LIBRARY MANAGEMENT SYSTEM

MEMBERS:
Hardik Sachdeva - 251188614
Pavithra Gopalakrishnan - 251201688
Piyush Adya - 251235087
Piyush Piyush - 251221825

Problem Definition

The library management system is a desktop-based application developed for assisting both librarians and students in borrowing and managing books. Keeping track of a manual entry for this system is not an efficient method. By implementing a digital system with optimal storage, the librarian's work is simplified. The student can view available books and view a list of previously issued books. The application maintains a record of books, allows adding and deleting books, and allows the librarian to update which books have been borrowed and returned.

Project Objectives

- Allow Librarian to login to the system
- Allow librarian to add books to the inventory and delete books from the inventory.
- Allow librarian to issue books and return books
- Allow student to login to the system
- Allow student to view available books
- Allow student to view previously borrowed books
- Maintain clear flow between all functionalities
- Ensure it is easily installable on the librarians'' computer systems

STAKEHOLDERS LIST

1. Librarian
2. Students
3. Application Developers
4. Project Manager

SUCCESS CRITERIA

Stakeholders	Success Criteria
Librarian	The application has been developed on time
	The application fulfills all the requirements discussed
	The final application is functional
	The application is easy for the librarian to use
Students	The application has been developed on time
	The application fulfills all the requirements
	The final application is functional
	The application is easy for the student to use
Application Developers	The developers are satisfied with the results of the application
	The documentation of the development process is clearly provided which can be helpful in the future.
Project Manager	The project manager is satisfied with the results of the application
	The documentation of the development process is clearly provided which can be helpful in the future.

USE CASE DIAGRAM

Fig.1 shows the Use Case Diagram for our application. The two actors are the librarian and the student. The activities are Login, Signup, Delete Book, Add Book, View previously issued books, View available books, Issue books, Return books, and View inventory. This diagram summarizes the details of the systems' users and their interactions with the system.

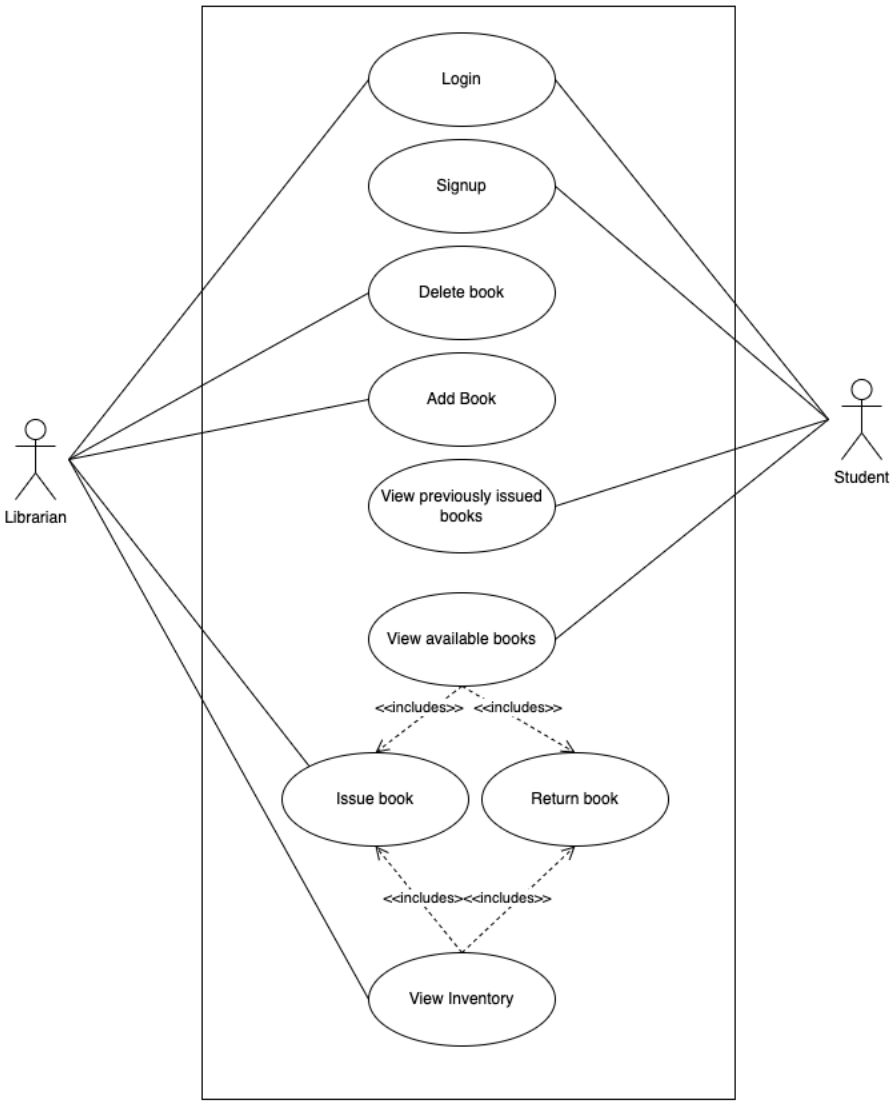


Fig.1 Use Case Diagram

USE CASE DESCRIPTION

Table.1 shows the Use Case description for the librarian side of the application. The flow of activities and conditions have been mentioned in detail.

