SOFTWARE ENGINEERING CW1 REPORT AND PRESENTATION- ARFANUL SHOUID BEGUM M00915474

INTRODUCTION

The Library Management System is a robust solution designed to efficiently organize and track available books and members in a small library setting. Utilizing a flexible data source in the form of a CSV file, the system allows the librarians to have access to user-friendly functionalities to manage member information, book transactions, and overdue fines seamlessly.

UML Diagrams:

A screenshot of a computer program

Description automatically generated

DESIGN:

This is the UML diagram we have based our project as it gives us each of the classes that we had to use in our project.

This are my use case diagram and activity diagram:

A diagram of a diagram

Description automatically generated

A diagram of a flowchart

Description automatically generated

The use case diagram shows the functionality of the system displaying how actors play a role in the system and interact with it. The diagram shows the functionalities that the librarian can perform like add members, issue books, return books, display books that are borrowed and display all books in the csv file. It also shows how the other actors are linked to each other and play a part related to the system.

The activity diagram visualizes the model for what the system should perform like. It helped me build my code as I used it like a blueprint and looked at it to help me understand how my code should work and what features it requires. It is good practice to build an activity diagram as it helps to visualize the workflows, processes and necessary steps needed to take to build a robust project.

My activity diagram shows how the program starts and how it follows according to the inputs given to it by the user/librarian. Program starts with the user having to type the csv file name for the program to locate it and use it as the list of books that can be borrowed by members of the library. If the file isn’t found the program tells the user to input the file name again. Once the correct file is inputted the programme is set in motion and the librarian has the freedom to use the menu according to their needs and responsibilities like add members, issue books, return books, display books that are borrowed and display all books in the csv file. There are also validations used in case inputs are not given correctly like the program not being able to issue a book to a member that does not exist, or the program not being able to display a book that has never been borrowed. Program will tell user to use the program correctly in a way that makes sense.

MAKEFILE

It links and compiles together all your files into one final program that you can execute.

VERSION CONTROL

I have uploaded my code to GitHub and each time that I have made any significant changes I have reuploaded it again with commits which describe very concisely, what I have uploaded. I did this using the commands: git status, git add, git commit and git push. When I am working on a project, the making of the code takes time and goes through changes to better suit my needs and wants. Which is why version control is used to update my GitHub on those changes.

A screenshot of a computer

Description automatically generated

IMPLEMENTATION/SOFTWARE DEMONSTRATION

The way I approached and translated the design was by having different classes like librarian, member, person and book like the UML diagram.

When we talk about the person class, we represent basic personal information to represent the individuals in the library system. With attributes like name, address, email. We can assign the personal information and access to it. The member class derives from the person class, and it represents the library members added to the system. Each member has their unique id and book id that is stored and can keep track of borrowed books, the functions in the code help and provide information about borrowed books and the members. The librarian class represents the librarian or the user in the library system, it’s the central core of the program and it interacts with members and books, it has the functionality of adding members issuing and returning books and handling files. The book class are the books in the library system, it stores book information and it’s related to the aspect of books being borrowed. It manages the libraries stock and tracks the borrowing of books. The main.cpp file carries the logic for the library management system, it interacts with the librarian class and has the main functions which it embodies to push forward the flow of the system.

CONCLUSION:

What I have developed in my project is library management system that has functionalities that allow for a librarian to be able to input csv file names into the program to read the books stored in the file, add people to the library management system so they become members, issue members books from the file, take back books and return to the system, display books that are borrowed from the system and also display all the books in the files.

The limitations of my work were that I wasn’t able add a feature that gave a return date to books borrowed which if they went past the three days mark the member would be issued a fine. The reason for this is because of the tight schedule where I run out of time to develop that feature and because of being ill. The next I approach a project like this I will make sure to give myself plenty of time to do this project and be more proficient and work smarter to achieve my goals.

Thank you.