Gavin Huang: Date counter, File operation, recommend books, all librarian operations, search books, moving operations from midterm project to each class in the final project, printAll

Jiajie: User authentication, student operations, borrow, reserve, renew, cancel reservations, password mask, adjust ending date, favor, penalties

We are using windows.

Parameters are passed between functions through identifiers like ISBN, id, and usernames which pertain to a unique user, book, or copy. From these, all other data can be accessed and changed through functions which search the vectors and output the memory address of the specific user, book, or copy. User input is requested through the console for functions that require it. This was done so we can pass the least number of parameters between functions. Any other information that is needed can be gathered through the vectors and accessor functions.

Overloading is used for the insertion and extraction operators. This allows them to be used on the classes. With them, we can access data from files and output to the files, or to the console.

Template is used in cases where we need to apply the exact same function to all types of classes. It is used to print out every item in the library since every class has their own overloaded extraction operator to output every bit of data about them.

The way file operations work is that the insertion operator is used to write the information of each class in a formatted manner. The extraction operator is meant to read this format and put the data back into the system. We use a function to read the files and one to output the files in the library header. In these functions, the txt files which store all the data are read or written depending on whether we want to input to the system or output to the files. The writeFiles() function is used to update the text files every time a change is made to the library.

The date counter works by setting a global variable t\_start which marks the ticks where the clock begins. Afterwards, we can call clock() and subtract by t\_start to find the number of ticks that have elapsed, and by dividing it by the ticks per second, we can find the number of seconds that passed, and then the number of days that passed. This is placed in copies to keep track of the day for each copy.

Recommend books works by taking the vector containing all the books in the library and sorting it by popularity. Popularity is determined by number of reserves. If the user has not borrowed any books, it lists the 10 most popular books unless there are less than 10 books, in which it would list every book. If a user has borrowed a book, the most recently borrowed book is used as the category that would be recommended to the user. In this case, the 10 most popular books of the same category are recommended. If there are less than 10, it lists every book of the same category.

Search books first asks for a type to search by. For ISBN, it checks every book to see if it shares the same ISBN, and if it does, lists the available copies. When searching by other types, it checks every book to see if it has the parameter that is wanted. If it is found, it is pushed into a vector that stores each book. Once it is found, every available copy related to each book is outputted into the console.

Add books is implemented by adding a book if there exists no book in the library and the ISBN is a new ISBN. If there is a book, it adds a new copy and increments count. A new book’s index is added by taking the last index, incrementing it by 1, then using it as the new index. Adding a new ID is done by taking the last ID of a copy, incrementing it by 1, then using it as the new ID.

Deleting books first checks if the copy exists or is available. If these conditions are fulfilled, it checks the count and deletes the book and copy if the count is 1. If the count is higher, it deletes the copy and decrements the count.

Adding new users asks for a type of user to be added as and checks if the new username is already used. If it is not used, the user is added to the library and written into its specific text file.

Search users works by using the overloaded operators to output students, teachers, or librarians. It takes a username and outputs a user if they exist.

Deleting users works by taking username inputs, then removing them from their specific vector that stores every user of its type. The only except is librarians, in which the last user cannot be deleted and that removing a librarian logs them out.

To borrow a book, you must login to your account. If you do not have an account, you must ask a librarian to make an account for you. Once you are logged in, you must check which book you want to borrow. To do this, you must search for books to find the ISBN of the book you want to borrow. If you want a recommendation, you can choose to recommend a book instead. Once you have the book you want, you select borrow a book and input the ISBN. If there are copies available, you will receive the copy. To return a book, you must choose to return a book and then choose the ID of the copy you wish to return. The ID of the copy is given when you borrow the copy, but it is also shown when you select my information. This returns the book.

In order to login, the system asks for a username and password. The password is masked to protect the user’s credentials. The system then takes the username and checks the library’s vectors for students, teachers, and librarians. The system also keeps track of the type of user as well as the index of the user in that specific user vector. This allows the system to use functions on the correct user. After the system has logged the user in, it displays the menu and displays the options for each user. For the inputs on the main menu for each user, it uses an overloaded function in the user class. That means that there is a base, generic menu for a user and a different menu for the derived classes.

In order for the user to change do anything in the system, they input a number signifying what action they want the system to do. The system also checks what type of user they are in order to accept different types of commands for a reader or a librarian. Each command also has a different case for if the user is a student or a teacher.

Borrowing a book requires the user to input a value for the ISBN of the book they wish to borrow. The system checks if the ISBN is of a book that exists and ends the function if the ISBN is invalid. It also checks through all the copies for if it has already been borrowed by the user. It also checks if the user has any overdue books to stop them from borrowing. It then checks for any available copy to borrow and sets the appropriate information in the copy of the book as well as adds it into the user’s borrowed vector. It also works if the user is the first one on the reserve list.

Returning a book requires the user to input a value for the id of the copy they wish to return. It checks if book being returned is overdue and gives a penalty to the user. If the penalties are a multiple of 5, it reduces the maxCopies by 1. It resets all the information in the copy and the user. After all that, it gives a poll to the user on whether they enjoyed the book or not.

Reserving a book requires the user to input a value for the ISBN of the book they wish to reserve. It checks if the ISBN for validity and also checks if a user has reserved the book to which it stops the command. It adds the user to the reserve list and puts the user in the copy if they are the only reserve there.

Cancelling a reserve requires the user to input a value for the ISBN of the book they wish to cancel the reserve. It checks for the validity of the ISBN and erases the user from the reserve list. If there is no reserve left, it returns the reserve in the copy to the default state.

Renewing requires the user to input an id for the book they wish to renew. The system checks for if it is inside their borrowed copies vector and adds another instance of the maxBorrowingPeriod.