CST2550 Coursework

Name : Thanesha Hemrage

Student ID : M00911454

Agenda

- Introduction
- UML Diagrams
- Implementations
- Testing
- Conclusion

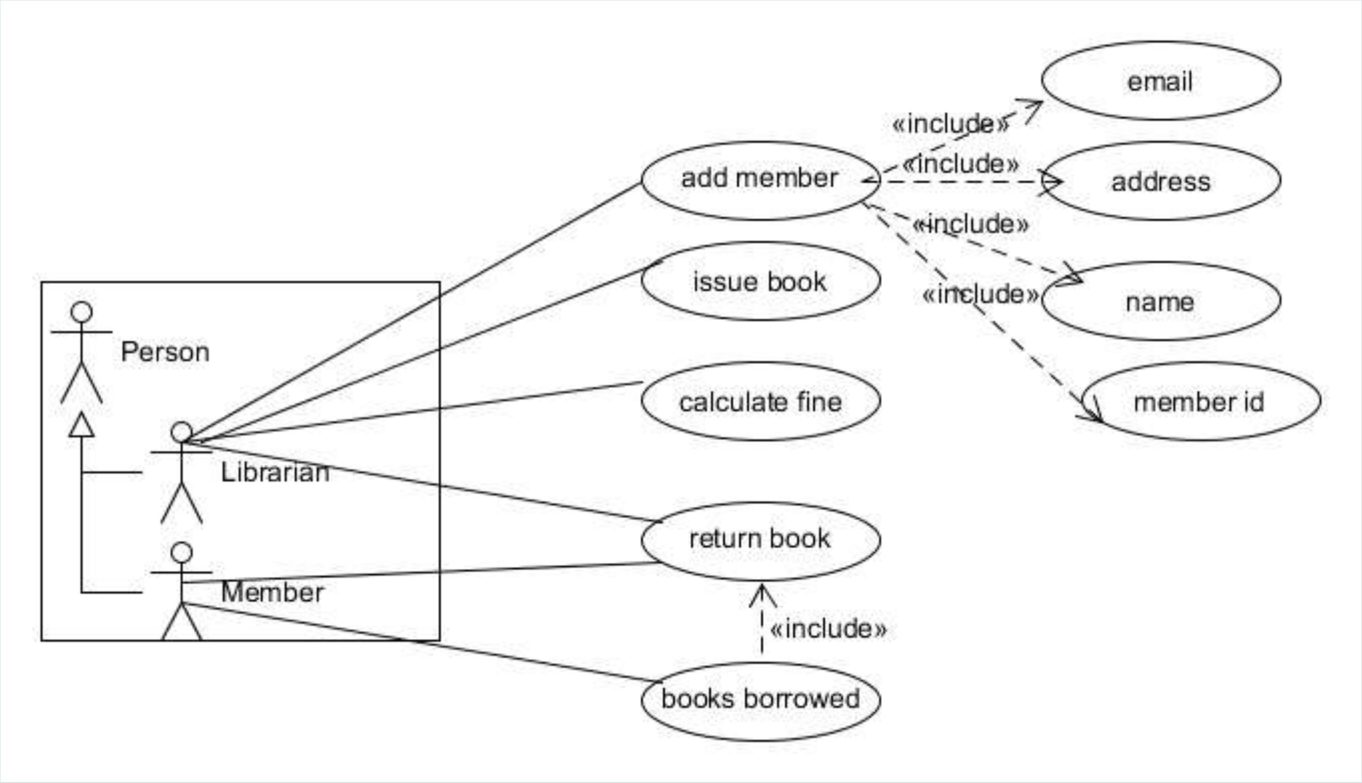
Introduction

- The project focuses on a small library that requires a system to manage the details about its members and the books that are provided. The librarian will handle all of the tasks, including adding new members, issuing books, collecting books that are returned from members, displaying all member book returns, and collecting fines from members who have past-due books.

