

Library Management System

¹Shreya Baliga, ¹Nisha Patil, ¹Anuja Patil

¹Information Systems, Northeastern University

{baliga.s, patil.nis, patil.anuj}@northeastern.edu

Abstract— Library Management system allows for the management of books, users, and borrowing activities in a library. The project includes a graphical user interface (GUI) using JavaFX that enables easy interaction with the system. The GUI allows users to search for books, borrow and return books, and manage user accounts.

This system has two actors: Admin and User. Admins are responsible for managing the books, users, and borrowing activities, while the users can only search for and borrow books. The system includes various features such as user authentication, book reservation, and overdue book tracking. The project utilizes object-oriented programming concepts and implements various design patterns, such as Singleton and Factory pattern, to ensure efficient and scalable code.

The outcome of this project is the development of a library system that aims to promote reading habits and knowledge among individuals who face access or financial barriers to reading. The library system is designed to provide readers with a wide range of book options and allow them to borrow books that they might not be able to access or afford otherwise.

The library system is expected to enhance the reading experience for individuals and promote a love for books and literature. The project aims to make a positive impact on the community by providing access to books and fostering a reading culture. The system also incorporates features such as user authentication, book reservation, and overdue book tracking to ensure efficient management of the library's resources.

Overall, the library system is expected to serve as a valuable resource for readers who are passionate about books and knowledge. The project highlights the importance of promoting reading culture and provides a practical solution for individuals who face barriers to accessing books.

Keywords—*Javafx,mysql,Singleton,Factory pattern,design patterns, Object oriented principles.*

I. PROBLEM DESCRIPTION

The habit of reading books is one of the habits that is tried to be given to us from an early age. Both in the fairy tales our parents read to us when we were little children, and the 'reading-writing' actions we learned when we started school have been important elements that tried to make the book a part of our lives.

Despite the fact that access to books is widely available nowadays, many individuals still wish to read but are unable to do so because of their financial situation or the conditions of their homes. In addition, a lot of people acquire library culture and adore the ambiance despite not having access or financial issues.

The purpose of our project is to design a library where we provide the readers with options for books they might enjoy and, we can improve their reading habits and knowledge. It is a library system that can help folks who enjoy reading by letting them borrow the books that are out of their grasp because of access issues or budget constraints.

II. ANALYSIS (RELATED WORK)

According to WANG, X. Y.'s work [1], The LMS is designed with the basic features such as librarian can add/view/update/delete books and students' details in it. Once he/she ingresses into the system they can modify any data in the database using C# and SQL. On the other hand, Sourabh Sharma's work [2] the LMS is a practical solution to the traditional library system. The application mainly consists of three modules which are admin module, librarian module and student module. The admin module will be managed by the system administrator. He manages the overall functioning of the application. The librarian module will be accessed by the librarian. He can perform various operations inside the application such as adding new students, new books to the database, issuing and returning of books, updating student's details, book's details, generating weekly/monthly reports etc. The student module can be accessed by registered students only. The operations that can be performed by the student include view all books available in the library, search the availability of a particular book, number of books he has issued, overall fine he must pay etc. These three modules are interconnected with each other and with the database. The application is built using Java technology and SQL database.

Chaulagain, B's work[3] aims to develop a computerized system to maintain all the daily work of library. This project has many features which are generally not available in normal library management systems like facility of user login and a facility of teachers login .It also has a facility of adminlogin through which the admin can monitor the whole system .It also has facility of an online notice board where teachers can student can put up information about workshops or seminars being held in our colleges or nearby colleges and librarian after proper verification from the concerned institution organizing the seminar can add it to the notice board . It has also a facility where student after logging in their accounts can see list of

books issued and its issue date and return date and also the students can request the librarian to add new books by filling the book request form. The librarian after logging into his account i.e, admin account can generate various reports such as student report , issue report, teacher report and book report.

Tsega Weldu Araya's work[4] aims to design the Web-based Digital Library Management System (DLMS) that will help to create a limitless amount of digital information and make it accessible to the world in parallel. This system planned to obtain research resources for the ACCE and then distribute it to users in their remote areas. The design parameters are Java Programming language, PHP, HTML, and Database MYSQL.

As Asmait Futsumbrhan explained that, a library is a place where a collection of books and other informational materials are made accessible to people for reading, study, or reference in their daily life activity. The library collections have almost contained a diversity of materials making it much easier for everyone who has an interest in reading and finding new things regarding their interests. Contemporary libraries preserve collections that contain printed materials such as manuscripts, books, newspapers, magazines, maps, and photographs. However, we found her explanation limited to the usage of the library. We are converting all the paperwork activities to a computer system and although adding a new eBook system, so the users can get access inside the library room and outside in digital format.

