Barkin Library Management Tool (BarkinLibTool) Technical Documentation

Summary

This project presents a Python-based desktop application for managing library books using the PyQt framework and MySQL database. It includes essential features such as user registration, login, book search and filtering, and loan request handling. Staff members can manage loan requests through a separate interface. The application is developed using object-oriented programming and designed with UML and ER diagrams offering a practical solution for basic library operations.

Recep Barkın Topcu

May 2, 2025

CONTENT

PERSONAL GAINS

At the initial stage of the project, I focused on defining the system requirements and creating UML diagrams to visualize the overall architecture. The project involved developing a Library Book Management System using the Qt Framework, where I learned to implement GUI-based desktop applications with Python and PyQt.

During the development process, I used Qt Designer to design and customize the user interface, which helped me gain a solid understanding of different approaches to building frontends in Qt.

For the backend, I set up the database using MySQL and handled database operations with mysql-connector-python

The application is developed using an object-oriented programming (OOP) approach. By effectively coordinating the frontend and backend components, I successfully developed a fully functional application.

This project provided me with valuable experience in desktop app development, database integration, and GUI design, while also deepening my understanding of Python and SQL.

KEY TAGS: SQL, Python, QT, MySQL and mysql-connector-python

1. Introduction

Libraries use various software solutions to manage and control physical books more efficiently. These solutions help track books, organize lending and return processes, and facilitate effective inventory management. In this project, the goal is to support library book management processes by developing a simple, Python-based desktop application.

2. Technical Design

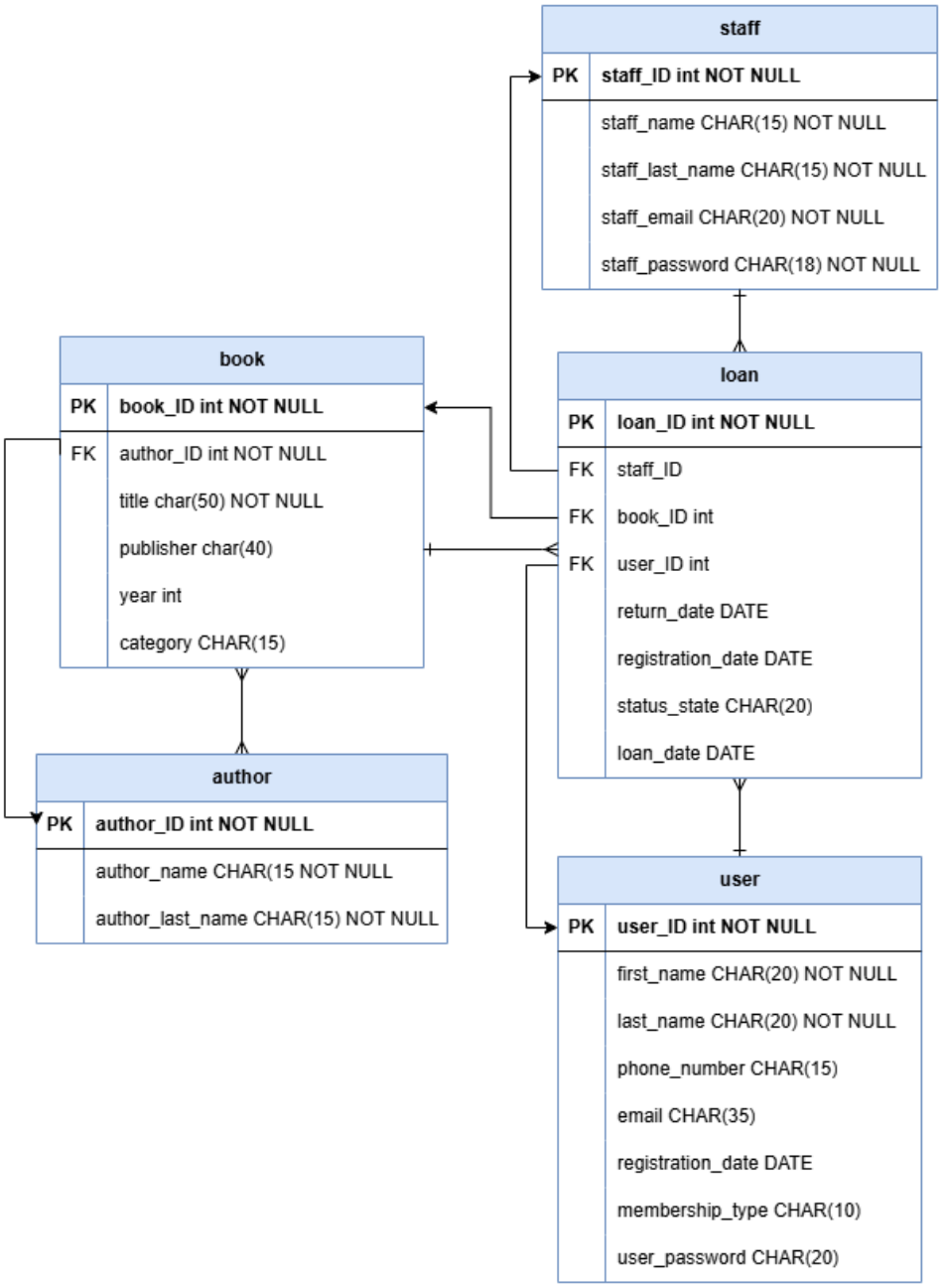


Figure 1. ER relationship of the tables.