Use Case Name	Library Management System	
Scenario	Librarian Login	
Triggering Event	Login	
Brief Description	When librarian logs in, deletes or adds books, issues and returns books, and views inventory	
Actors	Librarian	
Related Use Cases	Issue Book, Return Book (Includes)	
Stakeholders	Librarian for logging into the system Library Management for hosting the system	
Preconditions	Librarian must exist Librarian must login successfully	
Postconditions	Inventory must be updated for student viewing	
Flow of Activities	Librarian	System
	1. Librarian logs into system 2. Librarian adds book or deletes book 3. Issue books and return books from students 4. View Inventory	1.1 Verify librarian credentials 2.1 Update Inventory accordingly 3.1 Update list of books available
Exception Conditions	1.1 If Librarian does not exist, then system will not allow user to perform any updates to inventory. 3.1 If book is not available for issuing, the librarian cannot issue it. 3.3 A book that has not been issues, cannot be returned.	

Table.1 Use Case Description (Librarian)

Table.2 shows the Use Case Description for the student side of the application. The flows of activities and conditions have been mentioned in detail.

Use Case Name	Library Management System	
Scenario	Student login in or Creates account	
Triggering Event	Login	
Brief Description	When students log in or sign up	
Actors	Student	
Related Use Cases	Issue Book, Return Book (Includes)	
Stakeholders	Student for logging into the system Librarian to issue books and manage return books Library Management for hosting the system.	
Preconditions	Student must exist Librarian must exist Both actors must successfully login	
Postconditions	Inventory must be updated after issue process and return process for student viewing	
Flow of Activities	Student	System
	1. Student signs up or logs in 2. Student views available book 3. Student requests to be issued a book 4. Students returns a book	1.1 Verify student credentials or create new credentials for student 3.1 Update list of books available 4.1 Update list of books available
Exception Conditions	1.1 If student does not exist, books cannot be issued or returned	

Table.2 Use Case Description (Students)

SEQUENCE DIAGRAMS

Fig.2 shows a sequence diagram made for the add and delete book functionalities. The interaction is between the librarian, the application, and the databases. Each interaction shows the messages exchanged between the objects.

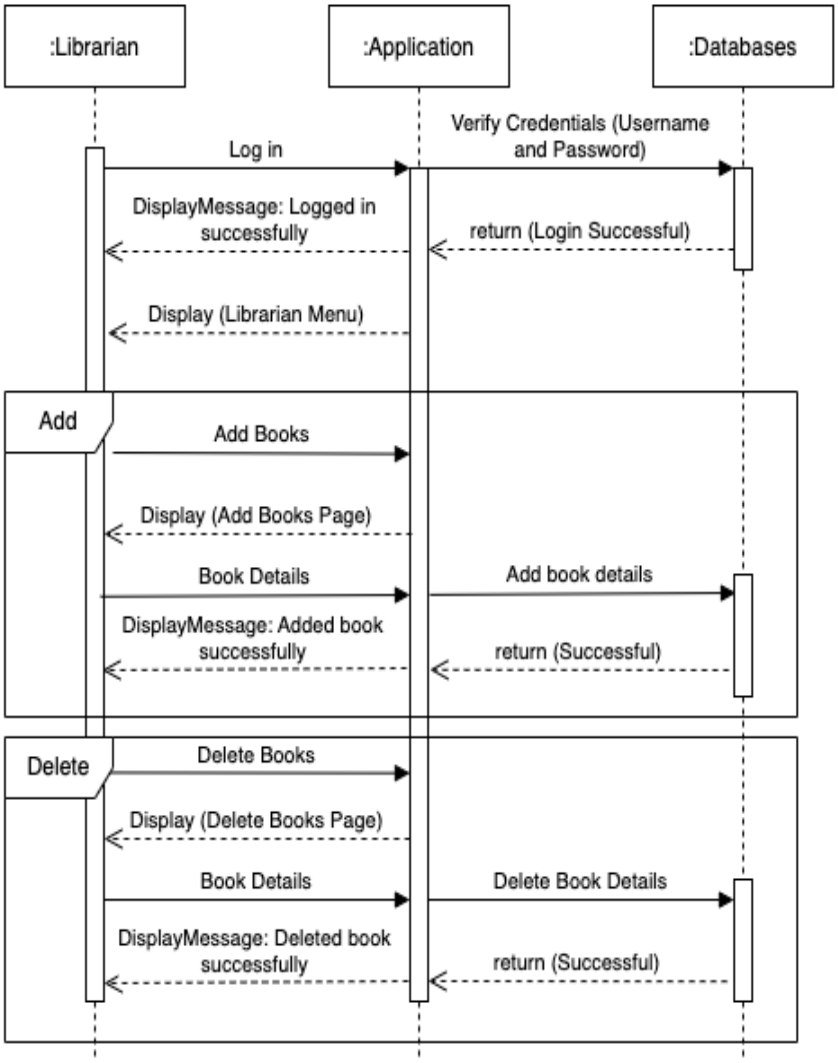


Fig.2 Sequence Diagram (Add and Delete)

Fig.3 shows a sequence diagram made for the issue book and return book functionalities. The interaction shown is between the librarian, the application, and the databases. Each interaction shows the messages exchanged between the objects.