Concurrently to Harsh S.B's work [5], the author explained the Library System as a fast-growing database for information retrieval, which targets in developing computerized systems to preserve the daily work of the library. Well, computerized library will be a benefit for users with fast and prompt services. Their library system denotes to the systematization of library maintenance and activities mainly used computerized. Their work has many structures that are mostly not available in manual LMS like facility of user login. In addition to the similarity of their work, we also proposed the digital archive, which can store the electronic files. This newly developed system has a capability of admin login through which the admin can monitor the whole system. Objectives of this study are to develop and build the database for the records and other facilities to the computer application. To provide numerous search selections to check the convenience of books in the library to generate the list of books.

Gustman's work [6] uses the same idea as our paper in converting an analog format of records to digital format. Their research also covers converting multimedia documents but our paper limits for creating a digital platform for the books and changing library daily activities to a computerized system.

Park N's work [7] briefly describe how to create a digital system, which has a higher acceptance rate and adoption with the users. Their focus was on decreasing the mismatches between system design and local user's realities. To satisfy our users and the library staff's needs we are implementing a user-friendly web-based system.

III. SYSTEM DESIGN

The use case diagram shows how people interacting with the system interact, in the most general way. In our 'Library Management System' project there are two actors interacting with the system Admin and User. The admin is the person responsible for the basic works of the library, who interacts more with library than the user. User, on the other hand, is the person responsible who can borrow books from the library, list the books that has bought in the past, search the books in the library and have the rights to list the books by logging into the library with name and password determined by the user as shown in the Figure 1.

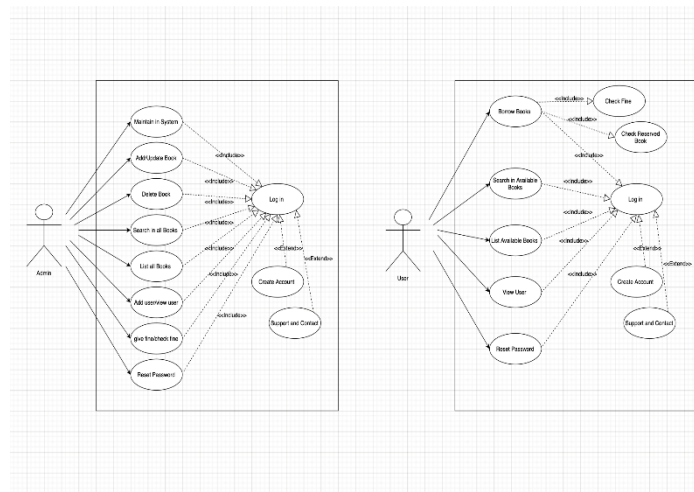


Figure 1. Use Case Diagram

The below diagram Figure 2. shows the class diagram for this project.

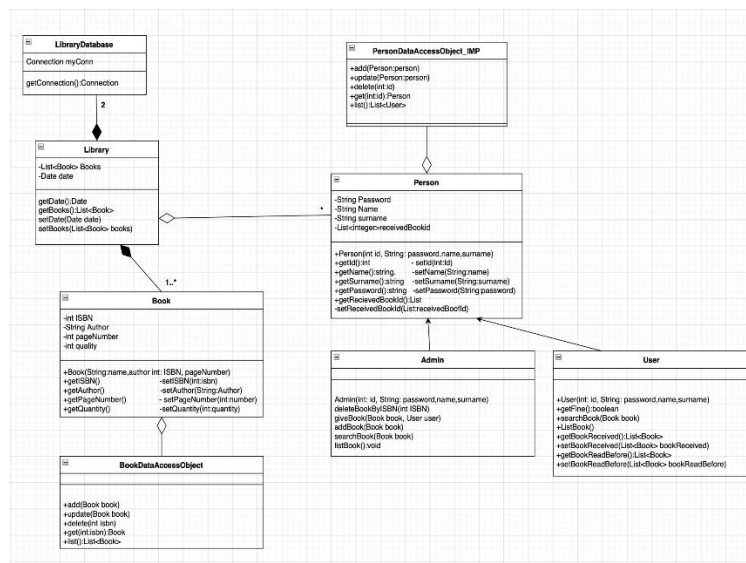


Figure 2. Class Diagram

Person: Each person has his/her name, surname, password, and books he/she bought. The person has a constructor containing these properties, and there are public getter and private setter commands for this information. Each admin and user is a person.

