

Efficient Record-Keeping: A GUI-Based Python Library Management System

Barabad, Marcmiel C., Baldeo, Redrick John C., Jatol, Richmond O., Hilario, Squall Leonhart B.,
Caubang, Angel C., Paleracio, Angelica D., Engr. Fabro, Meshelle N.

Adviser / Instructor, Eulogio “Amang” Rodriguez Institute of Science and Technology (EARIST)
Nagtahan, Manila, Philippines

DOI: <https://doi.org/10.51583/IJLTEMAS.2026.150600032>

Received: 07 June 2026; Accepted: 12 June 2026; Published: 03 July 2026

ABSTRACT

Manual library record-keeping often leads to misplaced records, slow book searching, inaccurate borrower tracking, and difficulty in monitoring available and borrowed books. To address these problems, this study developed **Efficient Record-Keeping: A GUI-Based Python Library Management System**, a desktop-based application designed to help manage library book records and borrowing transactions in a faster, more organized, and more accurate way. The system was developed using Python as the main programming language, a graphical user interface for user interaction, and SQLite as the database management system for permanent record storage. The system includes essential library management features such as adding book records, viewing available books, searching book information, editing book details, deleting records, borrowing books, returning books, and generating reports. Through database integration, book and borrower records remain saved even after the program is closed. The system also reduces manual work by allowing users to retrieve records quickly and update book status efficiently. Overall, the developed Library Management System provides a practical and user-friendly solution for improving library record-keeping, book monitoring, and borrowing management.

Keywords: Library Management System, Python, SQLite, Graphical User Interface, Book Records, Borrowing Management, Record-Keeping

INTRODUCTION

Background of the Study

Libraries are a crucial foundation in education, particularly in research, homework's and other academic activities. Libraries provide books, references, and space for the student to study. While it helps highly in education, Libraries are hard to maintain using traditional methods for borrowing books, tracking inventory, and students' information. However this problem can easily be solved by integrating technology and the use of automated systems.

Traditional operations in libraries highly rely on logbooks or physical records to track and monitor books and students in the library. Without the use of technology and automated systems libraries can be inefficient and prone to inaccurate information. These limitations slow overall transaction tracking and reduce the productivity affecting the quality of library services for both students and staff.

Libraries play an important role in supporting learning, research, and access to information. They serve as organized spaces where books and other learning materials are stored, borrowed, returned, and monitored. However, many small libraries and school-based libraries still rely on manual record-keeping methods such as logbooks, index cards, printed lists, and handwritten borrower records. Although these methods may work for a limited number of books, they can become inefficient as the number of library materials and borrowers increases.

Manual library management can lead to several problems. Book records may be misplaced, borrower information may be incomplete, and tracking borrowed or returned books may take a long time. Library staff may also

experience difficulty in searching for specific books, updating book details, checking availability, and preparing reports. These problems can affect the efficiency of library operations and may cause delays for both library staff and users.

With the help of computerized systems, library management can become faster, more accurate, and more organized. A computerized Library Management System allows users to store book records digitally, search information quickly, update book details, monitor borrowed books, and generate reports. It reduces the need for repetitive manual work and helps prevent common record-keeping errors.

This study focuses on the development of **Efficient Record-Keeping: A GUI-Based Python Library Management System**. The system is designed to help library staff manage book records and borrowing transactions through a simple graphical user interface. It uses Python for system development and SQLite for database storage. Through the graphical interface, users can add, view, search, edit, delete, borrow, and return book records without using complicated commands.

The use of SQLite allows the system to store records permanently. This means that book records and borrowing information remain available even after the application is closed. The system also helps improve record accuracy because library information is stored in an organized database rather than scattered manual records.

Overall, the system was developed to provide an efficient and user-friendly solution for library record management. It aims to improve book monitoring, reduce manual errors, speed up record retrieval, and support better library operations through a computerized and database-connected application.

Statement of the Problem

This study aims to develop a GUI-based Python Library Management System that can improve the efficiency and accuracy of library record-keeping.

Specifically, this study seeks to answer the following questions:

1. How can the system automate the process of managing library book records?
2. How can the system help users search, edit, delete, borrow, and return book records efficiently?
3. How can database integration improve the storage and retrieval of library records?
4. How can a graphical user interface improve the usability of the system for library staff?
5. How can the system reduce common errors found in manual library record-keeping?

OBJECTIVES OF THE STUDY

General Objective

The general objective of this study is to develop **Efficient Record-Keeping: A GUI-Based Python Library Management System** that can help manage library book records and borrowing transactions in a more organized, accurate, and efficient way.

Specific Objectives

Specifically, this study aims to:

1. Design a system flow for managing library records and borrowing transactions.
2. Develop the system using Python programming language.
3. Create a user-friendly graphical user interface for easier system navigation.
4. Integrate an SQLite database for permanent storage of book and borrower records.

5. Provide library management functions such as adding, viewing, searching, editing, deleting, borrowing, and returning books.
6. Improve the speed and accuracy of library record retrieval.
7. Generate reports that help monitor library records and book status.
8. Test the functionality of the system to ensure that it performs according to its intended purpose.

Significance of the Study

This study is significant because it provides a computerized solution for improving library record-keeping and book management.

Library Staff. The system can help library staff manage book records more efficiently. It allows them to add, search, update, delete, and monitor book records without relying on manual logbooks.

Students and Library Users. Students and other library users may benefit from faster book searching and borrowing transactions. The system helps reduce delays in checking book availability and borrower records.

School or Institution. The system can help the school or institution improve the organization and monitoring of library materials. It supports better record accuracy and more efficient library operations.

Future Researchers and Developers. This study can serve as a reference for future developers who want to create or improve similar library management systems using Python, GUI development, and database integration.

SCOPE AND LIMITATIONS

This study focuses on the development of a desktop-based Library Management System using Python and SQLite. The system includes basic library management functions such as adding book records, viewing records, searching books, editing book information, deleting records, borrowing books, returning books, and generating reports.

The system is intended for academic and small library use. It is designed to run locally on a computer and does not require internet access. However, it does not yet support advanced features such as online access, cloud storage, barcode scanning, mobile application support, multi-user network access, email notifications, or automatic overdue penalty computation.

Future improvements may include user login authentication, barcode scanner support, due-date reminders, borrower account management, cloud backup, and web-based access.

REVIEW OF RELEVANT THEORY, STUDIES, AND LITERATURE

This chapter presents the theories, studies, and literature relevant to the development of **Efficient Record-Keeping: A GUI-Based Python Library Management System**. It discusses library management principles, database systems, computerized record-keeping, graphical user interface applications, Python programming, and previous studies on library management systems. These concepts provide the foundation for understanding the importance of developing a computerized system to manage library records efficiently.

Library Management Systems

Library Management Systems (LMS) are designed to automate and streamline the handling of books, borrowing transactions, and member records. Traditionally, libraries relied on manual methods such as ledger books and index cards, which often led to misplaced records, delayed transactions, and difficulty in tracking borrowed books. A well-developed LMS enables libraries to store, search, and manage book information systematically while also monitoring borrowing and return activities.

According to Koohang et al. (2017), effective LMS solutions allow staff to manage book cataloging, member registration, borrowing, and returning efficiently. A GUI-based LMS reduces human error, improves accessibility of records, and speeds up transaction processing.

Computerized management systems

Computerized management systems are highly used in handling records, processing data, and improving operational efficiency. In contrast to manual record-keeping, computerized systems provide faster data entry, easier access to information, and reduced human error. In a Library Management System, a computerized system allows users to browse books, borrow books, record books, and add books to the library system. Moreover, the system allows users to track and review transactions which significantly makes library operations faster and efficient. According to Mardiyati et al. (2025) A web-based book inventory management system is crucial in reducing administrative workload which improves inventory accuracy and enhances users' experience in school libraries. Furthermore Lathifuddin et al. (2026) Library management system simplifies and digitalizes library operations, reducing inefficiencies and enhancing user experience.

Database Management System (SQLite)

A Database Management System (DBMS) is essential in software applications for storing, organizing, and retrieving data efficiently. SQLite is a widely used embedded relational database because it is lightweight, serverless, and easy to integrate into applications without requiring complex setup.

According to Urwin (2024) explained that SQLite is a self-contained database engine that stores data in a single file, making it highly portable and suitable for small-scale and standalone applications. Its simplicity allows developers to focus more on application logic rather than database configuration.

According to GeeksforGeeks (2025) further described SQLite as a fast, reliable, and zero-configuration database system commonly used in mobile and desktop applications. It is also widely supported in programming languages such as Python, making it ideal for student-developed systems like library management projects.