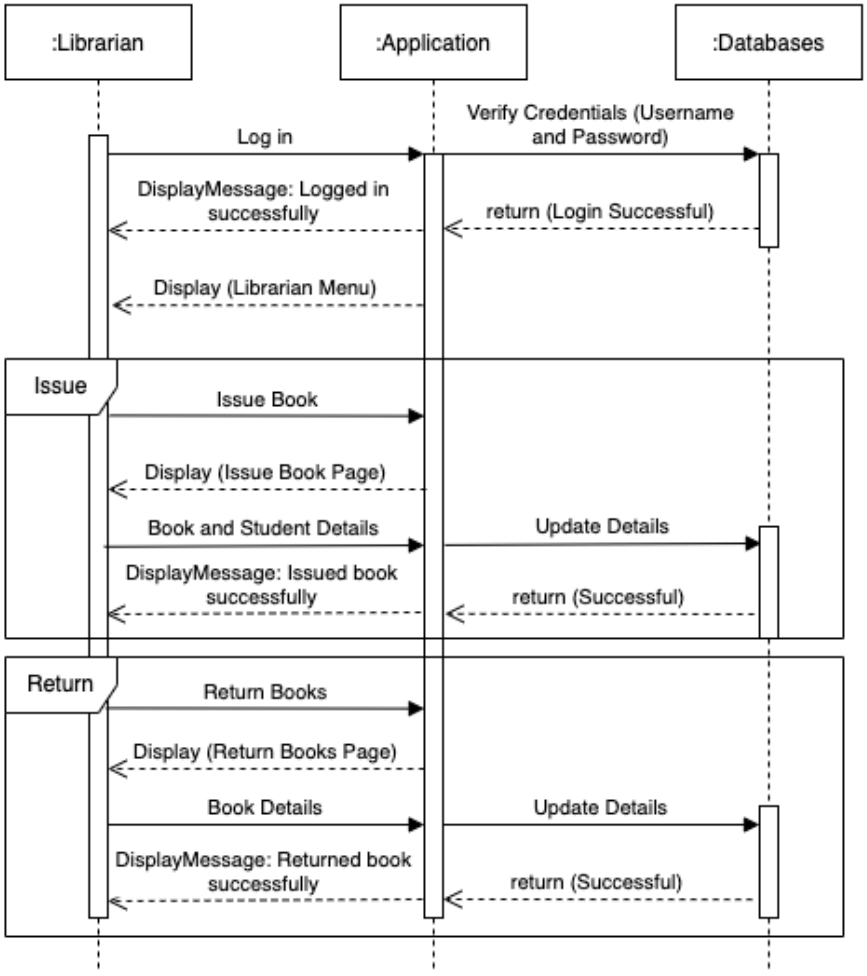


Fig.3 Sequence Diagram (Issue and Return)

Fig.4 shows a sequence diagram for the view book functionality. The interaction shown in the diagram is between the librarian, the application, and the databases. Each interaction shows the messages exchanged between the objects.

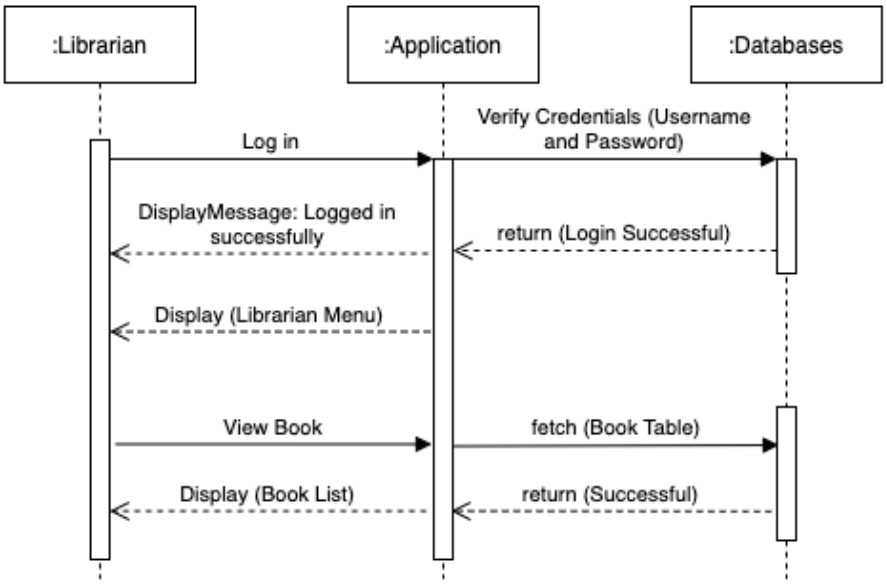


Fig.4 Sequence Diagram (View Books)

Fig.5 shows a sequence diagram for the student side of the application. Student can login, register, view available books and view books previously issued to them. The interaction shown in the diagram is between the student, the application, and the databases. Each interaction shows the messages exchanged between the objects.