Person Data Object Access implementation: In this class, the links between database and person are provided by functions. It is available in all classes, but we used one for class diagram display, the logic of all of them is the same.

Admin: Admin also has its own constructor and has functions delete iBook(delete book), giveBook(give the book to the user), addBook (add the book to the library), searchBook(search among all books), listBook(list books).

User: In addition, the user has its own constructor and functions getFine(search for penalty), searchBook (search among current books), getBookReceived (display received books), getBookReadBefore(view previously read books).

Library: There is a private list that holds the books in the library and there is a private variable named Date that shows the date. As a function, these variables have getters and setters.

Book: Books are available in the library. So the library has 1 or more books. Books have ISBN numbers, author names, page numbers, quantities and names. As a function, it has its getters and setters.

LibraryDataBase: The library has 1 database. However, this database has 2 tables, so we needed to indicate in the diagram that there are 2 tables, DataBase has a local connection(myConnection). It also has getConnection:Connection as a function.

The below diagram Figure 3 shows the sequence diagram for user.

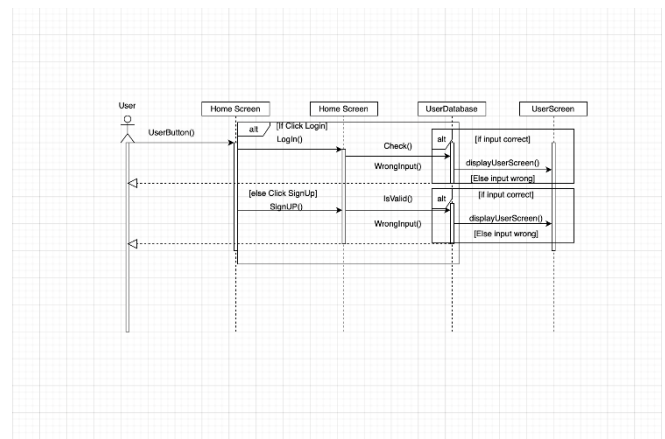


Figure 3. Sequence Diagram for user

The below diagram Figure 4 shows the sequence diagram for admin.

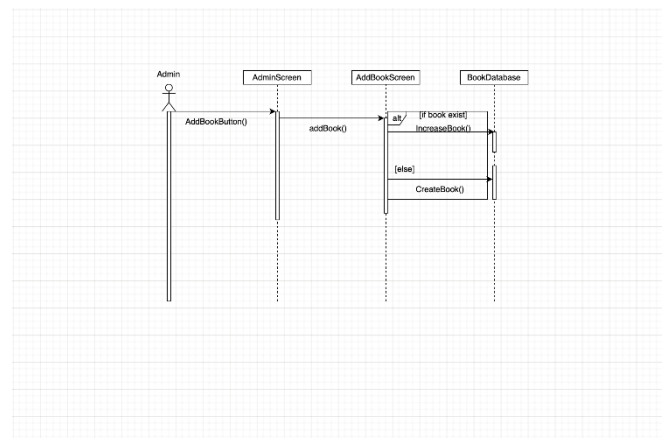


Figure 4. Sequence Diagram for admin

IV. IMPLEMENTATION

A. General Layout of FX

The tree structure is used for transition between the pages.

A class named SceneController has been used for these transitions. Each page is linked to the java class which performs the necessary functions.

Back Key: To go to the previous page there is a back key at the top left of the page. When pressed, it goes to the previous page.

Message Handling: If any operation which is performed is wrong, a warning message is displayed. This is valid for all

transitions. Also, if all the transitions are correct, successful message is displayed on the screen.

B. Main Screen

The system has two actors named user and admin.

The users and admins can click on their respective buttons and continues the program.

The person can switch between the dark/light mode from the top right by clicking on the moon button and if the person wants to exit then he can exit by clicking on the close/exit button.

C. Login Screen

The person can login as a user or as an admin. There are two separate login screens for user and an admin. Authentication and Authorization is taken care of.

The checklogin function is used to authenticate the login by verifying the id and password entered by the person with the one present in the database and sends the necessary messages to the person.

D. Register Screen

The person can register as a user or as an admin. There are two separate register screens for user and an admin. Authentication and Authorization is taken care of.

The checkRegisterForAdmin function is used to check whether an admin exists already with that id and password.

The checkRegisterForUser function is used to check whether an user exists already with that id and password.

The changePassword function compares the current password in the database with the password entered by the user and if correct updates the user's password if entered correctly.

E. Home Screen

The user and admin have different home screen.

