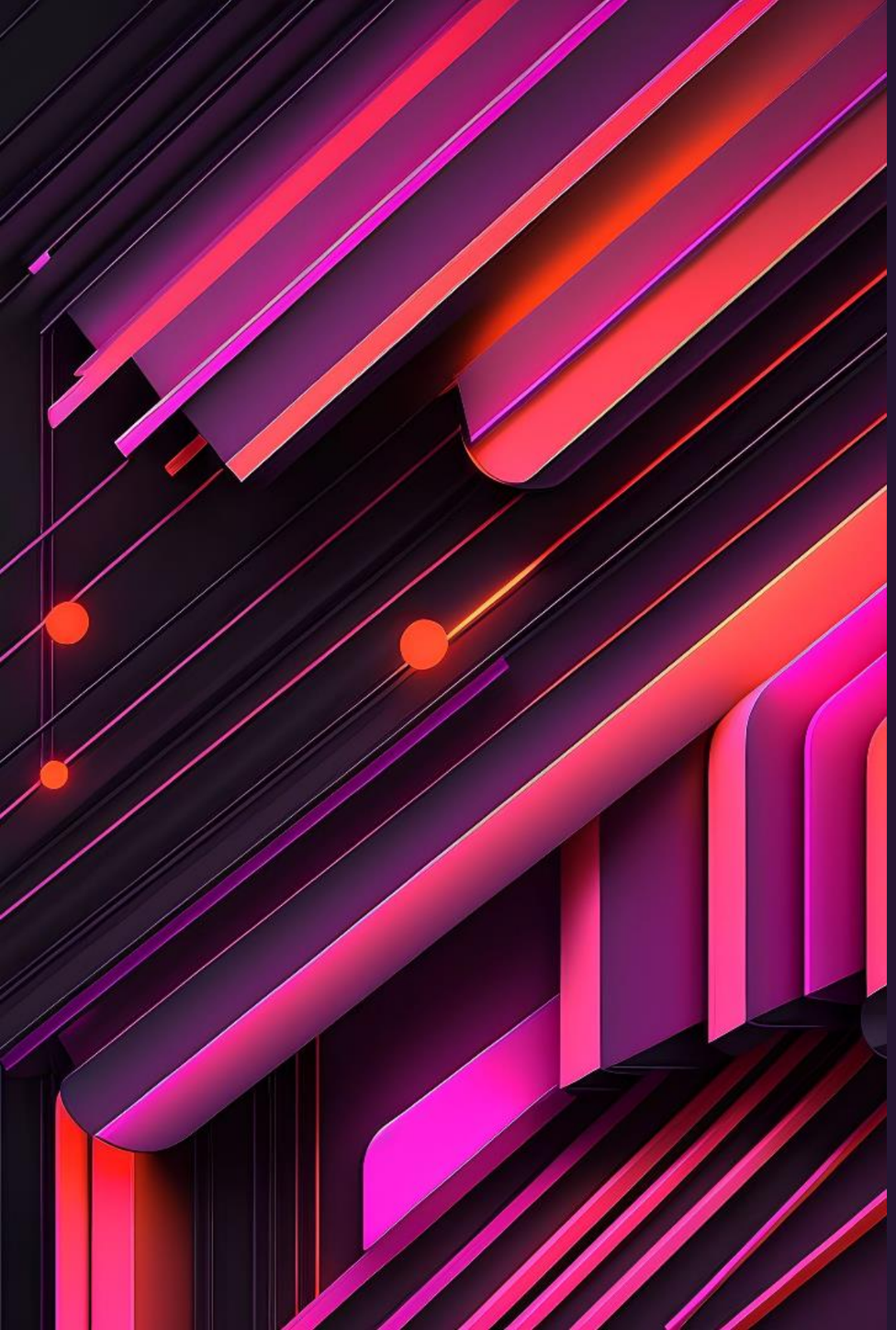
library management system

- M00968419

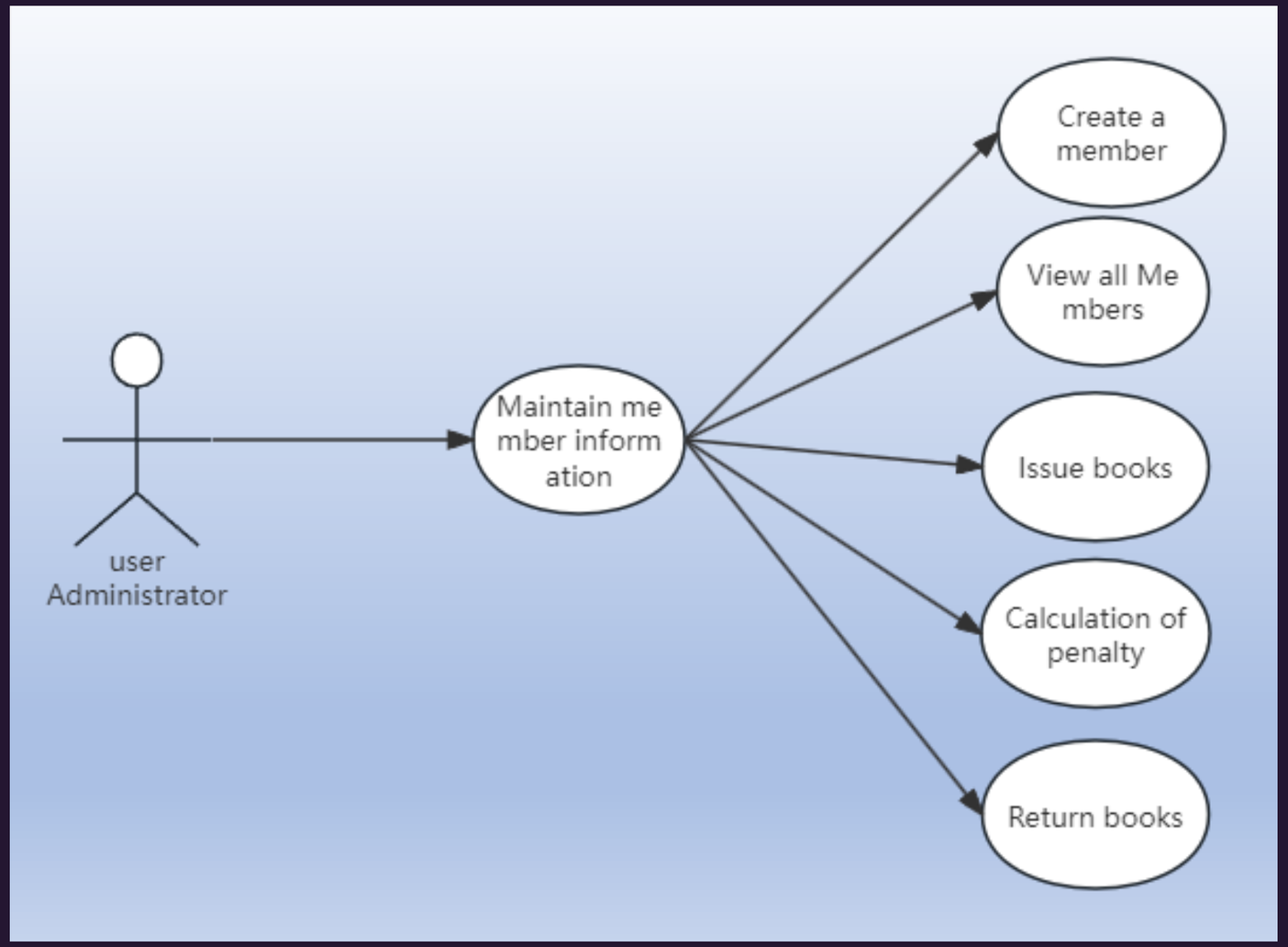


Introduction

Project Description: This is a library management system specifically for administrators. It allows administrators to add information, record and manage book information, and return books. Administrators can add members, issue books members, and handle book returns. The system also for displaying books borrowed by members and calculating fines. Written in C++, it has a simple and intuitive console operates through digital options to ensure the accuracy of input. This system is for administrator use only and focuses management of books and member information. It does not personal privacy or payment information.