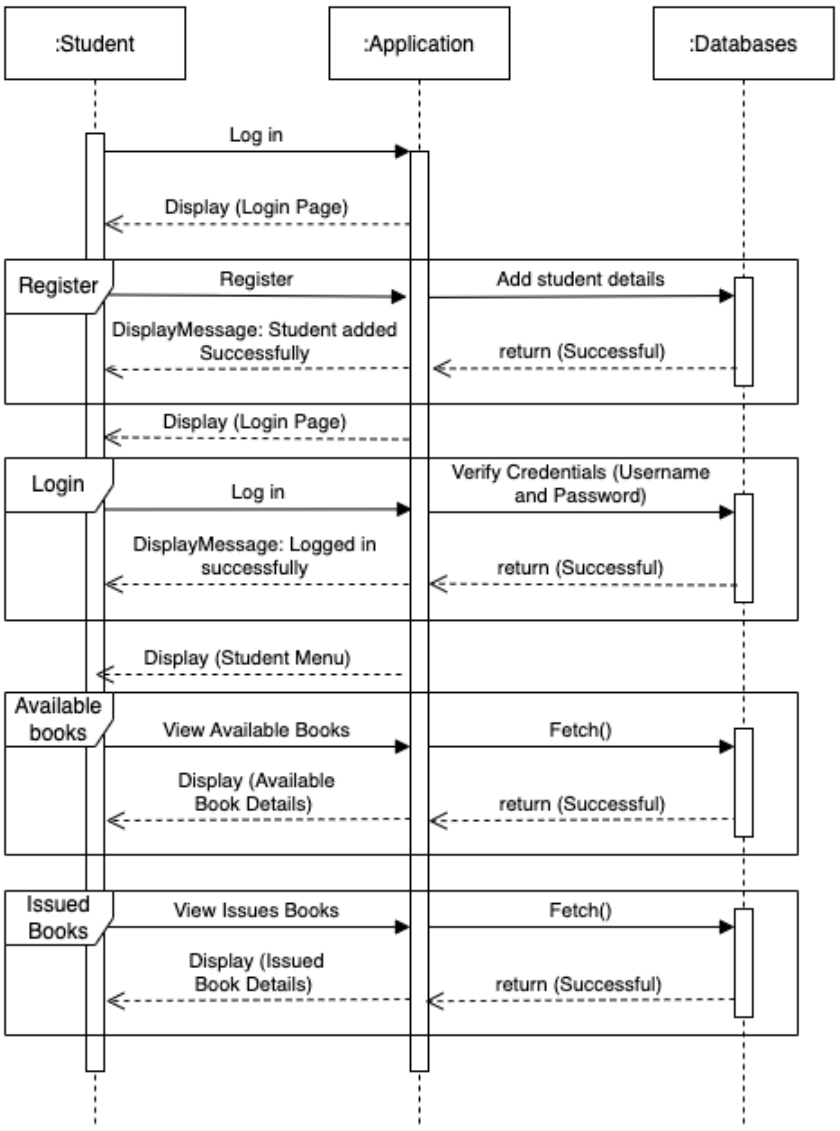


Fig.5 Sequence Diagram (Student View)

SYSTEM ARCHITECTURE

Fig.6 is a diagram of the application's system architecture. The application follows a microservices architecture. Microservices architecture is best suited for small-scale applications such as this. Each functional component of the application is a service of its own. The architecture has two components, the librarian component, and the student component. Each component has a GUI of its own. The services offered by each component is listed in the diagram. Both these components are also connected to three separate tables.