However, despite its advantages, SQLite has architectural limitations, such as limited concurrency support and reduced performance in large-scale multi-user networks. Because of this, it is more suitable for small to medium applications where simplicity and local data storage are prioritized. Overall, the literature shows that SQLite is an effective database solution for GUI-based library management systems due to its portability, efficiency, and ease of integration.

Library Management System

According to Singh (2024), The Library Management System (LMS) is a computerized application developed to streamline and manage the fundamental operations of a library efficiently. Its primary purpose is to provide a user-friendly platform that automates the administration of library resources, member records, and transaction processes, thereby enhancing operational efficiency and minimizing the manual workload of library personnel. The proposed system is capable of performing several essential library functions, including book cataloging, member registration, book borrowing, book return processing, and due date monitoring. It also contains modules that facilitate the addition of new books to the library collection, modification of existing book details, and removal of outdated or unavailable materials from the database. In terms of member management, the system enables the registration of users, updating of personal information, and monitoring of borrowing records and transaction histories.

Furthermore, the Library Management System incorporates an advanced search feature that allows users to locate books efficiently using various search parameters such as title, author, genre, and International Standard Book Number (ISBN). The system also supports the generation of comprehensive reports related to library utilization, overdue materials, and member activities. These reports can assist library administrators in decision-making processes and in improving the overall quality of library services.

GUI-Based Applications

Graphical User Interface (GUI)-based applications play a pivotal role in optimizing the efficiency and usability of modern library management systems. Compared to conventional manual methods or command-line interfaces, these applications offer an interactive, user-friendly environment for managing library records, thereby streamlining operations such as borrowing, searching, and updating data while minimizing operational errors (Valdez, 2022).

According to Kumar and Singh (2022) observed that traditional, manual library systems are inherently inefficient and highly susceptible to human error. They emphasized that computerized systems significantly enhance cataloging, searching, and data management, which ultimately improves overall system reliability.

Furthermore, interface design directly influences system performance and user reception. Siregar and Wahyuni (2023) demonstrated that well-organized interfaces optimize navigation and elevate the user experience. In alignment with these findings, According to Lwoga and Sangeda (2021) asserted that usability, functionality, and reliability serve as critical determinants of user satisfaction and system adoption, noting that poorly designed interfaces often discourage engagement even if the system is feature-rich.

According to Ariyanti and Supriyanto (2023) further highlighted that implementing proper UI and UX design principles minimizes user friction and learning curves. This is supported by Valdez (2022), who emphasized that factors such as learnability and operational efficiency directly govern how effectively users interact with library software. Additionally, the capacity for system adaptability is crucial; Rijal (2024) noted that interface customization—such as that seen in platforms like KOHA—empowers librarians to maximize system features and tailor the software to specific institutional needs.

Overall, the existing literature underscores that GUI-based library management systems drastically reduce manual workloads, enhance record accuracy, and improve user satisfaction. However, a significant gap remains: current research heavily favors web-based systems, leaving a notable scarcity of literature regarding desktop-based GUI applications. This creates a distinct opportunity for further development and exploration utilizing local, Python-based desktop frameworks.

Automation of the System

Existing studies on library management systems are highly productive for education. According to Jaiswal and Pandey (2024), Library Automation reduces manual process and errors, allowing the staff to focus on user engagement and service improvement, while improving the accuracy of information.

The system can automate and organize library operations by using a GUI-based application. By using GUI-based applications, the researchers can replace the manual processes with digital tools that are much easier, faster, and much more accurate to use.

However, these systems impose challenges according to Dagogo (2024), Library Automation in the 21st century faces challenges like insufficient funding, and technology. But proper planning, and management support can improve efficiency and maintain a quality system.

Input Validation and Error Handling

Input validation ensures that only correct and complete information is stored in the database. For library management, validation can prevent duplicate book IDs, invalid member numbers, incorrect return dates, or missing fields. Error handling further enhances reliability by notifying users of invalid entries without crashing the system. According to Birajdar (2025), robust input validation is crucial in LMS to maintain accurate library records and prevent data corruption.

Related Studies

Several studies have explored computerized library management systems:

1. **Koohang et al. (2017)** developed a GUI-based LMS that automated cataloging and borrowing processes, reducing manual workload and improving record accuracy.
2. **Dewan et al. (2020)** implemented a Python-based LMS with SQLite database for a school library, enabling real-time book availability tracking and automated report generation.
3. **Hossain et al. (2019)** designed an LMS with VS Code interface, emphasizing user-friendly navigation and effective record-keeping for both books and members.

These studies highlight that computerized LMS solutions improve efficiency, reduce errors, and allow administrators to monitor library operations effectively.

Synthesis of Reviewed Literature

The reviewed literature emphasizes the importance of integrating **database management, Python programming, GUI design, and input validation** to create an efficient and reliable Library Management System. Manual methods are prone to errors and inefficiencies, whereas computerized systems streamline operations, improve accessibility, and maintain accurate records.

Based on previous studies and LMS principles, the proposed system in this study focuses on:

- Storing book and borrower information in an SQLite database
- Automating borrowing and return processes
- Providing search, edit, and delete functionality
- Implementing GUI for easy navigation
- Applying input validation to prevent incorrect entries
- Generating reports for monitoring book status and library transactions

This framework ensures a practical, user-friendly, and efficient solution for managing library records in an educational setting.

This chapter presents the research design, system development process, system design, database design, tools and technologies used, system features, testing procedure, and evaluation method used in the development of **Efficient Record-Keeping: A GUI-Based Python Library Management System**. It explains how the system was planned, designed, developed, tested, and implemented to provide an efficient and user-friendly platform for managing library book records and borrowing transactions.

METHODOLOGY

Research Design

The researchers used a system development approach in creating the Library Management System. The system aims to provide an organized and efficient way of managing and organizing through a graphical user interface (GUI). Python programming language was used for the development of the application. The researchers utilized Figma to design the components of the GUI. While SQLite database was integrated into the system to provide an accessible digital library management system for users.

The Library Management System was designed to carry out important library operations, including adding books, borrowing books, updating book information, deleting books, and viewing book records. To maintain accuracy and dependability, input validation and error-handling features were incorporated into the system. The

researchers followed a systematic development approach consisting of planning, designing, coding, testing, and implementation to ensure the successful development of the system.

This study employed a **system development approach**, which is suitable for developing a functional desktop-based library management system. This approach focuses on planning, designing, developing, testing, and implementing a system to solve a real-world problem, in this case, manual library record-keeping.

The system was developed to address common problems in manual library operations, such as misplaced book records, slow searching, inaccurate borrower tracking, and difficulty in generating reports. The study aimed to provide a computerized solution that ensures accurate record management, efficient borrowing and returning, and reliable reporting.

The system was developed using **Python** for programming logic, a graphical user interface for user interaction, and **SQLite** as the database management system for permanent storage.

System Development Process

The development of the Library Management System followed a series of structured phases to ensure the system's efficiency, reliability, and user-friendliness.

Planning

During the planning stage, the researchers identified the problems in manual library management, including misplaced records, difficulty tracking borrowed books, and inefficient record retrieval. They also determined the main system requirements, including book record management, borrower management, borrowing and return processing, search functionality, editing and deleting records, and report generation.

System Design

System design involved creating flowcharts and interface mockups to guide the development process. The design defined the main system modules, including:

- **Book Management Module** – for adding, viewing, editing, and deleting book records.
- **Borrower Management Module** – for registering library users.
- **Borrow/Return Module** – for processing book loans and returns while updating availability.
- **Search and Reports Module** – for searching books and generating transaction summaries.
- **Exit/Logout Module** – for safely closing the application.

The system was designed to use a **GUI interface** to make navigation and record management intuitive for library staff.

Development

In this phase, the researchers carried out the actual coding and implementation of the system. The application was developed using the Python programming language together with the Figma library for the graphical user interface. Furthermore, a SQLite database was integrated to manage the storage and retrieval of library records efficiently. Essential functions such as Add, View, Edit, Delete, Search, Borrow, and Report generation were successfully incorporated into the system.

Database Integration

An **SQLite database** was integrated into the system for permanent storage of library records. The database stores the following tables:

- **books** – stores book ID, title, author, year, category, and availability status.
- **borrowers** – stores borrower ID, name, and contact information.
- **transactions** – records book loans and returns with dates.
- **activity_log** – records all administrative actions such as adding, editing, deleting books or transactions.

The database ensures that all records remain saved even when the system is closed. Python's built-in sqlite3 library was used to execute SQL commands such as INSERT, SELECT, UPDATE, and DELETE.

Testing

Functional testing was conducted to verify that all features worked correctly. This included testing:

- Adding, editing, and deleting book records.
- Borrowing and returning books.
- Searching for books by title, author, or category.
- Generating borrowing and transaction reports.
- Input validation to prevent invalid or incomplete entries.