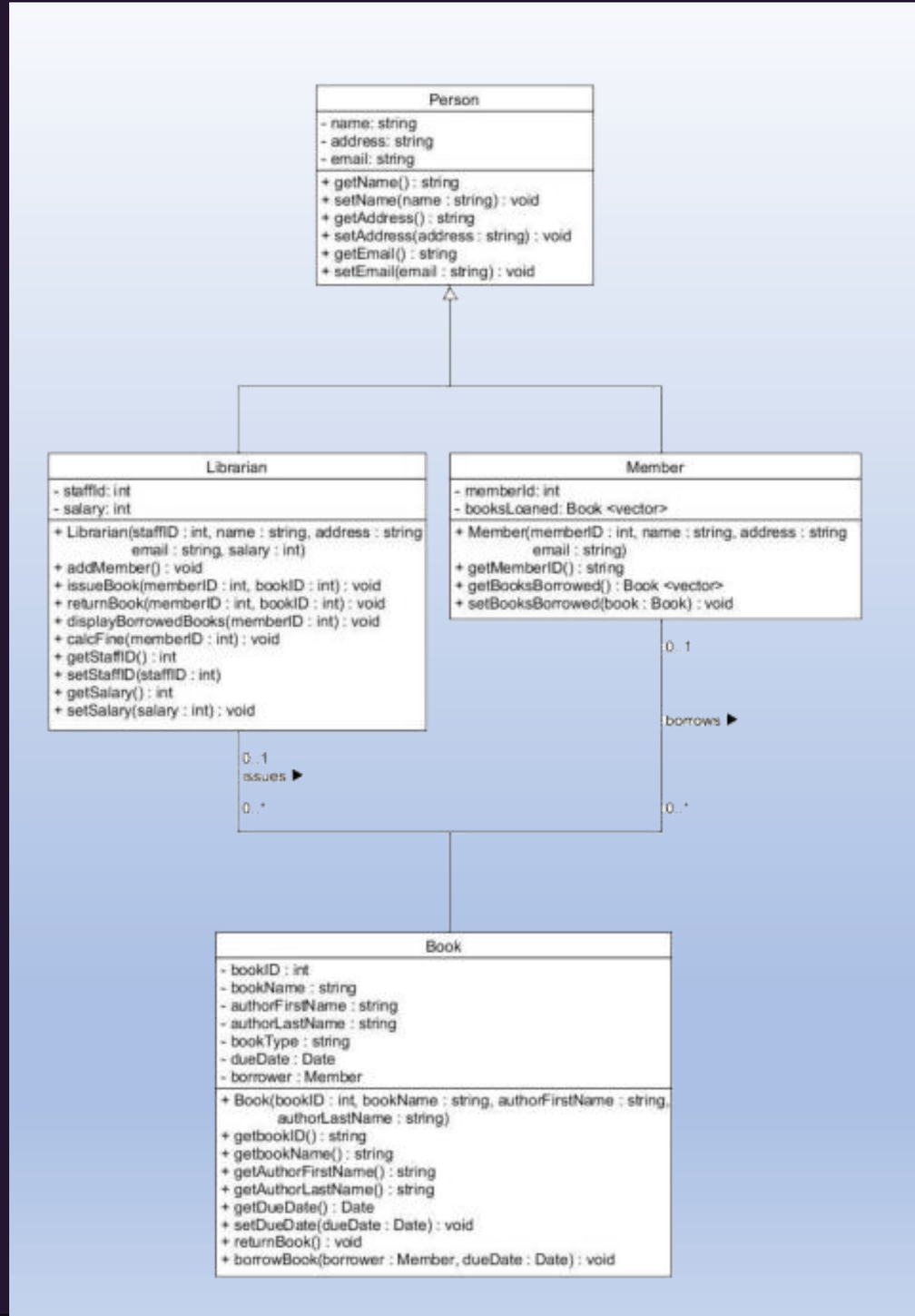
UML diagram-Use Case Diagram

The users of this library management system are librarians. The librarian can create a membership account for the member, record the member's basic information, view all member information, and then issue a book to the member with a 3-day borrowing time limit、the administrator can calculate the member's fine amount at any time when the member returns the book, and finally the administrator can return the book.