On the left they have navigation panels which consists of the operations they perform.

F. Display Tables in FX

The same logic is used in all table representations. We give the necessary names to the table and the columns in the table in our fxml file.

For example in the search book function. All the necessary FX naming has been done as seen in figure x. Then a java class is opened for the function searchBook. In this class ids are introduced and linked with FXML file. ArrayLists to be displayed later are transferred to ObservableList. To set them correctly, setCellValueFactory is used. It keeps List Book objects and when the value "name" is written, whichever object is used in the list, the value factory goes to the required class and extracts all data matching "name" and transfers it to the column. Since all this code is written inside the initialize function they are constantly updated. The elements kept in the Observable list are transferred to a new list structure called FilteredList, when each key is entered ,it is checked whether there is anything matching with if else's and then the matching ones are transferred to the sorted list structure.

Finally, the data matching the table structure is sent back.

G. Admin Screen Options

G.a Add Book

Book Data Access Object Implementation:

We need data access object class (interface) to facilitate access between database and the functions. The methods available in this class are:

When the admin enters, the admin can see the existing books in the library and how many there are. The admin can fill fields and add books.

While adding a book in the addBook function, the admin first checks whether the book exists in the database. If the ISBN number of the book matches one of the book ISBNs in the database, the number of that book is increased by 1, if there is no book then that book is added to the database.

The database accepts the object "Book" as a parameter and then adds this object to the database with the insert query.

G.b Delete a book

The admin sees the existing books in the library and when he or she clicks on the delete button, the book that matches the isbn entered by her/him is deleted.

The deleteBookByISBN function is responsible for performing the delete operation. In the deleteBookByISBN function, the ISBN of the book to be deleted is found by using the search function from the database. If the book is not found in the

database it will give an error message. If it is found, then it checks if there are multiple copies of that book. If only one copy is found then it reduces the count of that book by 1 but not completely delete it from the list, because when the book is returned, instead of creating a new book, the same book is replaced and its number is increased. If there are multiple copies then it decreases the count of that book by 1.

In the database, it deletes the book that matches the ISBN entered by user.

G.c Give Book

Admin sees the list of books and users in the table. The admin enters the ISBN of the book he or she wants to give and the id of the user to be given to and clicks on the give button.

In the giveBook function, the id of the book to be given and the id of the user to whom the book will be given are entered. If the user has a fine, the book is not given to the user and an error message is displayed. If not, then the book is given to the user and the count of the available books is reduced by 1.

G.d Return book

When the user returns the book and gives it to the admin, the admin is responsible for entering the ISBN of the book and the user's ID and click on the return button.

In the bookReturn function, the id of the user who will return the book and the ISBN of the book to be return are entered. If the user has fine then it will be removed on payment. Also, the book is included in the list of books he has read.

G.d Fine

The admin can see the information of the users with fine.

ListFine displays the users with fine. The users with fine are added to the usersFine list and the list is returned.

G.e Search

In the searchBook function, the word or string to be searched is entered. If there is an author or book name containing the entered input, it is listed.

In the database, search(String string: boolean isAdmin): It accepts a string and a boolean as arguments. Returns the books containing the string given as author or name in the database as ArrayList. "isAdmin" parameter helps in determining whether the query is made by user or admin.

G.f Update Book

Updates the data that matches the ISBN of the book given as argument in the database with the information of the project.

G.g Get Book

It accepts the ISBN as parameter, saves the information of the book that matches the ISBN entered in the database to an object and returns this object.

G.h List Books

All the books in the library are listed.

G.i Admin Profile

The admin can see all this profile information here. The admin can change his password in this section. The admin can logout by clicking on the logout button.

H. User Screen Operations

The syncData function updates the database when there is a change in the user or admin information.

Since both user and admin have common fields, we made use of abstraction using "person" abstract class.

H.a Books Read

The users can see a list of all the books he or she has read before and can search.

H.b Search Book

When the user logs in, the user sees a list of books in the library. Using the search function he can find a book if it exists or else it will throw an error.

H.c User Profile

The user can see all the information pertaining to him or her such as the books he or she currently holds, fine if any, the book to be returned. The user can also change the password.

I. Person Data Access Object Implementation:

I.a Add(Person person)

It accepts the "Person" object as a parameter and adds this object to the database using insert query. We have not created different tables for admin and user in our database ; we have separated it using a Boolean variable. The add function distinguishes the incoming object as admin or user using the instance query. Then it stores all the information in the table.

I.b *Update(Person person)*

It updates the information of a previously registered user or admin.