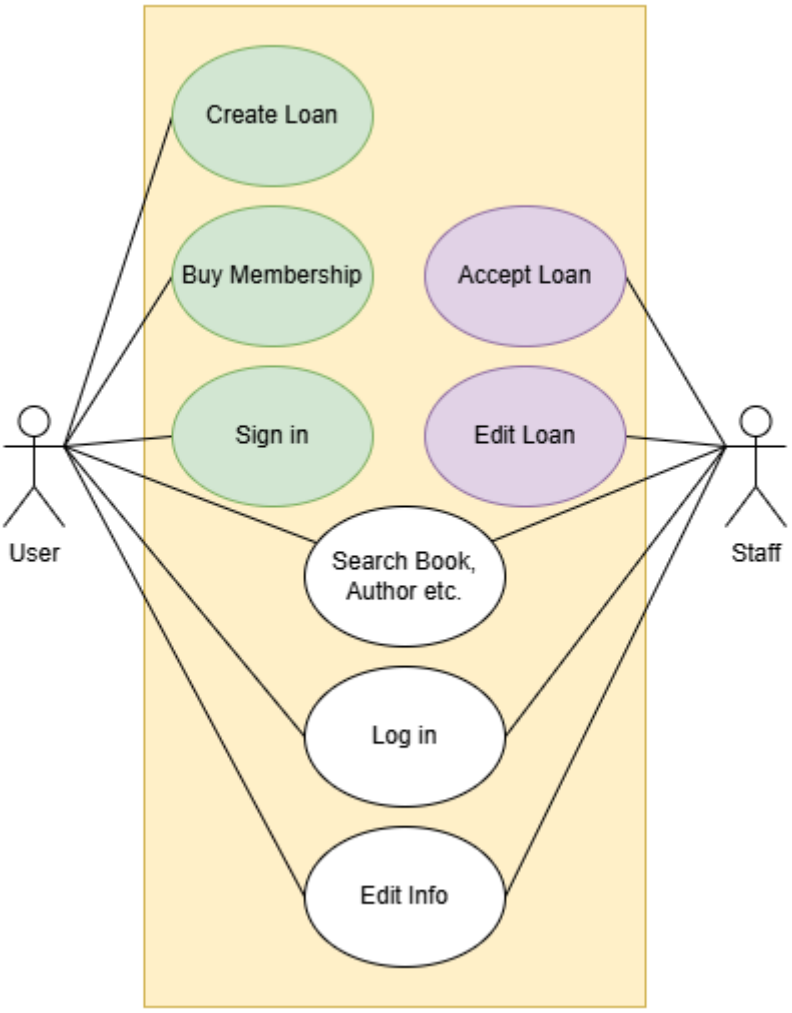


Figure 2. UML diagram of tool.

The relationship between the main entities in the database, such as books, members, and borrowing records is shown in Figure 1. This ER diagram defines how the tables are structured and linked, enabling efficient data organization for the library management system.

The overall system interactions and user roles within the application are shown in Figure 2. This UML use case diagram illustrates the functionalities available to two main actors: User and Staff. Users can perform actions such as signing in, buying a membership, creating a loan request, searching for books or authors, logging in, and editing their personal information. Staff members, on the other hand, are responsible for actions such as accepting loan requests and editing loan details, alongside shared functions like search, login, and editing information. This diagram provides a clear overview of how different roles interact with the system's core features.

3. Application Screenshots

3.1. Main Menu

When the application is launched, a login screen appears as shown in Figure 3. Users are required to enter their email and password to access the system. If incorrect credentials are entered, an error message is displayed, as illustrated in Figure 4. The email and password inputs are validated by checking the corresponding records in the database. Moreover, if the staff wants to log in, he/she must select the Staff box.