Errors and issues identified during testing were corrected through debugging to ensure reliability.

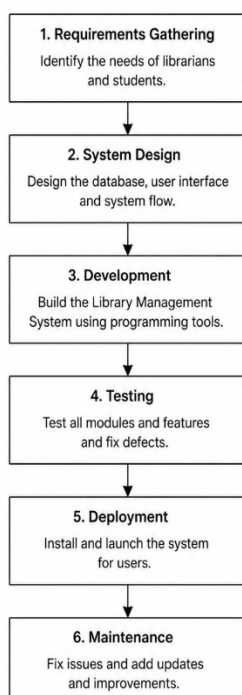
Implementation

After testing, the final system was implemented as a desktop-based Library Management System. The system allows library staff to manage book records, borrower information, borrowing and return transactions, and generate reports efficiently. Its GUI ensures ease of use, while SQLite provides permanent and reliable data storage.

Systems Design

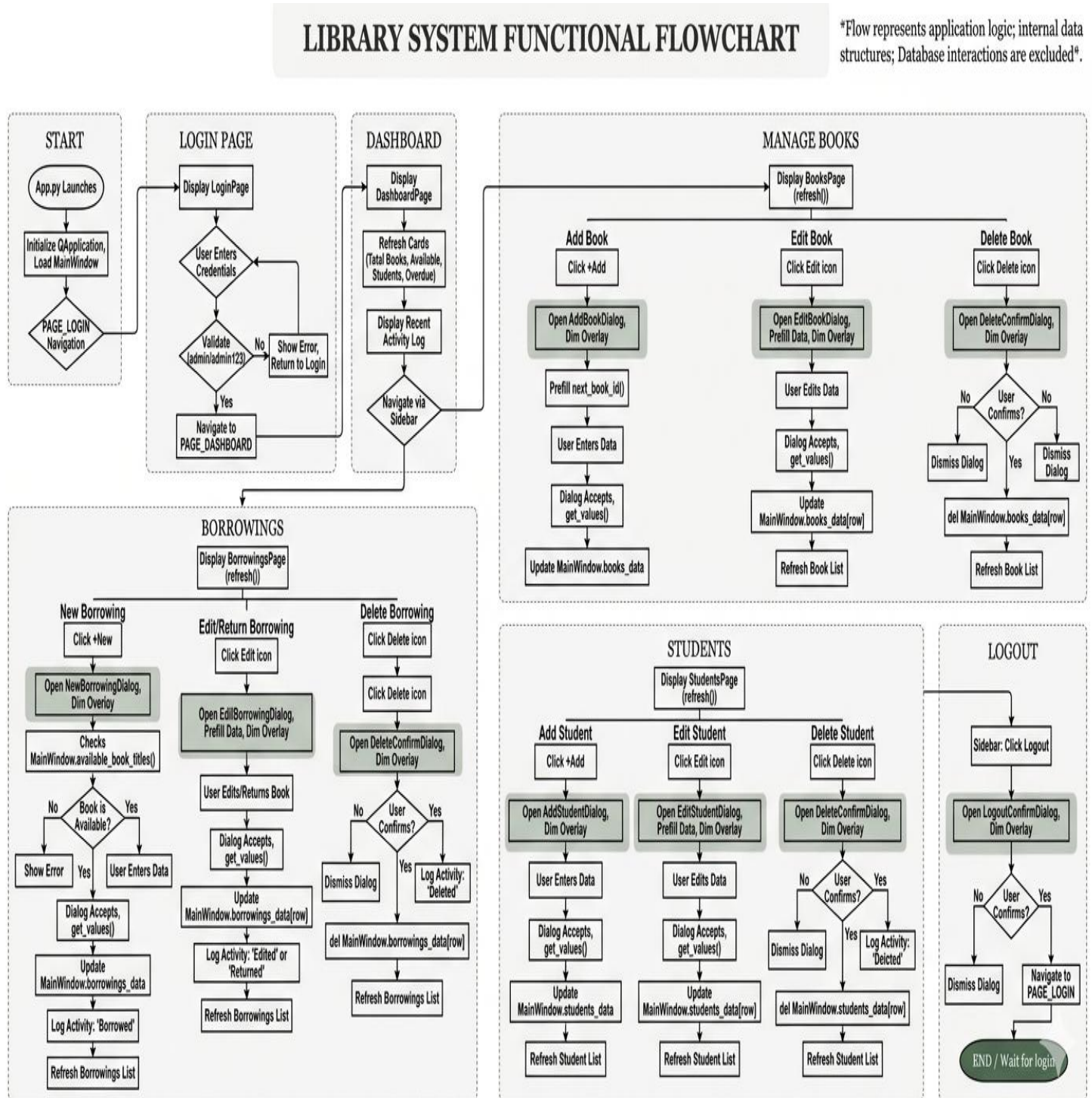
Figure 1. Waterfall Model of the System

Waterfall Model for Library Management System



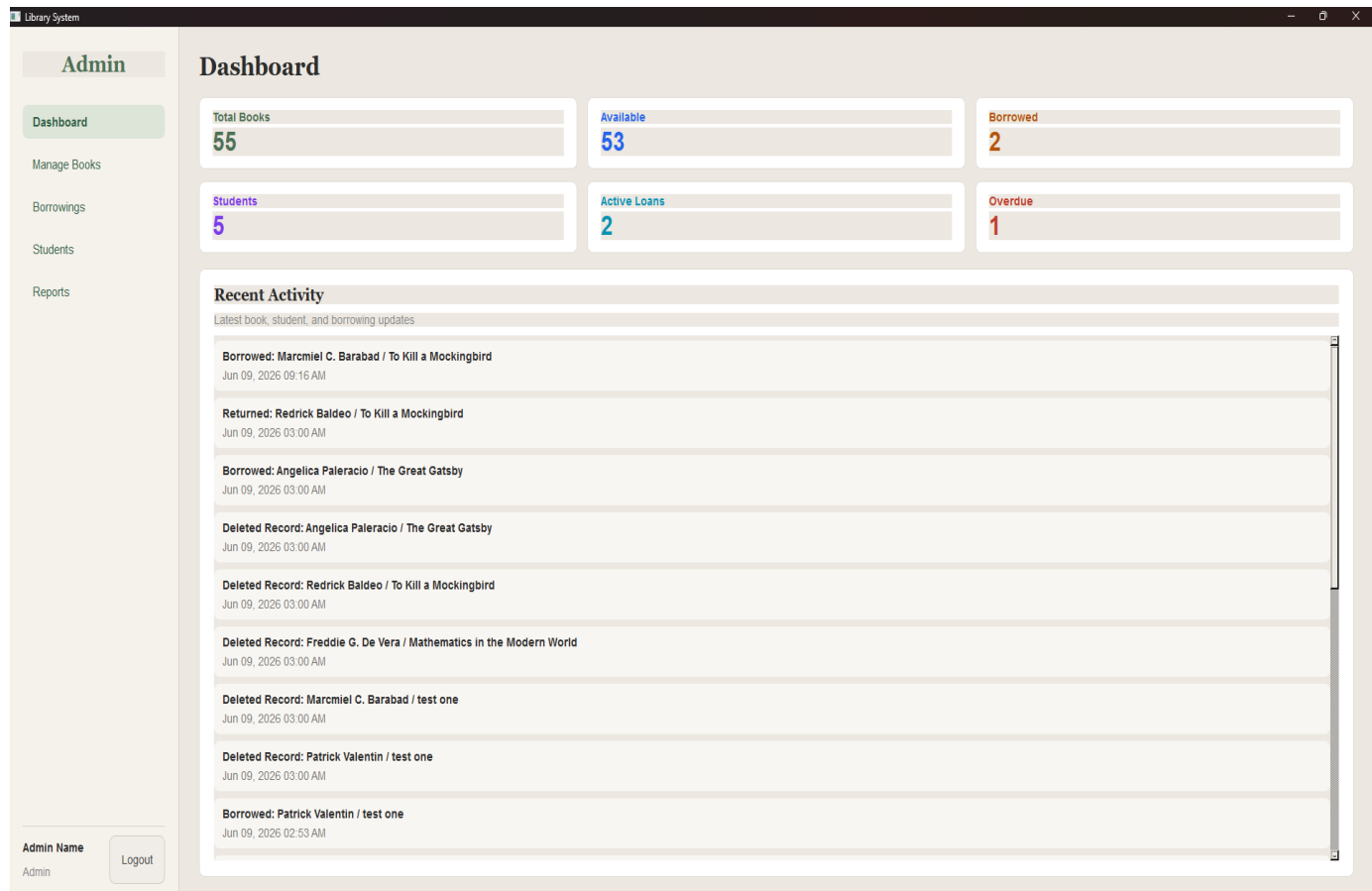
This figure illustrates the Waterfall Model followed in developing the Library Management System. It depicts the sequential phases of system development, including requirements gathering, system design, development, testing, deployment, and maintenance, providing a structured approach to ensure the system meets user requirements and operates effectively.

