Nathan Atchison

May 10, 2022

CS-172-1

Project Proposal and Requirements Specification

**What will I be doing for my project:**

The primary focus of this project is to develop a library database. Within this database, there will be two primary modes: a guest mode and an admin mode. In the guest mode, it will be like you are someone who has come into the library to find a book, check out a book or return a book. As for the admin side, it will allow someone to be able to do all the “behind the scenes” actions of updating the database such as removing books that will no longer be available or adding new books to the database that can be checked out.

**How do I plan to approach this project?**

For me to feel that my project has worked, I feel that there are a couple of key aspects that must be successfully implemented. First, I believe that the system must be user friendly. Whatever is shown to the user must make it clear what does what. For example, I want to interface for the database being accessed by a guest to clearly show how to return a book or check out a book. It needs to be clear so that anybody can use it. Another thing that I want this system to do is to make sure that there are separate actions that specific interfaces can take. For example, I want the guests to be able to be able to check out books or return books while not allowing them to completely remove a book from the system, as that is only an action that an admin should be able to do. Specific users should only be able to do specific things, and that’s how I want my system to work. Finally, I want a system that gives detailed information about the books being stored in the database. I want a system that will keep track of what books are available, how many are available, who has checked out what books and detailed information that adequately describes the books in the database. While not perfect, I want a database that can pass as something that a library could use effectively to keep track of what books the library has. There are a couple assumptions that come with this last aspect to make a successful system. The way I plan on keeping track of the books in the library is through file I/O. This means that if access to the file holding the information within the system is lost, all of the information for the system is lost as well. We assume that a system will always have access to these files, as without them the system would not function. We are also assuming that the files which will contain these information are already created before the program even first starts. The code will not create the file itself, it must be able to access the files and change them when necessary. Finally, we assume that someone who is using admin privileges will not have any use for actions that are restricted to the guests. For example, we assume that someone who is logged in as an admin is not there to check out or return books. They are there for the sole purpose of either removing/adding a book to the system or just doing a routine check of all the books in the system.

**UML Class Diagram**

Diagram

Description automatically generated

**Note:** All getter/setter functions as well as overloads are not included in the methods listed above. However, almost all classes have at least some getters/setters.

**Additional Information about process:**

On top of all of the class methods above, there will be several functions that will be created to be able to keep the database open until the user closes it down. The database is primarily for holding information that is being used throughout the process of the database being accessed however, it doesn’t do anything on it’s own. Finally, while the user class allows all of it’s subclasses to access actions of checking out a book and returning a book, the method will not allow it’s subclass of Guest to check out a book, because only registered users (or admins, who are in turn registered users) may access the ability to check out a book. Also, the return types of the class methods are subject to change as I create the functions that will allow the library database to function.

Src for writing a UML Diagram:<https://www.youtube.com/watch?v=UI6lqHOVHic&t=291s>