Introduction:
Name: Yehya Juma
Student # M00862523

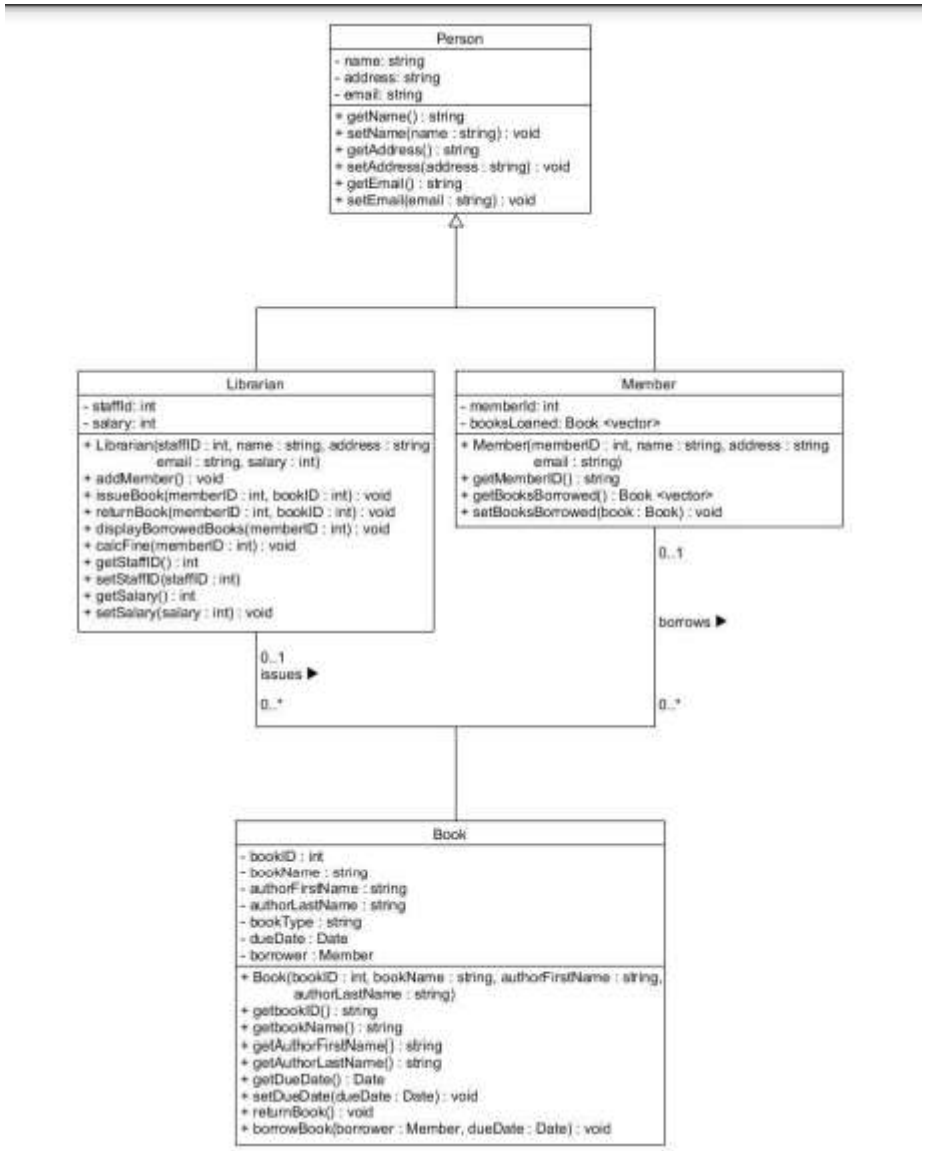
Library Management System

Design

UML Diagrams

Class Diagram:

- Person Class: Represents common attributes like name, address, and email.
- Member Class: Inherits from Person, includes member-specific attributes and methods.
- Book Class: Represents a book with attributes like bookID, bookName, genre, dueDate, and methods for borrowing and returning.
- Librarian Class: Inherits from Person, represents a librarian with staff-specific attributes and methods..



Implementation

Translating Design into Software

To implement the library management system, we followed the design specifications closely. We used a modular approach, breaking down the system into smaller components that could be developed and tested independently. This allowed for easier debugging and maintenance.