UML diagram-Class Diagram

According to the use case diagram of this library management system, three classes can be abstracted. These classes are librarian users, library members managed by librarians. Then based on the common attributes of these two classes, it is used as the base class Person class. Finally, there is the book information class of the library management, which facilitates the issuance of books to users, records the user's borrowing time, etc.



UML diagram-Activity Diagram

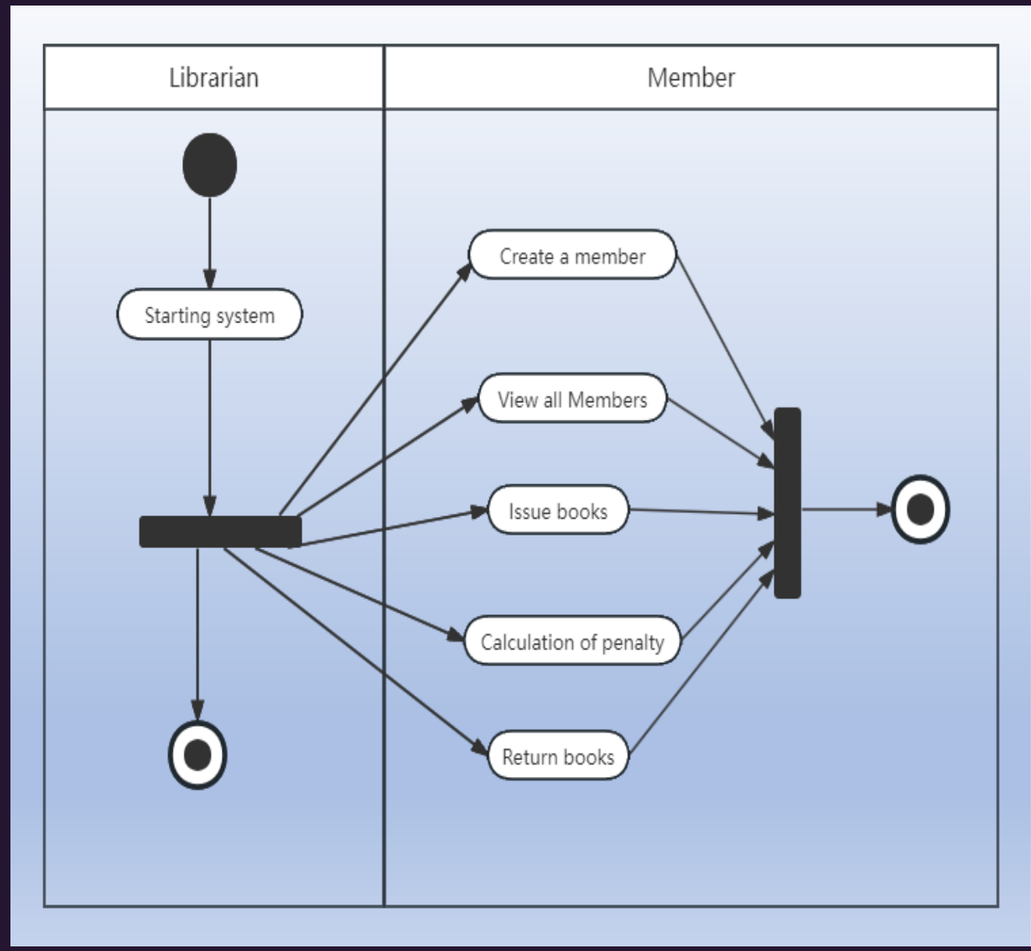
1.Create a member account
- Librarians may choose to create a new membership account.
- The administrator enters the member's basic information (such as name, address, email, etc.).
- The system records and creates a new member account.

2.View member information
- Librarians can view a list of all members'information.
- The system displays basic information including member name, ID, etc.

3.Release books
- The librarian chooses to issue a book for a specific member.
- Enter the member ID and book ID to set the borrowing period.
- The system records the loan status and expiration date of books.