Figure 2. Flowchart of the System



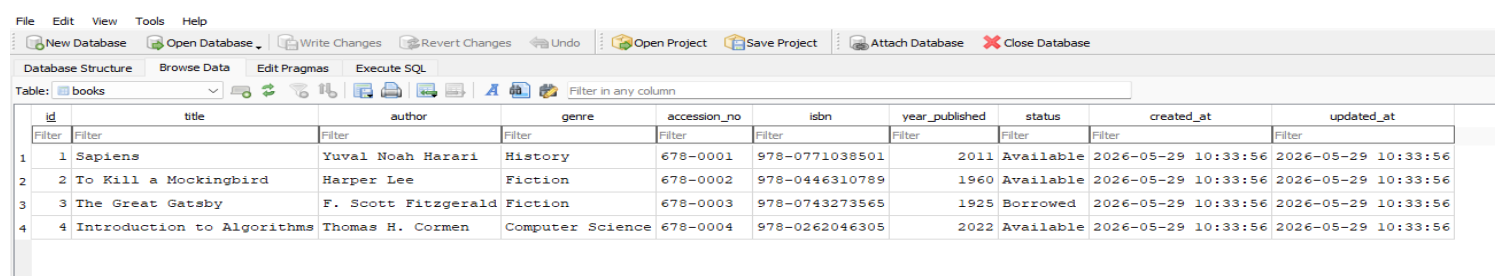
This figure illustrates the operational flow of the Library Management System. It begins with administrator login validation, proceeds to the main dashboard, and allows access to modules for student management, book management, borrowing, viewing records, and database operations. The flowchart shows the sequential logic of each module, including record checks, confirmations, database updates, and logout, providing a clear overview of system processes.

Figure 3. Library Management System Dashboard (GUI)



This figure shows the graphical user interface of the Library Management System dashboard. It provides an overview of the library's current status, including total books, available books, borrowed books, total students, active loans, and overdue items. The dashboard also displays recent activity such as book borrowings and returns, allowing administrators to monitor library operations efficiently.

Figure 4. Database Interface and Records in the Library Management System



This figure shows the Library Management System running in the Python terminal with the SQLite database connected. It displays stored book records, including book ID, title, author, and availability status. This interface demonstrates how the system retrieves and presents database records, supporting efficient record-keeping and management.

System Features

The developed system includes the following features:

- **Add Book Record** – register new books with details such as title, author, year, and category.
- **View Book Records** – display all books in a structured table.

- **Edit/Delete Book Records** – update or remove book information as needed.
- **Borrow/Return Processing** – manage book lending and returning while updating availability.
- **Search Function** – find books by title, author, or category.
- **Report Generation** – generate transaction summaries and book availability reports.
- **Activity Logging** – record all administrative actions for accountability.
- **Exit/Logout** – safely close the application and maintain data integrity.

Tools and Technology Used

The development of the Library Management System utilized the following tools and technologies:

- **Programming Language:** Python – Used for system logic, database interaction, and implementing the functionalities of the application.
- **GUI Framework:** PyQt5 – Employed to create an interactive and user-friendly graphical interface.
- **UI/UX Design:** Figma – Utilized for designing the system interface layout and improving user experience.
- **Database:** SQLite3 – Chosen for lightweight, file-based, and permanent storage of book and borrower records, as well as transaction logs.
- **Development Tools:** VS Code – Used as the integrated development environment (IDE) for writing, debugging, and testing the Python code.

This combination of technologies ensures a robust, functional, and visually appealing library management system with efficient database storage and user interaction.

Testing Procedure

Testing was conducted using **functional testing methods**. Each module was tested to verify that it produces the expected outcome. Both valid and invalid inputs were tested to ensure robustness, including:

- Blank or incomplete entries.
- Duplicate book or borrower IDs.
- Invalid date formats for borrow/return transactions.
- Attempting to borrow unavailable books.

Evaluation of the System

The system was evaluated based on:

- **Functionality** – ability to manage books, borrowers, and transactions effectively.
- **Accuracy** – correctness of book records, borrower details, and transaction records.
- **Usability** – ease of navigation and clarity of the GUI.
- **Reliability** – proper data storage and retrieval from SQLite.
- **Efficiency** – speed of operations such as search, borrow, return, and report generation.

Summary

The methodology of this study followed a system development approach consisting of planning, system design, development, database integration, functional testing, and implementation. The Library Management System was developed using Python, and SQLite, with OOP principles for better code structure.

Through this methodology, the researchers created a functional, user-friendly system that efficiently manages library records, book availability, borrowing and return transactions, and report generation. The system demonstrates the practical application of Python programming, GUI development, database management, and input validation in improving library operations.

RESULTS AND DISCUSSION

This chapter presents the results and discussion of the developed **Efficient Record-Keeping: A GUI-Based Python Library Management System**. It highlights the system overview, graphical interface, key modules, functional testing, and the overall performance of the library management system.

System Overview

The Library Management System is a desktop-based application developed to assist users in efficiently organizing, monitoring, and managing library records and transactions. The system was designed to address the need for a more organized, accurate, and accessible method of handling library operations without relying on manual record-keeping procedures.

The application operates through a menu-driven Graphical User Interface (GUI) created using Figma and is integrated with a SQLite relational database for reliable data storage and management. Upon launching the program, the system automatically retrieves existing library records from the database, allowing users to access information immediately without additional configuration.

The main objective of the system is to enhance the efficiency, accuracy, and organization of library management processes. This is achieved through input validation features that minimize encoding errors, systematic categorization of records, and report generation capabilities that provide users with a clear overview of library activities.

Overall, the Library Management System provides a comprehensive platform for managing library operations, including adding, viewing, editing, deletion, searching, and report books within a unified and user-friendly interface.

System Features and Results

Add Book Records

The Add Book Records feature allows the user to enter new books into the system database efficiently. This function stores important book details such as title, author, genre, ISBN, and publication year. It helps maintain an organized and updated collection of library resources. The system validates the entered information to reduce errors and duplication of records. Users can easily expand the inventory whenever new books are acquired by the library. This feature improves the accuracy of record management and minimizes manual documentation. It also ensures that all newly added books are accessible within the system for future transactions. Overall, the Add Book Records feature contributes to a more systematic and efficient library management process.

The Add Book module allows the user to enter a new book record, including details such as title, author, year, category, and availability status.

Result: The system successfully saved all valid book records in the SQLite database. Input validation prevented incomplete or duplicate entries from being saved. This ensured accurate and organized book records.

View Book Records

The View Records feature enables users to access and examine all stored book records in the database. It provides a comprehensive display of available books and related transaction details. This function allows users to quickly search and retrieve information when needed. Users can monitor the availability and status of books more efficiently through this feature. It also supports better organization of library resources by presenting records in a structured format. The feature minimizes the time required to locate specific information manually. Additionally, it improves decision-making by providing accurate and updated data regarding library materials. Overall, the View Records feature enhances accessibility and convenience in managing library information.

The View module displays all stored books in a structured table format. Users can see book titles, authors, categories, years, and availability status.

Result: All stored book records were displayed correctly. Users could efficiently review the library's collection, improving record accessibility compared to manual logbooks.

Editing Book Records

The Editing Book Records feature allows authorized users to modify existing book information in the system. This function ensures that all records remain accurate and up to date. Users can edit details such as book title, author, genre, publication year, and publication information whenever necessary. The feature helps maintain consistency and reliability within the database. It also prevents the accumulation of outdated or incorrect information in the system. By allowing updates, the system can adapt to changes in the library inventory more effectively. This function contributes to improved record accuracy and efficient management of resources. Overall, the Editing Book Records feature supports the continuous maintenance of a reliable library database.

Delete Book Record

The Delete Book Record feature enables users to remove unnecessary or outdated book records from the system database. This function is useful for managing books that are damaged, lost, or no longer available in the library. It helps maintain the accuracy and relevance of stored information. The feature also prevents clutter and duplication within the database. Authorized users can securely delete records while ensuring proper management of library data. This process contributes to a more organized and efficient record-keeping system. Additionally, it improves the overall performance of the application by eliminating inactive records. Overall, the Delete Book Record feature supports effective database maintenance and resource management.

Borrow Book Record

The Borrow Book Record feature manages the process of issuing books to users. It records essential transaction details such as borrower information, borrowed books, issue dates, and return deadlines. This feature ensures proper monitoring of borrowed materials within the library system. It helps librarians track book circulation more accurately and efficiently. The system also minimizes the risk of lost or unreturned books through organized transaction records. Borrowing information can be retrieved easily whenever needed for verification purposes. This function improves accountability among borrowers and library staff. Overall, the Borrow Book Record feature enhances the efficiency and reliability of library transaction management.