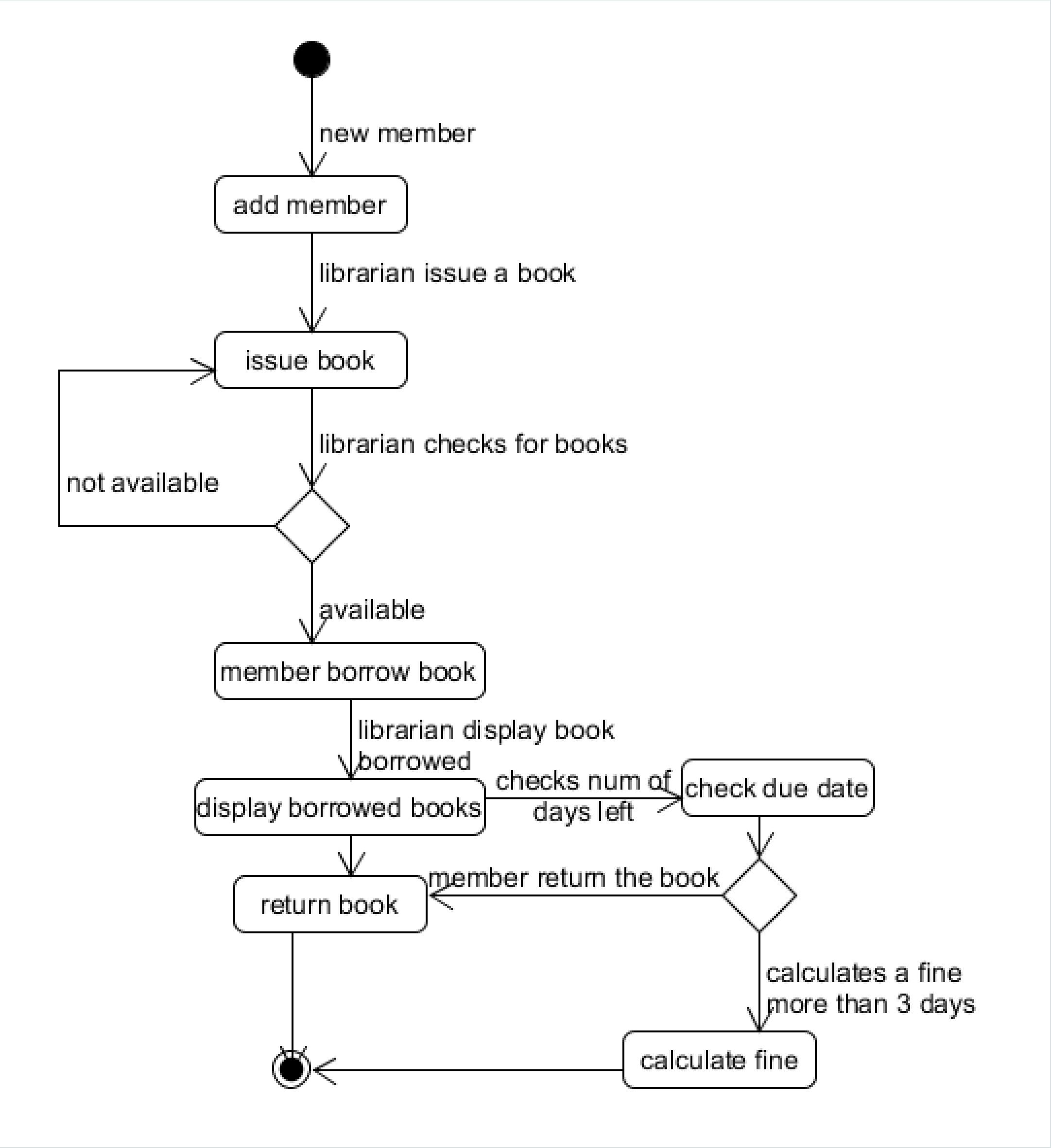
- Therefore, the UML diagrams for the library system, the way they are implemented, and the testing strategy will all be covered in this presentation.

