Text

Description automatically generated with medium confidence

**CST2550**

**Software Engineering Management and Development**

**Coursework 1**

**Autumn/Winter term**

**2023/2024**

**Date of Submission: 16/01/2024**

**Student ID Number: M00885480**

**Lab Tutor: Aditya Santokhee**

# Introduction

In this project I have created a library system to read a csv library file create members, issue books to members, handle returns etc.

I created a this program solely in C++ with the aid of some of its standard libraries. Through this report I will highlight the key aspects of the project, including but not limited to, the design, implementation, testing, what I found challenging, and lessons I learned.

# Design

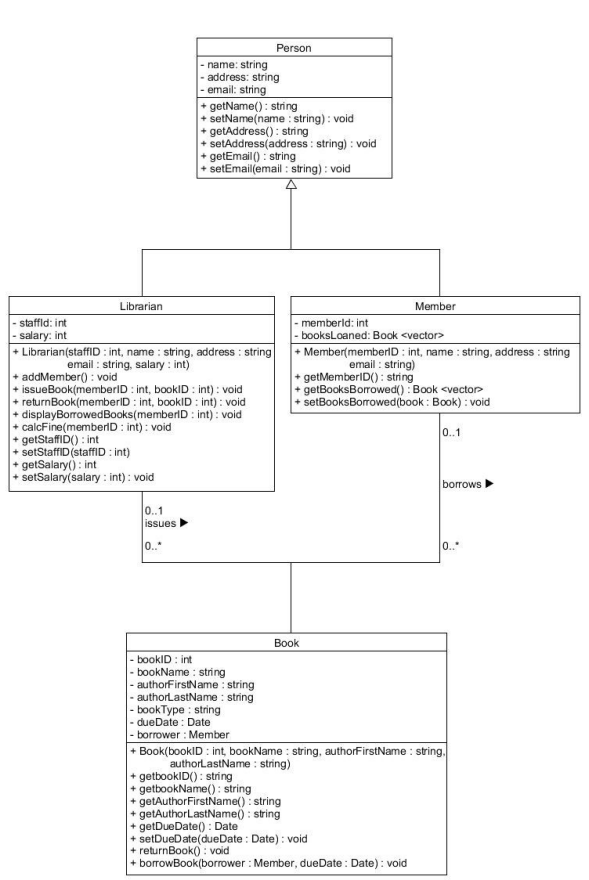


Figure Class diagram

Using the Class diagram provided in figure 1, I was able to successfully create the four classes and header files with all the described functions and functionality.

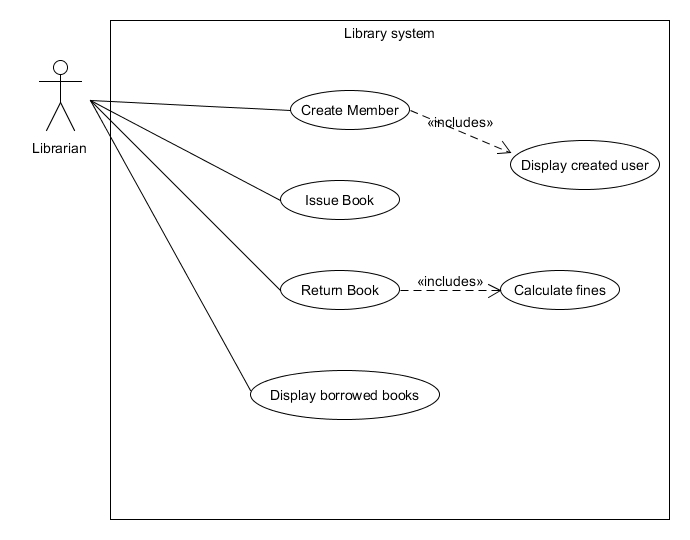


Figure Use case diagram

The use case diagram shows the actors involved in operating the Library system and the functionality they employ.