Exit System

The Exit System feature enables users to safely close the Library Management System application. This function ensures that all ongoing transactions and records are properly saved before termination. It helps prevent data loss and protects the integrity of the database. Users can conveniently end their session after completing library operations. The feature also contributes to system security by preventing unauthorized access when the application is no longer in use. Proper system termination supports the stable performance of the software. Additionally, it provides users with a simple and efficient way to exit the application. Overall, the Exit System feature promotes safe and reliable system operation.

Search Function

The search module allows the user to find a book by title, author, or category.

Result: Search results displayed the correct book(s) matching the query. This made retrieving specific records faster and more accurate than scanning through manual logs.

Report Generation

The Report module generates summaries of book availability, borrowing, and return transactions.

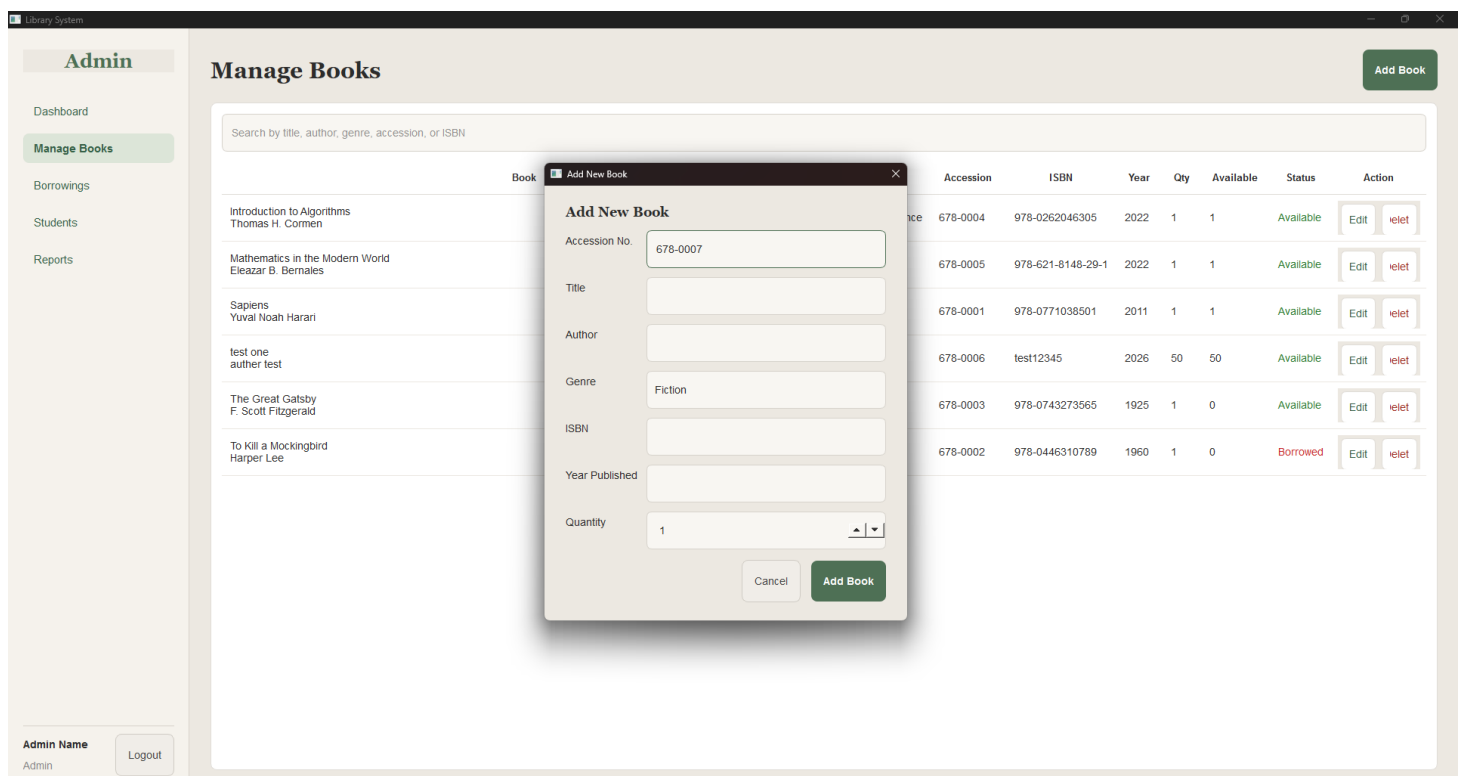
Result: Reports accurately reflected the stored database records, providing useful information for library management and decision-making.

System Interface

The Library Management System utilizes a sidebar-based navigation interface developed using Figma. The navigation panel is positioned on the left side of the application window, allowing users to conveniently access major system modules such as Dashboard, Manage Books, Borrowings, and Students. This layout improves accessibility, organization, and ease of navigation within the system. Additionally, the interface was designed to provide a clean, user-friendly, and efficient environment for managing library operations. The development and implementation of the Library Management System were successful. Student data and books, as well as borrowing records, can be managed efficiently using the system. It was found that the system is capable of accurately documenting transactions, storing, retrieving, and updating data. Through automation and by limiting mistakes on part of the human component, the system proved to increase both the accuracy and effectiveness of library processes. Monitoring and obtaining information through a unified database became easier. Overall, the objectives of the system in providing an efficient and reliable way of managing the resources were met.

Add Book Records

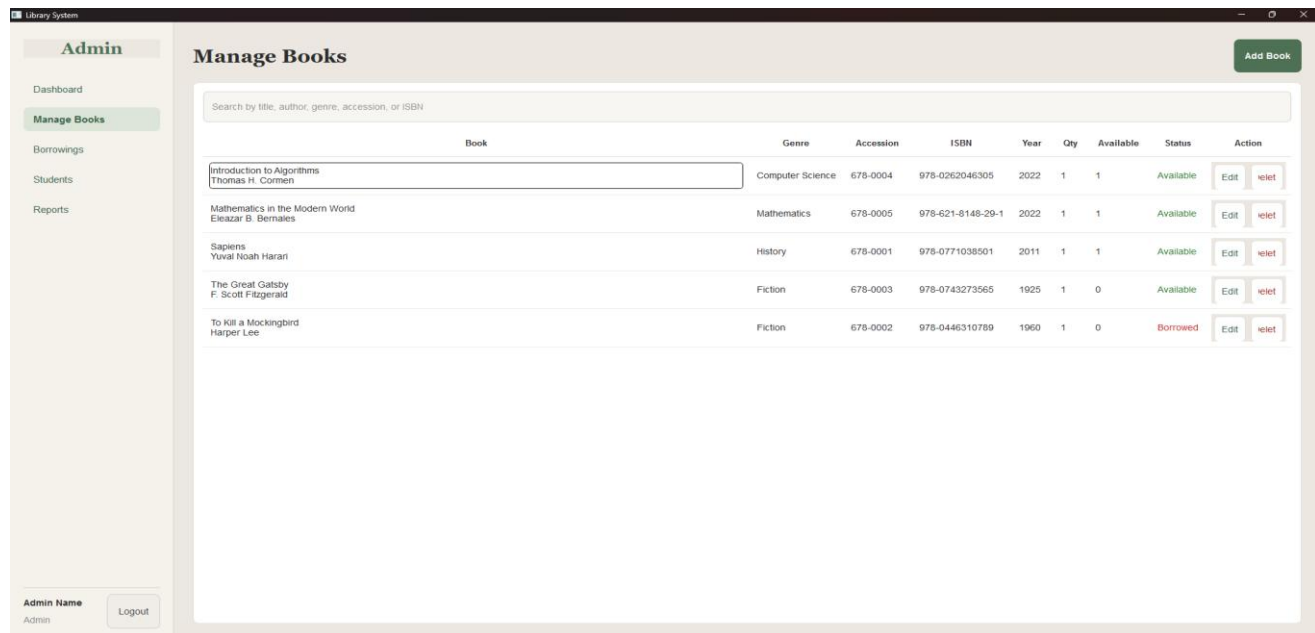
Figure 5. Add Book Interface of the Library Management System



This figure shows the Add Book interface of the Library Management System. The interface allows the administrator to input new book details such as accession number, title, author, genre, ISBN, and year published. This feature ensures accurate entry of book information into the database and updates the system inventory upon submission