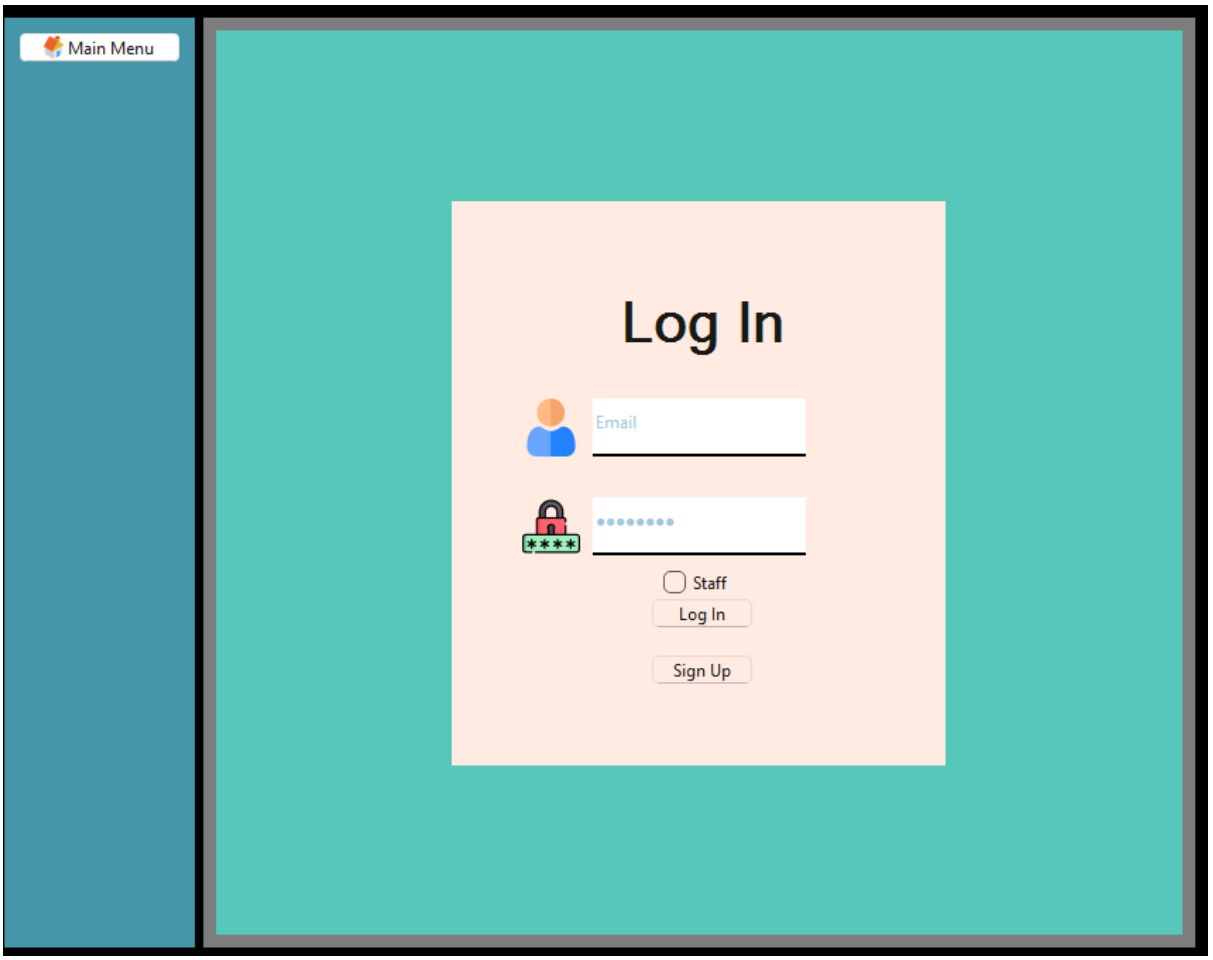


Figure 3. The login screen of the application.

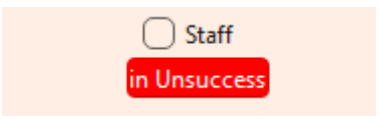


Figure 4. Unsuccess error message of log in.

Users can sign up by clicking the Sign Up button. After that, the registration page will be displayed, where users can enter their information to create a new account as shown in Figure 5. After the success login users can see their info in the Main Menu.

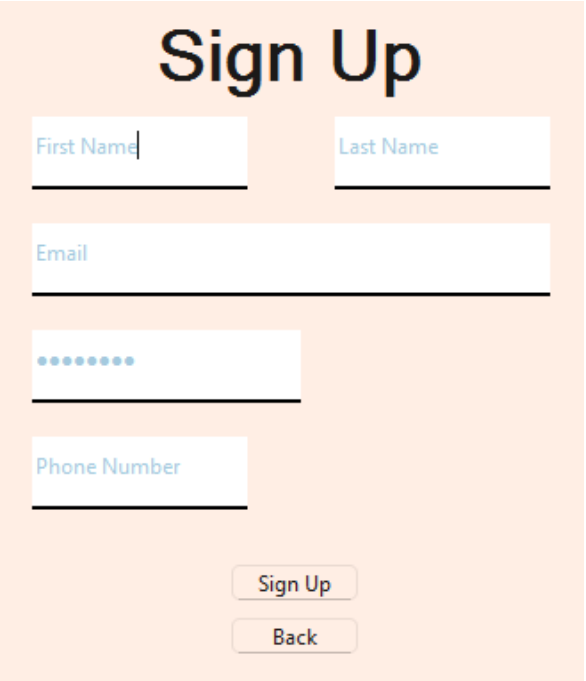


Figure 5. Sign up page of the application.