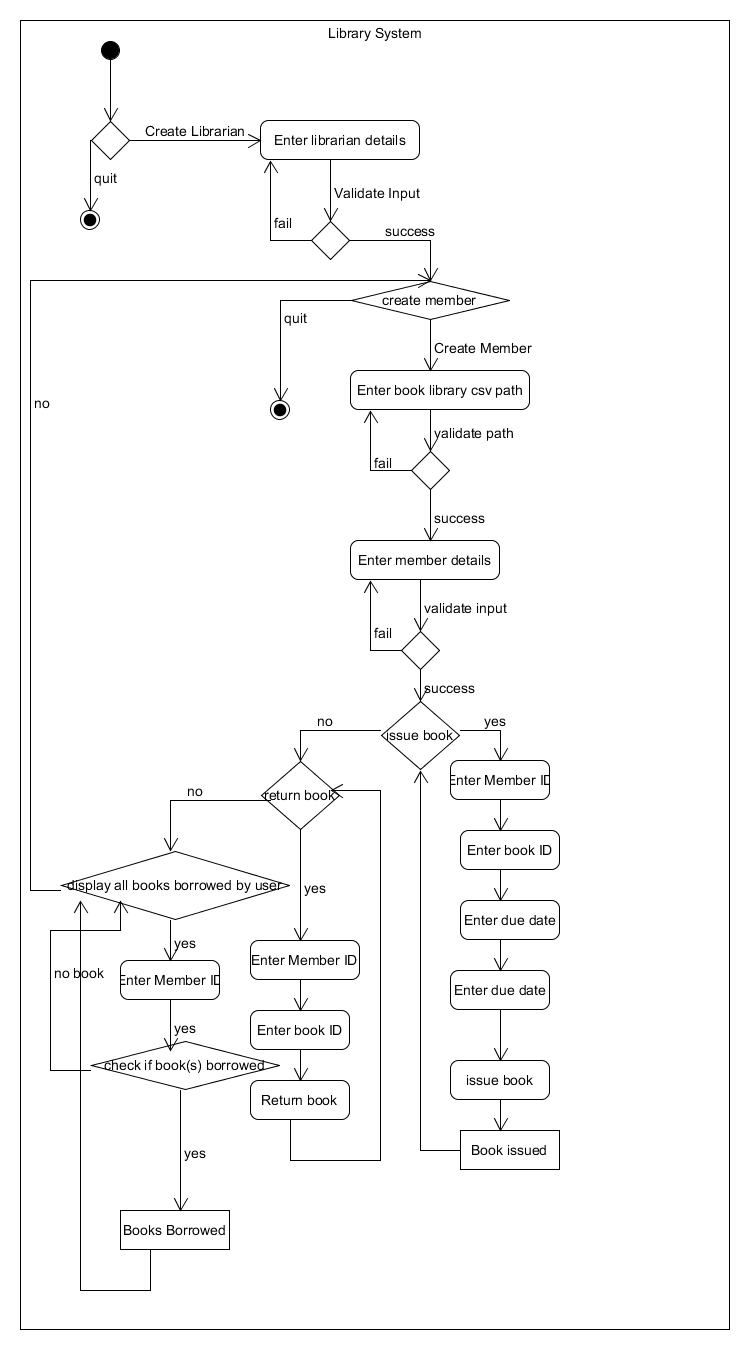


Figure Activity diagram

The Activity diagram above in figure 3 shows the flow of the Library system and how it should operate.

By using these 3 diagrams I was able to successfully complete the program and have it run as expected. The design process helped me smoothly transition from one task to the next without having to waste my time resource.

# Implementation

To compile a project of this size, with multiple files and to avoid circular dependency errors I use a makefile.

The makefile also includes multiple c++ flags to catch multiple compile errors.

I used GitHub for version control to manage file organization, back-up the program, keep track of modifications to the files and enable collaboration.