4.Calculate fine
- Administrators can choose to calculate a fine when a member returns a book.
- Enter member ID and the system will calculate the fine amount for overdue books.

5.Return books
- The librarian chooses to return the book.
- Enter the member ID and book ID to mark the book as returned.
- Update borrowing records and calculate fines.



System implementation

1.The implementation of the system is indispensable without the assistance of UML tools. Here, the following classes are created based on the UML class diagram of the previous design.

- Book class: includes book attributes and methods, such as book title, author, type, loan status, etc.
- Member class: stores member information and a list of borrowed books.
- Librarian class: Administrator's attributes and operations, such as adding members, issuing books, returning books, calculating fines, etc.
- Person class: As a base class, it stores shared user information, such as name, address and email.

2. Mapping of use case diagrams to functions

Functionalization of use cases: mapping use cases in use case diagrams to functions in code. Each use case usually corresponds to one or more functions to implement its functionality.

Here the following methods are implemented based on the system use case diagram ：

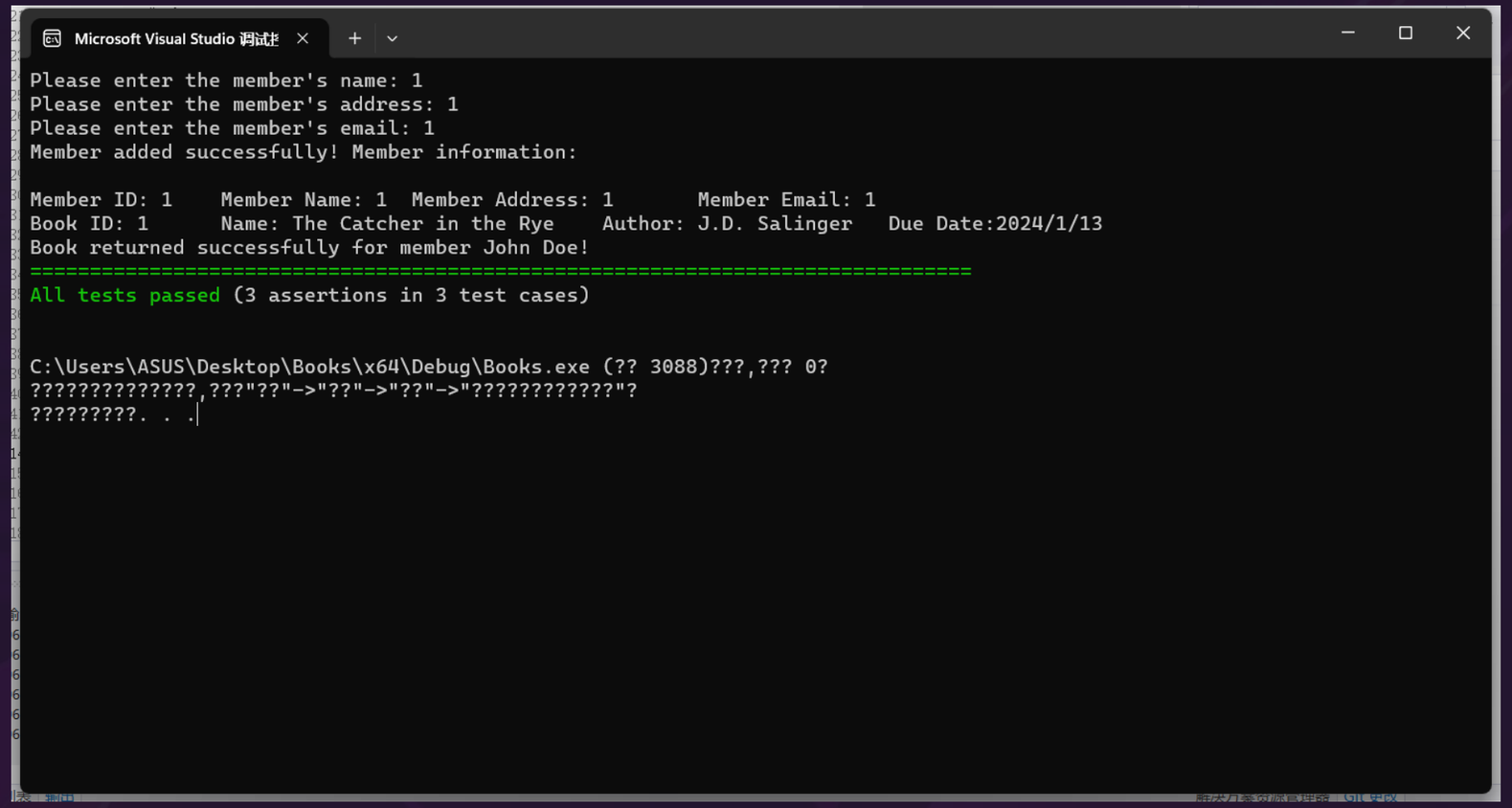
addMember、issueBook,、returnBook,、displayBorrowedBooks,、calcFine, etc.