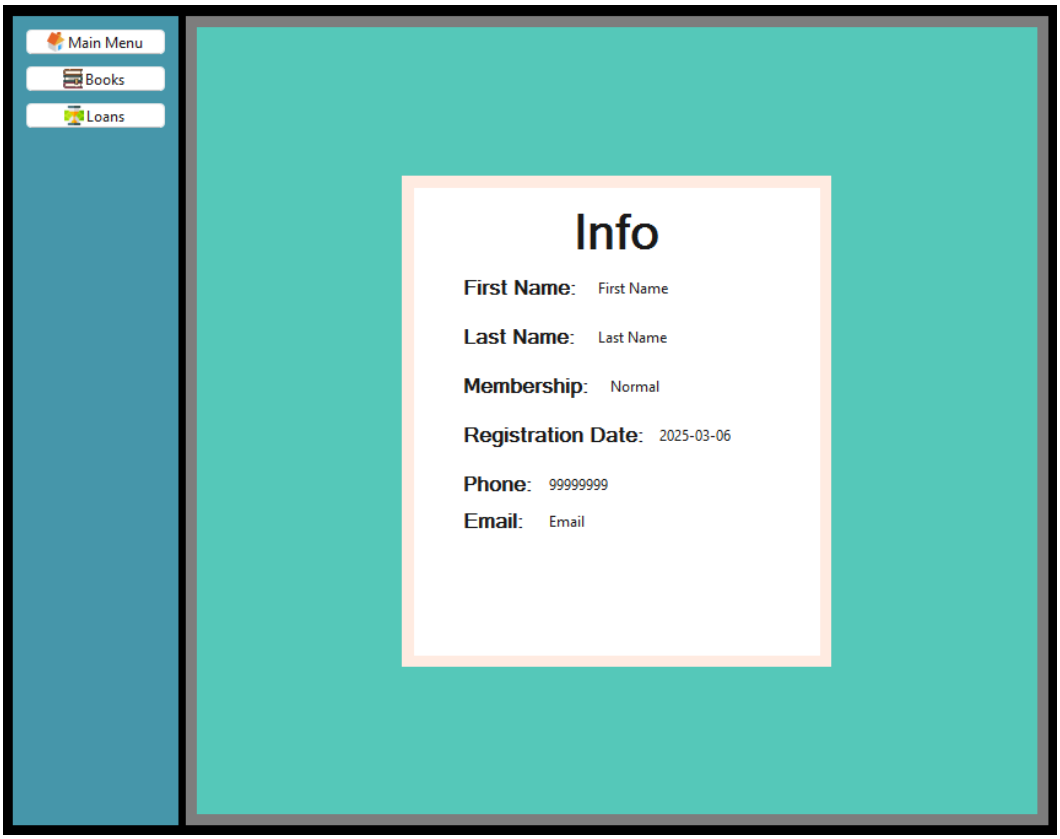


Figure 6. User/staff information of the application.

3.2. Books Menu

Users can search for books using the Books menu located on the left side of the application, as shown in Figure 7. On this page, users are able to search and filter books based on different criteria. If a user wants to search for a specific book by name or a part of the name, they can enter the title and click the Search button. Alternatively, users can leave the search field empty to display all available books. Additionally, they can search for books by a specific author. These cases are demonstrated in Figures 7, 8, and 9.

Moreover, users can filter the book list by categories such as Romance, Horror, Fantasy, and others. They also have the option to search for books published within a specific time range. These filtering options are shown in Figures 10 and 11.

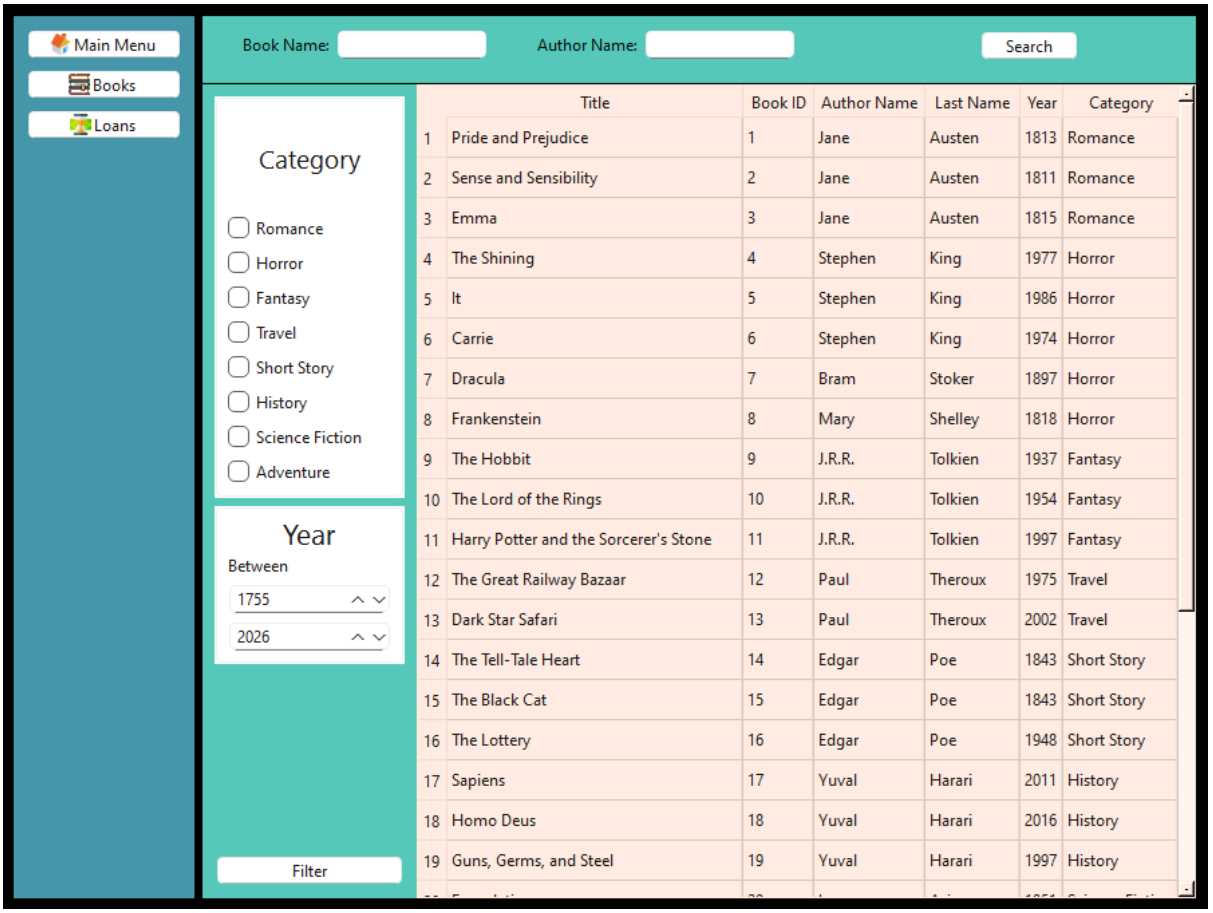


Figure 7. Books menu of application.