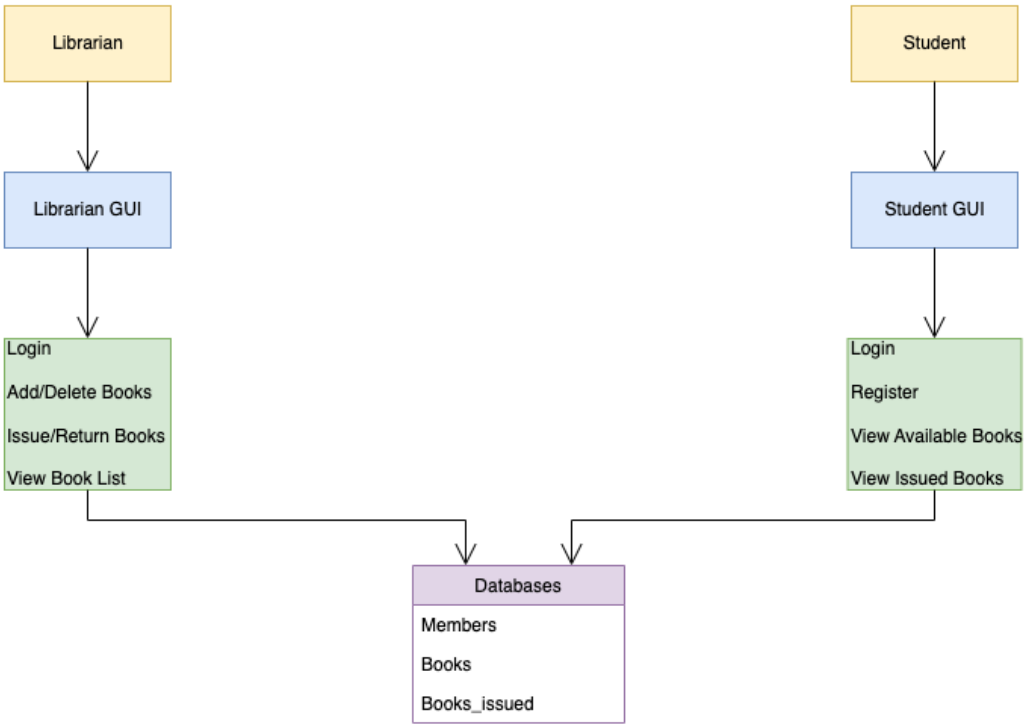


Fig.6 System Architecture

CLASS DIAGRAM

Fig.7 shows a detailed class diagram for the modules implemented in our application. The flow starts from the Main Menu module. Since our application is split into different modules depending on the functions offered, the class diagram represents the relationships between each of those modules.

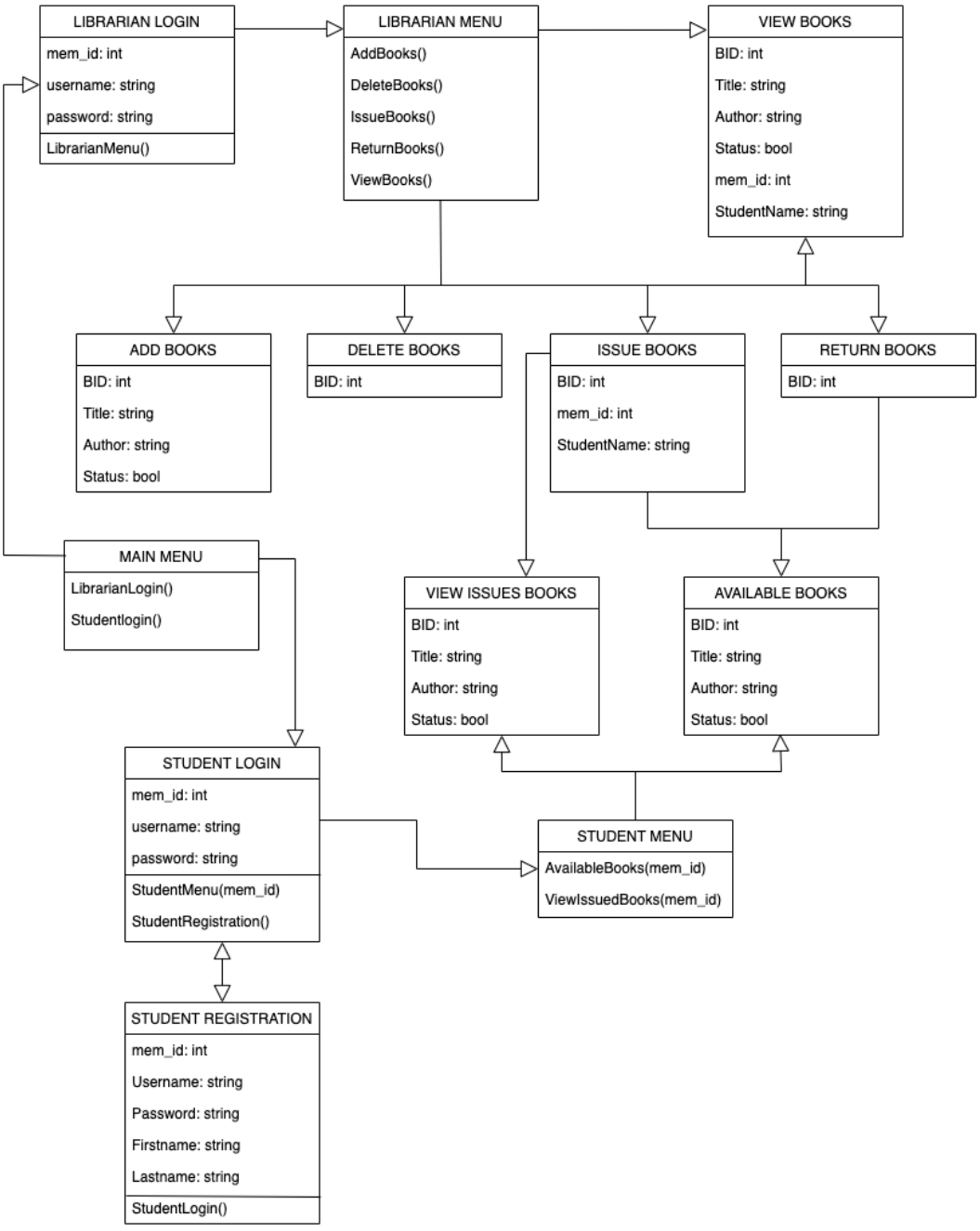


Fig.7 Class Diagram

STATE MACHINE DIAGRAMS

Fig.8 is the state machine diagram for our application. It shows the different states at which our librarian object can be throughout two different processes. The first process shows the states that the object can be at when the librarian adds a book to or deletes a book from the inventory. The second process shows the states at which the librarian object can be at when they issue a book to a student or return a book from the student.