3. Implementation from activity diagram to method

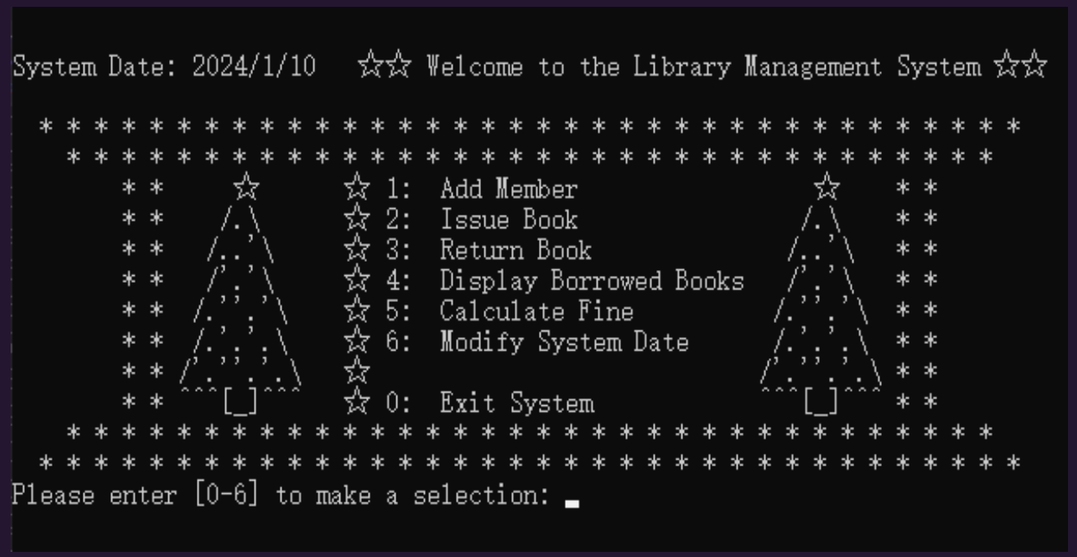
Convert activities to methods: Convert the processes in the activity diagram into actual methods. These methods represent various operations performed by objects or users in the system