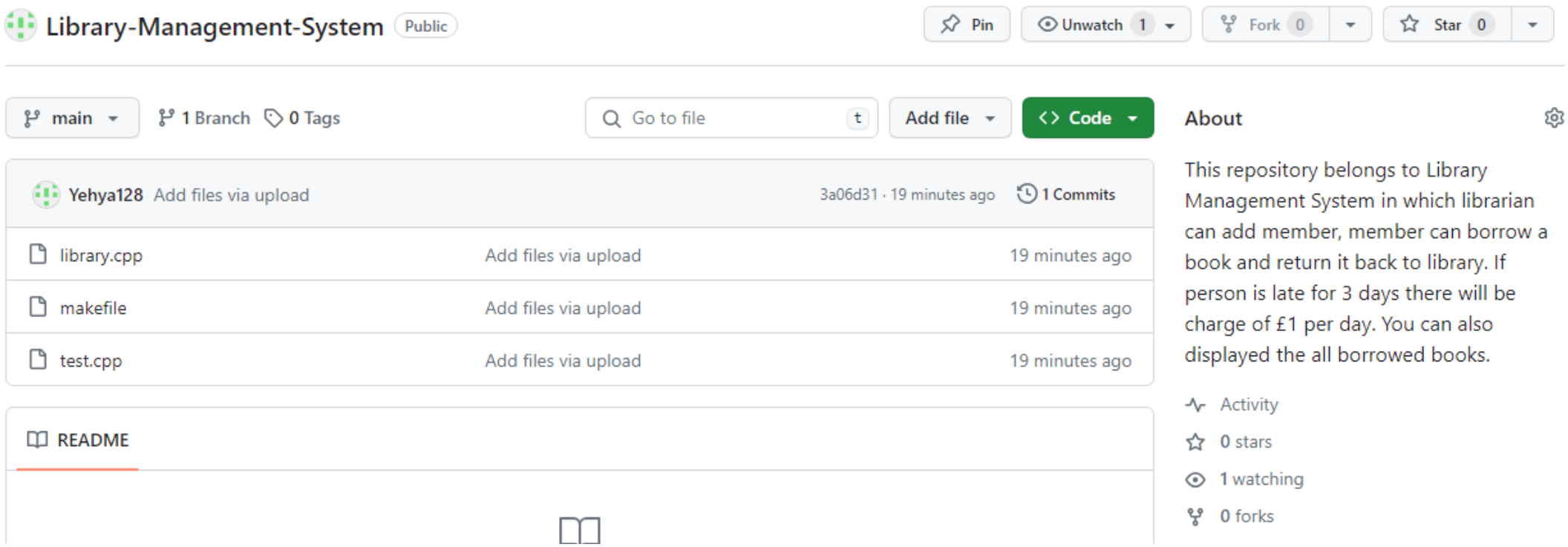
Makefile

We utilized a Makefile to automate the build process and manage dependencies. The Makefile included commands for compiling the source code, linking libraries, and generating the executable file. This streamlined the development process and ensured consistent builds across different environments.

Version Control

We employed version control using GitHub to track changes and collaborate on the project. This allowed multiple developers to work on different features simultaneously while maintaining a centralized code repository. We followed best practices, creating branches for new features and merging them back into the main branch after testing and code review.

GitHub Repo:


Library-Management-System
Public
Pin
Unwatch 1
Fork 0
Star 0
main
1 Branch
0 Tags
Go to file
t
Add file
Code
Yehya128 Add files via upload
3a06d31 · 19 minutes ago
1 Commits
library.cpp
Add files via upload
19 minutes ago
makefile
Add files via upload
19 minutes ago
test.cpp
Add files via upload
19 minutes ago
README
About
This repository belongs to Library Management System in which librarian can add member, member can borrow a book and return it back to library. If person is late for 3 days there will be charge of £1 per day. You can also displayed the all borrowed books.
Activity
0 stars
1 watching
0 forks

Testing Approach

Unit Testing

To ensure the functionality and correctness of individual components of the library management system, we conducted unit testing. This involved testing each module and function in isolation to verify that they produce the expected output for a given input. Test cases were designed to cover different scenarios and edge cases, such as invalid inputs and boundary conditions.

Integration Testing

Once the individual components were tested, we performed integration testing to verify the interaction and compatibility between different modules of the library management system. This involved testing the flow of data and communication between modules, ensuring that they work together seamlessly. Test cases were designed to cover different integration scenarios and identify any issues or inconsistencies.