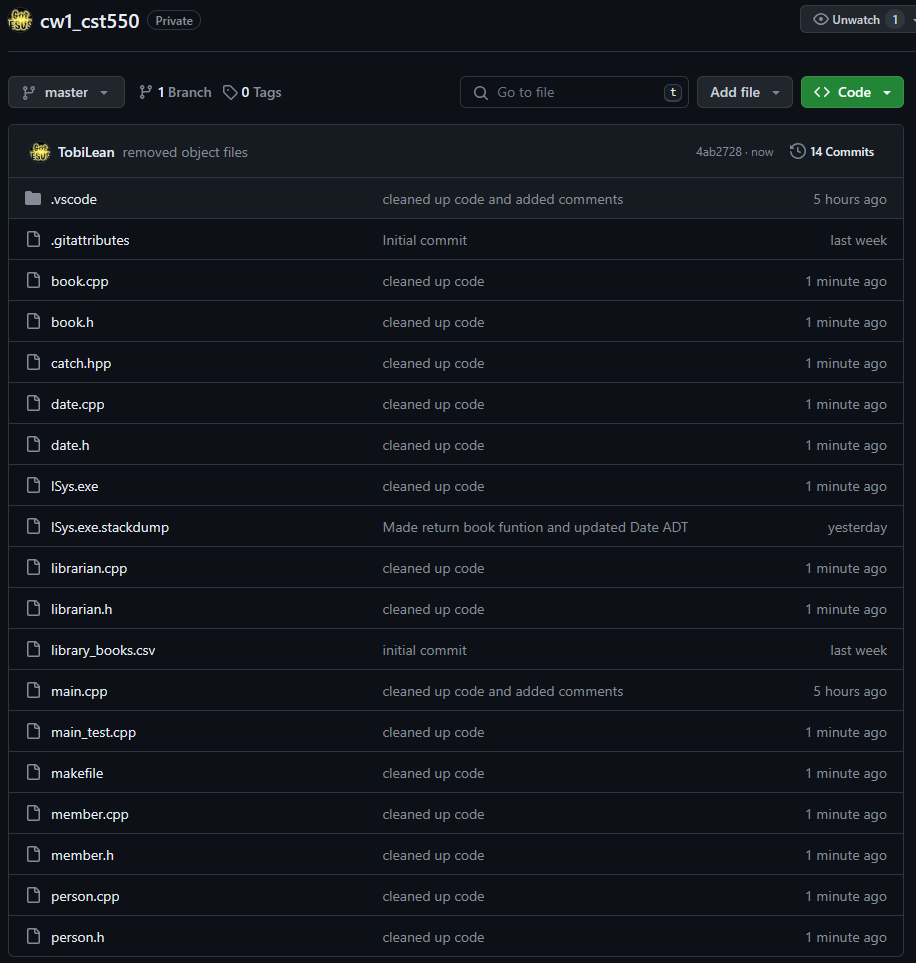


Figure git repository

# Testing

For testing I used the catch2 testing framework.

I made a test file to test all the constructors of each class and some of their member functions. I used a makefile to compile the program’s files and run the test. After the tests were run, I got some errors and used the information given to fix the program until no test cases failed.

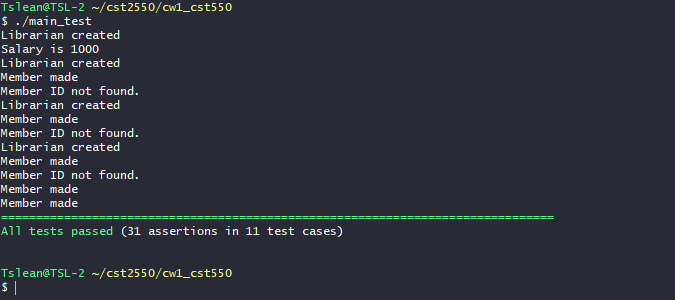


Figure test result

# Conclusion

To conclude, I reflect on the project and would point out that, although challenging, this project helped me grasp the fundamentals of OOP in C++.  
Some area I struggled with were with pointers, return types and copy elision.

I found that during the later stages of the project having a to-do list to summarize my tasks for the project increased my efficiency and performance. Something I would change on my next project is employ solutions like a to-do list earlier in the project life cycle.