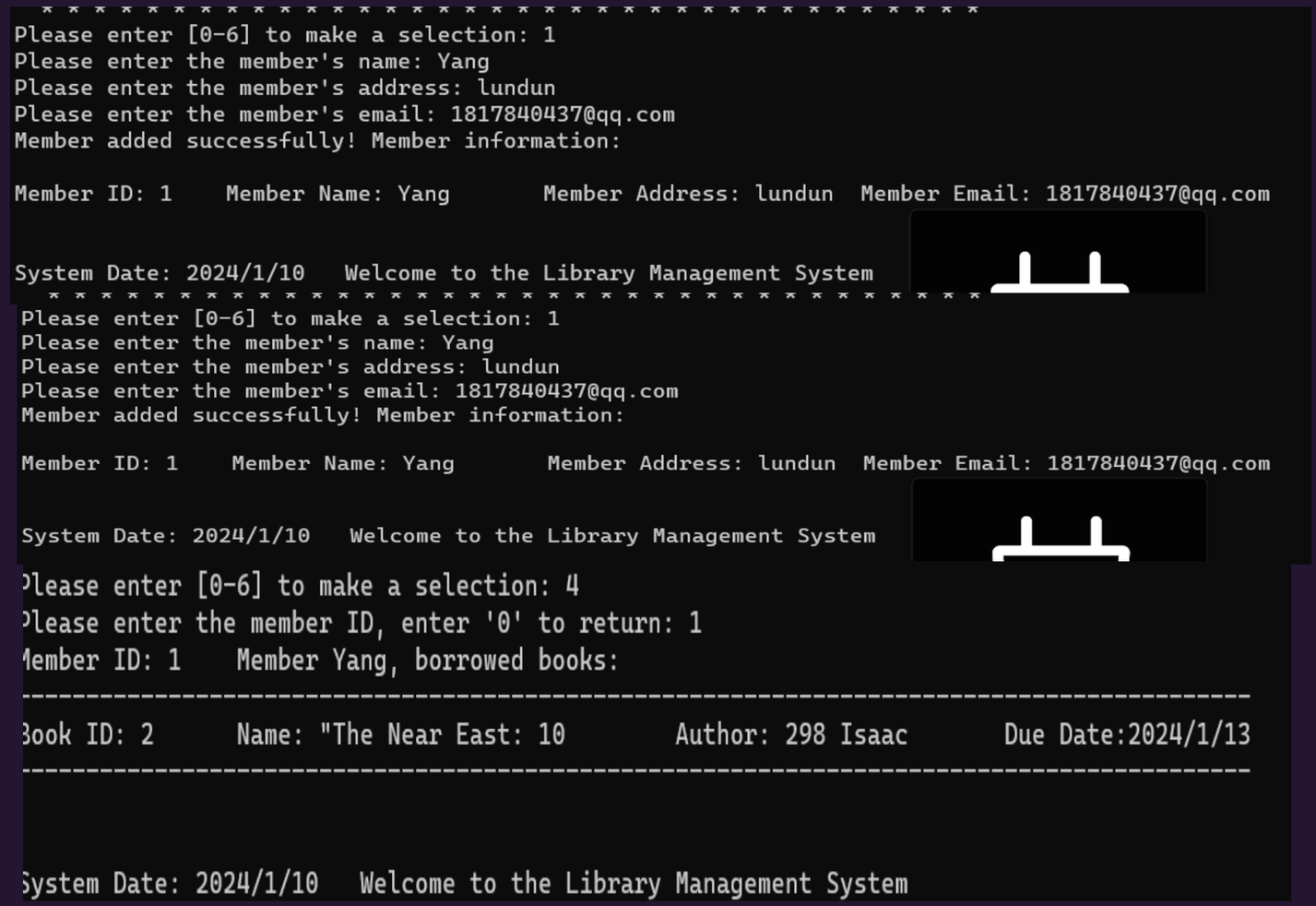
System testing

If we want to execute the following test case, comment out the main function. In order to facilitate system testing, I added a method myself. Through this method I can modify the time variable I created. In my program, this method is used as system time to calculate the fine amount. In the right is a screenshot of the test.