System Testing

To evaluate the overall functionality and performance of the library management system, we conducted system testing. This involved testing the system as a whole, including all modules and their interactions, to ensure that it meets the specified requirements. Test cases were designed to cover different user scenarios and simulate real-world usage of the system.

Software Demonstration

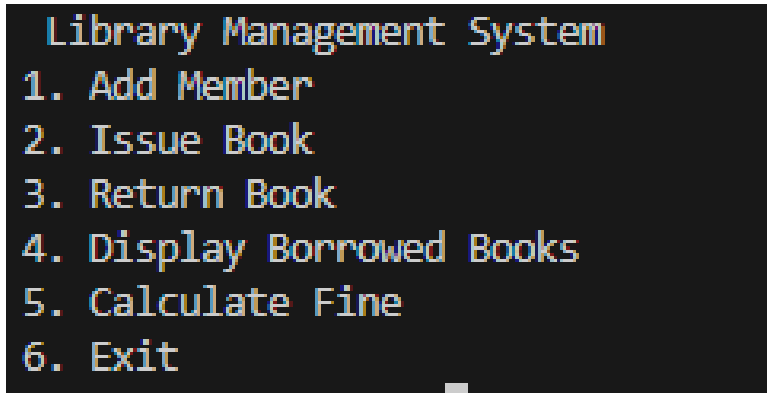
Implementation of the Library Management System

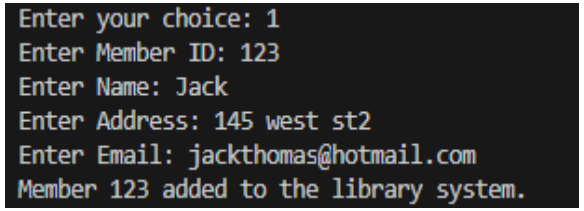
Our library management system has been implemented with a user-friendly interface.

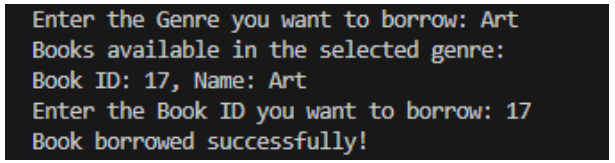
Program Functionality

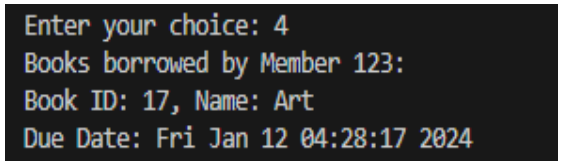
During this demonstration, we will showcase the following key functionalities of the library management system:

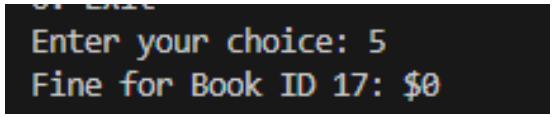
- User registration
- Book cataloging and classification
- Borrowing and returning books
- Fine calculation

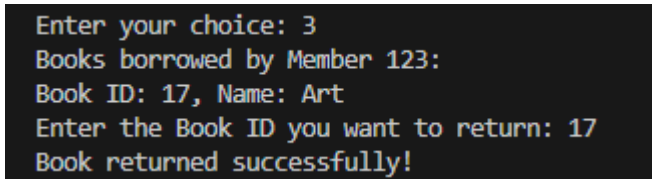












Conclusion

Summary of Work Done

Throughout the design, implementation, and testing phases of the library management system, we have successfully developed a robust and user-friendly platform for managing library resources. The system allows librarians to efficiently handle tasks such as cataloging books, managing memberships, and tracking borrowing and returning of items.

Limitations and Future Approach

While the library management system meets the core requirements and provides significant value to library operations, there are a few limitations to be aware of. Firstly, the system currently lacks integration with external library databases, limiting the availability of comprehensive book information. Additionally, the system could benefit from more advanced reporting and analytics features to provide deeper insights into library usage and trends.

In future projects similar to this, we would approach the development process with a focus on these limitations. We would prioritize integrating with external databases to enhance the book information available to users. Furthermore, we would allocate resources to develop advanced reporting and analytics capabilities, allowing libraries to make data-driven decisions and optimize their operations.