Figure 8. Filtered search of the books menu with a book name.

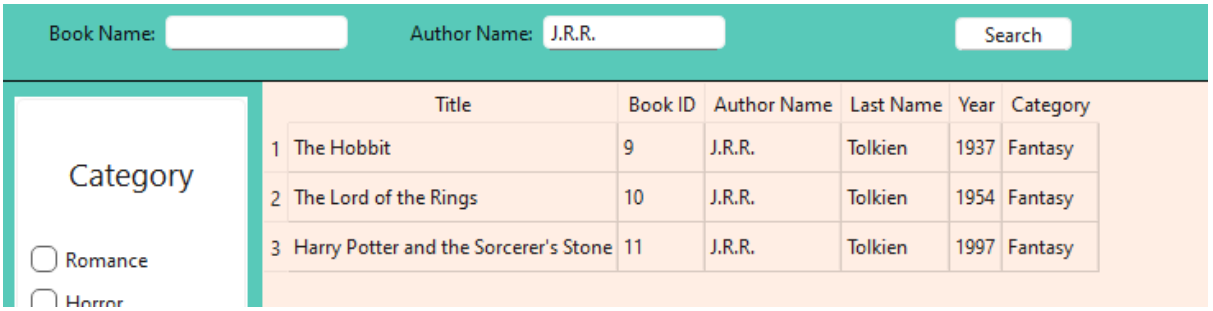


Figure 9. Filtered search of the books menu with an author name.

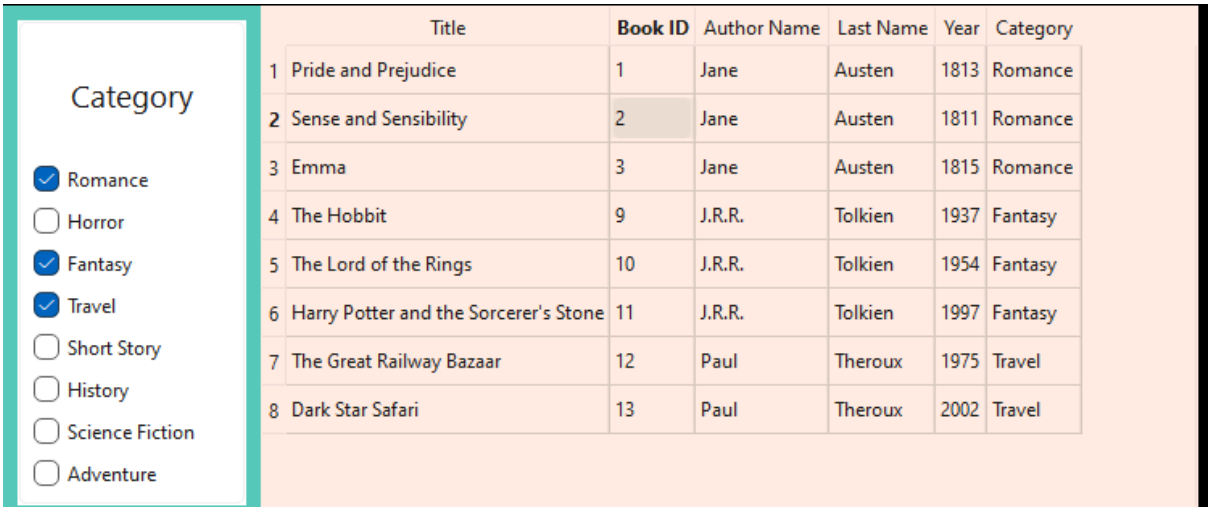


Figure 10. Filtered search of the books menu with a category filtered.