I.c *Delete(int id)*

It performs deletion over id.

I.d *Person get(int id)*

It transforms a person from the database into an object using only an id like the Delete method and returns it. isAdmin creates an object according to it using the boolean variable. This method creates an object with only name, surname, and password at first, then syncs the created object and updates its other information. This sync function is called every time the application is opened.

I.e *listAdmins()*

It returns a list containing only admins in the database using the boolean variable isAdmin and the WHERE condition. It creates an Admin object in each loop, then appends it to the list.

I.f *listUsers()*

The same applies here as for the admin. Differently, by calling the sync method on the user, other variables are also synced. Other variables: "bookReceived", "bookReadBefore" and "deadlines".

I.g *listUserBooks(int id, boolean getReceived):*

Returns the previously received books or the list of books currently in the user, depending on the second argument, getReceive, of the user whose id is given. Both of these variables are saved as string in the database and ArrayList in local. It is a method for users only.

I.h *getDeadlines (int id):*

Returns the user's deadlines as a list, which is saved as a string in the database and an ArrayList in the local. It is a method for users only.

J. *Library Operations*

J.a *LibraryDB*

To control object creation. limiting the number of objects to only one we used Singleton Design pattern in this class. This

class provides an object to This class returns an object to access the database.

J.b *Library*

All fields and methods of this class are static. Since we only have one library and we can access it from every class, we followed this path. It has ArrayList holding all books, all users, and all admins, and it keeps today's date as static.

J.c *syncData():*

It pulls the book, user, and admin array lists from the database at the start of the program and saves them in the locale. For this, it uses the list methods of data access objects. Thus, the data is synchronized at every startup.

J.d *getBookByISBN(int ISBN):*

This method checks all the books in the book arrayList. Returns the matching book object if it matches the ISBN.

J.e *getUserByID(int ID):*

This method checks all the users in the user arrayList. Returns the matching user object if it matches the ID.

J.f *checkFines (User user):*

This method checks all the users in the user arrayList using for loop. Inside in this loop also checks time between today and users deadlines in another loop. Converts the hasFine boolean status of expired users to True.

V. EVALUATION

The below pictures Figure 5, Figure 6, Figure 7, Figure 8, Figure 9, Figure 10, Figure 11, Figure 12, Figure 13, Figure 14, Figure 15, Figure 16, Figure 17, Figure 18, Figure 19 are the screenshots of our project.

