Muhammad Ahmed | M00898450

Library management system report

About the Task:

The Library Management System is designed to streamline book borrowing, returning, and member management. The implemented system ensures organized handling of books, members, and transactions. Object-oriented design with classes like Book, Member, Librarian, etc.

Class Hierarchy:

The system defines several classes, including Date, Person, Member, Book, and Librarian. Date class handles date-related operations. Person is a base class for Member and Librarian, encapsulating common attributes like name, address, and email. Member class represents library members, storing information such as member ID and a vector of borrowed books. Book class represents library books, containing details like book ID, title, author, due date, and borrower. Librarian class extends Person and includes additional attributes like staff ID, salary, and vectors for storing members and books.

Data Types:

Used vectors to store members (and their details). Used vector to store the books (along with book id, author name, book name etc) when reading in from the .csv file. Mainly string and integer is used for basic manipulation of data.

UML diagram around which the system is built:

A diagram of a computer program

Description automatically generated with medium confidence

Keeping the UML diagrams in mind, first, all the classes were declared, followed by getter and setter methods. Any necessary functions were carefully thought on how they would help achieve the desired outcome. There are not much of commits during the process because I was having some problem (authentication error) pushing code to git repo which I figured out just a few days before deadline.

Functions/Methods Used:

Member Creation: librarian uses create member function to add a member with details such as name, email, and address.

Issue Book: librarian can issue book to the member choosing from a list of books.

Check Borrowed Books: librarian can check for every (one at a time) member that how many books that member has borrowed.

Fine Calculation: if a book is past its due date, librarian has the control to check which member has gone past the due date to collect fine from them.

Algorithm:

When you run the program, it gives a menu like interface to let user select what they want to do then with each input they are assisted with commands that make it easier for the user to use the system without knowing much about how the program is coded or how things work in the back end of the system.

Problems:

The code works fine for most part of the functionality requirements, I had issues with makefile and testfile which I have attempted to make but I don’t feel confident about that but I have still uploaded them in the repo. My github account had some authentication issues, due to which I wasn’t able to commit and push during initial development of the program. The issue is now resolved and since then I started adding commits to the file.

Future Use:

My approach towards the task remains the same, which is to read the documentation carefully and understand what requirement of the task is and try to stick to the functionality wanted by the task and not get carried away. I can always improve on what I have done, and I plan (after the coursework is marked) to work on it more and make a fully functional and have more features than what was requested for the coursework, as these projects will improve my portfolio and I will have things to impress my employer in the future.