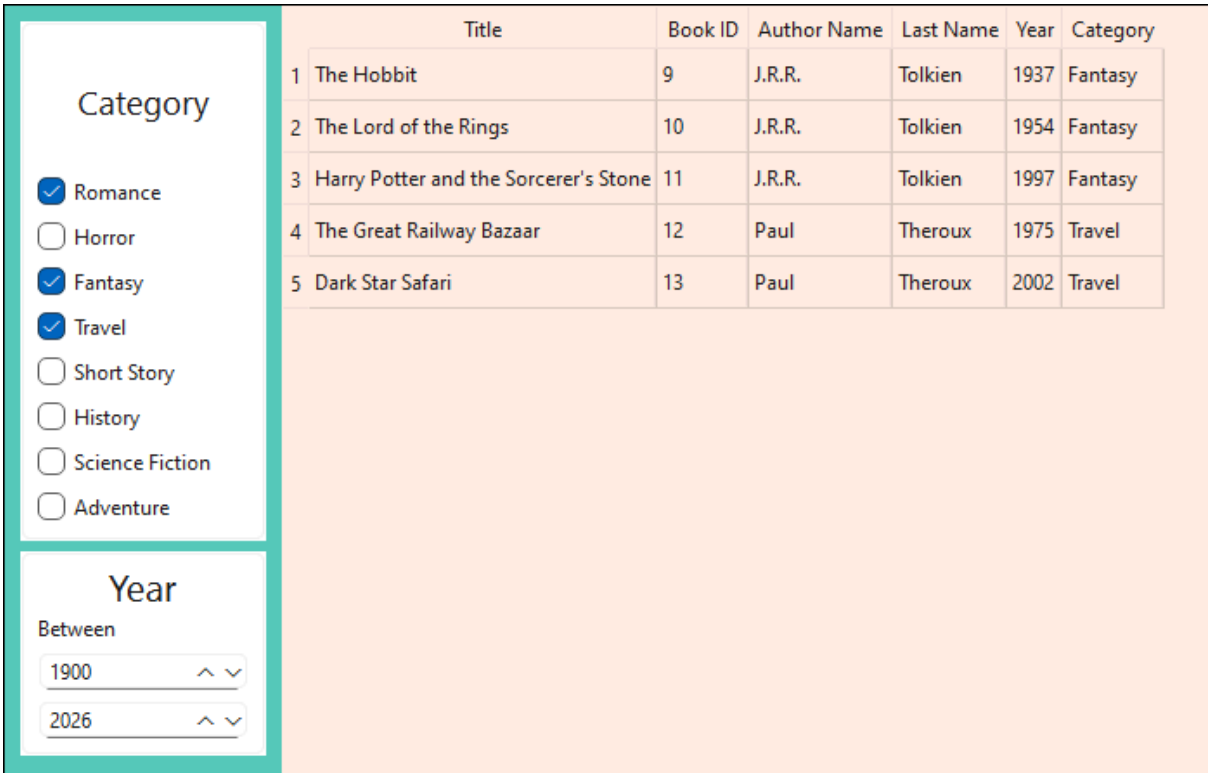


Figure 11. Filtered search of the books menu with category and time filtered.

3.3. Loans Menu

In the Loans menu, users can send loan requests for books, cancel existing loan requests, or view all of their current requests, as shown in Figure 12. To request a book loan, the user enters the Book ID of the desired book and submits the request. Afterward, by clicking the Show All Loans button, the user can check the status of their loan requests.

The loan information displayed includes the Loan ID, Book ID, Book Title, Loan Date, and Loan Status. As illustrated in Figure 13, the user sends a loan request for a book with Book ID 11, and in Figure 14, they query and view this request along with previous loan records.

To delete an existing loan request, users can use the Loan ID. In Figure 15, the user deletes the loan request with Loan ID 3, and in Figure 16, they perform a query to view their updated list of loan requests.

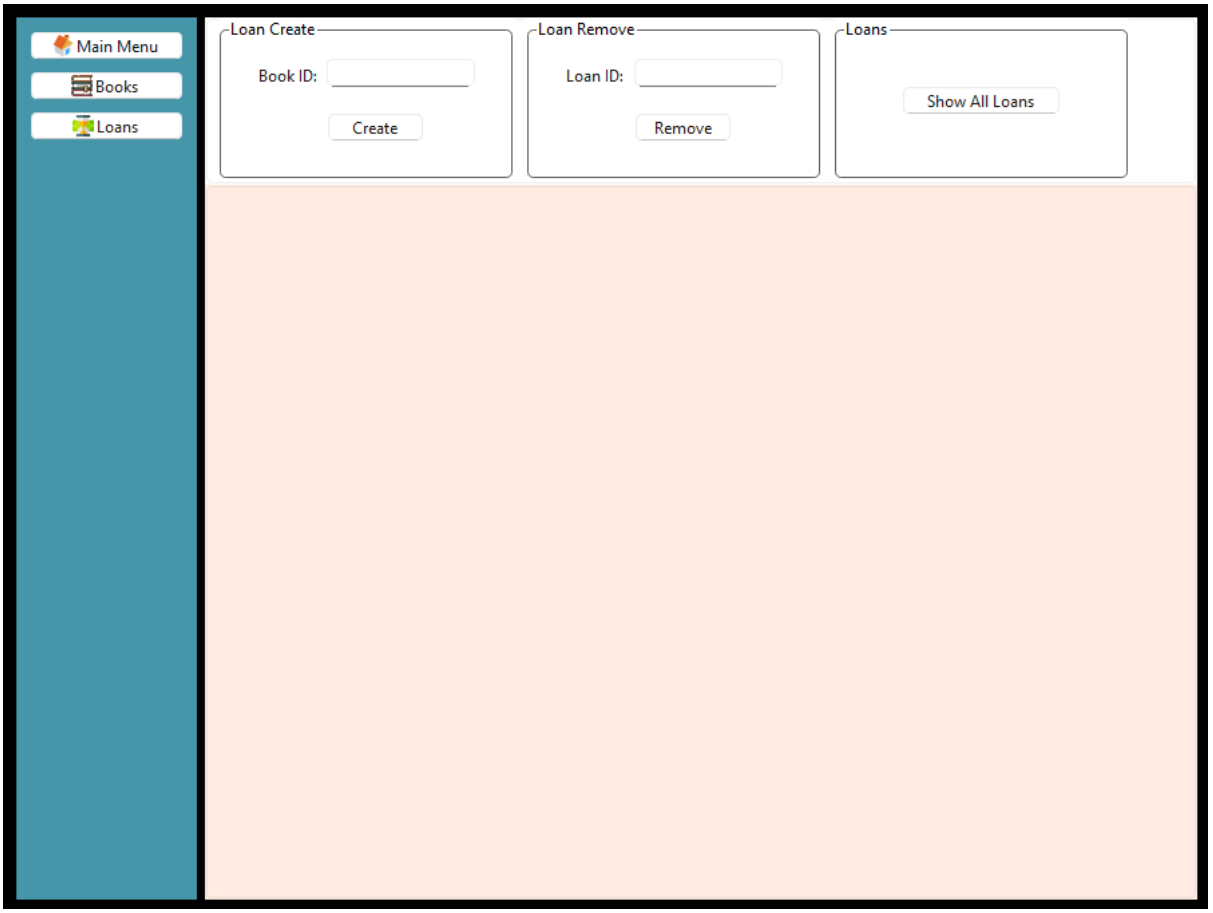


Figure 12. Loan page of the application.