View Records

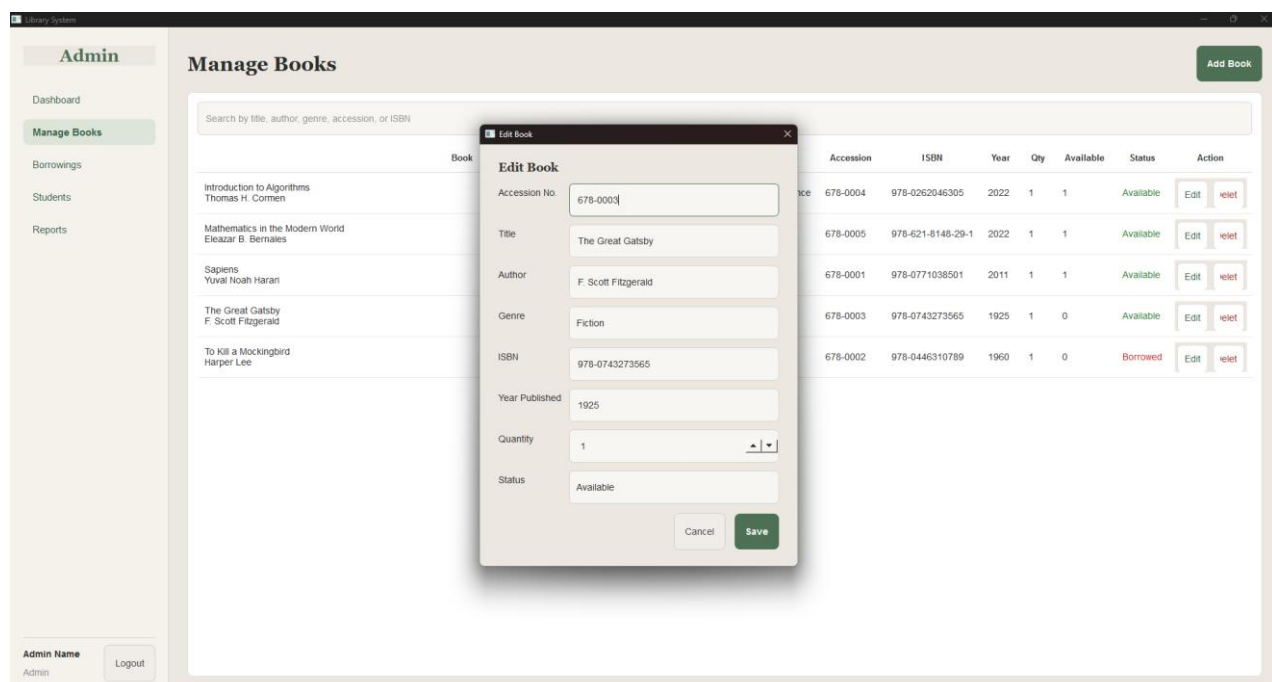
Figure 6. View Book Records Interface of the Library Management System



This figure shows the View Book Records interface of the Library Management System. It allows administrators to see all registered books in a structured table format, including details such as book title, author, year, ISBN, and status. This feature helps library staff monitor available and borrowed books efficiently.

Editing Book Record

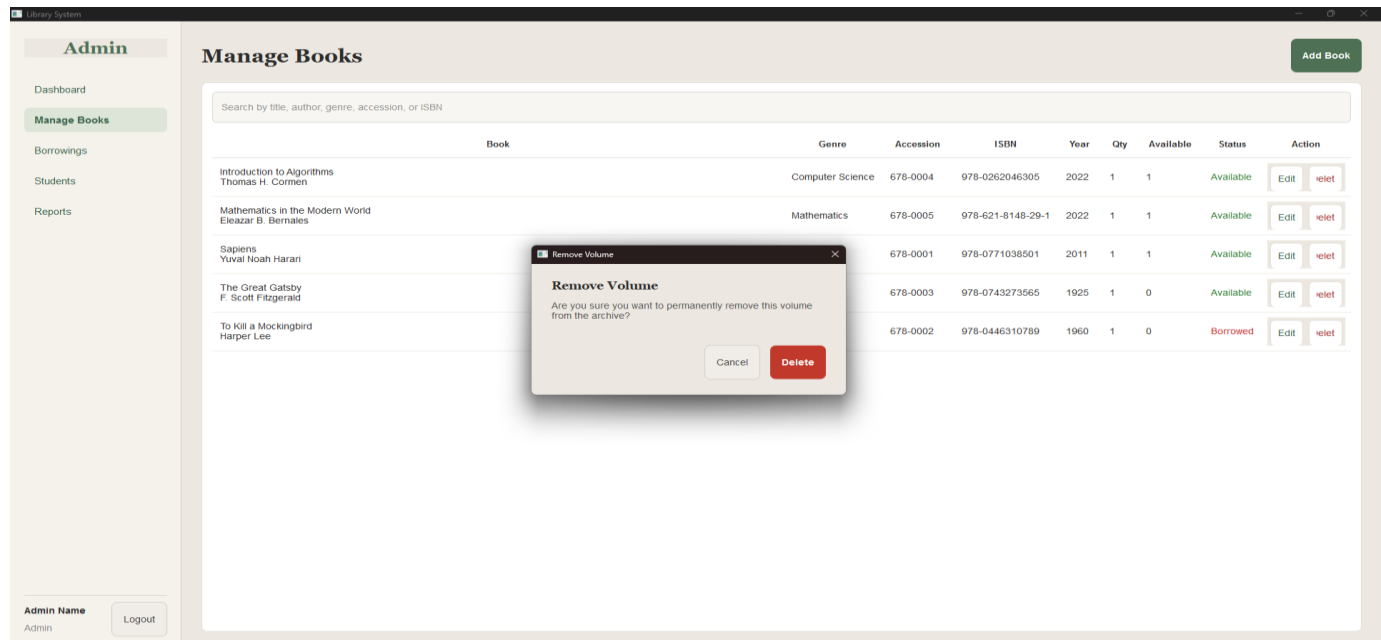
Figure 7. Edit Book Interface of the Library Management System



This figure shows the Edit Book interface of the Library Management System. It allows administrators to update the details of an existing book, including title, author, genre, ISBN, year published, and availability status. This feature ensures that book information in the database remains accurate and up-to-date.

Delete Book Record

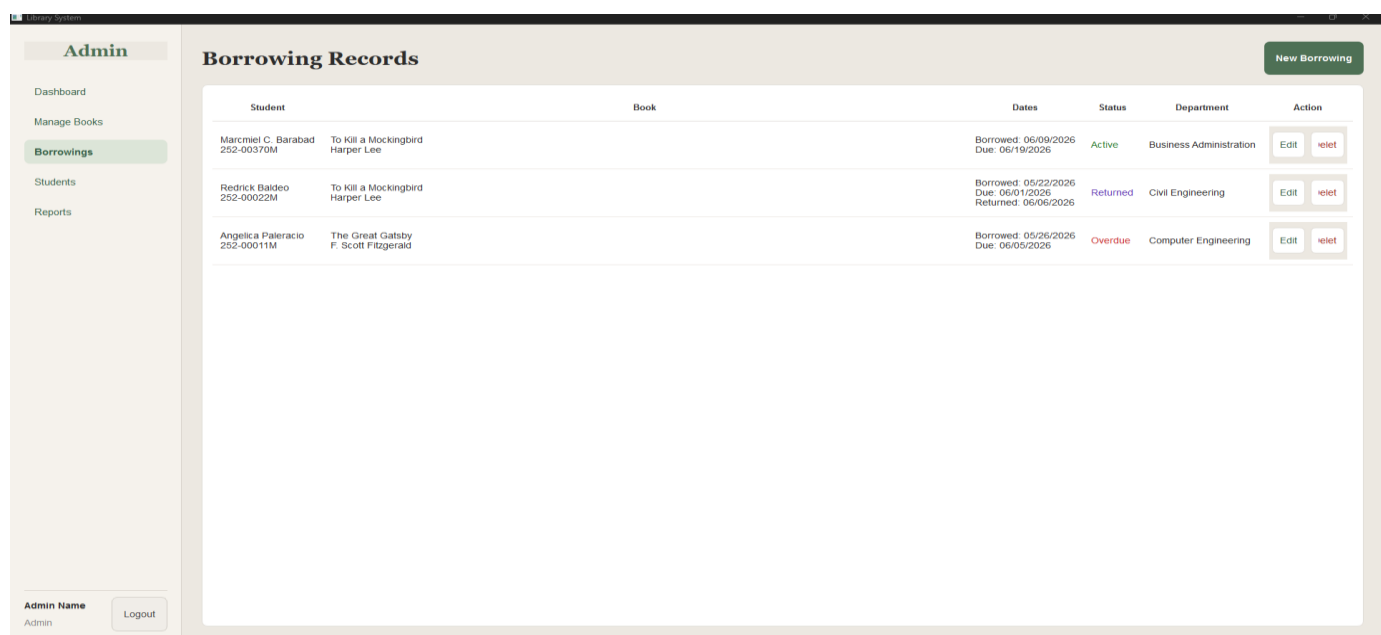
Figure 8. Delete Book Interface of the Library Management System



This figure shows the Delete Book interface, where the system prompts the administrator to confirm the permanent removal of a book record from the archive. This confirmation step ensures that deletions are intentional and prevents accidental loss of book data.

