get all lines in the text file. Once all lines are read in the text file, it will be displayed by the same while loop.
 2. The output strictly follows the text file format, and is usually used before most functions in this system
 - *deleteBooks ();*
 - i. This function has one main goal: detect a similar book number inputted by the user and delete all information with that book number. If there are multiple books, the program should change the remaining book numbers according to the remaining books.
 1. The program will first call on the showBooks(); function to show the available books to the user. Once the user has finished looking at the available books, the user will be prompted to select a book to delete by typing its book number in the terminal.
 2. Once the user has inputted his choice, the program will call on the deleteBooks function. First the program will call on the fstream function to open the text file with ios::in (read only)
 3. The function will run a while loop that reads all of the lines in the text file. In addition, we put an if statement in this while loop. Once the while loop has found a “.” in the line, will trigger a sequence.
 4. This sequence would begin with the program creating a struct variable. Then the program would use a getline function to store inputs from the text file, but specifically inputs 2 spaces after the “.” character. Since year is an int variable and not a string variable, we used “stoi” to convert the int data type to a string data type so that the getline function could be used.
 5. Then, all the data that was gathered from the getline functions would be all stored in a vector. Once in a vector, the program will use the books.erase() function specifically on the index that starts with (bookNum - 1)
(-1 was added because the indexes start from 0)

6. After the specific index has been deleted, the program will open the text file in a write only mode (`ios::out | ios::trunc`) to run a for loop that returns all the data from the vector back to the text file. This data is formatted in ascending order
- *updateBooks ()*;
 - i. This process has one goal: to detect a matching book number in the program. Once a matching value has been found, the user should be allowed to edit text inside the text file.
 1. In the main switch statement, first the program will display the current books available in this library system with the `showBooks()` function. Once the user has finished reading, the user is prompted by the system to choose a book to edit.
 2. Once the user has inputted the book number of the book that they want to edit a while loop will be called to read all lines into a vector. In this while loop, we will use the `find()` function to find two things: a digit that is `!< && !=0`, and if there is a “.” present in the line.
 3. During the while loop mentioned in Step 2, we will initiate a variable named `currentBookNum`. Once a “.” and a digit `!< && !=0` is detected, will increment itself by one.
 4. After the while loop has finished gathering all lines of information from the text file, will store itself in a vector called `fileLines`
 5. Once in a vector, the program will run a for loop using `size_t`. This for loop is used to “count” indexes in the vector array. This will be used to find the index (book) that the user wants to update from his choice from the switch statement

Guest Login

1. If credentials match, the user is prompted to choose the following options:
 - Enter 1 to rent a book
 - Enter 2 to return a book
 - Enter 3 to search for a book’s title
 - Enter 0 to exit the program
2. After each option is run, the program will be sent in a do-while loop asking the user if he wants to do another action. (Note: This option doesn’t apply for Option “0”)
3. These options are mostly done with the help of “struct” and “vector.” These two data types are the main parts of our admin functions which we will discuss later

Guest Functions:

- *rentBooks ()*;
 - i. This program uses a similar type of function with the `updateBooks()` function, however only has the ability to update the “Status” in the book struct.
 1. The program will call the `showBooks()` function to display the current books in the library program for the user. Once the user has finished reading the available books, the user will be prompted to select a book to rent.