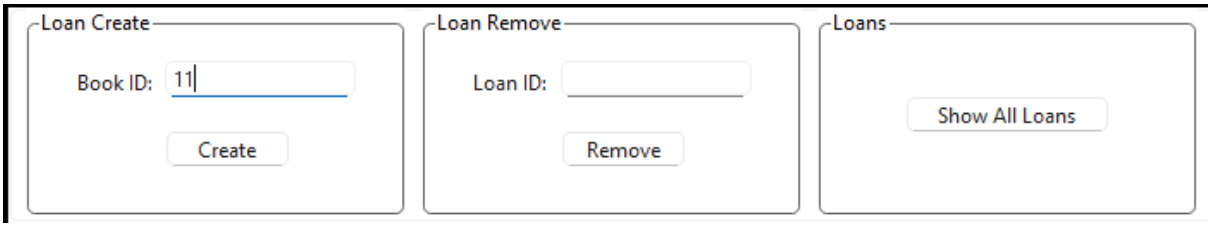
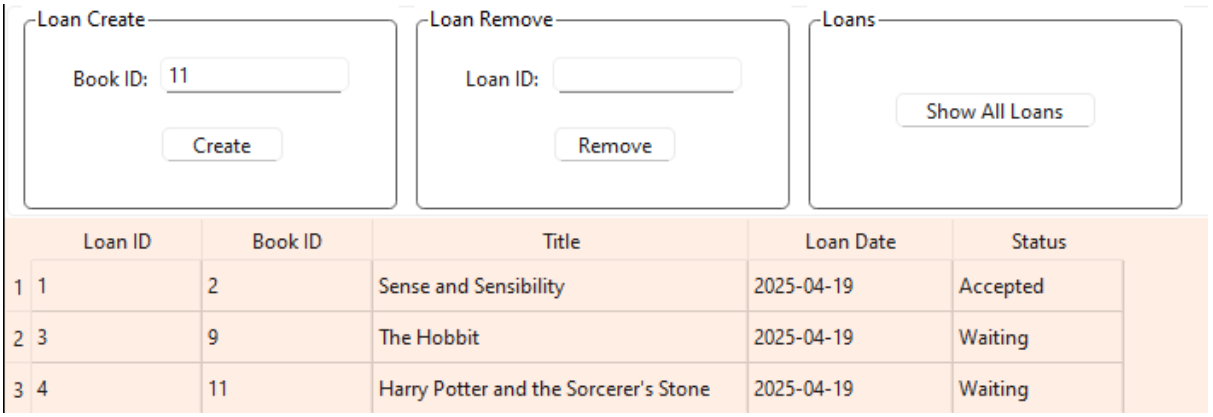
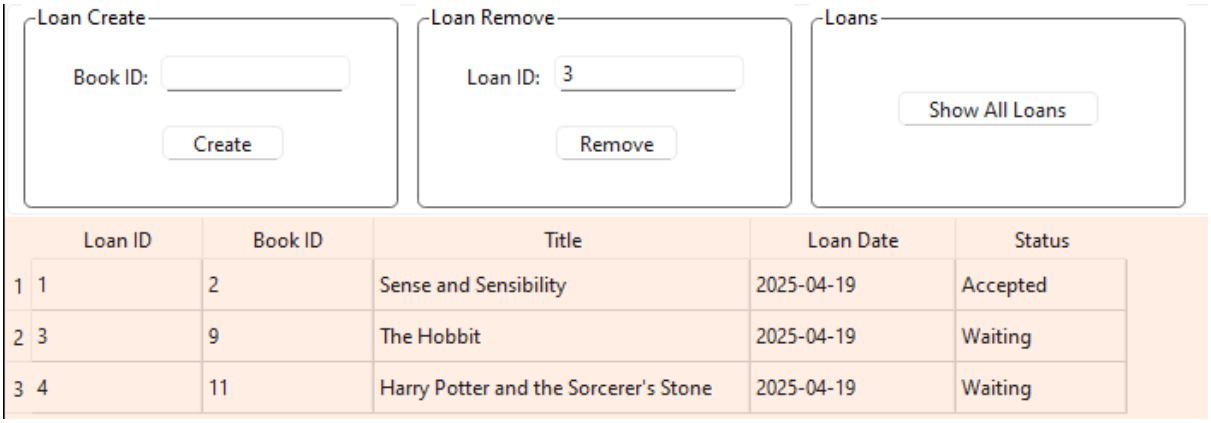


Figure 13. A loan request made for a book with Book ID 11.



	Loan ID	Book ID	Title	Loan Date	Status
1	1	2	Sense and Sensibility	2025-04-19	Accepted
2	3	9	The Hobbit	2025-04-19	Waiting
3	4	11	Harry Potter and the Sorcerer's Stone	2025-04-19	Waiting

Figure 14. List of all loan requests including the newly created one.