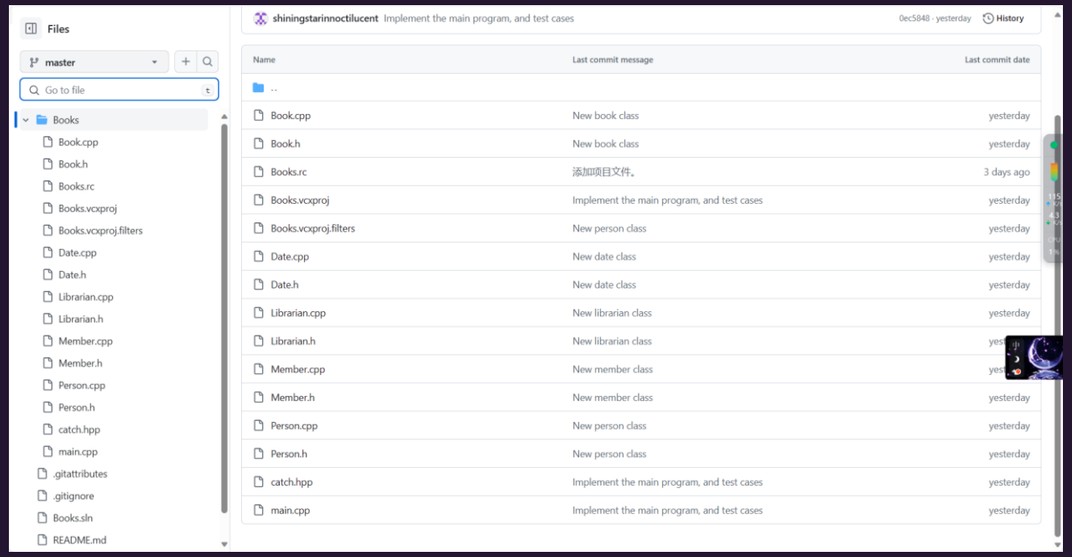
system demonstration screenshot

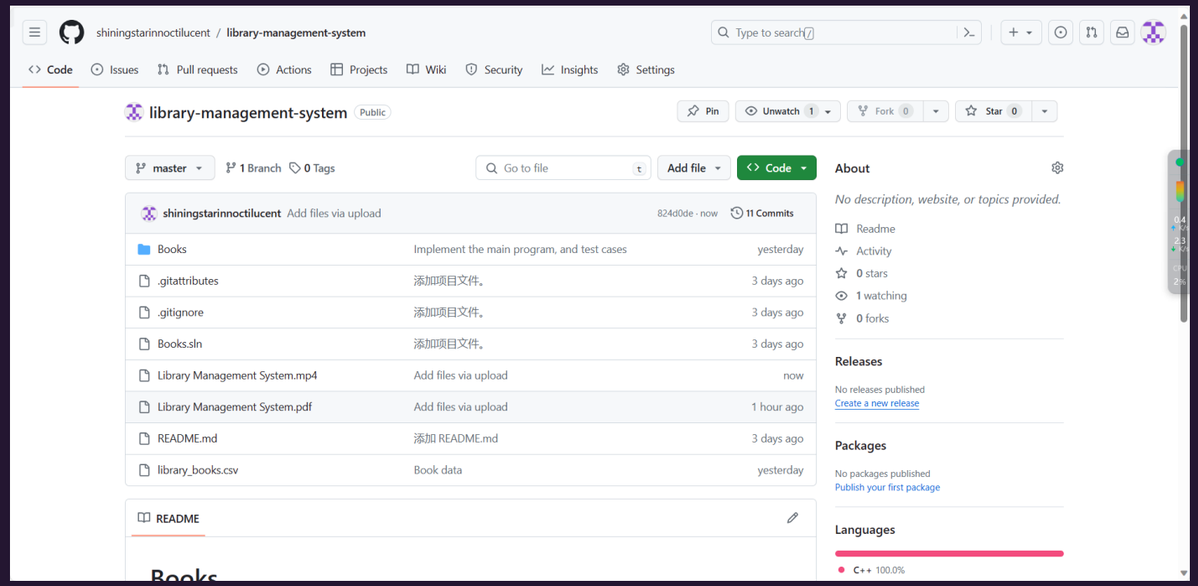




This is the system I implemented, which allows librarians to interact through the console. The system performs corresponding operations based on the librarian's input of different numbers 0-6.

Screenshot about github





conclusion

In this project, I designed a library management system to implement basic librarian and member management functions. The base class Person is used, and two classes, librarian and member, are derived, as well as a book information class. Associations are established between these classes, conforming to the design of UML diagrams.

The main challenge in the work is to ensure the correctness of the relationships between classes and the implementation of functions. The implementation of the code needs to ensure that each class and method performs its function correctly. When dealing with similar projects, I will pay more attention to the needs analysis of the project, ensure that more details are taken into account in the design phase, and pay attention to the readability and maintainability of the code when writing code.

The system is currently fully functional, but there may still be some potential room for improvement. For example, you can consider adding more error handling mechanisms to improve the stability and robustness of the system. In addition, you can consider adding more administrator operation functions, such as book inventory management, statistical analysis, etc., to further improve the functionality and practicality of the system.