Borrow Book Record

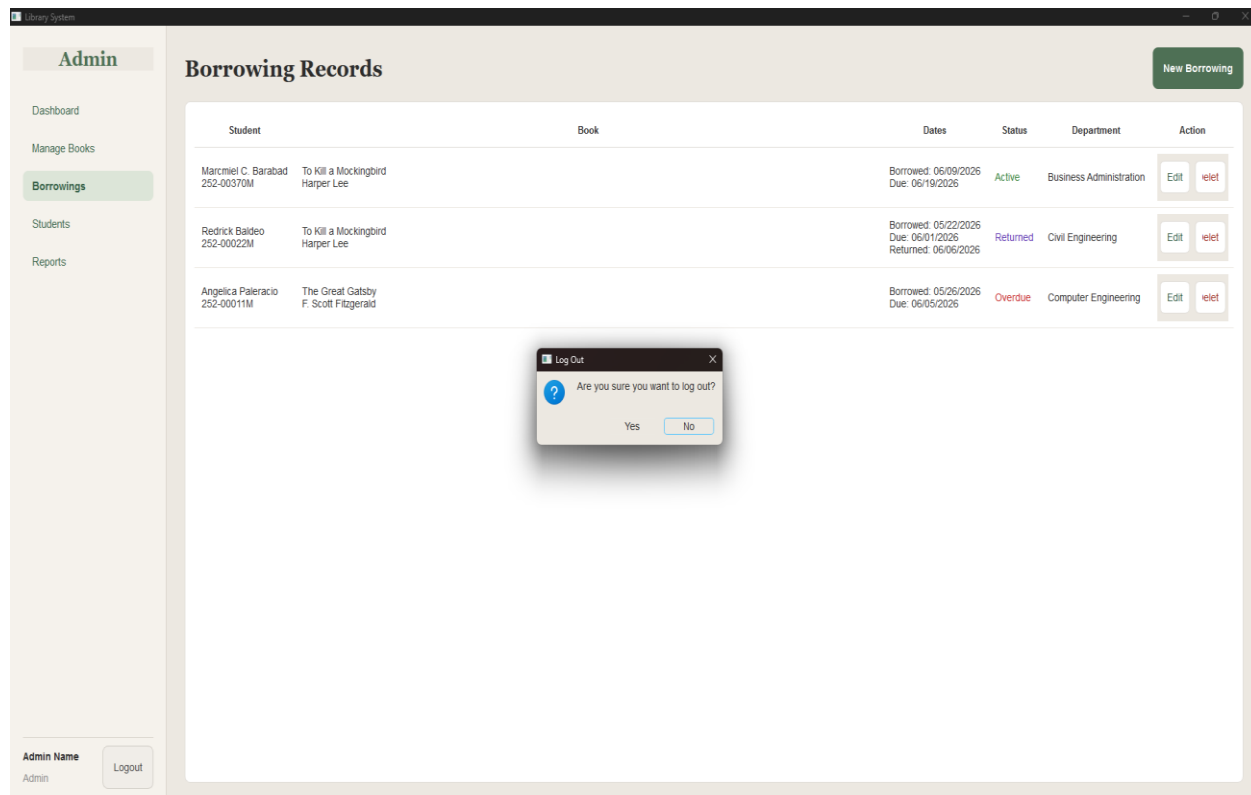
Figure 9. Borrow Book Interface of the Library Management System



This figure shows the Borrow Book interface where the administrator or library staff can record the borrowing transaction. It allows selecting a student and a book, automatically checks availability, and saves the borrowing details in the system. This interface helps track borrowed books and updates their status efficiently.

Exit System

Figure 10. Exit System Interface of the Library Management System



This figure shows the Exit System interface, which allows the administrator to safely close the Library Management System. It ensures that all data is saved, database connections are properly closed, and no records are lost during the exit process.

System Testing Results

Table 1.

Test Case	What Was Done	Expected Result	Actual Result	Status
Invalid Login Credentials	Entered wrong username or password	System should display "Invalid Username or Password"	System displayed error message	PASS
Empty Field (Login/Add Record)	Left a required field empty	System should prompt user to complete required fields	System displayed warning message	PASS
Add Record	Entered valid book details	Record should be saved in the database	Record saved successfully in the database	PASS

Delete Record	Removed a book record	Book should be removed from the database	Record successfully removed from the database	PASS
Search Record	Searched by title, author, or category	System should retrieve matching record(s)	Matching records displayed	PASS
Borrow Book	Borrowed a book	System should record borrowing details and update book status	Borrowing transaction recorded and book status updated	PASS
Return Book	Returned a borrowed book	Book availability should be updated	Return transaction recorded and book status updated	PASS
Invalid Input	Entered invalid characters or data type	System should display an error message	Error message displayed	PASS
Duplicate ID/ISBN Entry	Entered a book with existing ID/ISBN	System should show duplicate record warning	Duplicate record warning displayed	PASS
Borrow Unavailable Book	Attempted to borrow a book that is already borrowed	System should block transaction	Transaction blocked with warning	PASS
Edit Book Record	Updated existing book details	Changes should reflect in the database	Changes saved successfully	PASS
Generate Report	Requested a summary report	Report should match database records	Correct report generated	PASS

All test cases passed successfully, showing that the system performs its intended functions reliably.

DISCUSSION OF FINDINGS

The results demonstrate that the Library Management System successfully addressed issues inherent in manual record-keeping. The system allows library staff to manage book records efficiently, accurately track borrowed and returned books, and generate reports with minimal errors.

Database integration with SQLite ensures that records are saved permanently and organized in tables for efficient access. Input validation prevents incorrect entries, reducing errors in book and borrower records.

The GUI enhances usability by providing an intuitive interface that simplifies navigation and operation. Staff can perform key tasks such as adding, searching, editing, and deleting records without prior technical training.

Overall, the system improves library operations by increasing accuracy, reducing manual workload, improving record accessibility, and enabling quick retrieval of information for decision-making and reporting.

CONCLUSIONS AND RECOMMENDATIONS

Conclusions

Based on the development, testing, and evaluation of the **Library Management System**, the researchers concluded that the system successfully achieved its main objective of providing a functional, GUI-based desktop application to manage library book records and borrowing transactions efficiently.

The system effectively addressed the challenges of manual record-keeping, including misplaced records, delayed book searching, inaccurate borrower tracking, and difficulties in monitoring book availability. Through the use of Python, VS Code, and SQLite, the system provides an organized and reliable method for storing and retrieving book and borrower information.

The system successfully implemented key functionalities, including:

- Adding, viewing, editing, and deleting book records
- Borrowing and returning books with automatic status updates
- Searching for books by title, author, or category
- Generating transaction reports and summaries
- Input validation to prevent incorrect or incomplete entries

Functional testing confirmed that the system worked as intended. All test cases passed, demonstrating that the system reliably processed transactions, updated inventory status, handled invalid inputs, and preserved database integrity.

The graphical user interface contributed to system usability, allowing library staff to interact with the system intuitively without requiring technical knowledge of databases or programming. Overall, the system is a practical and user-friendly solution for improving the efficiency, accuracy, and organization of library record management.

Recommendations

Based on the findings and observed limitations, the following recommendations are proposed to improve the Library Management System:

1. **User Authentication:** Implement user login and role-based access to differentiate between administrators, library staff, and general users for enhanced security.
2. **Barcode Integration:** Integrate barcode scanning to speed up book check-in and check-out processes.
3. **Report Enhancements:** Include advanced reporting features such as daily, weekly, and monthly summaries, with export options to PDF or Excel.
4. **Overdue Tracking:** Add a module to track overdue books and generate notifications or reminders for borrowers.
5. **Cloud Backup:** Implement cloud-based data backup to prevent data loss and allow remote access to records.

6. **Multi-User Support:** Enable multi-user network access so that several library staff members can operate the system simultaneously.
7. **Mobile Accessibility:** Develop a mobile version or responsive interface for easier access in libraries with limited desktop stations.
8. **User Interface Improvements:** Enhance the GUI design with modern themes, clearer navigation, and visual feedback for actions.
9. **Advanced Search:** Implement filtering and sorting features to allow searches by multiple criteria, such as availability status, category, or borrower.
10. **Usability Testing:** Conduct further usability studies with actual library staff to gather feedback and refine system features for real-world application.