	Loan ID	Book ID	Title	Loan Date	Status
1	1	2	Sense and Sensibility	2025-04-19	Accepted
2	3	9	The Hobbit	2025-04-19	Waiting
3	4	11	Harry Potter and the Sorcerer's Stone	2025-04-19	Waiting

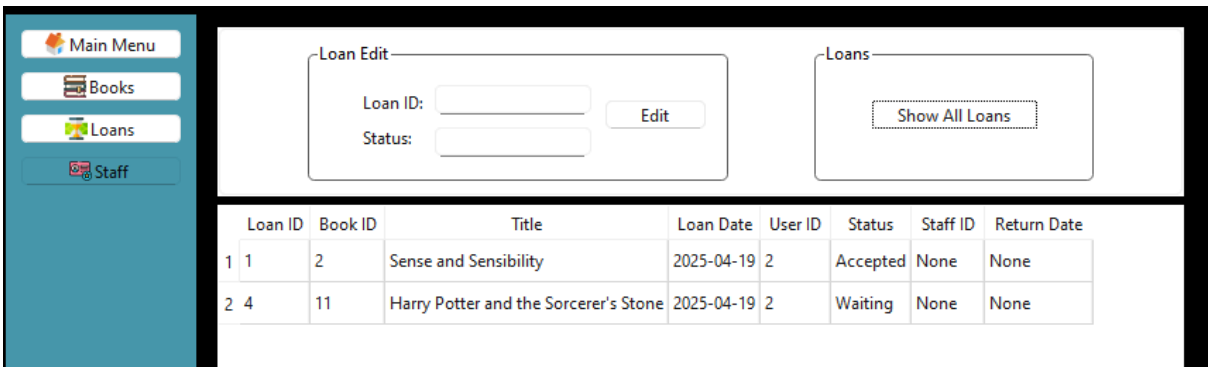
Figure 15. Deleting a loan request using Loan ID 3.

	Loan ID	Book ID	Title	Loan Date	Status
1	1	2	Sense and Sensibility	2025-04-19	Accepted
2	4	11	Harry Potter and the Sorcerer's Stone	2025-04-19	Waiting

Figure 16. Updated list of active loan requests after deletion.

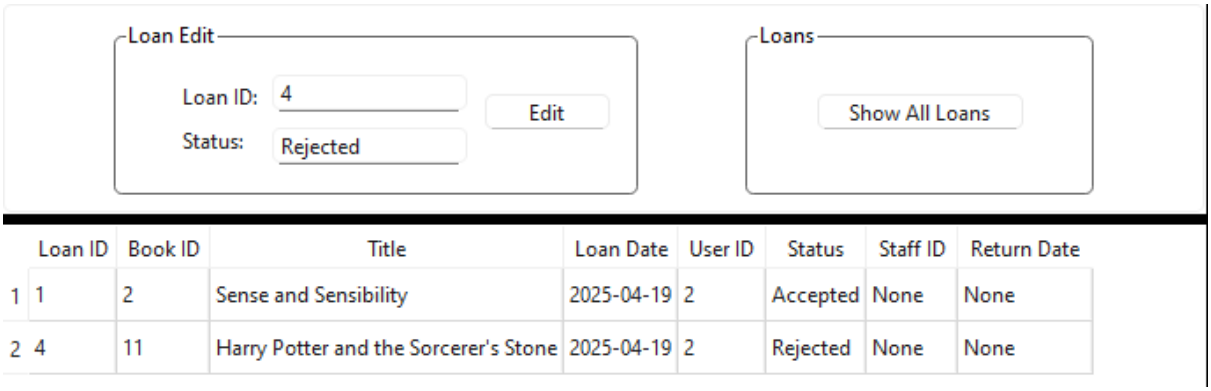
3.4. Staff Menu

If a staff member logs into the application, a Staff menu appears on the left side of the interface. Through this menu, staff members can view all loan requests, as shown in Figure 17. They are able to manage pending loan requests by entering the corresponding Loan ID and updating the status accordingly. An example of a rejected loan request is shown in Figure 18.



	Loan ID	Book ID	Title	Loan Date	User ID	Status	Staff ID	Return Date
1	1	2	Sense and Sensibility	2025-04-19	2	Accepted	None	None
2	4	11	Harry Potter and the Sorcerer's Stone	2025-04-19	2	Waiting	None	None

Figure 17. Staff interface displaying all loan requests in the system.



	Loan ID	Book ID	Title	Loan Date	User ID	Status	Staff ID	Return Date
1	1	2	Sense and Sensibility	2025-04-19	2	Accepted	None	None
2	4	11	Harry Potter and the Sorcerer's Stone	2025-04-19	2	Rejected	None	None

Figure 18. Example of a loan request that has been rejected by a staff member.

4. Conclusion

In this project, a simple yet functional desktop application was developed to support library book management processes. By using Python in combination with the PyQt framework, a graphical user interface was designed to enable users and staff members to interact with the system efficiently. The integration of a MySQL database allowed for reliable data storage and retrieval, including user information, book records, and loan transactions.

The application offers core features such as user registration, login authentication, book search and filtering, and loan request management. Role-based functionality was also implemented, allowing staff members to review and manage user loan requests through a dedicated interface.

Throughout the development process, UML and ER diagrams were used to design and plan the system architecture. Qt Designer was utilized to create the user interface visually, streamlining the frontend development process. Object-oriented programming principles were applied to maintain a clean, modular code structure.

Overall, the project provided valuable experience in software design, UI development, database integration, and full-stack desktop application development using Python technologies.

Additional Information

This project was developed using the Qt framework with the PySide6 library for Python.

The icons and logos used in the application interface were sourced from Flaticon and are credited to the user Freepik.