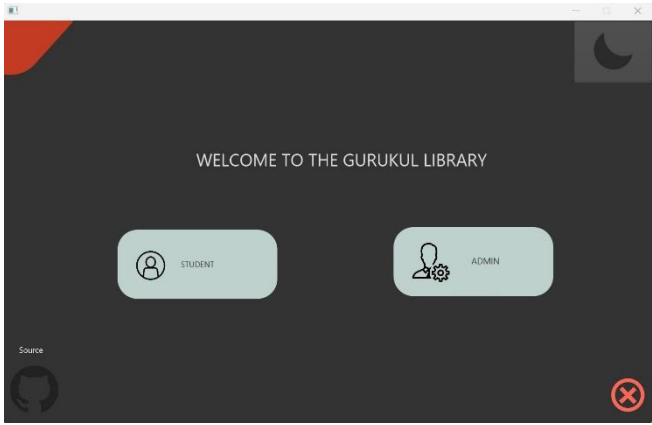


Figure 5. Welcome Page in the dark mode

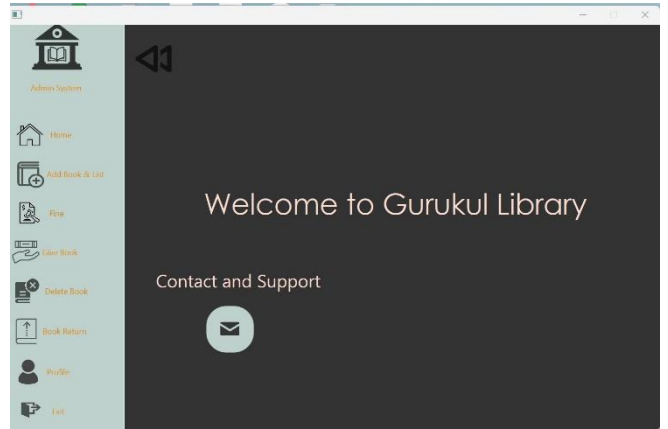


Figure 8. Admin Home Page

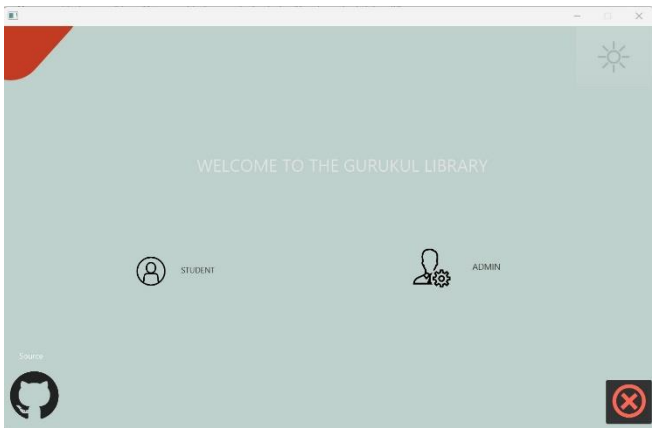


Figure 6. Welcome Page in the light mode.

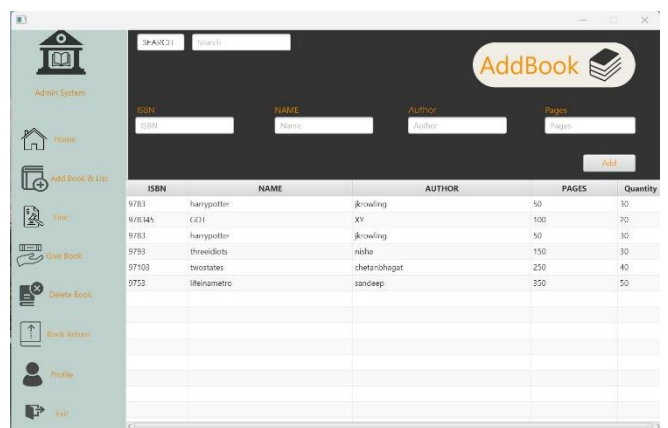


Figure 9. Add Book Page

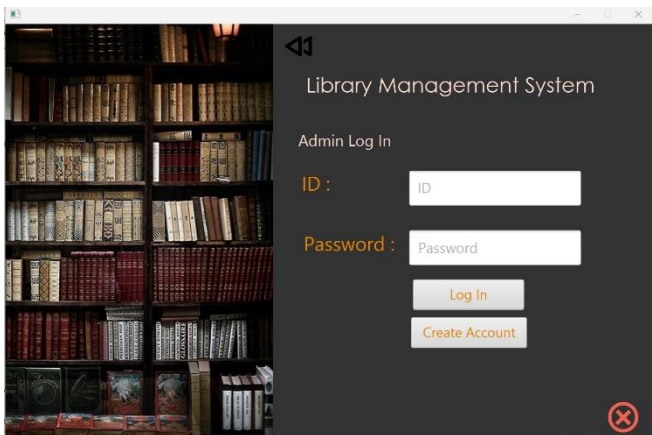


Figure 7. Admin Login Page

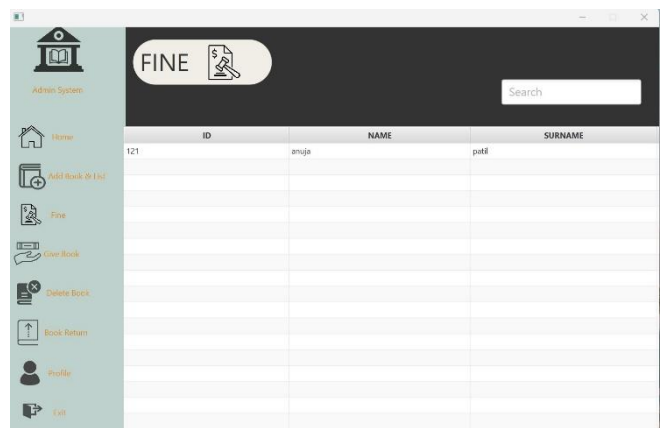


Figure 10. Records all the Fines.

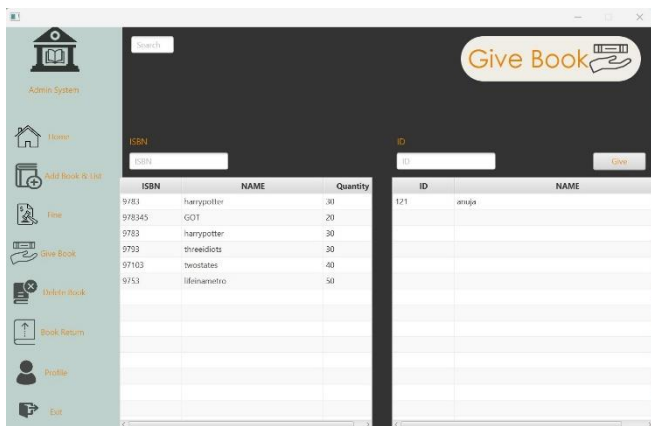


Figure 11. Keeps track of the books lent to the user.