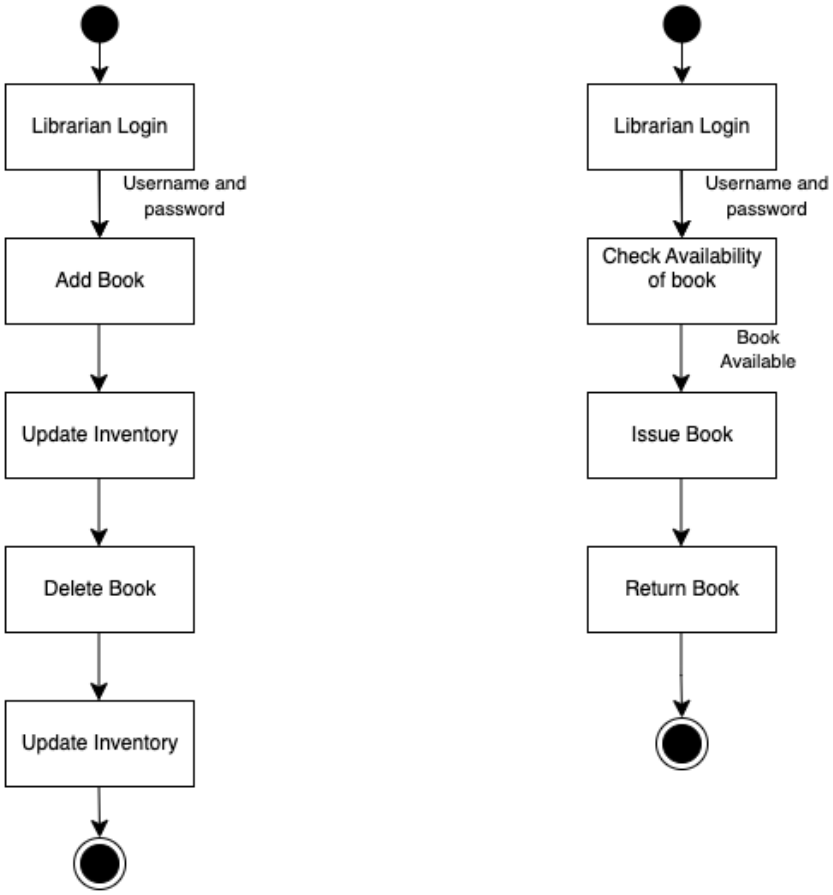


Fig.6 State Machine Diagrams

ENTITY RELATIONSHIP DIAGRAM

Fig.9 shows the ER diagram of the application. The three entities are Books, Book_issued, and Members. This diagram shows a high-level data model of the application. It shows how each database is linked with each and what each one is composed of.