In conclusion, while the Library Management System currently meets its intended goals and performs all essential functions, future enhancements could improve usability, accessibility, security, and reporting, making it a more robust and comprehensive tool for modern library management.

ACKNOWLEDGEMENT

The researchers would like to express their heartfelt appreciation to **Engr. Meshelle N. Fabro, PCpE**, for her dedicated mentorship, professional guidance, and expert supervision throughout the completion of this study. Her valuable insights, constructive suggestions, and continuous encouragement significantly contributed to the successful development and improvement of this research.

The researchers also extend their sincere gratitude to the **Eulogio “Amang” Rodriguez Institute of Science and Technology (EARIST)**, particularly the **Department of Computer Engineering**, together with its faculty members, for providing the academic knowledge, technical resources, and institutional support necessary for the accomplishment of this project. The institution’s commitment to excellence in education, innovation, and research played an essential role in the success of this study.

Special thanks are also given to the families and friends of the researchers for their unwavering support, patience, understanding, and motivation. Their encouragement served as a source of inspiration and strength throughout the challenges encountered during the research process.

Above all, the researchers offer their deepest gratitude to **Almighty God** for the wisdom, guidance, strength, and perseverance bestowed upon them throughout this endeavor. This work is humbly dedicated to His glory, acknowledging that every achievement was made possible through His grace and blessings.

REFERENCES

1. Ariyanti, A., & Supriyanto, W. (2023). Optimizing user interface and user experience: Exploring design improvements for the school library system. *International Journal of Scientific and Academic Research*, 5(2), 45–53. <https://ijsar.net/index.php/ijsar/article/view/144>
2. Dagogo, D. (2024). Library automation in the 21st century: Challenges and prospects. *Information Managers: A Journal of Nigerian Library Association, Rivers State Chapter*. <https://doi.org/10.61955/liaukn>
3. Eldaniz, M., & Dashqin, M. (2025). Modern problems of library and information resources management. Zenodo (CERN European Organization for Nuclear Research). <https://doi.org/10.5281/zenodo.17187844>
4. GeeksforGeeks. (2025, July 11). Introduction to SQLite. <https://www.geeksforgeeks.org/introduction-to-sqlite/>

5. Jaiswal, G. K., & Pandey, P. K. (2024). Assessing the challenges and issues of implementing library automation in library: A study. *ShodhKosh Journal of Visual and Performing Arts*, 5(6). <https://doi.org/10.29121/shodhkosh.v5.i6.2024.5238>
6. Kumar, R., & Singh, P. (2022). Design and development of a web-based library management information system. *International Conference on Business Information Systems*. <https://icobits.ubhinus.ac.id/index.php/ICoBITS/article/view/81>
7. Lathifuddin, A. S., Zakariya, A., Rahmawan, A. G., Saragih, D., & Bekti, F. K. (2026). Design and development of a web-based library management information system. *ICoBITS*, 1, 701–704. <https://doi.org/10.32664/icobits.v1.81>
8. Lwoga, E. T., & Sangeda, R. Z. (2021). The impact of usability, functionality and reliability on users' satisfaction during library system adoption. *Tanzania Journal of Information Technology*, 3(1), 15–27. <https://journals.iaa.ac.tz/index.php/tji/article/view/13>
9. Mardiyati, S., Pauziah, U., & Sutrisno, D. (2025). Design and creation of book inventory management system in school library. *RIGGS Journal of Artificial Intelligence and Digital Business*, 4(2), 5524–5530. <https://doi.org/10.31004/riggs.v4i2.1405>
10. Nurfadillah, S., & Rahman, A. (2023). Usability testing on Online Public Access Catalog (OPAC) interface of SLiMS 8.0 Akasia in BNPB library. *Jurnal Pustakawan Indonesia*, 22(1), 34–44. <https://journal.ipb.ac.id/index.php/jpi/article/view/25673>
11. Rijal, M. A. (2024). Customization of KOHA integrated library system (ILS) interface for better usability. *Access: An International Journal of Nepal Library Association*, 3(1), 55–67. <https://www.nepjol.info/index.php/access/article/view/59004>
12. Siregar, F., & Wahyuni, D. (2023). Analysis of user interface and user experience on Senayan Library Management System (SLiMS). *Lentera Pustaka*, 9(1), 21–30. <https://journal.uinmataram.ac.id/index.php/light/article/view/8805>
13. Singh, P. (2024). Library management system. *Gurukul Multidisciplinary Research Journal*, 407–417. <https://doi.org/10.69758/jubi7297>
14. Urwin, M. (2025, April 24). What is SQLite?. <https://builtin.com/learn/sqlite>
15. Valdez, J. P. (2022). Usability evaluation of library management system: A user experience approach. *Library Philosophy and Practice*. <https://scholarworks.sjsu.edu/libphilprac/8165>

ABOUT THE AUTHORS

Redrick John C. Baldeo is a Computer Engineering (CPE) student at the Eulogio “Amang” Rodriguez Institute of Science and Technology and a member of the Library Management System development team. He contributed to the design and development of the system by implementing Python-based functionalities, assisting in database management, participating in system testing, and helping improve the system's overall performance and usability. Passionate about technology and innovation, he continuously seeks opportunities to expand his knowledge in software development, database systems, and computer engineering. He is particularly focused on enhancing his problem-solving and teamwork skills, recognizing their importance in developing effective solutions and collaborating successfully in technical projects. Through dedication, continuous learning, and hands-on experience, he aspires to become a competent, innovative, and successful computer engineer in the future.

Marcmiel C. Barabad is a Computer Engineering (CPE) student at Eulogio “Amang” Rodriguez Institute of Science and Technology (EARIST) and a member of the Library System Management development team. He contributed to maintaining the project's structure and ensuring the overall functionality of the Library Management System.

Driven by his goal of becoming a Network Engineer, he is committed to continuously expanding his knowledge of networking, mechatronics, and computer systems. Despite the challenges he faces in studying mathematics, he remains focused and determined to master its concepts, recognizing its importance in developing logical and analytical thinking. He believes that wisdom is gained through experience and that failure is an essential part of growth and learning.

He is also a TESDA ICT-CSS NC II certificate holder, reflecting his dedication and passion into developing practical technical skills and immersing himself in the field of information technology.

Angel C. Caubang is a Computer Engineering (CPE) student and one of the developers of the Library Management System. As a recipient of a Research Award in the Information and Communications Technology (ICT) strand during senior high school, she developed a strong foundation in technology and innovation. Driven by her passion for computing, She continues to pursue advanced knowledge and skills in the field through her Computer Engineering studies. Her contributions to the project include system development, testing, and documentation, helping ensure the successful implementation of the Library Management System.

Squall Leonhart B. Hilario is a Computer Engineering (CPE) student in Eulogio "Amang" Rodriguez Institute of Science and Technology. He is involved in the creation of the Library Management System. He contributed to software development, troubleshooting, and system enhancement, ensuring that the application met the project's objectives. Driven to develop his knowledge and practical skills, he continuously strives to improve his competence in the field of computer engineering. He has a strong interest in understanding how technological devices operate and often examines and disassembles them to gain deeper insights into their functions. With his passion for learning and determination, he aims to build a successful career in computer engineering.

Richmond O. Jatol is a Computer Engineering (CPE) student at the Eulogio "Amang" Rodriguez Institute of Science and Technology (EARIST) and a member of the Library Management System development team, he contributed to the project by designing, developing, and managing the system's database, helping enhance its organization, efficiency, and overall functionality. Passionate about technology and system development, he continuously works to expand his knowledge and technical expertise in computer engineering. Despite academic challenges, he remains dedicated to strengthening his analytical thinking, problem-solving, and programming skills. Through perseverance, continuous learning, and self-improvement, he aims to become a competent and accomplished computer engineer capable of developing innovative solutions that contribute to the advancement of technology.

Angelica D. Paleracio is a Computer Engineering (CPE) student who contributed to the development of the Library Management System. She participated in project planning, documentation, and system evaluation, helping ensure the effectiveness and quality of the final product.

Engr. Meshelle N. Fabro is a licensed Professional Computer Engineer and educator at the Eulogio "Amang" Rodriguez Institute of Science and Technology (EARIST). She is currently pursuing a Master of Science in Computer Engineering at Bulacan State University and has professional industry experience as a Technical Engineer at Hewlett-Packard (HP) and an IT Engineer at IBM. She is also a TESDA CSS NC II and Trainers Methodology Level I (TM1) holder, demonstrating her commitment to technical excellence and competency-based education. Passionate about teaching and innovation, she is dedicated to guiding Computer Engineering students in becoming industry-ready, competent, and highly skilled individuals equipped for the evolving demands of the technology industry.