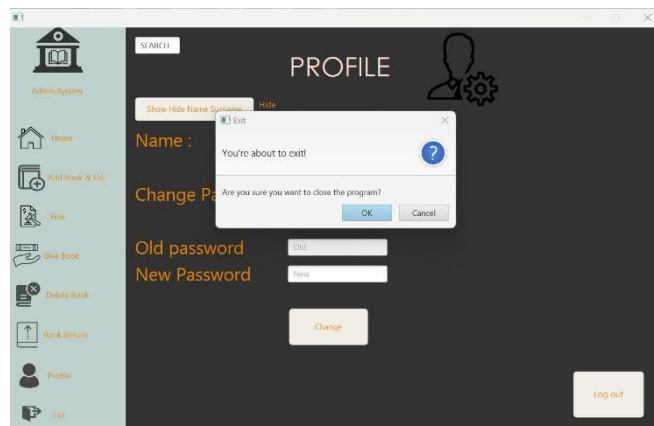


Figure 14. Exit Functionality

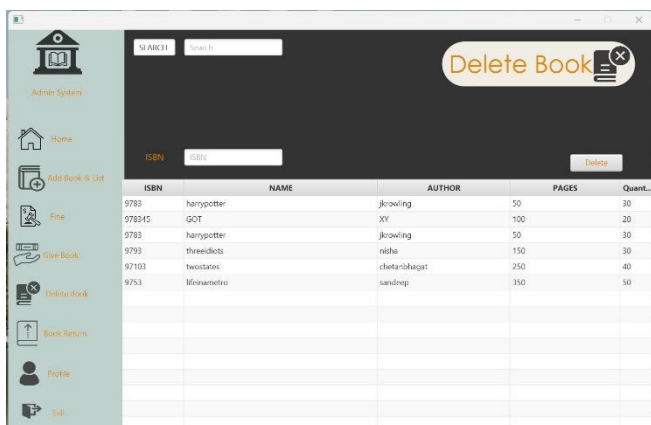


Figure 12. The admin can delete any book

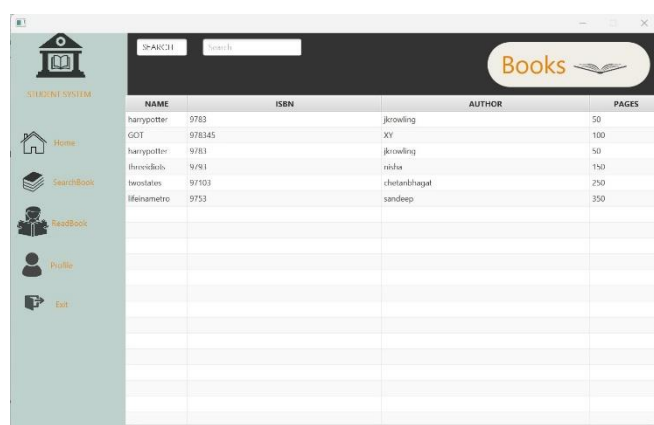


Figure 15. Search Functionality

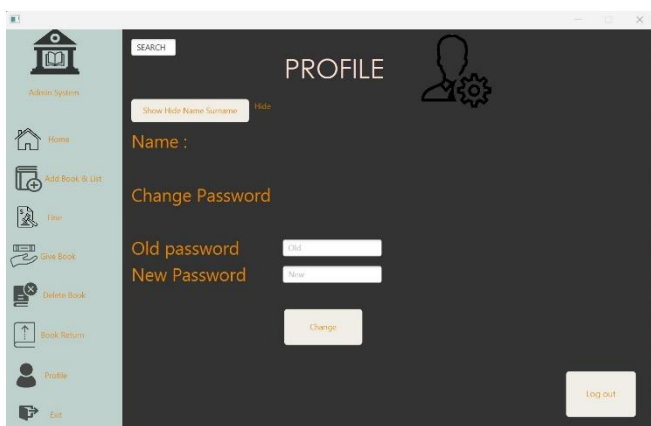


Figure 13. Admin Profile Page



Figure 16. Keeps track of books read previously.