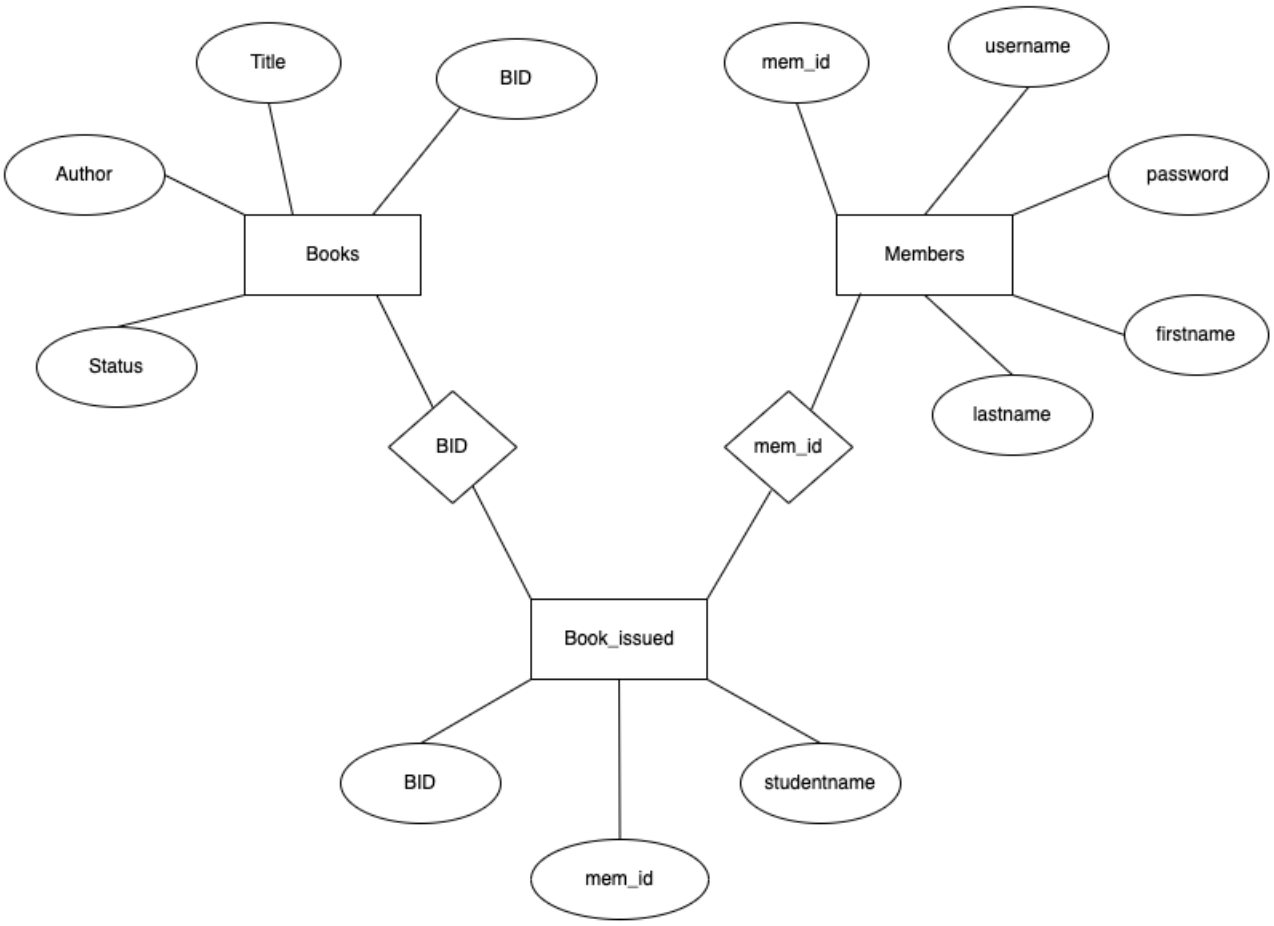


Fig.9 Entity Relationship Diagram

CONCLUSION

The Library Management System is a user-friendly and customized software for the librarian and the students. It has been developed to manage and automate the many tasks of a library management system. The application is now capable of allowing the librarian to perform tasks such as adding books to the inventory, deleting books from the inventory, issuing books to students, returning books from students, and viewing the list of books. It is also capable of allowing the students to view a list of available books to be issued and a list of books that have been previously issues to them. This application is very flexible to any additional functions that need to be added and can be upgraded according to the managements' requirements.

