User's_Guide_of_the_Library_Project

Who_Are_We_and_What_We_Did?

Dung Dinh: Implemented ***customer.cpp, customer.h*** *and* ***class-relative part*** *of* ***main.cpp***

Armando Mata: Implemented ***librarian.cpp, librarian.h*** *and* ***class-relative part of main.cpp***

Songwen Xue: Implemented the ***User.h***, ***User.cpp*** and the ***password verifying part*** of ***main.cpp***

What's_In_Our_ Program?

Our program mainly contains three classes, which are ***User(Based class), Customer(Derived class) , and Librarian(Derived class),*** and two files, which are ***books.txt*** and ***accounts.txt.***

- ***User Class:***

  As a base class, it contains most of the methods that we are going to use in the ***Customer Class*** *and* ***Librarian Class.*** *Its primary job is to copy the information from the files to the two-dimensional vectors, which will make it more convenient for modifying the files later on. It also contains different* ***Getter functions, Setter functions*** *and three* ***virtual functions*** *which later being used as* ***polymorphism.***

- ***Customer Class:***

  As a derived class, it inherits all of the non-private members of its base class - User Class. It's primary job is to give a customer that has successfully login the option to either checkout or return a book. If a customer chooses a valid option, the current library would be printed out. The program would then ask for a title and compare that title for a match in the current library. If it finds a match, the program checks to see if that book is available for checkout/return depending on the user's input. If successful, the program changes the book's availability and updates the text file. However, if the user tries to return a book that has not been checked out yet or checkout a book that is not there, the program notifies the user of the book's current status and exits. Otherwise if the program does not find a matching title, it notifies the user that there is no such book in the current library.

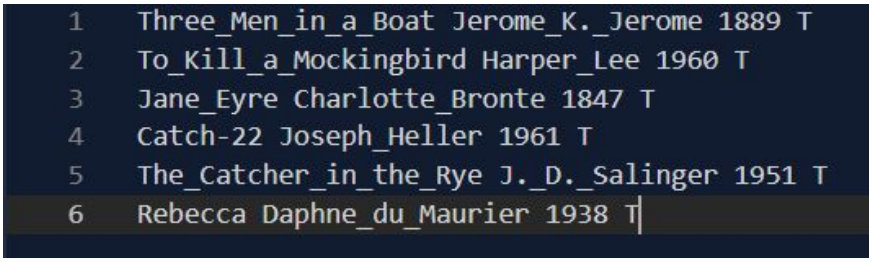
- ***Libralian Class:***

  The librarian class is a derived class of the User class. As such, it has access to the setter/getter functions of User, as well as the other helpful classes for displaying and updating information. As a librarian, the user has access to the database of books and is able to modify the library as they see fit; either by adding or removing books. When the librarian logs in they are greeted with their specific menu allowing them to either add a

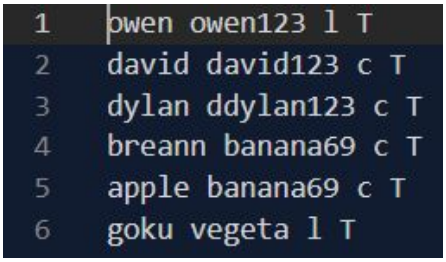
book, remove a book, see what books are currently in the database, or log back out. When adding a book the librarian must enter the name, author, and year of publication, then the program will update the book database with the new book. In order to remove a book from the library, the librarian will be shown the current books in the database and then he/she must type in the name of the book they wish to remove. Lastly, if the librarian simply desires to see the current books available in the database, they may do that also. After completing either of the three options, they will be brought back to the main menu where they can choose from another one of the first three options. If they no longer need to see/modify the books database, they can press the number 0 to log out.

- ***Books.txt File:***

  This is the file that contains all the information of the books. As the picture shown on the left below, the information of each book will take up one line. For each line, we will have the title of the book, the author, publish year and the status of the book, they are separate by **one** space. If there are spaces in the title and the author's name, use "_"s to replace those spaces. For the book availability, we use "T" for available and "F" for unavailable, which means it has been checked out. ***Remember not to include any empty spaces at the end of each line or any empty lines at the end of the file, otherwise this program may collapse.***



books.txt



accounts.txt

- ***Accounts.txt File:***

  This file is used to store the users' information. As the picture shown on the right above, the information of each user takes up one line. On each line, it contains the user name, password, user type ("l" for librarian and "c" for customer), and account status ("T" for valid and "F" for invalid or frozen). Each of those elements are separated by a **single** space. ***Remember not to include any empty spaces at the end of each line or any empty lines at the end of the file, otherwise this program may collapse.***

How_To_Use?

*Our program is very **user-friendly**. Follow the steps below.*

1. *Launch the program.*
2. *Type in your name and hit "Enter", the program will automatically identify the type of your account and give you a feedback.*
3. *Type in your password*

    a. *For some reason, you will see what you are typing, so make sure no one is standing behind you at that moment.*
    b. *If you fail to type in the correct password for 3 times, which you will be notify by our system, your account will be automatically frozen, you won't be able to unfreeze it until you contact us.*
4. *Once you log into our program, we will provide a menu for you based on your account type. Have Fun!*
    a. *For **customers'** accounts:*
        i. *If you want to check out books, type in "checkout" and hit "Enter". Then you will see a list of our books and would have to provide the name of the one you want to check out. Same idea for returning books.*
    b. *For **librarians'** accounts:*
        i. *If you want to add books to or remove them from our booklist, after you picking the corresponding menu option, you are require to type in the information of the book. **Keep in mind that for those elements you input, if there are spaces, replace them with "_"s.***

Where_Are_the_Bonous?

- ***Inheritance ---** We used the **User Class** and have the **Customer Class** and the **Librarian Class** inheriting from it.*

- ***Polymorphism ---** In our **main.cpp,** we allocate the **User type** pointer to the **Customer** and **Librarian** variables and have **virtual functions BookIn, BookOut, and menuOption** in **User.h .***

- ***Exception handling --** we used it in the **files handling** in **User.cpp***