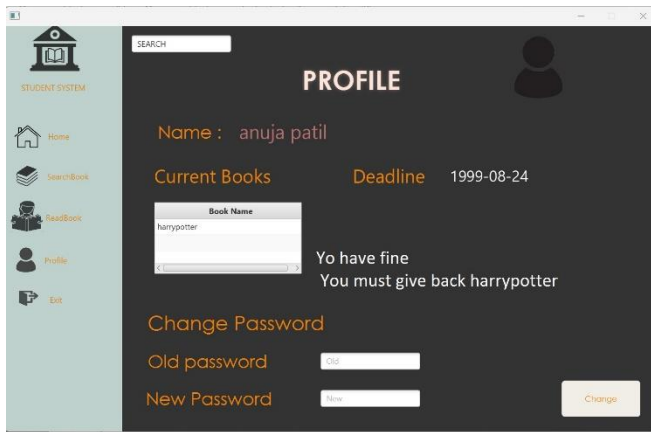


Figure 17. User Profile

VII. DISCUSSION (REFLECTION)

The project's main objective is to provide a user-friendly platform for admins to manage their library collections and the users to access the resources. The project's key features include a user authentication system, a book management system, a borrowing system, and a search engine for library resources. The user authentication system allows the library staff to login to the system and access its features, while the users can login to the system and can search for the resources and then request for the book borrowing through the portal. The book management system allows the admin to add new books to the system, edit book information such as author, publication date, and ISBN, and delete books from the library collection. The system also provides an interface for managing the book borrowing process, including tracking the status of borrowed books and generating due dates for returned books. The search engine allows users to search for books by author, title, or ISBN, and provides an option to search for books by category, such as fiction, non-fiction, or reference books.

VIII. CONCLUSIONS AND FUTURE WORK

This project includes all the features that a library needs. The primary goal of this project was to provide the people with a user friendly application using which the admins can easily manage the books and the users can easily find the books which interests them.

To assist employees in documenting the time and effort they put into their different work areas. To make the most of resources by increasing their productivity through automation. The system creates a variety of data that can be utilized for a variety of reasons.

The project is fast and stable. We kept all user and book data in the database. User information is also securely stored. In addition, because we do not aim for any income, and we think that having this feature will further make this project better.

There is a future scope of this facility that many more features such as reading books over the application, audio book feature,

interface development, interface optimization according to the age of the user, pricing system, users can add books to the system etc. We are very hopeful for future features as this open source project can be developed according to user needs.

IX. JOB ASSIGNMENT

The team members' contributions to the project.

- Shreya Baliga: Use Case Diagram, Front-end and back-end, database connectivity, report.
- Anuja Patil: Class Diagram, Front-end and back-end, database connectivity, presentation.
- Nisha Patil: Sequence Diagram, Front-end and back-end, database connectivity, presentation.

REFERENCES

- [1] WANG, X. Y., & Zhang, Q. S. (2013, September 18). Design and Implementation of Library Management System Based on .NET. *Advanced Materials Research*, 760–762, 934–937. <https://doi.org/10.4028/www.scientific.net/amr.760-762.934>
- [2] Sharma, S., Mishra, S., Gupta, S., & Kumar, S. (2022, May 31). Library Management System. *International Journal for Research in Applied Science and Engineering Technology*, 10(5), 889–893. <https://doi.org/10.22214/ijraset.2022.42375>
- [3] Chaulagain, B., & Awan, B. (n.d.). Library Management System Mini Project Report On LIBRARY MANAGEMENT SYSTEM. (PDF) Library Management System Mini Project Report on LIBRARY MANAGEMENT SYSTEM | Bibek Chaulagain and Bilal Awan - Academia.edu. https://www.academia.edu/37726542/Library_Management_System_Mini_Project_Report_On_LIBRARY_MANAGEMENT_SYSTEM
- [4] Tsega Weldu Araya. (2020, October 18). Designing Web-based Library Management System. *International Journal of Engineering Research And*, V9(10). <https://doi.org/10.17577/ijertv9is100131>
- [5] Harsh, S.B.: 'Management information systems', in Editor (Ed.)^(Eds.): 'Book Management information systems' (Michigan State University, 2011, edn.), pp
- [6] Gustman, S.: 'Digital library system', in Editor (Ed.)^(Eds.): 'Book Digital library system' (Google Patents, 2000, edn.), pp.

[7] Kumar, H.V., Jayaram, M., and Vikas, S.: 'WEB-BASED DIGITAL LIBRARY MANAGEMENT SYSTEM', initiatives, 2017, 4, (2)