UML Diagrams

USE CASE DIAGRAM



ACTIVITY DIAGRAM

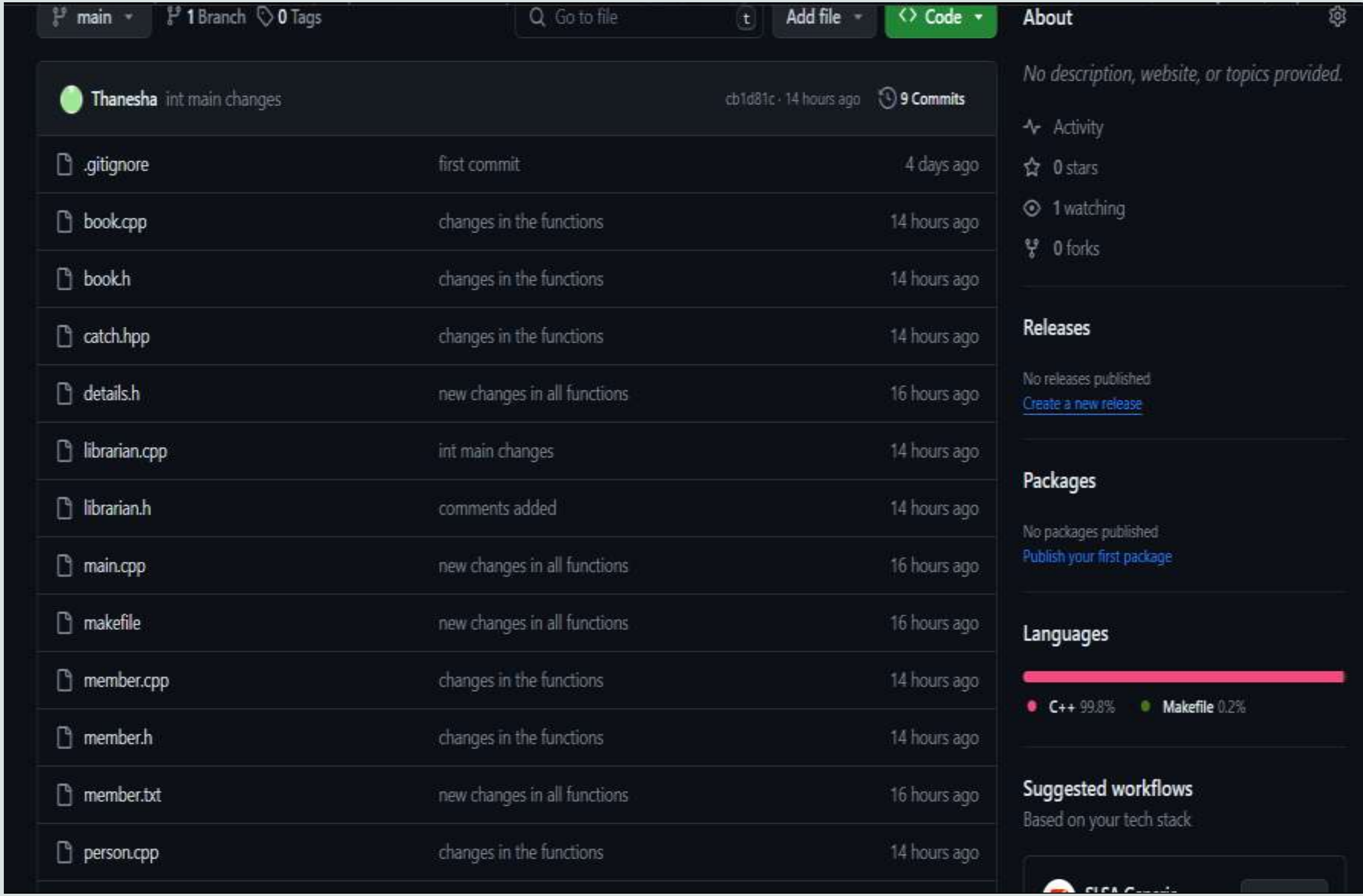


Implementations

- This library system was developed using an incremental and iterative process. I began by implementing the Person, Member, Book, and Librarian classes. The main elements in the library system were represented by these classes I next created methods for adding new members, issuing books, collecting them , displaying the books that were borrowed, and calculating fines as these methods corresponded to the main responsibilities of the librarian in the library system
- The library system's building process approach was automated with the help of the Makefile. I was able to construct the library system with only one command because of the Makefile, which enabled me to provide the compilation instructions for each source file and the linking command to generate the final executable.
- I used Git version control to keep a consistent history of changes. This made it possible for me to keep track of modifications made to the source code, revert over to previous iterations as needed

Implementations

- GitHub Repository:



Thanesha int main changes cb1d81c · 14 hours ago 9 Commits

.gitignore	first commit	4 days ago
book.cpp	changes in the functions	14 hours ago
book.h	changes in the functions	14 hours ago
catch.hpp	changes in the functions	14 hours ago
details.h	new changes in all functions	16 hours ago
librarian.cpp	int main changes	14 hours ago
librarian.h	comments added	14 hours ago
main.cpp	new changes in all functions	16 hours ago
makefile	new changes in all functions	16 hours ago
member.cpp	changes in the functions	14 hours ago
member.h	changes in the functions	14 hours ago
member.txt	new changes in all functions	16 hours ago
person.cpp	changes in the functions	14 hours ago

No description, website, or topics provided.

0 stars

1 watching

0 forks

No releases published

Create a new release

No packages published

Publish your first package

● C++ 99.8% ● Makefile 0.2%

Based on your tech stack

Testing approach

- I utilized test-driven development in this library system. With this method, test cases were created for each function before the corresponding code was placed. By using this strategy, I made sure the library system operated accurately and effectively.

- I utilized test cases to verify that the methods and attributes used in the Person, Member, Book, and Librarian classes were valid. Also, if the files were valid.

- I used a testing framework like Catch2 for C++ to run the test cases. With the help of this framework, test cases could be easily executed and comprehensive reports on the success or failure of each test could be produced.

Conclusion

- The program allows the librarian to manage books and members, also issue books to members, adding new member and many other functions. The program handles edge cases, such as members reaching their book borrowing limit and ensuring that only available books can be issue.

- In the future, this project could be improved by adding a graphical user interface (GUI), adding additional search capabilities, and growing the system to provide other functions like book reservations and renewal alerts..

Thank you