GITHUB LINK

https://github.com/Piyushadya/libraryMangmentSystem

APPENDIX

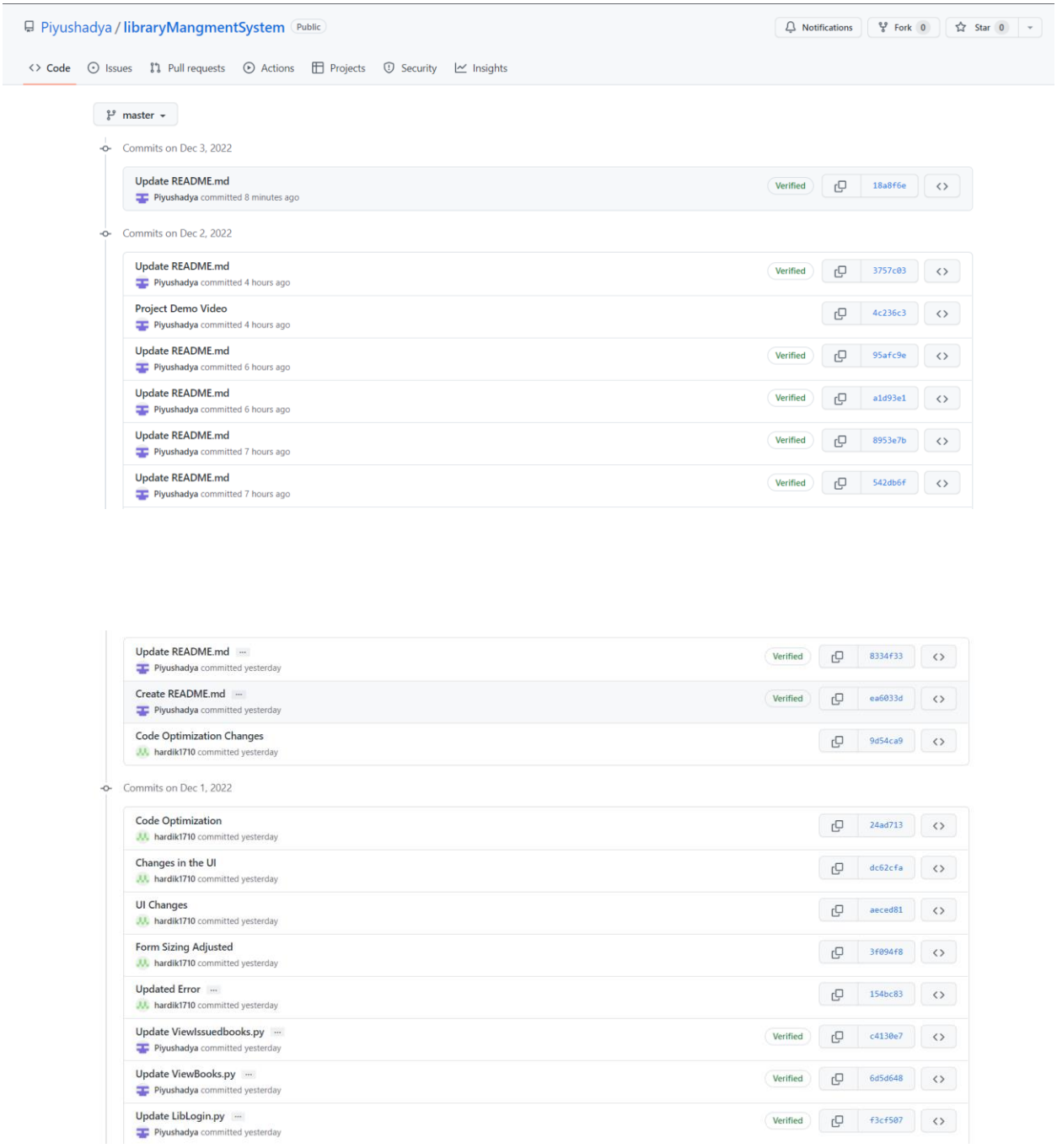
PROJECT WBS

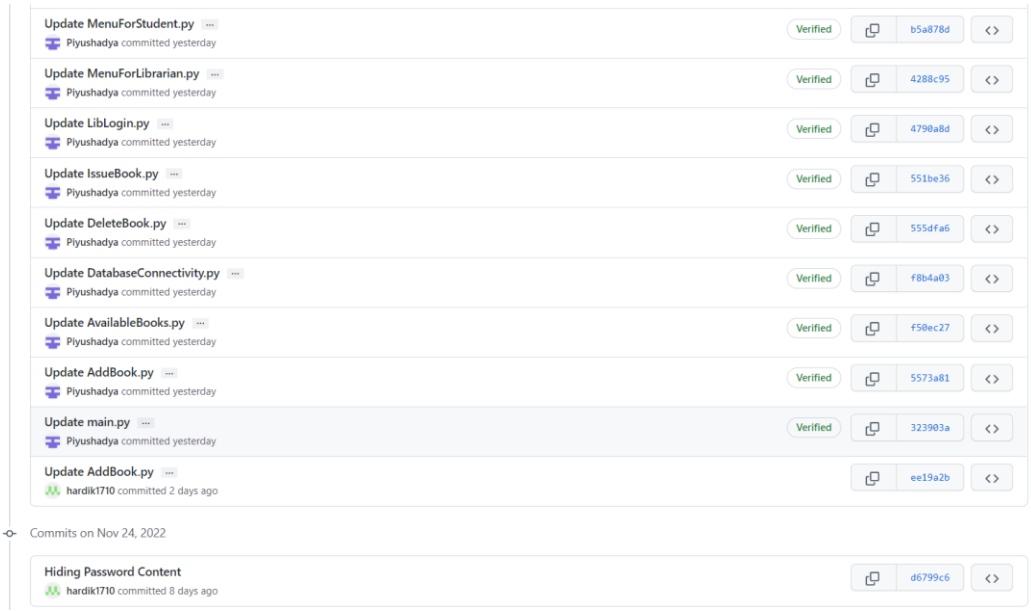
TASK ALLOCATION MATRIX

TASK	TASK OWNER	SUPPORT
Kick Off Meeting	Project Manager	Developer 1, Developer 2, Developer 3
Requirements Gathering	Project Manager, Developer 1, Developer 2, Developer 3	
Finalize Requirements	Project Manager	Developer 1, Developer 2, Developer 3
Design Selection	Project Manager, Developer 1, Developer 2, Developer 3	
Implementation	Developer 1, Developer 2, Developer 3, Developer 4	Project Manager
Testing	Developer 1, Developer 2, Developer 3, Developer 4	Project Manager
Deployment	Developer 1, Developer 2, Developer 3, Developer 4	Project Manager
Diagrams	Developer 1, Developer 2, Developer 3, Developer 4, Project Manager	
Documentation	Developer 1, Developer 2, Developer 3, Developer 4, Project Manager	
Project Closure	Project Manager, Developer 1, Developer 2, Developer 3, Developer 4	

Project Manager – Pavithra Gopalakrishnan
Developer 1 – Piyush Piyush
Developer 2 – Piyush Adya
Developer 3 – Hardik Sachdeva
Developer 4 – Pavithra

SAMPLE COMMITS ON GITHUB




Update MenuForStudent.py
Piyushadya committed yesterday
Verified
b5a878d
Update MenuForLibrarian.py
Piyushadya committed yesterday
Verified
4288c95
Update LibLogin.py
Piyushadya committed yesterday
Verified
4790a8d
Update IssueBook.py
Piyushadya committed yesterday
Verified
551be36
Update DeleteBook.py
Piyushadya committed yesterday
Verified
555dfa6
Update DatabaseConnectivity.py
Piyushadya committed yesterday
Verified
f8b4a03
Update AvailableBooks.py
Piyushadya committed yesterday
Verified
f50ec27
Update AddBook.py
Piyushadya committed yesterday
Verified
5573a81
Update main.py
Piyushadya committed yesterday
Verified
323903a
Update AddBook.py
hardik1710 committed 2 days ago
ee19a2b
Commits on Nov 24, 2022
Hiding Password Content
hardik1710 committed 8 days ago
d6799c6