2. Once the user has selected a book to rent, the program will call on the `rentBooks();` function, where the first thing it will do is open the text file in read only to read all lines of the text file using a `getline(while)` loop
 3. In the while loop, once the line has detected a “.” and a digit that is not less than or equal to 0, will initiate a variable called `currentBookNum++`
 4. Once the while loop has finished reading all lines, will store all data from the text file into a vector.
 5. Then the program will run a for loop to find out the location of the book that the user wants to edit, using the same parameters on Step 3.
 6. When the user’s input and for loop’s counter match, the program knows that the book’s index has been found.
 7. Earlier, we had to update 4 things in the struct, namely the Title, Author, Date, and Genre. However, since we are just renting a book, we just need to edit the 5th line which is the status.
 8. The program will first display the book’s data, then prompt the user to input the date on which the book is being rented. The date will follow a strict format of YY-MM-DD.
 9. Once that process has finished, the program will call on a stringstream variable called `rent`. This stringstream will allow us to edit the format for our rental date.
 10. Once the process has finished formatting the date of rent, will replace the old status with the date of rent in the vector.
 11. Once all other processes have been finished, the program will call on a for loop that will input all data from the vector back to the textfile.
- *returnBooks ();*
 - i. This program uses a similar type of function with the `updateBooks();` function, however only has the ability to update the “Status” in the book struct.
 1. The program will call the `showBooks();` function to display the current books in the library program for the user. Once the user has finished reading the available books, the user will be prompted to select a book to rent.
 2. Once the user has selected a book to return, the program will call on the `returnBooks();` function, where the first thing it will do is open the text file in read only to read all lines of the text file using a `getline(while)` loop
 3. In the while loop, once the line has detected a “.” and a digit that is not less than or equal to 0, will initiate a variable called `currentBookNum++`
 4. Once the while loop has finished reading all lines, will store all data from the text file into a vector.
 5. Then the program will run a for loop to find out the location of the book that the user wants to edit, using the same parameters on Step 3.
 6. When the user’s input and for loop’s counter match, the program knows that the book’s index has been found.

7. Earlier, we had to update 4 things in the struct, namely the Title, Author, Date, and Genre. However, since we are just returning a book, we just need to edit the 5th line which is the status.
 8. The program will just automatically input “Available” on the Status line in the vector
 9. Once all other processes have been finished, the program will call on a for loop that will input all data from the vector back to the textfile.
- *searchTitle ()*;
 - i. This function has one goal: to match user input with any book.title in the text file. If input matches, will display the book details. Else, will output that the “book is not available”
 1. First, the program will open the text file as “read only”. Then will prompt the user to enter the title of the book that they want to search.
 2. Once the user has entered their choice, two data types will be initialized: a bool data type named *isFound* (automatically set to false), and a string data type named “lines”.
 3. Once the variables have been initialized, the program will run a while(getline) loop to find two things in the text file: the string “Title:” && the specific title of the book.
 4. If both conditions are satisfied, the bool value “*isFound*” is set to true. Then the program will run a for loop with these parameters: for (int i = 0; i < 4, ++i). This for loop is to display the book’s contents to the user.
 5. If both conditions are not satisfied, the program will display that there is no book.

Conclusion

Iloilo Shareholders Library Management system is the culmination of all our collective planning and hardwork this semester. Our collaborative efforts as well as the professor suggestions made the process fun and worthwhile and in the end, the Iloilo Shareholders (Group 6) not only made a finished Library System but have successfully defended its creation. To close this chapter, We would like to give our heartfelt gratitude to sir John Cairo Minerva for his patience and understanding through every phase of our process. With that, we know the drill. semester dismissed.



References:

for Admin/Guest Login System:

<https://github.com/abdulsamie10/Library-Management-System>

<fstream> and its different functions:

https://www.youtube.com/watch?v=EaHFhms_Shbw&t=370s

array indexing, a vital point in our system:

https://en.cppreference.com/w/c/types/size_t#:~:text=size_t%20is%20commonly%20used%20for,on%2032%2Dbit%20modular%20arithmetic.

the STOI() <sstream> function:

<https://www.geeksforgeeks.org/convert-string-to-int-in-cpp/>

information about vectors:

https://www.w3schools.com/cpp/cpp_vectors.asp

<https://www.youtube.com/watch?v=RXzzE2wnnlo&t=397s>

<https://www.youtube.com/watch?v=QFrJQq6Iox8>

<https://cplusplus.com/forum/beginner/284500/>

Opinion from Reddit: better to use filename instead of “.txt”:

<https://stackoverflow.com/questions/36824225/use-string-or-array-of-char-for-filename>